

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

NENAD LJUBOJA

**SPLETNE STORITVE XML
V SISTEMU SAP R/3**

DIPLOMSKO DELO VISOKOŠOLSKEGA
STROKOVNEGA ŠTUDIJA

MENTORICA:
DOC. DR. MOJCA CIGLARIČ

LJUBLJANA, 2008

ZAHVALA

Zahvaljujem se mentorici doc.dr.Mojci Ciglarič za pomoč in nasvete pri izdelavi diplomske naloge. Za koristne pogovore, ter pomoč pri razumevanju mojega dela se želim zahvaliti tudi zaposlenim v podjetju IBM Slovenija, ekipi SAP. Na koncu bi se še rad zahvalil družini za podporo moje študijske poti, vsem prijateljem in še posebej bratu Ljuboja Danijelu, ki mi je ves čas stal ob strani. Hvala tudi vsem, ki niste bili posebej omenjeni, pa ste kakorkoli pomagali pri izdelavi diplomske naloge.

KAZALO

1. UVOD.....	2
2. INFORMACIJSKI SISTEM SAP R/3.....	3
2.1 Arhitektura R/3 sistema	4
2.1.1 Aplikacije sistema SAP R/3	6
2.2 Varnost v sistemu SAP s tehničnega vidika.....	7
2.3 Povezovanje varnostnih vidikov in infrastrukture.....	9
2.4 SAP Web application server	12
2.4.1 Tehnične lastnosti sistema.....	13
3. OSNOVE SPLETNIH STORITEV	16
3.1 Kaj so spletne storitve.....	17
3.1.1 Način komuniciranja	17
3.1.2 Vmesnik	19
3.1.2 Register spletnih storitev	19
3.2 Paradigma.....	21
3.2.1 Kakšna je razlika med dokumentnimi in objektnimi vmesniki	21
3.2.2 Granulacija storitev	23
3.3 Kdaj in kje uporabiti spletne storitve.....	24
3.4 Integracija je ključ do učinkovitih spletnih storitev	26
3.5 Zagotavljanje varnosti in transakcijske integritete	28
3.6 Zaključek poglavja.....	30
4. PRIMER UPORABE SPLETNE STORITVE	31
4.1 Analiza primera	31
4.1.1 Opis problema.....	31
4.2 Možne rešitve - izbira	33
4.3 Izvedba	35
4.3.1 Priprava grobega osnutka	35
4.3.2 Plan priprave.....	36
4.3.2 Dogovor o tehniki prenosa podatkov	36
4.3.3 Dogovor o podatkovnih skupinah in smeri prenosa	37
4.3.4 Povezava SAP – zunanji sistem.....	38
4.3.5 Opis uporabe vmesnika (klic storitve in pošiljanje)	38

4.3.6 Povezava med zunanjim sistemom in SAP	41
4.3.7 Opis uporabe vmesnika (sprejem klica in kreacija pošiljke – transportnega naloga).....	42
4.4 Zaključek poglavja.....	45
5. ZAKLJUČEK.....	46
6. VIRI	47
7. IZJAVA.....	49

POVZETEK

V diplomski nalogi je opisan primer prenosa podatkov med različnima informacijskima sistemoma, ki sta podprta s tehnologijo spletnih storitev.

Namen diplomske naloge je predstaviti primer prenosa podatkov s pomočjo tehnologije spletnih storitev, ki je danes zelo razširjena v svetu računalništva in čedalje bolj priljubljena v poslovnih sistemih. Opisana tehnika je standardna rešitev. V diplomski nalogi bom na kratko opisal tudi par drugih standardov za povezave informacijskega sistema SAP.

V okviru diplomskega dela je bil izveden del projekta, povezava SAP informacijskega sistema z drugim zunanjim sistemom. Namen projekta je bil izmenjava podatkov med SAP informacijskim sistemom in zunanjim sistemom na področju odpreme.

1. UVOD

V diplomski nalogi je predstavljena tehnologija in primer povezovanja dveh različnih sistemov, ki sta podprta z informacijsko tehnologijo. Brez povezovalnega standarda s spletnimi storitvami, bi bilo danes zelo oteženo delo v večini podjetij, saj podjetja poleg osnovnega informacijskega sistema uporabljajo tudi druge informacijske sisteme na področju različnih panog (npr. ena od teh je odprema). Podjetja imajo svoj informacijski sistem rešen z uporabo več informacijskih sistemov, ki so med seboj povezani z različnimi tehnologijami prenosa podatkov tako imenovanimi vmesniki. Tehnologija, ki je v zadnjem času doživela največjo rast uporabe in ji je posvečena velika pozornost, je prenos podatkov s pomočjo spletnih storitev (XML Web Services). Z uporabo tehnologije spletnih storitev si podjetja želijo zmanjšati stroške, zmanjšati neučinkovitost delovne sile ter rešiti marsikatero oviro pri prenosu podatkov med različnimi sistemi.

Tehnologije spletnih storitev ne smemo zamenjati s pomenom ponujanja nekaterih storitev ali prodaje izdelkov preko spleta. Spletna storitev se kot besedna zveza pogosto uporablja za oznako popolnoma nove zvrsti programske opreme.

Namen diplomske naloge ni podroben opis tehnologije spletnih storitev, ampak opis uporabe spletne storitve v informacijskem sistemu SAP. Na konkretnem primeru bom pokazal uporabo in izdelavo vmesnika za prenos podatkov med dvema različnima sistemoma. Z implementacijo na konkretnem primeru sem želel preizkusiti pridobljeno znanje iz šolanja in preveriti prenos teoretičnega znanja na praktičen primer. Za uspešno vpeljavo tehnologije v prakso je poleg teoretičnega znanja potrebno tudi seznanitev z orodji in procesi v dejanskem okolju. Brez upoštevanja vseh parametrov okolja ni mogoče uspešno prenesti tudi tako dobre tehnologije v dejansko uporabo.

V drugem poglavju bom predstavil informacijski sistem SAP R/3. V tretjem poglavju predstavim osnove spletnih storitev. Znanje iz prvih dveh poglavij je osnova za rešitev in izdelavo vmesnika oziroma implementacijo aplikacije. Le ta predstavlja praktični del te diplomske naloge, ki je predstavljen v tretjem poglavju. Opisan je primer uporabe spletne storitve in potek priprave - izdelave vmesnika. V petem poglavju strnem svoje ugotovitve v zaključek diplomske naloge. Sledijo še viri, ki so v šestem poglavju.

2. INFORMACIJSKI SISTEM SAP R/3

V tem poglavju bom predstavil splošne značilnosti sistema SAP R/3, komu je namenjen in kaj omogoča, v nadaljnjih poglavjih pa se bom vse bolj približeval bistvu diplomskega dela. V tem poglavju bom predstavil arhitekturo sistema in varnostne vidike tehnične zasnove sistema. V poglavju o spletnih storitvah bom predstavil osnovne pojme spletnih storitev in predstavil, zakaj se uporabljajo spletne storitve. Razumevanje slednjega je ključno za razumevanje praktičnega dela v četrtem poglavju.

Leta 1972 je pet nekdanjih IBM-ovih tehnologov ustanovilo podjetje SAP (System Analysis and Program Development) v majhnem nemškem mestu Mannheim. Imeli so vizijo, razviti programski paket, ki bi lahko integriral poslovne rešitve, torej zagotavljal boljši pretok informacij. Skušali so izdelati programsko rešitev, ki bi v celoti podpirala informacijske potrebe podjetja in s tem zmanjšala redundantnost in izboljšala konsistentnost podatkov, kar bi rešilo enotno podatkovno okolje z modularno zgradbo. Ta rešitev naj bi imela enotno strukturo in uporabniški vmesnik. Uporabljena naj bi bila za več različnih panog in podprta v različnih jezikih.

Približno leto kasneje je prišel na trg njihov prvi izdelek – programski paket, namenjen finančnemu računovodstvu. Paket je bil znan kot R/1, pri čemer črka »R« pomeni pretok informacij v realnem času (real-time data processing). Ta sistem je bila osnova za nadaljnji razvoj in kmalu je sledilo rojstvo SAP R/2. To je bila SAP-ova prva generacija ERP rešitev, ki se je uporabljala na velikih računalnikih (mainframe computers). Ta rešitev je prav tako vsebovala možnost izbire različnih jezikov in valut. Poleg tega in vseh ostalih inovacij je rast podjetja SAP dosegla visoko raven.

Preteklo je kar nekaj časa, ko je SAP poslal na trg svojo novo generacijo celovite informacijske rešitve za podjetja SAP R/3, ki je po vsebini enaka R/2, ampak z izvedbenega vidika popolnoma nova aplikacija z uporabo tro-nivojske arhitekture odjemalec/strežnik in ne zgolj naslednik starejšega sistema.

SAP R/3 je odprt programski sistem, ki deluje v okolju odjemalec/strežnik. Osnovan je za učinkovito upravljanje podatkov podjetja, predstavlja celovito rešitev za številne funkcije v organizaciji ter lahko deluje kot podpora za vse glavne procese. Celotni tok podatkov v R/3

sistemu deluje integrirano, kar pomeni, da je potrebno podatke vnesti le enkrat, sistem pa avtomatsko sproži oziroma posodablja druge logično povezane funkcije ali podatke. Tako so vsi poslovni procesi podjetja med seboj povezani in razporejeni tako, da se sprememba na enem področju podjetja izraža na drugem področju. Ključno za uspešnost celovite rešitve SAP R/3 je strategija izdelave odprtih rešitev. To pomeni, da aplikacije lahko delujejo na različnih platformah (Unix, NT, OS/400) in podatkovnih zbirkah (Oracle, MS SQL, DB2, Infomix). Aplikacije zato niso vezane na določenega ponudnika programske oz. strojne opreme. Sistem SAP R/3 je zgrajen iz modulov, ki pokrivajo osrednje in specifične poslovne procese, z razmahom interneta pa razširja svoje funkcionalne zmogljivosti pod rešitvijo mySAP.com. Ta omogoča tudi elektronsko poslovanje, upravljanje odnosov s kupci in dobavitelji ipd.

V zadnjih nekaj letih pa je SAP naredil še korak naprej. Ponudba novih funkcionalnosti je bila razširjena, v sistem pa so bile vgrajene najnovejše tehnologije in aplikacije. Poglavitni korak v tem preskoku je prehod iz arhitekture, ki je zasnovana na ABAP/4 programskem jeziku, na novo arhitekturo SAP NetWeaver, ki med drugimi vsebuje komponente SAP Enterprise Portal, SAP Exchange Infrastructure, J2EE in mobilno infrastrukturo. Ne eni strani prihod vseh teh novih tehnologij in izboljšav funkcionalnosti povečuje možnosti za integracijo podjetij ter naročnikov (kupcev), na drugi strani pa zahtevajo več pozornosti pri zmanjševanju nevarnosti, ki jih nove pridobitve prinašajo s seboj.

2.1 Arhitektura R/3 sistema

V uvodnem poglavju smo spoznali splošne značilnosti sistema od začetka njegovega razvoja pa do sedaj in smo ugotovili, da se je sistem precej spremenil. V tem poglavju se osredotočim na predstavitev troslojne zasnove trenutne različice SAP R/3 sistema, še posebej pa predstavim organizacijo aplikacijske plasti, ki je v pomoč razumevanju besedila, ki sledi v četrtem poglavju.

SAP R/3 sistem je popolnoma na novo razvit sistem in kot mogoče narekuje ime, ni le izboljšava njegovega predhodnika SAP R/2. Ko se je začel razvoj sistema R/3 sredi osemdesetih let 20. stoletja, se je SAP odločil za razvoj tro-nivojske odjemalec/strežnik arhitekture, ki se deli na tri plasti:

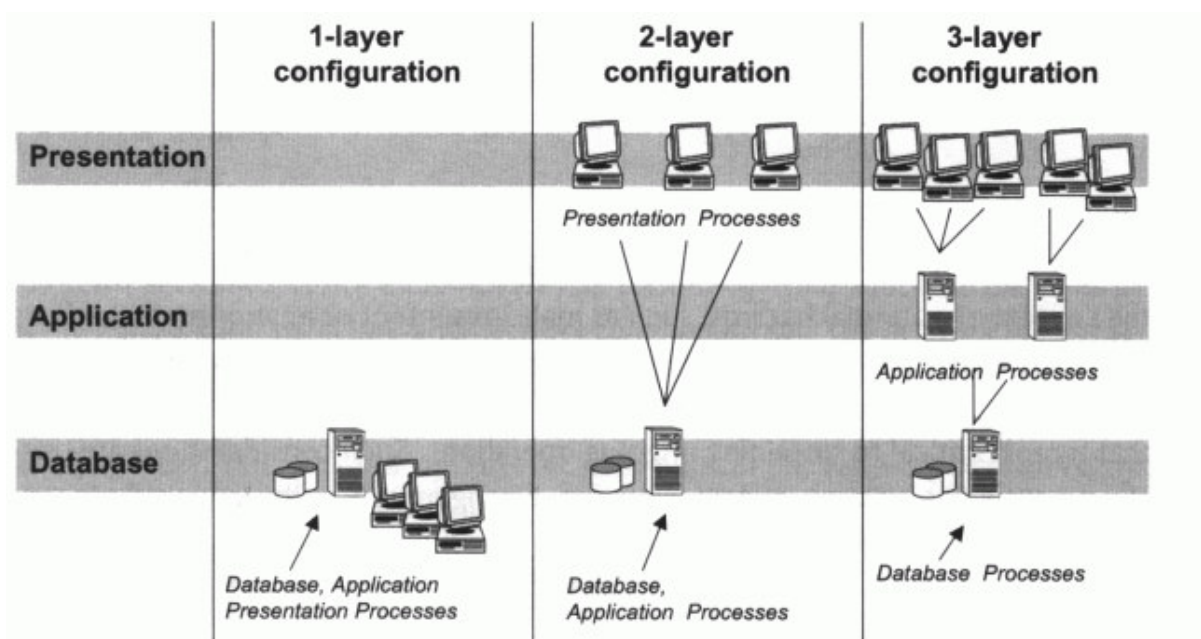
- podatkovna plast;
- aplikacijska plast;

- prezentacijska plast.

Hranjenje podatkov se izvaja v podatkovni zbirki (podatkovna plast), medtem ko se obdelava izvaja na aplikacijskem strežniku (aplikacijska plast). Vmesnik med uporabnikom in sistemom predstavlja SAP GUI in se tako uvršča na prezentacijsko plast.

Slika 1 predstavlja arhitekturo odjemalec/strežnik SAP R/3 sistema in povezave med prezentacijsko in aplikacijsko plastjo ter aplikacijsko in podatkovno plastjo v odvisnosti od števila nivojev arhitekture.

Vir: SAP AG, SAP Authorization System, 2003, str. 20



Slika 1 - Tronivojska arhitektura odjemalec/strežnik

Prezentacijska plast običajno zajema končno število osebnih računalnikov z nameščeno aplikacijo SAP GUI. SAP GUI ni emulator terminala, ampak samostojna aplikacija, ki je zadolžena za prikaz podatkov v sistemu R/3. Med SAP GUI odjemalci in R/3 sistemom zato ni posebno visokih zahtev, kar se tiče komunikacije. Vse poizvedbe, ki so zagnane iz vmesnika (SAP GUI), se obdelujejo izključno na preostalih dveh nivojih.

Aplikacijske strežnike, kjer tečejo glavne SAP R/3 aplikacije, uvrščamo v aplikacijsko plast. Če se pojavi zahteva po večjih in hitrejših obdelavah v določenih SAP R/3 aplikacijah, je to možno rešiti z dodajanjem novih strežnikov.

Zadnji-podatkovni nivo pa predstavlja zgolj strežnik, na katerem teče kateri od obstoječih sistemov za upravljanje z zbirkami podatkov (Oracle, DB2, Informix ipd.).

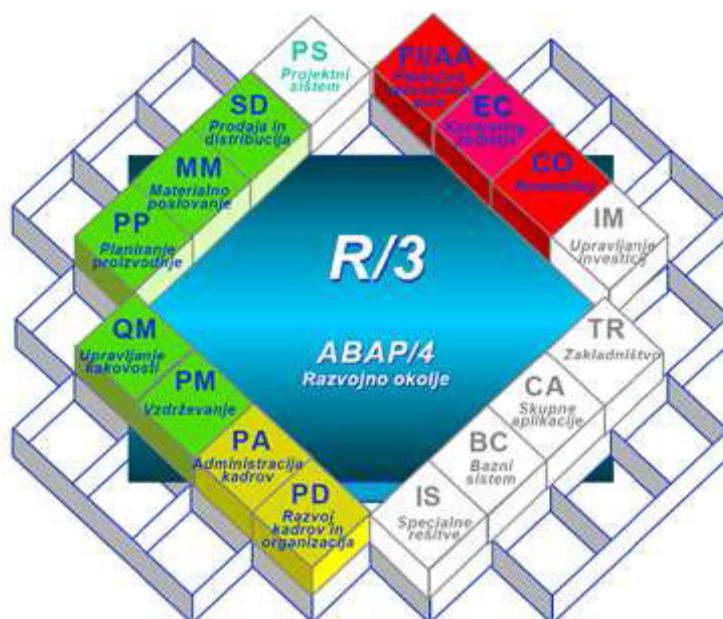
2.1.1 Aplikacije sistema SAP R/3

V tronivojski arhitekturi sistema SAP R/3 se ena izmed plasti osredotoča prav na izvajanje aplikacij, kjer se prepletata tako varnostni kot vsebinski vidik. Na tem mestu je aplikacijski nivo predstavljen le iz vsebinskega vidika, torej razdelitve aplikacij v posamezne module oz. funkcijska področja, v kasnejših poglavij pa je predstavljeno tudi zagotavljanje nadzora dostopa pri izvajanju aplikacij znotraj SAP okolja.

Modularnost sistema omogoča, da se podjetja lahko odločijo za uvedbo celotne programske rešitve SAP R/3 ali posameznih modulov. Standardni paket pokriva vse funkcije, ki jih večja in srednja podjetja uporabljajo, vključno s proizvodnjo, financami, prodajo in distribucijo ter kadrovskimi sistemi, do specifičnih aplikacij prilagojenih posamezni dejavnosti podjetja. Module sistema R/3 lahko okvirno združimo v tri osnovne skupine modulov oz. funkcijskih področij.

1. Finance, ki ponujajo uporabniku celovito sliko računovodskih funkcij z mnogimi pripomočki za izdelavo poročil za podporo poslovodstvu pri odločanju. Primerne so tudi za mednarodne organizacije, saj podpirajo uporabo različnih valut in večjezičnost.
2. Logistika, ki predstavlja najširše področje aplikacij in obsega največje število podmodulov. Upravljajo vse procese, vključene v nabavno verigo: od oskrbe s surovinami do dostave izdelka kupcu.
3. Kadrovski sistem pa vsebujejo podporo vsem potrebnim poslovnim procesom za učinkovito upravljanje s človeškimi viri pri zaposlovanju, razporejanju in razvoju človeških virov ter obračunu plač.

Vir: Hernandez, SAP R/3 Handbook, 2000, str. 13



Slika 2 - Prikaz (pod)modulov osnovnih funkcijskih področij

Slika 2 prikazuje natančnejši prikaz zgoraj navedenih osnovnih funkcijskih področij. Z rdečo barvo so označeni moduli, ki so zajeti v osnovni skupini finance, moduli zelene barve spadajo v skupino logistika, rumeni moduli pa v kadrovski sistem. Funkcionalnosti posameznih modulov po skupinah v nadaljnjem besedilu nisem podrobneje razlagal, ker se vsebina posameznih modulov ne prekriva s področjem zagotavljanja nadzora dostopa in varnosti v sistemu in gre za popolnoma ločeni vsebini.

2.2 Varnost v sistemu SAP s tehničnega vidika

V prejšnjem poglavju smo spoznali tronivojsko arhitekturo sistema SAP R/3 in organizacijo aplikacijske plasti znotraj sistema. Za naslednje poglavje v katerem bom opisoval spletne storitve, je dobro tudi predstaviti področje varnosti znotraj sistema SAP R/3 tako, da s tem poglavjem spoznajmo varnost v SAP R/3 z vidika tehnične zasnove.

Tehnična varnost se deli v tri skupine:

- omrežna varnost;
- aplikacijska varnost;
- fizična varnost.

Omrežna varnost

Izraz omrežna varnost obsega vso tehnologijo, ki podpira varnost omrežne infrastrukture in omrežne komunikacije. Omrežna infrastruktura je sestavljena iz fizičnega omrežja in vseh komponent, ki upravljajo komunikacijo znotraj omrežja (kot so usmerjevalniki, prehodi, požarni zidovi). Omrežna komunikacija pa se nanaša na možnosti za komunikacijo med dvema končnima točkama. Primer v SAP sistemu je lahko naslednji: »Ali lahko SAP GUI (FrontEnd), ki teče na računalniku A, v omrežju B komunicira z aplikacijskim strežnikom C in omrežjem D?« Zato ker je potrebno vsako vrsto komunikacije znotraj omrežja na uporabljeni infrastrukturi ustrezno zaščititi, je potreben po meri narejen varnosti koncept.

Aplikacijska varnost

V to skupino varnosti je potrebno vključiti vse potrebne varnostne vidike, ki nastopijo znotraj aplikacije. To pomeni, ko se uporabnik preko SAP GUI prijavi na SAP R/3 aplikacijski strežnik, lahko izvede le tiste operacije, za katere je pooblaščen (npr. vpogled v prodajne dokumente, kreiranje materiala ipd.). Varnost na nivoju aplikacije mora prav tako določiti, katere aktivnosti (operacije) uporabnik ne sme izvesti, kot je spreminjanje svojih osebnih podatkov ali podatkov o plači.

Oba elementa varnosti, prijavo in identifikacijo, je potrebno vključiti v postopek zagotavljanja varnosti za posamezno aplikacijo. Primer je lahko zagon aplikacije SAP GUI, ki pomeni v prvem koraku prijavo na aplikacijski strežnik. Aplikacijo zase pa predstavlja tudi operacijski sistem, na katerem tečejo aplikacije, konkretnije SAP GUI. Za prijavo v operacijski sistem so uporabnikom dodeljena uporabniška imena in gesla, na podlagi katerih so kreirani uporabniški profili. Preko teh so uporabnikom dodeljene pravice za izvajanje operacij na sistemu (npr. namestitve nove aplikacije ali zagon lastnih programov). Sistem SAP R/3 upravlja z določenimi podatki tudi na nivoju operacijskega sistema, na katerem teče (večinoma UNIX ali Windows NT). Vsak uporabnik, ki se lahko prijavi na nivoju operacijskega sistema, predstavlja svojevrstno nevarnost, saj ta nivo dopušča nepooblaščen dostop do R/3 zbirk podatkov ali datotek kot tudi spreminjanje programov na razvojnem sistemu. Izvajanje sistemskih ukazov preko ABAP funkcij ima prav tako velik vpliv na varnost v sistemu.

Zato mora obstajati koncept nadzora dostopa za posamezno aplikacijo – za operacijski sistem in za aplikacije, ki tečejo na tem sistemu. Mehanizem SAP avtorizacij omogoča nadzor dostopa le za aplikacije, ki tečejo znotraj SAP okolja.

Fizična varnost

Fizična varnost se nanaša na integriteto posameznih računalnikov, omrežnih komponent in podatkov, mišljeno v kontekstu nevarnosti, kot zlonamerno dejanje ali pa vpliv okolja (npr. požar ali ostale naravne katastrofe). Ko si poskusimo predstavljati celotni koncept varnosti v SAP R/3 okolju, moramo upoštevati fizično varovanje računalnikov in podatkov, ki imajo največji vpliv na dejavnost(-i) v podjetju. Taka upoštevanja ponavadi zajamemo v plan obnove po nesrečah. Opravila na tem področju zajemajo zrcaljenje podatkov, pogoste varnostne kopije in varnostna merila za fizično varovanje tehničnih okolij – požarna vrata, klimatske naprave, rešetke na oknih itd.

Še bolj natančno deljenje posamezne skupine v podskupine omogoča bolj podrobno predstavo različnih tveganj, katerim je arhitektura SAP lahko podvržena. Razdelitev v tri skupine, kot smo jo predstavili tukaj, služi kot osnovna predstava, kako zasnovati varnostni koncept, možne pa so tudi drugačne razdelitve. Meje med posameznimi kategorijami niso fiksne. Če pogledamo omrežno in aplikacijsko varnost lahko ugotovimo, da sta precej povezani. Na podlagi vnaprej določenih zahtev je možno nekatere varnostne težave in tehnologije uvrstiti tudi v druge kategorije.

2.3 Povezovanje varnostnih vidikov in infrastrukture

V poglavju o varnosti s tehničnega vidika smo spoznali pojme omrežne, aplikacijske in fizične varnosti. V poglavju povezovanja varnostnih vidikov in infrastrukture opišemo prepletanje prej omenjenih pojmov s posameznim slojem arhitekture SAP R/3 sistema – prezentacijskim, aplikacijskim in podatkovnim.

Arhitektura, ki smo jo opisali v prejšnjem razdelku, je osnova za analizo in preučevanje različnih varnostnih vidikov obsežnejših SAP R/3 okolij. Gledano z vidika dostopnih dovoljenj in restrikcij, morajo biti pooblastila (avtorizacije) v sistemu SAP dodeljene v skladu s celotnim

konceptom varnosti v SAP okolju, saj le tako lahko dosežemo pravilno izvedbo in razmejenost med vlogami.

- Infrastruktura omrežja

Prvi korak je določitev celotne mrežne infrastrukture. Ta povezuje podatkovni, aplikacijski in prezentacijski nivo in je kritična točka varnosti v sistemu, saj predstavlja prvi nivo zaščite. Vsi nepooblaščen poskusi dostopa morajo biti zavrženi. Model omrežne infrastrukture za aplikacijo kot je SAP R/3, ki ni edina v tem modelu in je večinoma del kompleksnejšega okolja, potrebuje dosledno skrb pri načrtovanju in kasnejšem zagotavljanju varnosti. Podjetje SAP zagotavlja varnost v omrežju s produktoma SAProuter in Secure Network Communication (SNC).

- SAProuter SAProuter je namenska naprava - požarni zid, ki podpira filtriranje in usmerjanje paketov specifičnih SAP-ovih komunikacijskih protokolov. Medtem ko običajni požarni zidovi pustijo katerikoli vrsto SAP-ove komunikacije, ki prihaja iz določenega omrežja, SAProuter prepušča le povezave od verificiranih uporabnikov. S tem načinom SAProuter pripomore k bolj natančnemu nadzoru SAP R/3 prometu po omrežju, kot to omogočajo običajni požarni zidovi.

- SNC SNC predstavlja aplikacijski nivo, ki je vgrajen v vsako komponento SAP R/3 sistema in omogoča souporabo z zunanjimi produkti za zagotavljanje varnosti. Za primer, ko SNC zagotavlja varnost na aplikacijskem nivoju, lahko navedemo komunikacijo med vmesnikom SAP GUI in aplikacijskimi strežniki ali komunikacijo med dvema aplikacijskima strežnikoma. Omogoča nam tri nivoje varnosti:

- najnižji nivo varnosti vsebuje le verifikacijo, ki preveri veljavnost identitet, ki med seboj komunicirata;
- na srednjem nivoju je dodana še funkcionalnost, ki zagotavlja integriteto podatkov; vsako spremembo podatkov, ki se zgodi med prenosom, je možno zaznati na tem nivoju;
- najvišji nivo zagotavlja vse funkcije prejšnjih nivojev, omogoča pa še zaščito podatkov med prenosom tako, da jih pred prenosom šifrira.

Poglejmo si še opis posameznih nivojev – podatkovni, aplikacijski in prezentacijski nivo – v okviru varnostnih vidikov sistema (omrežni, aplikacijski in fizični).

- Podatkovni nivo

V večini namestitev SAP R/3 sistema je v podatkovni nivo zajet eden ali več strežnikov s sistemom za upravljanje s podatkovnimi zbirkami. Ta strežnik se nahaja v ločenem omrežju, ki dopušča komunikacijo samo z aplikacijskimi strežniki. Aplikacijska varnost podatkovnega nivoja pokriva varnost operacijskega sistema strežnika, sistema za upravljanje s podatkovno zbirko in ostalih posebnih aplikacij, ki nudijo podporo pri shrambi podatkov. Za zagotavljanje fizične varnosti sistema se morajo strežniki nahajati v strežniških sobah, ki so opremljene s kakšnim od omenjenih sistemom nadzora dostopa. Neposreden dostop do podatkovne baze mora biti vsem uporabnikom onemogočen, strežnik pa popolnoma izoliran od omrežja podjetja, ker lahko le tako onemogočimo nepooblaščen dostop do podatkov, ki se nahajajo na podatkovnem strežniku.

- Aplikacijski nivo

Kot že omenjeno se v aplikacijski sloj uvrščajo aplikacijski strežniki na katerih tečejo SAP R/3 aplikacije. Aplikacijski strežniki se lahko nahajajo v ločenem omrežju ali pa v istem omrežju kot podatkovni strežniki. V tem primeru mora SAProuter nadzorovati celotno komunikacijo v preostalem delu omrežja. Na tem nivoju je še posebej pomembna aplikacijska varnost SAP R/3 sistema, kjer mora biti za varnost najprej poskrbljeno v gostujočem operacijskem sistemu. V SAP R/3 aplikacijah je na tem nivoju poskrbljeno za varnost z vsemi pomembnejšimi preverjanji pooblastil (avtorizacij), ki se odvijajo na tem nivoju. SAP-ova aplikacija preveri, če je posamezni uporabnik pooblaščen za izvajanje zahtevanih operacij. Za fizično varnost veljajo enake zahteve kot na podatkovnem nivoju.

- Predstavitveni nivo

Tipično arhitekturo prezentacijske plasti predstavlja množica osebnih računalnikov, ki imajo nameščen vmesnik SAP GUI. V večini ti računalniki kar pripadajo internemu omrežju podjetja in imajo s tem že omejen dostop s kako od metod nadzora dostopa.

Nedolgo nazaj je postala praksa, da so se podjetja začela odločati za uporabo zunanjega (oddaljenega) dostopa svojim upraviteljem sistema do svojih SAP R/3 sistemov. V takem primeru mora biti požarni zid podjetja pravilno nastavljen, da je tako povezavo sploh lahko

vzpostaviti, kar pa s seboj prinese (nujno) uporabo dodatnih nadzornih mehanizmov za preprečevanje nepooblaščenega dostopa.

Na aplikacijskem nivoju varnosti predstavlja gostujoči operacijski sistem prvo kontrolno točko, ko se mora uporabnik prijaviti v sistem. Naprednejše verifikacijske metode so lahko uporabljene in so priporočljive (biometrija, pametne kartice). Shranjevanje podatkov na lokalni medij je prav tako možno onemogočiti na nivoju SAP GUI vmesnika.

Na prezentacijskem nivoju tudi fizični vidik varnosti ni zanemarljiv in ga je potrebno podrobno preučiti, četudi ni tako kritičen kot na aplikacijski in podatkovni plasti. Varnostno preverjanje je v največ primerih omejeno na preverjanje identitete na vhodu v zgradbo podjetja.

2.4 SAP Web application server

Do tukaj smo spoznali sistem SAP do te mere, da vemo katere funkcionalnosti nam ponuja, in kako je v osnovi zgrajen (troslojna arhitektura). Tekoče poglavje se v povezavi z opisom SAP-ovega spletnega aplikacijskega strežnika, kot glavne tehnične komponente SAP NetWeaver arhitekture, osredotoči na predstavitev postopkov avtentikacije z uporabo SSO in področja varnosti v SAP aplikacijah. Zlasti zadnje področje je natančneje opisano, saj je ključno za razumevanje praktičnega dela v četrtem poglavju.

SAP Web AS je glavna tehnična komponenta SAP NetWeaver arhitekture in služi kot aplikacijska plast za izvajanje vseh aplikacij, razvitih v programskih jezikih ABAP ali Java. Kot osnova služi tudi komponentam SAP Enterprise Portal, SAP eXchange Infrastructure in SAP Mobile Infrastructure.

V preteklosti je bila vloga SAP Application Server komponente izključno zagotavljanje tehnične osnove SAP R/3 sistema. Vsi programi znotraj SAP sistema so bili napisani izključno v programskem jeziku ABAP. S prihodom nove arhitekture – SAP NetWeaver – je bilo okolje, ki je teklo izključno s podporo ABAP-a, razširjeno tudi s podporo izvajanju Java aplikacij na osnovi J2EE virtualnega stroja. Sto potezo je SAP stopil v korak s časom, v katerem se je uporaba internetnih tehnologij precej razmahnila. Od tod tudi sprememba imena v spletni aplikacijski strežnik (SAP Web Application Server). Ime mogoče zavaja, vendar SAP-ov

aplikacijski strežnik ni le »klasični« strežnik za izvajanje R/3 odjemalec-strežnik aplikacij, ampak bi ga lahko bolj točno opisali kot vmesni nivo (middleware technology platform) za izvajanje spletnih aplikacij.

Poleg izvajanja ABAP aplikacij, SAP Web AS podpira tudi Business Server Pages (BSP), Java Server Pages (JSP) in Web Dynpro za ABAP in Javo. Aplikacije, ki so izvedene z Web Dynpro tehnologijo, bodo postopoma nadomestile starejše ABAP Dynpro aplikacije.

Poleg običajnih opravil, ki jih izvaja aplikacijski strežnik za različna programska okolja, pa na osnovi spletnih storitev SAP Web AS omogoča tudi integracijske storitve kot sta:

- SOAP (Simple Object Access Protocol) protokol in
- klici oddaljenih funkcij (RFC – Remote Function Calls) za BAPI-je (Business Application Programming Interfaces).

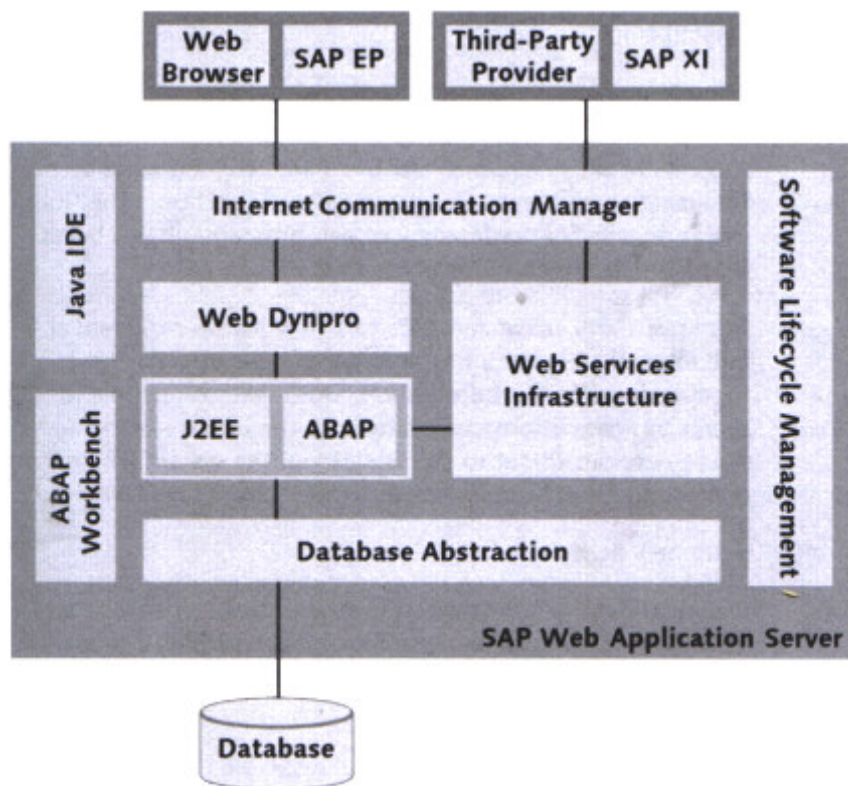
2.4.1 Tehnične lastnosti sistema

Centralna komponenta sistema SAP Web AS je ABAP-kopica (ABAP Stack), ki predstavlja okolje za izvajanje ABAP aplikacij. To so aplikacije ali BSP strani, ki so razvite neposredno z ABAP programskim jezikom. Komponenta sistema, imenovana ABAP Workbench, se lahko uporabi za razvoj in analizo delovanja takih programov, s čimer odkrivamo in odpravljamo napake tekom razvoja. Aplikacije za dostop do podatkovne zbirke uporabljajo abstrakcijo podatkovne zbirke, ki ga predstavlja nevtralni nivo za dostop do zbirk podatkov različnih proizvajalcev. SAP Web AS ima tudi svojo logično podatkovno zbirko, ki pa je ločena od SAP Web AS J2EE virtualnega stroja.

Kot je bilo že omenjeno v začetku poglavja, je mesto v novi arhitekturi dobila Java. Uvedba Jave je bila nujna poteza, zaradi bolj uspešne integracije sistema s sodobnimi internetnimi tehnologijami kot so aplikacijski strežniki Java (IBM WebSphere, BEA WebLogic) in .NET (Microsoft). SAP Web AS J2EE je v bistvu J2EE aplikacijski strežnik (Java 2, Enterprise Edition), ki omogoča okolje za izvajanje Java aplikacij kot so JSP spletne strani in Enterprise JavaBeans. SAP Web AS J2EE prav tako dostopa do svoje podatkovne baze z uporabo abstrakcije podatkovne zbirke. Obstaja tudi komunikacija med SAP Web AS ABAP in SAP Web AS J2EE, ki omogoča preko Javine povezovalne arhitekture (Java Connection

Architecture) izvajanje klicev RFC. Java ima tako kot ABAP tudi svoje razvojno okolje – Java IDE.

Vir: SAP AG, SAP Security And Authorizations, 2006, str. 137



Slika 3 - Tehnična zasnova SAP Web AS sistema

Web dynpro aplikacije lahko razvijemo bodisi z ABAP programskim jezikom ali pa z Javo. Vmesnik ICM (Internet Connection Manager) z uporabo protokola HTTP(S) upravlja vse spletne aplikacije – programe, ki so dostopni na svetovnem spletu in tiste, ki so dostopni v lokalnem omrežju. To pa ne velja le za namestitve, ki vsebujejo SAP Web AS J2EE, ampak tudi za ABAP dynpro aplikacije, saj je ITS (Internet Transaction Server) že integriran v ICM. ICM nadzoruje tudi spletne storitve in zato predstavlja integracijski in komunikacijski vmesnik uporabniku in ostalim storitvam, ki izvajajo klice določenih funkcij znotraj ICM. SAP-ov sistem za izmenjavo informacij (SAP Exchange Infrastructure) tudi uporablja vmesnik ICM. Podprtih je večina standardnih protokolov, vključno z HTTP(S) in SOAP (XML).

Za upravljanje življenjskega cikla aplikacij je znotraj sistema standardiziran postopek tako za razvoj in testiranje ABAP kot Java aplikacij. Tudi Javanske aplikacije imajo lahko klasično okolje s tremi sistemi (razvojno, testno in produkcijsko), čeprav se razvojno okolje razlikuje od

okolja ABAP Workbench in ABAP Transport Management (sistem za prenos objektov na ostale sisteme).

3. OSNOVE SPLETNIH STORITEV

V prejšnjem poglavju sem predstavil informacijski sistem SAP R/3, kateri je v povezavi s slednjim ključen za razumevanje praktičnega dela v četrtem poglavju. V tem poglavju bom predstavil splošne značilnosti spletnih storitev, kaj so spletne storitve, kako delujejo, komu so namenjene in kaj nam omogočajo. Zakaj ravno spletne storitve bom pojasnil v četrtem poglavju, kjer sem opisal možne rešitve povezovanja dveh različnih sistemov med seboj.

Naj s kratkim povzetkom predstavim slednje poglavje. Spletne storitve se uveljavljajo kot ključna tehnologija za elektronsko poslovanje. Omogočajo povezljivost med informacijskimi sistemi ne glede na programski jezik, platformo ali operacijski sistem. Se sliši znano? Res je! Natančno to sta obljubljala in omogočala tudi komponentna modela CORBA in COM+. V tem poglavju bom analiziral, v čem se spletne storitve razlikujejo od obstoječih komponentnih modelov, kje so njihove prednosti, kdaj jih uporabiti in kako zasnovati arhitekturo za učinkovite spletne storitve. Osvetlil bom področja, ki jih spletne storitve še ne naslavljajo in odgovoril na vprašanje, ali spletne storitve zadoščajo za elektronsko poslovanje.

Spletne storitve so še vedno najbolj vroča tehnologija na področju informatike. Po mnenju vodilnih analitikov bi se naj znašle spletne storitve na vrhu, znane pod imenom »hype-curve«. Pozicija na vrhu ne pomeni ničesar drugega, kot veliko medijsko podporo, veliko pisanja, pogovarjanja, predvsem pa precenjena pričakovanja – nekako v smislu nove tehnologije, ki bo čudežno rešila vse obstoječe probleme. Vse tehnologije se bolj ali manj hitro premikajo po tej krivulji. Spletne storitve izstopajo, saj so vrh dosegle v približno letu in pol. To je precej hitreje od XML-a, pa tudi veliko hitreje od Java in .NET.

Spletne storitve danes obljublajo veliko. Revolucijo razvoja programske opreme, nov model svetovnega spleta, rešitev problemov integracije aplikacij, ključ do uspešnega elektronskega poslovanja in še bi lahko naštevali. Z drugimi besedami, spletne storitve so promovirane kot revolucionarna tehnologija, kot nekaj, česar na področju informatike še nismo imeli.

Vsem, ki pa poznamo »tradicionalne« komponentne modele, kot so CORBA (Common

Object Request Broker Architecture), COM+ (Component Object Model), RMI (Remote Method Invocation) in EJB (Enterprise Java Beans), se nam primerjava s spletnimi storitvami vsiljuje sama od sebe. Kar preveč zapeljivo je gledati na spletne storitve kot na le še en komponentni model in tudi obljube o vseh revolucijah in novostih sprejemamo zadržano – in bolj ko se poglobljamo v podrobnosti spletnih storitev, več »deja-vu« prizorov doživljamo.

V isti sapi pa se poraja vprašanje, kakšna usoda čaka tradicionalne komponentne modele. Ali z razširitvijo spletnih storitev ne bodo obstoječi komponentni modeli izgubili pomena in bodo postali nepomembni?

V tem poglavju bom poskušal odgovoriti na zastavljena vprašanja. Ugotovil, da spletne storitve kljub vsemu prinašajo nekaj novosti v razvoj informacijskih sistemov in se odlično dopolnjujejo s tradicionalnimi komponentnimi modeli.

3.1 Kaj so spletne storitve

3.1.1 Način komuniciranja

Enostavna tehnična definicija spletnih storitev se glasi: spletne storitve so komponente, ki komunicirajo preko protokola SOAP (Simple Object Access Protocol). Protokol SOAP za oblikovanje sporočil, ki si jih izmenjajo spletne storitve, uporablja XML. To pomeni, da sporočila oblikuje tekstovno, vsi dosedanji protokoli pa so jih oblikovali binarno. Prednost takšnega pristopa je seveda v tem, da so sporočila v XML obliki veliko lažje čitljiva za človeka. Za računalnik pa predstavljajo dodatno obremenitev, saj zasedejo več prostora, potrebujejo več časa za prenos preko omrežja, pa tudi procesor je bolj obremenjen pri razpoznavanju takšnih sporočil, saj mora delati veliko primerjave nizov.

Podobno kot klasične komponente tudi spletne storitve definirajo vmesnike, v katerih povedo, katere operacije (ki jim tukaj radi pravimo storitve) nudijo. Vmesniki, ki jih definirajo spletne storitve, temeljijo na izmenjavi sporočil. To pomeni, da nas pri razvoju spletne storitve zanima predvsem, kakšna sporočila (seveda po protokolu SOAP) bo spletna

storitev sprejela – temu pravimo zahteve - in kakšna sporočila bo oblikovala v odgovor – tem pravimo odgovori. Tak pristop daje veliko fleksibilnosti pri razvoju.

SOAP sporočila se med odjemalcem in komponento prenašajo preko standardnih Internetnih protokolov. Najpogosteje je v ta namen uporabljen HTTP (Hyper Text Transfer Protocol), lahko pa uporabimo tudi SMTP (Simple Mail Transfer Protocol) – torej prenos po elektronski pošti, ali pa FTP (File Transfer Protocol) ali celo kak lasten protokol.

Najpomembneje je zagotovo dejstvo, da spletne storitve ne predpisujejo komunikacijskega modela. Čeprav je marsikje zapisano, da spletne storitve temeljijo na RPC (Remote Procedure Call) paradigmi, je to le pogojno resnično. RPC način komuniciranja je v osnovi sinhron, kar pomeni, da odjemalec proži neko operacijo in čaka na odgovor, ki naj bo takojšen. Spletne storitve pa niso omejene zgolj na RPC komunikacijo.

Spletne storitve lahko komunicirajo tudi asinhrono, na način, ki smo ga vajeni pri sporočilnih sistemih (MOM – Message Oriented Middleware). To pomeni, da odjemalec ne pričakuje takojšnjega odgovora. Običajno bo spletna storitev obvestila odjemalca o končanju operacije s povratnim klicem takrat, ko bo pač izvršila zahtevano procesiranje.

Tako zastavljen komunikacijski model je ključnega pomena za to, da bodo spletne storitve lahko izpolnile svoje poslanstvo. Spletne storitve so tehnologija za integracijo informacijskih sistemov med seboj. Integracija informacijskih sistemov med seboj se bistveno razlikuje od integracije enega informacijskega sistema – torej znotraj podjetja. Pri integraciji med informacijskimi sistemi ne moremo govoriti o enotnih arhitekturnih zasnovah in o komplementarnosti rešitev in pogledov. Zaradi tega je pri integraciji med informacijskimi sistemi ključnega pomena, da temeljimo na predpostavki o šibki sklopljenosti. Komunikacijski model spletnih storitev je prilagojen ravno slednjemu. Podpora sinhronemu, še pomembneje pa asinhronemu komunikacijskem modelu, preko Internetnih protokolov, omogoča izgradnjo vmesnikov za integracijo, ki so prilagojeni delovanju preko internetnih povezav z omejeno hitrostjo, pasovno širino in zanesljivostjo. Učinkovite spletne storitve bodo temeljile na asinhronem komunikacijskem modelu, saj je ideja spletnih storitev v izmenjavi dokumentov. Več o tem bom povedal v nadaljevanju.

3.1.2 Vmesnik

Mimo definicije vmesnika ne moremo tudi pri spletnih storitvah. V ta namen uporabimo WSDL, ki je kratica za Web Services Description Language, torej jezik za opis spletnih storitev. WSDL prav tako temelji na XML – z drugimi besedami, WSDL opis vmesnika ni nič drugega kakor XML dokument. WSDL specificira naslednje stvari:

- podatkovne tipe,
- sporočila,
- operacije,
- povezave,
- vrata in tipe vrat,
- storitve.

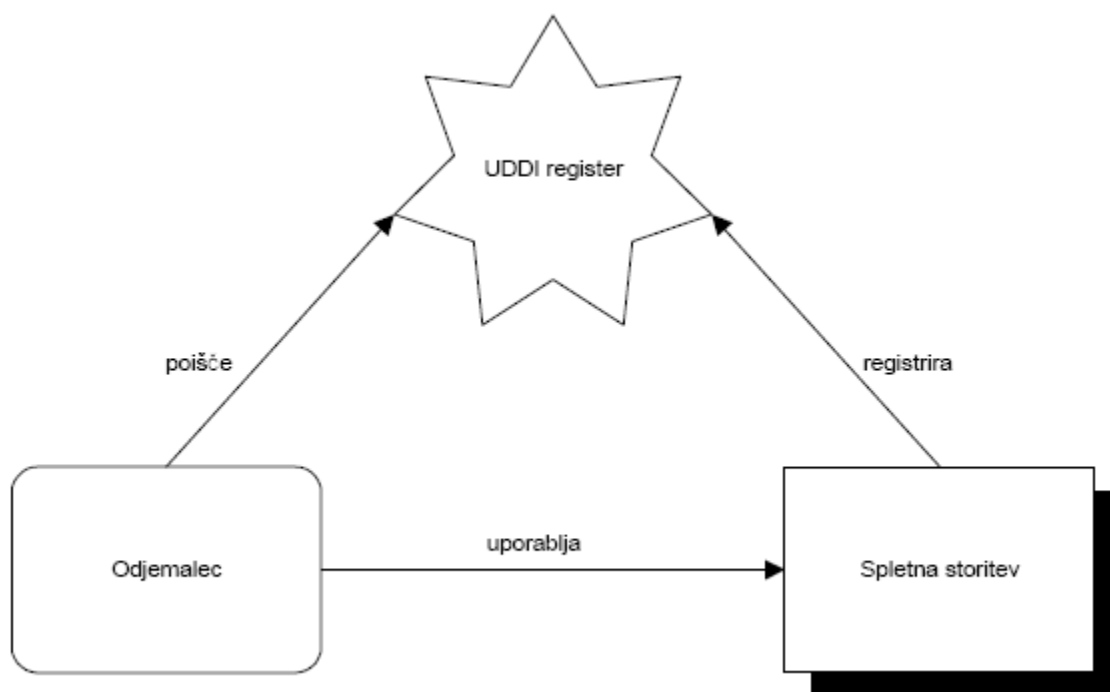
Specifikacija vmesnikov v WSDL se razlikuje od jezikov za definicijo vmesnikov (IDL) in je vsaj na prvi pogled veliko bolj kompleksna. Vendar moramo vedeti, da WSDL ni bil zasnovan, da bi ga pisali »na roko«. WSDL opise vmesnikov bodo za nas oblikovala orodja za razvoj spletnih storitev.

3.1.2 Register spletnih storitev

Tretji ključni gradnik vsakega komponentnega modela je register oz. imenska storitev, v kateri ponudniki registrirajo komponente in omogočijo, da jih odjemalci najdejo in uporabijo. Pri spletnih storitvah to vlogo igra register UDDI (Universal Discovery, Description and Integration). UDDI prav tako temelji na XML, v njem pa hranimo WSDL in ostale pomembne dokumente, ki opisujejo spletno storitev in so pomembne za odjemalca. Odjemalec mora namreč poznati način, kako uporabiti storitve, ki jih določena spletna storitev ponuja. Običajen postopek je identičen tistemu iz klasičnih komponentnih modelov:

1. Ponudnik spletne storitve le-to registrira v UDDI registru.
2. Odjemalec v UDDI registru poišče ustrezno storitev, pridobi WSDL in na oblikuje posrednika (proxy ali stub), ki naredi storitev lokacijsko neodvisno.

3. Odjemalec uporabi storitve izbrane spletne storitve preko protokola SOAP. To je prikazano na sliki.



Slika 4. UDDI register

Seveda odjemalec tudi za povpraševanje po UDDI registru uporablja protokol SOAP. UDDI register omogoča povpraševanje po imenih, za kar pa moramo vnaprej poznati ime spletne storitve. To vedno ni izvedljivo, zato je drugi način povpraševanje po tipu storitve, ki jo iščemo (neke vrste rumene strani). UDDI registri predvidevajo tudi tretjo možnost, ki jo imenujejo zelene strani, povpraševali pa bi po načinu, na katerega bi radi opravljali elektronsko poslovanje. Ta faza še ni realizirana, zato tudi ne moremo povedati več o tem, ali deluje.

V zvezi z UDDI registri je trenutno še nejasen tudi problem fragmentacije, torej način kako bodo registri organizirani v Internetu. Bodo povezani v enoten register, podoben DNS (Domain Name Server) modelu, ali pa bomo morali uporabljati več med seboj ločenih UDDI registrov. Trenutno nekaj podjetij ponuja javno dostopne UDDI registre. Popularna sta predvsem IBM-ov in Microsoftov register, celoten seznam pa najdete na www.uddi.org. Predvidevamo, da se bo s časom število registrov povečevalo, njihova nepovezanost pa bi otežila iskanje ustreznih storitev.

3.2 Paradigma

Če poznate tradicionalne komponentne modele, vas tehnični opis spletnih storitev skoraj zagotovo ni prepričal in verjetno se sprašujete, kaj je pravzaprav tako novega v spletnih storitvah. Ali ne gre le za stvari, ki jih poznamo že nekaj časa, le da so tokrat preoblečene in začinjene z XML.

Takšno razmišljanje pa nas lahko hitro zavede in povzroči uporabo spletnih storitev na način komponentnih modelov, to je z uporabo objektne paradigme oz. objektnega načina razmišljanja pri zasnovi in načrtovanju vmesnikov. Še več, zelo zapeljivo je razmišljati preprosto o uporabi obstoječih komponent (pa naj bodo to CORBA, COM+, ali EJB komponente) in enostavno izvoziti njihove vmesnike kot spletne storitve.

Tak pristop pa ni priporočljiv. Spletne storitve imajo namreč drugačno zgodovino. Ne pozabimo, da temeljijo na XML-u, ki je način zapisa podatkov v dokumente. XML je dokumentno orientiran in za učinkovito uporabo spletnih storitev je potrebno tudi vmesnike spletnih storitev načrtovati dokumentno! Spletne storitve je potrebno razumeti kot način za implementacijo sistemov za izmenjavo dokumentov.

3.2.1 Kakšna je razlika med dokumentnimi in objektnimi vmesniki

Osnova dokumentnih vmesnikov so dokumenti. Vmesniki so le način, kako dokumente izmenjujemo. To se bistveno razlikuje od objektnih vmesnikov, kjer so ključnega pomena operacije (metode), podatkovne strukture pa le uporaben način za izmenjavo podatkov, potrebnih za delovanje operacij. Dokumentno-orientirani vmesniki se od objektnih razlikujejo v naslednjih točkah:

1. Delujejo nad kompleksnejšimi podatkovnimi strukturami. Te so tako kompleksne, da jih imenujemo dokumenti.
2. Bogatejša vsebina dokumentov omogoča, da je sklopljenost med deli sistema veliko manjša, kot pri objektnih vmesnikih.
3. Manjša sklopljenost omogoča uporabo sinhronnega komunikacijskega modela.

Primerjavo lahko naredimo tudi malo manj formalno. Komponentni sistemi, katerih vmesniki so objektni, se obnašajo kot ljudje med pogovorom. Sistemi, ki pa temeljijo na dokumentnih vmesnikih, pa so bolj podobni pošiljanju pisem drug drugemu. Pri pogovoru je sklopljenost med sogovornikoma zelo velika. Recimo, da želimo kupiti nov računalnik. Objektni pristop bi izgledal nekako tako: »Imate računalnik z uro 2.4 GHz?« »Nimamo tega modela.« »Imate kak podoben model?« »Imamo z uro 2.2 GHz.« »Koliko pomnilnika ima?« ... in tako naprej. Bodite pozorni na naslednja dejstva:

- Enostavne podatkovne strukture.
- Tesno sklopljenost – odgovori so brez pomena, če ne poznamo konteksta vprašanja. Vedeti moramo tudi s kom komuniciramo.
- Komunikacija je sinhrona. Zaradi tesne sklopljenosti potrebujemo odgovore takoj, saj jih bomo le tako lahko smiselno povezali z vprašanji. Naslednje vprašanje (zahteva) pogosto temelji na predhodnem odgovoru.

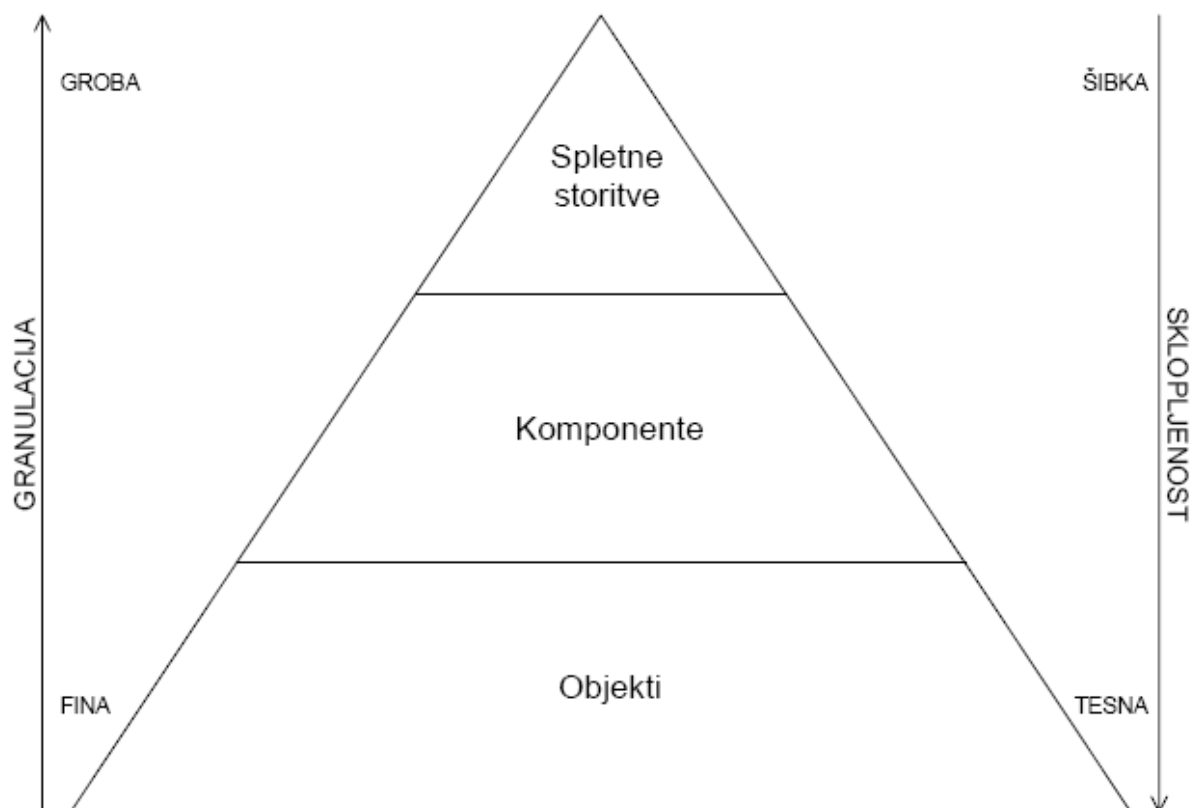
Dokumentni pristop h komuniciranju se zelo razlikuje. Tukaj bi zahtevali katalog vseh računalnikov, ki so na voljo. Na osnovi kataloga bi izbrali takšnega, ki je primeren za nas, oblikovali naročilo in ga poslali dobavitelju. Tukaj bodite pozorni na naslednja dejstva:

- Izmenjujemo kompleksne, samo-opisne podatke – dokumente, kot je na primer katalog in naročilo.
- Šibka sklopljenost – zahteve in odgovori niso direktno povezani. Naročilo ni direktno vezano na prej poslani katalog izdelkov. Prav tako katalog ni direktno vezan na zahtevo po katalogu, saj je samo-opisen. To da je katalog ugotovimo iz besednjaka (sheme) dokumenta in iz zahteve.
- Komunikacija je asinhrona in je pogojena s samo-opisnostjo dokumentov.

3.2.2 Granulacija storitev

Še enkrat – spletne storitve niso komponente v klasičnem smislu. Vmesniki spletnih storitev naj bodo veliko bolj grobo granulirani od vmesnikov komponent (ne pozabimo pa, da so že vmesniki komponent bolj grobo granulirani od vmesnikov objektov). Rezultat preslikave komponent direktno v spletne storitve bo, slaba arhitektura spletnih storitev in ponovile se bodo podobne težave, kot v prvih poskusih uporabe komponent, ko so razvijalci zgolj preslikali vmesnike objektov v komponente. Takšni sistemi ne bodo nudili ustreznih zmogljivosti, krivda pa ne bo v tehnologiji, pač v arhitekturi sistema.

To pa ne pomeni, da spletne storitve in komponente niso povezane. Ravno nasprotno! Komponentne platforme so idealna osnova za razvoj učinkovitih spletnih storitev. Spletne storitve bodo običajno uporabljale eno ali več poslovnih komponent in jim delegirale ustrezne naloge za izpolnitev storitev, ki jih nudijo. Za vse poznavalce je analogija med spletnimi storitvami in poslovnimi komponentami očitno podobna že prej omenjeni analogiji med poslovnimi komponentami in objekti. Iz izkušenj vemo, da lahko kot poslovno komponento izpostavimo katerikoli objekt v Javi, C++, ABAP ali drugem programskem jeziku. Vemo pa tudi, da tak način v praksi ne deluje. Dobro načrtovane komponente morajo združevati pomensko popolnejše storitve in običajno uporabijo več objektov za implementacijo komponente. Hierarhija je očitna in je prikazana na naslednji sliki:



Slika 5 - Hierarhija granulacije storitev

3.3 Kdaj in kje uporabiti spletne storitve

Komponentne platforme, od modela CORBA preko J2EE (Java 2 Enterprise Edition) in EJB do COM+ so zelo uspešne pri razvoju in integraciji aplikacij v podjetju. Dobro načrtovani komponentni sistemi zagotavljajo visoke zmogljivosti in skalabilnost, kot je bilo že bilo prikazano, ko je model CORBA dosegel mejo 75000 objektnih transakcij na sekundo z uporabo največje žive podatkovne baze (s področja telekomunikacij) velikosti 111 terabytov s 10^{14} zapisi. To je pet krat več transakcij, kot jih obdela vseh pet največjih svetovnih telekomunikacijskih podjetij skupaj.

Komponentni sistemi pa za takšno učinkovitost zahtevajo dobro definirano arhitekturo in predvsem jasno postavljene in stabilne vmesnike med komponentami. Zato je pri izdelavi komponentnih arhitektur ključen poudarek na definiciji vmesnikov. Če jih bomo definirali dobro, bodo naše arhitekture dejansko pokazale prednosti komponentnega pristopa. Če pa

vmesnikom v začetku ne bomo namenili dovolj pozornosti, bomo ceno plačevali kasneje, ko jih bomo prisiljeni popravljati.

Komponentni sistemi zahtevajo tudi, da se znotraj informacijskega sistema uskladimo o infrastrukturi in tehnologijah. To pa je običajno problem, kakor hitro zapustimo okvire podjetja in začnemo govoriti o integraciji med podjetji. Med poslovnimi partnerji običajno ni mogoče doseči konsenza glede tehnologij in infrastrukture. Največ, na kar lahko računamo je dogovor o poslovnih procesih in oblikah dokumentov.

Izmenjava dokumentov v elektronski različici pa se ne razlikuje bistveno od tradicionalne izmenjave dokumentov na papirju, s katero se poslovni ljudje srečujejo vsak dan in jo razumejo zelo dobro! Izmenjava dokumentov z uporabo spletnih storitev dejansko reflektira tradicionalno poslovanje med podjetji.

Tukaj se skriva tudi največji pomen spletnih storitev, saj omogočajo izmenjavo dokumentov z uporabo XML in ostalih internetnih standardov, ki smo jih že omenili. Ti standardi so danes tudi praktično edina izbira, ki jo imamo pri prenosljivi predstavitvi dokumentov in povezljivosti.

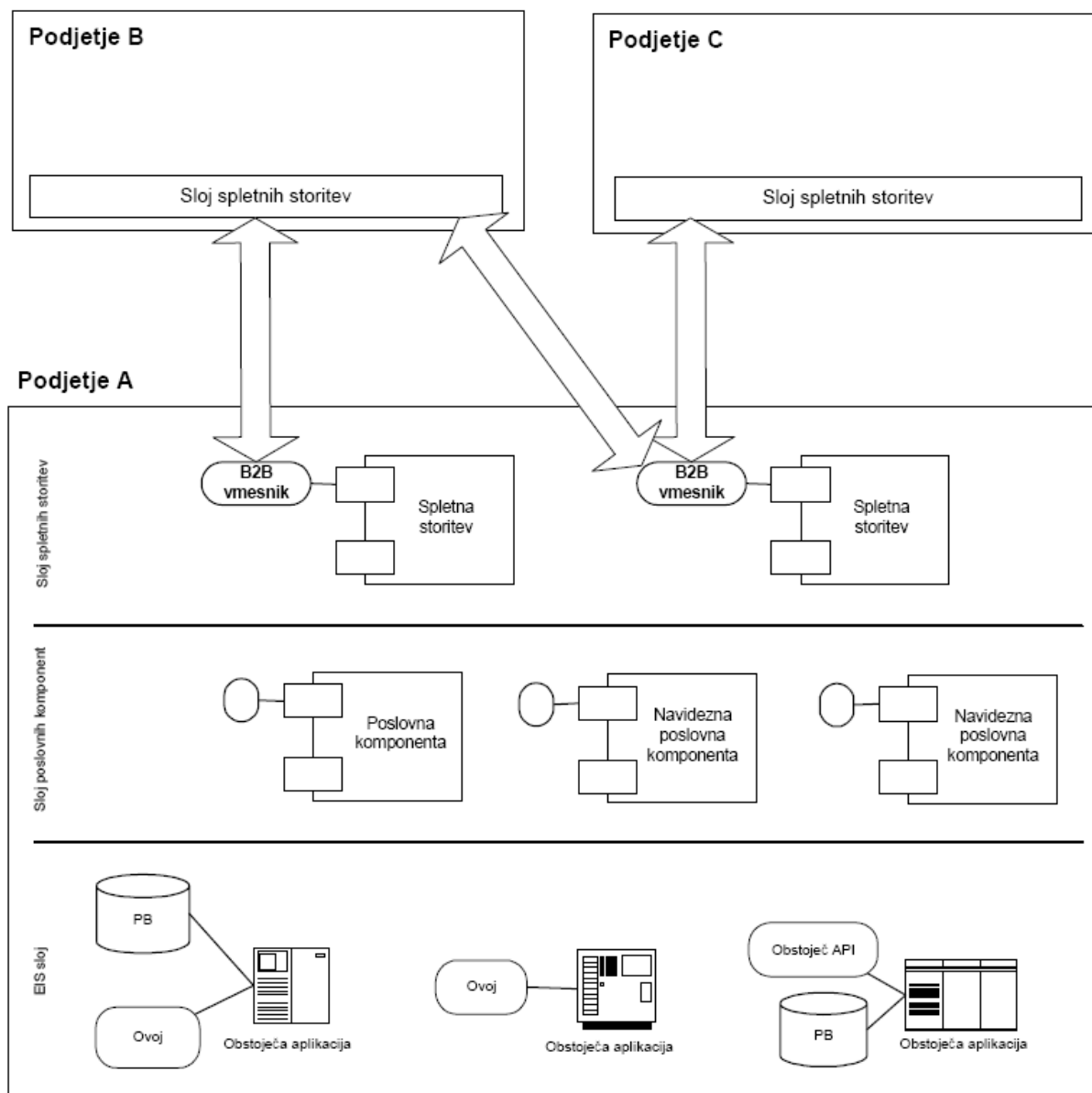
Spletne storitve pa plačujejo ceno enostavnosti in splošnosti pri zmogljivostih. Spletne storitve ne bodo nikdar dosegale zmogljivosti komponentnih modelov. Razlog je enostaven. Predstavitev podatkov v XML, njihov prenos in razpoznavanje zahtevajo dodaten čas.

Spletne storitve trenutno tudi še ne podpirajo vseh sistemskih storitev, ki smo jih vajeni iz komponentnih modelov. Predvsem pomembne so transakcije, varnost in zagotavljanje trajnega stanja, ki v spletnih storitvah še niso podprte. Tudi zaradi tega spletne storitve niso prava izbira za interno arhitekturo informacijskih sistemov, kjer si delovanja brez podpore transakcij in ustrezne varnostne politike ne moremo predstavljati.

3.4 Integracija je ključ do učinkovitih spletnih storitev

Kratek odgovor kdaj in kje uporabiti spletne storitve je torej:

- Komponentne modele uporabimo za izdelavo zapletenih informacijskih sistemov, kjer so pomembne zmogljivosti in skalabilnost, kjer je arhitektura sistemov dobro določena, kjer so vmesniki stabilni, komunikacije zanesljive in kjer si lahko privoščimo tesnejšo sklopljenost komponent – to pa ni nikjer drugje kot v informacijskih sistemih znotraj podjetja.
- Spletne storitve uporabimo tam, kjer želimo povezati neodvisne informacijske sisteme, kjer temeljimo na izmenjavi dokumentov in na deljenih poslovnih procesih, kjer govorimo o šibki sklopljenosti in asinhroni komunikaciji, kjer komunikacije med sistemi niso tako pogoste in potekajo preko Internetnih povezav, kjer ne moremo uskladiti arhitekture informacijskih sistemov, ki jih povezujemo (ali pa bi to zahtevalo preveč napora) – to pa ni nič drugega kakor integracija med podjetji, ali elektronsko poslovanje tipa B2B (Business to Business).



Slika 6 - Integracijska arhitektura

Posledica takšnega razmišljanja pa je tudi dejstvo, da bomo za izdelavo učinkovitih spletnih storitev potrebovali dobro definirano notranjo arhitekturo informacijskega sistema, ali z drugimi besedami, dobro integriran informacijski sistem znotraj podjetja, kar nas bo slej kot prej pripeljalo do potrebe po EAI, Enterprise Application Integration. Le dobro integrirani informacijski sistemi bodo na dolgi rok uspešni pri zagotavljanju učinkovitih spletnih storitev, ki bodo zagotavljale hitre in natančne podatke poslovnim partnerjem.

Pomena dobro zastavljene arhitekture zato ne gre zanemariti. Na sliki 3 je prikazana integracijska arhitektura, ki na osnovi komponent zagotavlja integracijo znotraj podjetja,

spletne storitve pa uporablja za integracijo med podjetji. Pri definiciji arhitektur pa igrajo pomembno vlogo tudi načrtovalski vzorci, ki jih vsekakor velja upoštevati.

Dokumentno orientirane spletne storitve bomo izdelovali nad poslovnimi komponentami na vmesnem sloju. Komponente igrajo vlogo robustnih, skalabilnih storitev z visoko stopnjo razpoložljivosti. Spletne storitve pa so visoko dostopne vstopne točke v sistem. Spletne storitve omogočajo avtomatizacijo deljenih poslovnih procesov med partnerji, katerih interna struktura informacijskih sistemov je zelo različna.

Orodja za razvoj danes omogočajo dva pristopa k gradnji spletnih storitev. Prvi je pretvorba obstoječih komponentnih vmesnikov v spletne storitve. Če bomo uporabili ta pristop, potem bo prava pot ta, da bomo najprej definirali komponente z ustreznimi vmesniki in jih nato pretvorili v spletne storitve. Drugi pristop je razvoj spletnih storitev na novo, kjer bomo morali definirati nove vmesnike, operacije pa bomo običajno delegirali več poslovnim komponentam oz. obstoječim sistemom, ograjenim s komponentami.

3.5 Zagotavljanje varnosti in transakcijske integritete

Kljub temu, da so spletne storitve obetavna tehnologija, pa trenutno ne naslavlja dveh zelo pomembnih področij: varnosti in transakcijske podpore. Spletne storitve odpirajo nove varnostne luknje, ki jih z obstoječimi varnostnimi tehnologijami ne moremo rešiti v celoti. Uporaba SSL (Secure Socket Layer), TLS (Transport Layer Security) in digitalnih certifikatov je učinkovita le pri komuniciranju med dvema partnerjema. Sodobno elektronsko sodelovanje pa zahteva vpletenost več partnerjev. Takrat pa je smiselno razmišljati o tem, da lahko določene dokumente ali njihove dele prebirajo le določeni partnerji, da je potrebno zagotoviti, da delov dokumentov partnerji ne bodo spreminjali (hote ali nehote) in da bo zagotovljena transakcijska integriteta poslovnih operacij.

Varnost je danes zagotovo najbolj vroča tema razvoja spletnih storitev. Trenutno obstaja množica specifikacij, ki rešujejo posamezne vidike varnosti. Najpomembnejše so našteje v nadaljevanju, zavedati pa se je potrebno, da gre za tehnologije v nastajanju:

- XML Digital Signature (xml-dsig), ki definira, kako digitalno podpisati dele ali celoten

- XML dokument. xml-dsig nastaja kot združen projekt W3C (World Wide Web Consortium) in IETF (Internet Engineering Task Force).
- SOAP-SEC: definira varnostno politiko SOAP sporočil in način njihovega digitalnega podpisa. Nastaja v sodelovanju IBM-a in Microsofta pod okriljem W3C.
 - WS-Security: je Microsoftov predlog za definicijo varnosti protokola SOAP in dodaja nekatere nove elemente v SOAP sporočila. WS-Security je precej bolj podrobna specifikacija od SOAP-SEC in naslavlja več od digitalnega podpisovanja SOAP sporočil. Videli pa bomo, ali bo WS-Security nadomestil SOAP-SEC.
 - SAML: Security Assertion Markup Language nastaja pod okriljem OASIS in definira načine za izmenjavo avtentikacijskih in avtorizacijskih podatkov med spletnimi storitvami. SAML omogoča enkratno avtentikacijo uporabnikov (podobno single-signon konceptu pri integraciji znotraj podjetja).
 - XACML: XML Access Control Markup Language, ki prav tako nastaja pod okriljem OASIS, je dopolnilna tehnologija k SAML, ki omogoča, da z uporabo XML določimo pravice dostopa do posameznih storitev.
 - XML Encryption nastaja pod okriljem W3C in naslavlja področje kriptiranja XML dokumentov, ki pa za razliko od klasičnega kriptiranja lahko naslavlja tudi dele dokumentov in vsebuje uporabne metapodatke o načinih prebiranja takih dokumentov.
 - XKMS: XML Key Management Specification pa obljublja poenostavitev uporabe protokolov javnih ključev, ki lahko precej otežijo postavitve varnih spletnih storitev. XKMS je bil predan v obravnavo W3C. V tem kontekstu sta pomembna še X-KISS (XML Key Information Service Specification) in X-KRSS (XML Key Registration Service specification).

Transakcije med podjetji se razlikujejo od klasičnih transakcij znotraj podjetja. Slednje običajno trajajo kratek čas, posebej v primerjavi s poslovnimi transakcijami, ki lahko trajajo od nekaj ur pa tudi do nekaj tednov. Uporaba klasičnih transakcijskih mehanizmov z zaklepanjem in dvofaznim protokolom potrjevanja se v teh razmerah ne obnese najbolje. Spletne storitve tako potrebujejo nov transakcijski model. Trenutno nastajata dva predloga: prvi pod imenom TIP (Transaction Internet Protocol) in drugi pod imenom BTP (Business Transaction Protocol).

3.6 Zaključek poglavja

Na prvi pogled se morda zdi, da so spletne storitve in komponentni modeli le različice porazdeljenega računalništva, ki se razlikujejo v protokolih in ostalih tehničnih podrobnostih. V naslednjem poglavju pa smo ugotovili, da temeljijo spletne storitve in komponente na povsem različnih paradigmah. Spletne storitve so namenjene dokumentno-orientirani komunikaciji, komponente pa komunikaciji na osnovi operacij. Zato so spletne storitve najprimernejše za implementacijo deljenih poslovnih procesov med podjetji in predstavljajo fasado do specifične informacijskih sistemov posameznih podjetij. Komponente pa so predvsem primerne za izgradnjo in integracijo informacijskih sistemov znotraj podjetja. Obe tehnologiji delujeta z roko v roki: komponente so osnovna arhitektura informacijskih sistemov podjetij, spletne storitev pa način integracije med podjetji.

Spletne storitve poenostavljajo elektronsko poslovanje, saj nudijo enotno tehnološko osnovo. Kljub vsemu ostajajo pomembne razpoke v tehnologiji spletnih storitev, nanašajo pa se predvsem na varnost in podporo transakcijam. Rešitve so v pripravi in šele po njihovi vpeljavi bodo spletne storitve resnično pripravljene za elektronsko poslovanje. Ne smemo pa pozabiti, da spletne storitve ne povedo ničesar o vsebini elektronskega sodelovanja. Tukaj pa vstopijo v igro tehnologije, kot so ebXML (Electronic Business XML), WSFL (Web Services Flow Language), BPML (Business Process Markup Language), XLang in ostale.

4. PRIMER UPORABE SPLETNE STORITVE

4.1 Analiza primera

4.1.1 Opis problema

V podjetju, ki se ukvarja s proizvodnjo gradbenih elementov, so želeli posodobiti in izboljšati del logističnega procesa. Glede na veliko število različnih izdelkov z različnimi lastnostmi in kompleksno logistiko dostave teh izdelkov do svojih naročnikov so ugotovili pomanjkanje programske podpore. Že pred implementacijo informacijskega sistema SAP so imeli težave na tem segmentu. Toda med implementacijo informacijskega istem niso posvečali dovolj pozornosti temu problemu. Poleg tega se niso zavedali kompleksnosti in možnost napak zaradi človeškega faktorja.

Podjetje je analiziralo proces logistike in so ugotovili da je velik del težav povezan s planiranjem in naročanjem transporta.. Težave so se vrstile zaradi raznolikih izdelkov produkta s katerimi se ukvarjajo, jih proizvajajo, to so: profili, paneli, plošče, itd. ter so raznoraznih dimenzij, tež, volumna itn. Seveda je pa težava v tem tako, kot vsepovsod »človek«, človeške napake so tiste, ki so pripeljale podjetje do težav s prevozom blaga. Prevoz blaga je potrebno dobro organizirati – potrebno je poskrbeti, da transportno vozilo ustreza za sam prevoz izdelkov. Organizacije so se lotili z uporabo nepreglednih informacijskih orodij in z uporabo stare a neobhodne tehnologije, papirologije. In tu so se začeli problemi. Ni bilo urejenosti, ažurnosti in tudi zgodovine ne. Dejanski problemi so se začeli, ko so naročali tovornjake nepravilnih dimenzij ali je pa sam prevoznik poslal napačen tovornjak Dogajalo se je tudi, da prevoznik ni uspel izpeljati naročila. Tako se je proces naročanja transporta podaljšal zaradi iskanja novega ustreznega prevoznika. In po tem takem je bila tudi stranka nezadovoljna. S finančnega vidika takšno početje ni bilo profitno, saj je zaradi zamujanja dostave naročenega blaga kupec dobavitelju zaračunal penale. Prav tako je zaradi nepregledne statistike podjetje izgubilo možnost uveljavljanja določenih popustov pri prevozniku. Po vsem tem kaosu, se je podjetje odločilo poiskati neki zunanji sistem, ki jim bo olajšal delo z organizacijo, iskanjem prevoznikov.

Odločili so se, da potrebujejo novo programsko rešitev specializirano za planiranje in organizacijo transporta. Tako so prišli do ponudbe nekega podjetja, ki se ukvarja z borzo prevoznitva. V ponudbi so dobili informacijsko podporo za upravljanje s prevozi.

Obvladovanje transportne logistike, ki postaja za podjetja vse bolj vitalnega pomena. Z urejeno logistiko so podjetju zmanjšali stroške, zagotovili nemotenost proizvodnje in točnost dobav svojim kupcem. Pri povečanju obsega poslovanja so problem obvladljivosti rešili z ustrežno programsko rešitvijo, ki je namenjena podjetjem, ki:

- nimajo urejene informatike na področju prevozniške logistike
- potrebujejo sorazmerno veliko število prevozov (več dnevno), ki jih ne pokrivajo z lastnim voznim parkom;
- delujejo globalno
- želijo izboljšati pregled nad izvedenimi prevozi tako po kvaliteti kot obseg (izkoriščenost vozil, točnost dostav, cene na enoto, ...)

Prednosti ponujenega:

- boljša preglednost zunanje logistike po obsegu (število prevoženih kilometrov, količini prepeljanega blaga, destinacijah, ...) ter kvaliteti izvedenih storitev (hitrosti, točnosti, reklamacijah, ...);
- zmanjšanje stroškov naročenih storitev zaradi: urejenega poslovanja, direktne konkurence in obvladljivosti ponudnikov, pravočasne dostave
- izboljšana pogajalska pozicija na osnovi statističnih podatkov o opravljenih prevozi in s tem povečan vpliv nad prevozniki
- možnost spremljanja statistike izvedenih storitev po različnih nivojih organizacije;
- zmanjšanje obremenitev zaposlenih, zmanjšanje stroškov in dvig kvalitete koordinacijskega dela;
- nepristranskost naročanja storitev pri prevoznikih oz. podprevoznikih;
- postopen prehod k avtomatizaciji poslovanja tudi na področju logistične koordinacije in optimizacije.
- integriranosti v obstoječi informacijski sistem
- spletna aplikacija; ni potrebe po dodatnih instalacijah na posameznih delovnih mestih

Čeprav je zunanji sistem prinesel veliko novosti in olajšal delo, je obstajal še vedno en problem. Ta problem je bila povezava med dvema sistemoma to je, informacijskim sistemom SAP in informacijsko podporo za upravljanje s prevozi. Povezavo med njima je bilo potrebno sprogramirati oziroma razviti vmesnik za komunikacijo med njima.

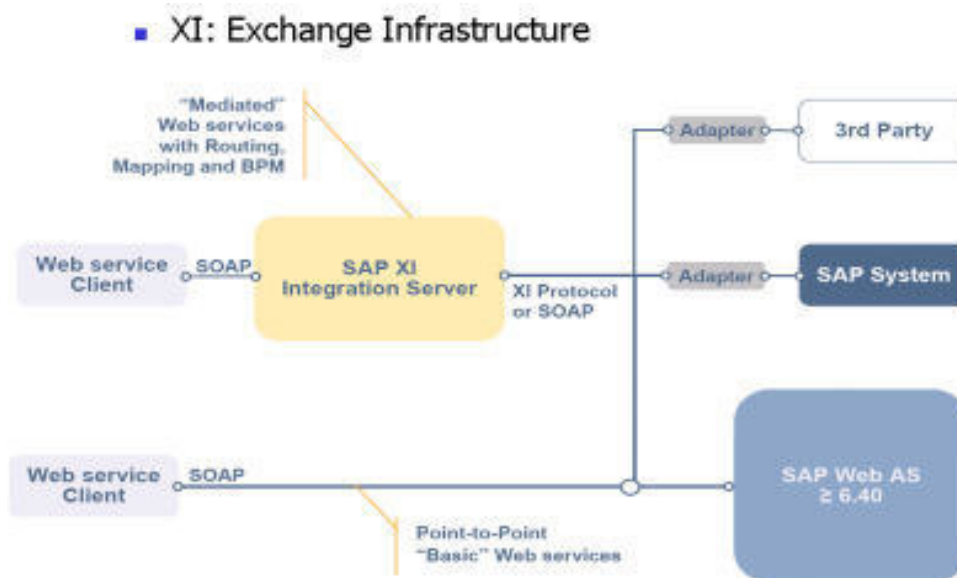
4.2 Možne rešitve - izbira

Kot sem zgoraj omenil, da je obstajal problem povezave dveh informacijskih sistemov in sam načina povezave, saj sta sistema popolnoma drugačna. Seveda, da bi bila najlažja rešitev ta, da bi dvojno vodil dva sistema a kaj ko pa morata biti povezana zaradi poslovnega procesa in sledljivosti dokumentov. Tako je prišlo do razvoja vmesnika. A tudi to ni kar tako, saj smo se morali odločiti kakšne vrste povezavo bomo naredili.

Tu bom opisal možne načine, standarde in rešitve povezave dveh različnih sistemov.

Eden od načinov je vsekakor najnovejša SAP-ova rešitev XI (Exchange Infrastructure), ki ponuja zelo veliko in je zelo na visokem nivoju upravljanja, a kaj ko ni vsaka rešitev primerna za vse. XI je bolj namenjen za podjetja, ki imajo zelo veliko povezav med različnimi sistemi ter imajo zelo veliko prenosov. Imajo pa eno zelo slabo lastnost, vsaj po mojem mnenju: XI je SAP-ova komponenta, ki jo je potrebno posebej plačati, to so dodatni stroški za podjetje.

Primer XI (Exchange Infrastructure) vidimo na sliki.

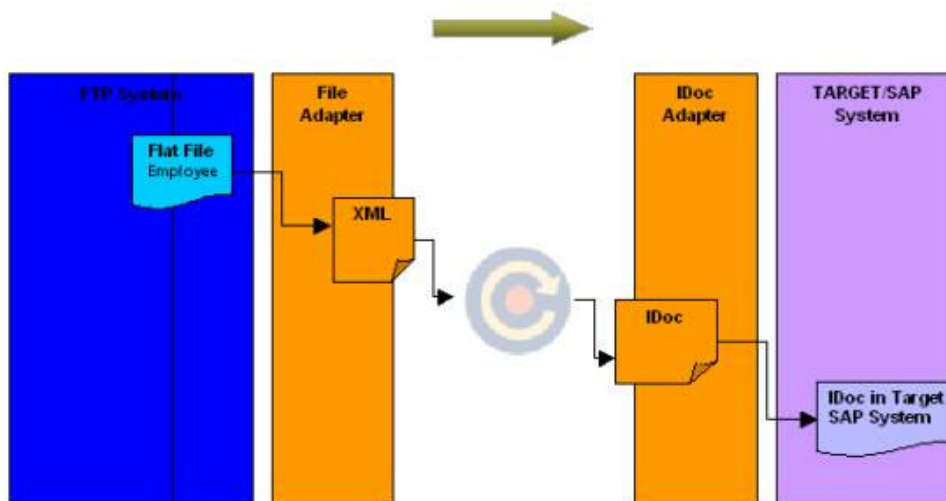


Slika 7 - Web services z SAP XI

Druga rešitev je bila spet SAP-ova komponenta ali adapter IDoc. IDoc je neke vrste povezava prednastavljenih vmesnikov, ki so že v naprej definirani. To je tudi ena od

prednosti, saj je adapter preddefiniran v SAP-u in ga ni potrebno več programirati. Za drugo stran to predstavlja slabost, ker je potrebo precej programerskega dela, da se uskladi s SAP-om. Slabost je tudi ta, da IDoc-i delujejo asinhrono in zahteve se kopičijo v vrsto tako, da pošiljatelj ne ve kaj se je zgodilo z IDoc-om, ker ni prejel potrdila o uspešnem prejemu in izvršitvi. Tako, da smo tudi idejo o IDoc-ih opustili.

Primer IDoc-a vidimo na sliki.



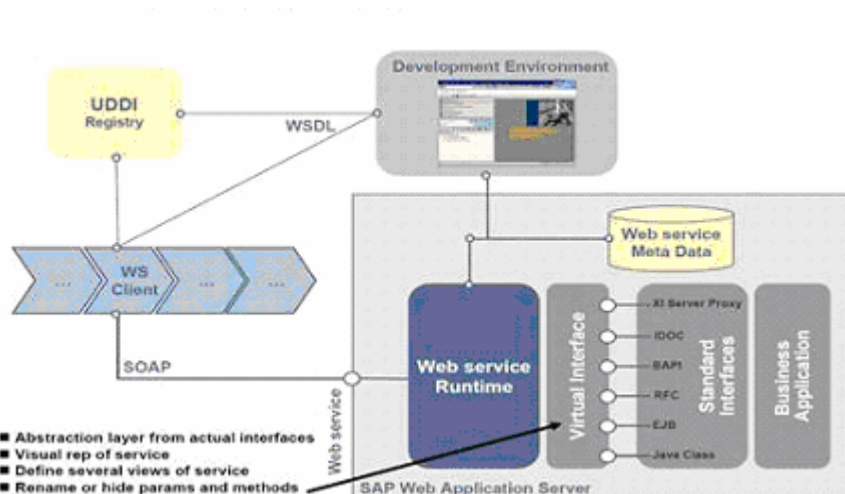
Slika 8 - Primer sprejema IDoc-a

Spraševali smo se tudi o izmenjavi datotek in ugotovili, da je ena povezava že tako in tako vezana na dokumente (izmenjava XML datotek) in celo s to tehnologijo povezave smo se že srečali tudi v drugih podjetjih, potemtakem smo se odločili za spletne storitve (WS – Web Services), zato sem v prejšnjem poglavju naredil bolj detajlni opis spletnih storitev, kaj so, kje se uporabljajo itn.

Pri izbiri spletnih rešitev smo se strinjali, da prednosti prevladujejo nad slabostmi. Tako mi kot zunanji sistem smo to tehnologijo že uporabljali in je preizkušena na obeh programskih rešitvah. Ocenjena velikost prenesenih datotek bo zmerna in ne bo vplivala na performaco. Glede na to, da naročnik želi takoj potrditev o uspešnem prenosu podatkov, je bila izbira spletnih storitev prava.

Glede izbire smo se strinjali tako mi, razvijalci drugega sistema kot podjetje za katerega smo izdelali rešitev. Izbrali smo spletne storitve WS (Web Services).

Primer Web Service vidimo na sliki 9.

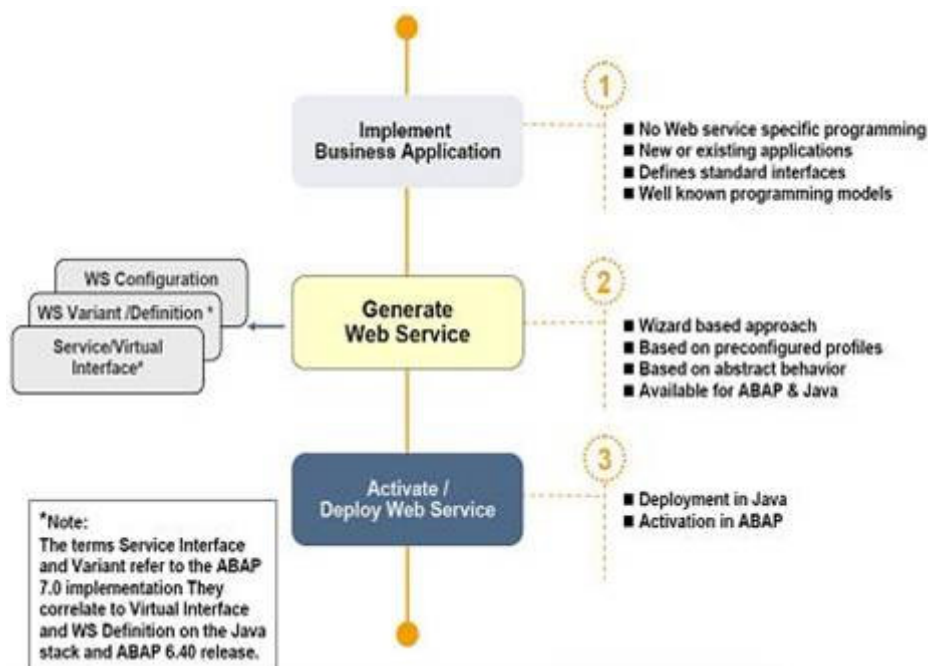


Slika 9 - Web Service v SAP-u

4.3 Izvedba

4.3.1 Priprava grobega osnutka

Želja naročnika-podjetja je bila iskanje prevoznikov za naročila svojih strank. Ideja je bila, da bi se na podlagi prodajnih naročil – izhodnih dobav našel prevoznik, ki bi dostavil naročilo stranki. Odločili so se za iskalnik prevoznikov, za katerega so dobili ponudbo zunanjega izvajalca, ki jim je ponudil borzo prevoznitva. Sistem naj bi bil postavljen v internem omrežju podjetja, zaradi izločanja morebitnih težav medmrežja. A vseeno je bil potreben vmesnik med informacijskim sistemom SAP in zunanjim sistemom (borzo prevoznitva). Naloga vmesnika bi bila pošiljanje dobavnic ter sprejemanje pošiljk. Se pravi, iz informacijskega sistema SAP bi se pošiljale izhodne dobave za katere bi zunanji sistem našel prevoz ter na podlagi tega kreiral pošiljko v SAP-eju. Grobi osnutek priprave spletne storitve prikazuje spodnja slika 10.



Slika 10 - Primer grobe kreacije spletne storitve v SAP-u

4.3.2 Plan priprave

Pri planu priprave, je bil potreben dogovor z naročnikom in tudi dogovor s podjetjem zunanjega sistema. Po dogovoru je bila potrebna določitev število dni v kolikšnem možnem času je možna realizacija zelenega vmesnika ter do kdaj si želijo oziroma do kdaj je možno izdelati vmesnik za preklon v živo, zagon vmesnika na produkcijskem sistemu.

4.3.2 Dogovor o tehniki prenosa podatkov

Dogovor je potekal tudi o tehnologiji prenosa podatkov, saj obstaja več možnih tehnik, kot so npr.:

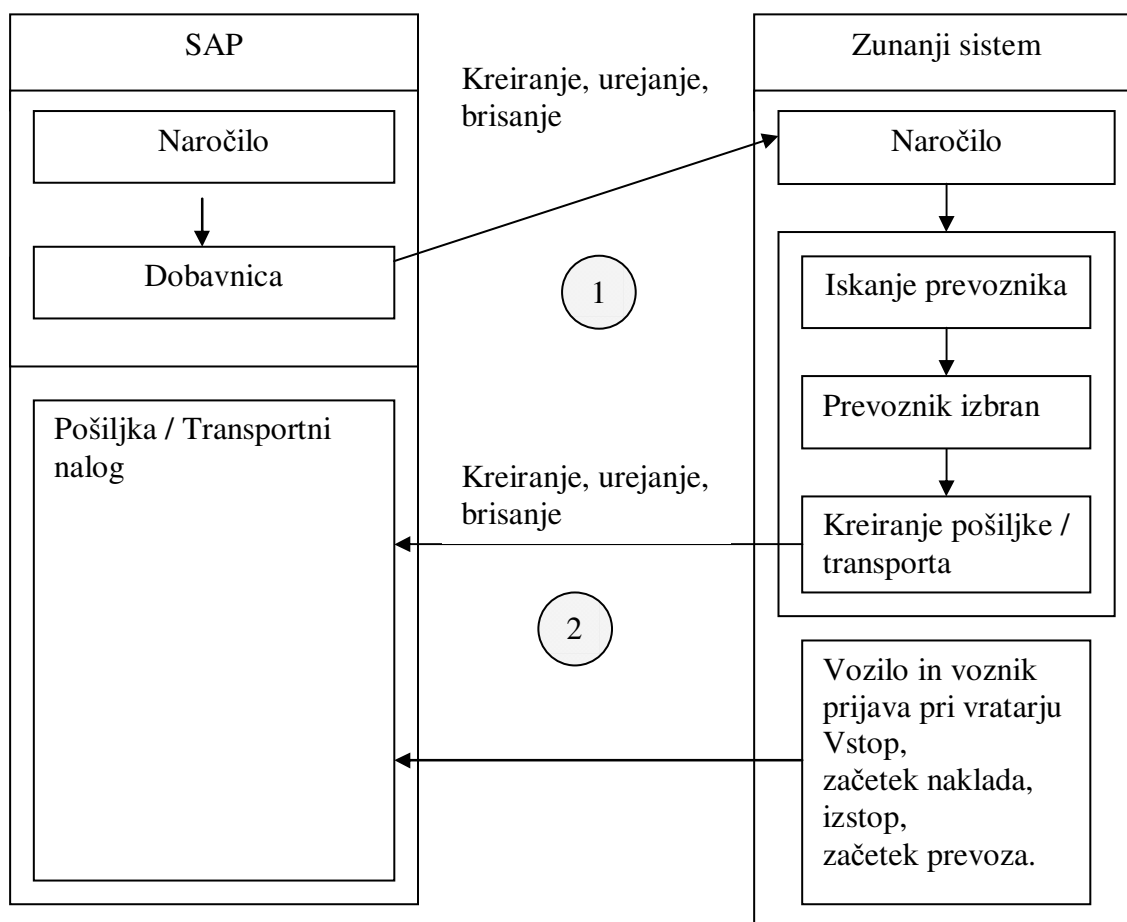
- XI
- IDoc
- Izmenjava datotek
- RFC
- Spletni servisi (Web Service)

Pri dogovoru smo se odločili za tehniko prenosa podatkov preko spletnih storitev (Web Service), kajti zunanji sistem določenih tehnik ne podpira, določene tehnike so prepočasne tako, da najmanj slabosti smo našli pri tehniki prenosa podatkov spletnih storitev.

4.3.3 Dogovor o podatkovnih skupinah in smeri prenosa

Pri dogovoru o podatkovnih skupinah ter smeri prenosa, se je bilo potrebno dogovoriti o vrstnem redu prenosa podatkov. Naloga zunanjega sistema je bila pripraviti spletno storitev (objava WSDL-ja) za sprejem dobavnice, katero je klical naš sistem pri pošiljanju dobavnic. Naša objava spletne storitve (objava WSDL-ja), je bila potrebna pri kreaciji pošiljke, katero je klical zunanji sistem za kreacijo pošiljke. Spodnja slika, slika 11 prikazuje potek izmenjave podatkov.

Točke povezav, število vmesnikov:



Slika 11 - Potek povezave vmesnikov za izmenjavo podatkov.

Točka 1: Informacijski sistem SAP kliče spletno storitev zunanjega sistema pri pošiljanju dobavnice.

Točka 2: Zunanji sistem kliče spletno storitev informacijskega sistema SAP pri kreaciji pošiljke.

4.3.4 Povezava SAP – zunanji sistem

Prva povezava predstavlja pošiljanje SAP-ejeve dobavnice v zunanji sistem. V SAP-eju se na osnovi prodajnega naročila kreira izhodna dobava (dobavnica). Dobavnica je v SAP-eju osnovni dokument za celoten proces odpreme blaga. Podatki na dobavnici, ki so potrebni za »odpremo«, so vsaj: prejemnik blaga, točka odpreme, blago, količina, enota mere, itn. Ti podatki so tudi osnova za pošiljanje dobavnice preko spletne storitve. Ob tem SAP kliče spletno storitev zunanjega sistema, to je objavljeni WSDL (XML), preko katerega se posredujejo podatki. Na podlagi podatkovnih struktur, ki v SAP-eju predstavljajo podatke dobavnice, smo se dogovorili za strukturo podatkov, ki se bodo izmenjevali. Struktura predstavlja polja s podatkovnimi tip ter medsebojna razmerja polj. Tako smo definirali večnivojsko strukturo za prenos dobavnice.

4.3.5 Opis uporabe vmesnika (klic storitve in pošiljanje)

V tem poglavju opisujem, kako poteka klic spletne storitve (Web Service-a) ter kako smo se lotili pisanja programa za pošiljanje dobavnic.

Preden smo začeli z delom smo morali definirati natančen opis podatkovnih struktur vmesnika. Primer strukture dobavnice »send order« vidimo na sliki 17.

Method	SEND_ORDER	
Importing	INPUT	ZSEND_ORDER_SOAP_IN
ORDER		Z_ORDER
ORDER		STRING
LOADING_TO		STRING
LOADING_FROM		STRING
UNLOADING_TO		STRING
UNLOADING_FROM		STRING
STRICT_LOADING_DATE		INT4
STRICT_UNLOADING_DATE		INT4
CREATION_DATE		STRING
Exporting	OUTPUT	ZSEND_ORDER_SOAP_OUT
SEND_ORDER_RESULT		STRING

Slika 12 - Struktura dobavnice »send order«

Pri podatkovni strukturi smo morali biti zelo pazljivi na podatkovne tipe polj, saj se nekateri podatkovni tipi razlikujejo med seboj (na primer: podatkovni tip »boolean« se v razvojnem okolju .NET razlikuje od SAP-ejevega razvojnega okolja) tako, da smo dosti časa porabili pri ugotavljanju ustreznosti podatkovnih tipov. Rešitev za večino primerov je bil podatkovni tip »string«, saj nam le ta ni povzročal težav.

Ko smo imeli definirano podatkovno strukturo, so razvijalci zunanega sistema objavili spletno storitev, od katere so posredovali URL na katerem se nahaja opis XML-ja (WSDL). Struktura dobavnice »send order« v WSDL obliki prikazuje slika 13.

```

- <s:element name="SendOrder">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="Order" type="tns:Order" />
</s:sequence>
</s:complexType>
</s:element>
- <s:complexType name="Order">
- <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="Order" type="s:string" />
  <s:element minOccurs="0" maxOccurs="1" name="LoadingTo" type="s:string" />
  <s:element minOccurs="0" maxOccurs="1" name="LoadingFrom" type="s:string" />
  <s:element minOccurs="0" maxOccurs="1" name="UnloadingTo" type="s:string" />
  <s:element minOccurs="0" maxOccurs="1" name="UnloadingFrom" type="s:string" />
  <s:element minOccurs="1" maxOccurs="1" name="StrictLoadingDate" type="s:int" />
  <s:element minOccurs="1" maxOccurs="1" name="StrictUnloadingDate" type="s:int" />
  <s:element minOccurs="0" maxOccurs="1" name="CreationDate" type="s:string" />
</s:sequence>
</s:complexType>

```

Slika 13 - Struktura dobavnice »send order« v WSDL obliki

Po objavljenem WSDL-ju, je bila moja naloga kreirati »client proxy« v katerega je potrebno vpisati destinacijo objavljenega WSDL-ja, ki ga SOAP predela v XML strukturo.

V nadaljevanju sem napisal program (aplikacijo) v katerem se napolnijo potrebni podatki podatkovne strukture podanih dobavnic, nato se izvede priprava objekta za klic servisa ter sam klic servisa za pošiljanje preko spletnega servisa (priprava objekta in klic servisa vidimo na sliki 19). Ker pa vsak sistem ima kakšne slabosti in tudi spletni servisi jih imajo (nedosegljiv spletni servis oz. napaka pri vzpostavitvi povezave), sem naredil pomožni zasilni program, ki bo v primeru, da spletni servis ni dosegljiv, avtomatično na določenem intervalu ponovno vzpostavila povezava s servisom ter poslala ne poslano dobavnice. Primer uporabe: Uporabnik kreira izhodno dobavo (dobavnico) na kateri so vsi potrebni podatki, nato jo poda v aplikacijo za pošiljanje in jo pošlje (slika 14 prikazuje primer aplikacije za pošiljanje dobavnic). Uporabniku ni potrebno skrbeti ali je dobavnica poslana ali ne, saj zato skrbi zasilni program, ki to počne samodejno.

```

* priprava objekta za klic servisa
  m_logical_port = 'ZCO_WS_SOAP'.
  TRY.
*   priprava objekta
    CREATE OBJECT m_servis
      EXPORTING
        logical_port_name = m_logical_port.

    CATCH cx_ai_system_fault INTO m_sys_exc.

    CALL METHOD m_sys_exc->get_text
      RECEIVING
        result = m_result.
    PERFORM log_err TABLES it_spor USING wa_act m_napaka m_result
      'ws_soap/cx_ai_system_fault'.
  ENDTRY.

* klic servisa za pošiljanje dobavnic
  TRY.
    CALL METHOD m_servis->send_order
      EXPORTING
        input = m_input
      IMPORTING
        output = m_output.

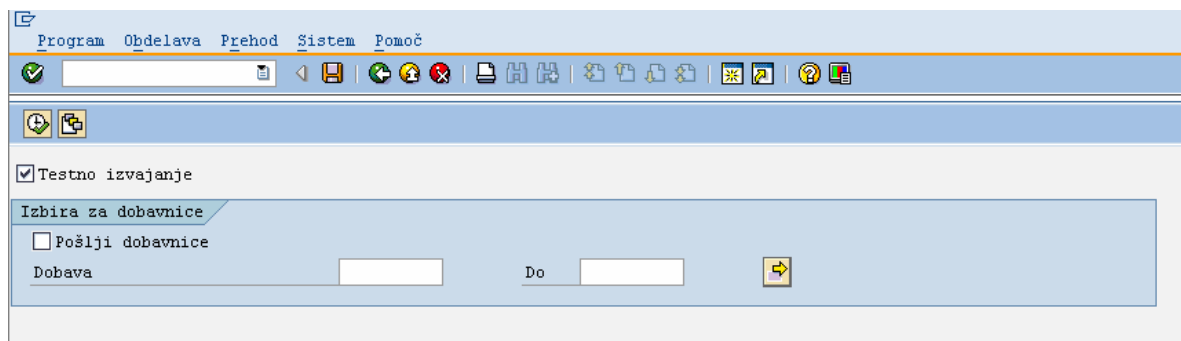
    CATCH cx_ai_system_fault INTO m_sys_exc.
    CALL METHOD m_sys_exc->get_text
      RECEIVING
        result = m_result.
    PERFORM log_err TABLES it_spor USING wa_act m_napaka m_result
      'ws_soap->send_order/cx_ai_system_fault'.

    CATCH cx_ai_application_fault INTO m_apl_exc.
    CALL METHOD m_apl_exc->get_text
      RECEIVING
        result = m_result.
    PERFORM log_err TABLES it_spor USING wa_act m_napaka m_result
      'ws_soap->send_order/cx_ai_application_fault'.

  ENDTRY.

```

Slika 15 - Priprava objekta in klic servisa za pošiljanje dobavnic.



Slika 14 - Primer aplikacije za pošiljanje dobavnic.

4.3.6 Povezava med zunanjim sistemom in SAP

Na podlagi poslanih dobavnic v borzo prevoznitva, se prične iskanje prevoza za le te. Po uspešno najdenem prevozniku-prevozu, zunanji sistem vzpostavi povezavo-klic spletnega servisa s SAP-ejem. Po objavljenem WSDL-ju za kreacijo pošiljke/transportnega naloga, se sproži akcija na SAP-ejevi strani. Po kreiranju prevoza je možno tudi urejanje same pošiljke, kot na primer vozilo je prispelo pred podjetje, vpis voznika, je vstopil, pričetek nakladanja, je izstopil in pričetek transporta.

Ko se prevoznik prijavi, dobimo dejanske podatke (voznika, vozila, registrska številka, gabarite, itn.). Ko vstopi, SAP dobiti status »je vstopil« (podatki na pošiljki / transportnem nalogu se osvežijo). Ko viličarist prične s komisioniranjem, se status pošiljke SAP-eja spremeni v začetek nakladanja. Ko je nalaganje končano (potrditev zadnje postavke relevantne pošiljke), se status pošiljke v SAP-eju zopet spremeni v konec nalaganja. V tem trenutku je obveščen logist, ki prične s pripravo dokumentov. Voznik pri vratarju prevzame zaključene dokumente ter zapusti podjetje. Ko se pošiljka v SAP-eju postavi na status začetek transporta, se v ozadju samodejno knjiži izdaja blaga is skladišča.

4.3.7 Opis uporabe vmesnika (sprejem klica in kreacija pošiljke – transportnega naloga).

Kako sem se lotil kreiranja spletne storitve (Web Service-a), objave WSDL-ja.

Vse aplikacije v SAP-eju se prožijo s transakcijami. Tako, da uporabnik, kadar kreira pošiljko to naredi ročno s transakcijo za kreiranje pošiljk. Z uporabo tehnologije spletnih storitev, sem naredil vmesnik za kreacijo pošiljk, ki bo namesto fizičnih uporabnikov, obstajal sistemski uporabnik »zunanji sistem«, ki bo samodejno kreiral pošiljko. Transakcija za kreiranje pošiljk vsebuje snop funkcij, ki opravljajo svoje delo pri kreiranju pošiljk. Moja naloga je bila narediti program ali funkcijo RFC (remote function control) s katero se bo polnila natančno določena podatkovna struktura, ki je prav tako, kot iz prejšnjega primera dogovorjena z razvijalci zunanjega sistema. Slika 16 prikazuje strukturo za kreacijo pošiljke.

Component	RTy	Component type	Data Type	Length	Decim	Short Text
SHIPMENT_FKEY	<input type="checkbox"/>	CHAR10	CHAR	10	0	Dolžina polja znaka = 10
SHIPMENT_TYPE	<input type="checkbox"/>	CHAR10	CHAR	10	0	Dolžina polja znaka = 10
CARRIER_FKEY	<input type="checkbox"/>	CHAR10	CHAR	10	0	Dolžina polja znaka = 10
CARRIER_SUBFKEY	<input type="checkbox"/>	CHAR10	CHAR	10	0	Dolžina polja znaka = 10
STATUS_TENDERING	<input type="checkbox"/>	CHAR2	CHAR	2	0	Komponenta številke verzije
SHIPPING_CONDITION	<input type="checkbox"/>	CHAR2	CHAR	2	0	Komponenta številke verzije
PRICE	<input type="checkbox"/>	CHAR10	CHAR	10	0	Dolžina polja znaka = 10
CURRENCY_FKEY	<input type="checkbox"/>	CHAR3	CHAR	3	0	Polje z dolžino 3 bajtov
TRANSPORT_START_DATE	<input type="checkbox"/>	CHAR20	CHAR	20	0	Znak20
TRANSPORT_END_DATE	<input type="checkbox"/>	CHAR20	CHAR	20	0	Znak20
VEHICLE_REG_PLATE	<input type="checkbox"/>	CHAR10	CHAR	10	0	Dolžina polja znaka = 10
TRAILER_REG_PLATE	<input type="checkbox"/>	CHAR10	CHAR	10	0	Dolžina polja znaka = 10
DRIVER	<input type="checkbox"/>	CHAR20	CHAR	20	0	Znak20
DRIVER_GSM	<input type="checkbox"/>	CHAR10	CHAR	10	0	Dolžina polja znaka = 10
ACTUAL_ARRIVAL_DATE	<input type="checkbox"/>	CHAR20	CHAR	20	0	Znak20
ACTUAL_ENTRY_DATE	<input type="checkbox"/>	CHAR20	CHAR	20	0	Znak20
ACTUAL_LOAD_START_DATE	<input type="checkbox"/>	CHAR20	CHAR	20	0	Znak20
ACTUAL_LOAD_END_DATE	<input type="checkbox"/>	CHAR20	CHAR	20	0	Znak20
ACTUAL_DEPARTURE_DATE	<input type="checkbox"/>	CHAR20	CHAR	20	0	Znak20
CANCEL	<input type="checkbox"/>	INT4	INT4	10	0	Naravno število
CANCEL_REASON	<input type="checkbox"/>	CHAR50	CHAR	50	0	Komentar
NAROCILA	<input type="checkbox"/>	NAROCILA		0	0	Podatki naročila

Slika 16 - Struktura pošiljke »data_posiljke«

Tudi tu smo morali biti pazljivi pri podatkovnih tipih polj, kot v prejšnjem primeru. Ko je bila struktura narejena, je bilo potrebno narediti spletni servis in objava WSDL-ja. Dostop do WSDL-ja sem razvijalcem zunanjega sistema posredoval URL. Spletni servis je dejansko funkcija, ki neprestano čaka na klic zunanjega sistema. Ko zunanji sistem pokliče spletni servis za kreacijo pošiljke in s tem pošlje podatke, mora biti poslana struktura identična zahtevani strukturi, saj v nasprotnem primeru pride do napake, ki jo javi že sam SOAP zunanjega sistema. Primer XML strukture vidimo na sliki 22. V primeru, da so vhodni parametri veljavni, se izvajanje funkcije nadaljuje s kreiranjem pošiljke. Uspešno kreirano pošiljko vidimo na sliki 23.

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <m:POSILJKA_IN xmlns:m="urn:sap-com:document:sap:rfc:functions">
      <POSILJKA>
        <SHIPMENT_FKEY>0001000000</SHIPMENT_FKEY>
        <SHIPMENT_TYPE>0001</SHIPMENT_TYPE>
        <CARRIER_FKEY>0009385189</CARRIER_FKEY>
        <CARRIER_SUBFKEY></CARRIER_SUBFKEY>
        <STATUS_TENDERING>AC</STATUS_TENDERING>
        <SHIPPING_CONDITION></SHIPPING_CONDITION>
        <PRICE></PRICE>
        <CURRENCY_FKEY></CURRENCY_FKEY>
        <TRANSPORT_START_DATE>25.05.2008 23:14</TRANSPORT_START_DATE>
        <TRANSPORT_END_DATE></TRANSPORT_END_DATE>
        <VEHICLE_REG_PLATE>LJ</VEHICLE_REG_PLATE>
        <TRAILER_REG_PLATE>LJ1</TRAILER_REG_PLATE>
        <DRIVER></DRIVER>
        <DRIVER_GSM></DRIVER_GSM>
        <ACTUAL_ARRIVAL_DATE></ACTUAL_ARRIVAL_DATE>
        <ACTUAL_ENTRY_DATE></ACTUAL_ENTRY_DATE>
        <ACTUAL_LOAD_START_DATE></ACTUAL_LOAD_START_DATE>
        <ACTUAL_LOAD_END_DATE></ACTUAL_LOAD_END_DATE>
        <ACTUAL_DEPARTURE_DATE></ACTUAL_DEPARTURE_DATE>
        <CANCEL>0</CANCEL>
        <CANCEL_REASON></CANCEL_REASON>
        <NAROCILA>
          <item>
            <ORDER_FKEY></ORDER_FKEY>
            <ORDER_SAP_FKEY></ORDER_SAP_FKEY>
            <PROJECT></PROJECT>
            <SALES_ORDER></SALES_ORDER>
            <LOADING_PARTNER_FKEY></LOADING_PARTNER_FKEY>
            <UNLOADING_ADDRESS></UNLOADING_ADDRESS>
            <CARGO />
            <TRANSPORT_PURPOSE_FKEY />
            <EXTERNAL_REMARKS />
            <INTERNAL_REMARKS />
          </item>
        </NAROCILA>
      </POSILJKA>
    </m:POSILJKA_IN>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Slika 17 - Primer XML strukture in vhodnih podatkov za kreacijo pošiljke.

The screenshot displays a SAP transport management system window titled "Pošiljka iz projekta 1000000 Spreml.: Pregled". The interface includes a menu bar with options like "Transport", "Obdelava", "Prehod", "Okolje", "Sistem", and "Pomoč". Below the menu is a toolbar with various icons. The main content area is divided into tabs: "Obdelava", "Identifikacija", "Izračun tovorn.", "Nadzor", and "Upravlj.". The "Identifikacija" tab is active, showing a form with the following data:

Vrsta trans.	0001 Pošiljka iz pro	TočPlanaTransp	0001
Celotni status	1 Mat. plan. zaključ	Stat. razpisa	AC Sprejeto od špediterj
Špediter	385189		
Pot transporta	020 20 DNI	Vozni red poti	
Vrsta odpreme	01 Tovornjak	Pogoj odpreme	
Vrsta transp.	1 Tovor	Posebna obdel.	
Registracija	LJ	Registr.prikol.	LJ1
DovŠkupTež	KG		

Below the form, there are tabs for "Termini", "Nadalj.termini", "Segmenti", "Partner", "Teks.", "Trajanje in distanca", "Status", "Razpis", and "Dod.podatki". The "Termini" tab is active, showing a table with columns for "Planiranje" and "Izvedba". The "Izvedba" column shows a date and time of 26.05.2008 00:17 with a green checkmark, indicating successful execution. A list of activities is shown on the right, including "Dispozicija", "Registracija", "Začetek nalaganja", "Konec nalaganja", "Odprema", "Začetek transp.", and "Konec transporta".

Slika 18 - Uspešno kreirana pošiljka, ki je pripravljena za nadaljnje delo.

Test in šolanje uporabnikov pred preklpom v živo (prenos na produkcijski sistem)

Test vmesnikov smo izvedli skupaj z uporabniki. Prvi test je bil izveden na razvojnem sistemu. Pri testu smo odkrili največ napak in pomanjkljivosti vmesnikov, kot so podatkovni tipi, nepopolna polja, vsebina polj, itn. Težave z odkrivanjem napak smo imeli zaradi protokola SOAP, ki ne ponudi popolnega opisa napake. Kljub temu smo ugotovili vzrok napak, katere smo takoj odpravili. Po odpravljenih napakah smo izvedli test na testnem sistemu, ki je bil uspešen. Hkrati smo izvedli tudi izobraževanje uporabnikov, ki je predpogoj za delo na produkcijskem sistemu. Po testu in izobraževanju uporabnikov, so bili vmesniki pripravljene za zagon na produkcijskem informacijskem sistemu SAP in na produkcijskem sistemu borze prevoznitva.

4.4 Zaključek poglavja

Na konkretnem primeru implementacije in uporabe vmesnika smo se naučili, da je sama tehnologija le majhen del celotne vpeljave nekega vmesnika. Že na začetku pri sami definiciji problema se moramo posvetiti pravim problemom, ki vodijo naročnika po zahtevi za pripravo vmesnika. Vmesnik ni nujno prava rešitev problema. Z napačno definicijo lahko prispemo do neuspešne implementacije.

Po odločitvi o pripravi vmesnika se moramo najprej odločiti o tehnologiji povezovanja. Glede na vse parametre okolja in z uporabo preteklih izkušenj se odločimo za najprimernejšo rešitev. Tudi ta korak je izredno pomemben, saj s seboj prinaša vse posledice odločitve. V našem primeru smo se odločili za uporabo spletnih storitev, ki pav drugačnih okoliščinah morda ne bi bila nujno najboljša rešitev.

Tudi pri sami izvedbi se je pokazalo, da je zelo pomembna priprava. S pravilno definicijo problema in odločitvami o uporabljeni tehnologiji izvedba ni težavna. Seveda smo se še vedno srečevali z nepredvidenimi situacijami, ki smo jih hitro rešili s timskim delom. S pravilno pripravljenim in izvedenim testiranjem smo odpravili še zadnje pomanjkljivosti. Vmesniki so bili predani v uporabo v predvidenem roku.

Uspešna implementacija je prinesla izboljšavo delovanja organizacije prevozov. Uporabniki nimajo več težav pri koordiniranju prevozov, ni več netočnosti pri naročilu tovornjakov, zdaj se že v naprej ve kake so potrebe po velikosti tovornjaka in prevoznik take tudi pošlje. Ni več težav pri sledljivosti, saj je vse elektronsko arhivirano.

Se pravi da so z informacijsko podporo za upravljanje s prevozi in informacijskim sistemom SAP le prišli do izboljšav.

Do kakšnih izkušenj sem prišel pri razvoju vmesnika med sistemoma? Ugotovili sem, da spletne storitve imajo tako prednosti kot slabosti, ki pa jih je potrebno preveriti na vsakem konkretnem primeru posebej. Le te so odvisne od množice parametrov konkretnega okolja ki pa se lahko zelo razlikujejo od primera do primera.

5. ZAKLJUČEK

Namen diplomske naloge je bil predstaviti konkretno uporabo spletnih storitev, ki vodijo v povezovanje distribuiranih in do sedaj nezdružljivih informacijskih sistemov, v katerih je ogromna količina prenosa poslovnih in drugih podatkov. Pri tem smo nadgradil obstoječi sistem s podporo izmenjave XML sporočil z zunanjim sistemom (borza prevoznništva).

V nalogi sem opisal postopek priprave spletne storitve, pregled obstoječih standardov - XML, WSDL, SOAP in preučil njihovo delovanje, kaj vsebujejo in kaj se na podlagi teh da narediti. Seznanil sem se s programskim okoljem informacijskega sistema SAP. Ugotovil sem da je podpora spletne storitve v novejših verzijah informacijskega sistema SAP zelo visoka. Na podlagi ugotovljenega smo začeli z izdelavo dvosmernega vmesnika za prenos podatkov med informacijskim sistemom SAP in borzo prevoznništva. Med razvojem vmesnikov ni šlo vse tako gladko kot smo si predstavljali, naleteli smo tudi na težave (kot so razlike pri podatkovnih tipih in pri dolžin polj v XML strukturi). Uspešno smo odpravili vse napake in tako ustvarili stabilen vmesnik za prenos podatkov med heterogenima sistemoma.

Izkazalo se je, da tehnologija spletnih storitev ni vedno najboljša rešitev za povezovanje dveh sistemov. Za vsak konkretni primer moramo preučiti vse možne načine povezovanja in se z ocenjevanjem prednosti in slabosti odločiti o načinu povezovanja. Pokazalo se je tudi, da so spletne storitve dovolj razvita tehnologija za uporabo v praksi in da to tehnologijo podpira veliko programskih rešitev. To je tudi osnova, da sploh lahko implementiramo spletne storitve v praksi z zadovoljivim uspehom.

6. VIRI

- [1.] SAP PRESS: SAP Authorization System, Design and Implementation of Authorization Concepts for SAP R/3 and SAP Enterprise Portals, Galileo Press, 2003
- [2.] SAP PRESS: ABAP Objects, ABAP Programing in SAP Netweaver, Galileo press, 2007
- [3.] SAP PRESS; Mario Linkies, Frank Off: SAP Security and Authorizations, Galileo Press, 2006
- [4.] Wikipedia, zadnji dostop december 2006, http://en.wikipedia.org/wiki/OSI_model
- [5.] SAP Press, zadnji dostop november 2006, <http://www.sap-press.com/>
- [6.] IT-Audit.de, zadnji dostop november 2006, <http://www.it-audit.de>
- [7.] Wikipedia, zadnji dostop december 2006,
[http://en.wikipedia.org/wiki/Firewall_\(networking\)](http://en.wikipedia.org/wiki/Firewall_(networking))
- [8.] Wikipedia, zadnji dostop december 2006,
http://en.wikipedia.org/wiki/Public_key_certificate
- [9.] Seymour Bosworth, M. E. Kabay: Computer Security Handbook, 4th Edition, John Wiley & Sons, 2002
- [10.] ITRFC 2828 - Internet Security Glossary, zadnji dostop december 2006,
<http://www.faqs.org/rfcs/rfc2828.html>
- [11.] Wikipedia, zadnji dostop december 2006,
http://en.wikipedia.org/wiki/Trusted_Computer_System_Evaluation_Criteria
- [12.] Wikipedia, zadnji dostop september 2008,
http://en.wikipedia.org/wiki/Web_services
- [13.] Wikipedia, zadnji dostop september 2008, <http://en.wikipedia.org/wiki/ABAP>
- [14.] SAP Press, zadnji dostop september 2008, <http://www.sap-press.com/>
- [15.] Matjaž B. Jurič with Ramesh Nagappan, Rick Leander, S. Jeelani Basha, Professional J2EE EAI, Wrox Press Inc; ISBN: 186100544X; 1st edition, december 2001
- [16.] Matjaž B. Jurič, Nadia Nashi, Mike Swainston Rainford, Mack Hendricks, John Carnell, J2EE Design Patterns Applied, Wrox Press Inc; ISBN: 1861005288; 1st edition, junij 2002

- [17.] ITP. G. Sarang, Kyle Gabhart, Matjaž B. Jurič, Rahim Adatia, et. al., Professional EJB, Wrox Press Inc; ISBN: 1861005083, julij 2001
- [18.] Matjaž B. Jurič, Bojan Kraut, ebXML in Insurance Software: OdaTeam Messaging Spec, XML Europe 2002, ebXML Track, Barcelona, april 2002
- [19.] Matjaž B. Jurič, J2EE for Enterprise Application Integration and e-business Integration, sprejeto na ACM OOPSLA'2002, Seattle, Washington, november 2002
- [20.] Matjaž B. Jurič, Marjan Heričko, J2EE, EAI & Web Services, Web Services Journal, May 2002, Vol 2, Issue 5
- [21.] Matjaž B. Jurič, Ivan Rozman, J2EE as the Platform for EAI, Java Developers Journal, March 2002 in April 2002, Vol 7, Issue 3 in 4
- [22.] Matjaž B. Jurič, Integrating EJBs with CORBA, JavaWorld, <http://www.javaworld.com/javaworld/jw-03-2002/jw-0329-corba.html>
- [23.] . Matjaž B. Jurič, Aleš Živkovič, Ivan Rozman, Are Distributed Objects Fast Enough?, More Java Gems, Cambridge University Press, 2000
- [24.] Matjaž B. Jurič, Ivan Rozman, Simon Nash, Java 2 Distributed Object Middleware Performance Analysis and Optimization, ACM SIGPLAN Notices, 2000, Vol. 35, No. 8, str. 31-40
- [25.] Matjaž B. Jurič, Optimizing Enterprise Java Beans's Performance, vabljeno predavanje na SIGS Conference for Java Development, San Jose, October 28 - November 1, 2001
- [26.] Matjaž B. Jurič, Reusing Legacy Applications in Modern e-commerce Solutions, vabljeno predavanje na Wrox Professional JAVA Developer Conference, Nehru Centre, Bombay, 1-3 March, 2001

7. IZJAVA

Spodaj podpisani Nenad Ljuboja izjavljam, da sem diplomsko nalogo izdelal samostojno pod mentorstvom doc.dr.Mojce Ciglarič. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

Datum

Kandidat