

UNIVERZA V LJUBLJANI  
EKONOMSKA FAKULTETA

**DIPLOMSKO DELO**

MARJAN AMBROŽIČ

UNIVERZA V LJUBLJANI  
EKONOMSKA FAKULTETA

DIPLOMSKO DELO

**INTEGRACIJA POSLOVNIH PROCESOV Z ORODJEM  
BIZTALK STREŽNIK**

Ljubljana, november 2003

MARJAN AMBROŽIČ

## IZJAVA

Študent **Marjan Ambrožič** izjavljam, da sem avtor tega diplomskega dela, ki sem ga napisal pod mentorstvom **dr. Mojce Indihar Štemberger**, in dovolim objavo diplomskega dela na fakultetnih spletnih straneh.

V Ljubljani, dne 3.11.2003

Podpis \_\_\_\_\_

## Kazalo

<b>1. Uvod</b> .....	<b>1</b>
<b>2. Integracija poslovnih procesov in aplikacij</b> .....	<b>2</b>
2.1. Integracija poslovnih aplikacij .....	3
2.2. Upravljanje poslovnih procesov .....	5
2.3. XML.....	5
2.4. Projekt e-SLOG.....	8
<b>3. Integracija z uporabo strežnika BizTalk</b> .....	<b>10</b>
3.1. Pobuda BizTalk.....	10
3.2. BizTalk ogrodje.....	11
3.2.1. Protokol SOAP .....	11
3.3. Knjižnica XML shem.....	13
3.4. BizTalk strežnik .....	13
3.5. Orodja BizTalk strežnika.....	14
3.5.1. Nadzorna plošča (BizTalk Management Desk) .....	15
3.5.2. Orodje za vodenje trgovanja med partnerji (Trading partner management).....	15
3.5.3. Urejevalnik pogodb (Agreement Editor).....	15
3.5.4. Urejevalnik (BizTalk Editor).....	15
3.5.5. Prevajalnik (BizTalk Mapper) .....	16
3.5.6. Orodje za razvijanje poslovnih procesov (BizTalk Orchestration Designer).....	16
3.5.7. BizTalk Messaging Port .....	19
3.6. Storitve upravljanja (BizTalk Orchestration Services).....	19
3.6.1. Lastnosti storitev upravljanja.....	20
3.7. Sporočilne storitve (BizTalk Messaging Services) .....	21
3.8. BizTalk strežnik in N-nivojska arhitektura .....	22
3.9. Delovanje strežnika.....	23
3.10. Prednosti in slabosti BizTalk strežnika .....	24
<b>4. Primer uporabe BizTalk strežnika v podjetju Krka d. d.</b> .....	<b>26</b>
4.1. Podjetje Krka d. d.....	26
4.2. Opis poslovnega procesa sprejema naročil .....	26
4.3. Opis implementiranega sistema .....	26
4.4. Možne izboljšave sistema.....	34
<b>5. Sklep</b> .....	<b>34</b>
<b>6. Literatura</b> .....	<b>36</b>
<b>7. Viri</b> .....	<b>37</b>

## Seznam slik:

Slika 1: Vloga tehnologije XML.....	7
Slika 2: Predaja računa .....	10
Slika 3: SOAP sporočilo .....	11
Slika 4: Primer zahteve v obliki sporočila SOAP .....	12
Slika 5: Primer odgovora spletne storitve v obliki sporočila SOAP .....	13
Slika 6: Programska arhitektura .....	14
Slika 7: BizTalk Orchestration Designer .....	18
Slika 8: XLANG risba in XLANG program .....	19
Slika 9: Primer dela naročila v obliki EDIFACT .....	27
Slika 10: Nastavitve sprejemne funkcije .....	28
Slika 11: Nastavitve sprejemne funkcije .....	28
Slika 12: Del XML sheme za EDIFACT naročilo .....	29
Slika 13: BizTalk Mapper.....	30
Slika 14: Del XML sheme za tekstovno datoteko (izhodni dokument).....	31
Slika 15: Del tekstovne datoteke (izhodni dokument).....	31
Slika 16: Del SOR sporočila .....	32
Slika 17: Grafična predstavitev implementiranega sistema podjetja KRKA d. d.....	33

## **1. Uvod**

Podjetja in organizacije vse bolj poslujejo preko interneta. Prenos poslovanja na internet predstavlja podjetjem zahtevno nalogo, saj poslovanje prek interneta prinaša nove, večje zahteve. Internetno poslovanje zahteva povezovanje med seboj ponavadi popolnoma različnih sistemov, od katerih se zahtevajo razpoložljivost, učinkovitost in razširljivost. Tem zahtevam pa obstoječi sistemi, zgrajeni za ne-internetno poslovanje, niso kos.

V današnjem času, času globalizacije, sodelovanja med podjetji in razmahom interneta je vse bolj pomembno povezovanje in interakcija med različnimi sistemi. Stranke od informatikov pričakujejo, da bodo aplikacije delovale skupaj, ne glede na sisteme in platforme, na katerih temeljijo in da bodo podatki iz ene aplikacije na voljo drugi v realnem času. Tako na področju informatike postaja eden od ključnih elementov razvoj rešitev, ki bi na čim bolj enostaven in učinkovit način omogočale komunikacijo med sistemi.

V diplomski nalogi se ukvarjam s tematiko, ki postaja čedalje bolj aktualna pri nas, in sicer kako čim bolj učinkovito zagotoviti, da bodo obstoječe aplikacije izkoristile internet in vse možnosti, ki jih ta ponuja. Povezljivost različnih sistemov preko interneta je glavno vodilo te naloge.

Na trg programske opreme je v letu 2000 prišel Microsoft-ov BizTalk strežnik (Microsoft BizTalk Server) - orodje, ki nam ponuja prav to povezljivost; enostavno povezovanje obstoječih sistemov, avtomatizacijo in nadzor le teh ter varno komunikacijo preko interneta.

Vendar pa je BizTalk strežnik razmeroma novo orodje, ki se še ni uveljavilo in splošno prijelo, zato je bil moj cilj to orodje podrobneje spoznati. Določenega orodja pa po mojem mnenju ni možno spoznati samo iz literature, ampak so potrebne tudi praktične izkušnje. Zato je bil tudi eden izmed ciljev te diplomske naloge, da najdem podjetje, ki dejansko uporablja ta produkt, da bi njegovo uporabnost lahko preučil na realnih primerih. Diplomska naloga tako obravnava teoretično in praktično uporabnost tega orodja. Glede na to, da je potrebno poznati širok spekter tehnologij pri spoznavanju takšnega orodja, sem se v diplomskem delu poskušal navezati tudi na aktualne projekte iz tega področja, ki se izvajajo v Sloveniji. Tako GZS<sup>1</sup> v sodelovanju z nekaterimi

---

<sup>1</sup> GZS – Gospodarska zbornica Slovenije

slovenskimi podjetji izvaja projekt e-SLOG<sup>2</sup>, katerega namen je izdelati XML<sup>3</sup> sheme za dokumente, ki se uporabljajo v poslovnem svetu. Vzporedno s tem pa nekatera podjetja na svetovnem spletu že objavljajo svoje interne XML sheme za dokumente, ki so potem dostopne za vse njihove poslovne partnerje.

Namen razvoja orodja Microsoft BizTalk strežnik in izvajanja teh projektov pa je čimbolj učinkovita in enostavna integracija aplikacij in poslovnih procesov znotraj same organizacije in tudi z drugimi organizacijami.

## ***2. Integracija poslovnih procesov in aplikacij***

Podjetja se neprestano večajo, širijo svoje poslovanje z namenom povečevanja tržnega deleža. Prilagoditi se morajo tudi spremembam, ki jih narekuje moderna ekonomija. Podatke morajo kar najhitreje pretvoriti v informacije in te naprej v znanje. Zaradi tega se spreminjajo tudi njihovi poslovni procesi<sup>4</sup> – lahko se razširijo, lahko se pojavijo novi poslovni procesi, ki do določene velikosti podjetja niso bili potrebni. S tem se spreminjajo tudi poslovne aplikacije, ki podpirajo te poslovne procese.

Za takšna podjetja predstavlja rešitev uporaba strategije integracije poslovnih procesov (angl. BPI – Business Process Integration). Integracija poslovnih procesov omogoča takšnim podjetjem, da povežejo interne poslovne procese in informacije znotraj podjetja in tudi s poslovnimi procesi partnerjev, kupcev in dobaviteljev. Z integracijo poslovnih procesov morajo podjetja izkoristiti obstoječo informacijsko tehnologijo in ji dodati novo vrednost. Novo vrednost pa predstavljajo nove potrebe podjetja, ki se pojavijo z rastjo podjetja ali s prilagoditvijo na razmere elektronskega poslovanja. Eden od ključnih ciljev integracije poslovnih procesov je ohranjanje obstoječih investicij v informacijsko tehnologijo.

*Integracija poslovnih aplikacij* (angl. EAI - Enterprise Application Intergration), *upravljanje poslovnih procesov* (angl. BPM – Business Process Management) in *elektronska izmenjava podatkov* (angl. EDI – Electronic Data Interchange) skupaj tvorijo celovito rešitev za integracijo poslovnih procesov, katere centralna točka so poslovni

---

<sup>2</sup> e-SLOG - Elektronsko poslovanje slovenskega gospodarstva

<sup>3</sup> XML – angl. eXtensible Markup Language. Ker v slovenski literaturi ne obstaja poslovenjena kratica za razširljivi označevalni jezik, bom v nadaljevanju dela uporabljal angleško kratico XML.

<sup>4</sup> Poslovni proces je vnaprej planiran odziv podjetja na poslovne dogodke. Poslovni proces se v informacijskem sistemu odraža kot pretok informacij med aplikacijami. Posameznim akcijam v poslovnem procesu ustreza en ali več prenosov sporočil med aplikacijami. Vejanje v poslovnem procesu pa implementiramo kot odločitve na podlagi prejetih sporočil (Dolenc, 2003, str. 116).

procesu in ne povezave med aplikacijami. Medtem ko se integracija aplikacij ukvarja z izmenjavo informacij med dvema aplikacijama ne glede na poslovni proces, integracija poslovnih procesov upošteva celoten potek poslovnega procesa in različne aplikacije potrebnih za dokončanje posla. Integracija poslovnih procesov je torej strategija, ki uporablja poslovna pravila za opredelitev vpliva ene aplikacije na drugo, da bi poslovanje potekalo čimbolj učinkovito (Selčan, 2003, str. 108).

Največji problem integracije poslovnih procesov je, da so definicije poslovnih procesov običajno preveč ohlapne. Ker je njihov namen predvsem opis poslovanja, ne vsebujejo vseh podatkov, ki so potrebni za implementacijo v informacijskem sistemu. Zato težko prevedemo opise poslovnih procesov v obliko, ki jo razume integracijski posrednik<sup>5</sup>, ne da bi poznali podrobnosti samega informacijskega sistema.

Poleg tega se razlike med poslovnim in informacijskim pogledom na proces pojavijo takrat, ko proces vključuje akcije in odločitve ljudi oziroma le teh ne moremo avtomatizirati. Pogost primer je vejanje v poslovnem procesu, kjer se veja izbere glede na človeško odločitev, ki ni direktno sporočena informacijskemu sistemu. Zato moramo modele procesov prilagoditi; npr. tako, da sklepamo na odločitev iz poznejših dogodkov (Dolenc, 2003, str. 116).

## **2.1. Integracija poslovnih aplikacij**

Integracijo poslovnih aplikacij lahko opredelimo kot proces povezovanja različnih aplikacij v podjetju, ki so bile razvite neodvisno, ki uporabljajo različno neprilagojeno tehnologijo in jih neodvisno tudi upravljamo. Integracija aplikacij pomeni izmenjavo informacij med aplikacijami, kar vključuje pripravo aplikacij za pošiljanje, sprejem in reagiranje na informacije. Povezovanje različnih aplikacij kot celovite poslovne rešitve (angl. ERP – Enterprise Resource Planning) in poslovnih partnerjev je kompleksna zadeva. Problem predstavlja predvsem dejstvo, da večina aplikacij temelji na lastni (interni) obliki za vnos in izvoz podatkov, kar onemogoča direktno izmenjavo podatkov. Za izmenjavo podatkov med dvema različnima aplikacijama se je potrebno dogovoriti za »skupen jezik« (Selčan, 2003, str. 107). Vedno več se kot »skupni jezik« uporablja razširljivi opisni jezik XML. To temo obravnava poglavje 2.3. v nadaljevanju diplomske naloge.

---

<sup>5</sup> Ker bi pri povezovanju vsake aplikacije z vsemi drugimi število povezav (in s tem kompleksnost sistema) raslo s kvadratom aplikacij, se običajno odločimo za uporabo integracijskega posrednika. Vloga posrednika je, da ponuja skupno infrastrukturo za integracijo, kot so npr. sporočilne vrste, repozitorij oblik sporočil, nadzor in upravljanje sistema z ene točke, vmesniki za standardne aplikacije... (Dolenc, 2003, str. 115).



Podjetja se z integracijo obstoječih poslovnih aplikacij izognejo stroškom, zato raje vzdržujejo in povezujejo že obstoječe aplikacije, kot da bi razvijala ali kupovala nove. Integracija poslovnih aplikacij tako temelji na ideji, da lahko samostojne, že obstoječe programe med seboj povežemo brez večjih sprememb le-teh. Tako povezan sistemi naj bi bili cenejši in bolj prilagodljivi na spremembe kot pa monolitne in vseobsegajoče rešitve.

S povezovanjem aplikacij pa pridobimo tudi nove zmožnosti informacijskega sistema. Tako povezan sistem predstavlja zelo dobro osnovo za izgradnjo celostnih pogledov na informacijski sistem, npr. sisteme za podporo odločanja<sup>6</sup>, skladiščenje podatkov<sup>7</sup> in rudarjenje podatkov<sup>8</sup> (Dolenc, 2003, str. 114).

Pri integraciji poslovnih aplikacij se pojavljata dva problema (Dolenc, 2003, str. 114):

1. **Na kakšen način prenesti informacije iz ene aplikacije v drugo?** Prenos informacije lahko razdelimo na osnovna opravila. Najprej moramo informacijo zajeti, nato prevesti v standardno obliko ali v obliko, ki jo razume ciljni sistem. Šele tedaj lahko informacijo usmerimo in prenesemo v ciljno aplikacijo. Tam jo bo treba morda spet prevesti in na koncu naložiti v ciljno aplikacijo. Ves čas pa je treba preverjati, da informacija ustreza vnaprej določenim poslovnim pravilom in da ima pravilno obliko. Vrsto prenosa na grobo razdelimo na sinhroni (pošiljatelj čaka na odgovor prejemnika) in asinhroni prenos (pošiljatelj takoj nadaljuje z delom). Pri asinhronem prenosu pogosto uporabimo posrednika. V primeru, ko je naslovnik trenutno nedosegljiv, posrednik zanj hrani njemu namenjena sporočila. Ko postane naslovnik spet dosegljiv, mu posrednik sporočila dostavi. Običajna implementacija posrednika je sporočilna vrsta.
2. **Kako obvladati kompleksnost integriranega sistema?** Kompleksnost sistema lahko zmanjšamo z uporabo enega orodja in implementacije za posamezno področje ter s standardnimi formati sporočil. Raznolikost sistema močno raste že s samim dodajanjem aplikacij (ker ima vsaka svoj format sporočil in svoj nabor pravil, omejitev in predpostavk o podatkih). Zato si je treba prizadevati, da raznolikosti ne povečujemo še z integracijo samo.

---

<sup>6</sup> Sistem za podporo odločanju je orodje, ki analizira poslovne podatke in jih predstavi na tak način, da uporabniki lahko lažje sprejemajo poslovne odločitve. Navadno se sistemi za podporo odločanju uporabljajo pri ogromnih količinah podatkov iz katerih s pomočjo različnih orodij/aplikacij iščejo podatke in iz njih izluščijo uporabne informacije (RRC, 2003).

<sup>7</sup> Podatkovna skladišča se uporabljajo za poizvedovanje in analizo podatkov. Poleg same relacijske baze, podatkovno skladišče navadno tvorijo tudi sistemi za uvoz, predelavo in polnjenje podatkov, orodja za analizo in prikaz podatkov na strani uporabnika in druge aplikacije, ki uravnavajo procese zbiranja podatkov in njihove predstavitve poslovnim uporabnikom (RRC, 2003).

## 2.2. Upravljanje poslovnih procesov

Bistvo upravljanja poslovnih procesov je, da omogoči oblikovanje, analizo, optimizacijo in avtomatizacijo poslovnih procesov. To dosežemo tako, da ločimo poslovno logiko od aplikacij, ki to logiko izvajajo, nadziramo povezave med udeleženci procesa, povezujemo notranje in zunanje procesne vire in nadziramo izvajanje procesov (Selčan, 2003, str. 110).

Prednosti, ki jih upravljanje poslovnih procesov nudi organizacijam, lahko strnemo v (Selčan, 2003, str. 110):

- avtomatizacija ponavljajočih aktivnosti (nalog),
- spremljanje in nadzor nad učinkovitostjo izvajanja poslovnih procesov,
- spreminjanje poslovne logike morajo izvajati poslovni ljudje brez potrebe po pomoči informatikov, saj procese najboljše oblikujejo in izvajajo tisti, ki zanje odgovarjajo.

Za definiranje, spremljanje in nadzor poslovnih procesov uporabljamo orodje za krmiljenje delovnih procesov (angl. Workflow management system). Ta orodja ne podpirajo le posameznih delovnih postopkov, kot so sprejemanje naročil, pač pa informatizirajo celoten proces; kot je npr. delovni proces, ki zajema celotno pot od sprejema naročila do odpreme (Selčan, 2003, str. 110).

Na podlagi napisanega lahko rečemo, da se poslovni procesi odražajo le v usmerjanju sporočil znotraj integracijskega posrednika. Še eden od ciljev integracije poslovnih procesov je, da bi lahko usmerjanje sporočil določalo tudi ne-tehnično poslovno osebje oziroma, da bi se lahko poslovni procesi, ki jih pripravijo poslovni analitiki, enostavno prenesli v informacijski sistem organizacije.

## 2.3. XML

Razširljivi opisni jezik XML predstavlja del integracijske tehnologije v integraciji poslovnih procesov. Uvršča se v področje elektronske izmenjave podatkov, ki skupaj z integracijo poslovnih aplikacij in upravljanjem poslovnih procesov tvorijo celovito rešitev za integracijo poslovnih procesov.

Elektronska izmenjava podatkov je danes pomemben del mnogih poslovnih procesov, saj omogoča izmenjavo podatkov podjetja s kupci, dobavitelji in poslovnimi partnerji. Z uporabo interneta in objavo niza standardov je elektronska izmenjava podatkov postala uporabna za neomejeno število sodelujočih partnerjev. Podobno kot integracija aplikacij

---

<sup>8</sup> Rudarjenje podatkov je tehnika poglobljenega pregledovanja podatkov z iskanjem vedno bolj podrobnih podatkov, z namenom odkrivanja pomembnih vzorcev skritih v podatkih (RRC, 2003).

je tudi elektronska izmenjava podatkov usmerjena bolj na izmenjavo informacij, kot pa na povezovanje poslovnih procesov (Selčan, 2003, str. 110).

Večina dokumentov, ki so danes na internetu, je shranjenih in se prenaša v obliki HTML<sup>9</sup>. To je za učenje in rabo preprost jezik, ki omogoča hipertekstovne povezave in enostavno oblikovanje dokumentov. HTML temelji na standardnem metajeziku SGML<sup>10</sup>, ki je bil oblikovan za opisovanje in rabo formatov dokumentov. SGML omogoča dokumentom, da opišejo svojo vsebino oziroma določijo množico oznak, ki jih dokument uporablja, in strukturne relacije med njimi. Jezik HTML ima v skladu s specifikacijami SGML definirano majhno množico oznak, ki je standardno določena za vse dokumente in ni vključena v same dokumente. Po eni strani to pomeni lažje ustvarjanje novih dokumentov, po drugi pa onemogoča (Leskovar, 1999):

- ustvarjanje lastnih oznak ter atributov, s katerimi bi podatke tudi semantično ovrednotili,
- opis vsebinske strukture dokumentov, ki je npr. potrebna za predstavitev tabel podatkovnih baz ali objektne hierarhije,
- preverjanje pravilnosti podatkov.

SGML na drugi strani omogoča vse, česar HTML ne, in še veliko več. Vendar je jezik SGML preveč kompleksen za vsakodnevno uporabo v svetovnem spletu.

Leta 1995 je zato organizacija World Wide Web Consortium ([www.w3c.org](http://www.w3c.org)) predlagala razvoj jezika, ki bi bil tako enostaven za uporabo kot jezik HTML in močan in razširljiv kot SGML. Prvi osnutek je bil objavljen leta 1996, XML verzija 1.0 pa februarja leta 1998; standard so podprla vsa svetovno najbolj znana podjetja s področja informacijske tehnologije (Sun, IBM, Oracle, Microsoft, ...) (Dečman, 2000, str. 52). XML je tako nastal kot posledica slabosti in pomanjkljivosti, ki jih imata njegova predhodnika: standardni splošni označevalni jezik SGML in jezik za označevanje hiperteksta HTML.

Bistvene prednosti in pomen predstavitve podatkov v jeziku XML (Sturm, 2000) so:

- *ločitev podatkov od njihove predstavitve*: prednost se pokaže v neodvisnem spreminjanju predstavitve podatkov in v predstavitvi istih podatkov na več načinov,
- *neodvisnost podatkov*: predstavitev v XML-u je neodvisna od programske in strojne opreme ter neobčutljiva za njihove nadgradnje,
- *večja uporabnost podatkov*: podatke v XML-u lahko uporabi katera koli programska rešitev, ki podpira jezik XML,
- *razširljivost*: lahko definiramo nove slovarje glede na potrebe konkretnih programov, gospodarske panoge ali načina komunikacije med poslovnimi partnerji,

---

<sup>9</sup> HTML – angl. HyperText Markup Language

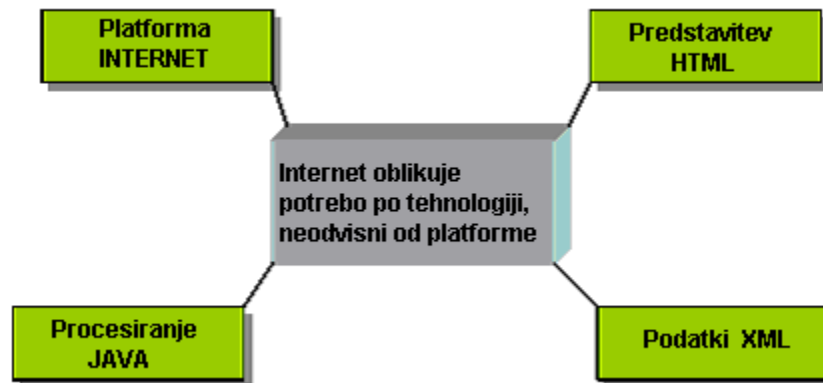
<sup>10</sup> SGML – angl. Standard Generalized Markup Language

- *enostavna uporaba*: XML je samo malo bolj zahteven kot HTML. Vedno več je programov in urejevalnikov, ki podpirajo XML, tako da je verjetno, da ga bo uporabljalo vedno več ljudi,
- *internacionalizacija XML-a*: XML bazira na Unicode<sup>11</sup> sistemu. Ta omogoča uporabo več znakov, kar pomeni, da lahko v XML-u uporabimo tudi druge tuje slovnice. SGML in HTML uporabljata ASCII<sup>12</sup> standard, kar predstavlja problem s tujimi slovnici,
- *podpora razvoju*: dobra podpora razvijalcem in veliko dostopnega gradiva za izobraževanje, predvsem na svetovnem spletu.

Slabost XML-a je v pomanjkanju standardiziranih definicij zaznamkov. Tako se pojavljajo množice definicij zaznamkov s strani različnih podjetij, za podobne ali celo enake rešitve, v različnih ali istih panogah (Dečman, 2001, str. 54).

Ker je XML koncentriran na vsebino, omogoča sistemu na oddaljeni lokaciji izmenjavo in interpretacijo dokumentov brez človeškega posredovanja. XML dokument je le poljubno besedilo. Da bi opisali dejanska pravila, po katerih se moramo ravnati, potrebujemo dodatni mehanizem (Selim et al., 2002). Slika 1 prikazuje mesto in pomen jezika XML.

Slika 1: Vloga tehnologije XML



Vir: Pušnik, 2002.

<sup>11</sup> Standard *Unicode* je univerzalni znakovni kodirni standard za predstavitev besedila, ki ga procesirajo računalniki. Unicode določa konsistenten način kodiranja znakov za vse najpomembnejše svetovne jezike. Določa tudi uporabo matematičnih in drugih tehničnih znakov. Unicode temelji na enostavnosti sistema ASCII in je z njim tudi usklajen (Kovačič, 2003).

<sup>12</sup> ASCII – angl. American Standard Code for Information Interchange

Zaradi kombinacije razširljivosti, strukture in preverjanja je XML uporaben ne le za elektronsko izdajanje, ampak tudi za oblikovanje poslovnih sporočil, ki si jih izmenjujejo podjetja. V primeru velikih procesov se morajo trgovski partnerji strinjati glede celotnega upravljanja poslovanja, preden določijo individualna sporočila. XML besednjak lahko definira poslovne funkcije, najdene v več industrijah, ali pa celotna ogrodja, ki nudijo povezljivost vseh industrij in funkcionalnosti. Programska oprema in sistemi, ki jih podpira XML shema, ne morejo pokriti vseh potankosti, saj bi s tem povečali kompleksnost sistema in drastično povišali stroške. Največja ovira uporabe XML za poslovanje je v tem, da ne nudi združljivosti z različnimi besednjaki, napisanimi za izmenjavo poslovnih sporočil (Selim et al., 2002).

Zaradi svoje fleksibilnosti je postal XML zelo popularen, tako, da se je pojavila množica različnih implementacij XML sintakse za poslovne in druge namene. Zato je bil za namene elektronskega poslovanja razvit enotni standard jezik ebXML, kot plod sodelovanja med podjetjema UN/CEFACT<sup>13</sup> in OASIS<sup>14</sup> (Katić, 2002, str. 9).

ebXML definira uporabo XML za elektronsko poslovanje na globalnem trgu na tak način, da se ohranijo vse prednosti in fleksibilnost XML, hkrati pa obstaja nabor pravil, ki omogočajo globalno kompatibilnost standarda in enostavno povezovanje vseh sodelujočih na globalnem trgu. ebXML v veliki meri temelji na standardu XML, vendar vključuje tudi določene elemente drugih standardov in specifikacij.

## **2.4. Projekt e-SLOG**

Ko se v Sloveniji podjetje odloči, da bo povežalo svoje poslovne procese s poslovnimi procesi svojih partnerjev s katerimi posluje (kupci, dobavitelji, ...), se sreča s problemom pomanjkanja splošno sprejetih standardov, ki bi to omogočili. Zato podjetja povezujejo svoje poslovne procese na individualni ravni (npr. podjetje se z vsakim partnerjem posebej dogovori o formatu dokumentov, ki si jih bosta izmenjevala). Takšno dogovarjanje je zamudno in neproduktivno, zato se je Gospodarska zbornica Slovenije odločila, da bo uvedla splošno sprejete standarde na področju elektronskega poslovanja, ki so v svetu že uveljavljeni.

Gospodarska zbornica Slovenije skupaj s partnerji in pridruženimi člani izvaja projekt e-SLOG z namenom priprave in uveljavitve standardov za elektronsko poslovanje v

---

<sup>13</sup> UN/CEFACT je mednarodna organizacija s sedežem v Ženevi. Njena osrednja dejavnost je razvoj mednarodnih protokolov in standardov za podporo elektronskemu poslovanju. Podpirajo razvoj in implementacijo odprtih, interoperabilnih globalnih standardov in rešitev za elektronsko poslovanje.

<sup>14</sup> OASIS je neprofitni mednarodni konzorcij, predan pospeševanju adaptacije od produktov neodvisnih formatov, ki so zasnovani na standardih v javni lasti. Člani so v veliki meri ponudniki, uporabniki ali specialisti za tehnologije, na katerih ti standardi temeljijo.

gospodarstvu. Partnerji pri projektu so: Mercator, Petrol, Lek, Krka, Pivovarna Laško, tehnološki nosilci rešitev projekta pa so Microsoft, Skupina Atlantis in IBM. Osnovni namen projekta je seznanjanje in praktično usposabljanje slovenskih podjetij za elektronsko poslovanje na skupno dogovorjenih standardih. Projekt ima predvsem naslednje cilje (Prva priporočila projekta e-SLOG, 2003):

- priprava standardnih dokumentov za elektronsko poslovanje med podjetji po internetu,
- varovanje elektronskega poslovanja na temelju elektronskega podpisa in PKI<sup>15</sup> sistema,
- povezovanje različnih tehnoloških rešitev in s tem večji izbor rešitev za posamezna podjetja, ki omogočajo uporabo elektronskega poslovanja v večjih ter tudi v srednjih in malih podjetjih.

Končni cilj pilotnega projekta je uveljavitev celotnega nabora standardnih poslovnih dokumentov: naročilnice, dobavnice in računa.

Kot izhodišče pri izdelavi vsebinskih standardov so bile uporabljene XML sheme za osnovne poslovne dokumente, ki jih je izdal EAN<sup>16</sup> International in UCC<sup>17</sup>. Ker je delovna skupina za vsebinske standarde ugotovila, da je objavljena shema za račun nepopolna, sta Mercator in Merkur izdelala ekstrakt računa iz standarda EANCOM<sup>18</sup>. Po testiranju in uskladitvi elementov računa z ostalimi člani delovne skupine, so račun pretvorili v XML shemo. Tako je sedaj izdelana prehodna shema računa, ki odstopa od objavljenih EAN XML shem. To shemo bodo testirali in preverili pridruženi člani projekta e-SLOG. Skupina je pripravila tudi dokumentacijo XML računa, ki vsebuje opise, XML specifikacije, in primer uporabe specifikacije v konkretnem računu (Prva priporočila projekta e-SLOG, 2002).

Skupina za tehnološke rešitve je uspešno izvedla predajo računa med dvema BizTalk strežnikoma, lociranim na lokacijama A in B (med podjetjem Atlantis in GZS). Obe lokaciji sta imeli isto platformo in avtomatizirano okolje, pri čemer ima lokacija A funkcijo izdajanja računa (izvoz), lokacija B pa funkcijo sprejemanja (uvoz) (glej sliko 2).

---

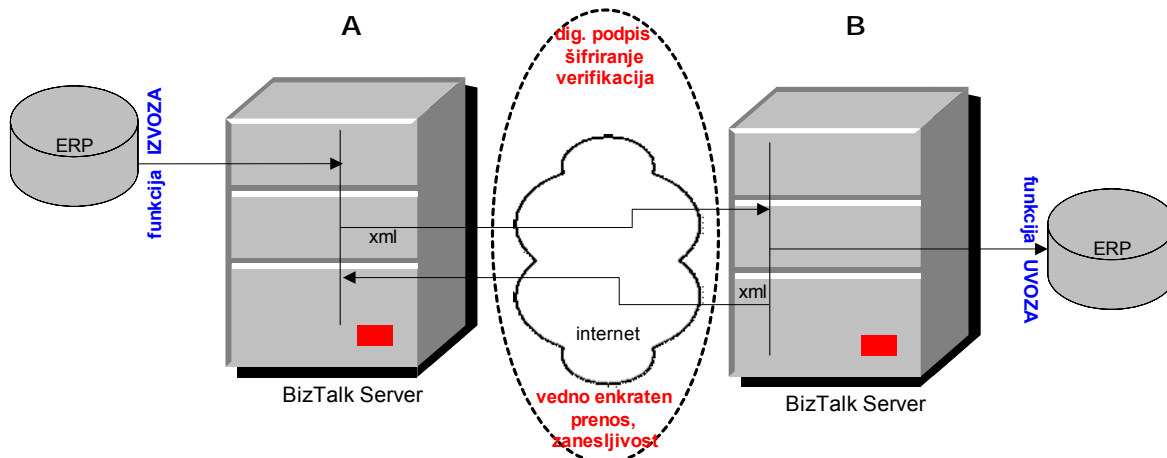
<sup>15</sup> PKI – angl. Public Key Infrastructure

<sup>16</sup> EAN – angl. European Article Numbering

<sup>17</sup> UCC – angl. Uniform Code Council

<sup>18</sup> EANCOM je standard, ki vsebuje definicije sporočil za elektronsko izmenjavo poslovnih dokumentov – poslovnih sporočil. Nastal je leta 1987, kot posledica zahtev podjetij po elektronskem poslovanju. EANCOM vsebuje podмноžico specifikacij standarda UN/EDIFACT in vključuje samo tista sporočila, ki so namenjena elektronskemu poslovanju v preskrbovalnih verigah.

Slika 2: Predaja računa



Vir: e-SLOG Tehnološko navodilo, 2003.

Postopek se začne na lokaciji A, ki iz svojega ERP sistema izvozi račun. Lokacija A poskrbi, da je račun v XML obliki, ga digitalno podpiše in šifrira ter pošlje na lokacijo B. Tam je poskrbljeno, da se sporočilo verificira in uvozi v njihov informacijski sistem. Na lokaciji B morajo imeti možnost potrjevanja prejetega dokumenta. Za vse naloge, ki se izvajajo na obeh lokacijah, poskrbita BizTalk strežnika (e-SLOG Tehnološko navodilo, 2003).

### 3. Integracija z uporabo strežnika BizTalk

#### 3.1. Pobuda BizTalk

Pobuda BizTalk-a predstavlja veliko več kot le produkt BizTalk strežnik. Predstavlja množico vlaganj s strani Microsofta, da bi čim hitreje dosegli cilj, in sicer avtomatizacijo poslovnih procesov, ki bi bila neodvisna od operacijskih sistemov, programskih modelov ali programskih jezikov. Sprožilo jo je podjetje Microsoft z namenom razrešiti problem prenosa poslovnih dokumentov med poslovnimi partnerji.

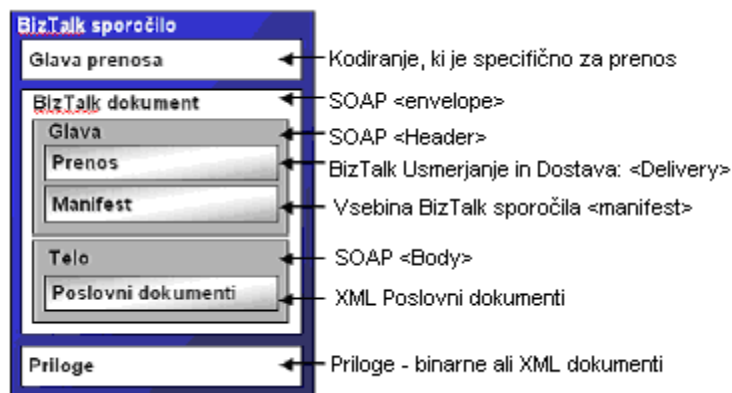
Pobudo BizTalk sestavljajo štiri različni deli:

- neodvisna specifikacija BizTalk ogrodja (*BizTalk Framework Independent Document Specification*),
- knjižnica XML shem,
- strežniki, ki so združljivi z BizTalk ogrodjem (*BizTalk Framework Compliant – BFC – server*). To so strežniki, ki lahko obdelujejo BizTalk dokumente.
- BizTalk strežnik.

## 3.2. BizTalk ogrodje

BizTalk ogrodje je množica navodil za implementacijo sporočilnih protokolov. Vsakdo lahko navodila uporabi za definiranje poslovnih podatkov z uporabo XML standarda. BizTalk ogrodje ni nov standard, ampak gradi na obstoječih standardih. Protokol za sporočila, ki je opisan v specifikacijah BizTalk ogrodja verzije 2.0 je napisan v skladu s SOAP 1.1 specifikacijami. To pomeni, da se BizTalk dokumenti preko interneta prenašajo v bistvu kot SOAP sporočila (glej sliko 3).

Slika 3: SOAP sporočilo



Vir: XML and SOAP Programming for BizTalk Servers, 2000.

BizTalk dokument je SOAP 1.1 sporočilo, v katerem telo sporočila vsebuje poslovne dokumente, glava sporočila pa vsebuje BizTalk-specifične podatke, ki narekujejo kaj in kako delati s sporočili. Element <delivery> vsebuje podrobnosti o usmerjanju in dostavi in ta vsebuje element <reliability>, ki določa, ali je od ciljnega strežnika zahtevano sporočilo o potrditvi prejetja sporočila. Element <manifest> vsebuje spisek vsebine sporočila (vsebuje tudi spisek prilog). Dejanski XML poslovni dokument se nahaja znotraj elementa <Body>.

### 3.2.1. Protokol SOAP

V drugi polovici devetdesetih je skupina programerjev z uporabo preprostih dokumentov XML poskušala preko interneta dostopati do oddaljenih spletnih storitev<sup>19</sup>. Zato so samoiniciativno razvili majhen protokol za izmenjavo sporočil o klicih in rezultatih izvedenih procedur oziroma spletnih storitev. Zaradi svoje narave so ga poimenovali XMLRPC (angl. XML Remote Procedure Call). Za protokol so se zanimali tudi v

<sup>19</sup> Spletna storitev je košček poslovne logike, ki je dostopna preko interneta s pomočjo standardiziranega sporočanja v obliki XML. Omogoča izvršitev specifične operacije ali niza operacij samostojno v okviru spletne storitve, lahko pa v povezavi z drugimi spletnimi storitvami izvede skupek kompleksnih operacij ali poslovnih transakcij. Spletna storitev je opredeljena v standardni notaciji XML, ki se imenuje opis spletne storitve (angl. service description) (Ribič, 2002, str. 47).



Microsoftu in kmalu je nastala osnovna različica tega, kar danes poznamo kot SOAP. Pri nadaljnjem razvoju protokola so sodelovala tudi druga velika podjetja in ga priredila tako, da je uporaben v industriji, vseeno pa s seboj nosi veliko navidezne preprostosti.

Cilji protokola SOAP so (Jovanovski, 2001):

- *omogočiti komuniciranje med različnimi sistemi*; zaradi velike množice različnih računalnikov vseh velikosti in pojavnih oblik se je pojavila težnja po zmožnosti sodelovanja in povezovanja med njimi. Pri tem je treba podpreti pošiljanje sporočil (enosmerna komunikacija) ter izmenjevanje zahtev in odgovorov nanje (dvosmerna komunikacija),
- *preprostost gradnje in razgradnje sporočil*; desetletja prakse so pokazala, da je najmanj napak pri prenašanju podatkov v tekstovni obliki. Poleg tega jih človeški opazovalec najlažje razume,
- *uporaba obstoječe infrastrukture*; na internetu je prisotnih kar nekaj protokolov za prenos podatkov. Standard sam ne zapoveduje protokola za prenos podatkov med aplikacijami, vendar so se avtorji posebej orientirali na protokol HTTP kot prevozno sredstvo za sporočila SOAP.

Sliki 4 in 5 prikazujeta primer zahteve odjemalca (slika 4). Odjemalec od strežnika zahteva izvedbo funkcije »get StateName«, ki sprejme parameter »statenum« tipa integer in vrne vrednost tipa string. Odjemalec bo strežniku poslal naslednjo zahtevo:

Slika 4: Primer zahteve v obliki sporočila SOAP

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope
  SOAP-
  ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance">
  <SOAP-ENV:Body>
    <m:getStateName xmlns:m="http://www.soapware.org/">
      <statenum xsi:type="xsd:int">41</statenum>
    </m:getStateName>
  </SOAP-ENV:Body>
```

Vir: Jovanovski, 2001.

Glede na to, da je 41. država v ZDA Južna Dakota, bo strežnik na to zahtevo odgovoril z naslednjim besedilom (slika 5):

*Slika 5: Primer odgovora spletne storitve v obliki sporočila SOAP*

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance">
  <SOAP-ENV:Body>
    <m:getStateNameResponse xmlns:m="http://www.soapware.org/">
      <Result xsi:type="xsd:string">South Dakota</Result>
    </m:getStateName>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Vir: Jovanovski, 2001.

Protokol SOAP torej odgovarja na vprašanje, kako izmenjati podatke s standardnimi mehanizmi na katerikoli strojni in/ali programski arhitekturi, v kateremkoli programskem jeziku, z uporabo kateregakoli omrežnega protokola in na osnovi kateregakoli objektnega modela.

### **3.3. Knjižnica XML shem**

Do julija 2002 se je nahajala na spletnem naslovu BizTalk.org, potem je Microsoft umaknil to stran. Kot razlog so navedli uveljavitev XML-a kot splošnega standarda, kar je bil tudi namen knjižnice BizTalk.org in prisotnost drugih podobnih knjižnic (npr. XML.org). Knjižnice omogočajo organizacijam iskanje shem, za katere se zanimajo. Prav tako lahko organizacije objavijo svoje sheme.

### **3.4. BizTalk strežnik**

BizTalk strežnik je sistem za prenos podatkov med udeleženci in na poslovnem trgu nudi orodja in infrastrukturo za izmenjavo med različnimi platformami in operacijskimi sistemi. Nudi portal za pošiljanje in sprejemanje dokumentov preko interneta. Sporočila, ki se prenašajo, temeljijo na ogrodjih XML shem in so v skladu z industrijskimi standardi (odprta, fleksibilna, stabilna, zanesljiva, z nizko stopnjo tveganja).

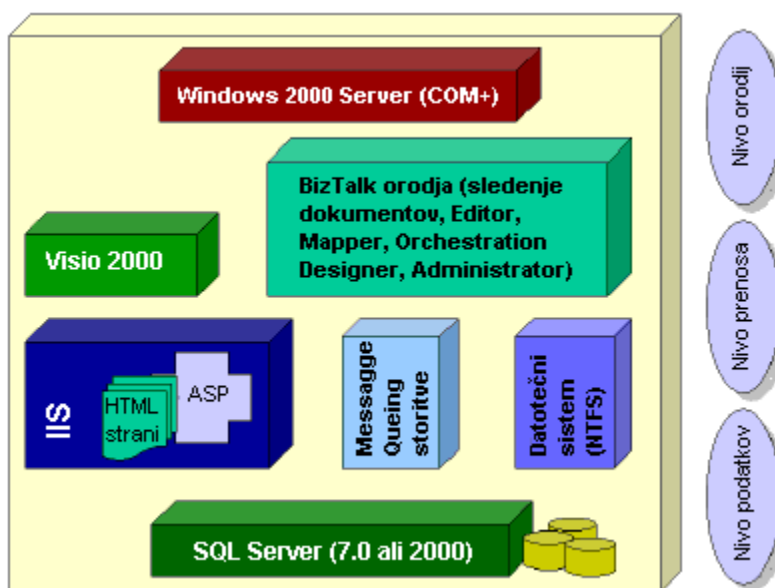
Udeleženci so lahko celotna podjetja, komponente, aplikacije, deli operacijskega sistema, ..., skratka vsi deli poslovnega procesa, ki potrebujejo dostop do podatkov. Ker vsak sistem potrebuje vhod in izhod, BizTalk podpira več protokolov, preko katerih

lahko prenaša podatke, vključno s SMTP, FTP<sup>20</sup>, HTTP, MSMQ<sup>21</sup> (Ritonja et al., 2000, str. 3).

BizTalk strežnik omogoča učinkovito in zmogljivo spletno trgovanje, procesno integracijo in integracijo med podjetji. Podjetjem omogoča združevanje in vodenje poslovnih procesov z izmenjavo poslovnih dokumentov kot so naročilnica in fakture, med poslovnimi aplikacijami znotraj ali izven organizacijskih omejitev.

Windows 2000 predstavlja temelj BizTalk platformi (celotno platformo nam prikazuje slika 6). Vse različice Windows 2000 sistema imajo vgrajeno polno podporo XML-u (XML, XSL<sup>22</sup>, XML Namespaces, XML Shema, ...).

Slika 6: Programska arhitektura



Vir: BizTalk, 2000.

### 3.5. Orodja BizTalk strežnika

Vsa orodja, ki jih nudi BizTalk strežnik, imajo grafični vmesnik. Vsako je namenjeno določeni funkcionalnosti, ki je podprta s strani strežnika, tako da skupaj tvorijo zaključeno celoto.

<sup>20</sup> FTP – angl. File Transfer Protocol

<sup>21</sup> MSMQ – angl. Message Queuing

<sup>22</sup> XSL – angl. eXtensible Stylesheet Language

### 3.5.1. Nadzorna plošča (BizTalk Management Desk)

Orodje BizTalk Management Desk je spletni grafični vmesnik, ki se lahko uporablja na strežniku ali na daljavo prek interneta. Namenjeno je kreiranju novih in urejanju obstoječih relacij med poslovnimi partnerji, kar je bistveno pri izgradnji uspešne rešitve elektronskega trgovanja.

### 3.5.2. Orodje za vodenje trgovanja med partnerji (Trading partner management)

Vodenje trgovanja med partnerji oziroma kreiranje in ohranjanje odnosov s partnerji je ključnega pomena pri gradnji uspešnih rešitev elektronskega trgovanja. BizTalk strežnik zagotavlja pospešeno izmenjavo dokumentov, pri čemer poslovni partnerji pojasnjujejo lastnosti, kot so (Ritonja et al., 2000, str. 3):

- organizacija: certifikati, izmenjava identitet, kontaktna imena, ipd.,
- profil prenosa: enkripcija, kodirni format (MIME<sup>23</sup>), internet naslov,
- pogodbe: proces definiranja specifične izmenjave dokumentov (prevajanje, potrdila in logiranje),
- dokumenti: shema, enkripcijski certifikat in digitalni podpisi.

### 3.5.3. Urejevalnik pogodb (Agreement Editor)

Urejevalnik pogodb se lahko uporablja za kreiranje novih pogodb med organizacijami – trgovskimi partnerji in internimi aplikacijami. Pogodba predstavlja temeljna pravila, prej dogovorjena v izvorni organizaciji in ciljni organizaciji, ki regulira elektronsko izmenjavo podatkov med izvorno in ciljno organizacijo.

### 3.5.4. Urejevalnik (BizTalk Editor)

Urejevalnik oziroma BizTalk Editor je orodje, ki ga lahko uporabljamo za kreiranje in urejevanje specifikacij. BizTalk uporablja specifikacijo za prevod formata vhodnega dokumenta v začasni XML format, s katerim potem BizTalk strežnik operira. Zatem uporabi drugo specifikacijo za prevod obdelanih podatkov v ustrezen format izhodnega dokumenta.

BizTalk Editor specifikacije nudijo običajne opise podatkov, katere lahko BizTalk Mapper uporablja za transformacijo podatkov preko raznovrstnih formatov, in tako omogočijo prenosljivost podatkov med poslovnimi procesi.

Z njim naredimo XML sheme, ki bodo določale strukturo uporabljenih dokumentov. Seveda lahko tudi prevzamemo sheme, ki so jih pripravili drugi. Po načinu dela je ta del podoben urejevalniku za izdelavo ali izpolnjevanje tabele v relacijski zbirki. Za vsak

---

<sup>23</sup> MIME – angl. Multipurpose Internet Mail Extensions

atribut lahko določimo parametre podobno kot pri zbirkah podatkov, attribute pa lahko opredeljujemo tudi hierarhično.

### 3.5.5. Prevajalnik (BizTalk Mapper)

Orodje BizTalk Mapper je namenjeno oblikovanju pretvorbenih tabel, ki povejo kako združiti specifikacije dokumentov. Vsebuje komponento za prevajanje besedilnih slogov (angl. style sheet) in kreiranje XSL datoteke, ki je potrebna za pretvorbo. Vsebuje veliko zelo koristnih pripomočkov, s katerimi lahko kreiramo, pregledujemo in testiramo prevajanje XSL pretvorbenih datotek.

Orodje BizTalk Mapper zagotavlja enostavno uporaben in razširljiv format za izmenjavo pretvorbenih specifikacij. Z njim poenostavimo integracijo aplikacij, ki uporabljajo različne podatkovne formate. Z orodjem BizTalk Mapper lahko izkoristimo moč XML-a, ki nam ponuja pretvorbo med različnimi podatki.

Če je potrebno dodatno računanje ali procesiranje za povezavo med poljem v izvornem in ciljnem dokumentu, uporabljamo t.i. »functoid«. To so že napisane funkcije, ki jih uporabljamo za dodatne izračune ali pretvorbe podatkov. V orodju najdemo knjižnico že obstoječih »functoid«, ki so namenjene za različne pretvorbene scenarije. Alternativen način za uporabo dodatnih »functoid« je, da uporabimo ali napišemo COM<sup>24</sup> komponente.

Orodje je praktično za uporabo, saj preslikave opravljamo na grafičnem modelu, ki na zaslonu nazorno kaže, kakšno preslikavo želimo imeti. Tu lahko povemo, da se npr. polje »ime osebe« preslika v polje »ime«. Pri preoblikovanju se lahko atributi tudi dodatno obdelajo.

### 3.5.6. Orodje za razvijanje poslovnih procesov (BizTalk Orchestration Designer)

Orodje BizTalk Orchestration Designer je namenjeno predvsem modeliranju poslovnih procesov. Tradicionalen pristop h gradnji poslovnih procesov je sestavljen iz dveh ločenih faz. V prvi fazi poslovni proces modeliramo, v drugi pa nastopi kodiranje. V prvi fazi se ponavadi izvaja analiza obstoječega poslovnega procesa (na primer proces nabave blaga) ter kreiranje procesnih diagramov poteka ali diagramov interakcij, s katerimi se obstoječi proces opiše. Faza kodiranja se ponavadi izvaja ločeno. V pristopu, ki ga uvaja BizTalk Orchestration orodje, se kreira abstraktni model poslovnega procesa, ki ga potem povežemo z implementacijskim ogrodjem.

---

<sup>24</sup> COM – angl. Component Object Model

Ena najpomembnejših lastnosti orodja BizTalk Orchestration Designer je integracija dveh prej omenjenih ločenih faz znotraj enotnega razvojnega orodja. Dodatna lastnost, ki jo orodje BizTalk Orchestration Designer ponuja pri integraciji modeliranja in implementacije, je sposobnost kreiranja in upravljanja robustnih, dolgo trajajočih poslovnih procesov, ki niso tesno povezani (niso povsem odvisni drug od drugega) in se lahko raztezajo čez več organizacij, ki temeljijo na različnih platformah in imajo instalirane različne aplikacije.

Grafični vmesnik orodja BizTalk Orchestration Designer sloni na orodju MS Visio, ki ga uporabljamo za risanje XLANG<sup>25</sup> razporedov (slika 7).

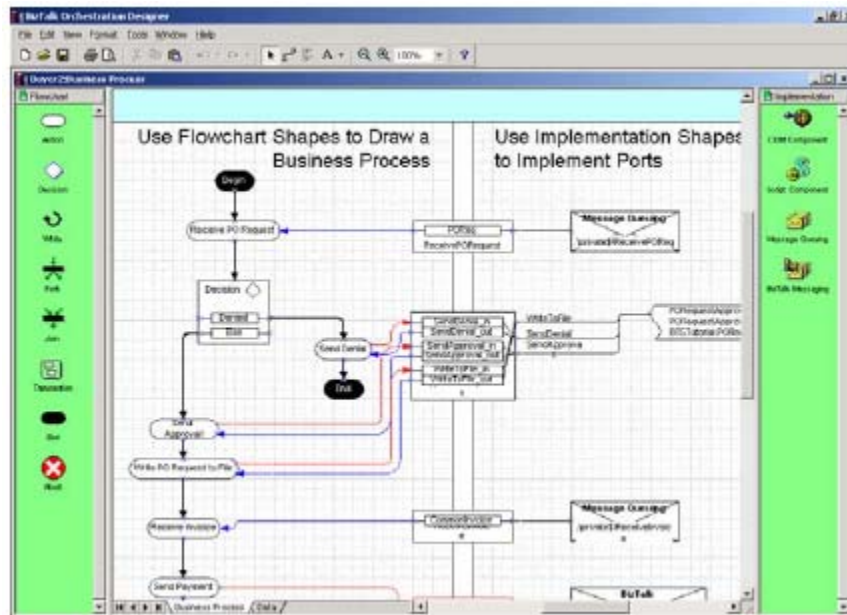
X Language se uporablja kot glavna tehnologija za opis poslovnih procesov znotraj iniciative BizTalk, katere glavni cilj je avtomatizacija poslovnih procesov z uporabo standardnih XML tehnologij.

XLANG je »XML poslovni jezik«, ki omogoča združevanje aplikacij z možnostjo, da razvijalci vključijo aplikacije kot komponente v poslovne procese. Cilj XLANG-a je formalno specificirati poslovne procese. Poslovni procesi vsebujejo vedno več kot samo enega udeleženca. Poln opis procesa mora vsebovati ne samo dejavnosti vsakega udeleženca, ampak tudi kako so te dejavnosti udeležencev usklajene pri izgradnji celotnega procesa. Namen XLANG-a je specificiranje samo dejavnosti, za katere udeleženec želi, da bi jih drugi razumeli v izgradnji svojih storitev (Kosar, 2001, str. 2).

---

<sup>25</sup> XLANG – angl. X Language

Slika 7: BizTalk Orchestration Designer

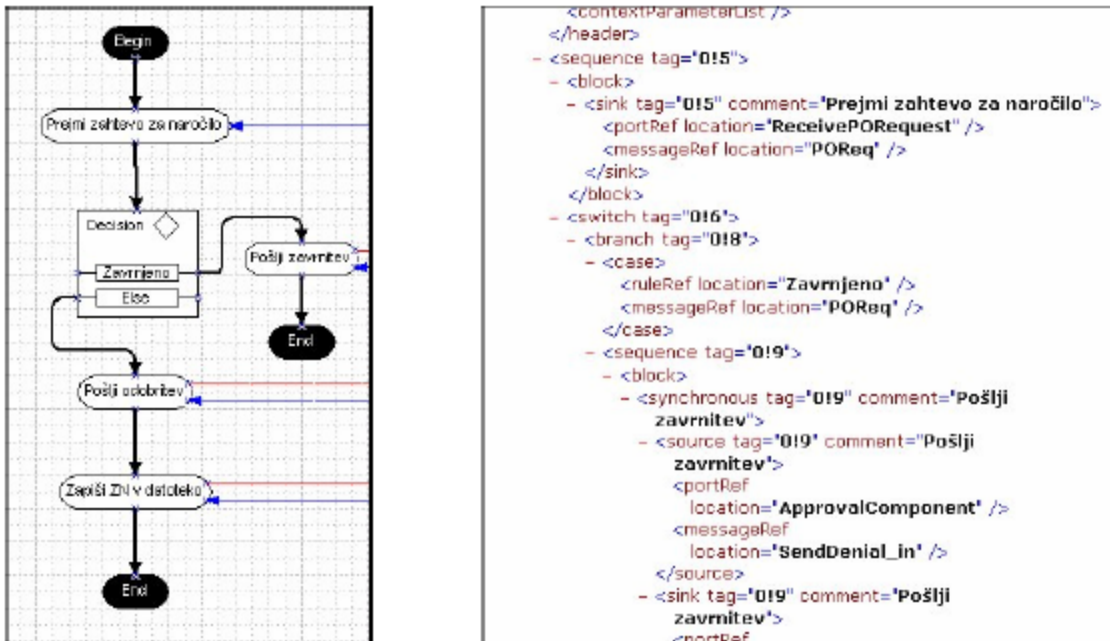


Vir: BizTalk, 2000.

Orodje Orchestration Designer omogoča (Ritonja et al., 2000, str. 7):

- kreiranje skic XLANG razporedov, ki opisujejo poslovne procese,
- da dosežemo implementacijo poslovnih procesov s povezavo določene akcije znotraj risbe z izhodi, ki predstavljajo lokacije, na katere se sporočila pošiljajo in od koder sporočila prihajajo. Izhodi so poimenovane lokacije, sporočila predstavljajo podatke, ki se pošiljajo in sprejemajo med akcijami in izhodi,
- definiranje toka podatkov med sporočili znotraj poslovnih procesov, prevajanje skic XLANG razporedov v XLANG razporede. Skica XLANG razporeda je grafična predstavitev nekega poslovnega procesa v podjetju. XLANG razpored je izvršna XML datoteka, v kateri so shranjene informacije, ki jih vidimo na skici XLANG razporeda (slika 8).

Slika 8: XLANG risba in XLANG program



Vir: BizTalk, 2000.

Prav tu se vidi ločitev določanja izvajanja. Na podlagi teh shem se oblikuje dokument XLANG – postopek za avtomatizacijo, ki ga nato strežnik uporablja v praksi.

### 3.5.7. BizTalk Messaging Port

S pomočjo orodja BizTalk Messaging Port določimo, kako si dve entiteti izmenjujeta podatke. To je točka, ki pove preko katerega transportnega mehanizma bo prišel dokument, kakšen je tip dokumenta in ali uporabljamo dodatne varnostne mehanizme. Vsak dokument lahko postavimo v elektronsko kuverto, s katero določimo kodiranje, tip digitalnega podpisa in druge attribute. Tu lahko določimo tudi alternativni transportni mehanizem, če primerni ne deluje.

## 3.6. Storitve upravljanja (BizTalk Orchestration Services)

Sistem za upravljanje izvajanja poslovnih procesov mora poskrbeti, da se poslovni procesi izvedejo pravilno in da se v celoti končajo. Skrbeti mora, da se sporočila med poslovnimi subjekti pošiljajo v pravilnem vrstnem redu kot to narekuje definicija procesa. Za integracijo in avtomatizacijo izvajanja poslovnih procesov morajo takšni sistemi zagotavljati storitve, kot so (Casati, 2000):

- prilagodljiv in dinamičen tok procesa – sistemi za snovanje, spreminjanje in upravljanje poslovnih procesov bi morali omogočati modeliranje prilagodljivih procesov,



- nadzor transakcij – ta lastnost je zelo pomembna predvsem za obvladovanje izjem in napak pri izvajanju procesov,
- upravljanje znanja poslovnih procesov – sistem za upravljanje procesov lahko poleg analize izvajanja poslovnih procesov nudi tudi avtomatično prepoznavanje procesov, ki so kritični in potrebni preнове.

Storitve upravljanja poslovnih procesov predstavljajo najvišji nivo BizTalk strežnika. Uporabljajo se za modeliranje poslovnih procesov, ki upravljajo celotno poslovno logiko. Pri definiranju poslovnega procesa moramo določiti logično zaporedje akcij in ustrezen tok sporočil.

Storitve upravljanja v BizTalk strežniku vključujejo (Ritonja et al., 2000, str. 34) :

- okolje za vizualno modeliranje (orodje BizTalk Orchestration Designer),
- množico vezi za povezovanje definicij poslovnih procesov z različnimi tehnologijami (COM komponente, razne MSMQ vrste, skriptne komponente, lastnostmi sporočilnega sistema BizTalk strežnika in aplikacijami drugih proizvajalcev),
- modul za izvajanje poslovnih procesov, vzdrževanje njihovega stanja, nadzorovanje in razhroščevanje poslovnih procesov.

### 3.6.1. Lastnosti storitev upravljanja

Storitve upravljanja omogočajo gradnjo in izvajanje poslovnih procesov, ki lahko vključujejo različne akterje (organizacije, aplikacije in ljudi). Namen uporabe storitev upravljanja je, da dosežemo skalabilnost, dinamičnost in robustnost procesov, ki se izvajajo med temi akterji.

Lastnosti storitev upravljanja so (Ritonja et al., 2000, str. 36):

- *Ločitev definicije od implementacije* – programske komponente, ki podpirajo proces ali del procesa, ne smejo imeti direktnega vpliva na definicijo procesa. Danes je definicija poslovnega procesa še vedno tesno povezana z implementacijo. Koda za implementacijo aplikacij in komponent je zapisana skupaj s kodo, ki kreira in nadzoruje procese. Ko pride do spremembe poslovnega procesa, lahko implementacijo hitro in enostavno na novo definiramo.
- *Dinamični procesi* – poslovni procesi niso statični. Živijo in se spreminjajo ter sodelujejo z udeleženci v procesu. BizTalk strežnik zagotavlja močno funkcionalnost, ki omogoča dinamično spreminjanje procesa ob izvajanju glede na komunikacijo z aplikacijami, podatki in elementi, ki so udeleženi v procesu.
- *Integracija vsak z vsakim* – proces mora v času izvajanja komunicirati s programsko opremo. Programska oprema lahko vsebuje aplikacije in komponente, ki tečejo na

različnih sistemih. BizTalk strežnik ima arhitekturo, ki je odprta za povezovanje (v strežnik so vgrajeni adapterji za razne finančne aplikacije in aplikacije drugih proizvajalcev).

- *Sočasnost in sinhronizacija* – sočasnost je skupna vsem poslovnim procesom, a je redko podprta v programskih jezikih in okoljih. V BizTalk strežniku je opisovanje in sočasno izvajanje podprocesov popolnoma podprto. V orodju za vizualno modeliranje procesov takšne procese enostavno narišemo in jih povežemo s programskimi komponentami, ki jih implementirajo. BizTalk strežnik potem poskrbi za sočasno izvajanje in sinhronizacijo procesov.
- *Sposobnost sestavljanja* – zelo redko je v procesu en udeleženec odgovoren za celotni proces, ki se lahko razteza čez več organizacij. V BizTalk strežniku je lahko definicija procesa porazdeljena preko različnih organizacijskih enot. Vsako vlogo pri procesu lahko definira vsak posamezen udeleženec. Na podlagi takega pristopa lahko posamezne komponente definirajo in vzdržujejo tisti, ki najbolj poznajo domeno v kateri te komponente delujejo.
- *Upravljanje in nadzor* - stanje vseh procesov, ki se izvajajo, je stalno v odprti XML shemi in se lahko shrani v katerikoli podatkovni bazi, ki podpira XML. Ta lastnost omogoča enostavno povpraševanje po stanju procesov in napredno upravljanje, kontroliranje in razhroščevanje procesov.
- *Dolge transakcije* - nekateri procesi lahko trajajo po več dni, tednov ali celo mesecev preden se zaključijo. BizTalk strežnik omogoča, da takšne procese obravnavamo skupaj kot enojno, tesno povezano transakcijo. V primeru, da takšna dolga transakcija ne uspe, definiramo nadaljne korake, ki se naj izvedejo.

### **3.7. Sporočilne storitve (BizTalk Messaging Services)**

Storitve sporočilnega sistema («BizTalk Messaging Services») predstavljajo spodnji nivo, ki zagotavlja podporo storitvam upravljanja. Uporabljajo se za pošiljanje in sprejemanje sporočil med poslovnimi partnerji in med poslovnimi aplikacijami in XLANG razporedi znotraj ene poslovne enote. To nam omogoča integriranje poslovnih partnerjev in obstoječih aplikacij v splošne poslovne procese, ki jih lahko upravljamo s storitvami upravljanja.

Sporočilne storitve so orodje za pošiljanje in prejemanje sporočil med poslovnimi partnerji z uporabo internet protokolov. Prav tako so namenjene prenosu sporočil med aplikacijami in XLANG programi znotraj poslovne organizacije.

Dodatno sporočilne storitve omogočajo (Ritonja et al., 2000, str. 25):

- branje in potrditev prihajajočih sporočil,
- sledenje sporočil, ki prihajajo in so namenjena ven,

- izdelavo potrdil o prejemu,
- uporabo map za spreminjanje strukture in oblike podatkov,
- zagotavljanje integritete in varnosti podatkov.

Sporočilne storitve so narejene tako, da podpirajo prejemanje sporočil, ki potem tečejo v poslovni proces, in pošiljanje sporočil, ki tečejo iz procesa. Storitve upravljanja in sporočilne storitve so oblikovane tako, da delujejo skupaj. Sporočilne storitve zagotavljajo storitvam upravljanja nivo, ki podpira prejemanje in dostavo.

Če hočemo dostaviti sporočilo do poslovnega procesa, moramo sporočilo poslati preko orodja BizTalk Orchestration Designer. Ne smemo kreirati monolitnih komponent za integracijo aplikacij, ki bi se obnašale kot komponente za poslovni proces. To bi podrlo ključni razvojni cilj BizTalk strežnika – ločitev poslovnega procesa od njegove implementacije.

### **3.8. BizTalk strežnik in N-nivojska arhitektura**

Tronivojska arhitektura je najenostavnejši primer N-nivojske ali večnivojske arhitekture. Pri tem se število nivojev nanaša na število dodanih poslovnih strežnikov. Aplikacija je lahko na več načinov razdeljena na več platform. Pomembno je, da s takšno gradnjo sistem postane bolj zapleten, vendar s tem povečamo zmogljivost, varnost ali zmanjšamo omrežni promet. S tem se tudi povečuje modularnost sistema, omogoči se enostavni in hitrejši razvoj, olajša se vzdrževanje, še posebno s povečanjem števila uporabnikov sistema (Sajko, 1997).

Lastnosti N-nivojske arhitekture so (Sajko, 1997):

- primernost za aplikacijsko kritične posle (zahtevajo veliko procesorske moči),
- omogoča delo velikemu številu uporabnikov (sto ali tisoč),
- heterogeno računalniško okolje (aplikacije se lahko uporabljajo na različnih platformah),
- distribuiranost podatkov različnih tipov,
- distribuirano procesiranje (s tem povečamo zmogljivost sistema),
- dolg življenjski cikel aplikacije.

V običajnih večnivojskih arhitekturah najdemo tri glavne nivoje, in sicer nivo predstavitve, nivo poslovne logike in nivo podatkov.

Naloga BizTalk strežnika ni prikazovanje podatkov. Zagotavlja lahko vmesnik med nivojem poslovne logike in nivojem predstavitve podatkov (na primer med ASP<sup>26</sup> datoteko, ki se nahaja na spletni trgovini in BizTalk Orchestration nivojem). Vloga sporočilnih storitev je zagotavljanje nivoja sposobnega zagnati XLANG razpored, ki upravlja poslovne procese v odgovor na prejeta sporočila. Z uporabo storitev upravljanja se nivo poslovne logike vzdržuje v centralni enoti, do nivoja poslovne logike pa lahko dostopa neomejeno število odjemalcev.

BizTalk strežnik razširja in podpira N-nivojsko arhitekturo na dva načina (Ritonja et al., 2000, str. 24):

1. dodaja nov nivo («Orchestration Services» - storitve upravljanja) nad nivojem poslovne logike. Storitve upravljanja omogočajo to, da vzamemo svoje obstoječe komponente poslovne logike in jih sestavimo skupaj v poslovne procese. Poslovni procesi se zelo hitro spreminjajo, storitve upravljanja pa nam omogočajo enostavno spreminjanje poslovnih procesov,
2. deluje kot komponenta za dostopanje do podatkov (sporočilne storitve) v nivoju poslovne logike. Komponente za dostopanje do podatkov nam zagotavljajo tehnično infrastrukturo potrebno za pošiljanje v nivo shranjevanja podatkov (podatkovne baze) in sprejemanje iz njega. Sporočilne storitve zagotavljajo sposobnost pošiljanja in sprejemanja sporočil (kot so fakture in naročilnice) v in iz podatkovnih virov. Sporočilne storitve v ta namen podpirajo veliko prenosnih protokolov, kot sta na primer TCP/IP<sup>27</sup> (HTTP/SMTP) in FTP. Podatkovni viri so lahko datotečni sistemi, poštni predali in Microsoft Exchange strežniki, ASP strani in sporočilne vrste.

### **3.9. Delovanje strežnika**

BizTalk strežnik je orodje, ki povezuje različne programe in poslovne partnerje prek različnih komunikacijskih sistemov in tako usklajuje poslovne procese. Program skrbi za prenos, preoblikovanje in usklajevanje podatkov. Pri svojem delu podpira različne tipe dokumentov, ne samo XML. Vhodni ali izhodni podatki so lahko tudi povsem navadne tekstovne datoteke, podatki iz podatkovnih strežnikov in celo dokumenti v zapisu EDIFACT. Toda pri vsakem prenosu se dokumenti najprej interno pretvorijo v obliko XML, ne glede na to, kaj in kje sta izvir in ponor podatkov.

---

<sup>26</sup> ASP – angl. Active Server Pages

<sup>27</sup> TCP/IP – angl. Transmission Control Protocol/Internet Protocol

Za delovanje sistema je potrebno povedati, kje so podatki in kako do njih dostopamo. Za komunikacijo med različnimi informacijskimi sistemi lahko BizTalk strežnik uporablja vrsto transportnih mehanizmov. To so lahko podatkovni strežniki, datoteke na izbranih imenikih ali elektronska pošta.

Strežnik skrbi za zasledovanje dokumentov. Za učinkovito izmenjavo podatkov moramo za vsak dokument vedeti, kdaj se prenaša, kaj se z njim dogaja in kje je, ni pa nujno, da je vsak v pravilnem zapisu. Podatki o dokumentih so shranjeni v posebni zbirki, ki jo lahko shranimo v podatkovnem strežniku, v njej pa lahko tudi nastavimo, katere podatke o dokumentih hočemo shraniti. Za prilagodljivost delovanja obstajajo tudi čakalne vrste, v katerih lahko dokumenti čakajo, da se prejemnik poveže in jih prevzame. Vsi dogodki v zvezi s prenosom dokumentov se seveda beležijo v svoji zbirki. Pri tem je pomembno, da BizTalk-ov mehanizem za prenos omogoča povezave med enim virom in več ponori, ne samo med dvema točkama.

Ko imamo določene sheme in dokumente XLANG, sledi postopek spremljanja in sledenja dokumentov. V zbirki se shranjujejo sporočila, podatki za zasledovanje, opredelitve usklajevanja in drugi podatki. Prav zato, ker se za vsak dokument lahko izbira kar precej podatkov, je priporočljivo, da pri večjih transakcijah to opravljamo v ločenem strežniku. Poleg prilagodljivosti nabora podatkov o dokumentih obstaja tudi programski vmesnik za dostop do podatkov o zasledovanju dokumentov, tako da lahko ta del vgradimo v lastne programe. S priloženim orodjem je mogoče iskati po dokumentih, jih razvrščati in filtrirati.

BizTalk je zelo koristno orodje, ki na pregleden in urejen način omogoča avtomatizacijo zbiranja in posredovanja podatkov. Čeprav se znajde tudi pri slabše urejenih dokumentih, ne more preseči pravila, da so ciljni podatki lahko le tako dobri kakor vhodni (Čepin, 2002, str. 19).

### **3.10. Prednosti in slabosti BizTalk strežnika**

Glede na dosedanji opis lahko prednosti in slabosti strežnika BizTalk strnemo v naslednjih točkah.

**Prednosti** (Čepin, 2002, str. 19):

- Microsoft je same standarde in specifikacije povzel in jih zato ni posebej razvijal. Pozornost so tako preusmerili k poslovni logiki in implementaciji rešitev na višjem nivoju. V BizTalk strežniku je na voljo veliko shem za en poslovni dokument.

Preslikava med različnimi vrstami dokumentov z vmesniki, ki so nam na razpolago, ne predstavlja težav.

- BizTalk strežnik ponuja nabor orodij in aplikacij, ki omogočajo gradnjo poslovnih procesov in tako že v osnovi hitrejše združevanje aplikacij. Takšne vmesnike so podjetja v preteklosti izdelala sama ali pa so jih kupovala po naročilu, kar pa v obeh primerih ni bilo poceni.
- BizTalk strežnik je zelo koristno orodje, ki na pregleden in urejen način omogoča avtomatizacijo zbiranja in posredovanja podatkov. Čeprav se znajde tudi pri slabše urejenih dokumentih, ne more preseči pravila, da so ciljni podatki lahko le tako dobri kakor vhodni.
- Uporaba aplikacij je možna brez znatnega programskega kodiranja in integriranja novih aplikacij, poslovnih procesov in vseh elementov poslovne verige.
- Arhitektura BizTalk strežnika omogoča hitro vključevanje in avtomatizacijo dinamičnih poslovnih procesov z izmenjavo poslovnih dokumentov znotraj ali med organizacijskimi enotami.
- Kljub temu da se veliko podjetij poslužuje EDIFACT standarda za elektronsko izmenjavo podatkov, to ni ovira za uporabo BizTalk strežnika, saj ta podpira več vrst dokumentov, vključno z XML, EDI (EDIFACT in X.12), pa tudi druge oblike prilagojenih in nestrukturiranih datotek.
- Vgrajene transportne storitve, ki se uporabljajo za pošiljanje dokumentov aplikacijam znotraj podjetja in med podjetji, omogočajo zanesljivo dostavo dokumentov.
- Dobro podprta varnost. Podpira vse trenutno uporabljane standarde za varnost, ki zagotavljajo zadostno zaščito elektronskega poslovanja.

### **Slabosti:**

- BizTalk pobuda ni deležna spodbud in podpore institucij (npr: UN/CEFACT) (Čepin, 2003, str. 20).
- Uporaba je vezana le na Microsoftove operacijske sisteme.
- BizTalk strežnik ne ponuja v celoti t.i. »rešitve iz škatle« zato je v določenih primerih še vedno potrebna pomoč programerja (Čepin, 2003, str. 20).
- Omejenost sporočilnih vrst. Sporočilne vrste, ki jih uporablja strežnik BizTalk za prenos podatkov med posameznimi moduli strežnika (kanali, vrata,...) imajo omejeno dolžino prenosa. Ta je sicer omejena na 4 MB, kar je za večino potreb dovolj, vendar pa moramo upoštevati, da se med sistemi lahko prenašajo tudi zelo velike datoteke (Čepin, 2003, str. 20).
- Cena. Čeprav strežnik BizTalk nudi ogromno stvari, pa je cena, ki jo trenutno ima strežnik BizTalk, le malce previsoka. Poleg cene samega strežnika BizTalk, ki stane

okoli 25.000 \$ na procesor, je namreč potrebno prišteti tudi ceno operacijskega sistema Microsoft Windows 2000, ceno podatkovnega strežnika Microsoft SQL Server 2000 in še ceno orodja Microsoft Visio 2000. Vse skupaj pa stane toliko, da se manjša podjetja rada obrnejo stran (za velike je ta cena sprejemljivejša).

#### **4. Primer uporabe BizTalk strežnika v podjetju Krka d. d.**

##### **4.1. Podjetje Krka d. d.**

Krka d. d. Novo mesto sodi med vodilna generična farmacevtska podjetja v srednji in vzhodni Evropi. Predmet poslovanja podjetja je farmacevtsko-kemijska dejavnost, v katero sodi poleg proizvodnje in prodaje zdravil na recept in izdelkov za samozdravljenje tudi proizvodnja in prodaja veterinarskih in kozmetičnih izdelkov. Svoje zmogljivosti podjetje usmerja v razvoj sodobnih, pretežno generičnih zdravil. Preostala Krkina zdravila so plod sodelovanja z vodilnimi svetovnimi multinacionalnimi farmacevtskimi podjetji. Podjetje ima 3.792 zaposlenih, s svojimi izdelki pa so prisotni v več kot 70 državah sveta (Krka d. d. – Predstavitev, 2003).

Podjetje Krka d. d. se je odločilo za uporabo orodja BizTalk strežnik z namenom avtomatiziranja procesa sprejema naročil.

##### **4.2. Opis poslovnega procesa sprejema naročil**

V podjetju imajo implementiran sistem za avtomatsko obdelavo naročil s trgovskim podjetjem Poslovni sistem Mercator d. d..

V grobem ta proces poteka tako, da podjetje Mercator pošlje naročilo za določene izdelke v elektronski obliki z določenega elektronskega naslova. To naročilo prispe na določen elektronski naslov podjetja Krka. Naročilo Krkin sistem obdela in Mercatorju odgovori – pošlje potrditev o prejemu naročila. V odgovoru ni datuma dobave, odobrenih količin, odobrenih proizvodov, ipd., ampak le potrditev o prejemu naročila.

##### **4.3. Opis implementiranega sistema**

Podjetje Mercator pošlje sporočilo v obliki elektronske pošte iz naslova »mercator.nabava@mercator.si«<sup>28</sup> po protokolu SMTP. To sporočilo ima pripeto datoteko v obliki EDIFACT (glej sliko 9), ki vsebuje naročilo posameznih izdelkov. V

---

<sup>28</sup> Zaradi poslovne tajnosti je to izmišljen naslov.

zadevi tega sporočila je navedena EAN koda naročnika. Sporočilo Mercator pošlje na naslov »edi@krka.si«.

Ko sporočilo prispe na ta naslov, ga obdela Krkin interni servis (program zapisan v programskem jeziku Visual Basic). Servis se sproži vsaki dve minuti in preverja, če je prispelo kako novo naročilo. Interni servis pobere iz sporočila pripeto EDIFACT naročilo in ga zapiše na določeno lokacijo na datotečnem sistemu (na primer: »C:\fs\_BizTalk\_inp«). Na katero lokacijo v datotečnem sistemu bo servis zapisal pripeto naročilo je odvisno od zadeve sporočila, kjer je zapisan naročnik. Servis pripete datoteke ne odpira. V tej fazi obdelave nastopi orodje BizTalk strežnik, ki s pomočjo svojih sprejemnih funkcij prebere EDIFACT naročilo in ga pretvori v navadno tekstovno datoteko, primerno za nadaljno obdelavo v Krkinem poslovnem sistemu.

*Slika 9: Primer dela naročila v obliki EDIFACT*

```
UNB+UNOA:3+3838606900014+3838989119492+030703:0900+18781+++++1'  
UNH+1+ORDERS:D:96A:UN:EAN008'  
BGM+220+3838606970789-13103993+9'  
DTM+137:200307030831:203'  
DTM+2:200307070831:203'  
RFF+ON:3838606970789-13103993'  
NAD+BY+3838606900014::9+PS MERCATOR D.D.'  
NAD+DP+3838606970789::9+@ivila in ne`ivila'  
NAD+SU+3838989119492::9+KRKA, TOVARNA ZDRAVIL, D.D.,'  
LIN+1+++3838989718121:EN'  
PIA+1+:SA+414044:IN'  
IMD+F++::MLEKO V PRAHU PIKOMIL1 450G'  
QTY+21:144.00:PCE'  
LIN+2+++3838989718145:EN'  
PIA+1+:SA+414043:IN'  
IMD+F++::MLEKO V PRAHU PIKOMIL2 450G'  
QTY+21:144.00:PCE'  
UNS+S'  
CNT+2:2'  
UNT+19+1'  
UNH+2+ORDERS:D:96A:UN:EAN008'  
BGM+220+3838606970789-13103994+9'  
DTM+137:200307030833:203'  
DTM+2:200307070833:203'  
FTX+PUR+++HIPER LJ'  
RFF+ON:3838606970789-13103994'  
NAD+BY+3838606900014::9+PS MERCATOR D.D.'  
NAD+DP+3838606970789::9+@ivila in ne`ivila'  
NAD+SU+3838989119492::9+KRKA, TOVARNA ZDRAVIL, D.D.,'  
LIN+1+++3838989714932:EN'  
PIA+1+:SA+001604:IN'  
IMD+F++::^AJ BEZEG KRKA F20 30G'
```

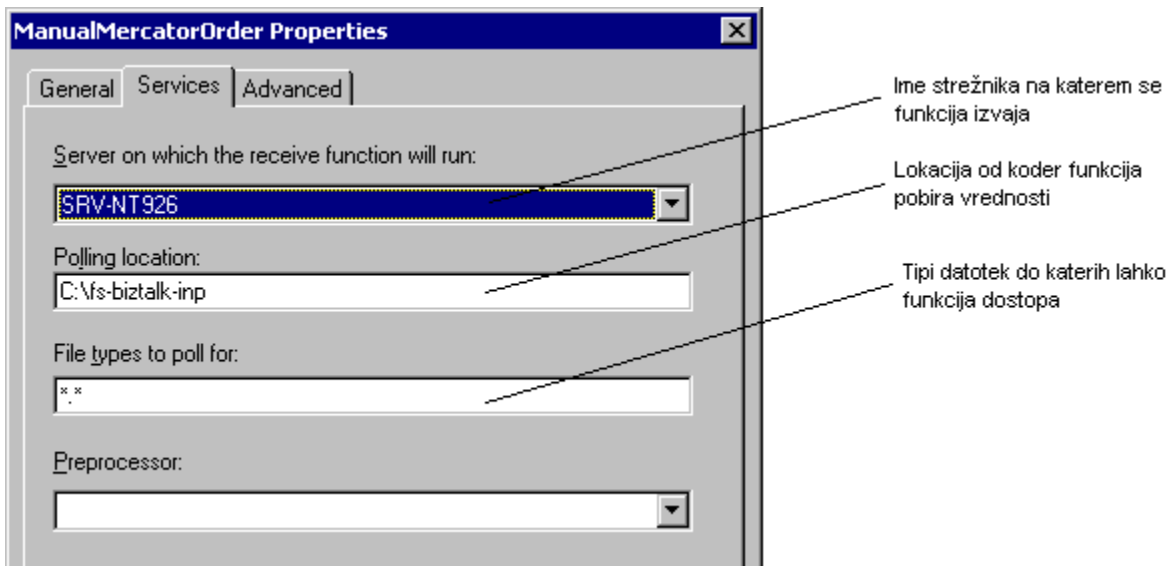
Vir: Interna gradiva podjetja Krka d. d., 2003.

Sprejemne funkcije so pomemben del BizTalk strežnika, saj tu določimo protokol po katerem naj BizTalk pobere podatke (v našem primeru gre za datoteke), lokacijo, od koder naj pobere podatke (v našem primeru »C:\fs\_BizTalk\_inp«) in pa iz katerih tipov



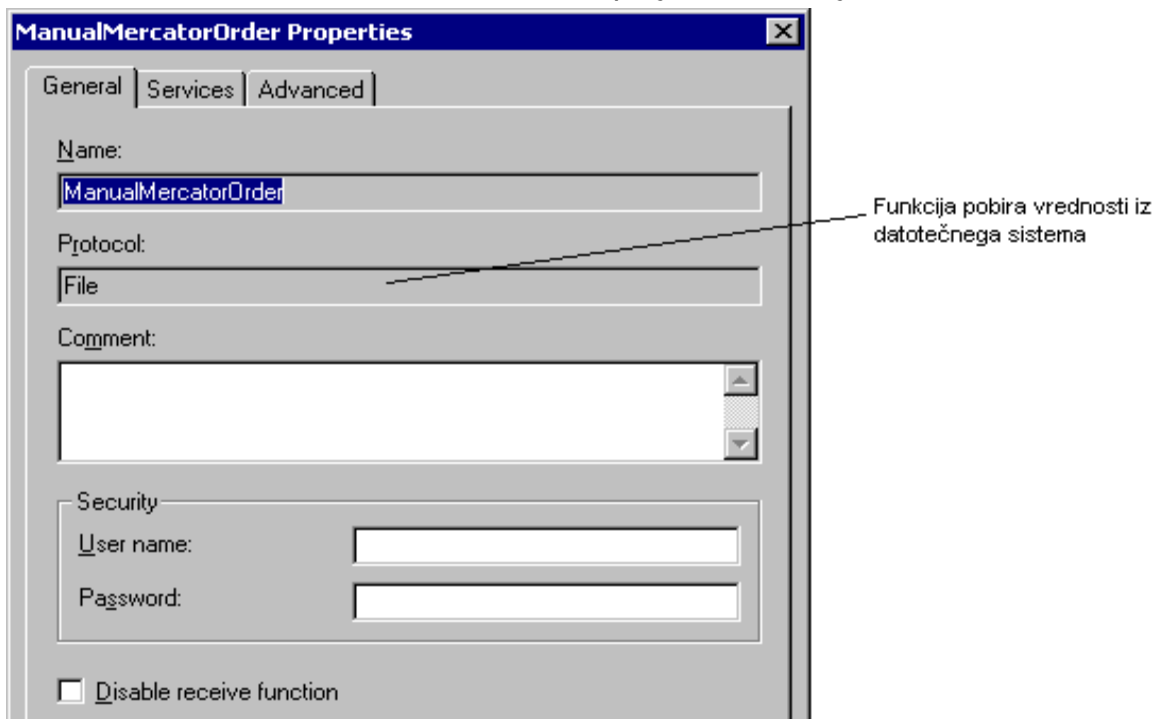
datotek naj pobere podatke (v našem primeru so bili nastavljeni vsi tipi datotek – glej sliki 10 in 11).

Slika 10: Nastavitve sprejemne funkcije



Vir: Interna gradiva podjetja Krka d. d., 2003.

Slika 11: Nastavitve sprejemne funkcije



Vir: Interna gradiva podjetja Krka d. d., 2003.

Pri pobiranju EDIFACT naročila iz datotečnega sistema BizTalk najprej preveri, če je to naročilo veljavno. To pomeni, da morajo elementi tega naročila ustrezati določenim zahtevam glede strukture. Strukturo elementov lahko definiramo na dva načina:

- v definiciji tipa dokumenta DTD – Document Type Definition,
- v XML shemi – XML Schema.

V našem primeru je struktura elementov zapisana v XML shemi. BizTalk torej primerja EDIFACT naročilo z XML shemo, in če je dokument skladen z XML shemo, potem naročilo označi kot razpoznano in veljavno, drugače pa je preusmerjeno v vrsto, kjer se zbirajo nerazpoznana sporočila. Del XML sheme za naročilo prikazuje slika 12.

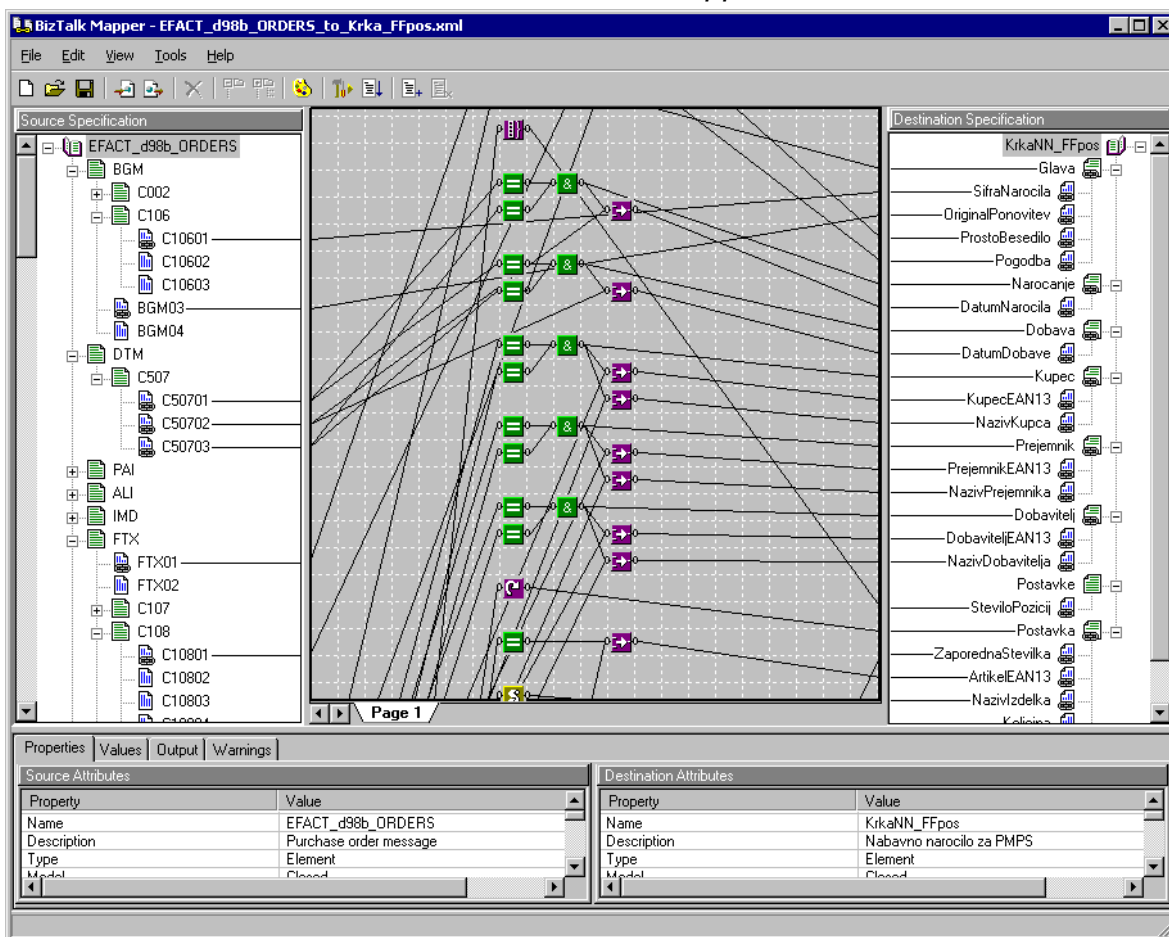
*Slika 12: Del XML sheme za EDIFACT naročilo*

```
<?xml version="1.0"?>
<!-- Generated by using BizTalk Editor on Tue, May 21 2002 09:09:42 PM -->
<!-- Copyright (c) Microsoft Corporation. All rights reserved. -->
<Schema name="EFACT_d98b_ORDERS"
b: BizTalkServerEditorTool_Version="1.5"
b:root_reference="EFACT_d98b_ORDERS" b:schema_type="ORDERS"
b:version="1.0" b:is_envelope="no" b:standard="EDIFACT"
b:standards_version="d98b" xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:b="urn:schemas-microsoft-com: BizTalkServer" xmlns:d="urn:schemas-
microsoft-com:datatypes">
<b: SelectionFields/>
<ElementType name="UNS" content="empty" model="closed">
<description>Section control</description><b: RecordInfo tag_name="UNS"
structure="delimited" delimiter_type="inherit_field" field_order="prefix"
escape_type="inherit_escape" count_ignore="no"/>
<AttributeType name="UNS01" d:type="string" d:maxLength="1"
d:minLength="1"><description>Section identification</description>
<b: FieldInfo edi_datatype="AN" justification="left"/></AttributeType>
<attribute type="UNS01" required="yes"/>
</ElementType><ElementType name="TOD_2" content="eltOnly" model="closed">
<description>Terms of delivery or transport</description><b: RecordInfo
tag_name="TOD" structure="delimited" delimiter_type="inherit_field"
field_order="prefix" escape_type="inherit_escape" count_ignore="no"/>
<AttributeType name="TOD02" d:type="string" d:maxLength="3"
<attribute type="TOD01" required="no"/>
```

Vir: Interna gradiva podjetja Krka d. d., 2003.

Pretvorba naročila iz oblike EDIFACT v tekstovno datoteko poteka s pomočjo orodja BizTalk Mapper, ki skrbi za preslikavo atributov iz izvornega v ciljni dokument. Orodje je uporabniško prijazno, saj preslikave opravljamo na grafičnem vmesniku, ki na zaslonu nazorno kaže, kakšno preslikavo želimo imeti. Pri preoblikovanju se lahko atributi tudi dodatno obdelajo. Na sliki 13 vidimo povezave med izvornimi in ciljnim elementi.

Slika 13: BizTalk Mapper



Vir: Interna gradiva podjetja Krka d. d., 2003.

Pri pretvorbi naročila v navadno tekstovno datoteko BizTalk strežnik poskrbi, da je izhodni dokument veljaven. To pomeni, da elementi izhodnega dokumenta ustrezajo zahtevam glede strukture, ki je določena v XML shemi. (Glej sliko 14)

*Slika 14: Del XML sheme za tekstovno datoteko (izhodni dokument)*

```
<?xml version="1.0"?>
<!-- Generated by using BizTalk Editor on Wed, May 22 2002 06:39:50 AM -->
<!-- Copyright (c) Microsoft Corporation. All rights reserved. -->
<Schema name="KrkaNN_FFpos" b: BizTalkServerEditorTool_Version="1.5"
b:root_reference="KrkaNN_FFpos" b:def_record_delim="0xd" b:standard="FLATFILE"
xmlns="urn:schemas-microsoft-com:xml-data" xmlns:b="urn:schemas-microsoft-
com: BizTalkServer" xmlns:d="urn:schemas-microsoft-com:datatypes">
<b: SelectionFields/>

<ElementType name="Prejemnik" content="empty" model="closed">
<b: RecordInfo tag_name="Prejema::" tag_position="1" structure="positional"
append_newline="yes" skip_CR="no" skip_LF="no"/>
<AttributeType name="PrejemnikEAN13" d:type="string"
d:maxLength="13"><description>NAD C08201 if (NAD01=DP and NAD
C08203=9)</description>
<b: FieldInfo/></AttributeType>
<AttributeType name="NazivPrejemnika" d:type="string"
d:maxLength="35"><description>NAD C05801 if (NAD01=DP and NAD
C08203=9)</description>
<b: FieldInfo/></AttributeType>
<attribute type="PrejemnikEAN13" required="no" b:posStart="11" b:posEnd="23"/>
<attribute type="NazivPrejemnika" required="no" b:posStart="24" b:posEnd="58"/>
</ElementType><ElementType name="Postavke" content="empty" model="closed">
```

Vir: Interna gradiva podjetja Krka d. d., 2003.

Na koncu procesa pretvorbe EDIFACT naročila torej dobimo tekstovno datoteko, ki je kreirana za interne potrebe Krkinega sistema. Sistem v nadaljevanju operira s to datoteko. Slika 15 prikazuje primer dokumenta, ki ga dobimo s pretvorbo iz oblike EDIFACT z BizTalk strežnikom.

*Slika 15: Del tekstovne datoteke (izhodni dokument)*

```
Glava:::::3838606970819-09201437          9
3838606970819-09201437
Naroceno::200307030719
Dobaviti::200307040719
Kupuje::::3838606900014PS MERCATOR D.D.
Prejema:::3838606970819Galanterija (G)
Dobavlja::3838989119492KRKA, TOVARNA ZDRAVIL, D.D.,
Postavke::10
Postavka::1  3838989758653GNOJ.ASEF TE.CVET.R.500ML      336.00    PCE
Postavka::2  3838989758363GNOJ.ASEF TE.CVET.RAS.1L        216.00    PCE
Postavka::3  9000100067010GNOJ.SUBSTRAL 1L                          288.00    PCE
Postavka::4  9000100056328GNOJ.SUBSTRAL 500ML                       192.00    PCE
Postavka::5  3838989702502GNOJ.ASEF-CV.BALK.RASTL.1KG                35.00
```

Vir: Interna gradiva podjetja Krka d. d., 2003.

Ko je takšna tekstovna datoteka kreirana, jo BizTalk strežnik zapiše spet nazaj na neko lokacijo v datotečnem sistemu. Ob tem pa kreira tudi odgovor za stranko, ki je poslala

naročilo (v našem primeru Mercator) v obliki SOR<sup>29</sup>. Ta odgovor ne vsebuje nobenih podatkov, ki bi se nanašali na naročilo, ampak je le potrditev prejema naročila. Že prej omenjeni interni servis pošlje SOR sporočilo na naslov »mercator.nabava@mercator.si« po protokolu SMTP. SOR sporočilo bi lahko poslal tudi sam BizTalk strežnik, le da ima ta možnost pošiljanja sporočil samo po protokolih: SMTP, HTTP, FILE, MSMQ, ne pa tudi po protokolu X.400. V podjetju Krka pa uporabljajo tudi ta protokol, zaradi česar so razvili svoj interni servis, ki omogoča pošiljanje sporočil tudi po protokolu X.400. Primer odgovora v obliki SOR prikazuje slika 16.

*Slika 16: Del SOR sporočila*

```
UNB+UNOA:2+3838989119492+3838606900014+030714:0810+73'UNH+1+ORDRS
P:D:96A:UN'BGM+231+3838989119492-SOR-
00000074+12'DTM+69:200307040719:203'DTM+137:200307140810:203'RFF+ON:3
838606970819-09201437'DTM+171:200307030719:203'RFF+POR:3838989119492-
SOR-00000074'NAD+BY+3838606900014::9+PS MERCATOR
D.D.'NAD+DP+3838606970819::9+Galanterija
(G)'NAD+SU+3838989119492::9+KRKA, TOVARNA ZDRAVIL,
```

Vir: Interna gradiva podjetja Krka d. d., 2003.

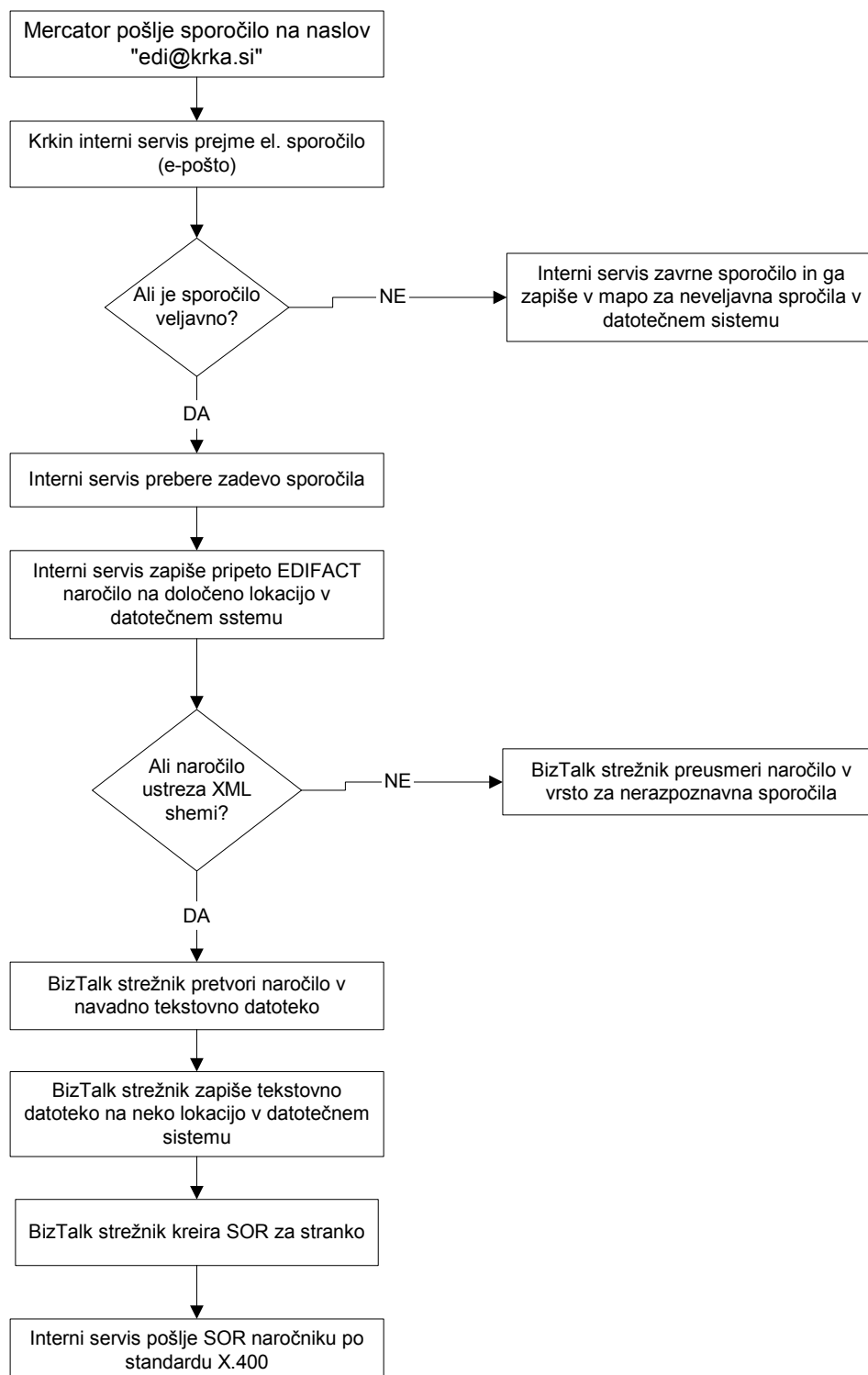
Ko je tekstovna datoteka zapisana na neki lokaciji v datotečnem sistemu, jo tam pobere interna poslovna programska oprema (deluje na operacijskem sistemu Open VMS). Ta programska oprema je napisana v programskem jeziku Cobol in v nadaljevanju procesa obdelave naročila operira s to datoteko.

Grafično predstavitev implementiranega sistema prikazuje slika 17.

---

<sup>29</sup> angl. Standard Order Response

Slika 17: Grafična predstavitev implementiranega sistema podjetja KRKA d. d.



#### **4.4. Možne izboljšave sistema**

Na zgoraj opisanem proces vidimo, da v podjetju Krka pri svojem poslovanju s podjetjem Mercator niso v celoti izkoristili vseh možnosti, ki jih nudi orodje BizTalk strežnik.

Glavno pomanjkljivost vidim v tem, da odgovor, ki ga sistem pošlje nazaj v Mercator, ne vsebuje nobenih informacij razen potrditve sprejema naročila. Pomembno bi lahko izboljšali sistem tako, da bi ta avtomatsko pregledal razpoložljive zaloge in plan proizvodnje, na podlagi teh informacij pa bi lahko v odgovor vključili tudi datum dobave naročenih izdelkov. Vendar pa so mi v podjetju na to vprašanje odgovorili, da to zaenkrat ni mogoče, ker imajo proizvodnjo deficitarno in je zato planiranje proizvodnje kompleksnejši proces.

Kot drugo izboljšavo bi podjetju svetoval vpeljavo izstavljanja računov v elektronski obliki. V obstoječem sistemu podjetje pošilja račune v klasični obliki, kar pomeni veliko dela z ročnim obdelovanjem računov na strani pri prodajalcu in pri kupcu. Orodje BizTalk strežnik omogoča tudi pretvorbo elektronskih računov v XML format in obratno, če je definirana XML shema takšnega računa in pošiljanje (je v vlogi sporočilnega sistema). Seveda pa je od prejetja naročila do izstavitve računa kar dolga pot. V tem primeru bi v podjetju morali v obstoječi sistem vključiti tudi dobavnico (navezuje se na naročilo), za katero je prav tako potrebna XML shema.

### **5. Sklep**

V okviru diplomske naloge sem dodobra spoznal strežnik BizTalk, tako na teoretičnih primerih kot tudi na praktičnem. Ugotovil sem, da je možno probleme, ki jih prinaša povezava sistemov preko interneta, z uporabo strežnika BizTalk učinkovito odpraviti.

Orodje se je v veliki meri približalo temu, čemur je bilo po mojem mnenju tudi namenjeno. To je integracija aplikacij in poslovnih procesov na »out of the box« način. To pomeni, da lahko povežemo različne aplikacije, ki tečejo na različnih platformah, in si izmenjujejo podatke v različnih formatih, skorajda brez znatnega programskega kodiranja, ampak samo z nastavitvami posameznih parametrov v orodju. Seveda je za to potrebno obširno poznavanje samega orodja in pa tehnologij, ki so potrebne za delovanje BizTalk strežnika. Ko pa izpolnemo ta dva kriterija, lahko rečem, da so možnosti, ki jih ponuja BizTalk strežnik, skoraj neomejene. Seveda pa je cena orodja temu tudi primerna. Zato se za ta produkt odločajo bolj ali manj samo velika podjetja, ki

imajo z veliko partnerji dnevne transakcije. Tako je bilo tudi v primeru podjetja Krka d. d., ki sem ga preučil v diplomski nalogi. Vendar pa tudi pri njih BizTalk strežnik ni bil optimalno izkoriščen, saj so ga zaenkrat povezali samo z enim poslovnim partnerjem (Mercator d. d.). Zanimivo pa je, da so v podjetju Mercator d. d. v ta namen razvili svojo programsko opremo (namesto BizTalk strežnika). Očitno je res potrebna podrobna stroškovna analiza, ko se odločamo za nakup takšnega orodja. Brez dvoma pa postajajo orodja kot so BizTalk strežnik vedno bolj popularna in iskana, saj so v večini podjetij prisotni stari sistemi, ki so se skozi leta malo ali nič vzdrževali, in jih je potrebno sedaj vključiti v internetno poslovanje.

Lahko rečemo, da je stanje na področju integracije poslovnih procesov v Sloveniji dobro. Tako so strokovnjaki iz podjetja Zaslon d. d. že v letu 2000 pripravili prve slovenske finančne XML sheme in jih objavili v knjižnici BizTalk.org (Zaslonove novice, 2000). Ne ve se, kaj se je zgodilo s temi shemami po juliju 2002, ko je Microsoft umaknil to knjižnico, vendar se sheme lahko vedno objavijo tudi na kakšnem drugem naslovu (npr. XML.org). Prav tako je z uporabo enostavnega računa v praksi pričel tudi Microsoft Slovenije. Račun oz. XML shema za račun je bila razvita v okviru projekta e-SLOG, kar sem tudi predstavil v diplomski nalogi. Račun, ki ga Microsoft Slovenije pošlje svojim kupcem, lahko kupci obdelajo v elektronski obliki, za kar je Microsoft dobil tudi soglasje DURS<sup>30</sup>-a. Ta podatek nam pove, da se tudi država vključuje v področje integracije poslovnih procesov, kar bo še pospešilo razvoj elektronskega poslovanja pri nas.

---

<sup>30</sup> DURS – Davčna uprava Republike Slovenije



## 6. Literatura

1. Casati Fabio, Ming-Chien Shan: Process Automation as the Foundation for E-Business. Morgan Kaufmann, 2000, str. 688-691.
2. Čepin Peter et al.: BizTalk. Seminarska naloga. Ljubljana: Ekonomska fakulteta, 2002. 23 str.
3. Dečman Mitja: Elektronsko poslovanje in XML. Uporabna informatika, Ljubljana, 8(2000), 1, str. 51 – 56.
4. Dolenc Boštjan: Povezovanje poslovnih aplikacij in upravljanje poslovnih procesov. Zbornik posvetovanja DSI 2003. Ljubljana: Slovensko društvo Informatika, 2003, str. 113-117.
5. Jamnik Anka: Razširljivi označevalni jezik XML in njegove razširitve ter pomen pri razvoju poslovnih internetnih storitev. Magistrsko delo. Ljubljana: Ekonomska fakulteta, 2001. 98 str.
6. Jovanovski Viktor: SOAP. [URL:<http://www.-ai.ijs.si/viktor/papers/soap.html>], 1.9.2003.
7. Katić Uroš: ebXML: Globalni standard za poslovanje na internetu. Diplomsko delo. Ljubljana: Ekonomska fakulteta, 2002. 39 str.
8. Kastelic Bojan: Povezava informacijskega sistema DURS z uporabo programskega orodja BizTalk strežnik 2000. Diplomsko delo. Ljubljana: Fakulteta za računalništvo in informatiko, 2001. 53 str.
9. Kosar Tomaž: XLANG. [URL:<http://lisa.uni-mb.si/student/predmeti/ipp/seminarske/2001-2002/XLANG.pdf>], 10.9.2003.
10. Leskovar Robert T.: V novo tisočletje z Javo in XML. [URL:<http://cotl.uni-mb.si/cotl/99jesenzima/Leskovar99.html>], COTL – spletni časopis Centra za objektno tehnologijo, jesen/zima 1999.
11. Pušnik Maja: Primerjava tehnologij za povezovanje poslovnih procesov med podjetji. Diplomsko delo. Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko, 2002. 85 str.
12. Ribič Miroslav: Implementacija elektronskega poslovanja med podjetji. Magistrsko delo. Ljubljana: Ekonomska fakulteta, 2002. 111 str.
13. Ritonja Boštjan et al.: BizTalk [URL:<http://lisa.uni-mb.si/student/predmeti/ipp/seminarske/2001-2002/BizTalk.pdf>], 20.6.2003.
14. Sajko Uroš: Razvoj sistemov tipa odjemalec - strežnik z orodjem Power-Builder. [URL:<http://lisa.uni-mb.si/student/mk-95/uross/diploma/pogl1.html#N-nivojska%20arhitektura>], 1.9.2003.
15. Selčan Livija: Integracija poslovnih procesov. Zbornik posvetovanja DSI 2003. Ljubljana: Slovensko društvo Informatika, 2003, str. 107-112.

16. Sturm Jake: Developing XML Solutions. Redmond, Washington : Microsoft Press, 2000. 415 str.
17. Travis Brian E.: XML and SOAP Programming for BizTalk Servers. Redmond (WA): Microsoft Press, 2000. 428 str.

## **7. Viri**

1. Active Server Pages Guide.  
[URL:<http://www.microsoft.com/windows2000/en/datacenter/iis/default.asp?url=/WINDOWS2000/en/datacenter/iis/html/asp/aspguide.htm>], Microsoft, 17.9.2003.
2. Aissi Selim, Pallavi Malu, Krishnamurthy Srinivasan: E-Business Process Modeling: The Next Big Step: [URL:<http://www.computer.org/computer/co2002/r5055abs.htm>], 1.9.2003.
3. EAN Slovenija: EANCOM. [URL: <http://www.ean.si/sntportal.asp?p=18&m=19>], 13.10.2003.
4. GZS Portal: e-SLOG Tehnološko navodilo. [URL: <http://www.gzs.si/e-poslovanje/standardi/>], 9.12.2002.
5. GZS Portal: Prva priporočila projekta e-SLOG. [URL:<http://www.gzs.si>]. 9.12.2002.
6. Interna gradiva podjetja Krka d. d.
7. Kovačič Iztok: Uvod v XML.  
[URL:<http://ucilnica.fgg.uni-lj.si/sola/RaclnfUni/arthiv.2001-2002/Gradiva/Uvod%20V%20XML/uvodVXML.htm>], 9.10.2003.
8. Krka d. d. – Predstavitev. [URL: <http://www.krka.si/si/krka/predstavitev/>], 17.5.2003.
9. RRC: Novi trendi. [URL: <http://www.rrc.si/novitrendi/Default.htm#3>], 9.10.2003.
10. Zasloneve novice: Pripravljena izhodišča za oblikovanje standarda za enotno elektronsko poslovanje podjetij z bankami.  
[URL:<http://www.zaslon.si/novice/20000919-BizTalk.htm>], 17.5.2003.