UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

ANALIZA REPLIKACIJE NA PRIMERU SPLETNEGA PORTALA NLB

Ljubljana, maj 2004 ROK LAVRIČ

IZJAVA

Študent Rok Lavrič izjavljam, da sem avtor tega magistrskega dela, ki sem ga napisal pod mentorstvom doc. dr. Indihar Štemberger Mojce in skladno s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah dovolim objavo magistrskega dela na fakultetnih spletnih straneh.

V Ljubljani, dne 14.05.2004

Podpis:_____

KAZALO

KAZALO SLIK

KAZALO TABEL

KAZALO PRILOG

1. UVOD

1.1 Uvod v magistrsko delo

Razmere na trgu se pogosto zelo hitro spreminjajo. Temu se mora prilagoditi tudi poslovno okolje, če hoče pravočasno poiskati odgovore na nove izzive, ki nam ne prizanašajo pri sprejemanju hitrih in pogosto tveganih odločitvah. Spremenjeni poslovni viziji in strategiji se morajo prilagoditi tudi poslovni procesi (Kovačič, 1998, str. 37). Upravljavski center za informacijsko tehnologijo v podjetju mora slediti spremembam in potrebam poslovnih procesov ter zniževati stroške in čas za njihovo izvajanje.

Zaposleni v Upravljavskem centru za informacijsko tehnologijo v NLB za potrebe novih ali prenovljenih poslovnih procesov prirejamo, popravljamo, ustvarjamo programsko in vzdržujemo strojno opremo. Veliko pripomorejo že vgrajene možnosti, ki jih nudi proizvajalec programske ali strojne opreme. Če te možnosti dobro poznamo, jih lahko v kritičnih trenutkih pomembno izrabimo v svoj prid. Tako lahko pridemo do cenejših in zanesljivejših rešitev, predvsem pa so le-te hitrejše.

Proizvajalci programske opreme za svoj produkt razvijejo različna orodja, ki nam omogočajo boljši vpogled in nadzor nad posameznimi procesi v aplikaciji. Nekatera lahko dobimo že ob nakupu programskega paketa, druga pa se nam ponujajo kot doplačilo. Seveda je količina pridobljenih orodij v osnovnem paketu (brez doplačila) odvisna tudi od naših pogajalskih sposobnosti ob nakupu.

Sestavni del Sistema za upravljanje baz podatkov (SUBP) priznanih proizvajalcev so tudi procesi, ki podpirajo izvedbo in delovanje podatkovne replikacije, to je podvajanje podatkov na isti ali drugih SUBP. Za aktiviranje teh procesov moramo imeti potrebno tehnično zanje in (še bolj pomembno) poznati področja in namen replikacije, da nam ta lahko koristi.

Replikacijo lahko uporabljamo na različne načine in si tako z ostalimi produkti prikrojimo okolje, ki najbolje odgovarja našim poslovnim zahtevam. V osnovi služi podvajanju podatkov. Orodje lahko izkoristimo za:

- hranjenje vrednejših podatkov (v primeru odpovedi sistema omogočimo hitri preklop na drugi delujoči sistem z majhno verjetnostjo izgube podatkov),
- izgradnjo arhitekture, ki nam v nekaterih primerih ne omogoča ene centralne podatkovne baze (zato potrebujemo sinhronizacijo podatkov med njimi),

- optimizacijo (podatke podvojimo in omogočimo hitro iskanje po podvojenih tabelah, medtem ko podatke spreminjamo in vnašamo samo v glavno tabelo),
- prenos podatkov na podsisteme (podatki se lahko obdelujejo centralno, uporabniki pa dostopajo samo lokalno, preko aplikacij do pristojnih področij),

Mnogo informatikov ne pozna te možnosti ali pa je ne znajo primerno izkoristiti za podporo poslovnim procesom v podjetjih.

Cilj magistrskega dela je seznanitev z replikacijo in možnostmi, ki jih le-ta ponuja. Namen analize je ugotoviti primernost metode replikacije na konkretnem primeru spletnega portala Nove Ljubljanske banke d.d. (v nadaljevanju NLB) in ugotovitev ustreznosti izbranega procesa. Pri oceni sem preko podrobnejše analize tehnične realizacije ovrednotil različne poglede s strani uporabnikov, skrbnikov ter podjetja. Zbrane rezultate sem kritično ovrednotil in na koncu podal oceno ustreznosti.

Predstavitev svojega magistrskega dela sem razdelil na štiri glavna poglavja z ustreznimi podpoglavji.

Jedro naloge je prestavitev podatkov na lokalno bazo spletnega portala. Zato sem najprej predstavil rabo in pomen Interneta v Sloveniji, podal nekaj splošnih informacij in pojasnil vzroke za projekt prenove spletnega portala NLB.

V banki uporabljamo veliko različnih računalniških sistemov in komponent, ki bolj ali manj podpirajo vse veje poslovanja. Zato sem se v drugem delu lotil razlage sistemov, ki jih uporabljajo v procesu snovanja, in predstavitve spletnega portala. Tako sem omogočil lažjo razumljivost posameznih procesov in rešitev v nadaljevanju naloge.

V tretjem delu sem opisal replikacijo. Navedel sem splošne značilnosti postopka in nadaljeval s predstavitvijo konkretne rešitve na spletnem portalu NLB. Na koncu sem prikazal konkretno rešitev, ki jo uporabljamo.

Analizo sem opravil v četrtem delu. Ker je zaščita podatkov zelo pomemben dejavnik, sem opisal vidike varnosti in ranljivosti sistema. Pri tem sem se osredotočil predvsem na arhitekturo replikacije in, zaradi zaščite podatkov, delno tudi na rešitev banke. V tem delu sem tudi ocenil, v kolikšni meri je rešitev izpolnila pričakovanja. Na koncu sem podal sklep z možnimi izboljšavami, v kolikor bi bile potrebne.

V nalogi sem uporabil teoretično in praktično znanje, ki sem ga pridobil pri študiju na Ekonomski fakulteti, na izobraževanjih in z delom v podjetju NLB. V poglobljenih razlagah sem se naslonil na tehnično dokumentacijo, članke in ostalo strokovno literaturo priznanih avtorjev. Pri pregledu gradiva sem upošteval več različnih SUBP različnih proizvajalcev, saj je osnovni koncept replikacije povsod enak. Ker je večji

del literature v angleškem jeziku, sem za izvirne angleške izraze našel in tudi navedel ustrezne slovenske prevode.

Naj na koncu uvoda dodam, da se bosta besedi replikacija in propagacija vseskozi navezovali na temo podatkovnih baz, kot specifična procesa za upravljanje podatkov. Vsekakršne druge interpretacije so izključene.

1.2 Predstavitev podjetja NLB d.d.

Zgodovina bančništva na področju Slovenije se prične leta 1862, ko so ustanovljene prve slovenske hranilnice - leta 1862 v Mariboru, leta 1865 v Celju in leta 1889 Mestna hranilnica ljubljanska kot predhodnica NLB, Ljubljana.

Organizacijsko in strukturno se začne razvijati leta 1955, ko se ustanovi Komunalna banka Ljubljana. Od takrat naprej se pridruži še več drugih bank, začno poslovati s tujino, odpirajo tuja predstavništva v tujini in tudi po vsej takratni državi.

Leta 1990 se preobrazi bančna skupina Ljubljanske banke v delniške družbe. Skladno z novo bančno zakonodajo bančna skupina Ljubljanske banke formalno preneha delovati. Članice skupine nadaljujejo poslovanje popolnoma samostojno, ohranjene pa so še določene kapitalske in poslovne povezave.

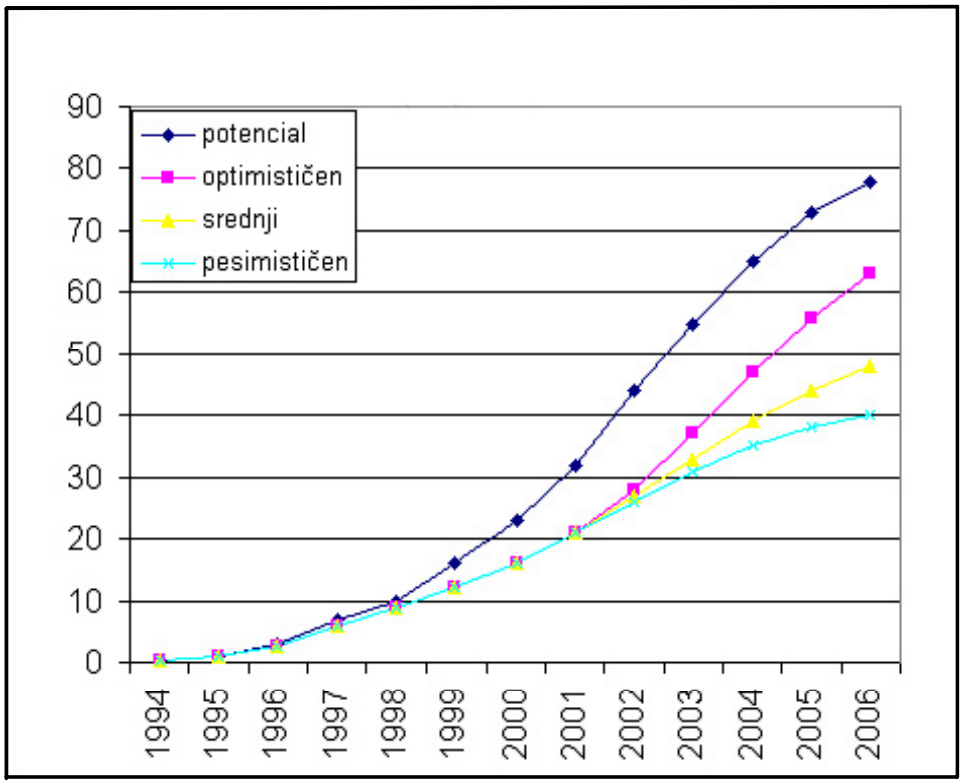
Nato se začne proces sanacije Ljubljanske banke d.d., Ljubljana - prenos slabih terjatev na Agencijo za sanacijo bank in hranilnic. 16. julija 1997 se uradno zaključi proces sanacije NLB. Banki podelijo ocene naložbenega tveganja štiri svetovno priznane rating agencije. Ocene NLB so med najvišjimi ocenami bank v srednji in vzhodni Evropi.

NLB je delniška družba in je kot taka univerzalna banka z licenco Banke Slovenije za opravljanje vseh bančnih poslov v Sloveniji in tujini. NLB opravlja dejavnosti poslovnega bančništva, investicijskega bančništva in varčevanja. S svojo razvejeno poslovno mrežo nudi storitve doma in v tujini za individualne stranke, javne ustanove, obrtnike ter mala, srednja in velika podjetja. Preko svojih hčerinskih podjetij upravlja tudi investicijske sklade, se ukvarja z nepremičninami, opravlja lizinške storitve, posle izvoznega faktoringa in finančnega svetovanja. NLB je največja slovenska banka, saj ima največje tržne deleže na ključnih področjih poslovanja. Trenutno zaposluje preko 5000 ljudi na vseh področjih organizacijske strukture.

2. SPLETNI PORTAL

Internet je zaradi svoje dostopnosti in velike količine podatkov prava meka za iskalce informacij. In teh vsekakor ni malo. Statistika (na sliki 1) nazorno prikazuje rast rabe interneta in projekcijo do konca leta 2006 v Sloveniji. Pri tem je zanimivo, da je odstotek obiskov v razmerju med tujimi in domačimi stranmi v korist slednjih. To je močna spodbuda za podjetja, ki so "zaspala" in šele sedaj prihajajo na spletni trg.

Slika 1: Delež uporabnikov v celotni populaciji - projekcije



Vir: Indikatorji interneta, 2002.

Slovenija na področju rabe interneta za evropskimi državami zaostaja za dve leti. Kljub temu pa so slovenska podjetja sorazmerno dobro opremljena z računalniki in dostopom do svetovnega spleta, saj ga zaposlenim omogoča 90 odstotkov podjetij. Več kot 40 odstotkov zaposlenih v naši državi uporablja računalnik, več kot 33 odstotkov pa tudi internet. V Sloveniji imamo na 100.000 prebivalcev 2.000 računalnikov, ki imajo dostop do interneta. V letu 2003 je tako bilo na spletu okrog 30.000 spletnih mest in štiri milijone dokumentov (Slovenija glede rabe interneta za dve leti zaostaja za Evropo, 2004). Velja zelo groba ocena, da je Slovenija svoj

zaostanek za Evropsko Unijo nekoliko zmanjšala. Sedaj znaša odstotek rabe interneta v Sloveniji 47%, medtem ko v Evropski Uniji dosega 56,5% (Uporaba interneta: Gospodinjstva, 2003).

V letu 2002 je število uporabnikov interneta naraščalo po najbolj optimističnem scenariju. Napoved o 540.000 uporabnikih, kar naj bi po ocenah srednjega scenarija iz junija 2001 Slovenija dosegla šele v juniju 2002, je bila namreč presežena že v marcu 2002. Če se bo Slovenija še naprej gibala po optimističnem scenariju, bo junija 2006 več kot 1.2 milijona uporabnikov interneta in ne samo 960.000, kot je bilo po srednjem scenariju predvideno v projekcijah iz leta 2001 (Indikatorji interneta, 2002). Potreba podjetij po predstavitvi svojih uslug, proizvodov in storitev na internetu je tako očitna.

Za poslovanje prek interneta se odloča tudi vse več slovenskih podjetij. Raziskava, ki so jo v okviru projekta "Raba Interneta v Sloveniji" naredili s pomočjo Fakultete za management med 750. slovenskimi podjetji, kaže, da jih dve tretjini že uporablja elektronsko poslovanje, s katerim želijo skrajšati in izboljšati kakovost notranjih in zunanjih procesov (Slovenija glede rabe interneta za dve leti zaostaja za Evropo, 2004).

Uporabniki se srečujemo s portali različnih grafičnih podob in preglednosti. Zelo cenimo sprotnost predstavljenih podatkov in posredovanje odgovorov na morebitna zastavljena vprašanja. Dobra spletna mesta pritegnejo veliko ljudi.

2.1 Spletni portal Nove Ljubljanske banke d.d.

Novi NLB portal je zaživel oktobra 2002. Požel je zelo pozitivne kritike in prejel priznano nagrado Netko, ki so jo podelili ocenjevalci Gospodarskega vestnika med 180 prijavljenimi na razpis za najboljšo predstavitev na spletnem portalu. Trenutno obsega že približno 6000 dokumentov. Ta številka še ni končna, saj se neprestano širi z novimi vsebinami.

Strani spletnega portala so vsebinsko bogate in pregledne (slika 2). Informacije, ki jih uporabniki bančnih storitev potrebujejo, so preprosto dostopne. Enostavna, vendar lična grafika uporabnikom ne otežuje brskanja po straneh. Ponuja več oblik elektronskega poslovanja za pravne in fizične osebe. Zagotavlja spoštovanje zasebnosti obiskovalca, omogoča hitro nalaganje strani in preglednost (Predstavitev banke na internetu, 2002).

Slika 2: Preglednost spletnega portala

Vir: NLB statistika po predstavitvi Simone Muc "Spletni portal NLB – Med strategijo in izvedbo", 2003.

NLB se je odločila za prenovo spletnega portala, ker želi slediti sodobnim trendom na tem področju ter zagotoviti obiskovalcem portala enostavno uporabo. Stranke želi informirati s pestro vsebino in omogočiti neposredno povezavo ponudbe z opravljanjem storitev banke. Obiskovalec portala NLB lahko na njem pridobi vrsto dnevno pomembnih informacij ali se naroči na prejem tečajnih list in elektronskih novic. Poleg tega lahko opravi informativni izračun za najem posojila, varčevanje, menjavo valute, ali na zemljevidu poslovne mreže banke poišče sebi najbližjo poslovalnico ...

Spletni portal NLB je dopolnjen z vrsto interaktivnih orodij, s katerimi nam olajša pot do najboljše bančne rešitve. Vsebinsko je razdeljen na štiri sklope (Predstavitev banke na internetu, 2002):

- Osebne finance – področje je namenjeno prebivalstvu. Vsebuje predstavitve storitev in možnih poti do banke ter centre za posamezne segmente za otroke in odrasle. Do storitve nam pomaga virtualni bančni svetovalec in ostala orodja.
- Poslovne finance - podjetniške strani so ravno tako sestavljene iz centrov (za zasebnike, mala podjetja ter velika in srednja podjetja) in iz predstavitve storitev. Tudi tukaj nam svetovalec pomaga izbrati pravo storitev, s finančnim računalom pa bomo hitro izračunali, katera storitev je za naše potrebe najprimernejša.
- O banki - področje je namenjeno predstavitvi Nove Ljubljanske banke. V njem bodo koristne informacije našli novinarji, iskalci zaposlitve in tudi vsi ostali, ki jih zanima največja slovenska banka.
- O skupini NLB – predstavitev zajema Bančne skupine Nove Ljubljanske banke, shemo, lastniške deleže in osnovne podatke o posameznih povezanih družbah.

Internet se pojavlja kot nov, vsestranski medij, ki odpira vedno nove neslutene možnosti v odnosu (komunikaciji) med posamezniki in podjetji. Zavedanje le-tega pomeni potrebo po prisotnost podjetja v njem. Predstavitve storitev in izdelkov uspešno dopolnjujejo tradicionalne metode (brošure, letake, plakate, oglaševanje preko televizije, radia…). Spletna predstavitev NLB in njenih storitev se uspešno sklada s celovito strategijo trženja bančne ponudbe.

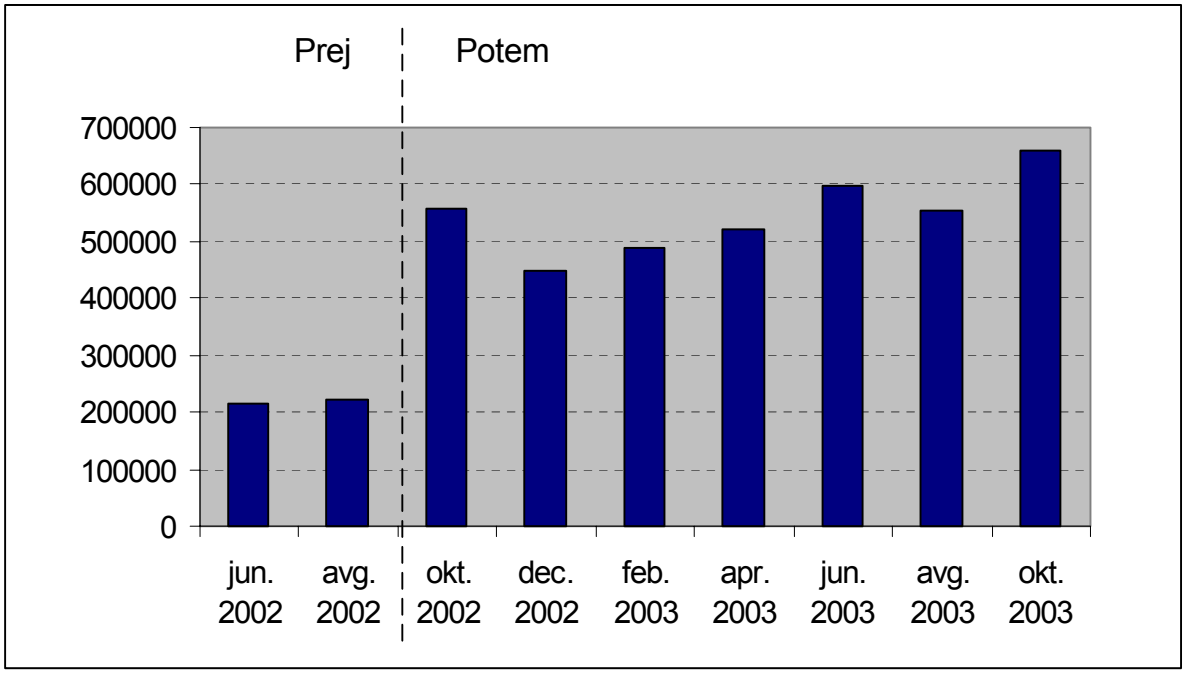
Internet ponuja hitreje in spretneje. Ljudje sami brskajo in pridobivajo informacije brez sodelovanja uslužbencev, kar prihrani in usmerja energijo zaposlenih v nove proizvode in storitve. Internet predstavlja konkurenčno prednost in priložnost, ki jo nudi informacijska doba.

2.1.1 Zakaj novi spletni portal Nove Ljubljanske banke ?

Prvi spletni portal NLB se je odprl v svet 1994. leta. Takrat je bila NLB ena prvih bank, ki je predstavila svoje delovanje na internetu. Prvo leto so bile spletne strani samo v angleškem jeziku, leto kasneje pa se jim je pridružila še slovenska različica. Večjo prenovo so doživele 1997. leta. Omogočal je nekaj osnovnih funkcionalnosti, ki pa sčasoma v obstoječi konfiguraciji niso bile več nadgradljive. Boljša programska orodja so omogočala izdelavo oblikovno sodobnejših in preglednejših spletnih portalov. Tako je prišlo do odločitve za izdelavo popolnoma novega in z zunanjo podobo NLB skladnega portala.

Cilj NLB je bil postaviti sodoben spletni portal, ki bo povečal ugled banke, pospešil prodajo storitev in bil prijazen za uporabo. Portal naj bi uporabnikom nudil enostaven dostop do informacij NLB. Obiskovalce spletnih strani bi spremljali ter jim poleg splošnih informacij ponudili tudi osebno svetovanje. Mnenje odgovornih je, da so bili cilji doseženi.

Slika 3: Obiskanost spletnih strani NLB pred in po uvedbi novega portala



Vir: NLB statistika po predstavitvi Simone Muc "Spletni portal NLB – Med strategijo in izvedbo", 2003.

Vodstvo je od vsega začetka podpiralo strategijo razvoja novega internetnega portala in je bilo na koncu tudi zadovoljno s končnim izdelkom. Če sodimo po obisku, so zadovoljnejši tudi uporabniki. Obisk se je namreč povečal za več kot 100%, kar prikazuje slika 3. Najbolj obiskane strani so tečajne liste, posojila za nepremičnine, osebna posojila, informacije o Kliku in tolarski depoziti.

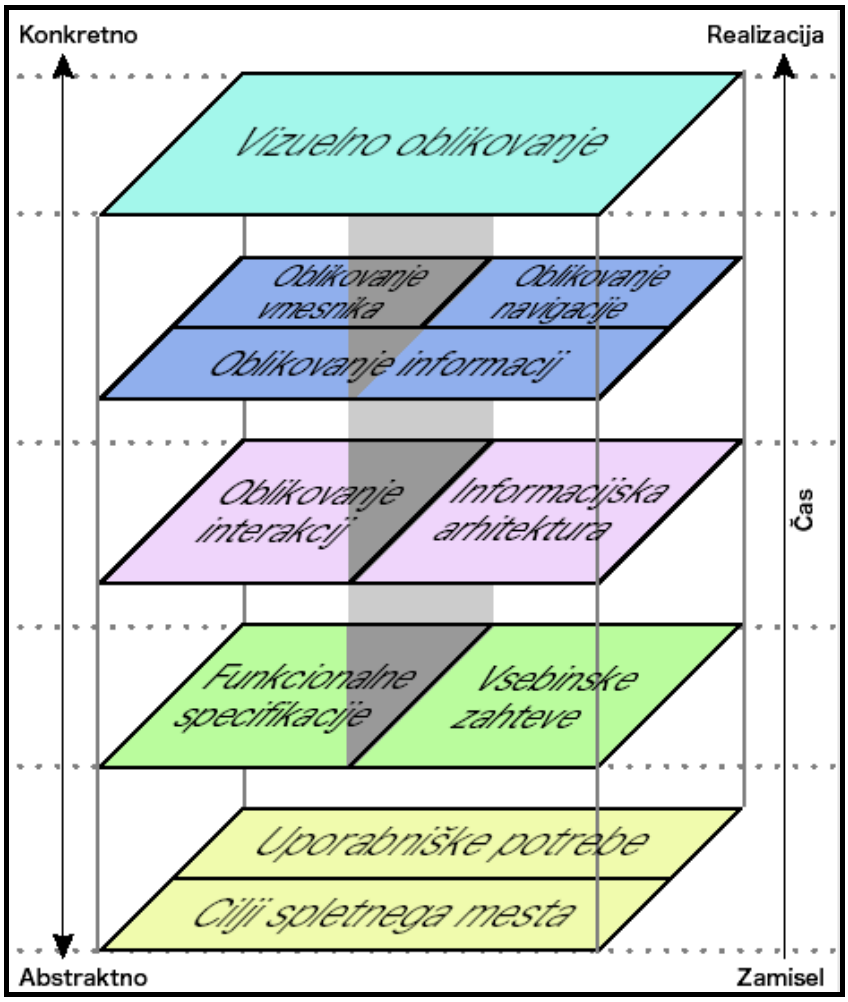
2.1.2 Postopki od zamisli do uresničitve izgradnje portala

Pri določanju osnovnih smernic projekta "Izgradnja bančnega portala NLB" so se projektni vodje naslonili na model Jesse James Garretta, ki ga prikazuje slika 4. Seveda vse ni potekalo popolnoma po načrtu, zato je prišlo do odstopanj in popravkov v posameznih fazah, kar bo razvidno v spodnjih odstavkih.

Strani slike 4 se nanašata na:

- leva stran - splet kot uporabniški vmesnik (naravnanost na naloge),
- desna stran - splet kot hipertekstualni sistem (naravnanost na informacije).

Slika 4: Elementi uporabniške izkušnje



Vir: Elementi uporabniške izkušnje, 2000.

1. faza

Uporabniške potrebe: od zunaj opredeljena namembnost spletenega mesta, ugotovljena skozi raziskave uporabnikov.

Cilji spletnega mesta: poslovni, kreativni in od znotraj opredeljeni cilji spletnega mesta.

- julij 2001 - postavljeni so bili prvi temelji novemu spletnemu portalu NLB. Določi se načina dela na projektu "Izgradnje bančnega portala NLB"

2. faza

Funkcionalne specifikacije: nabor funkcij - natančni opisi funkcij, ki jih spletno mesto mora imeti, če hoče zadovoljiti potrebam uporabnikov.

Vsebinske zahteve: definicija vsebinskih elementov, ki jih spletno mesto rabi, če hoče zadovoljiti potrebam uporabnikov.

- avgust 2001 – določi se vsebina portala
- december 2001 – potrjena vsebinska zasnova portala
- maj 2002 – vsebinski popravki
- julij 2002 – dodatne funkcionalnosti na področju osebnih in poslovnih financ
- oktober 2002 – spremembe zaradi odprave vsebinskih in funkcionalnih napak

3. faza

Oblikovanje interakcij: razvoj toka aplikacij, ki omogočajo, da uporabniki opravljajo naloge. Definicije, kako uporabnik interaktivno sodeluje z funkcionalnostjo spletnega mesta.

Informacijska arhitektura: oblikovanje strukture informacijskega prostora, ki omogoča intuitiven dostop do vsebin.

- julij 2001 - izbira sistemske programske in strojne opreme
- september 2001 – logični model sistema
- maj 2002 – vsi sistemi postavljeni; realizacija propagacije

4. faza

Oblikovanje informacij: oblikovanje predstavitve informacij, ki olajša razumevanje.

Oblikovanje vmesnika: kot v tradicionalnem vmesniku človek-računalnik - oblikovanje elementov vmesnika, ki olajšajo interakcijo uporabnika z funkcijami.

Oblikovanje navigacije: oblikovanje elementov vmesnika, ki omogoča premikanje uporabnika med stranmi.

- avgust 2001 – prvi prototip grafičnega vmesnika
- januar 2002 – predstavitev grafične podobe upravi

5. faza

Vizualno oblikovanje: grafična obdelava elementov vmesnika; vizualna obdelava besedil, grafičnih elementov strani in navigacijskih komponent.

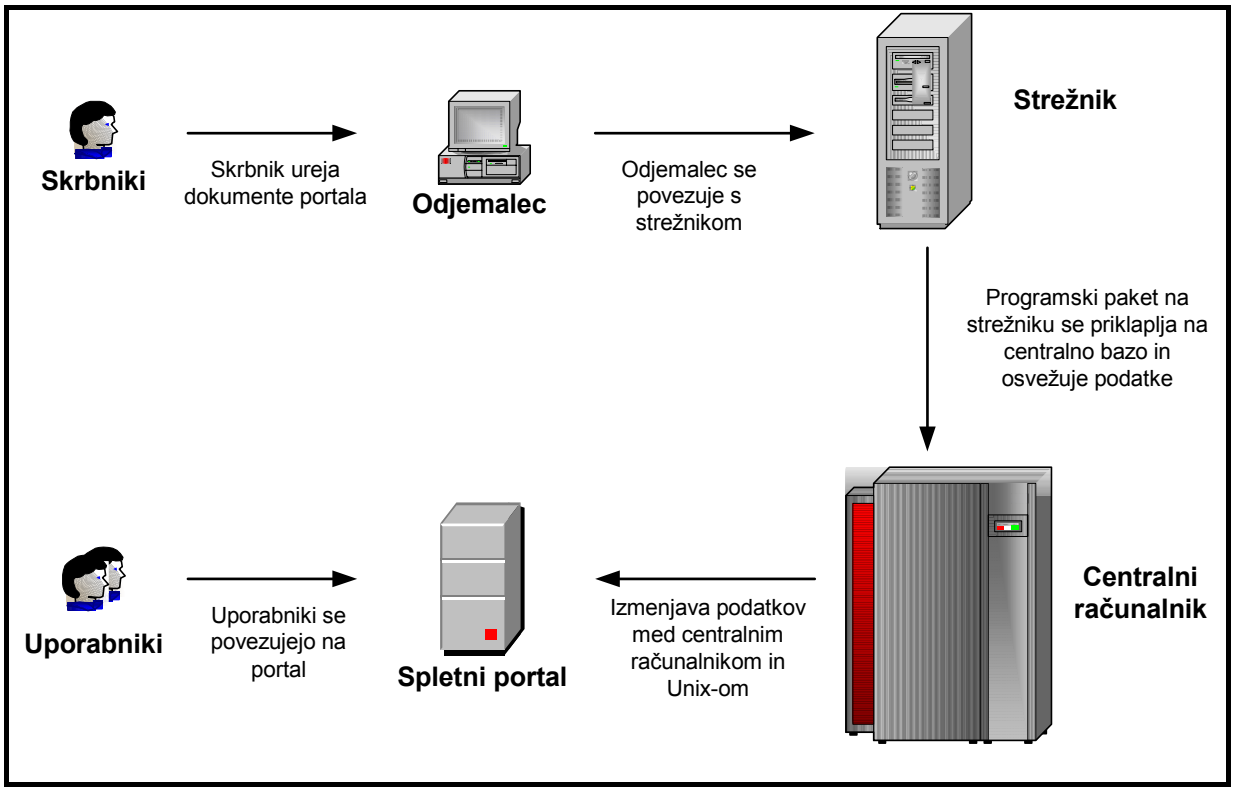
- februar 2002 – končan koncept celotne grafične podobe portala

Za postavitev portala so bile potrebne investicije za nakup programskega paketa, ki smo ga izbrali na natečaju in Unix strežnika. Vse vmesne sisteme in glavni računalnik smo že imeli na voljo. Prav tako nismo imeli dodatnih stroškov glede programskih licenc za operacijske sisteme in SUBP.

Končni tok podatkov od kreiranja novega dokumenta (ali popravka obstoječega) do prikaza na spletnih straneh prikazuje slika 5. Skrbnik kreira ali popravi dokument in ga shrani v podatkovno bazo na centralnem računalniku. Ves postopek opravlja preko odjemalec/strežnik aplikacije. Preden se dokument potrdi, ga pregleda še drugi skrbnik, pri čemer ugotavlja skladnost dokumenta z njegovo vsebino. Potrjen

dokument se nato podvoji s procesom replikacije na strežniku spletnega portala, do katerega lahko dostopajo uporabniki.

Slika 5: Podatkovni tok spletne strani



3. SISTEMI V NLB ZA PODPORO DELOVANJA SPLETNEGA PORTALA

Celoten informacijski sistem v banki je zelo kompleksen, kar je razumljivo, saj pokriva območje celotne države in vzdržuje povezave s tujino. Povezuje vse poslovalnice, bankomate in še množico lokalnih omrežij, ki so razporejena po upravnih stavbah.

Zaradi kompleksnosti in lažje razumljivosti sem okvirno predstavil posamezne sisteme, ki služijo tudi za podporo delovanja spletnega portala. V grobem jih lahko razdelimo na tri logične enote:

- sistem za upravljanje z bazo podatkov
- omrežja
- varnost.

3.1 Sistem za upravljanje z bazo podatkov

SUBP je del sistemske programske opreme, ki uporabniku omogoča definiranje, oblikovanje in vzdrževanje podatkovne baze, hkrati pa omogoča nadzor nad dostopom do podatkovne baze oziroma je posrednik med podatkovno bazo in uporabniškimi programi (Bobek, Strnad, 2004, str. 27). SUBP v današnjem času spadajo med najpomembnejšo programsko opremo, ki vodi informacijsko dobo.

Včasih so datoteke in baze podatkov vsebovale manjšo količino podatkov. Ti podatki so bili shranjeni na centralizirani lokaciji, ki je zagotavljala dostop do omejenega števila ljudi preko nepriročnih metod. Ko se je vse več tipov podatkov selilo v elektronske formate in so se, pogosto z malo načrtovanja, močno razširile baze, se je pojavilo povečanje uporabnikov, dostopnih metod, prenos podatkov med komponentami in kompleksnost nižje-ležečih tehnologij.

V NLB imamo naslednje SUBP:
- DB2
- Supra
- SQL Server
- ORACLE
- Universe.

Vrstni red ponazarja tudi pomembnost podatkov, ki jo posamezni podatkovni sistem hrani. Sistem DB2, kakor tudi ostali SUBP, uporablja jezik SQL. Namenjen je za opis in uporabo podatkov.

Naštete lastnosti v nadaljevanju poglavja (robustnost, zmogljivost, nadgradljivost) v celoti veljajo le za SUBP DB2. Seveda najdemo veliko skupnih točk tudi z SUBP SQL Server in ORACLE, ki pa v nalogi ne bosta predmet razprave. Centralna baza DB2 namreč hrani podatke spletnega portala in podpira ter izvaja replikacijo na lokalno DB2 bazo, kjer so ti podatki dostopni v svetovni splet. Podrobnosti arhitekture sem opisal skozi nadaljnja poglavja.

3.1.1 Skrbniki podatkovnih baz

SUBP upravljamo skrbniki baz podatkov (DBA - DataBase Administrator). In kdo smo skrbniki baz podatkov? Odgovor sem povzel po Mullins S. Craig (Mullins, 2002b): "Skrbnik je informacijski tehnik, ki je zadolžen za zagotavljanje operacijske funkcionalnosti in učinkovitosti SUBP in aplikacij, ki dostopajo do nje. Skrbnik SUBP nima nikakršnega vpogleda v fizično zgradbo podatkovne baze. Vse opravita sistem za upravljanje baz in operacijski sistem."

Poslovne zahteve so mnogokrat zelo kompleksne. Tehnološke rešitve zagotavljajo več možnih poti. Od strokovnjakov za SUBP se pričakuje poznavanje arhitekture, načrtovanja, izgradnje baznih komponent in poslovnih rešitev, uporabo najnovejših tehnologij in zagotavljanje znanj s področja baznih tehnologij za programerske arhitekte, razvojnike in uporabnike (Komer, 2001).

Podatki so življenjska tekočina računalniških aplikacij, ki so razvite za podporo poslovnim procesom. Skrbniki baze podatkov smo v centru razvojnega življenjskega cikla - zagotavljamo, da imajo aplikacije učinkovite in točne dostope do podatkov nekega podjetja. Tako pogosto delujemo kot vmesnik med različnimi skupinami uporabnikov: tehniki, programerji, končnimi uporabniki, strankami in izvajalci. Skrbniki zato potrebujemo izjemne komunikacijske sposobnosti (Mullins, 2002a).

3.1.2 Lastnosti SUBP DB2

Tradicionalno se informacijska infrastruktura za podjetja uporablja kot jedro za podporo poslovnim procesom. Zato mora biti zelo robustna, saj pogosto zahteva 24-urno delovanje. Kadar produkcijski sistemi ne delujejo zaradi okvare, zaposleni ne morejo delati, draga oprema se ne uporablja, material in storitve se ne morejo tržiti. Stroški nedelovanja so zato lahko zelo visoki (Beyer, 1999).

DB2 je SUBP in je izdelek IBM-a. Temelji na relacijskem podatkovnem modelu. Zaradi zanesljivosti se ponavadi nahaja v ustanovah, ki bi imela (ali imajo) zelo visoke stroške z nedelovanjem sistema (Beyer, 1999). Tradicionalno so bili to sistemi zasnovani na MVS ali OS/400 platformah. Sedaj nekatere organizacije uporabljajo Unix ali celo Windows operacijske sisteme, čeprav je odstotek delovanja teh platform precej nižji. Če obstaja zahteva po relacijski bazi, je DB2 pogost spremljevalec. Razlog za večji tržni delež je robustnost, ki je kritična v 24x7 delovanju sistema. Druga dobra lastnost so zmogljivosti, saj omogoča izvajanje dinamičnih in statičnih poizvedb (SQL - Structured Query Language).

DB2 produkti se na različnih platformah v 90% nanašajo na isto osnovno programsko kodo. Ostalih 10% vzamejo povezave z operacijskim sistemom. Podjetja lahko razvijajo produkte na Windows operacijskem sistemu, čeprav bo izvedba potekala na Unix-u ali Linux-u (Yevich, 2001).

Kakor povsod, se dogajajo spremembe tudi na področju orodij DB2. Tako je tudi z verzijo 7. Najbolj opazna sprememba je, da je družina postala samo en svet – DB2. DB2 za OS/390 je dobil podporo za procese, ki so bili poznani samo na drugih platformah in Unix, Windows in OS/2 so prav tako dobili podporo za procese iz OS/390 (Lawson, Hubel, Mordini, 2000).

Očitno je, da s to verzijo IBM cilja na niz orodij za razvoj aplikacij, ki jih stranke lahko uporabljajo za razvoj na obeh DB2 (OS/390 na eni in Unix, Windows in OS/2 na drugi strani). IBM prav tako stopnjuje skupne skrbniške možnosti skozi družino DB2, z omogočanjem delovanja nadzornega centra in centra za izvajanje komand (oba sta grafična vmesnika) na vseh platformah.

3.1.2.1 XML podpora

DB2 je v verziji 7 razširila funkcionalnost z XML (eXtensible Markup Language) podporo. Sedaj je mogoče kombinirati strukturo podatkov v XML formatu s tradicionalnimi relacijskimi podatki. XML dokument lahko shranimo kot XMLCLOB (XML Character Large Object) ali pa preslikamo vsebino kot tradicionalne podatke v relacijskih tabelah (Lawson, Hubel, Mordini, 2000).

Podpora za XML dodaja možnosti iskanja po bogatih podatkovnih tipih XML elementov. Prihajajoči XML dokumenti se lahko razstavijo v tradicionalne SQL podatkovne tipe in shranijo v stolpce. V primeru uporabe pa sestavijo v izhodni XML dokument. Izvedbene metode omogočajo transformacijo med XML dokumenti in relacijskimi podatki. Tako lahko razbijemo XML dokument na enega ali več kosov za shranjevanje v tabelah in obratno.

3.1.2.2 Objekti v sistemu DB2

Osnova gradnika konceptualnega podatkovnega modela sta tabela in baza podatkov. Tabela je določena z imenom in obsega poljubno število stolpcev ter poljubno število vrstic, medtem ko baza podatkov logično zaokrožuje skupino tabel, ki služijo določenemu poslovnemu procesu. Stolpci so ravno tako določljivi le z imenom. V okviru vsakega tipa podatkov obstaja tudi nedefinirana oziroma ničelna vrednost (NULL). Več logično povezanih tabel zberemo v eno podatkovno bazo. SUBP pa lahko upravlja z več različnimi bazami podatkov.

DB2 vsebuje množico pogledov ter tabel. V okviru nekega pogleda lahko priredimo tiste podatkovne elemente, ki uporabnika zanimajo. Podatki, ki tvorijo pogled, so lahko deli različnih tabel, ali že oblikovanih pogledov. Sistem sam izlušči zahtevane podatke in iz njih sestavi logični zapis, ki je definiran s pogledom. Sistem tudi sam določi način dostopa do želenih podatkov (s pomočjo indeksa ali sekvenčno). Poenostavljena je tudi zaščita dostopa do podatkov. Uporabniku lahko preprečimo uporabo določenega stolpca, lahko pa mu dovolimo le uporabo vrstic tabele, ki zadoščajo nekemu pogoju.

Indeks je ukazni niz kazalcev na podatke v DB2 tabeli. Shrani se ločeno od tabele. Vsak indeks je baziran na vrednostih podatkov v enem ali več stolpcih tabele. Ko kreiramo indeks, ga DB2 ohranja, toda mi ga lahko preverimo, popravimo ali reorganiziramo.

Indeks neke tabele lahko kreiramo vedno, kadar najprej kreiramo tabelo in ni pogojen z vnosom podatkov. Razen sprememb v izvajanju, se uporabniki tabel ne zavedajo, da so bili indeksi uporabljeni. V DB2 vgrajeni algoritmi skrbijo za to, kdaj se bo uporabljal indeks za dostop do tabele. Indekse lahko uporabljamo za:

- izboljšanje izvajanja (v mnogih primerih je dostop do podatkov hitrejši z indeksom kot brez),
- zagotavljanje originalnosti (tabela z unikatnim indeksom ne more imeti dveh vrstic z enakimi vrednostmi v stolpcu ali stolpcih, ki tvorijo indeksni primarni ključ).

Poleg tabel, pogledov in indeksov največ uporabljamo še tabelarični prostor, ki zagotavlja fizični prostor za tabelo na pomnilni enoti, in prožilce, ki prožijo programirane aktivnosti glede na vnos, popravek ali brisanje zapisa.

3.1.2.3 Beleženje sistema DB2

Replikacija zahteva specifično nastavitev beleženja SUBP DB2. Ob neprimerni izbiri nam ne bo uspelo vzpostaviti procesa za podvajanje podatkov. Izbiramo lahko med cirkularnim in arhivskim načinom (Yakushin, 2002).

Značilnosti cirkularnega načina so:

- ob kreiranju baze je določeno število primarnih beleženj (minimalno tri) z določeno velikostjo alociranih na fizični pomnilni enoti (število in količina beleženj se lahko spreminjata, vendar samo navzgor od privzetih vrednosti),
- primarna beleženja se polnijo cirkularno,
- vsako beleženje je ponovno uporabno,
- če SUBP zmanjka primarnih beleženj, uporabi sekundarne. Ti niso alocirani ob kreiranju baze, ampak se kreirajo po potrebi. Prav tako se samodejno zbrišejo, ko jih ne potrebujemo več,
- ne omogoča "online" varnostne kopije, replikacije.

Značilnosti arhivskega načina so:

- primarna beleženja niso ponovno uporabna, ampak se zaporedno nizajo na pomnilni enoti, dokler niso fizično zbrisana,
- sekundarna beleženja se uporabljajo na enak način kot pri cirkularni metodi,
- podpira "offline" in "online" izdelavo varnostnih kopij in izvajanje replikacije,

- Skrbnik baz podatkov mora poskrbeti za hranjenje beleženj (na disku ali tračni enoti), ki jih potrebuje, in odstraniti tiste, ki jih ne.

Delovanje SUBP urejamo preko različnih parametrov. V njih na primer določimo število in velikost beleženj, kodno tabelo, avtorizacije, format datumskega zapisa… Kadar naredimo spremembo na določenem parametru, se le-ta odraža šele po ponovnem startu SUBP. Trend je, da bi se spreminjanje večine parametrov izvajalo dinamično, kar bi pomenilo hitrejši odziv na spremembe in temu primerno boljše delovanje, oziroma optimizacijo (Beyer, 1999).

3.1.3 Centralna baza

Centralna baza v NLB deluje na IBM-ovem centralnem sistemu (veliki računalnik). Ime nam pove, da osnovna arhitektura za splošno rabo izhaja iz 60-ih let prejšnjega stoletja. Seveda pa je bil vseskozi izpostavljen tehničnim novostim in modernizaciji. Stare centralne aplikacije še vedno tvorijo velik del informacijskega sistema v mnogih podjetjih. Približno 70% svetovnih podatkov se hrani na teh sistemih in sprocesirajo 85% vseh poslovnih transakcij (Mattsson, 2003).

SUBP izkorišča vse prednosti arhitekture, kot so velika procesna zmogljivost, varnost, zanesljivost. Z verzijo 7 je pridobila tudi več fleksibilnosti v upravljanju podatkov in boljšo integracijo z družino DB2 (Administration Guide, 2001, str. 3).

DB2 na gostitelju uporablja varianto zgoraj opisanega cirkularnega beleženja. Razlika je v tem, da cirkularna beleženja po tem, ko jih napolni, arhivira. Arhiv se hrani poljubno dolgo (najprej na disku in po treh dneh na magnetnem traku), glede na izkušnje, potrebe aplikacij (beri stabilnosti)…

DB2PM je orodje za spremljanje in nadzor procesov, ki se vršijo v SUBP DB2. Skozi zadnjih nekaj let je bil za zagotavljanje sprotnega nadzora programski paket nadgrajen (Sherman, 2001). Tako nam pomaga zagotoviti najoptimalnejšo porabo virov in odzivni čas za uporabnike.

3.1.3.1 Centralni sistem

Jedro računalniškega sistema je v posebni sobi, ki je obenem tudi varnostna celica pred požari, vlomi in drugimi nevšečnostmi. Sistem sestavljata IBM-ova centralna računalnika, ki se ločita na strojni del (S/390) in operacijski sistem (OS/390).

Produkcijski S/390 služi za obdelavo podatkov v produkciji (aplikacije za podporo bančnemu okencu, bankomati, obdelave podatkov…). Testni pa služi testiranju posameznih aplikacij, preden te prenesejo v delujoč sistem. Zagotavlja nam širok razpon storitev in nam zagotavlja pomembne prihranke v energiji in stroških vzdrževanja. Strežniška platforma je standardno definirana za podjetniške potrebe, kar pomeni, da je izredno fleksibilna in funkcionalna. Arhitektura je skoncentrirana okoli sklopne kontrole in zelo hitrih sklopnih povezav za notranjo komunikacijo (Hoskins, Frank, 2002, str. 416). Tehnologija S/390 integrira strojno opremo in tako ponuja vsestransko rešitev za končne uporabnike, ki so podvrženi nestanovitnemu in tekmovalnemu poslovnemu okolju.

Operacijski sistem OS/390 izpolnjuje zahteve po stabilnosti (ki zagotavlja nemoteno delovanje sistema) in učinkovitosti (kar pomeni hitrejše razpošiljanje ukazov in krajši odzivni čas). Ima tudi dobre lastnosti na področju elektronskega poslovanja, kjer nudi visoke varnostne mere. Njegov naslednik Z/OS podpira še več funkcionalnosti in je celo predpogoj za migracije na višjo verzijo nekaterih produktov, kot je tudi sistem DB2.

3.1.4 Lokalne baze

Upravljanje distribuiranih ali lokalnih baz je mnogo težje, kot upravljanje centralnih (Thompson, 1997). Potrebno je več časa za načrtovanje in izvajanje dodatnega dela ter skrbeti za razpršene sisteme, mnogokrat na različnih lokacijah. Lahko pa uspešno razbremenimo centralni sistem z neodvisnimi in nezahtevnimi aplikacijami, ki lahko delujejo lokalno.

Lokalna baza vsebuje zmogljivosti, ki skrbnikom omogočajo izdelavo avtomatskih povzetkov tabel, statističnih funkcij, večje upravljanje nad procesom za masovne vnose podatkov v tabele (uslužnostni program za vnos – load utility) (Perna, 1999). Dodatna orodja nam pomagajo pri najoptimalnejši postavitvi indeksov, ki pomembno vplivajo na delovanje aplikacij.

Nekateri operacijski sistemi lahko gostijo različne aplikacije. DB2 UDB to učinkovito izrablja za podporo svojega delovanja, velikih podatkovnih skladišč, elektronskega poslovanja in samopostrežnega spletnega portala (Logan, 2000). Rezultat je, da pogosto dosegamo znatne prihranke in prednost obstoječe DB2 z izkušnjami pri razvoju DB2 programske opreme.

Privlačnost NT operacijskih sistemov je tipična podpora za opravila, ki zahtevajo manj strokovnosti kot tista v Unix-u ali centralnem računalniku. IBM je zato popolnoma na novo predelala vmesnik že v verziji 5. Produkt nadzornega centra

omogoča DB2 skrbnikom hitro in enostavno kreiranje, spreminjanje in upravljanje z vsem baznimi objekti (tabele, prožilci, medpomnilniki…) (Fosdick, 1998). In seveda so tu še drugi centri, ki omogočajo kontrolo nad izvedenimi ukazi in skriptami, beležijo bazne zmogljivosti in prožijo alarme v primerih potencialnih problemov.

IBM vključuje mnogo posebnosti v DB2 za NT, ki združujejo še ostale pomembne produkte in gredo celo korak dlje s podporo za multimedijske podatkovne tipe. Prav tako vsebuje optimizator poizvedb (query optimizer), ki na podlagi izvedene statistike nad podatki izbere najboljšo dostopno pot za vsako posamezno poizvedbo. Optimizator (Lindholm, 1996) lahko:

- optimizira več spajanj in jih tako naredi bolj učinkovite,
- ponovno napiše poizvedbo po združevanju in preverjanju pomenskosti.

Lokalna baza ne pozna arhiviranja cirkularnega beleženja kot na primer centralni računalnik. Zato moramo uporabljati arhivsko beleženje, ki edino omogoča izvedbo replikacije. To je varnostni mehanizem, ki zagotavlja daljše hranjenje beleženj in nam posledično omogoča večjo stabilnost.

3.1.5 Orodje za povezovanja med bazami

Za povezave med centralno in lokalnimi bazami skrbi dodatno orodje, ki je že vgrajeno v bazne sisteme. Potrebna je le nastavitev, ki omogoči nemoteno sporazumevanje med SUBP. Orodje se imenuje DB2 Connect in dovoljuje odjemalcem dostop do podatkov, shranjenih na različnih baznih strežnikih skozi interni DB2 DRDA protokol (Distributed Relational Database Architecture) (Yevich, 1999, str. 8). Sicer obstajajo še drugi produkti ostalih proizvajalcev, ki pa jih mi ne uporabljamo.

Replikacija v NLB prav tako uporablja DRDA protokol. Preko njega se poveže na centralni računalnik in izvaja predpisane aktivnosti. Vsak odjemalec, ki uporablja aplikacijsko ukazni protokol, se lahko poveže na katerikoli strežnik, ki ga predstavlja aplikacijsko strežniški protokol. DRDA je temelj IBM-ovega komunikacijskega protokola (Young, 2000).

DB2 Connect je produkt IBM-a, ki poskrbi za hitro in enostavno povezavo do obstoječih baz. Informacijska struktura je visoko nadgradljiva za povezavo do podatkov na internetu, Windows-ih, Unix-u (Linux-u), OS/2-ju, na S/390 in AS/400. Povezavo uporabljajo tudi aplikacije, kot so Java, dinamične in statične poizvedbe, grafični vmesniki in drugi.

DB2 Connect deluje s sprejemanjem zahtev (Young, 2001) povezav, poizvedb, baznih sprememb in potrditev od DB2 UDB za Unix, Windows in OS/2 odjemalcev, ki jih pošilja skozi DRDA do DB2 UDB za OS/390 ali AS/400 strežnikov. Prav tako zagotavlja točko koordinacije za aplikacije, ki izvajajo spremembe na več različnih sistemih s podatkovno integriteto z zagotavljanjem dvofazne potrditve.

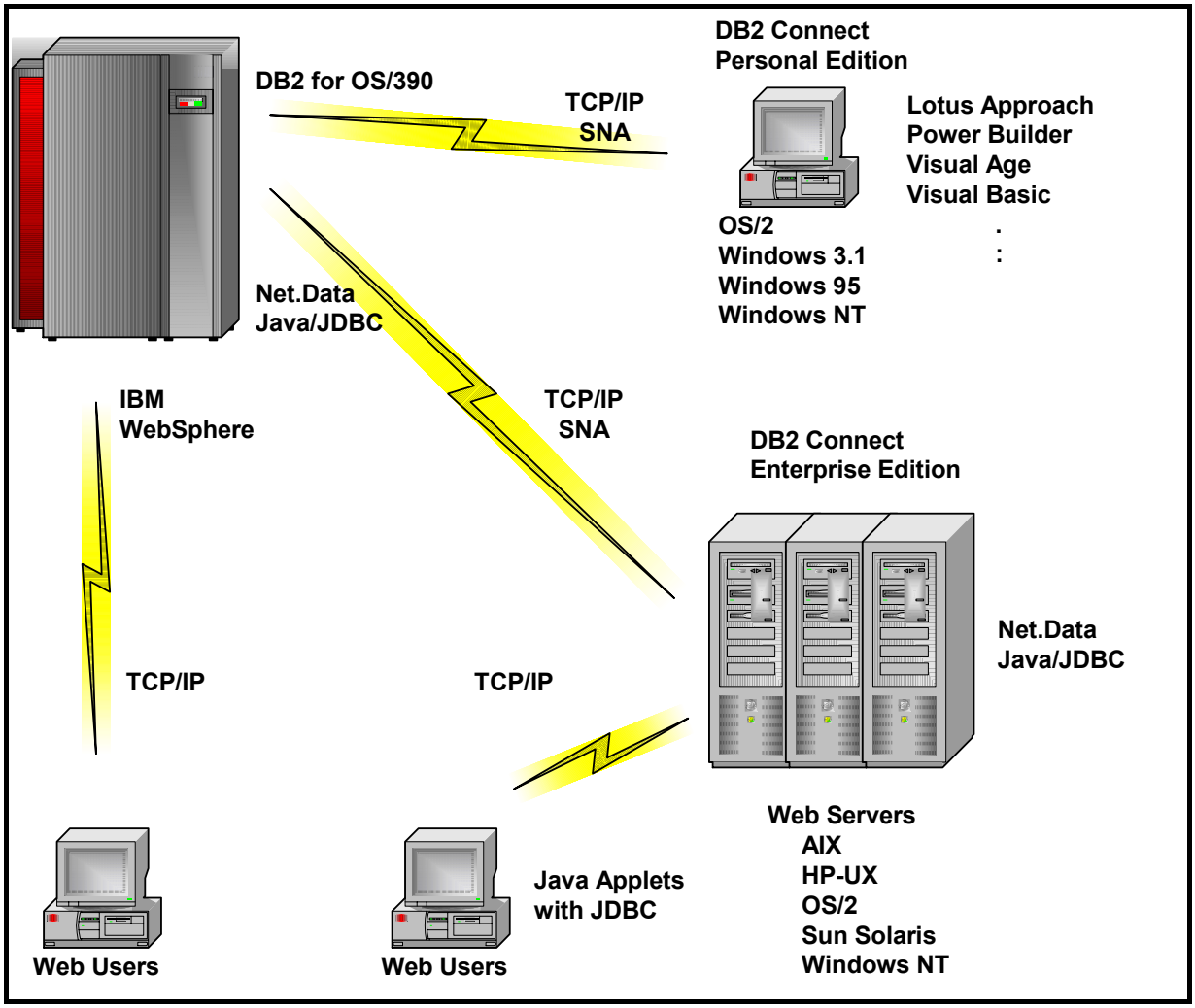
Vsako nadgraditev programske opreme je potrebno storiti samo na strežniku in ne na vsakem odjemalcu. Kadar odjemalec potrebuje povezavo do DB2 na OS/390, se kreira opravilni agent na DB2 Connect strežniku. Agent izvaja naslednje funkcije (Young, 2001):

1. kreira odgovarjajočo povezavo do DB2 na centralnem računalniku,
2. sprejme zahteve od odjemalca,
3. konvertira zahtevo odjemalca v DRDA protokolni format,
4. odpošlje DRDA zahtevo iz centralnega računalnika v različnem komunikacijskem protokolu (TCP/IP iz odjemalca in SNA (Systems Network Arhitecture) do DB2 UDB OS/390),
5. sprejme odgovor in konvertira dobljene podatke v kodno tabelo odjemalca,
6. formulira povratno sporočilo,
7. počaka na naslednjo zahtevo odjemalca. Če naslednji ukaz ne povzroči ukinitev povezave, se ponovi niz zgornjih šestih korakov. Drugače pa zaključi povezavo med odjemalcem in centralnim računalnikom in umakne strežniško opravilo iz sistema.

DB2 Connect vzdržuje povezavo do potrditve. Nato v navezavi z DRDA sprosti vez. Produkt nenehno beleži in usklajuje število povezav, ki jih ima do posameznega odjemalca, in ugotavlja, h kateremu mora usmeriti naslednjo povezavo. Uravnoteženost mrežnega prometa omogoča visoko stopnjo nenehnega delovanja (povezava več strežnikov v grozde). Napaka katerega koli strežnika med delom se odraža v distribuciji nalog na preostale aktivne člane v grupi. Tako se omogoča nemoteno opravljanje operacij. Raznovrstnost povezav, ki jih vzdržujemo preko DB2 Connect-a, so vidne na sliki 6.

Če povzamemo dejstva, ugotovimo, da je DB2 Connect močno programsko orodje, ki povečuje dostopnost podatkov na centralnem računalniku preko odjemalec/strežnik arhitekture ali Internet dostopa (Smith, Young, 2001).

Slika 6: Raznovrstnost povezav



Vir: Instalation guide, 2001, str. 462.

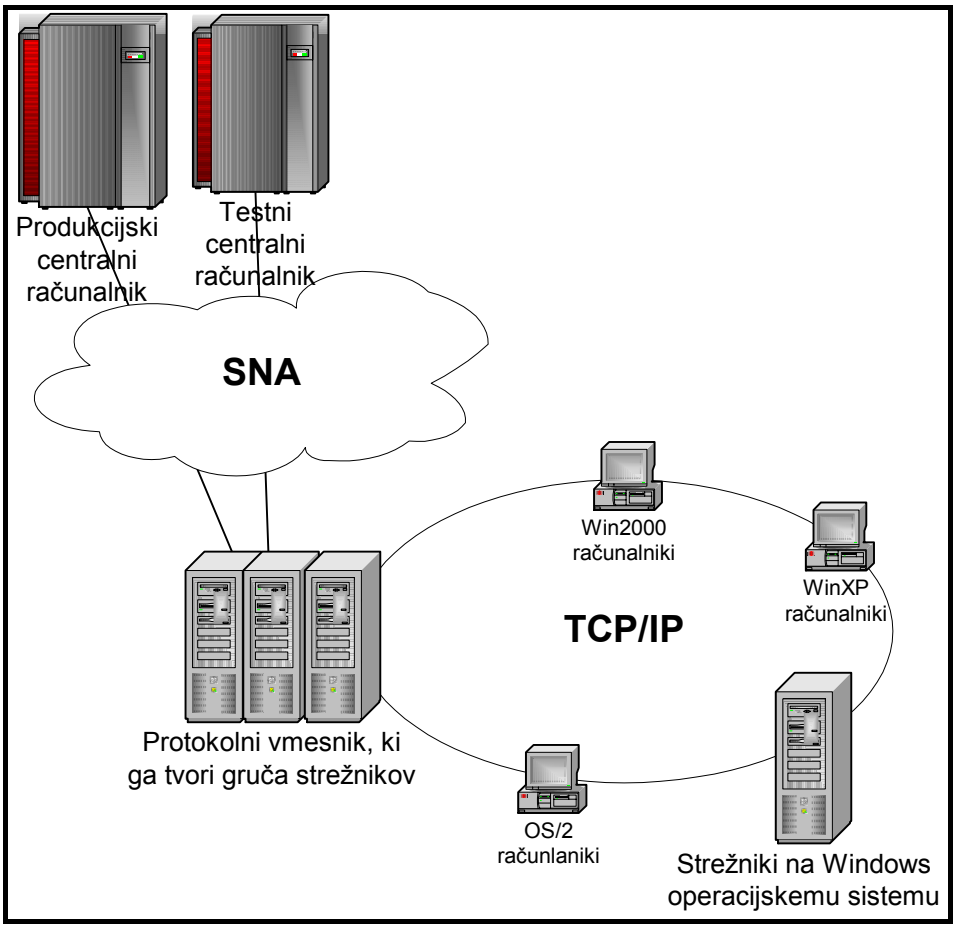
3.1.5.1 Prilagojeni sistemski programi

Prilagojeni sistemski programi kontrolirajo tok podatkov od izvora do cilja ali pa nam omogočajo vpogled v različne informacije skozi, na primer svetovni splet (Young, 2000).

Mnogo podjetij z nekaj različnimi podatkovnimi sistemi ima možnost migracije na en SUBP standard. To je vsekakor bolje, kot vzdrževati heterogene povezave. Vendar je v nekaterih organizacijah, med katerimi je tudi NLB, raznolikost tako kompleksna, da je uporaba prilagojenih sistemskih programov edina realna rešitev. S tem, ko se vse več organizacij sooča z izzivom mešanih sistemov zaradi decentralizacije poslovnih operacij, združitev, pridobitev, programskih nadgraditev, sprememb v poslovanju in

zahtev poslovnih procesov, bodo novejše in boljše rešitve za srednja okolja žela uspeh na trgu. Mešani SUBP in programska oprema vmesnega okolja bosta nedvomno hitro dozorevala z željami uporabnikov in zahtevami po večfunkcionalnosti.

Slika 7: Protokolni vmesnik



Vir: http://www.ibm.com, 2004.

Po eni strani prilagojeni sistemski programi privlačijo poslovno stran k novim podatkovnim virom in poizvedbam, ki obremenjuje sistemske inženirje in podporno osebje. Po drugi strani pa lahko (Marshall, 1998):

- dajejo večjim organizacijam možnost izkoriščanja informacij, za katere niti ne vedo, da jih posedujejo,
- da se hitro priključijo pri prenovi poslovanja,
- da izvajajo vse operacije na novih podatkih.

V to skupino spadajo tudi protokolni vmesniki (gateway). So postaje, ki v NLB skrbijo za transformacijo mrežnih paketov iz SNA v TCP/IP protokol in obratno. Dejansko so to vstopna vozlišča za dostop do podatkov na glavnem računalniku (slika 7). Vse aplikacije, ki za svoje delo potrebujejo bazo in so napisane za osebne računalnike, dostopajo do podatkov preko protokolnega vmesnika.

3.2 Varnost

Varnost predstavlja pomemben segment v zaščiti in varovanju programske in strojne računalniške opreme, podatkov, fizičnih naprav in omrežij. Delimo jo lahko na:

- fizično in
- logično.

3.2.1 Fizična varnost

Fizična zaščita pomeni ločitev prostorov in naprav informacijske tehnologije, zato da se zavarujejo pred prekinitvami in nepooblaščenim dostopom. Onemogoča krajo informacijskih sredstev, preprečuje dostop do občutljivih konzol in kriptografskih modulov ter zagotavlja zaščito pred nepooblaščenimi spremembami v opremi informacijske tehnologije. Največkrat se za fizično zaščito uporabijo posebni dobro zaščiteni prostori, ki so zaklenjeni, opremljeni z video nadzorom ter so varni pred požari in poplavami (Gril, 2003, str. 9).

V NLB so prostori v banki fizično zaščiteni v štirih conah z različnimi varnostnimi ukrepi. Bližje kot je cona računalniškemu centru, bolj restriktivni so varnostni ukrepi in zaščite ter manj zaposlenih ima dostop vanjo. Vsak prihod zunanjih sodelavcev v podjetje registrirajo varnostniki na vhodu. Zaposlenim v Upravljavskem centru za informacijsko tehnologijo je zagotovljen vstop s kartico in PIN kodo. V nadaljevanju pa lahko dostopamo še do podatkovne knjižnice (služi za arhiv podatkov in izdelavo varnostnih kopij) in glavnega sistema, če smo za to pooblaščeni in če nam dana situacija to omogoča.

3.2.2 Logična varnost

Logična zaščita dostopa do informacijskih sistemov je težji oreh. Ukvarja se z omejevanjem dostopa tam, kjer ni fizičnih zaščit ali kot dopolnilo (Gril, 2003, str. 9). V NLB ima vsak uporabnik omrežja svoje uporabniško ime geslo. Brez njiju delo na delovni postaji ni mogoče. Za delovanje mreže skrbi mrežni skrbnik. Posebno uporabniško ime in geslo sta potrebna tudi za dostop do centralnega računalnika.

Sistemski inženir tu skrbi za nadzor nad sistemom in dodeljevanjem uporabniških pravic. Vsak uporabnik namreč nima dostopa do posameznih ali vseh sistemov, če mu tega ne odobri sistemski inženir.

Sistemski inženirji za zaščito dostopa uporabljajo programski paket CA-TSS (Top Security System), izdelek podjetja Computer Associates. Namenjen je programski zaščiti računalniškega sistema, programske in strojne opreme. TSS omogoča nadzor nad dostopi do sistema, dostopi do aplikacij in podatkov, ki se nahajajo v datotekah. Omogoča tudi nadzor nad izvajanjem obdelav in programov, nad uporabo ukazov operacijskega sistema in podsistemov.

Poznamo več vrst omejitev dostopa do centralnega sistema in datotek. Prvi nivo zaščite je že v namestitvi emulacijskega programa, ki služi za dostop do centralnega sistema. Če do dostopa na sistem nismo upravičeni, se tega na osebni računalnik ne namesti.

Odobritev dostopa in nivo prioritete mora za uporabnika podpisati vodja oddelka in odobriti varnostni inženir. Šele nato sistemski inženirji določijo tvoje mesto v sistemu.

3.2.3 Zaščita podatkov v bazi podatkov

Naloga zaščite je preverjati ali je uporabnik pooblaščen za izvrševanje opravil, ki jih želi izvesti. Problem zaščite ima več vidikov:

- pravni, sociološki oziroma etični vidik (ali ima uporabnik pravico uporabljati določene podatke, npr. o višini bančnega računa),
- fizična zaščita (zaklepanje prostorov),
- operativni problemi (zagotoviti, da je geslo res tajno),
- možnosti kontrole, vgrajene v strojno opremo (ali vsebuje procesor dodaten, privilegiran nabor ukazov).

Zavedati se moramo predvsem tega, da podatki ne smejo biti na voljo vsakomur, predvsem pa mora biti podatkovna baza zaščitena pred namernimi ali naključnimi nepooblaščenimi posegi (spreminjanje, brisanje podatkov). Da lahko SUBP zadosti tem zahtevam, mora prepoznati oziroma identificirati vsakega uporabnika in mora "vedeti", kaj mu je dovoljeno. Zato mora obstajati orodje za dodeljevanje, oziroma odvzemanje pooblastil za delovanje znotraj SUBP.

V SUBP ima vsak uporabnik več identifikacijskih oznak. Primarna identifikacijska oznaka (primary ID) služi predvsem za dostop do SUBP samega. Na podlagi te oznake so uporabniki razdeljeni v več kategorij. Dodatne identifikacijske oznake (secondary ID) pa služijo predvsem za dodeljevanje dodatnih pravic uporabe

sistema. Obstaja še posebna identifikacijska oznaka (SQL – ID) za prirejanje identifikacije jezika SQL.

Ob izvedbi SUBP ima dostop do vseh objektov sistema le varnostni vzdrževalec podatkovnih baz (security administrator). Z ukazi pa lahko dodeljuje in odvzema pravico do kreiranja in uporabe celotne baze podatkov, posamezne tabele, pogleda, aplikacije, sistemskega kataloga in tudi do posameznih ukazov.

Problematika zaščite se nanaša tudi na komunikacijski medij, kamor se pošiljajo uporabniška imena, gesla in podatki. DB2 V7 zagotavlja strežniško podporo za zamenjavo kodiranih gesel. Pri beleženju na sistem se geslo, preden se pošlje do želenega sistema, kodira (Presentation Guide, 2001, str. 281). Avtentikacija se lahko izvaja tudi preko zunanjih DB2 varnostnih pripomočkov, ki so lahko del operacijskega sistema ali ločen produkt. Pri katalogizaciji posameznih baz moramo posebej izbrati opcijo strežniškega kodiranja. Tako se vsa uporabniška imena in gesla po mreži prenašajo kodirano (Yakshuin, 2002). Preden dodelimo IP naslove ali podomrežje za povezavo med strežniki in osebnimi računalniki, se moramo posvetovati z mrežnim skrbnikom, da zagotovimo varnostno politiko (preverimo dostope in stanje mrežnih naprav) (Simpson, 2002).

Na glavnem sistemu je več različnih aplikacij: operacijski sistem, CICS, DB2, … in TSS (Top Security Sistem). Procesor vsebuje tudi predprocesor, ki komunicira z ločenimi enotami in v tem pogledu razbremenjuje glavni procesor. Prav tako tudi vzpostavlja komunikacijo (SNA) s terminalskimi enotami, pri čemer varnostni sistem sproti preverja parametre in zaklene komunikacijski kanal. Potek je zelo hiter in je z vidika varnosti najoptimalnejši. V času komunikacije se namreč ne more vriniti noben drug računalnik, oziroma uporabnik. Linja je fiksna in rezervirana samo za kličoči terminal.

3.3 Omrežja

Omrežje v NLB je zelo razvejano in kompleksno. Zato se nisem spuščal v podrobnosti, temveč sem samo v groben nakazal njeno razvrstitev in širino.

3.3.1 Topologije računalniških omrežij v NLB

V NLB vzdržujemo tri topologije računalniških omrežij:

- topologija zvezde,
- topologija vodila in
- topologija obroča.

3.3.1.1 Topologija zvezde

Pri topologiji zvezde so vse delovne postaje povezane v eno skupno centralno vozlišče. Osnovna zveza je točka-točka. Postaje si delijo vire glavnega računalnika, ki ponavadi prevzame odgovornost za celoten potek komunikacije v tovrstnih sistemih (Simčič, 1988, str. 52).

Osnovna topologija v NLB je na principu zvezde. Vsi terminali so priklopljeni na centralni računalnik, kjer se vodi obdelava podatkov. Nekateri so vezani direktno, nekateri pa posredno preko večjih lokacijskih centrov.

3.3.1.2 Topologija vodila

Pri tem načinu so računalniki razporejeni okoli istega medija, ki si ga po potrebi med seboj tudi delijo. To je primer mnogotočkovne zveze. Osnovni komunikacijski medij se na obeh krajih zaključi s terminatorji. Sporočilo, ki je oddano od katerega koli računalnika, je takoj vidno vsem ostalim.

Naprava, ki želi komunicirati, se najprej prepriča, da je mreža prosta. Če je temu tako, začne s prenosom. Lahko pa se zgodi, da tudi druga delovna postaja začne oddajati v istem trenutku. V tem primeru obe napravi zaznata podvojene signale, kar pomeni, da je prišlo do trčenja, in takoj prenehata oddajati. Po določenem naključno pretečenem času, bo postaja zopet poskusila oddati sporočilo (Beyda, 1989, str. 158). Topologija je v svetu najbolj razširjena. Sedaj se vse bolj širi tudi v NLB in zamenjuje topologijo obroča, ki je dražja za vzdrževanje.

3.3.1.3 Topologija obroča

Bistvo mehanizma je žeton, ki se neprestano podi po mreži. Tista delovna postaja, ki žeton pobere, ima pravico na "igro" (Simčič, 1988, str. 87). Žeton nosi sporočilo in informacijo, komu je namenjeno. Ko ga prejme hotena delovna postaja, ta javi pošiljatelju status prenosa (uspešen, neuspešen). V kolikor je uspešen, se žeton sprosti in omogoča nadaljnje prenose med ostalimi postajami.

Žeton sam je kombinacija enic in ničel s posebnim pomenom in se lahko nahaja v prostem ali zasedenem stanju. Delovanje nadzira aktivni nadzorni mehanizem. Njegova naloga je zagotavljanje prvega prostega žetona ob zagonu mreže in generiranje novih ob neuspelih prenosih.

3.3.2 Komunikacijski protokoli v NLB

Obstoj mnogih različnih protokolov ni problem, če so različna omrežja ločena. Vsako od teh omrežij pa zahteva svojo opremo, rezervne dele, vzdrževalno osebje, osebje za tehnično podporo in strokovnjake. V NLB za komunikacijo uporabljamo dva komunikacijska protokola, to sta: SNA in TCP/IP. SNA je bil razvit za potrebe terminalov in centralnih računalnikov, TCP/IP pa je optimiziran za uporabo v omrežju širokih razsežnosti in ne za lokalna omrežja (Schnaidt, 1992, str. 261).

3.3.2.1 SNA

IBM–ov SNA je bil predstavljen 1974–tega leta (Beyda, 1989, str. 128-135). Postal je najbolj široko uporabljena komunikacijska struktura v svetu. Uspeh SNA gre pripisati sposobnosti, da lahko poveže široko področje naprav z različnimi zmožnostnimi v enotni mreži, katerih delo je skrito končnim uporabnikom. Težave pa povzroča vzdrževanje, saj se pojavi veliko dela ob novih ali spremenjenih definicijah. Mnogi komunikacijski protokoli so bolj odprti in "standardizirani" kot SNA (Hoover, 1997, str. 11).

3.3.2.2 TCP/IP

TCP je bil razvit v srednjih sedemdesetih letih za obrambno ministrstvo, tako da so lahko uporabniki ARPANET-a povezovali svoje računalnike, ki so bili lokacirani v vladi in vojski. Leta 1981 je bil izdan IP standard in leto kasneje je TCP/IP postal protokol uporaben na Internetu. In ker je bil zahtevan za Internet, so mnoge univerze posvojile ta novi protokol (Schnaidt. 1992, str. 271).

TCP/IP je paketno orientiran, plastni protokol. Kot ostali paketno orientirani protokoli SNA, DECnet in predvsem OSI, različne plasti komunicirajo samo s plastmi direktno nad ali pod njimi, omogočajoč določeno stopnjo samostojnosti in prostosti za različne plasti in aplikacije. Je štiri nivojski protokol, ki grobo sledi spodnje štiri nivoje OSI-jevega protokola. TCP/IP ni OSI-jev protokol, temveč absolutno en svet v dveh protokolih (Schnaidt. 1992, str. 272):

- TCP (Transmission Control Protocol)
- IP (Internet Protocol)

Vsak računalnik na TCP/IP omrežju je definiran z edinstvenim IP naslovom. Naslov mora biti globalno edinstven, kar pomeni, da računalnik ne more imeti istega IP naslova, kot katerikoli drug računalnik, s katerim lahko komunicira. Ohranjati naslove

globalno edinstvene v svetovnem omrežju, kot je internet, je velika reč. Na našem omrežju za to skrbi omrežni skrbnik.

3.3.3 Povezave v NLB

3.3.3.1 Centralni del

Vršni del piramide sestavlja sistem z dvema centralnima računalnikoma, od katerih je eden produkcijski, drugi pa testni. Vsak je povezan s svojim usmerjevalnikom. Zaradi varnosti je usmerjevalnik testnega sistema povezan še s produkcijskim centralnim računalnikom. Tako bi lahko v primeru odpovedi primarne povezave izvedli preklop na sekundarni vir. Povezava je na tem nivoju seveda optična (180 – 200 MB/s).

3.3.3.2 Distribuirani del

V lokalnih omrežjih sta obe topologiji nato povezani preko usmerjevalnika v centralni sistem. Pred usmerjevalnikom se nahaja še strežnik (protokolni konverter), ki nadzira pravilen prenos podatkov in pakiranje za prenos podatkov. Na lokalnih omrežjih so ponavadi priključeni še terminali in bankomati. Prenos se vrši tako, da obstoječe podatke IP zapakira v paket in ga na cilju zopet odpakira.

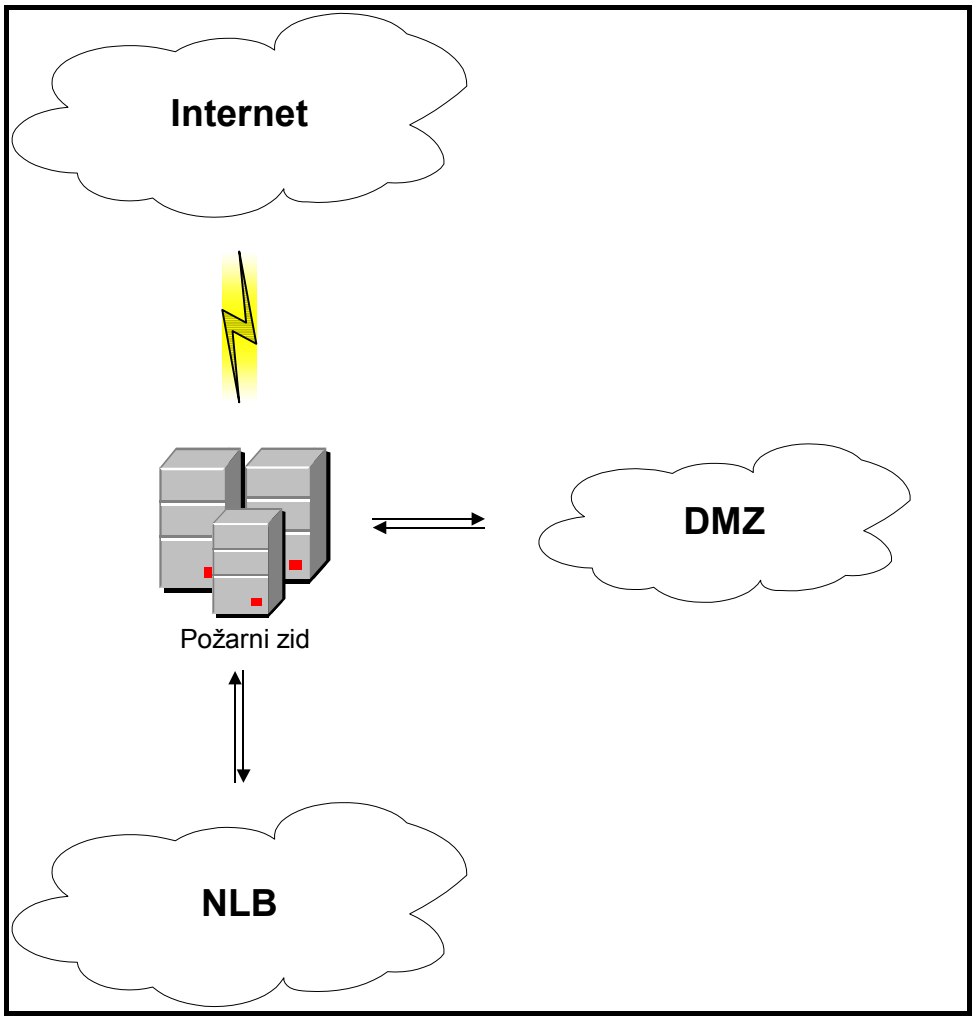
V NLB imamo veliko osebnih računalnikov, povezanih v lokalno mrežo (LAN – Local Area Network). LAN se nanaša na skupino računalnikov, ki so opremljeni s primernim mrežnim vmesnikom, programsko opremo in si priključeni s kablom delijo aplikacije, podatke in periferne enote (Schnaidt, 1992, str. 14).

Za prenose podatkov po mreži uporabljamo tudi multiplekser. Je zelo razširjena in uporabna naprava za povečanje učinkovitosti prenosa. Osnovni namen je, da več napravam dovoljuje delitev skupnega prenosnega medija, s tehniko delitve fizičnega kanala, na večje število takih ali drugačnih logičnih in neodvisnih kanalov (Simčič, 1988, str. 123).

V banki uporabljamo še kup manjših strežnikov, ki služijo za različne naloge:

- požarni zid (firewall), ki je v NLB zelo zmogljiv, saj mora sistem obvarovati pred kupom nevšečnosti, kot so virusi iz interneta, varnostni dostop v sistem, preprečevanje dostopa uslužbencem na nelegalne strani Interneta…,
- za registracijo zaposlenih ob prihodih in odhodih v službo,
- za skrbništvo lokalnih omrežij,
- …

Slika 8: Demilitarizirano okolje



V računalniških mrežah je demilitarizirano okolje (DMZ - DeMilitarized Zone) majhno nevtralno področje med privatnim omrežjem podjetja in javnim omrežjem (slika 8). Namen je preprečevanje direktnega dostopa do podatkov zunanjim uporabnikom (DMZ, 2002), ki lahko preko požarnega zidu dostopajo le do demilitariziranega okolja. Dostop naprej do notranje mreže NLB jim ni dovoljen.

4. REPLIKACIJA

Replikacija je proces, ki je namenjen podvajanju podatkov. V NLB ga uporabljamo pri selitvi sprotnih podatkov na spletni portal. S tehničnega vidika nam zagotavlja avtonomno delovanje in izpolnjuje vse predpisane varnostne zahteve.

Poslovni svet je vse bolj zainteresiran za replikacijo iz dveh razlogov (Hotek, 2002, str. 55):

- posli postajajo vse bolj geografsko razpršeni, zaposleni pa morajo še vedno imeti dostop do manjšega dela podatkov,
- podjetja hranijo velike količine podatkov, ki bi se lahko v primeru sistemske napake hitro restavrirali.

Za lažjo predstavo pojma sem izbral dve definiciji:

1. Replikacija je proces vzdrževanja izbranih podatkov na več kot eni lokaciji. Vsebuje kopiranje določenih sprememb iz ene lokacije (izvor - source) na drugo (cilj - target) in sinhronizacijo podatkov. Izvor in cilj sta lahko na istem ali različnih računalnikih v distribuiranem omrežju (Replication Guide and Reference, 2000, str. 3).
2. Replikacija je proces kopiranja in vzdrževanja baznih objektov v večih bazah podatkov, ki jih definira distribuiran bazni sistem. Spremembe, ki se pojavijo na eni strani, se zajamejo in lokalno shranijo, preden se pošljejo na oddaljene lokacije. Replikacija zagotavlja uporabnikom hiter dostop do podatkov, aplikacijam pa zagotavlja dostopnost zaradi dodatnih podatkovnih virov (Replication, 1999).

Hitro opazimo, da sta definiciji zelo podobni. Ista rdeča nit se nadaljuje tudi skozi ostale opise in razlage, ki sem jih uspel izbrskati skozi literaturo in internetne vire.

Seveda lahko podvojene podatke na različnih ali istih sistemih vzdržujemo na več možnih načinov. Replikacija je ena izmed uveljavljenih metod, ki sem jo pobližje spoznal. Iz izkušnje lahko potrdim vrednost definicij. Proces replikacije namreč s svojim podatkovnim modelom in programsko podporo zagotavlja konsistentnost podvojenih podatkov. Verjetno se pojavi vprašanje smiselnosti podvojevanja na istem sistemu. Na podlagi lastnih izkušenj poznam dva razloga:

- selitev dislociranega sistema na centralni računalnik je s prenosom objektov, podatkov in programske podpore že samo po sebi velik zalogaj. Selitev sicer omogoča ukinitev propagacije, vendar jo je v prvi fazni smiselno ohraniti in zmanjšati možni element napake,

- podvojevanje DB2 kataloga (katalоških tabel) je lahko učinkovit način razbremenitve primarnih tabel, saj se lahko večina ali del programov pri dostopu sklicuje na repliko. Obvezno pa moramo poskrbeti za primerno hitro osveževanje podatkov.

4.1 Ustreznost replikacije

Sistem replikacije mora zadostiti sledečim poslovnim zahtevam (Replication Server, 2002, str. 2):

- visoki dostopnosti podatkov: replikacija mora biti zanesljiva in ne sme izpostavljati poslovnih operacij računalniškim napakam,
- konsistentna dostava informacij: distribucijski sistem mora zaščititi podatkovno integriteto,
- visoke zmogljivosti: replikacija ne sme bremeniti podatkovnega vira, izkoristiti mora učinkovitost mreže in mora omogočati optimizacijske metode pri dostopanju do lokalnih podatkov,
- enostavno centralizirano skrbništvo: skrbnik mora biti sposoben enostavnega upravljanja z distribuiranimi komponentami replikacijskega sistema iz enega delovnega mesta,
- heterogen dostop do izvornih podatkov: replikacija naj bi bila sposobna seliti podatke skozi podatkovne vire različnih proizvajalcev,
- lokalna avtonomija: vsaka stran, ki dobiva podatke preko replikacije, bi se morala prosto odločati, katere podatke želi dobivati, kako bo te podatke prikazovala, do njih dostopala in jih spreminjala.

Četudi lahko podatkovna replikacija reši mnogo problemov (skrajša, poenostavi in tako pohitri podatkovne tokove) med centralnimi in distribuiranimi SUBP, je pomembno prepoznati situacijo, v kateri replikacija ni idealna. Morda je ne želimo uporabljati v naslednjih primerih (What is Database Replication?, 2001):

- obstaja veliko število sprememb na veliko replikah – aplikacije, ki zahtevajo pogoste spremembe na obstoječih zapisih v različnih replikah, lahko naletijo na več konfliktov, kot aplikacije, ki samo vnašajo nove zapise. Aplikacije z mnogo konflikti zahtevajo več skrbniškega časa, ker morajo biti težave odpravljene ročno,
- konsistentnost podatkov je vseskozi kritična – aplikacije, ki se nanašajo na ves čas pravilne informacije, kot so denarni tokovi, letalske rezervacije in sledenje paketnim pošiljkam, ponavadi uporabljajo transakcijsko metodo. Podatki, ki se izmenjajo med replikama skozi sinhronizacijo, so rezultat transakcije in ne transakcija sama,

- obstaja potreba po takojšnem prenosu sprememb vsem uporabnikom – aplikacije, ki izvajajo pogoste spremembe, niso dobri kandidati za replikacijo,
- uporabniki si ne morejo privoščiti prekomerne obremenitve SUBP – velikost naše glavne baze in replike lahko niha zaradi sprememb na podatkih ali sprememb na podatkovni strukturi. Nekateri uporabniki imajo verjetno omejitve glede porabe pomnilniškega prostora. Replicirana baza je lahko večja od originalne in porabi ves prostor na disku.

4.2 Lastnosti replikacije

Replikacija je samostojen produkt, ki je integriran v strukturo SUBP, kot na primer pri DB2, SQL Server-ju, Oracle-u. Pri namestitvi okolja ponavadi ni aktiven, saj se ga pogosto ne uporablja. V procesu vzpostavitve replikacije se le-ta avtomatično aktivira (kreira potrebne objekte in doda funkcionalnosti). V končni fazi to pomeni nekaj več porabljenih virov operacijskega sistema, ki pa ponavadi ni kritično obremenjujoča, jo je pa potrebno spremljati in nadzorovati.

Osnova je fizični podatkovni model, ki ga je potrebno vnesti v strukturo baze. Praktično to pomeni izgradnjo vseh pripadajočih logičnih (tabele, baze) in fizičnih (tabelarični prostor, indeksi) objektov. Šele od tu naprej lahko pričnemo z tvorjenjem definicij replikacije, ki se nanašajo na podatke v tabelah.

Vsaka stran (izvor in cilj) v replikacijskem okolju ima svoj katalog (opisan v prilogi na str. 1 in 2). To je zbirka tabel, ki vsebujejo skrbniške informacije o repliciranih objektih in grupah. Vsak strežnik, ki sodeluje v replikacijskem okolju, lahko avtomatizira replikacijo objektov z uporabo informacij iz svojega kataloga (Database Replication, 1999).

Replikacija postaja kritično orodje za zagotavljanje visoke razpoložljivosti, preživetja in učinkovitosti za aplikacije, ki dostopajo do podatkov. Za zagotavljanje uporabne replikacije moramo rešiti kompleksen problem vzdrževanja podatkovne konsistence med vsemi replikami (Yair, Cipirian, 2002, str. 1).

Močno razvejan sistem nam vedno ne omogoča dostopa po mreži do vseh podatkov (Stodder, 2003). Za ustrezno delovanje je potrebno prenesti nekatere podatke lokalno do uporabnika.

Če smo načrtovali aplikacije za več uporabnikov, lahko replikacija uporabnikom dostopnost do podatkov izboljša. Z uporabo replikacije lahko reproduciramo podatke iz baze in s tem omogočimo delo s svojo repliko baze v istem času različnim uporabnikom. Čeprav so replike lahko locirane na različnih delovnih postajah ali v

različnih pisarnah, ostajajo sinhronizirane med seboj (Why Use Database Replication?, 2004).

Zagotoviti je potrebno, da je proces propagacije skladen z aplikacijskim okoljem. Lahko bi namreč prišlo do nekaterih sistemskih sprememb, ki bi povzročile nekompatibilnost in posledično prenehanje ali napačno delovanje.

Pri spremembah nad podatkovno bazo se izvaja beleženje, iz katerega se nato izvrši prepis v vmesne tabele (proces zajemanja). Drug proces (proces dodajanja) nato periodično preverja vmesne table in izvaja spremembe na cilju ter sproti zagotavlja konsistentnost.

Zaradi realne možnosti odpovedi mrežnih povezav ali komponent, uporablja replikacija zanesljivo tehnologijo hranjenja in posredovanja podatkov. Podatek se posreduje, če je dislocirana enota dostopna. Če pa postane nedostopna, se shrani, dokler se povezava ponovno ne vzpostavi (izvede se samodejna sinhronizacija). Aplikacije lahko v času nedostopnosti posameznih sistemov normalno operirajo z lokalnimi podatki (Replication Server, 2002, str. 1).

Propagacija prav tako pomeni prenos in zamudo. Potrebno jo je nadzorovati. Le tako lahko spremljamo, če se procesi prenosa podatkov izvajajo korektno po pričakovanjih. Ne smemo:

- mešati repliciranih tabel s procesom replikacije. Replicirane tabele so objekti v SUBP,
- uporabljati uslužnostnih programov, ki ne opravljajo beleženja.

Spremembe, ki jih naredijo potrjene transakcije, se leno propagirajo iz primarne na sekundarno stran. To pomeni, da se propagacija pojavi nekaj časa po spremembi, potrjeni na primarni strani. Uporabljeni so različni mehanizmi za generiranje sprememb (za izvedbo propagacije) iz beleženja (prožilci, vohljači). Spremembe se neprestano beležijo po FIFO (First In First Out – sprememba, ki se prej pojavi, pride prej na vrsto) metodi (Daudjee, Salem, 2002).

Poudarek je na lahkotnosti uporabe in robustnosti. Nove registracije, spremembe in brisanja se lahko izvajajo poljubno brez potrebe po izključitvi zajemanja. Če bo prišlo do napak, ne bo trpel proces zajemanja, ampak bodo posamezne registracije neaktivne (Hamel, 2004).

Vsak večji ponudnik SUBP ima takšno ali drugačno replikacijsko rešitev in tudi ostali ponujajo alternativne metode za replikacijo podatkov. Replikacijski produkti zahtevajo resen pristop. Ne smemo nasedati lahkim konfiguracijam nekaterih izdelkov. Četudi uporabniški vmesnik poenostavlja implementacijo in skrbništvo, replikacija zahteva

previdno analizo in načrtovanje. Prav tako je pomembno celotno razumevanje mehanizma, ki ga SUBP uporablja za replikacijo (Thompson, 1997).

Pred očmi moramo vedno imeti razloge za nastavitve SUBP. Zato je najbolje, da jih dokumentiramo in dodamo diagrame mrežnih povezav med posameznimi bazami, definicije podatkov, ki jih potrebujemo replicirati in način replikacije. Tako dobimo dober vpogled nad nalogo, ki nas čaka (Thompson, 1997).

Problematika spreminjanja podatkovnega modela (tabel, primarnih ključev) ni nedolžen proces. Ob spremembi (na primer dodamo nov stolpec) je edini način za uspešno repliciranje novega stolpca zrušitev obstoječe strukture in njeno ponovno tvorjenje. Če je izvorna tabela velika, lahko proces prenašanja podatkov poteka tudi več ur, preden prenese vse preko mreže (Nanda, 2002). V primeru izvajanja sprememb ne smemo pozabiti, da se nam proces lahko izjalovi, kar je lahko posledica napačno tvorjenih definicij (naša nedoslednost ali napaka v generatorju) ali drugih napak v delovanju. V tem primeru je povratek najboljša rešitev, če smo seveda uspeli ohraniti prejšnjo strukturo, definicije, podatke… Zato morajo biti taki posegi vedno skrbno načrtovani in testirani.

Če je naš cilj visoka zmogljivost z uporabo replikacije, potem je potrebno razmisliti, kako bi bilo potrebno zasnovati bazo. Upoštevati bi morali predvsem dve stvari:

- da objavljamo samo tiste podatke, ki jih potrebujemo. Lahko namreč uporabimo vertikalno in horizontalno filtriranje in tako omejimo količino podatkov,
- da optimiziramo količino indeksov. Več indeksov namreč obstaja, počasnejši je replikacijski proces, ker mora biti vsak nov zapis indeksiran (Tips for Performance Tuning SQL Server Replication, 2001). Na cilju se namreč izvaja vnos podatkov v tabelo in tudi v vsak posamezen indeks.

Obstajajo določene tehnične zahteve za distribucijo aplikacij. Pozorni moramo biti, na primer pri pošiljanju podatkov po mreži (obremenjenost omrežja). Slepega prenašanja celotne baze v enem zamahu si verjetno ne želimo (paketne obdelave lahko spremenijo velike količine podatkov v eni transakciji). Torej moramo imeti tehnologijo, ki omogoča rezanje podatkov in s tem zagotoviti uporabniku podatke, ki jih potrebuje. Tako zagotovimo večjo efikastnost in se izognemo konfliktom (Johnston, 2003).

Replikacijski niz lahko vsebuje samo eno glavno bazo in poljubno število repliciranih. Le na glavni bazi so dovoljene spremembe. Če se spremeni katerokoli polje v vrstici, se ves zapis markira kot spremenjen in se kot tak prenaša v času sinhronizacije na replicirane baze. Polja, ki vsebujejo velike objekte (LOB – Large Object), se zaradi svoje obsežnosti ne prenašajo, razen če so spremenjena (Dove, 1999). Pozorni moramo biti na:

- primarni ključ – zapis je zavrnjen, če hočemo podvojiti zapis z istim enovitim ključem,
- istovetnost tabel – zapis vsebuje vrednost (eno ali več), ki ne odgovarja definiciji polja,
- zaklepanje – sprememba zapisa ni mogla biti izvedena v času sinhronizacije, ker je drug uporabnik zaklenil tabelo. Sinhronizacija se ponavlja, dokler se zaklepanje ne sprosti.

Nikoli ne smemo pozabiti na vzdrževanje kontrolnih tabel replikacije. Večino časa namreč namenimo tabelam na izvoru in cilju, pogosto pa spregledamo kontrolne tabele replikacije. V bazi namreč funkcionirajo kot običajne tabele z mnogo vnosi in brisanjem podatkov. Za boljše delovanje zajemanja in dodajanja in možnost povrnitve podatkov od padcu sistema prav tako potrebujejo primerne varnostne kopije, reorganizacije, statistiko… (Building the Operational Data Store on DB2 UDB Using IBM Data Replication, WebSphere MQ Family, and DB2 Warehouse Manager, 2001, str. 68).

4.2.1 Energična in lena propagacija

Propagacija se lahko zgodi znotraj ali zunaj transakcijskih mej. V prvem primeru govorimo o energični (eager), v drugem pa o leni (lazy) replikaciji. Energična omogoča detekcijo konfliktov preden transakcija izvede potrditev. Ta pristop zagotavlja podatkovno konsistentnost vnaprej in posledično močno vpliva na povečano obremenjenost komunikacij ter slabše odzivne čase (Kemme, Alonso, 2000, str 336). Tudi energična propagacija beleži slabe zmogljivosti zaradi distribuiranega zaklepanja, dvofaznega potrjevanja in posledično pogostejšega medsebojnega zaklepanja pri dostopu do podatkov. Algoritmi replikacije ne zagotavljajo, da se operacije pojavljajo v istem vrstnem redu povsod naenkrat, zagotavljajo pa periodično ponavljanje (I&K Resarch, 1998).

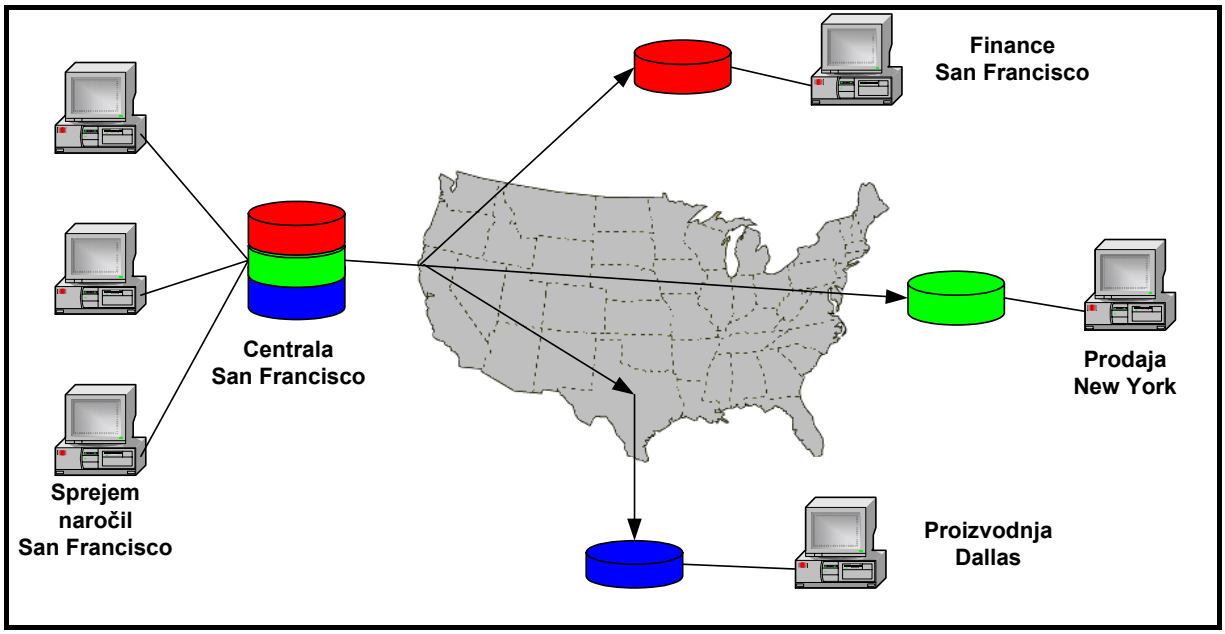
Za ohranjanje kratkih odzivnih časov, lena propagacija zadrži izvajanje sprememb do konca transakcij in jih nato izvede v ozadju. Ker se kopijam dopušča razhajanje, se lahko pojavi nekonsistentnost (Kemme, Alonso, 2000, str 336). Asinhronost ponavadi vključuje zahtevo po vmesnem shranjevanju in pošiljanju podatkov. Operacije, ki morajo biti replicirane iz izvorne SUBP, morajo biti, preden se propagirajo na ciljno bazo (eno ali več), na istem sistemu tudi shranjene (Rennhackkamp, 1997).

Z rastjo SUBP preko grozdov, postaja replikacija vse pomembnejša tema. Izziv je postavitev replikacije, ne da bi resno ogrozili vire sistema. Zaradi te težave današnje baze uporabljajo "leno" (asinhrono) (Darren, 2002, str. 6) propagacijo, ki je zelo učinkovita, lahko pa ogrozi konsistenco (Wiesmann et al, 2002).

4.3 Predstavitev možnosti replikacije

Čeprav je napredna replikacija dostopna že nekaj časa in se jo vidno uporablja, je osnovna replikacija (enostavna osvežitev celotne tabele v enem koraku) uporabljena v velikem številu namestitev. Podatki se prenašajo samo enosmerno, večinoma za dovajanje podatkov v podatkovno skladišče, v dostopne spletne baze in podobno (Hordila, 2002).

Slika 9: Replikacija kot povezava med dislociranimi enotami



Vir: An Architecture for Distributing and Sharing Corporate Information, 2002, str. 8.

Bazna replikacija se pojavlja v različnih oblikah. Nekatere vsebujejo dvofazno potrditev. V tem primeru replike v prvi fazi pošljejo svoje strinjanje z načrtovano spremembo v podatkovni bazi, v drugi pa se izvede replikacija sama. Dvofazna potrditev zahteva nenehno medsebojno delovanje vseh strežnikov in povezav. Ker pa je to težko zagotoviti, se večina organizacij raje odloča za "toplo" različico, ki periodično prenaša spremembe, kot pa "vročo" različico, kjer so vse transakcije po vnosu takoj dostopne (Ray, 1988).

Replikacija da uporabnikom hiter dostop do vseh podatkov, ki jih potrebuje ob majhni ceni aparaturne opreme na poljubni lokaciji. Ker se vsi podatki posledično nahajajo lokalno, je:

- dostop do podatkov poenostavljen,
- poveča se odzivni čas,

- komunikacijska obremenitev se zmanjšuje.

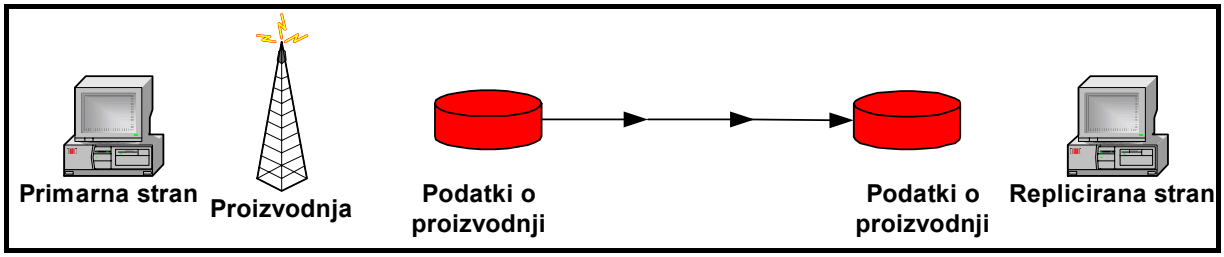
Ker potreba po močnejših strežnikih ne obstaja več, se zmanjšajo stroški strojne opreme.

Replikacijo se lahko uporabi (The Need for Replication Technology, 2004):

- za dokončanje postopne sistemske migracije,
- premikanje podatkov na rezervno lokacijo,
- za zagotavljanje in visoko dostopnost podatkov na našem sistemu,
- za zagotavljanje nemotenega prenosa (sistem za detekcijo in odpravo napak).

Replikacija (vsi člani v procesu dodajanje skupaj ali posamezna ali nobena) se lahko izvede zaporedno v enakih časovnih zamikih ali na določen dogodek (Yakushin, 2002).

Slika 10: Replikacija kot varnostna kopija



Vir: An Architecture for Distributing and Sharing Corporate Information, 2002, str. 9.

Preden se lotimo tvorjenja skript za registracijo tabel, moramo nujno poznati vse možnosti, ki nam jih replikacija nudi. Le tako bomo lahko našli optimalno rešitev, ki bo zadostila poslovnim potrebam:

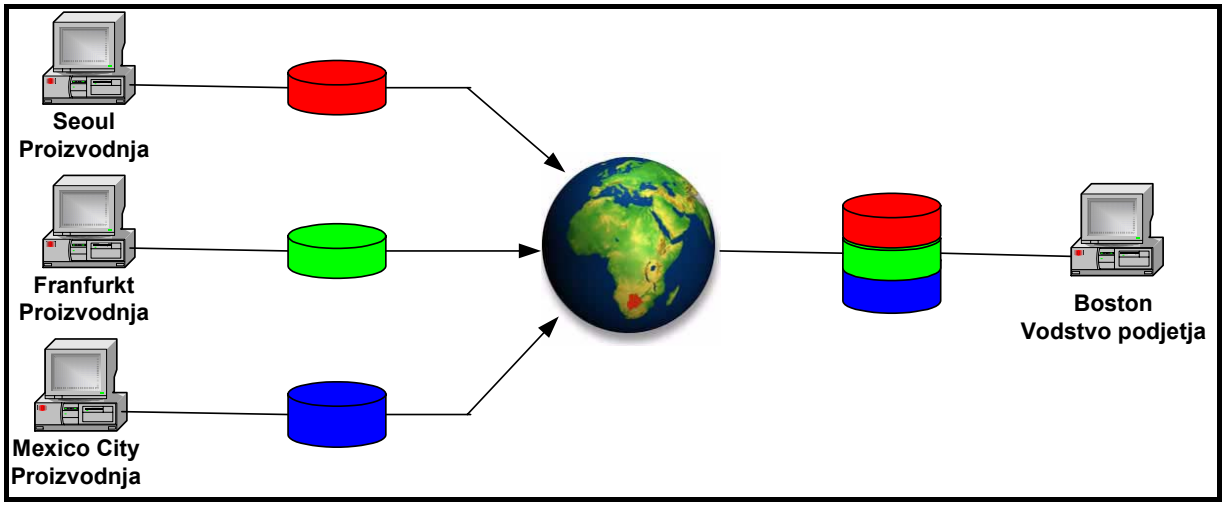
1. replikacija kot povezava med dislociranimi enotami (lokalni dostopi so hitri, podatki so urejeni po regijah, vsaka baza nudi varnostno kopijo drugi – slika 9),
2. replikacija kot varnostna kopija (velike katastrofe, kot so poplave, potresi, požari) – možni hitri preklop na sekundarno bazo (slika 10),
3. za zbiranje poročil – obremenilni procesi se nato izvajajo lokalno in ne obremenjujejo mreži in centralnega sistema, za mobilne uporabnike, podatkovno skladišče (slika 11).

Obstajajo tudi orodja (Sybase Replication Agents™, Data joiner…), ki omogočajo izmenjavo podatkov med SUBP različnih proizvajalcev, med katerimi so največkrat omenjeni trije: IBM DB2, MS SQL – Server, ORACLE.

Medtem ko nekateri proizvajalci že ponujajo grafične vmesnike za lažji nadzor replikacije, drugi to šele obljubljajo. V našem primeru si moramo pomagati z izborom posameznih poizvedb, saj preglednejšega orodja še nimamo.

Podatke lahko zajamemo po ali pred izvedeno spremembo zapisa na posamezni tabeli (Kemme, Alonso, 2000).

Slika 11: Repliciranje podatkov, potrebnih za generiranje poročil vodstvu podjetja



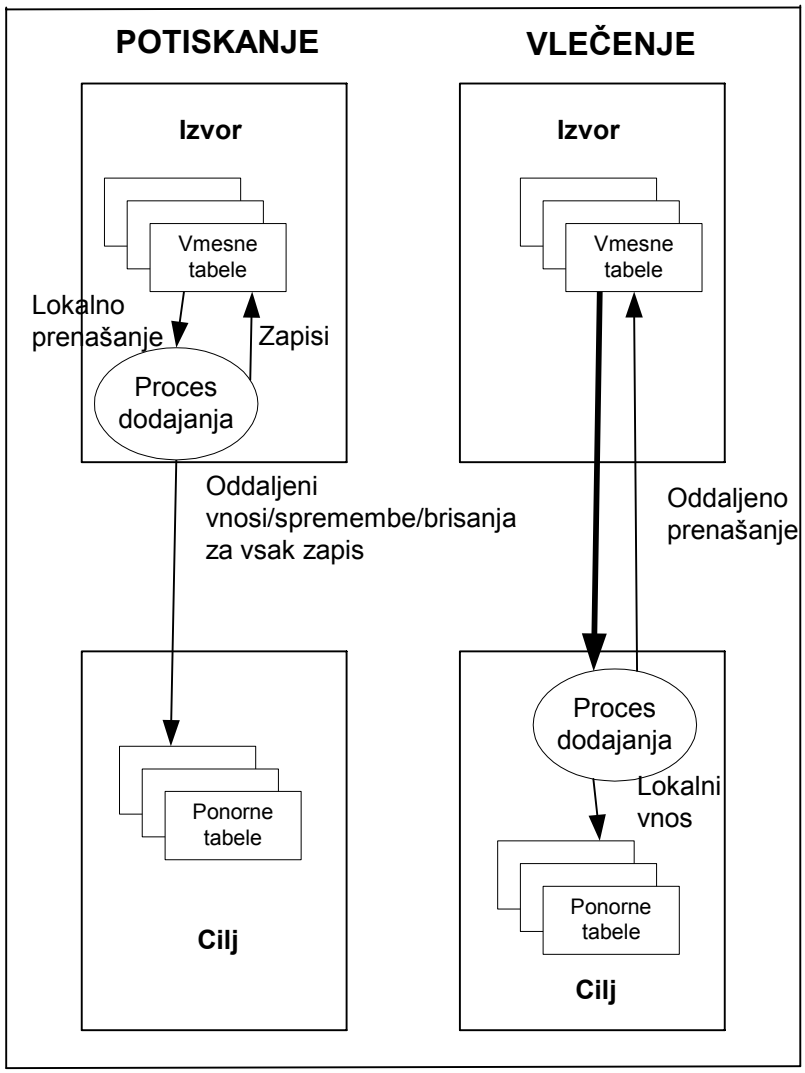
Vir: An Architecture for Distributing and Sharing Corporate Information, 2002, str. 10.

V fazi kreiranja definicij lahko izbiramo med naslednjima dvema možnostima (Kemme, Alonso, 2000):

- uporabniško kopiranje – to je popolno, zgoščeno kopiranje vsebine izvornih tabel, ki pa morajo vsebovati primarni ključ. Tabele na cilju bodo vzdrževale popoln duplikat podatkov originalnih tabel,
- kopiranje s časovno kontrolo - to je popolno, zgoščeno kopiranje vsebine izvornih tabel, ki pa morajo vsebovati primarni ključ in vsebujejo dodatni stolpec tipa timestamp (polje, ki združuje datumsko in časovno komponento) za datumsko in časovno indikacijo opravljene transakcije.

Vmesne tabele, znane tudi kot CD in CDD (Change Data table in Consistent Change Data table), imajo lahko več namenov, a so originalno zasnovane za zagotavljanje vmesnega območja, v katero se beležijo spremenjeni podatki. Proces dodajanja tako ne dostopa do originalnih tabel, temveč vmesnih tabel in prenaša podatke do cilja.

Slika 12: Prikaz potiskanja in vlečenja podatkov v procesu dodajanja



Vir: Building the Operational Data Store on DB2 UDB, 2001, str. 103.

Proces dodajanja lahko uporabi dve metodi, kot jih prikazuje slika 12:

- potiskanje podatkov - izvor pošlje podatke cilju in pri tem lahko čaka ali ne čaka (odvisno od naših potreb) na zahtevo procesa dodajanja. Zaradi potrebe po takojšnji dostopnosti podatkov v realnem času, transakcija replikacije uporablja metodo potiskanja.
- vlečenje podatkov – proces dodajanja periodično ali na nek izvršen dogodek zahteva podatke od izvora. Uporabljamo jo, ko imamo mnogo ciljev in nobene potrebe po takojšnji dostopnosti repliciranih podatkov (Otey, 1999).

V današnjem času ima lahko lokacija podatkov pomemben vpliv na odziv in dostopnost aplikacije. Centraliziran dostop upravlja samo z eno kopijo baze in je zato enostaven. Vendar ima dve glavni pomanjkljivosti (Yair et al, 2002, str. 1):

- problem zmogljivosti ob množičnem dostopu oddaljenih odjemalcev do strežnika in
- problem razpoložljivosti v času, ko strežnik ni aktiven ali ob pomanjkanju povezav.

4.4 Arhitektura in replikacija spletnega portala

V tem poglavju sem opisal svoje izkušnje in postopke pri gradnji propagacije za spletni portal. Pot je bila pogosto trnova, saj sem naletel na prenekatero težavo, ki je ni bilo lahko odpraviti. Prvič sem se srečal tudi s propagacijo velikih objektov in moral razrešiti mnogo nejasnosti. Rezultat vloženega truda je delujoča verzija.

Replikacija je bila izbrana kot najbolj zanesljivo sredstvo za prenos podatkov na spletni portal. Izbor je temeljil na strogih varnostnih zahtevah in naših dotedanjih izkušnjah. Strani spletnega portala se v osnovi generirajo na centralnem računalniku, nato jih s procesoma replikacije prepisujemo na internetni strežnik.

Za doseganje čim večje zanesljivosti delovanja smo postavili dve ločeni okolji – testno in produkcijsko. Takšno okolje tudi zagotavlja integriteto ter izboljšuje operativno delo (Mullins, 2002a).

Pri replikaciji podatkov med različnimi sistemi sem se hitro srečal s problemom, ki ga navaja že Craig S. Mullins (Mullins, 2002a): Upravljanje večih okolij otežuje delo skrbnikom SUBP. Pojavlja se cel kup različnih problemov in rešitev, ki jih je potrebno osvojiti.

Postopek povezave med sistemoma je zahteval posege tudi v nekatere nastavitve parametrov na več nivojih; operacijskega sistema in SUBP. Najprej pa sem se moral spopasti z nastavitvami SUBP. Posebej sem moral biti pozoren na:

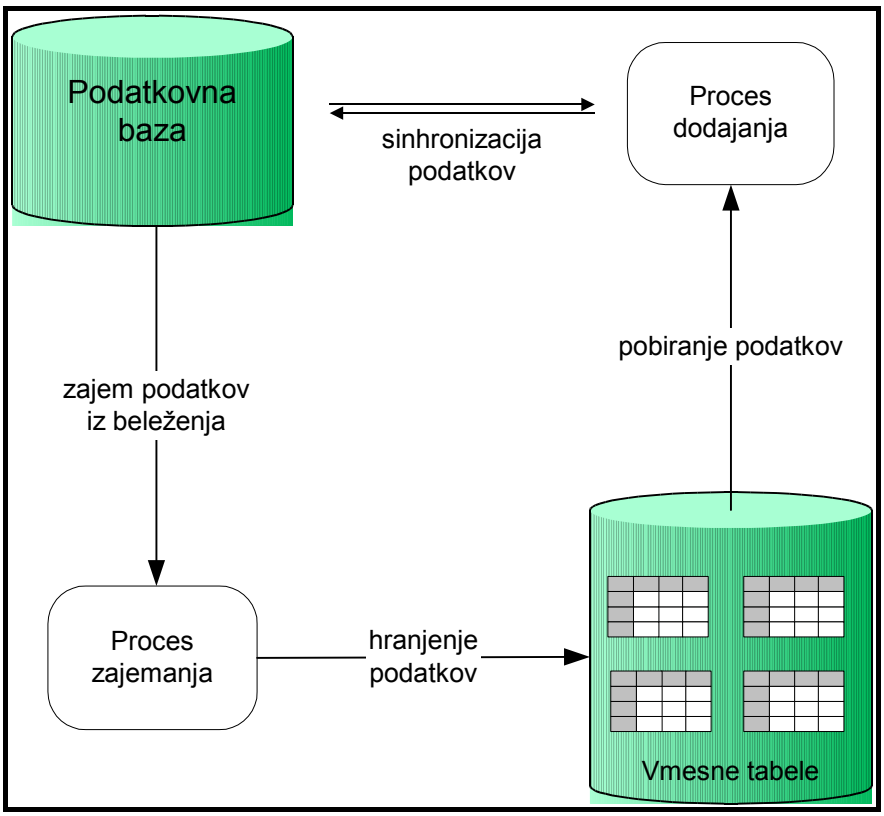
- mrežne nastavitve, ki jih potrebuje sistem za upravljanje podatkovne baze, zahtevajo znanje obeh: SUBP in komunikacij (Instalation guide, 2001, str. 462),
- medpomnilniki in alokacije spomina imajo na splošno globok vpliv na zmogljivosti DB2 baze, ne glede na platformo (Hayes, 2000).

Posebej moram poudariti, da DB2 podpira samo asinhrono replikacijo (Yair et al, 2002, str. 3). Tako sem zasnoval čim enostavnejši proces, ki bi podprl zahteve uporabnikov in bi bilo njegovo vzdrževanje lažje.

4.4.1 Splošne značilnosti

Pri praktični izvedbi sem ugotovil, da napisano v knjigah in navodilih ne drži povsem. Procesa zajemanja in dodajanja na centralnem računalniku sta nedosledna in samodejno pogosto (kar bi po pravilu sicer morala) ne prepoznata definicije za novo replikacijo, čeprav so za to možni ukazi (napisani v knjigi Replication Guide and Reference). Zato je najboljša rešitev, da proces zajemanja in dodajanja pred potrebno akcijo preprosto izklopimo, izvedemo operacijo (zagon skripte za proces zajemanja ali dodajanja) in ju ponovno vklopimo. Uspeh je v tem primeru zagotovljen (če nismo naredili napake kje drugje).

Slika 13: Prikaz delovanja procesov zajemanja in dodajanja



Procesi, programi in procedure morajo vedno biti ločeni in dokumentirani glede na to ali jih uporabljamo v testnem ali produkcijskem okolju. Tako imamo (tudi po imenu) ločeni proceduri za vključitev procesa zajemanja v načinu:

- hladni zagon, ki je primeren samo ob pojavitvi sistemskih težav procesa zajemanja ali SUBP DB2. Ob tem zbriše vso vsebino iz svojih kataloških tabel in prične z novim izvajanjem replikacije. To je impulz procesu dodajanja za popoln prepis vseh podatkov (popolna osvežitev) na ciljne lokacije,

- topli zagon, ki ga uporabljamo vedno ob vnosu novih definicij za propagacijo in kontroliranemu ustavljanju in zagonu SUBP. Interne tabele mu omogočajo spremljanje njegovih akcij, zato je vzpostavitev hitrejša in (s strani porabe virov) cenejša, saj po beleženju pobere samo podatke od zadnje sinhronizacije.

Večkrat so se že pojavile hujše težave v delovanju replikacije, ki so zahtevale takojšen poseg. Ker sem imel postopke že vnaprej pripravljene, ni bilo potrebno iskati ustrezne procedure in jo pred zagonom spreminjati (prilagajati). Stresne situacije že same povzročijo dovolj napak, zato si ne smemo dodatno oteževati dela. Če je zagon opravljen v hladnem načinu, nam to napiše ob startu procedure, drugače ne.

Vmesna tabela vsebuje neomejen izpis spremenjenih podatkov, ki jih proces zajemanja prebere iz DB2 beleženja. Obstaja samo ena CD tabela za vsako replicirano izvorno tabelo.

4.4.2 Sistemske tabele

Proces replikacije za svoje delo nujno potrebuje tudi niz tabel, kamor shranjuje vse podatke, potrebne za nemoteno delovanje. Ti objekti v začetni fazi niso kreirani, postopek izvedbe pa se razlikuje glede na tip baze:

- Centralna baza - ukazi za kreiranje so spravljeni v posebni knjižnici na centralnem računalniku. Skripte zaženemo ročno. Pred tem jih lahko tudi popravljamo in prirejamo, vendar moramo dobro poznati strukturo in namen. Če se namreč tabele ne kreirajo pravilno, replikacija ne bo delovala.
- Lokalna baza – postopek je tu bolj enostaven, saj je avtomatiziran. Ko se lotimo prvih korakov (definicij) za replikacijo, sistem to zazna in samodejno kreira potrebne objekte.

4.4.3 Zahteve

Spletni portal je postavljen v demilitarizirano okolje, ki sem ga že opisal in ga prikazuje slika 8. Na njem delujejo vsi potrebni procesi, ki omogočajo nemoteno delovanje, oziroma svetu odpirajo okno z različnimi informacijami s področja delovanja NLB (letna poročila, splošne informacije, bančne informacije, ponudbe, tečajne liste…).

Kot vsak tak sistem, je bil tudi naš podvržen strogi varnostni kontroli. Pogosto so prav spletni portali tarča napadov in vdorov. Zato smo določili specifično arhitekturo, ki podpira varnostne zahteve:

- glavna baza je na centralnem računalniku,

- za potrebe spletnega portala se postavi ločen strežnik s svojim operacijskim sistemom in lokalno bazo,
- vsi podatki se ažurirajo na glavni bazi, spremembe pa se preko replikacije prenašajo na lokalno bazo, kjer so dostopni zunanjemu okolju.

Iz zgornjega lahko vidimo, da je proces nadzora, prirejanja in popravljanja podatkov možen izključno iz notranje mreže NLB. Da bi zadostili še preostalim varnostnim elementom, bi morala replikacija podatke na spletni portal potiskati (push metoda). V tem primeru bi moral centralni računalnik postati tudi kontrolni strežnik (torej proces zajemanja in dodajanja na istem strežniku), kar pa v našem primeru ni bilo mogoče. Zato smo potrebovali še dodatni strežnik.

Zaradi možne nekonsistentnosti podatkov na lokalni bazi, so bili v aplikacijo vgrajeni dodatni varnostni mehanizmi, ki preprečujejo tovrstne pojavitve. Replikacija sama namreč služi samo za prenos podatkov in nima vgrajene napredne logike, ki bi omogočala prikazovanje samo določenih podatkov. Podatke, ki jih mi želimo prenašati, moramo zagotoviti sami. Replikaciji nato povemo samo izvor in cilj.

4.4.4 Kontrolni strežnik

Kontrolni strežnik opravlja zelo pomembno funkcijo prenosa podatkov v obe smeri. To v našem primeru pomeni, da mora biti aktiven vsaj en proces dodajanja. Informacijo o uspešnosti/neuspešnosti izvajanja pa beleži v svoje tabele, kar pomeni, da potrebuje v ozadju svojo delujočo lokalno bazo. Procesa dodajanja namreč ne moremo postaviti samostojno. Najprej moramo namestiti SUBP (lahko z minimalno konfiguracijo) in nato aktivirati proces dodajanja.

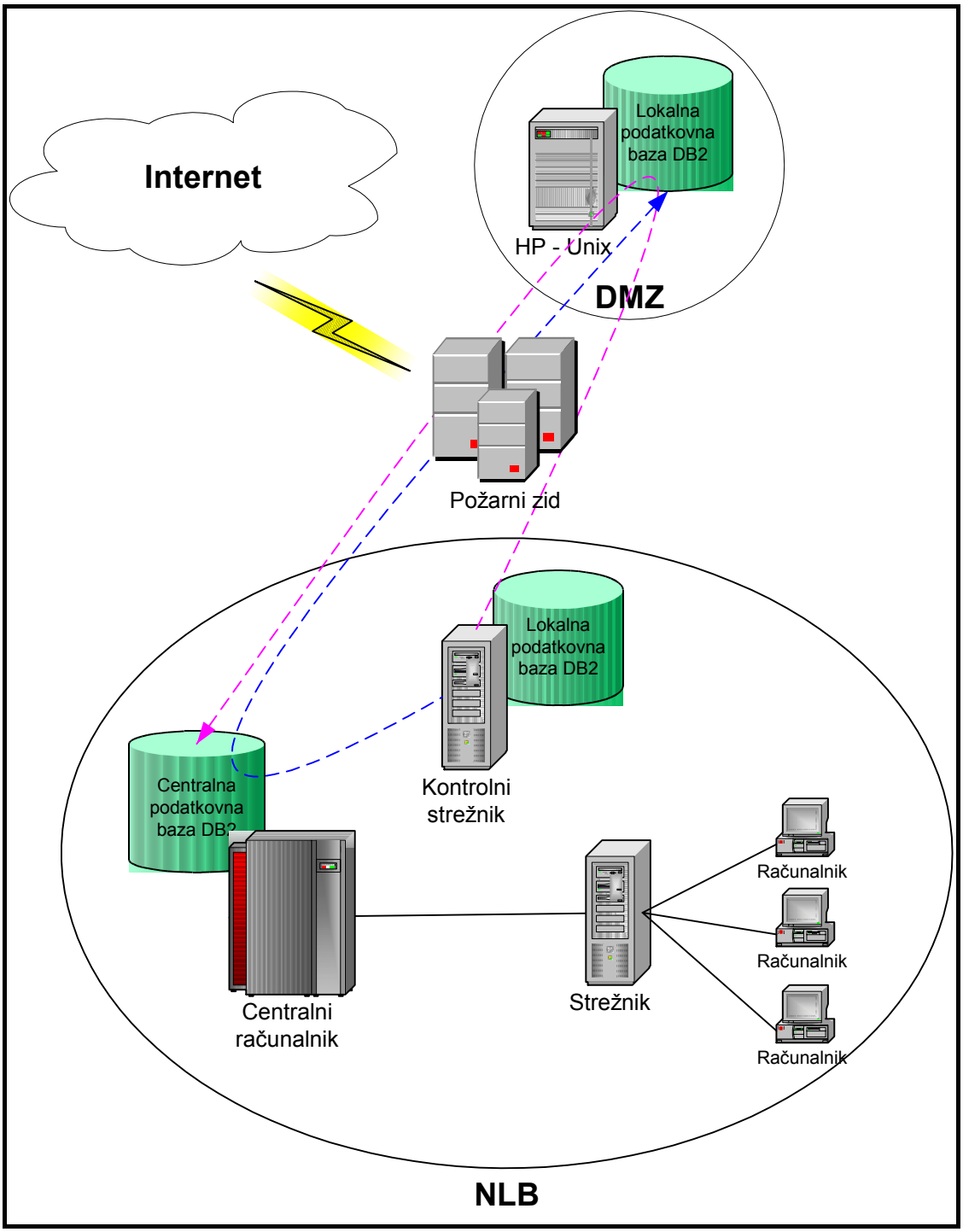
Za svoje sprotno delovanje sicer ne potrebuje veliko virov. Lahko pa nas preseneti (kakor je tudi mene) s trajanjem v primeru masovnih prenosov, ko smo osveževali vse podatke na cilju. Glavnino težav so povzročali veliki objekti.

Pravilna uporabniška identifikacija, ki jo replikacija uporablja za prijavljanje na ciljno bazo, mora biti definirana v replikacijski povezavi. Uporabniško ime za vzdrževalca mora biti kreiran v vsaki izvorni in ciljni bazi (Database connections, 2004). Ker se pri nas vse odvija preko kontrolnega strežnika, moramo tej zahtevi prirediti tudi uporabniško identifikacijo.

4.4.5 Izvedba

Ko sem imel potrjen in testiran podatkovni model, sem se lotil postavitve replikacije. Ta vedno poteka v dveh fazah. Najprej izvedemo skripte za proces zajemanja, v drugem delu pa še skripte za proces dodajanja.

Slika 14: Shema replikacije med centralnim računalnikom in spletnim portalom

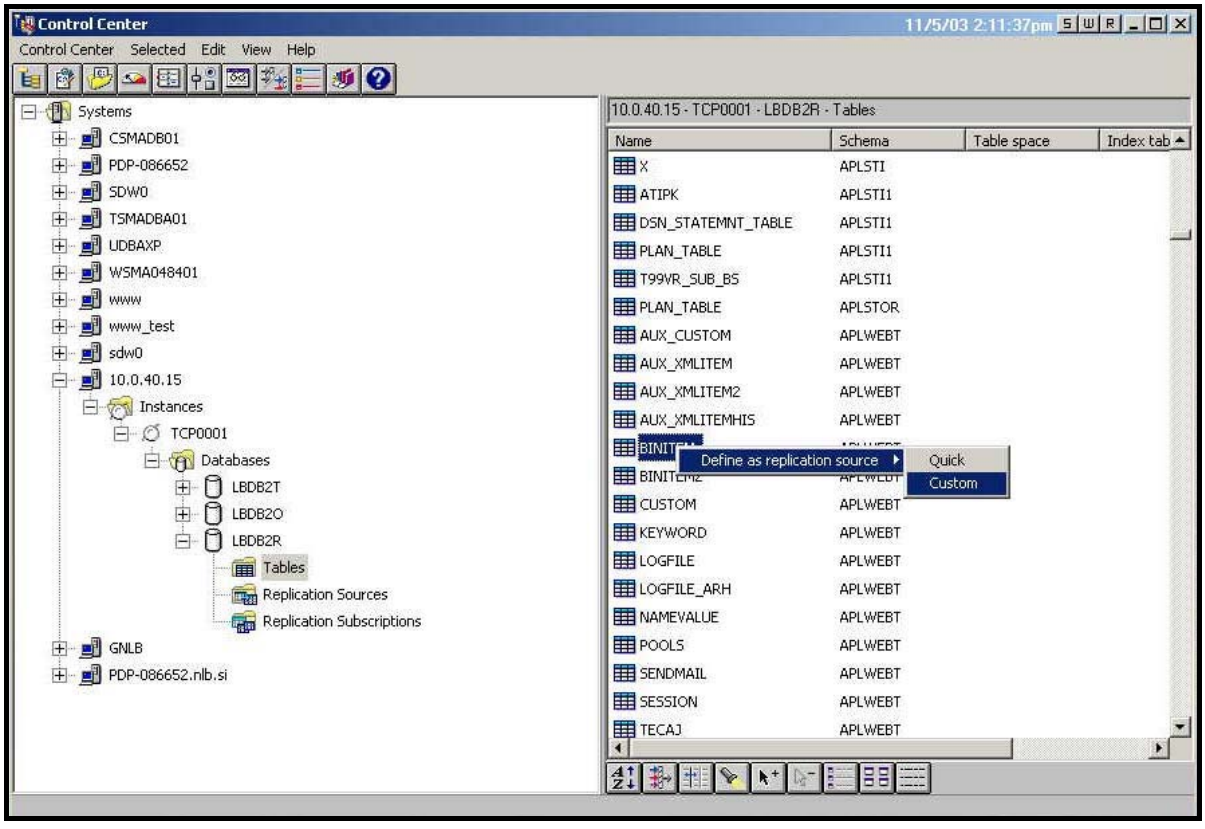


Slika 14 prikazuje replikacijo podatkov na strežnik spletenga portala in nazaj. Na strani NLB omrežja so postavljeni strežniki, ki služijo ažuriranju podatkov za potrebe spletnega portala in kontrolirajo propagacijo. Glavna baza se nahaja na centralnem računalniku in hrani vse potrebne podatke.

Podatki prihajajo na spletni portal preko požarnega zidu, kjer jih nato prikazuje spletna aplikacija. Uporabniki pregledujejo strani spletnega portala ter postavljajo vprašanja, pišejo pripombe in se registrirajo za uporabo tečajnih list… Ti podatki se nato prenašajo nazaj v centralno bazo, se obdelajo in prikazujejo skrbnikom portala. V dnevnem času tako poteka med dvemi ločenimi omrežji precej živahen promet.

4.4.5.1 Registracija v procesu zajemanja

Slika 15: Izbiranje tabel za potrebe replikacije preko nadzornega centra



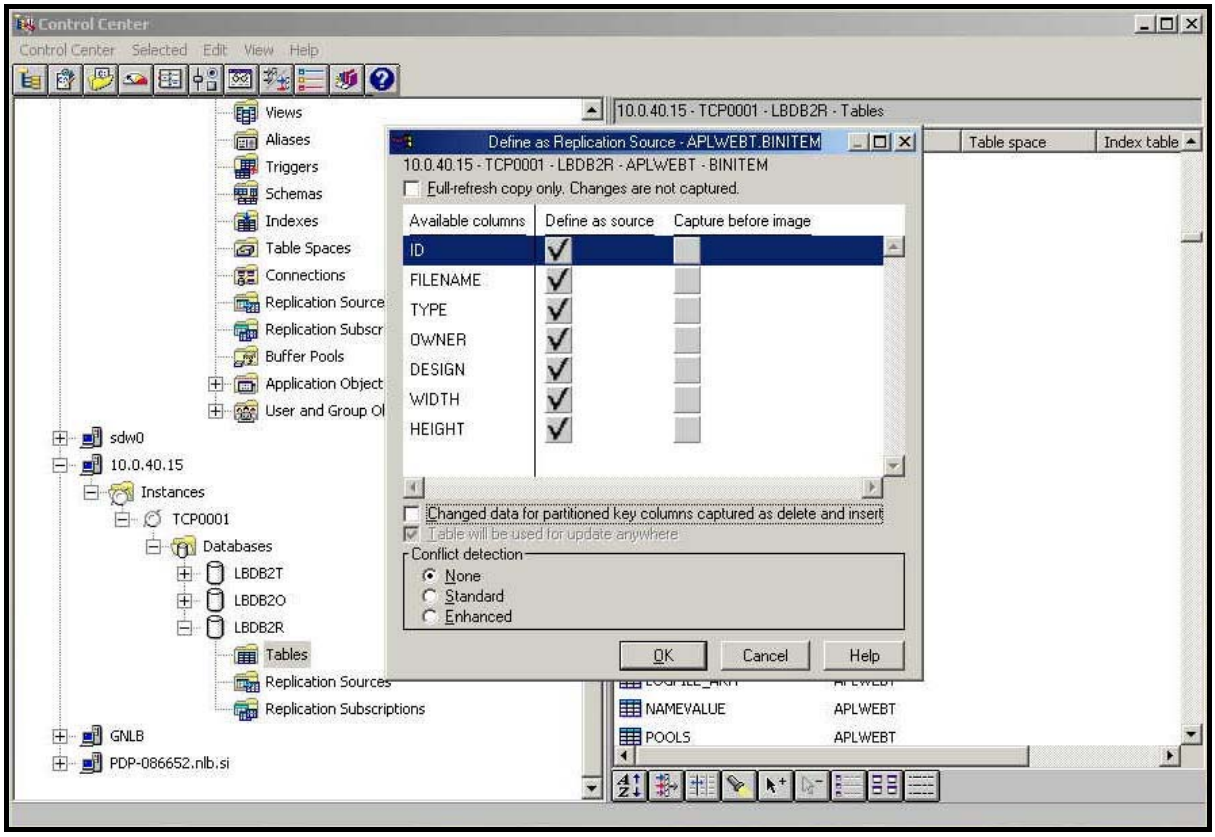
Pri registraciji v procesu zajemanja označimo tabele, ki jih želimo replicirati, in jim dodelimo funkcijo izvora. Pri tem postopku (in seveda tudi pri nekaterih ostalih, kar bom prikazal v nadaljevanju) si lahko učinkovito pomagamo z nadzornim in ukaznim centrom, ki nam zaradi grafičnega prikaza olajša delo (kot ga prikazuje slika 15). Že v

tej fazi si lahko izberemo samo določene stolpce tabele, ki jih bomo prenašali (slika 16). S klikanjem po menijih v ozadju dejansko generiramo stavek za poizvedbo.

Če ne želimo samo prenašati sprememb, se lahko odločimo za vsakokratno popolno osvežitev celotne tabele. Ker proces replikacije podatke prepisuje direktno iz tabele na izvorni strani v tabelo na ciljni, ne potrebujemo vmesne tabele. Popolna osvežitev je primerna za tabele z manjšo količino podatkov, sicer se lahko pojavi prevelika obremenjenost omrežja.

SQL skripte nikoli ne zaženem takoj (kar nam produkt omogoča), ampak jo shranim na disk in naredim varnostno kopijo, ki bi nam prišla prav ob izgubi podatkov v kataloških tabelah replikacije. Skripto tudi vedno pregledam in pri tem preverjam ustreznost in skladnost glede na zahteve.

Slika 16: Izbira stolpcev določene tabele za potrebe replikacije preko nadzornega centra



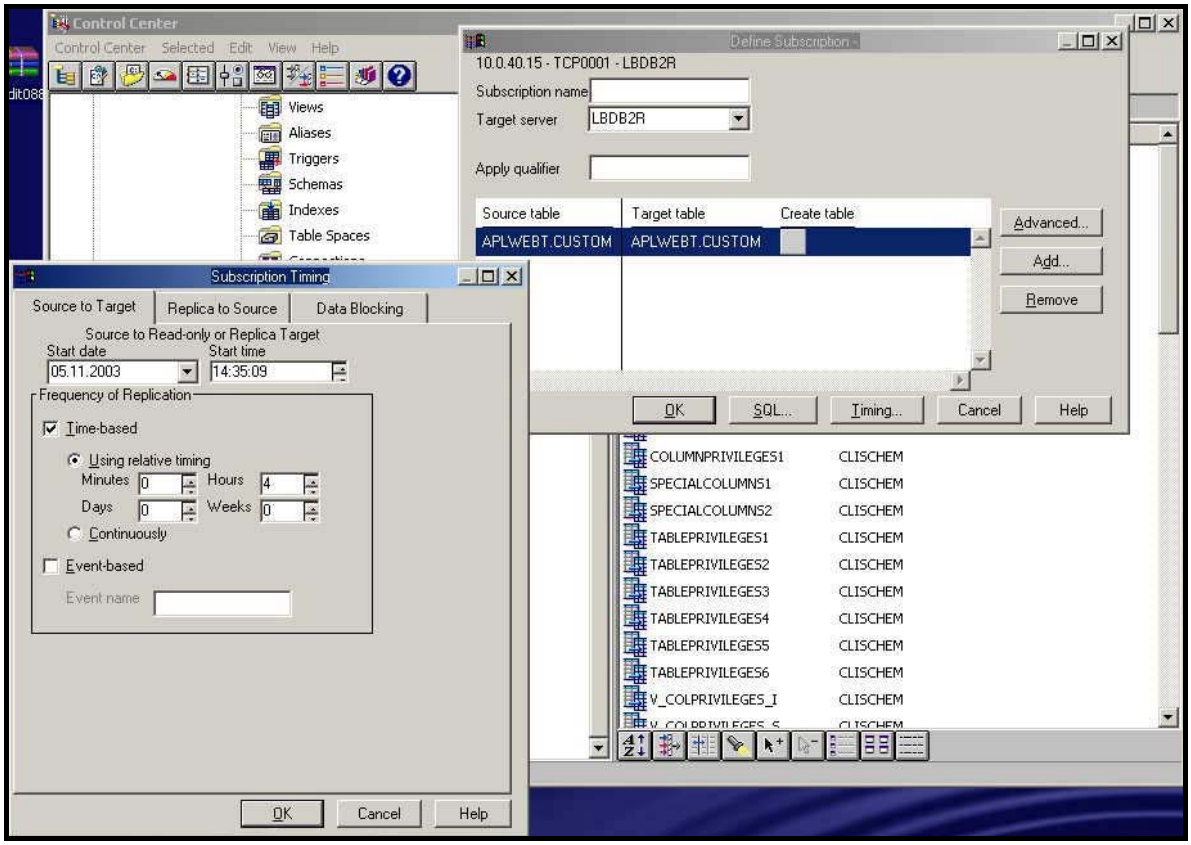
Primer generirane SQL skripte pri registraciji v procesu zajemanja lahko vidimo v prilogi na sliki 3 (glej v prilogo na str. 4). Najprej se izvrši povezava na izvorni SUBP. Z drugim stavkom sistemu podatkovne baze zabeležim novi status tabele (tabela se

propagira). Sistem se odzove z obsežnejšim beleženjem (sicer beleži manj in tako varčuje s prostorom), ki ga potrebuje proces zajemanja. Nato sledi kreiranje vmesne tabele, pripadajočega indeksa in vnos zapisa v kataloško tabelo replikacije za izvorni strežnik. Pomen kataloških tabel replikacije je posebej razložen v prilogi na straneh 1 in 2.

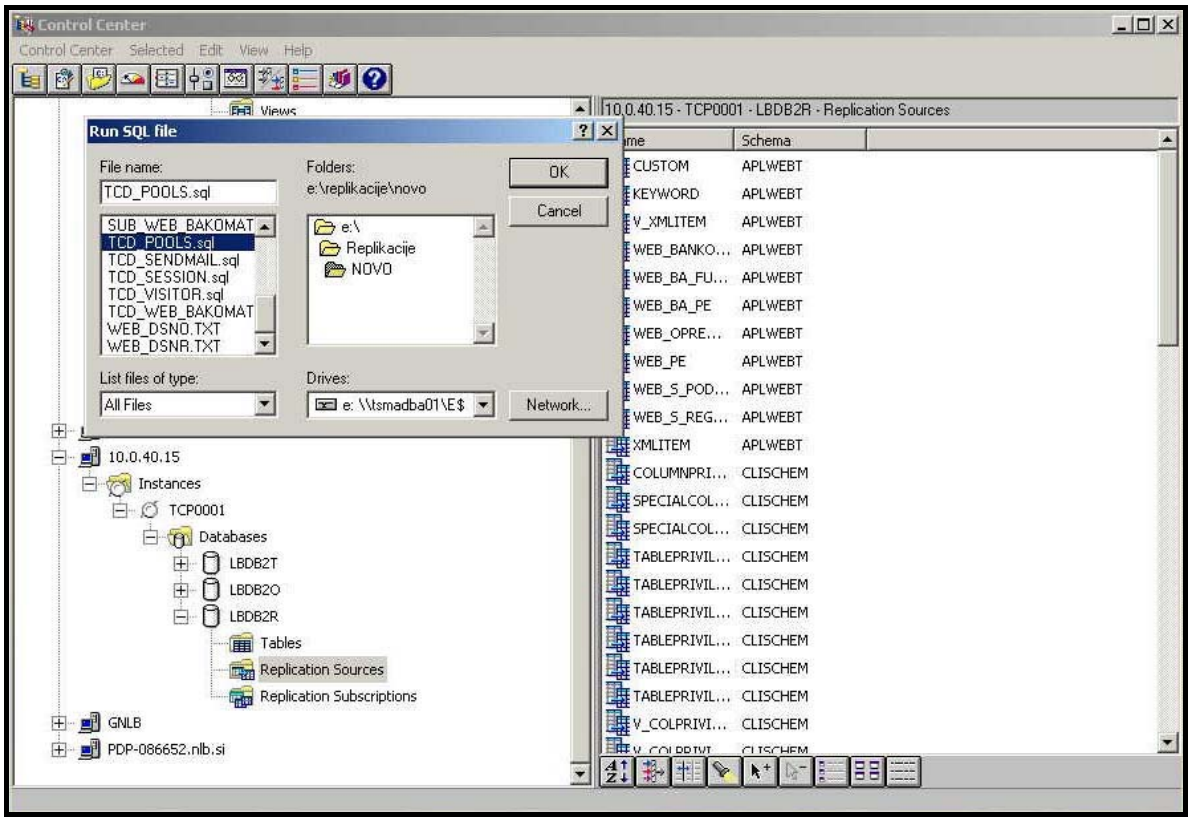
4.4.5.2 Registracija v procesu dodajanja

V drugem delu registrirano tabelo v procesu zajemanja na izvoru povežemo s tabelo na cilju. Ta dva objekta imata lahko različno poimenovanje stolpcev (in tabele), vendar morata imeti sorodna stolpca isti podatkovni tip, sicer replikacija ne bo delovala.

Slika 17: Definicija za proces dodajanja preko nadzornega centra



Slika 18: Zagon SQL skripte preko nadzornega centra



V nadaljevanju lahko izbiramo med načinom proženja preslikav. Imamo dve možnosti (slika 17):

- replikacija na čas – preslikave spremenjenih podatkov se izvajajo periodično na določeni pretečeni interval,
- replikacija na dogodek – preslikava se zgodi samo, ko mi (oziroma aplikacija) to želimo. Ukazni vrstici za poimenovanje dogodka (ime je lahko poljubno) in proženje preslikave za določeni dogodek lahko vidimo v prilogi na sliki 2 (glej v prilogo na str. 3).

Tudi v tem primeru generiramo poizvedbeneno skripto (slika 18), ki jo nato ročno zaženemo. Primer generirane SQL skripte pri registraciji v procesu dodajanja sem prikazal v prilogi na sliki 4 (glej v prilogo na str. 5) in 5 (glej v prilogo na str. 6). Zopet se najprej izvrši povezava na kontrolni SUBP (glej sliko 4 v prilogi na str. 5). Nato sledi niz vnosov v kataloške tabele replikacije za kontrolni strežnik (glej sliko 4 in 5 v prilogi na str. 5 in 6). Na koncu sledi še povezava na izvorni strežnik in vnos zapisa v koordinacijsko tabelo (glej sliko 5 v prilogi na str. 6), ki služi za sinhronizacijo med procesoma zajemanja in dodajanja.

4.4.6 Vzdrževanje

Podatki se v nekaterih sistemskih tabelah replikacije in vmesnih tabelah v vseh okoljih neprestano kopičijo. To prinaša vsaj dve nevšečnosti:

- zmogljivostne težave – veliko pregledovanja zapisov v tabelah, da se proces dodajanja dokoplje do potrebnih podatkov,
- prostorske težave – kopičenje podatkov lahko zapolni rezerviran prostor, kar na koncu povzroči ne delovanje replikacije.

Obema se lahko izognemo s stalnim nadzorom in izvajanjem potrebnih akcij. Zajemanja vsebuje dodatno funkcionalnost brisanja odvečnih podatkov, ki ob vzpostavitvi samodejno ni aktivna. Kakor sem že omenil, oba procesa med replikacijo neprestano komunicirata. Med tem si izmenjujeta tudi podatke o uspešnosti prenosa. Zapisi, ki so bili uspešno preneseni do cilja, proces označi, saj jih ne potrebuje več. Brisanje je zatorej zaželeno, vendar tudi samo obremenjuje sistem. Na centralnem sistemu se izvaja kot paketna obdelava v večernih urah, da ne moti tekočih obdelav, medtem ko je na portalu sprotna (po vsaki potrditvi prenosa).

Za kontrolne tabele replikacije, ki izvajajo beleženje, pa moram poskrbeti kar sam. Zaradi možnih analiz za nazaj, v primeru težav, ohranjam podatke v časovnem oknu zadnjih treh mesecev (ki ga lahko po potrebi ožim ali širim), starejše podatke pa tedensko brišem iz tabele.

4.4.7 Nadzor

Žal za nadzor replikacije nimam na voljo orodij, ki bi mi omogočala boljšo preglednost in pošiljala opozorila v primeru nastalih težav ali celo nedelovanja. Tako za morebitne težave največkrat zvem šele od skrbnikov spletnega portala, kar je sorazmerno pozno. Napaka se namreč že lahko odraža na straneh spletnega portala, kar seveda opazijo tudi uporabniki.

Sam si pomagam z izborom poizvedb (primer na sliki 19), ki jih prožim nad tabelami kontrolne baze v ukaznem centru:

Primer poizvedbe na sliki 19 izpiše vse vitalnejše podatke za vsako posamezno tabelo, ki jih potrebujem:

- ime tabele, ki se propagira,
- količino izvedenih vnosov, popravkov in brisanj za posamezen prenos,
- status replikacije,
- čas, ko se je izvedla sinhronizacija,

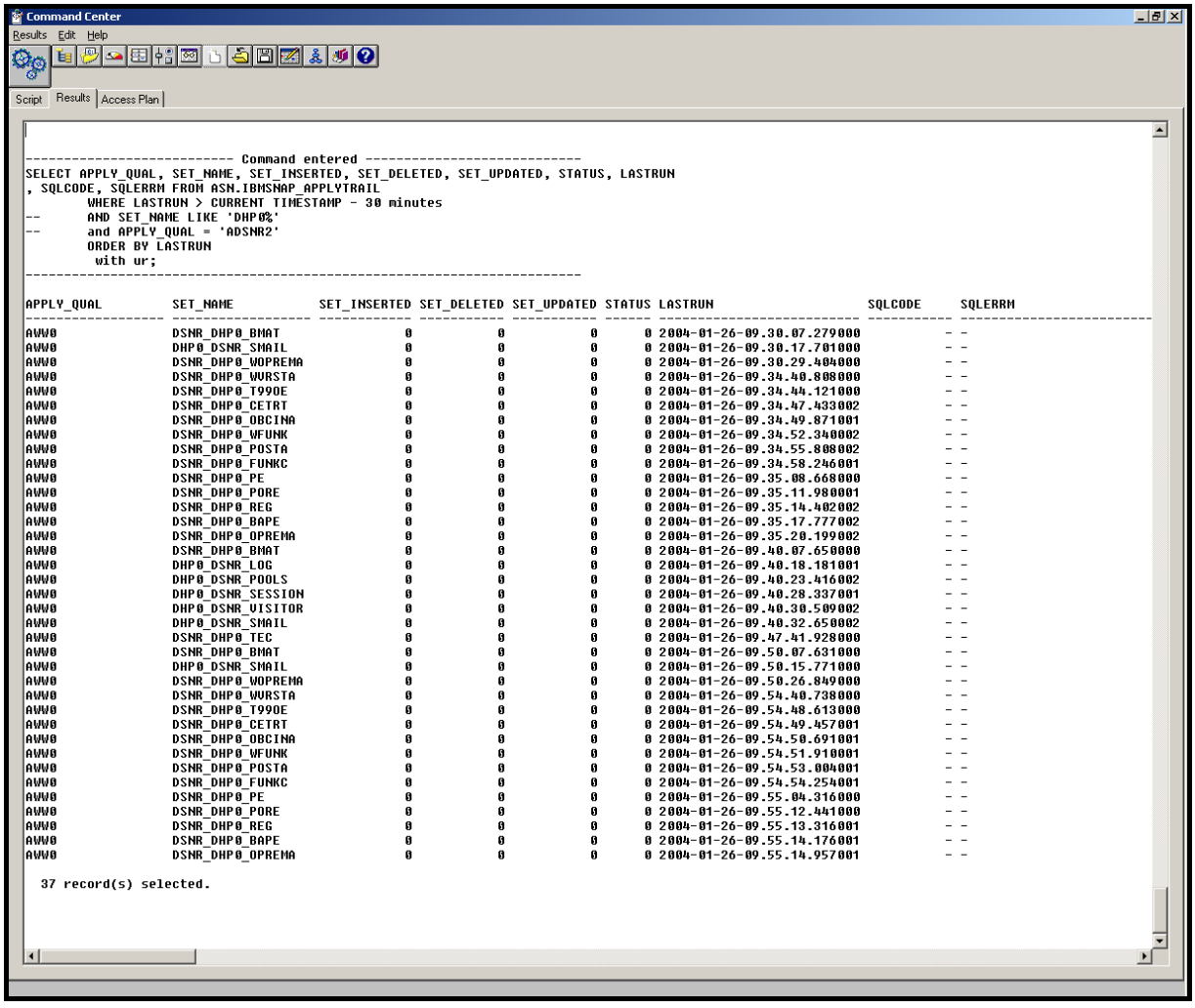
- SQL koda in sporočilo, če je koda manjša od 0 (ničla pomeni uspešen prenos, morebitne težave pa se odražajo s predpono "minus" in ustrezno kodo).

Grafični pregled izpisanih informacij je možen na sliki 20.

Slika 19: Primer poizvedbe nad tabelo, ki opravlja beleženje

```
SELECT
       APPLY_QUAL,  SET_NAME,  SET_INSERTED,  SET_DELETED,
       SET_UPDATED, STATUS,  LASTRUN,  SQLCODE,  SQLERRM
FROM ASN.IBMSNAP_APPLYTRAIL
WHERE LASTRUN > CURRENT TIMESTAMP - 30 MINUTES
ORDER BY LASTRUN
       WITH UR;
```

Slika 20: Izpis poizvedbe preko ukaznega centra



Preglednejše in s tem lažje spremljanje replikacije preko aplikacijskega vmesnika IBM obljublja v verziji 8.

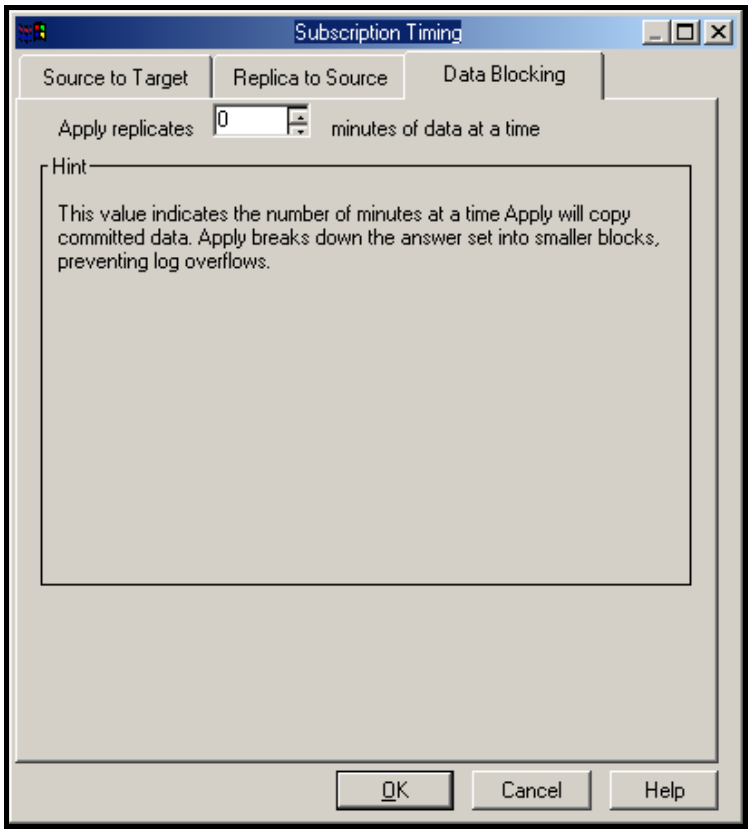
4.4.8 Veliki objekti

Propagacija velikih objektov ima nekatere posebnosti, na katere moramo biti pozorni, saj drugače le-ta ne deluje. Težavo predstavljajo tudi orodja za generiranje definicij nad posameznimi tabelami, saj se razlikujejo po verzijah in tako žal tudi po delovanju.

Posebnost se pojavi že pri kreiranju tabele, ki vsebuje velike objekte, saj potrebuje dva tabelarična prostora (običajna tabela fizično potrebuje en tabelarični prostor):

- enega za običajne podatke in
- drugega za zapise velikih objektov.

Slika 21: Določanje časa za blokiranje podatkov v procesu dodajanja



Povezava med fizičnima prostoroma je interna in jo vodi sistem DB2 preko identifikacijskega (ROWID) polja, ki je v bazi na centralnem računalniku viden, na

lokalni bazi pa ne. Ko torej izvedemo poizvedbo, se v ozadju izvede na dveh ločenih enotah in tako dobimo celoten podatek.

Ker veliki objekti ponavadi zasedejo veliko prostora (tudi do 2 MB in več), jih ne želimo beležiti, saj bi to pomenilo veliko obremenitev za sistem DB2. Zato ima tu proces zajemanja vgrajeno posebno logiko, ki zagotavlja, da se ob spremembi in s tem prenosom običajnih podatkov do cilja, vključi tudi polje z velikimi objekti. Tako v bistvu pošiljamo ogromne količine podatkov po mreži, ne glede na to, ali je prišlo do spremembe na polju velikega objekta ali ne. Ker smo to želeli preprečiti, smo temu priredili podatkovni model spletne aplikacije in izvajanje replikacije.

Osnovna naloga tabele, ki vsebuje velike objekte, je hranjenje starih, novih in spreminjajočih se XML dokumentov. Ker je ažuriranje pogosto, še posebej v času nastajanja novega dokumenta, smo dodali poseben stolpec, ki služi nadzoru replikacije. Tako se proces zajemanja proži preko pogleda, ki prikazuje le zapise, primerne za prenos. Tehnična rešitev je prikazana v prilogi na sliki 6 (glej v prilogi na str. 7). Tako smo preprečili pojavljanje pomanjkljivih dokumentov na portalu in zmanjšali promet po mreži.

Promet se po mreži lahko kontrolira tudi z izbiro možnosti, ki je predstavljena na sliki 21. V okence vneseno število minut predstavlja trajanje prenosa za določeno tabelo. Po pretečenem času prekine izvajanje in ga nadaljuje šele, ko obdela celoten cikel preostalih tabel. S tem razbremenimo operacijski sistem in si zagotovimo varnejše izvajanje.

Rešitev se je v praksi izkazala pri problemu, ki se je pojavljal pri popolni osvežitvi podatkov. Temu botruje tehnična rešitev procesa dodajanja, ki v fazi prenašanja podatkov velike objekte shranjuje kot datoteke na kontrolni strežnik in jih šele nato potiska naprej. Operacijski sistem pa pri tem zabrede v resne težave, ki mu jih povzroča veliko število datotek (preko 5000) na eni mapi. Pri prenosu zahteva vse več sistemskih virov, dokler ne pride do maksimalnih zmogljivosti. Malo kasneje pa ponavadi obstane. Potreben je ročni poseg, ki vključuje ponovni zagon strežnika. Nepredvideni težavi sem se izognil z vključitvijo možnosti blokiranja.

5. ANALIZA

Ko sem se podrobneje seznanil z izvedbo replikacije na spletnem portalu, me je čakala faza analize. Skušal sem ugotoviti naslednje:

- koliko postavljena struktura odgovarja sprejetim in priznanim varnostnim standardom,
- ali je uvedba replikacije pripomogla k učinkovitejšemu delovanju internetnega portala – ali z drugimi besedami: v kolikšni meri je replikacja upravičila svojo implementacijo in delovanje.

5.1 Varnostni vidik

Varnost je za banko ključnega pomena. V NLB se zavedamo, da so ljudje, objekti, informacije in sredstva predmet potencialnih in realnih groženj. Zato smo opredelili temeljne cilje in načela njihovega varovanja z namenom zagotavljanja zakonitosti in ugleda banke, ki jih upoštevamo na vseh ravneh in jih uveljavljamo vsi. Cilji varovanja so zagotavljanje varnih pogojev dela, varovanje integritete strank in delavcev v banki ter zaupnosti informacij povezanih z njimi, varovanja sredstev in premoženja, neprekinjenega poslovanja in omejevanje poslovne in materialne škode na najmanjšo možno mero.

Tako podatki niso samo shranjeni v SUBP, ampak jih različne aplikacije uporabljajo, kar pomeni, da se urejajo, obdelujejo, spreminjajo… in pogosto tudi selijo (Young, 2000). In ker se selijo, je na poti potrebno poskrbeti tudi za varovanje.

Vsak varnostni program mora zagotoviti varno upravljanje s ključi, dodeljevanje, zamenjavo, staranje. Kodiranje dodaja pomemben nivo varnosti pred vsiljivci, ki vdrejo skozi požarni zid ali preko pomanjkljivosti operacijskega sistema.

Hibridna rešitev kombinira tehnologije iz več področij (omrežij, aplikacij in baz) in zagotavlja najbolj optimalno cenovno rešitev za privatnost in varnost podatkov (Mattsson, 2003). Rešitev minimizira vpliv na aplikacijski nivo in kombinira moč in ločitev nalog iz zunanjega varnostnega sistema s podporo tesne bazne integracije.

Najboljša rešitev minimizira negativne vplive z nadzorovanjem samo tistih podatkov, ki so kritični z vidika varnosti, in ne celotne podatkovne baze. Privatnostno-varnostno pooblastilo in ostale poslovne zahteve definirajo, kateri podatki zahtevajo najvišji nivo

zaščite in revidiranja. Osredotočenje na samo občutljive podatke optimizira zmogljivosti in maksimizira uporabnost zaščitenih pregledovanj beleženja.

Varnostne rešitve morajo podpirati različne metode za prepoznavanje, vključno z geslom, varnostnimi disketami, pametnimi karticami in digitalnimi certifikati. Take rešitve bi morale dovoljevati tehnikom implementacijo in vzdrževanje varnostnega okolja iz ene nadzorne točke. Varnostna programska oprema bi morala biti nagradljiva z izvajalci lokalne kontrole na vsakem ciljnem strežniku (Mattsson, 2003).

Mnoge organizacije se zanašajo na mrežno kodiranje. Nekatere celo kodirajo podatke, ki so shranjeni v bazi, vendar je tak pristop drag in vpliva na (odzivnost) aplikacije in bazo. Lahko se odraža tudi kot resna omejitev v dostopu do podatkov skozi aplikacije in manj varnemu ter dražjemu upravljanju kodnih ključev.

Celovita varnostna politika mora varovati občutljive podatke v bazi v času hranjenja in procesiranja. Uveljavljati je potrebno privolitev v privatno pooblastilo in omogočati revidiranje na konsistenten in praktičen način skozi različne platforme.

Neprestani nadzor nad delovanjem strežnika zagotavlja pravočasno ugotavljanje ozkih grl, ki se nanašajo na zmogljivost in razpoložljivost strežnikov. Potrebno je zagotoviti alarme, ki javljajo zmanjšanje in pomanjkanje virov na strežniku, in avtomatsko pošiljanje obvestil skrbnikom, da se pravočasno zagotovi ustrezna nadgradnja (Gril, 2003 , str. 78).

Z replikacijo smo se do sedaj že podrobno seznanili. Ugotovili smo, da je sestavni del namestitve SUBP, ki sama prav tako izkorišča varnostne elemente operacijskega sistema. Zato sem varnostni vidik analiziral po parih – DB2 + pripadajoči operacijski sistem.

5.1.1 OS/390 in DB2

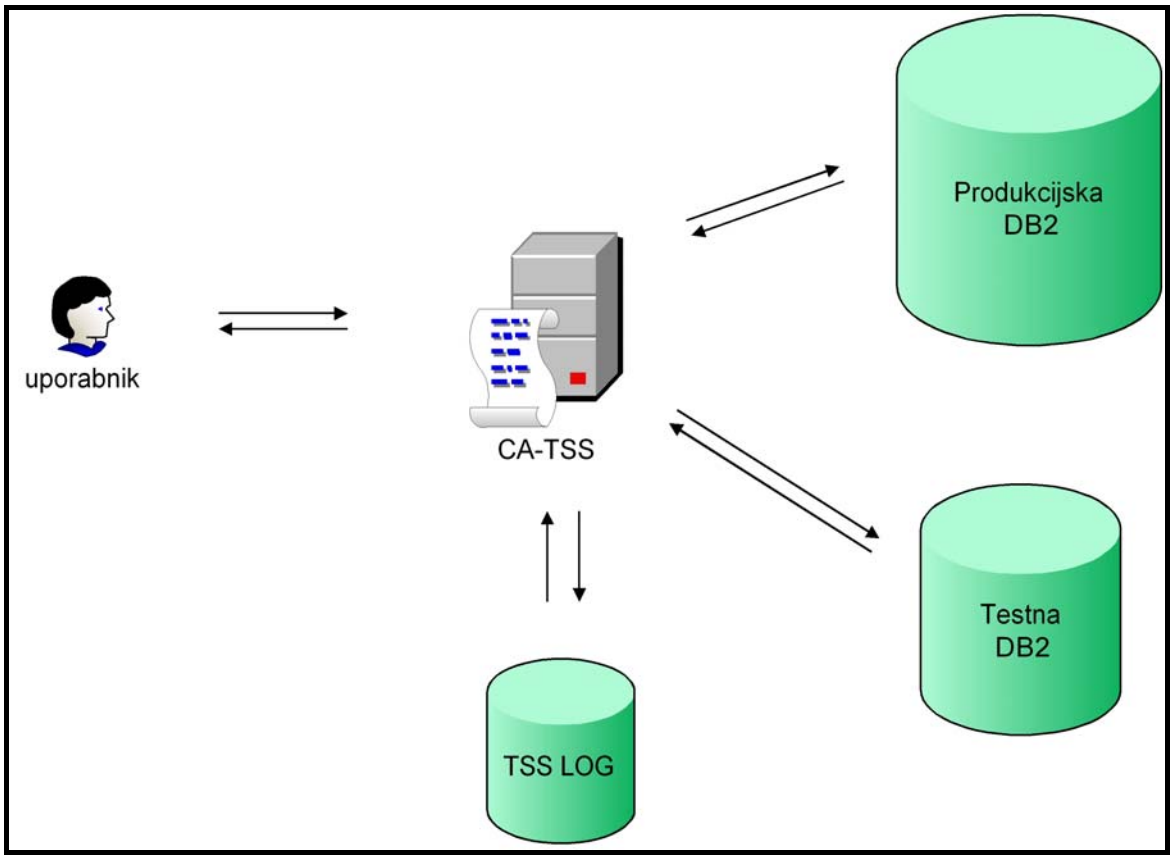
Do sistema OS/390 dostopamo po SNA komunikacijskem protokolu, ki je nefleksibilen in zapleten za vzdrževanja. Po funkcionalnostih DB2 na centralnem računalniku ponavadi nekoliko zaostaja za sorodnimi SUBP na lokalnih strežnikih. Za oba sistema pa velja, da sta stabilna in varna.

TSS je produkt, ki skrbi za varnost. Njegova naloga je nadzor pri prepoznavnosti uporabnikov (slika 22), ki se povezujejo na produkcijski in testni sistem in posledično tudi na DB2. Vse prijave tudi beleži in tako omogoča naknadno kontrolo avtoriziranih in neavtoriziranih povezav. Programski paket je zelo stabilen. Do sedaj še nismo imeli težav, ki bi jih le-ta lahko povzročil.

Neavtorizirane poskuse dostopa, spreminjanja, brisanja podatkov ali datotek tudi beleži v posebne zaščitene datoteke, do katere ima dostop samo skrbnik TSS, in ki se redno tudi pregledujejo.

Na OS/390 obstaja tudi več naprednih orodij za popolnejši nadzor nad delovanjem DB2. Poleg tega vsako novost vedno najprej kontroliramo na testnem sistemu. Kljub temu so se težave na produkcijski DB2 že pojavile, a smo jih uspeli rešiti brez posledic izgube podatkov ali zloma sistema.

Slika 22: Top Security System (TSS)



Podatki na diskih so v RAID (Redundant Array of Independent Disks) formatu 1, ki označuje zrcaljenje (isti podatki se hranijo na dveh neodvisnih diskih). Redno se izdelujejo tudi varnostne kopije na trakove.

5.1.2 Unix in DB2

Unix-ov operacijski sistem in baza sta v vsem ciklu propagacije še najbolj izpostavljena možnostim vdora. Zato se na sistemu redno izvajajo namestitve

varnostnih popravkov. Tedensko se sestaja tudi skupina, kjer sodelujejo vzdrževalci iz našega podjetja in tujih pogodbenih podjetij. Namen je ravno testiranje, vzdrževanje, implementacija obstoječih in novih varnostih mehanizmov.

Potrebno pa je vzdrževanje SUBP (brisanje odvečnih beleženj, reorganizacija podatkov…) in prenos podatkov z replikacijo. Požarni zid je nastavljen tako, da navzven do portala lahko dostopamo, obratno pa je to nemogoče. Kontrola replikacije se nahaja v notranjem omrežju in samo poriva ter vleče podatke preko požarnega zidu.

Do SUBP skrbniki dostopamo preko SSL (Secure Socket Layer) varnostnega protokola. Iznašel ga je Netscape leta 1994 (Stan, 2000). Omogoča povezavo med odjemalcem in strežnikom na varni šifrirani lini, ki je tako privatna (Jerman Blažič, 2001, str. 172). S svojim načinom delovanja zagotavlja zaupnost, ki je pomembna za obe strani (Jerman Blažič, 2001, str. 173). Deluje na TCP/IP povezavi. Po nekajminutni neaktivnosti se povezava samodejno prekine.

Tudi tu so podatki na diskih v RAID 1 formatu. Na trakove kopiramo le del podatkov (varnostno kopijo in podatke nekaterih tabel), saj se vsi vitalni že vzdržujejo in arhivirajo na centralnem računalniku.

5.1.3 Windows 2000 in DB2

Operacijski sistem je postavljen na strežniku, ki delujejo zelo zanesljivo. Dele občutljive strojne opreme ima namreč podvojene. Šibka stran je disk, ki je eden in ni podvojen. Če pride do njegove okvare, bi ostali brez podatkov. Ker sem se zavedal pomanjkljivosti, sem po vzpostavitvi replikacije arhiviral podatkovno strukturo kontrolnih tabel z vsemi potrebnimi podatki (definicijami replikacij) in jih spravil na varno lokacijo. Akcijo ponovim, ob vsaki dodani ali spremenjeni definicij.

Ob ponovni postavitvi sistema bi lahko v sorazmerno kratkem času vzpostavil staro stanje. Upoštevati bi moral dve stvari:

- izgubil bi podatke iz tabele za beleženje (ne)uspešnosti replikacij za posamezne tabele, kar ni tako problematično,
- ob zagonu bi replikacija izvedla popolno osvežitev podatkov na spletni portal, kar bi terjalo mojo pozornost pri nadzoru tako obsežne količine prometa po omrežju.

Veliko pozornosti si zasluži tudi operacijski sistem, saj se kritični varnostni popravki redno nameščajo. Za tovrstni operacijski sistem so dokaj pogosti in ob vsaki takšni namestitvi je potreben ponovni zagon sistema, ki je zelo občutljiv del, saj prekine

izvajanje replikacije. Zato redno dobivam informacije o novih namestitvah in ponovnih startih strežnika, da lahko izvajam učinkovit nadzor.

Linja je zavarovana in kodirana na celi poti (od protokolnega vmesnika do portala) prenosa podatkov, torej tudi v notranjem omrežju.

5.1.4 Revizija

Pomembna zahteva za vse varnostne in privatne (zasebne) implementacije je revidiranje SUBP. Beleženje aktivnosti izvajajo uporabniki. Kritični element so neavtorizirani poskusi dostopa.

Beleženja se morajo izvajati, ne glede na uporabo občutljivih podatkov, skupaj s prebranimi ali spremenjenimi zapisi določenega uporabnika. Upravljavci lahko uporabijo te zapise za sledenje trendov, analizo potencialnih nevarnosti, podporo varnostnemu načrtovanju in ocenijo učinek protiukrepov. Beleženje bi se idealno moralo koncentrirati na najbolj uporabne informacije za varnostne upravljavce, ki so "aktivnost okoli zaščitenih podatkov".

Revidiranje ni opravek na vse ali nič – mora biti selektiven. Z osredotočenjem na samo občutljive podatke selektivno revidiranje privarčuje čas in zmanjša skrbi. Omejevanje zbiranja revidiranih beleženj v tej smeri pomeni osvetlitev in vpogled nazaj v kritične varnostne dogodke.

NLB je definirala svojo revizijsko strategijo občutljivih podatkov, osnovano na poznavanju aplikacij in podatkovnih baz. Pomanjkljivo revidiranje zmanjšuje pravo odgovornost in ovira proces reševanja notranjih in zunanjih vrzeli. Učinkovite revizijske sledi so ključne za razumevanje, katere poteze moramo potegniti za zaščito občutljivih podatkov. Bistveno je direktno beleženje dogodkov v povezavi z občutljivimi podatki v podatkovni bazi.

5.1.5 Ostali elementi

5.1.5.1 Elektronska pošta

Programski paket za delovanje spletnega portala (od kreiranja dokumentov do prikazovanja na spletnih straneh) smo kupili od zunanjega proizvajalca. Že v fazi testiranja smo ugotovili posamezne pomanjkljivosti in nepravilnosti. Posledica so bile "živahne" komunikacije preko elektronske pošte. Ker smo se zavedali pomanjkljivosti

varovanja pri tovrstni komunikaciji, smo že na uvodnih sestankih določili za kodiranje s PGP Personal Privacy (An introduction to cryptography, 2000).

5.1.5.2 Vzdrževalne pogodbe

Za zmanjšanje grožnje odpovedi opreme in odpravljanje manjših težav pri okvarah so sklenjene vzdrževalne pogodbe, ki hitro zagotovijo nadomestne aparature. Vsi računalniki so v zaščitenih, klimatiziranih prostorih in tako dostopni samo pooblaščenim osebam.

5.1.5.3 Človeški faktor

Za prekinitve delovanja pri računalnikih so večkrat vzrok človeške napake. Posebej problematične so zato, ker jih ni možno vselej vnaprej predvideti in se posledično nanje ne moremo pripraviti. Včasih se zmanjša število napak že s tem, da imajo zaposleni dovolj časa in miru za opravljanje svojega dela in da niso podvrženi zunanjim vplivom, ki bi zmanjševali njihovo zbranost. Tveganja bi zmanjšale tudi dodatne kontrole, ki bi poskrbele za redkejše pojavljanje napak (na primer bolj strog nadzor). Potrebno bi bilo tudi pregledati, dopolniti ali na novo napisati navodila za delo nadomestnih skrbnikov v primeru odsotnosti glavnih skrbnikov sistemov (Gril, 2003 , str. 77).

5.2 Analiza pričakovanj in rezultatov

Naša pričakovanja najlepše ponazori naslednji citat: "Kar pričakujemo, je praktična rešitev, ki zagotavlja konsistentnost, zagotavlja jasne, natančne kriterije, razumno toleranco napak, se izogiba ozkim grlom in ima dobre zmogljivosti" (Kemme, Alonso, 2000, str. 1).

Uporabniki (kar sem pogosto tudi sam) bi si vsekakor želeli hitro in pravilno delovanje, brez povezav v slepe ulice in napačnih ter zavajajočih podatkov. V zvezi s hitrostjo je bilo veliko narejenega na optimizaciji spletne aplikacija že v času testiranja. Seveda pa spremljam razvoj dogodkov in odzive uporabnikov. Poslovna stran namreč nadaljuje z zahtevami po novih načinih uporabe podatkov in mnoge pristope še vedno razvijajo. Zato je moja naloga, da jim nudim potrebno podporo in jih seznanjam s potencialnimi omejitvami, ki se jih večinoma da odpraviti (na primer z razširitvijo sistema).

Na začetku je implementacija aplikacije in njenih nadgradenj predstavljala velik vir napak predvsem zato, ker so bile premalo testirane. Z dodatnimi ukrepi smo si izborili več časa za testiranje in odpravljanje napak. Popolnejša je tudi dokumentacija, ki sedaj zadostuje vsem kriterijem standarda ISO 9001/TickIT, ki smo ga prejeli leta 2001.

Uporabniki čutijo tudi slabše delovanje procesov zajemanja in dodajanja, ki niso najbolj optimalni. Dokler je v tabelah malo zapisov, ni težav, kasneje pa lahko postaneta precej potratna. Problem je tembolj očiten, ko se procesne zmogljivosti centralnega računalnika približujejo maksimalnim zmogljivostim.

S podrobnejšo analizo in pomočjo nekaterih orodij (DB2PM) sem uspel pridobiti poizvedbe, ki imajo predolge odzivne čase in uporabnikom povzročajo motnje v njihovem delovnem procesu. Sledilo je ponovno načrtovanje indeksov, ki je precej izboljšalo stanje, a ne povsem. Zanimivo je, da sem našel poizvedbe, ki jih pogosto zasledim – celo v IBM-ovi strokovni literaturi - kot slabe. Torej lahko le upam, da bo nova verzija popolnejša.

Spremembe na indeksih pogosto omogočajo aplikacijam boljše odzive in posledično večje zmogljivosti. Dobra stran so opazni prihranki na procesorju centralnega računalnika, zmanjšanje I/O (Input/Output) in manj razporejanja. Te prednosti se nanašajo na doma razvite in kupljene aplikacije in ne vplivajo na integriteto, saj primarnega ključa ne spreminjamo.

Dodajanje indeksov lahko deluje kot zdravilo za vse bolezni. Vendar moramo spremembe indeksov uveljavljati šele po razmisleku, kako bo to vplivalo na celotno delovanje aplikacije. Pogosto slabi analizi sledi prekomerno število indeksov, kar povzroči povečanje aktivnosti ob spremembah podatkov na tabelah (Hubel, 2001).

Redno izvajamo procedure, ki nam generirajo statistična poročila. Iz njih lahko odkrijejo presenetljiva dejstva o podsistemu, za katerega smo menili, da ga dobro poznamo (Catteral, 2003). Na podlagi pridobljenih rezultatov si ustvarimo čistejšo sliko porabljenih virov in obiska portala.

Zelo težko pa je izmerljivo zadovoljstvo uporabnikov. Anket se namreč vsi zelo radi izogibamo, med tem ko nikoli ne varčujemo s negativnimi kritikami. Nekoliko lažje je s skrbniki portala v NLB. Tu je informacija direktna in hitra. Pritožbe letijo predvsem na občasno počasno delovanje, ki pa je večinoma posledica procesnih špic na centralnem računalniku.

Težave pa ne izvirajo samo iz sistemskega, temveč tudi organizacijskega okolja. Del podatkov se namreč (še) ne nahaja v elektronski obliki in te nam povzročajo največ preglavic. Jasne rešitve pa žal še ni na vidiku.

5.3 Analiza stroškov in koristi

Analiza stroškov in koristi se uporablja za pomoč pri določanju prioritet in za izboljšanje ekonomske učinkovitosti. Kriteriji se nanašajo na kvantitativno izražen dobiček, tveganja pri vrnitvi investicij in zagotavljanja identifikacij za predlagane investicije (IT Cost-Benefit Analysis, 2001).

Slika 23: Kategorije koristi

Vir: Business Inteligence Roadmap, 2003.

Koristnosti so ponavadi težje ocenljive kot stroški. Zelo učinkovita metoda za zadovoljevanje izdatkov je direktna povezava na poslovni problem. Pri identificiranju koristnosti moramo biti zelo natančni, čeprav je težko podati točne izračune. Tako lahko dobimo ustrezno podporo vodstva in podporo za razvoj projekta (Moss, Atre, 2003).

Slika 23 prikazuje kategorije koristi, ki vsebujejo (Moss, Atre, 2003):

- povečanje dohodkov (identifikacija novih trgov in niš, nakazovanje uspešnejše prodaje, hitrejša identifikacija priložnosti, hitrejša pojavitev na trgu),

- povečanje dobička (bolj ciljno usmerjena elektronska sporočila, zgodnejše odkrivanje upada trga, identifikacija nezmogljivih proizvodnih lin ali produktov, identifikacija notranje neučinkovitosti, učinkovitejše upravljanje blaga),
- povečanje zadovoljstva stranke (razumevanje strankinih prednosti, ujemanje stranke in produkta, naknadna naročila strank, hitrejše razreševanje strankinih pritožb),
- povečanje prihrankov (zmanjšanje odpadnih ali časovno pretečenih izdelkov, zmanjšanje zahtev po rednih poročilih),
- povečanje tržnega deleža (povečanje števila strank na račun konkurence, večje število strank v primerjavi s preteklimi leti ali konkurenco).

Vrednost spletnega portala je prav tako cenovno težko ovrednotiti. Preko interneta banka širi svojo ponudbo in storitve. Ker ne izvajamo direktne prodaje, ne moremo operirati s konkretnimi zneski. Glede na kategorije koristi, ki sem jih opisal v prejšnjem odstavku, pa lahko zaključim naslednje:

- velika količina izračunanih posojil in depozitov razbremenjuje komercialiste, ki se lahko posvečajo pomembnejšim opravilom,
- predstavitev posebnih ponudb, ki jih banka lahko pravočasno najavi tudi preko spletnega portala, dviguje prodajo storitev,
- enostavna dostopnost do datotek (letna poročila, javni ključ…), ki jih komitenti potrebujejo ali želijo,
- velika obiskanost na straneh, ki promovirajo elektronsko bančništvo, kaže na povečano zanimanje strank za sodobne tržne poti,
- splošno zadovoljstvo strank zaradi preglednosti ključnih podatkov, kot so tečajne liste in podobno (veliko pozitivno naravnanih mnenj preko elektronske pošte).

Ob prejetju priznane nagrade za najboljšo spletno stran je bilo veliko napisanega v različnih tiskanih izdajah znanih časopisnih hiš, kar je bilo že samo po sebi najboljša reklama za obisk. Dober portal je NLB potrebovala, saj se tudi na tak način kaže odnos do strank.

5.4 PSPN (SWOT) analiza replikacije

Kratica PSPN (SWOT) je sestavljena iz naslednjih besed: prednosti (Strenghts), slabosti (Weaknesses), priložnosti (Opportunities), nevarnosti - pretnje (Threats). Pri tem se prednosti in slabosti nanašajo na raziskovani predmet, medtem ko se priložnosti in nevarnosti nanašajo na okolje.

V našem primeru je težko izvesti cenovno ocenjevanje, saj ne operiramo s konkretnimi vrednostmi, ker je le-te težko ovrednotiti. Programski paket za izvajanje

replikacije v DB2 okolju ni samostojen produkt, torej prosto ne nastopa na trgu. Postavitev procesov in implementacija je bila plod našega dela in izkušenj, pri čemer niso sodelovali zunanji strokovnjaki. Po drugi strani pa bi bila primerjava koristnosti zaradi specifičnega primera in okolja s podobnimi programskimi paketi nesmiselna.

Zato sem se odločil za subjektivno metodo. Pri tem načinu se vsaka podstruktura razstavi na posamezne sestavne dele, ki se jih oceni. Potem se te ocene povežejo v skupno oceno posamezne podstrukture (Pučko, 1996, str. 134). Osredotočil sem se na cilj, ki ga replikacija sedaj zagotavlja in ga analiziral. Rezultati so predstavljeni v dveh ločenih tabelah.

Tabela 1: Prikaz prednosti in slabosti

Prednosti	**Slabosti**
• varen prenos podatkov • kontinuiran prenos podatkov • delovanje portala tudi v primeru odpovedi replikacije • nizki stroški implementacije in delovanja	• popolna odpoved replikacije bi pomenila večurno zamudo pri osvežitvi podatkov • slab nadzor nad izvajanjem replikacije

V procesu kontinuiranega prenašanja podatkov nam ni potrebno skrbeti, kdaj in kako se bo sprožilo podvojevanje podatkov na ciljnem strežniku. Kontrola se namreč nahaja v notranjem omrežju NLB, kar omogoča varen prenos podatkov, in je ni moč nadzorovati od zunaj. Nedelovanje replikacije ne vpliva na delovanje portalskega strežnika, ki lahko nemoteno naprej opravlja svoje poslanstvo.

Stroški implementacije in delovanja so nizki, saj smo samo izkoristili dane možnosti, ki porabijo manjšo količino sistemskih zmogljivosti. Majhna poraba zmogljivosti pa nam zagotavlja delovanje tudi ob večjih obremenitvah sistemov.

Popolna odpoved replikacije pomeni, da je potrebno ponovno vzpostaviti procesa zajemanja in dodajanja ter zagotoviti podatke, ki določajo replicirane tabele v kontrolnih tabelah na obeh straneh. Po uspešnem zagonu procesov je edina možnost popolna osvežitev vseh podatkov za vsako posamezno tabelo, kar lahko vzame precej časa. Delovanje spletnega portala je v takšnem primeru moteno, saj pride do prikazovanja pomanjkljivih in zavajajočih podatkov.

Nadzor nad izvajanjem replikacije je slab, saj nam ne omogoča alarmiranja ob napakah ali odpovedih procesov zajemanja in dodajanja. Posledice so slabi odzivni

časi ob pojavu in odpravljanju težav, kar prizadene skrbnike in uporabnike spletnega portala.

Tabela 2: Prikaz priložnosti in nevarnosti

Priložnosti	Nevarnosti - Pretnje
• nova popolnejša orodja za nadzor replikacije • hitrejše izvajanje prenosa podatkov	• zahteve po kompleksnejših metodah replikacije • ukinitev replikacije • novejši proces replikacije, ki ne bi podpiral obstoječih metod

Na trgu se pojavljajo nova in posodobljena orodja za nadzor replikacije, ki vsebujejo ali so skladna z obstoječimi programi obveščanja po elektronski pošti ali preko kratkih sporočil na mobilni telefon. Zmogljivejši strežniki in komunikacijske naprave bodo še pospešile prenos podatkov po omrežju, kar bo skrajšalo čas, potreben za prehod podatkov na ciljni sistem.

Nevarno je implementirati novosti, če jih popolnoma ne obvladujemo, saj jih je težko ali pa (vsaj na začetku) celo nemogoče izvajati. Kompleksnejše metode izvajanja replikacije bi zahtevala nova znanja in angažiranost skrbnikov podatkovnih baz, kar bi pomenilo dodatne stroške v celotnem obvladovanju procesov spletnega portala.

Ukinitev replikacije ob nespremenjeni varnostni politiki bi pomenilo selitev podatkov preko drugih orodij. Sprememba načina prenosa bi vzela veliko časa. Ponovno bi se srečali z začetniškimi napakami, ki so sedaj večinoma že odpravljene in spremljajo vsak nov produkt. Podobne težave bi se pojavile tudi pri novejšem procesu replikacije, ki ne bi podpiral obstoječih metod.

6. SKLEP

Predstavljena naloga dokazuje, da obvladovanje obsežnih in kompleksnih procesov resnično ni mačji kašelj. V drugem poglavju sem v grobem najprej predstavil širok spekter uporabljenih tehnologij v NLB. Raznolikost nam da tudi različne profile ljudi, ki upravljajo s sistemi. Zato je dobro sodelovanje zelo pomembno, saj pomembno vpliva na dobro koordiniranost izvajanja akcij, predvsem pa je pomembno, da so te kvalitetne. Dobro upravljanje zmore samo dobro uglašen tim.

V četrtem poglavju sem se osredotočil na replikacijo. Opisal sem možnosti, ki jih ponujajo in obljubljajo proizvajalci. V praksi mnogokrat naletimo na različne težave pri uporabi produkta in tudi tokrat ni bilo nič drugače. Pomagati sem si moral z obvozi in prilagajanjem, kar sem večinoma tudi opisal. Morda bomo naleteli tudi na kontradiktorne izjave, ki so plod razhajanja med tistim, kar bi programski paket moral nuditi, in tistim, kar dejansko nudi.

Na koncu naj povzamem nekaj najpomembnejših sklepov, ki sem jih predstavil po smiselnih sklopih:

- Replikacija je izpolnila pričakovanja uporabnikov in zadostila strogim varnostnim pravilnikom banke. Sedaj, ko je led prebit, so odprte še druge možnosti uporabe. Širše gledano smo v NLB hitro in z minimalnimi stroški zagotovili prenos podatkov do oddaljene SUBP, pri čemer smo samo izkoristili dane možnosti vgrajenih orodij in procesov. Verjetno se tudi druga podjetja srečujejo s podobnimi težavami. Z dobrim poznavanjem poslovnih procesov na eni strani ter potrebami in željami na drugi, lahko učinkovito izkoristimo množico integriranih orodij , med katerimi se nahaja tudi replikacija.
  Iz opisanega v nalogi lahko sklepamo, da bomo morali v NLB vložiti še veliko truda za izboljšanje razmer. Nekatere naloge smo že uspeli realizirati, nekatere pa nas še čakajo.

- Pričenjamo tudi z izvajanjem testov na lokalnem SUBP, verzija 8, ki ponuja boljšo kontrolo nad izvajanjem replikacije. Alarmi bi potemtakem do nas prišli hitreje in temu primerna bi bila tudi odzivnost. Uredili bi pošiljanje alarmov na elektronsko pošto ali po kratkih sporočilih na mobilni telefon, saj se težave lahko pojavijo v kateremkoli delu dneva.

- Spremenjena bazna arhitektura (podatkovni model replikacijskih tabel) v novi verziji je učinkovito zmanjšala velike potrebe proces dodajanja, ki so postajali vsebolj kritični.

- Kontrolni strežnik v kratkem času čaka vidna posodobitev; migracija na Windows 2003 strežniški operacijski sistem (nudi večjo kontrolo nad dodeljevanjem sistemskih in procesnih virov), močnejši procesor ter hitrejše diske v RAID 1 formatu.
  Spremljali smo tudi delovanje Unix-a in ugotovili, da obstaja del neizkoriščenih virov. Dodelili smo jih lokalni bazi in tako pospešili delovanje spletnega portala.

- Poskrbeli smo, da se vsi strežniki in centralni računalnik napajajo preko UPS (Uninterruptible Power Supply) sistema, ki zagotavlja nemoten dotok električne energije in s tem omogoča neprekinjeno delovanje.

- Trenutno nimamo varnostnega sistema, ki bi ga bilo mogoče nadzorovati iz ene točke. Sam pripisujem velik pomen že temu, da se vodstvo zaveda te pomanjkljivosti. Gostili smo namreč že nekaj strokovnjakov tujih podjetij, ki ponujajo obsežne napredne programske pakete, ki to zmorejo. NLB pa sicer spremlja najnovejše trende na področju varnosti in jim sledi v najboljši možni meri.

- Zaradi novih funkcionalnosti in posameznih sprememb sledimo novim verzijam vse programske opreme, saj si le tako olajšamo delo in povečamo učinkovitost. Poleg naštetega nam je vedno na voljo tudi ustrezna podpora proizvajalca, ki je sicer za posamezno verzijo časovno omejena.

- Trend za področje SUBP DB2 je, da bi se spreminjanje večine parametrov izvajalo dinamično, kar bi pomenilo hitrejši odziv na spremembe in temu primerno boljše delovanje oziroma optimizacijo. V trenutni situaciji je v večini primerov znak spremembe viden šele po ponovnem startu baze, kar pa je redek in skrbno načrtovan proces.

- Smotrno bi bilo razmisliti in praktično stestirati prednosti, ki jih obljubljajo XML extenderji. V različnih strokovnih revijah sem že uspel prebrati nekaj spodbudnih novic. A po izkušnjah sodeč, se rezultati od primera do primera lahko zelo razlikujejo, zato pretiran optimizem tu ni dobrodošel.

- Smiselna bi bila tudi analiza mrežnih virov. V množici strojne opreme se namreč vzdržuje standardna povezava med sistemi. Ker gre za zahtevno področje servisiranja informacij vsem zunanjim uporabnikom, bi bila morda že višja prioriteta znak ponovnega dizajna povezav in posledično povečanje učinkovitosti in hitrosti prenosa podatkov do ciljne baze in nazaj.

Vidim še mnogo prehodov in odprtih vrat. Imamo torej še veliko možnosti, ki nam lahko vedno znova olajšajo in pospešijo procese. Moja želja in naloga je izpopolniti delovanje spletnega portala, saj bomo le tako zadovoljili uporabnike na njihovi poti do informacij in posledično ohranjali ugled NLB.

7. LITERATURA

1. Beyda J. William: Basic Data Communications. Englewood Cliffs: Prentice Hall, 1989. 268 str.

2. Beyer Thomas: E-business and DB2 - A Perfect Fit? The IDUG Solutions Journal. [URL: http://www.idug.org/idug/member/journal/nov99/feature5.cfm], november, 1999.

3. Catterall Robert: Just the Stats, Ma'am. DB2magazine. [URL: http://www.db2mag.com/db_area/archives/2003/q2/db2dba.shtml], 2. četrtletje, 2003.

4. Fosdick Howard: DB2 For NT: What's New And What's Next. The IDUG Solutions Journal. [URL: http://www.idug.org/idug/member/journal/mar98/db24nt.html], marec, 1998.

5. Gril Matej: Varnost In tehnološka zaščita informacijskega sistema v banki. Magistrsko delo. Ljubljana: Ekonomska fakulteta, 2003. 90 str.

6. Hayes Scott: Optimizing Buffer Pools in DB2 UDB for UNIX, Windows, and OS/2. The IDUG Solutions Journal. [URL: http://www.idug.org/idug/member/journal/mar00/optimize.cfm], marec, 2000.

7. Hoover Chuck: Broader Horizons With DB2 Version5. Farmington Hills: Compuware Corporation, 1997. 142 str.

8. Hordila Mike: Automated Replication Management. [URL: http://www.dbazine.com/hordila5.html], 2002.

9. Hoskins Jim, Frank Bob: Exploring IBM @server zSeries and S/390 Servers. Sedma izdaja. Gulf Breeze: Maximum press, 2002. 416 str.

10. Hotek Michael: Database Replication. Windows & .NET Magazine. [URL: http://www.insession.com/goldengate/DataReplicationBuyersGuide.pdf], januar, 2002.

11. Hubel Martin: A Simple Approach to Index Redesign. The IDUG Solutions Journal. [URL: http://www.idug.org/idug/member/journal/Nov01/articl06.cfm], november, 2001.

12. Jerman Blažič Borka et al: Advanced Security Technologies in Networking. Amsterdam: IOS Press, 2001. 257 str.

13. Johnston Britt: Replication is a Two-way street. Computerworld. [URL: http://www.computerworld.com/mobiletopics/mobile/story/0,10801,79091,00.html?SKC=mobile-79091], marec, 2003.

14. Kemme Bettina, Alonso Gustavo: A New Approach to Developing and Implementing Eager Database Replication Protocols. ACM Transactions on Database Systems, september, 2000. Št. 3, str. 333 – 379.

15. Komer Kathy: In the Game. DB2magazine. [URL: http://www.db2mag.com/db_area/archives/2001/q4/idug.shtml], 4. četrtletje, 2001.

16. Kovačič Andrej: Informatizacija poslovanja. Prva izdaja. Ljubljana: Ekonomska fakulteta, 1998. 214 str.

17. Lawson Susan, Hubel Martin, Mordini Gary: DB2 Universal Database Version 7 - The Database Solution for E-business. The IDUG Solutions Journal. [URL: http://www.idug.org/idug/member/journal/summer00/articl06.cfm], poletje, 2000.

18. Lindholm Elizabeth: Is DB2 Right for You?. Windows @ .Net Magazine. [URL: http://www.winnetmag.com/Articles/Index.cfm?ArticleID=2864], december, 1996.

19. Logan Mike: Made for Each Other: DB2 Universal Database V7.1 and Windows 2000. DB2magazine. [URL: www.db2mag.com/db_area/archives/2000/q3/logan.shtml], 3. četrtletje, 2000.

20. Mattsson T. Ulf: Protecting DB2 Data. DB2magazine. [URL: http://www.db2mag.com/db_area/archives/2003/q1/mattsson.shtml], 1. četrtletje, 2003.

21. Marshall Morwenna: A New Database Animal. DB2magazine. [URL: http://www.db2mag.com/db_area/archives/1998/q2/98smMrsh.shtml], 2. četrtletje, 1998.

22. Mullins Craig S.: What Is a DBA? DB2magazine. [URL: http://www.dbazine.com/dba_1.html], junij, 2002a.

23. Mullins Craig S.: What Is a DBA? — Part Two. DB2magazine. [URL: http://www.dbazine.com/dba_2.html], junij, 2002b.

24. Moss T. Larissa, Atre Shaku: Business Inteligence Roadmap. Boston: Addison Wesley, 2003. 576 str.

25. Nanda Arup: Altering the Master Table in a Snapshot Replication Environment without Recreating the Snapshot. [URL: http://www.dbazine.com/nanda2.html], 2002.

26. Otey Michael: Database Replication Components. SQL Server Magazine. [URL: http://www.sqlmag.com/Articles/Print.cfm?ArticleID=5677], avgust, 1999.

27. Perna Janet: Creating an e-business Universe: IBM DB2 in the New Millennium. The IDUG Solutions Journal. [URL: http://www.idug.org/idug/member/journal/nov99/feature1.cfm], november, 1999.

28. Pučko Danijel: Strateško upravljanje. Ljubljana : Ekonomska fakulteta, 1996. 394 str.

29. Ray Jerry: Database Replication - Developing applications that support remote offices and data warehousing - Part 1. Clarion Magazine. [URL: http://www.clarionmag.com/col/98-10-replication.html], oktober, 1998.

30. Rennhackkamp Martin: Mobile database replication. DBMS. [URL: http://www.dbmsmag.com/9710d17.html], oktober, 1997.

31. Schnaidt Patricia: Enterprise – Wide Networking. Carmel: Sams Publishing, 1992. 493 str.

32. Sherman Nicholette: Taking Database Performance to New Levels. The IDUG Solutions Journal. [URL: http://www.idug.org/idug/member/journal/Nov01/articl09.cfm], november, 2001.

33. Simčič Slobodan: Spoznajmo lokalne računalniške mreže. Ljubljana: ŠKD FORUM, 1988. 196 str.

34. Simpson David: Is Replication Appropriate? [URL: http://www.dbazine.com/simpson1.html], 2002.

35. Smith Hugh, Young Casey: Pooling Anybody? No, Not Car Pooling (Connection and Transaction Pooling in DB2 Connect). The IDUG Solutions Journal. [URL: http://www.idug.org/idug/member/journal/Sept01/articl05.cfm], september, 2001.

36. Stodder David: Bound to Succeed. DB2magazine. [URL: http://www.db2mag.com /db_area/archives/2003/q1/stodder.shtml], 1. četrtletje, 2003.

37. Thompson Charles: Database Replication. DB2magazine. [URL: http:// www.dbmsmag.com/9705d15.html], maj, 1997.

38. Yevich Richard et al.: DB2 Answers! Certified Tech Support. Berkeley: McGraw-Hill/Osborne Media, maj, 1999. 444 str.

39. Yevich Richard: Simply Irresistible: DB2 to Application Vendors. DB2magazine. [URL: http://www.db2mag.com/db_area/archives/2001/q4/yevich.shtml], 4. četrtletje, 2001.

40. Young Casey: Irrigating Your Information Fields – a Look at Middleware. The IDUG Solutions Journal. [URL: http://www.idug.org/idug/member/journal /summer00/articl07.cfm], poletje, 2000.

8. VIRI

1. An introduction to cryptography. [URL: http://download.nai.com/products /media/pgp/support/pgp/pgp71_win32/introtocrypto.pdf], Network Associates, 2002.

2. Bobek Samo, Strnad Simona: Programska oprema, Tema 12. [URL: http://epf-oi.uni-mb.si/clani/bobek/Informatika/tema12.pdf], 2004. 72 str.

3. Building the Operational Data Store on DB2 UDB Using IBM Data Replication, WebSphere MQ Family, and DB2 Warehouse Manager. USA: IBM, december, 2001. 334 str.

4. Darren Johnson: PostgreSQL Database Replication Options. [URL: http://conferences.oreillynet.com/presentations/os2002/johnson_darren.pdf], 2002. 29 str.

5. Database connections. [URL: http://manuals.sybase.com/onlinebooks/group-rs/rsg1250e/rshetrep/@Generic__BookTextView/8658], Sybase, 2004.

6. Database Replication. [URL: http://www.csee.umbc.edu/help/oracle8/server.815/ a67781/c31repli.htm#12636], Oracle, 1999.

7. Daudjee Khuzaima, Salem Kenneth: Lazy Database Replication with Freshness Guarantees. [URL: http://www.cs.uwaterloo.ca/cs-archive/CS-2002/31 /techrep.pdf], 2002.

8. DB2 UDB Server for OS/390 and z/OS Version 7: Presentation Guide Version 7. USA: IBM, Prva izdaja, 2001. 568 str.

9. DB2 Universal Database for OS/390 and z/OS: Administration Guide Version 7. USA: IBM, Prva izdaja, 2001. 1070 str.

10. DB2 Universal Database for OS/390 and z/OS: Installation Guide Version 7. USA: IBM, Prva izdaja, 2001. 582 str.

11. Dove Debra: Database Replication in Microsoft Jet 4.0. [URL: http://www.homelancs.com/Download_Files/dbrepjet.htm], Microsoft Corporation, januar, 1999.

12. DMZ. [http://www.webopedia.com/TERM/D/DMZ.html], 2002.

13. Garrett J. James: Elementi uporabniške izkušnje. [URL: http:// jjg.net/ia/elements_si.pdf], 2000.

14. Hamel Beth: New Features in DB2 Replication Version 8. [URL: http://www-3.ibm.com/software/data/dpropr/pres13/v8featuresprz.pdf], 2004.

15. http://www.ibm.com, 2004.

16. IAB. [URL: http://citeseer.nj.nec.com/cerf90internet.html], 2004.

17. IBM, Database2 for MVS/ESA – Administration Guide. USA: IBM Corporation, Prva izdaja, 1995. str.

18. IBM DB2 Universal Database: Replication Guide and Reference Version 7. USA: IBM, 2000. 430 str.

19. I&K Resarch - DRAGON: Database Replication based on Group Communication. [URL: http://www.inf.ethz.ch/department/IS/iks/research/dragon.html], 1998.

20. Indikatorji Interneta. [URL: http://www.ris.org/indikatorji/projekcije.html], RIS, april 2002.

21. IT Cost-Benefit Analysis. [URL: http://wwwoirm.nih.gov/itmra/cost-benefit.html], 2001.

22. Muc Simona: Spletni portal NLB – Med strategijo in izvedbo, 2003

23. Predstavitev banke na Internetu. [URL: http://www.nlb.si], 2002.

24. RAID. [URL: http://www.webopedia.com/TERM/R/RAID.html], 2004.

25. Replication. [URL: http://www.csee.umbc.edu/help/oracle8/server.815/a67781/c31repli.htm#12636], Oracle, 1999.

26. Replication Server: An Architecture for Distributing and Sharing Corporate Information. [URL: http://www.sybase.com/content/1019249/4427RepServerArch_3.v4.pdf], Sybase, 2002. 27 str.

27. Slovenija glede rabe Interneta za dve leti zaostaja za Evropo. [URL: http://www.comtron.si/News_Detail.asp?News_ID=334], STA, Siol, februar 2004.

28. Stan Cavin: An introduction to Secure Sockets Layer (SSL). [URL:http://www.eecs.umich.edu/~aprakash/585/html/ssl-cavin-2000.ppt], EECS, 2000.

29. The Need for Replication Technology. [URL: http://www.unifx.com/data/replication.html], Unif/x, 2004.

30. Tips for Performance Tuning SQL Server Replication. [URL: http://www.sql-server-performance.com/replication_tuning.asp], 2001.

31. Uporaba Interneta: Gospodinjstva 2003/2. [URL: http://www.sisplet.org/ris/ris/dynamic/readpublications.php?sid=36], RIS, januar 2004.

32. What is Database Replication? [URL: http://support.microsoft.com/default.aspx?scid=/support/access/content/repl/replovrvw.asp], Microsoft, 2001.

33. Why Use Database Replication? [URL: http://perso.wanadoo.fr/mnfauvel/Kb/html/using_da.htm], 2004.

34. Wiesmann Matthias et al: Database Replication Techniques: a Three Parameter Classification. [URL: http://www.cs.mcgill.ca/~kemme/papers/srds00.pdf], 2002.

35. Yair Amir, Cipirian Tutu; From Total Order to Database Replication. [URL: http://www.cnds.jhu.edu/pub/papers/AT02_icdcs.pdf], 2002. 10 str.

36. Yair Amir et al: Practical Wide-Area Database Replication. Johns Hopkins University. [URL: http://www.cnds.jhu.edu/publications], 2002.

37. Yakushin Igor: Selected features of DB2 & metadatabase replication. [URL: http://www.ligo.caltech.edu/docs/G/G020413-00.pdf], 2002.

PRILOGE

1. Kontrolne tabele za izvorni strežnik

To so interne tabele, ki se vedno nahajajo samo na izvornem strežniku. Uporabljata jih oba procesa, od tega večinoma proces zajemanja pri registraciji novih tabel in sledenju podatkov iz beleženja. Tabele so naslednje:

1. ASN.IBMSNAP_PRUNCNTL - tabela vsebuje kontrolne podatke o brisanju odvečnih zapisov za trenutni izvorni strežnik. Prevaja časovne naslove beleženj in koordinira brisanje. Tabela je locirana na izvornem strežniku,
1. ASN.IBMSNAP_REGISTER - tabela vsebuje podatke o replikacijskih izvorih, kot so imena repliciranih izvornih tabel, in njihove atribute,
2. ASN.IBMSNAP_TRACE - vsebuje podatke aktivnostih zajemanja (startu, brisanju, stopiranju..) in jih beleži. Če je zagon v hladnem načinu, potem se tabela izprazni in začne beležiti na novo,
3. ASN.IBMSNAP_CCPPARMS - tabela vsebuje parametre, ki so lahko spremenjeni za kontrolo zmogljivosti procesa zajemanja. Ob kreiranju so nastavljeni na privzete vrednosti,
4. ASN.IBMSNAP_UOW - tabela zagotavlja podatkovno integriteto z zapisovanjem transakcij, ki so bile potrjene (torej že prenesene na ciljni strežnik) na izvornem strežniku,
5. ASN.IBMSNAP_WARM_START - tabela vsebuje podatek, ki zajemanju pri toplem načinu zagona omogoča ponovno sinhronizacijo.

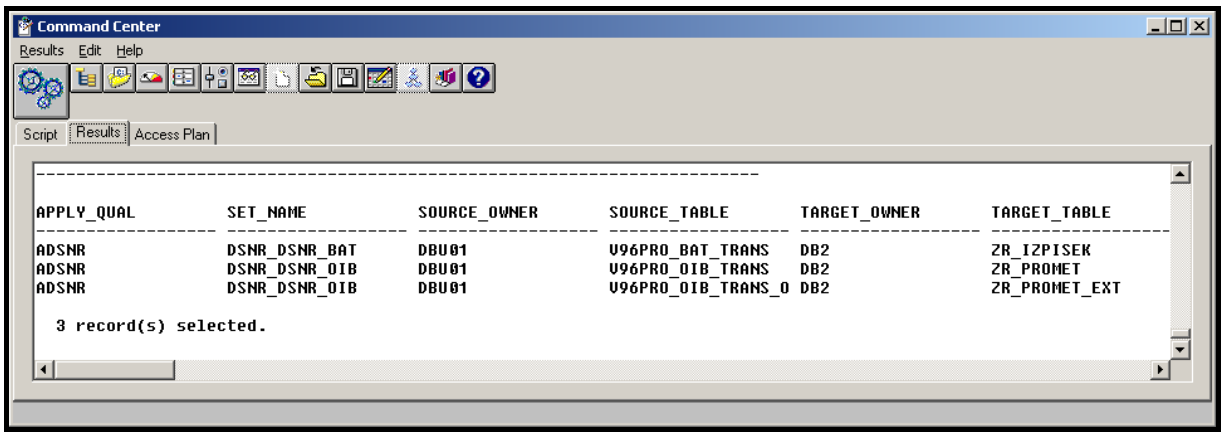
2. Kontrolne tabele za prenos podatkov

Nahajajo se na kontrolnem strežniku. To je lahko hkrati ciljna lokacija (v večini primerov zaradi boljših zmogljivosti) ali pa vmesna postaja (strežnik) med izvorom in ciljem podatkov. Kontrolne tabele so vedno tam, kjer je nameščen tudi proces dodajanja. Tabele so:

1. ASN.IBMSNAP_APPLYTRAIL - tabela beleži zgodovino posodabljanja. Uporabimo jo lahko za revizijo posodabljanja in statistik zmogljivosti. Razberemo lahko naslednje informacije:
    - pove količino vnesenih podatkov,
    - pove količino brisanih podatkov,
    - pove količino spremenjenih podatkov,
    - status (javi in opiše napako),
    - pokaže dnevnik procesa dodajanja storitev za posamezno tabelo,

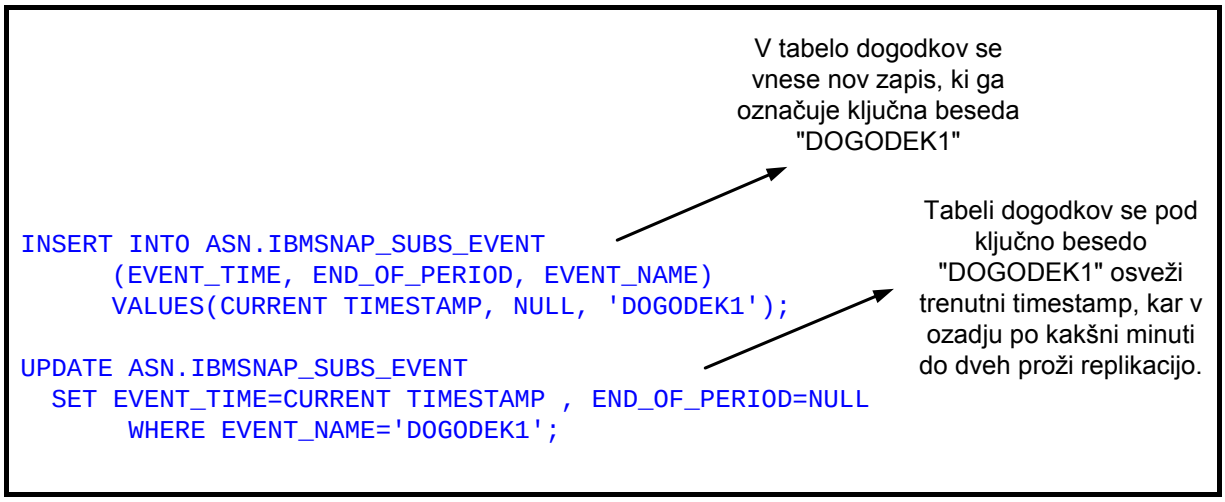
2. ASN.IBMSNAP_SUBS_SET - vsebuje vse zapise za naš kontrolni strežnik. Tu vidimo, če se propagacije v redu izvajajo in kdaj so bile nazadnje izvedene,
3. ASN.IBMSNAP_SUBS_COLS - tabela vsebuje opis stolpcev za vsako tabelo, ki se propagirajo,
4. ASN.IBMSNAP_SUBS_MEMBR - vsebuje zapis (podatek) za vsako ciljno tabelo, v katero se propagira.
   Primer:
   a. SET_NAME – ime niza, ki ga poimenujem sam. Določil sem si standard, ki mi olajša delo in nadzor. Prvi del mi pove ime izvorne baze, drugi ime ciljne baze, tretji pa predstavlja kratico za tabelo (na voljo imam skupaj 18 znakov),
   b. SOURCE – izvor vsebuje dve tabeli, za kateri je pomembna konsistentnost. Zato ju vključim pod istimi imenom niza. Če katerikoli tabeli prenos ne uspe, replikacija poskrbi za povrnitev na stanje pred začetkom prenosa (razveljavitev) in zabeleži napako (SQL kodo),
   c. TARGET – cilj vsebuje dve odgovarjajoči tabeli,
   d. APPLY_QUAL – dodatni kvalifikator, ki določa kontrolno bazo. V našem primeru je kontrolna baza hkrati tudi ciljna,

Slika 1: Izpis poizvedbe v ukaznem centru

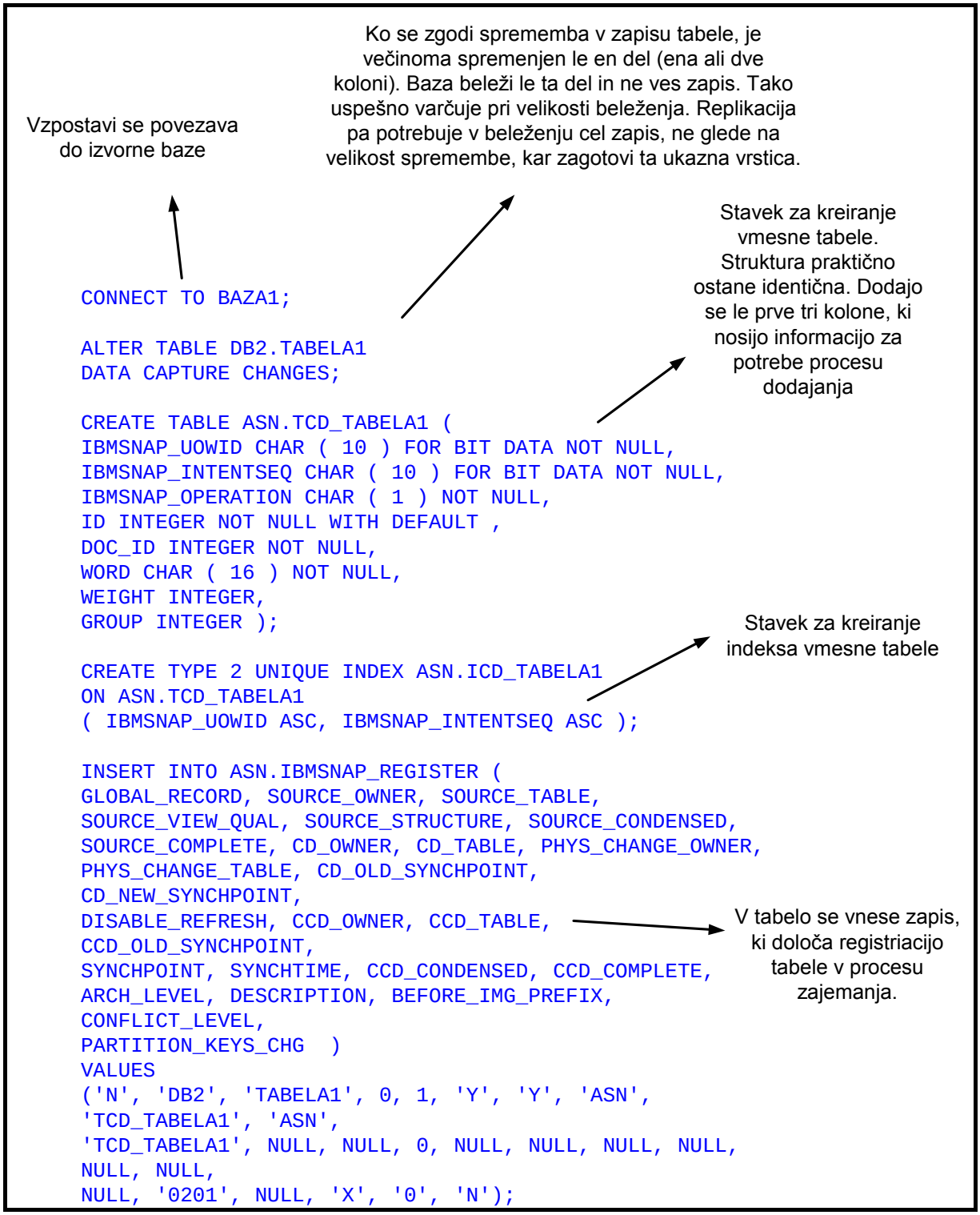


5. ASN.IBMSNAP_SUBS_EVENT - vsebuje podatke o tabelah, ki imajo definirano propagacijo na dogodek. Če želimo sprožiti propagacijo, spremenimo časovno nastavitev in jo določimo s trenutnim časom za izbrani dogodek. Sistem preko kataloških tabel replikacije zazna razliko med časom zadnjega proženja in trenutnim časom. Posledica je aktivacija prenosa za tabele, ki jih določa dogodek. Vnos in časovni popravek dogodka lahko vidimo na sliki 1.

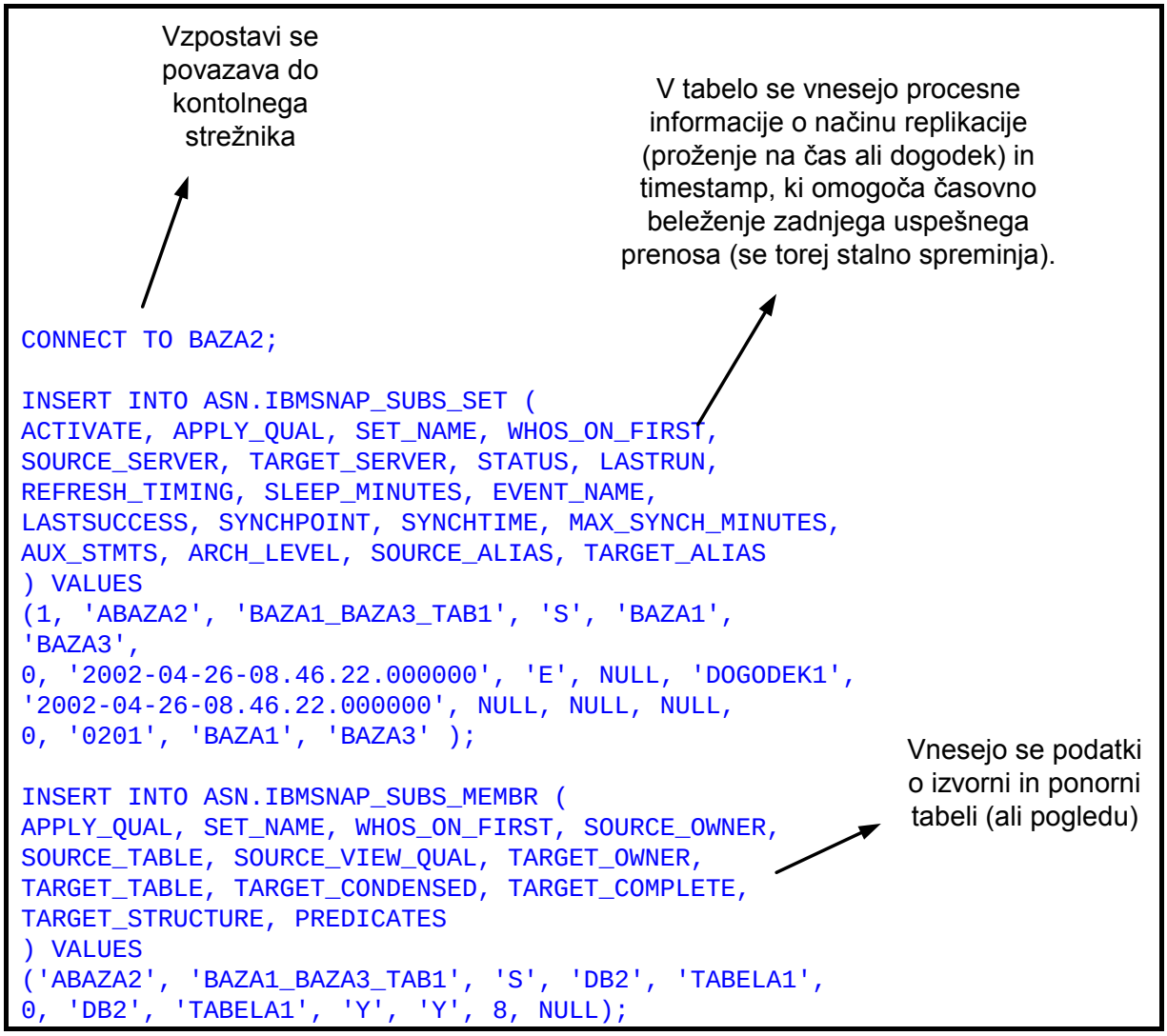
Slika 2: Vnos in popravek podatka za proženje propagacije na osnovi dogodka



Slika 3: Generirana poizvedba, pripravljena na izvedbo

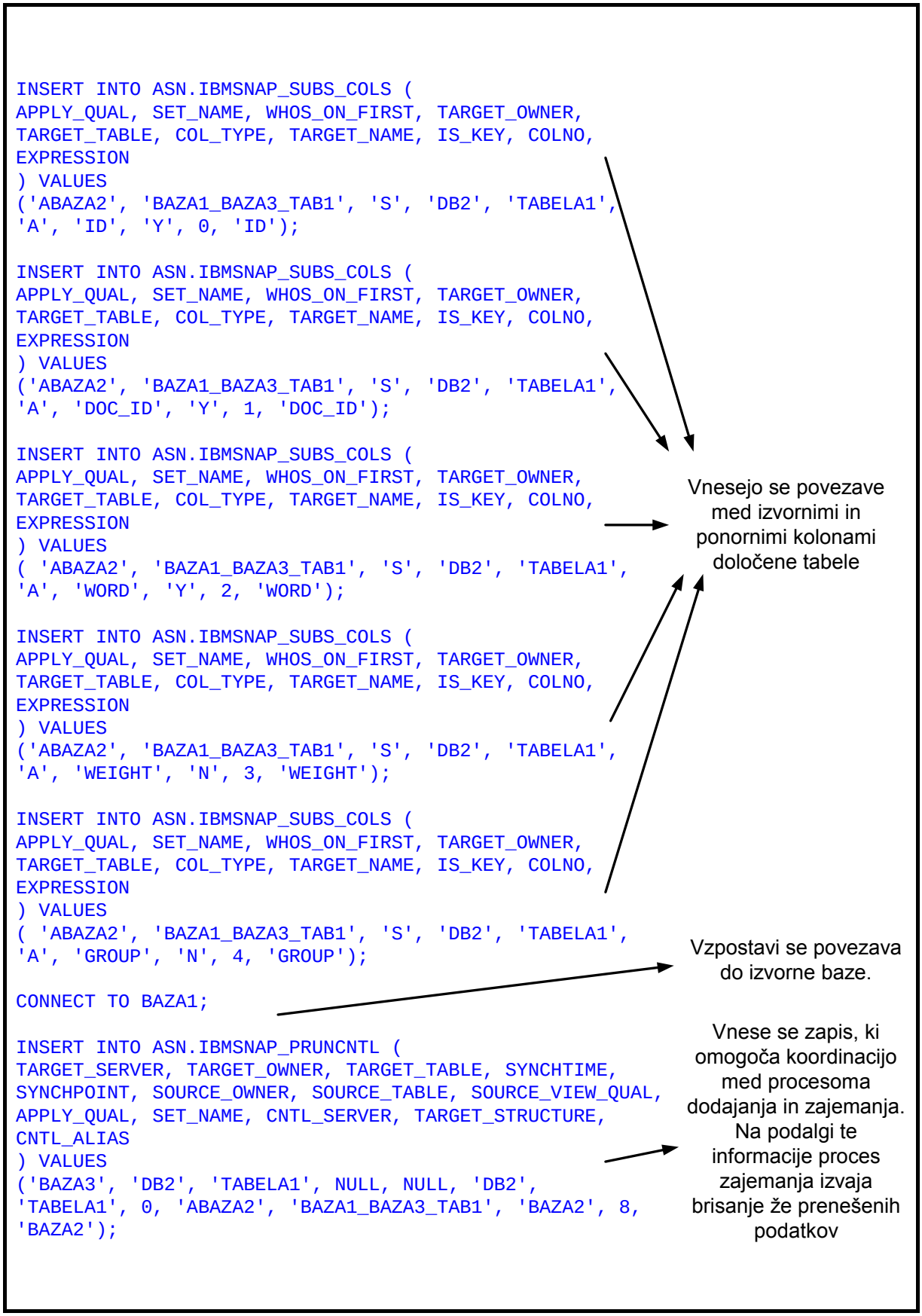


Slika 4: Prvi del generirane poizvedbe za proces dodajanja

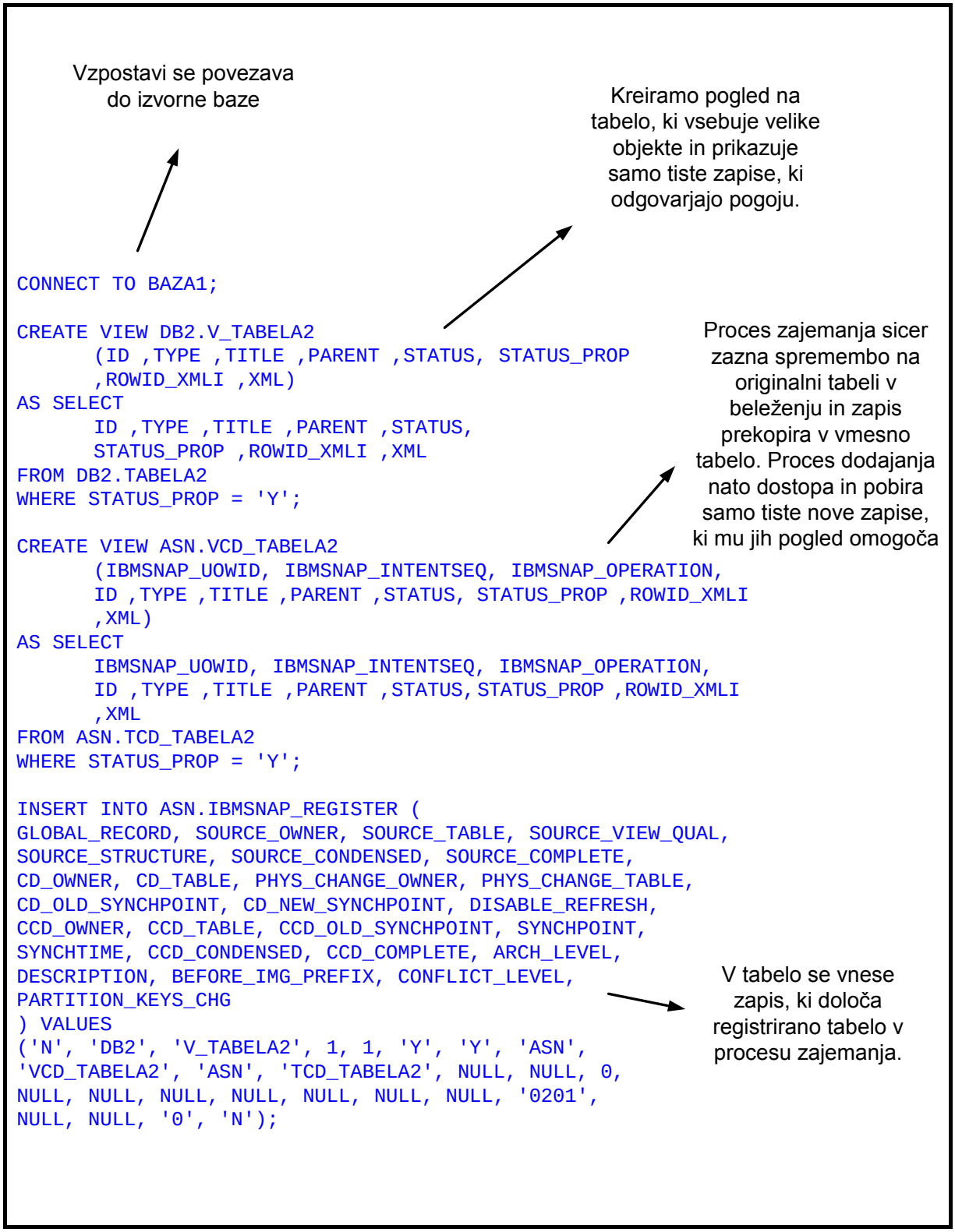



```
CONNECT TO BAZA2;

INSERT INTO ASN.IBMSNAP_SUBS_SET (
ACTIVATE, APPLY_QUAL, SET_NAME, WHOS_ON_FIRST,
SOURCE_SERVER, TARGET_SERVER, STATUS, LASTRUN,
REFRESH_TIMING, SLEEP_MINUTES, EVENT_NAME,
LASTSUCCESS, SYNCHPOINT, SYNCHTIME, MAX_SYNCH_MINUTES,
AUX_STMTS, ARCH_LEVEL, SOURCE_ALIAS, TARGET_ALIAS
) VALUES
(1, 'ABAZA2', 'BAZA1_BAZA3_TAB1', 'S', 'BAZA1',
'BAZA3',
0, '2002-04-26-08.46.22.000000', 'E', NULL, 'DOGODEK1',
'2002-04-26-08.46.22.000000', NULL, NULL, NULL,
0, '0201', 'BAZA1', 'BAZA3' );

INSERT INTO ASN.IBMSNAP_SUBS_MEMBR (
APPLY_QUAL, SET_NAME, WHOS_ON_FIRST, SOURCE_OWNER,
SOURCE_TABLE, SOURCE_VIEW_QUAL, TARGET_OWNER,
TARGET_TABLE, TARGET_CONDENSED, TARGET_COMPLETE,
TARGET_STRUCTURE, PREDICATES
) VALUES
('ABAZA2', 'BAZA1_BAZA3_TAB1', 'S', 'DB2', 'TABELA1',
0, 'DB2', 'TABELA1', 'Y', 'Y', 8, NULL);
```

Slika 5: Drugi del generirane poizvedbe za proces dodajanja



Slika 6: Poizvedbe za kreiranje "filtra" podatkov pred propagacijo




```
CONNECT TO BAZA1;

CREATE VIEW DB2.V_TABELA2
       (ID ,TYPE ,TITLE ,PARENT ,STATUS, STATUS_PROP
       ,ROWID_XMLI ,XML)
AS SELECT
       ID ,TYPE ,TITLE ,PARENT ,STATUS,
       STATUS_PROP ,ROWID_XMLI ,XML
FROM DB2.TABELA2
WHERE STATUS_PROP = 'Y';

CREATE VIEW ASN.VCD_TABELA2
       (IBMSNAP_UOWID, IBMSNAP_INTENTSEQ, IBMSNAP_OPERATION,
       ID ,TYPE ,TITLE ,PARENT ,STATUS, STATUS_PROP ,ROWID_XMLI
       ,XML)
AS SELECT
       IBMSNAP_UOWID, IBMSNAP_INTENTSEQ, IBMSNAP_OPERATION,
       ID ,TYPE ,TITLE ,PARENT ,STATUS,STATUS_PROP ,ROWID_XMLI
       ,XML
FROM ASN.TCD_TABELA2
WHERE STATUS_PROP = 'Y';

INSERT INTO ASN.IBMSNAP_REGISTER (
GLOBAL_RECORD, SOURCE_OWNER, SOURCE_TABLE, SOURCE_VIEW_QUAL,
SOURCE_STRUCTURE, SOURCE_CONDENSED, SOURCE_COMPLETE,
CD_OWNER, CD_TABLE, PHYS_CHANGE_OWNER, PHYS_CHANGE_TABLE,
CD_OLD_SYNCHPOINT, CD_NEW_SYNCHPOINT, DISABLE_REFRESH,
CCD_OWNER, CCD_TABLE, CCD_OLD_SYNCHPOINT, SYNCHPOINT,
SYNCHTIME, CCD_CONDENSED, CCD_COMPLETE, ARCH_LEVEL,
DESCRIPTION, BEFORE_IMG_PREFIX, CONFLICT_LEVEL,
PARTITION_KEYS_CHG
) VALUES
('N', 'DB2', 'V_TABELA2', 1, 1, 'Y', 'Y', 'ASN',
'VCD_TABELA2', 'ASN', 'TCD_TABELA2', NULL, NULL, 0,
NULL, NULL, NULL, NULL, NULL, NULL, NULL, '0201',
NULL, NULL, '0', 'N');
```

3. Slovarček

apply - dodati, priložiti
audit – pregled, revizija
authentication – overjanje, avtentikacija

backup - varnostna kopija
buffer - medpomnilnik

capture – zajemati
client - odjemalec
commit - potrditev

data capture - zajem podatkov
data output - izstopni podatki
data source - podatkovni vir
database - baza podatkov
database administrator - skrbnik podatkovne baze
design - zasnova

encryption - šifriranje
event log - dnevnik dogodkov
extend - razširjen, razširiti
extent - obseg, razteg, velikost

firewall – požarni zid, varnostni zid (za zaščito omrežja organizacije)

hardware - računalniške naprave
host - gostitelj

input - vnos
insert - vstaviti, vriniti
installation – namestitev
IT (information technology) – informacijska tehnologija

join – spajanje, spoj

LAN (Local Area Network) – lokalno omrežje
LOB (Large Object) – veliki objekt
load utility - uslužnostni program za vnos
log - beležiti
log file - dnevnik

mainframe - velik računalnik
middleware – prilagojeni sistemski programi

offline - stanje brez povezave
online - spletni
output - izložek

performance - zmogljivost
propagation - razširjanje

replicator - podvojevalnik
rollback - razveljavitev
router - usmerjevalnik

scalability – nadgradljivost
server – strežnik
sniffer - vohljač
software - programski paket, programje
source – izvoren
SSL (Secure Socket Layer) – varnostni protokol v skladovnici TCP/IP protokolov

table space - tabelarični prostor
target - ciljno
task - opravilo
trigger - prožilec

update - ažuriranje, posodobitev
utility - uslužnostni program

XML (Extensible Markup Language) - razširjeni označevalni jezik