

**UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA**

DIPLOMSKO DELO

SMOTRNOST UPORABE XML STANDARDA V PS MERCATOR D.D.

Ljubljana, avgust 2003

GREGOR ROJKO

IZJAVA

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

V Ljubljani, dne _____

Podpis: _____

KAZALO

1.	UVOD.....	1
2.	ELEKTRONSKO POSLOVANJE.....	2
2.1.	Računalniško izmenjavanje podatkov	2
2.2.	Standardizacija elektronskega poslovanja.....	4
2.2.1.	Standardizacija v Sloveniji	4
2.3.	Oblike elektronskega poslovanja.....	5
2.3.1.	Prednosti in slabosti elektronskega poslovanja med podjetji	7
3.	STANDARD XML	7
3.1.	Razvoj XML-a.....	8
3.2.	Prednosti XML-a	9
3.3.	Oblikovanje dokumentov XML-a	10
3.3.1.	Kaskadne stilne predloge – CSS.....	11
3.3.2.	Razširljiv jezik za stil – XSL.....	12
3.3.3.	Transformacija razširljivega jezika za stil – XSLT	12
3.3.4.	Jezik za oblikovanje izpisa XML – XSLF	13
3.4.	Vmesniki za procesiranje in razčlenjevanje XML-a	14
3.4.1.	Objektni model dokumenta – DOM	14
3.4.2.	Enostavni vmesnik uporabniškega programa za XML – XML SAX.....	15
3.4.3.	Primerjava DOM-a in SAX-a.....	18
3.5.	XML in prenos podatkov.....	19
3.6.	Izmenjavanje dokumentov XML.....	21
3.7.	XML in varnost podatkov	24
3.7.1.	Javni ključi in šifriranje podatkov	25
3.8.	Primerjava XML-a in RIP-a	26
3.9.	Nadgradnja standarda XML	28
4.	POSLOVNI SISTEM MERCATOR D.D.	29
4.1.	Predstavitev	29
4.2.	Dejavnost	29
4.3.	Uvedba RIP-a	30
4.4.	Smotnost uvedbe XML-a V PS Mercator d.d.	32
4.4.1.	Podproces prihod blaga v skladišče.....	33
4.4.2.	Podproces odhod blaga iz skladišča	36
5.	SKLEP.....	38
6.	LITERATURA	40
7.	VIRI.....	43

1. UVOD

Informatika prodira na vsa področja poslovanja. Razvoj informacijske tehnologije in vedno večja potreba podjetij po obvladovanju vseh tržnih segmentov je pripeljala do tega, da so v svoje poslovanje uvedli tudi nov segment – elektronsko poslovanje.

Podjetja nimajo več izbire elektronsko poslovanje da ali ne, sedaj morajo že izbirati, katere standarde uporabljati. Tudi razvoj elektronskega poslovanja napreduje iz dneva v dan, vendar se nekatera podjetja ne odločajo za nove standarde, ampak ostajajo pri uporabi starih, že vpeljanih. V preteklosti so se podjetja povezovala v zasebna omrežja in imela sistem računalniške izmenjave podatkov (RIP). Ta način izmenjave dokumentov je bil drag in primeren za večja podjetja.

V svet informatike je prišel internet in z njim tudi globalno omrežje, tako da podjetjem ni več trebna ustvarjati zasebnih omrežij. Skupaj z internetom pa je prišel tudi jezik XML in na njem temelječ globalni standard, ki bo pocenil RIP, da bodo pri izmenjavi dokumentov lahko sodelovala tudi manjša podjetja brez velikih stroškov.

Namen diplomskega dela je predstaviti standard XML in Poslovni sistem Mercator d.d. ter njihovo elektronsko poslovanje, svetovati kdaj, če sploh, in v kakšni meri prevzeti standard XML.

V drugem poglavju obravnavam elektronsko poslovanje, računalniško izmenjavo podatkov, standardizacijo elektronskega poslovanja v svetu in pri nas ter uveljavljen standard v Sloveniji UN/EDIFACT.

V naslednjem poglavju bom obdelal standard XML: njegov razvoj, orodja za oblikovanje dokumentov XML-a, vmesnika za procesiranje in razčlenjevanje XML-a DOM in SAX ter ju primerjal. Poleg tega poglavje obravnava prenos podatkov, izmenjavanje dokumentov XML in varnost, naredil pa bom tudi primerjavo standardov RIP in XML.

V četrtem poglavju bom predstavil Poslovni sistem Mercator d.d., s sedežem v Ljubljani, ki je vodilno slovensko trgovsko podjetje. Prav tako bom predstavil vpeljevanje elektronskega poslovanja v podjetje. Ogledali si bomo praktični

primer, kaj prinese vpeljava standarda XML v poslovni proces veleprodajno poslovanje.

Zaključil bom z oceno, ali naj Poslovni sistem Mercator d.d. v svojem nadaljnjem poslovanju uvede elektronsko poslovanje na podlagi standarda XML ali naj vztraja pri svojem obstoječem standardu.

2. ELEKTRONSKO POSLOVANJE

Elektronski način poslovanja je v uporabi že nekaj deset let. Uporabljajo ga predvsem velika podjetja in organizacije, kot so zavarovalnice, banke, vladne ustanove in drugi. Izmenjava poslovnih listin oz. dokumentov med velikimi organizacijami poteka na osnovi standarda računalniške izmenjave podatkov (RIP), kar je prevedeno iz angleške Electronic Data Interchange (EDI).

2.1. Računalniško izmenjavanje podatkov

Standard RIP je bil v uporabi še pred komercialno uporabo interneta, zato ima prav posebne zahteve glede programske opreme in je sestavljen iz posebne oblike poslovnih sporočil. Vzpostavitev sistema RIP je bila zelo draga, saj je bilo treba investirati tako v novo programsko kot novo strojno opremo, ki pa sta se razlikovali od že obstoječih v podjetju, ki so se uporabljale za druge rešitve. Tehnologija RIP je zaradi vseh dejstev omejena na zelo ozek krog ustanov in velikih podjetij, ki so si to tehnologijo lahko privoščili. Še preden se je začela poslovna raba interneta, so podjetja za komunikacijo uporabljala omrežja z dodano vrednostjo VAN (Value Added Networks). Arhiviranje, upravljanje poslovnih in pravnih zahtev, spremljanje delovanja v obliki dnevnikov, nadzor nad transakcijami in trgovinskimi profili ter obravnavanje napak so le nekatere izmed možnosti uporabe sistema RIP (Anderson, 2000, str. 581). Ena izmed značilnosti sporočil RIP-a je tudi težko razumljiva sintaksa, za prenos sporočil pa so uporabljali protokole, kot sta FTAM in X.400.

Standard RIP temelji na konceptu transakcij sporočil. Sporočila RIP se prenašajo z uporabo vnaprej definiranih komunikacijskih protokolov, zapisana

pa so po vnaprej definiranih in znanih pravilih in oblikah. Komunikacijski protokoli so večinoma statični in vezani na strojno opremo. Oblika sporočil je bolj dinamična. V posebnih skladiščih so shranjene najpogosteje uporabljene sheme oblik sporočil, ki si jih velike organizacije delijo, ko med seboj elektronsko poslujejo. Sheme do podrobnosti opisujejo obliko podatkovnih objektov, ki jih stranke izmenjujejo znotraj transakcij, podobno kot definicije tipa dokumentov DTD in sheme v XML-u.

V literaturi se pojavlja veliko definicij računalniške izmenjave podatkov. RIP je izmenjava elektronskih dokumentov v standardnem formatu, ki je poznan obema stranema v transakciji ter uporablja elektronski transakcijski medij za prenos poslovnih dokumentov (Huemer, 2000, str. 680). Nekateri avtorji poudarjajo, da RIP za razliko od elektronske pošte pomeni izmenjavo podatkov v standardizirani obliki in tako omogoča avtomatizacijo postopkov njihove obdelave oziroma računalniško komunikacijo brez človeškega vmešavanja (Bračko, 1997, str. 9). Računalniško izmenjavo podatkov najlažje razumemo kot zamenjavo papirnih naročil z elektronskim nadomestkom (Škedelj, 1999, str. 34).

Standardi za RIP so bili sprejeti še pred komercialno uporabo interneta, saj velika podjetja, organizacije in ustanove poslujejo elektronsko že nekaj desetletij. Zahteve glede RIP-a so bile včasih drugačne tako glede standardov kot tudi opreme. Najpogosteje uporabljeno omrežje za prenos podatkov, ki je bilo v uporabi pri RIP-u, je bilo omrežje razvito v skladu s standardom ISO OSI X.400. To omrežje pa ima precej pomanjkljivosti, v primerjavi z omrežjem, na katerem deluje internet (Tabela 1).

Tabela 1: Primerjava prednosti in slabosti omrežja ISO OSI X.400 ter interneta

ISO OSI X.400	internet
- zelo slabo razširjen	+ široka razširjenost uporabe
- dokaj počasen	+ velika hitrost
- draga programska oprema	+ poceni programska oprema
- precej drag	+ poceni
- zahteva predhodne povezave	+ ne zahteva predhodnih povezav
- veliko standardov	+ malo standardov
+ sledljivost je možna	- ni sledljivosti
+ relativna varnost	- ni varnosti

Vir: Škedelj, 1999, str. 36.

Potreba po razvoju oziroma uvedbi standarda, ki bi deloval bolje kot standard EDIFACT, se je pojavila z razvojem interneta kot omrežjem za prenos računalniških podatkov. Tu pa se kot možnost pojavlja nov jezik XML. Jezik XML omogoča, da se RIP sporočila prenašajo zelo enostavno in poceni preko interneta, direktno v programsko rešitev poslovnega partnerja.

2.2. Standardizacija elektronskega poslovanja

Razvoj mednarodno priznanega standarda računalniške izmenjave podatkov, imenovanega UN/EDIFACT, se je pričel leta 1986 v okviru Organizacije združenih narodov. Standard podaja sintaktična pravila za pripravo sporočil, ki se izmenjujejo med partnerji po elektronski poti. Pomembna lastnost UN/EDIFACT standarda je v tem, da so sporočila oziroma dokumenti v celoti šifrirani; standard pa je oblikovan tako, da je v celoti neodvisen od strojne in programske opreme. Besedni zaklad tvorijo podatkovni elementi, ki so predstavljeni v standardu ISO 7372, slovnico pa tvorijo podatkovni elementi, ki so zapisani v standardu ISO 9735. V standardu UN/EDIFACT je implementirana množica svetovno sprejetih standardov in navodil za računalniško izmenjavo podatkov, še posebej v panogah povezanih s trgovino med neodvisnimi, z informacijsko tehnologijo podprtimi, podjetji. UN/CEFACT si prizadeva, da bi vsaka posamezna država jasno opredelila vse potrebne papirne in elektronske postopke ter dokumente, objavila kriterije elektronskega poslovanja v državi ter oblikovala nacionalni standard elektronskih sporočil, ki bi bil v okviru celotnega standarda in s tem definirala nekakšen podstandard UN/EDIFACT.

2.2.1. Standardizacija v Sloveniji

Že od leta 1994 obstaja priporočilo za predlog slovenskega standarda, ki bi temeljil na standardu ISO 9735 in na njegovi podlagi oblikovanih elektronskih sporočil (UN/EDIFACT, SWIFT). Leta 1991 je Ministrstvo za znanost in tehnologijo ustanovilo Urad za UN/EDIFACT, ki naj bi zagotavljal povezavo s standardom in omogočil vpliv na njegovo oblikovanje ter prilagajanje nacionalnim potrebam. Urad za UN/EDIFACT je potreben, saj je slovensko gospodarstvo še prešibko, da bi lahko posamezna podjetja sodelovala s

svojimi strokovnjaki v delovnih skupinah pri Organizaciji združenih narodov (Damjan, 1990, str. 56). Urad naj bi skrbel predvsem za (Bračko, 1997, str. 14):

- razširjanje vsebine standarda, ki bo posebej prilagojena našemu prostoru ter
- usklajevanje vladnih in branžnih delovnih skupin pri dopolnjevanju vsebine standarda.

2.3. Oblike elektronskega poslovanja

Elektronsko poslovanje zajema poslovanje v elektronski obliki na daljavo z uporabo informacijske in komunikacijske tehnologije (Zakon o elektronskem poslovanju in elektronskem podpisu).

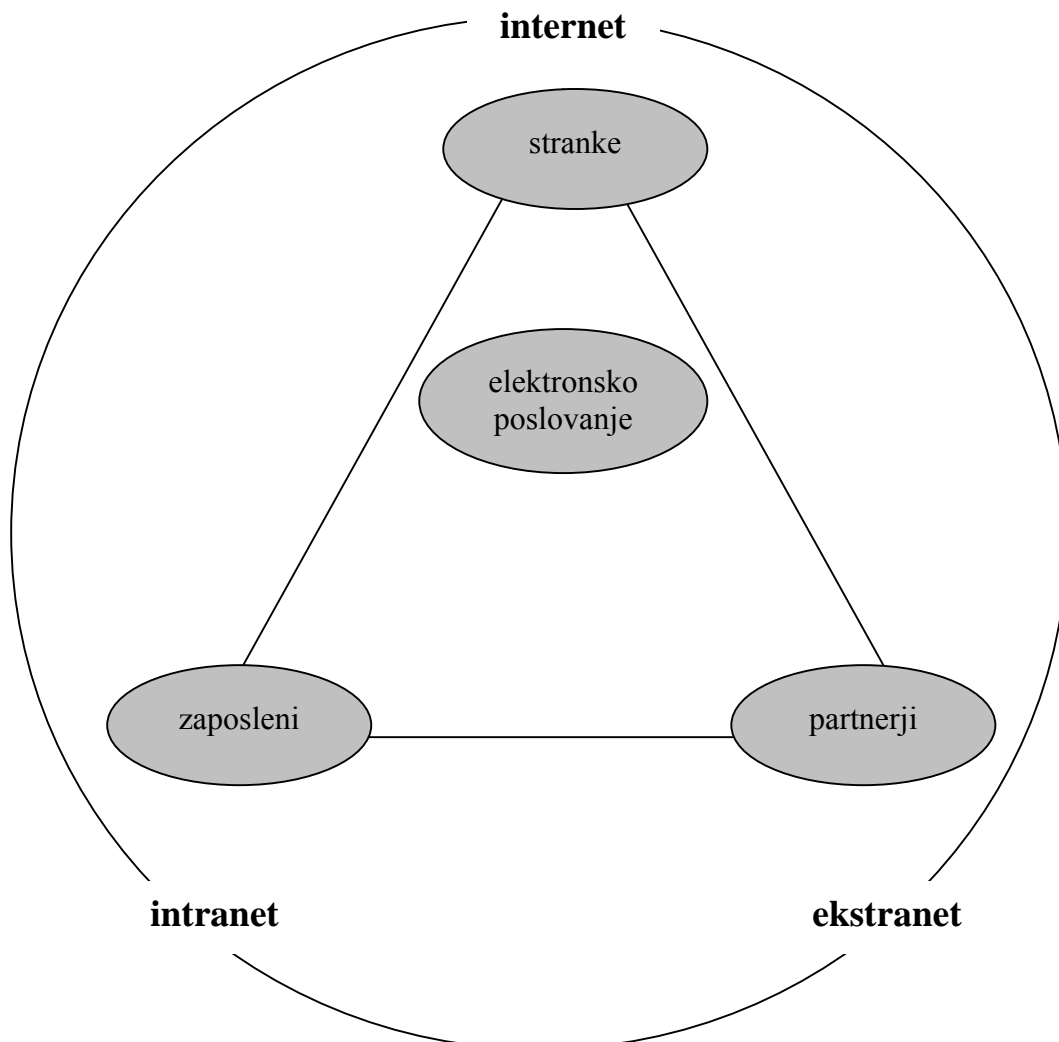
Internet že dolgo ni več samo medij za prikaz in izmenjavo statičnih dokumentov, ampak je prerasel v pomemben kanal komunikacije, oglaševanja ter prodaje za podjetja in organizacije po vsem svetu. Vse, kar je povezano z nakupom in prodajo po internetu, označujemo kot elektronsko trgovanje ali bolj natančno: elektronsko trgovanje pomeni vsako izmenjavo dobrin in storitev za denar med dvema ali več strankami preko svetovnega spleta. Ločimo tri glavne modele elektronskega poslovanja: direktna prodaja stranki (business to customer - B2C), transakcije med podjetji (business to business - B2B) ter delitev informacij in združevanje vsebin. (Anderson, 2000, str. 578)

Večina uporabnikov interneta se sreča z modelom prodaje, ki nam je zaradi tega tudi najbolj poznan. S pomočjo spletnega brskalnika se uporabnik poveže na spletno stran ponudnika, kjer ta predstavlja svoje storitve ali proizvode. Naročene proizvode ali storitve lahko stranka plača na več načinov. Najobičajnejši način je plačilo s kreditno kartico, nakar ponudnik dostavi blago na dom. Povečevanje dobička je eden izmed razlogov, zakaj proizvajalci svoje blago ponujajo na svetovnem spletu. Blago tako ponujajo direktno svojim potencialnim kupcem in se s tem izognejo posrednikom. Po drugi strani pa so z uporabo novih tehnologij povezani tudi novi stroški. Ponudniki, ki na svetovnem spletu predstavljajo svoje proizvode, morajo vlagati denar v oglaševanje svojih spletnih strani, ustreči morajo zahtevam in naročilom strank, opraviti morajo tudi distribucije majhnih količin naročenega blaga. Mnoga podjetja so se zaradi teh novih dodatnih stroškov in obveznosti

povezala s partnerskimi podjetji, ki so posredniki za prodajo na svetovnem spletu in ki v imenu partnerja skrbijo za določene aktivnosti, ki so potrebne pri elektronskem poslovanju.

Slika 1 nam prikazuje možne oblike elektronskega poslovanja. Na njej vidimo, da elektronsko poslovanje ni le poslovanje med dvema partnerjema. Lahko je tudi komunikacija med zaposlenimi v podjetju preko intraneta, sodelovanje s poslovnimi partnerji preko ekstraneta ali pa nam najbolj znana komunikacija s strankami preko interneta.

Slika 1: Oblike elektronskega poslovanja



Vir: Vavan, 2001.

Drugi način elektronskega poslovanja so transakcije med podjetji. Tu gre za transakcije med dvema podjetjema, v prejšnjem primeru pa je šlo za sodelovanje med podjetjem in fizično osebo. Prva razlika, ki se tu pojavi, je plačevanje naročenih storitev. Posamezniki največkrat naročeno blago plačajo s kreditno kartico, podjetja pa imajo pri svojih dobaviteljih največkrat že kar odprt račun. Komunikacija med podjetjema poteka po vnaprej dogovorjenem protokolu.

2.3.1. Prednosti in slabosti elektronskega poslovanja med podjetji

Podjetja se za uvedbo elektronskega poslovanja odločajo na podlagi naslednjih prednosti, obenem pa morajo biti pozorna tudi na slabosti.

Prednosti elektronskega poslovanja med podjetji (Jamnik, 2001, str. 64):

- zmanjšanje obsega papirne dokumentacije, kar pomeni hitreje izvedena naročila;
- avtomatizacija določenih poslovnih procesov (primer naročanje novih izdelkov je neposredno glede na stanje zalog in ga ni več treba izvesti ročno);
- poenostavljanje procesov;
- zmanjševanje števila aktivnosti;
- zmanjševanje stroškov na račun izvajanja poslovnega procesa.

Slabosti elektronskega poslovanja med podjetji:

- socialni nemir – zmanjšanje števila zaposlenih;
- potrebe po integraciji poslovnih procesov in aplikacij zahtevajo velika vlaganja in organizacijske spremembe;
- stroški uvedbe sistema za elektronsko poslovanje;
- stroški vzdrževanja sistema za elektronsko poslovanje.

3. STANDARD XML

Po besedah Johna Ousterhouta obstaja vrzel v elektronskem poslovanju med podjetji (B2B), ki jo trenutno lahko uspešno zapolni XML.

Znanstveniki komiteja ASC X12 so ugotovili, da je XML lahko most, ki bo popeljal standard RIP v vode spletnega elektronskega poslovanja tako, da bo uporabno znanje o standardu RIP olajšalo delo razvijalcem sistemov za spletno elektronsko poslovanje.

Razširljiv označevalni jezik XML (eXtensible Markup Language) je izpeljanka jezika SGML (Standard Generalized Markup Language). Iz jezika SGML so izpeljani tudi drugi jeziki, kot so: VML, MathML, CML in drugi (Mohorič, 2001, str. 55). Prav tako pa je tudi jezik HTML predhodnik jezika XML. Oba omenjena predhodnika imata določene slabosti in pomanjkljivosti. HTML je enostaven za uporabo, vendar ima dve pomanjkljivosti: podatki niso ločeni od oznak in drugih informacij za njihovo predstavitev ter slovar, omejen na nabor znakov, ki jih določa specifikacija in zato jezik ni razširljiv. Prav nasprotno pa velja za jezik SGML, ki je univerzalen jezik za označevanje dokumentov, a je preveč kompleksen za vsakdanjo rabo v svetovnem spletu (Pardi, 1999, str. 16).

Iskanje rešitve je prevzela skupina znotraj konzorcija za svetovni splet W3C – World Wide Web Consortium. Začel se je razvoj novega standarda, ki bo enostaven kot HTML in bo obenem temeljil na jeziku SGML. Vodila razvijalcev novega standarda za predstavitev podatkov so bila ustvariti standardni jezik za označevanje dokumentov, ki bo podpiral širok spekter aplikacij in bo primeren zlasti za široko uporabo na internetu (Bray, 2000).

3.1. Razvoj XML-a

Cilji, ki so jih upoštevali pri razvoju XML jezika, so (W3C):

- XML naj bo enostaven za uporabo preko interneta;
- XML naj podpira širok spekter programskih aplikacij;
- XML naj bo kompatibilen s SGML;
- pisanje programov, ki procesirajo XML dokumente, naj bo enostavno;
- število pomožnih dodatkov v XML naj bo čim manjše, idealno bi bilo brez njih;
- XML dokumenti naj bi bili pregledni in čitljivi;
- načrt XML naj se da hitro pripraviti;

- načrt XML naj bo formalen in jedrnat;
- XML dokumenti naj bodo preprosti za izdelavo;
- natančnost izboljšave XML je minimalnega pomena.

Bistvo XML-a je v tem, da je to zbirka pravil za definiranje pomenskih oznak (tags) in je programski jezik za opisovanje strukture in pomena podatkov. Kako naj se podatki prikazujejo na spletni strani, XML ne opisuje, vendar pa je način izpisa možno definirati s pomočjo ene od njegovih razširitev. Ne vsebuje vnaprej definirane množice oznak, kot to vsebuje HTML, ampak omogoča določanje lastnih oznak, ki pa morajo biti v dokumentu urejene glede na ustrezna pravila. V XML-u so podatki ločeni od svoje predstavitve, to pomeni, da ta jezik služi le za definicijo vsebine, kako pa bo grafično prikazan, pa je odvisno od vmesnika, ki ga uporabljamo.

3.2. Prednosti XML-a

Specifikacija jezika XML določa njegovo namembnost in iz nje so razvidne prednosti in pomen predstavitve podatkov v jeziku XML (Sturm, 2000):

- *Razširljivost:* Možnost definicije novih slovarjev glede na potrebe konkretne aplikacije, gospodarske panoge ali načina komunikacije med poslovnimi partnerji.
- *Izmenjava podatkov in poslovna komunikacija:* Izmenjava podatkov je možna med heterogenimi sistemi.
- *Neodvisnost podatkov:* Dokument XML je neobčutljiv za nadgradnjo programske in strojne opreme. Predstavitev v XML-u je neodvisna od programske in strojne opreme.
- *Večja uporabnost podatkov:* Podatke v XML-u lahko uporabi katerakoli programska rešitev, ki podpira jezik XML.
- *Internacionalizacija:* XML je zasnovan na 16-bitni kodi Unicode sistema, HTML in SGML pa temeljita na ASCII standardu. Zatorej lahko z XML-om uporabljamo tudi tuje slovnice, ne pa samo angleško, ki je osnova pri ASCII standardu.
- *Podpora razvoju:* Obstaja veliko dostopnega gradiva za izobraževanje uporabnikov, predvsem na svetovnem spletu, zato imajo razvijalci sistema zelo dobro podporo za črpanje znanja.

- *Ločitev podatkov od njihove predstavitve:* Iste podatke je možno prestaviti na več načinov. Način predstavitve podatkov lahko spreminjamo neodvisno od sistema, ki ga uporabljamo.
- *Enostavna uporaba:* Zaradi enostavne sintakse je jezik XML primeren z jezikom HTML in iz dneva v dan ga uporablja več ljudi. Posledično z večjim številom uporabnikov se na trgu veča tudi izbira programov, ki podpirajo XML.

3.3. Oblikovanje dokumentov XML-a

XML loči podatke od pravil, kako naj se ti podatki prikazujejo. Za predstavitev podatkov v XML-u se uporabljajo stilni jeziki, ki temeljijo na deklarativnem programiranju in transformirajo dokument XML v zbirko elementov, ki se slušno ali vidno zaznavajo. Deklarativno programiranje pomeni, da aplikacija dobi podatek, kaj želimo, ne pa tudi, kako naj pride do rezultata. Stilne jezike zaznamujeta dve skupini pravil: navodila za delovanje in pravila za ujemanje po vzorcu oziroma predlogi.

V uporabi sta dva mehanizma, ki izpišeta podatke:

- kaskadne stilne predloge CSS – Cascading Style Sheets;
- razširljivi jezik za stil XSL – Extensible Stylesheet Language.

Pri predstavitvi dokumenta XML v spletni aplikaciji sta dve možnosti prikaza: na strani odjemalca (uporabnika) ali pa na strani strežnika. V primeru da uporabnik uporablja spletni brskalnik, ki ne podpira XML-a, mora strežnik preoblikovati dokument v predstavitevno obliko in šele nato ga pošlje odjemalcu. V nasprotnem primeru, ko brskalnik uporabnika podpira XML, pa spletni strežnik enostavno pošlje dokument XML s pripadajočo stilno predlogo brskalniku uporabnika, ki nato dokument sam preoblikuje v predstavitevno obliko.

V naslednjih podpoglavjih so opisani različni načini, kako prikazovati XML dokumente. Marsikateremu uporabniku je vizualna oblika zelo pomembna, saj pripomore k lažji prepoznavnosti predstavljenih podatkov. Kako se ti podatki prikazujejo, kakšna so pravila prikazovanja in druge pomembne lastnosti oblikovnega vidika standarda XML si pogledjmo na naslednjih primerih.

3.3.1. Kaskadne stilne predloge – CSS

Kaskadne stilne predloge CSS - Cascading Style Sheets temeljijo na modelu škatel (boxes). Škatla je definirana kot pravokotno območje z osnovnimi značilnostmi, kot so dimenzije in odmiki od robov. Škatle so zložene ena v drugo. Tak model omogoča predstavitev v obliki drevesa pravokotnih območij. Drevo vseh škatel je vsebovano v krovnem pravokotniku, ki je odvisen od predstavitvenega medija. Lahko je nepretrgan kot v brskalniku, kjer se lahko pomikamo po zaslonu gor in dol, lahko pa je fiksnih dimenzij kot v primeru tiskanih medijev. Elemente v dokumentu XML povezujemo s škatlami preko pravil v jeziku CSS. Pravila sestavljajo množice lastnosti, ki so povezane z določenim tipom elementa ali elementov. Vsako pravilo je sestavljeno iz proceduralnega dela, ki vsebuje zbirko lastnosti ter iz vzorca za ujemanje, ki ga imenujemo selektor. Lastnosti opisujejo, kako naj bo škatla predstavljena, selektor pa določa ime elementa (Anderson, 2000, str. 669).

Model CSS je odvisen od strukture elementa, ki ga predstavlja. Drevo komponent dokumenta XML in drevo objektov za obliko in predstavitev sta pogosto enaka. Najpomembnejši predstavitveni objekti v CSS-u so:

- plavajoči objekt (float),
- tabela (table),
- vrstični objekt (inline),
- seznam (list) in
- blok (block).

Smer izpisa določa, v kakšnem vrstnem redu so bloki zloženi eden na drugega. V primeru da je smer izpisa od vrha proti dnu in od leve proti desni, potem so bloki v krovnem pravokotniku nanizani od vrha proti dnu. Vrstični objekti so vključeni v blok. Smer pisanja določa, kako so vrstični objekti vključeni v blok. Ta model imenujemo plavajoči model predstavitve, ker se objekti izpišejo eden za drugim v smeri pisanja in znotraj območja, ki jim je na voljo. Poleg tega omogoča CSS tudi postavljanje posameznih objektov na absolutne pozicije ali pa postavljanje posebnih plavajočih pravokotnikov, okrog katerih se razporedi vsebina ostalih objektov (Bos, 1998, 1999).

Z jezikom CSS lahko podatke prikažemo na več načinov:

- izpis na tiskalnik,
- izpis na ekran,
- posredovanje na govorno napravo,

- posredovanje na televizijo,
- posredovanje na napravo za slepe.

Vidimo, da lahko isti XML dokument s CSS-om prikažemo na različne načine, ne da bi posegali v podatke same. Zapis podatkov v dokumentu XML z jezikom CSS je neodvisen od aplikacije ali naprave, ki te podatke prikazuje.

3.3.2. Razširljiv jezik za stil – XSL

Jezik XSL opisuje kako prikazati XML dokument. Stil zahteva izvorni XML dokument, ki vsebuje informacije, ki jih bo XSL prikazal in dokument, ki določa pravila, ki jih bo stil XSL uporabil za prikaz.

Razširljiv jezik za stil omogoča:

- filtriranje podatkov v XML-u,
- transformacijo XML-a v drugo obliko XML,
- transformacijo XML-a v HTML,
- naslavljanje določenega dela dokumenta,
- oblikovanje dokumenta XML glede na določeno vrednost,
- sortiranje podatkov v XML-u.

V primerjavi s CSS-om je XSL bolj fleksibilen in uporaben, nastal je iz treh jezikov:

- pot XML-a je jezik za definiranje delov XML-u in vzorcev XML-a (patterns),
- oblikovni objekt XSLF (XSL Formatting Objects) je jezik za oblikovanje izpisa dokumentov XML,
- transformacije razširljivega jezika za stil XSLT so jezik za preoblikovanje – transformiranje dokumentov XML-a.

3.3.3. Transformacija razširljivega jezika za stil – XSLT

Transformacije razširljivega jezika za stil XSLT so del razširljivega jezika za stil XSL. XSLT omogoča prenos podatkov iz ene oblike XML-a v obliko, ki jo prepozna spletni brskalnik – HTML ali pa v drugo obliko XML-a. S transformacijo lahko dokumentu XML odstranujemo obstoječe ali dodajamo nove elemente, sortiramo glede na kriterije, določimo, kateri podatki se bodo

izpisali in kateri ne, ter preurejamo. Vsebinsko se loči tri tipe transformacij (Anderson, 2000, str. 370):

- kreiranje dinamičnih dokumentov – filtriranje, sortiranje ali preurejanje delov dokumenta XML,
- transformacija v predstavitveni jezik – dokument XML se transformira v brezžični označevalni jezik WML (Wireless Markup Language) za predstavitev na mobilnih telefonih ali pa v obliko HTML, ki je primerna za prikaz v spletnih brskalnikih,
- prevajanje – transformacija strukture, kjer se transformira en slovar XML-a v drugega.

3.3.4. Jezik za oblikovanje izpisa XML – XSLF

Transformacije dokumentov XML s pomočjo razširljivega jezika za stil XSL omogočajo veliko bolj prožen način prikazovanja dokumentov XML. V poglavju o transformacijah smo spoznali, kako lahko dokument XML preoblikujemo v drugo obliko XML ali v obliko XHTML za prikaz v spletnih brskalnikih ter WML za prikaz na mobilnih telefonih. Ena od možnosti uporabe razširljivega jezika za stil je, da ga uporabljamo za ustvarjanje drugih oblik predstavitvenih objektov, kot so predstavitve, ki temeljijo na XML-u in jih lahko implementiramo s transformacijami. Za uporabnika, ki nima spletnega brskalnika, ki podpira XML, je najbolj priročna transformacija v HTML, saj mu omogoča, da mu strežnik, ki transformira dokument, pošlje verzijo HTML, ki pa jo brskalnik za internet podpira.

XSL Formatting Objects je jezik za oblikovanje izpisa dokumentov XML. Je specifikacija, ki se ukvarja z oblikovanimi objekti in je del jezika XSL. XSLF omogoča transformacijo dokumenta XSL v obliko, ki ne temelji na osnovi XML, a je primerna za predstavitev podatkov. Vrste transformacij, ki jih podpira XSLF (Adler, 2000):

- pretvorba dokumenta XML v obliko dokumenta z obogatenim tekstom RTF – Rich Text Format,
- pretvorba dokumenta XML v obliko prenosljivega formata dokumenta PDF – Portable Document Format,
- pretvorba dokumenta XML v obliko dokumenta, napisanega z matematičnim besedilom, napisanega v jeziku TeX,

- transformacija dokumenta XML v drevo oblikovanih objektov XSLF – XSL deluje kot zbirka pravil na določenem vzorcu ali predlogi, ki iz dokumenta izbere določene elemente in jih prevede v obliko, kot je navedena v navodilih predloge,
- pretvorba v katero od poljubnih oblik, saj nam razširljivost jezika omogoča poljubno kreiranje glede na pravila, ki jih moramo upoštevati.

3.4. Vmesniki za procesiranje in razčlenjevanje XML-a

Za delo z dokumenti XML imamo na voljo dva vmesnika. Razlikujeta se v lastnostih in načinu delovanja. Kako delujeta, si pogledjmo v naslednjih dveh podpoglavjih.

3.4.1. Objektni model dokumenta – DOM

Objektni model dokumenta DOM – Document Object Model je programski vmesnik, ki omogoča spreminjanje, dodajanje, iskanje, brisanje in branje v XML dokumentih. Objektni model dokumenta je standard, neodvisen od platforme in programskega jezika.

Temelji na drevesni strukturi dokumenta XML. Delovanje vmesnika DOM je enostavno in si sledi v naslednjih fazah:

1. Preberi dokument.
2. Razčleni dokument na posamezne enote (besedilo, komentar, element, atribut, navodilo za procesiranje).
3. V spominu strukturiraj drevo vozlišč iz elementov iz točke 2.

Uporaba vmesnika DOM za izgradnjo in manipulacijo z dokumenti XML ima kar nekaj prednosti. Vmesnik DOM zagotavlja pravilno sintakso. Razvijalec dokumentov XML se ukvarja z drevesno podobo dokumenta in ne več z besedilno predstavitvijo. Ločitev predstavitve od same sintakse XML-a se omogoči z drevesno strukturo, ki predstavlja logični pogled na vsebino – kakšne informacije so v dokumentu in na kakšen način so povezane med seboj. Transformacija dokumenta XML z uporabo DOM-a pomeni spreminjanje drevesne strukture, kar je mnogo bolj enostavno in hitrejše kot delo z datotekami. Prenos informacij iz hierarhičnih in relacijskih baz v XML-u z

uporabo DOM-a je zelo enostaven, saj je način predstavitve povezav med elementi, ki jih podpira DOM, zelo podoben predstavitvam podatkov v omenjenih bazah (Anderson, 2000, str. 156).

DOM prebere celotno vsebino dokumenta XML v spomin in ga tam procesira, zato ni primeren za delo z večjimi oziroma ogromnimi datotekami. Aplikacije DOM-a se lahko uporabljajo na obeh straneh sodelujočih v procesu, ali na strani odjemalca ali pa na strani strežnika. Na strani odjemalcev se DOM uporablja za prikazovanje podatkov XML in za oblikovanje podatkov, ki jih uporabnik vnese na strani odjemalca in se potem prenesejo na stran strežnika. Na strani strežnikov pa se DOM uporablja za arhiviranje informacij in za izmenjavo podatkov med posameznimi notranjimi in zunanji poslovnimi procesi ali med bazami podatkov.

3.4.2. Enostavni vmesnik uporabniškega programa za XML – XML SAX

Za razliko od vmesnika DOM temelji enostavni vmesnik uporabniškega programa za XML SAX (Simple API for XML) na dogodkih. To pomeni, da razčlenjevalnik bere dokument in sporoči programu, ko najde določen simbol - na primer začetek ali konec elementa – določene znakovne podatke, ipd.

SAX ne prebere celotnega dokumenta XML v spomin, zato je primeren za delo tudi z zelo velikimi – ogromnimi datotekami. Poleg tega omogoča, da hranimo podatke v kompleksnih podatkovnih strukturah, ki jih definiramo v svoji aplikaciji, in jih spreminjamo, ko vmesnik sproži določen dogodek. Pogosto potrebujemo iz dokumenta XML samo majhen del informacije, ki jo nosi, na primer število določenih elementov. V tem primeru ni nobene potrebe, da preberemo dokument v celoti, saj SAX omogoča, da ignoriramo podatke, ki nas ne zanimajo. SAX je enostaven za uporabo in pomeni tudi najhitrejši način dostopa do določene informacije iz podatkov XML-a, ki zahteva enkraten zaporeden prehod skozi dokument. To, kar so v določenih primerih prednosti vmesnika SAX, so v drugih njegove slabosti. Ker se dokument ne nahaja v spominu, ni možno naključno dostopati do posameznih njegovih delov, kar postane problematično zlasti pri dokumentih, ki vsebujejo navzkrižne reference. Tudi implementacija kompleksnih iskanj v SAX je zahtevna, saj moramo sami definirati strukture, ki hranijo ustrezne podatke za nadaljnje poizvedbe. Poleg tega SAX ne nudi informacij o strukturi podatkov v DTD-ju

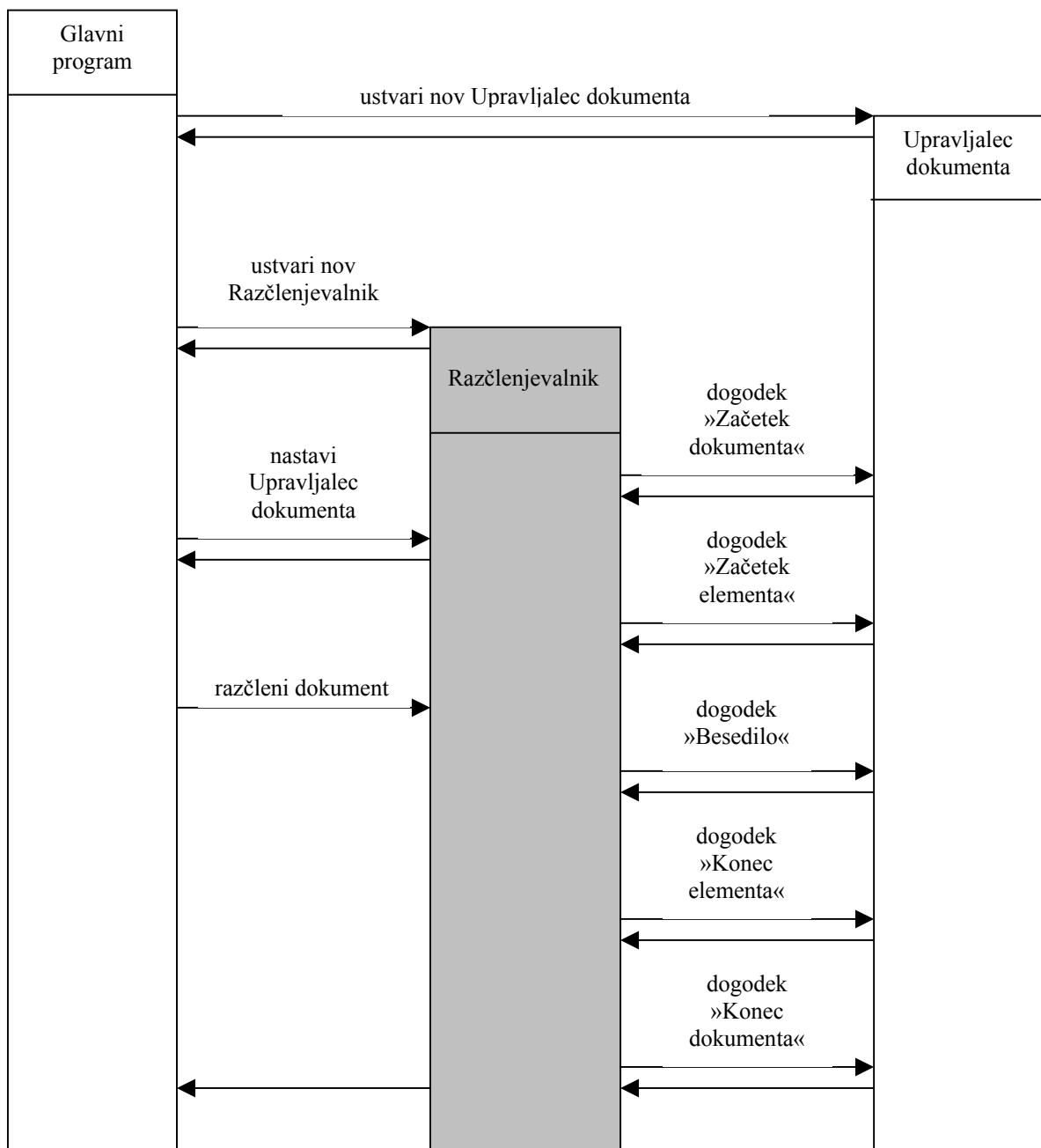
niti ne nekaterih informacij o načinu zapisa podatkov v dokumentu. SAX ni primeren za spreminjanje, nasprotno, pa je zelo primeren za kreiranje in branje dokumentov XML.

Aplikacija, ki jo uporablja SAX, je sestavljena iz razčlenjevalnika, ki je prilagojen standardu SAX in glavnega programa, programske kode, ki procesira vsebino dokumenta in jo imenujemo upravljalec dokumenta (Slika 2 na str. 17). V aplikaciji kreiramo razčlenjevalnik in upravljalec dokumenta, povemo razčlenjevalniku, kateri upravljalec dokumentov naj uporablja in začne procesirati vhodni dokument. Naloga razčlenjevalnika je, da obvesti upravjalca dokumenta na pomembne dogodke znotraj dokumenta. Na zahtevo aplikacije upravljalec procesira določene dogodke in izvede aktivnosti.

Ločimo več tipov dogodkov. Dogodke dokumenta tvorita metodi, ki označujeta njegov začetek in konec, prav tako imamo par metod, ki označujejo začetek in konec elementov. Nadalje vsebuje SAX še metodo, ki vrne vrednost danega atributa ter metode za znakovne podatke, navodila za procesiranje in odkrivanje napak. Napake, ki sprožijo dogodke, delimo v tri skupine: prve so napake pri odpiranju dokumenta in drugih virov, druge so napake, ki jih odkrije razčlenjevalnik in so povezane z dobro oblikovanostjo in veljavnostjo dokumentov, tretje pa so napake, ki jih javi sama aplikacija (Jamnik, 2001, str. 40).

Obravnavanje vseh dogodkov v večjem dokumentu z veliko različnimi elementi pogosto postane nepregledno in nestrukturirano, saj je težko razločevati med posameznimi dogodki, v vsakem odzivu na dogodek pa se izvaja vrsta med seboj neodvisnih aktivnosti. Zato obstajata dve metodi, ki sta namenjeni reševanju kompleksnejših problemov v vmesniku SAX. Prvo imenujemo filter skozi cev, kjer celotno procesiranje razdelimo na več vmesnih stopenj, od katerih ima vsaka svoj vhodni dokument in vrača kot rezultat neki izhodni dokument. Druga metoda temelji na zbirki pravil oblike »Če nastopi ta in ta dogodek, potem izvedi to opravilo.« in je v bistvu nadgradnja programiranja, ki temelji na dogodkih.

Slika 2: Delovanje enostavnega vmesnika uporabniškega programa za XML



Vir: Anderson, 2000, str. 190.

3.4.3. Primerjava DOM-a in SAX-a

Pri izbiri med vmesnikoma DOM in SAX je treba pretehtati prednosti in slabosti, ki jih ima določen vmesnik. Glede na naše potrebe in možnosti, ki nam jih ponujata vmesnika, lahko naredimo izbor iz naslednjih karakteristik.

Prednosti vmesnika SAX:

- hiter,
- enostaven za uporabo,
- primeren za delo z dokumenti poljubne dolžine,
- primeren za delo z majhnim delom podatkov znotraj dokumenta,
- primeren za izgradnjo lastnih podatkovnih struktur.

Slabosti vmesnika SAX:

- ne podpira DTD-ja,
- nedostopnost podatkov o strukturi,
- naključno dostopanje do delov dokumenta ni mogoče,
- spletni brskalniki ga še ne podpirajo,
- namenjen samo za branje,
- implementacija zahtevnih iskanj je otežena.

Prednosti vmesnika DOM:

- enostavna preslikava hierarhične ali relacijske strukture baze podatkov v predstavitev DOM,
- omogoča dobro oblikovanost dokumentov,
- omogoča pravilno sintakso,
- manipulacija z dokumenti XML je poenostavljena,
- sintaksa dokumenta XML se loči od logične predstavitve vsebine istega dokumenta.

Slabosti vmesnika DOM:

- neprimeren za delo z velikimi dokumenti.

3.5. XML in prenos podatkov

Prenos dokumentov XML med strankami in ponudniki imenujemo XML v elektronskem poslovanju. Dokument, ki se prenaša med poslovnimi partnerji, se imenuje sporočilo. Ponavadi je sporočilo sestavljeno iz glave (header), ki vsebuje podatke za nadzor stanja in prenosa sporočila, in telesa (body), ki vsebuje dejanske podatke. Za dodajanje dodatnih informacij sporočilom nam XML ponuja standardno sintakso za označevanje podatkov. S temi informacijami lahko določamo poslovna pravila (Anderson, 2000, str. 587).

Eden izmed najpomembnejših vidikov komunikacije med trgovinskimi partnerji je, kako se sporočila sklicujejo na lokalne vire in informacije. Ti podatki niso shranjeni v sporočilu, so pa nujno potrebni za aplikacije, ki obdelujejo sporočila.

Podatki za nadzor prenašanja sporočil se nahajajo v glavi, ki pa vsebuje še naslednje dodatne informacije:

- varnost,
- serijsko obvarovanje sporočil,
- identifikacija transakcij,
- sledenje informacij, povezanih s sporočili,
- usmerjanje sporočil,
- sporočanje o napakah.

Ti podatki v nekaterih aplikacijah niso shranjeni v obliki XML. V spletnih aplikacijah so ti podatki lahko shranjeni v glavi sporočila HTTP, ki se uporablja kot protokol za prenos podatkov.

V glavi dokumenta je definiran cilj, kamor morajo prispeti podatki, zapisani v telesu sporočila. Telo sporočila je glavni del vsebine sporočila in lahko vsebuje podatke za prodajo, naročilo, potrditev naročila, račun, ipd. Slovarje uporabljamo za označevanje podatkov, ki so navedeni v telesu sporočila. Od vsebine sporočila je odvisno, kateri slovar se bo uporabljal. V primeru da uporabniku ne ustreza noben obstoječi slovar, ga lahko napiše sam za svoje lastne potrebe (W3C, 2000).

V gospodarstvu narašča število novih standardov za različne panoge, ki temeljijo na XML-u, zato je zelo verjetno, da že obstaja standard za rešitev

problema uporabnika oziroma obstaja vsaj standard, ki bi ga lahko uporabnik v rešitvi vzel za osnovo in nato razširil. Težava pa je najti ustrezn standard. Najprej se lahko uporabnik obrne na svojega partnerja, s katerim bo sodeloval in mogoče že uporablja katero od obstoječih shem. Druga možnost pa je uporaba interneta, saj na internetu že obstajajo tako imenovana skladišča XML-a na strežnikih organizacij in združenj, ki se ukvarjajo s tem. Za dostop do skladišča uporabljamo aplikacijo organizacije, na strani katere se nahajajo podatki, ki jih potrebujemo. To je na primer orodje Biztalk, ki ga proizvaja podjetje Microsoft. Skladišča vsebujejo različne definicije tipa dokumenta in sheme za obstoječe slovarje. Za svoje transakcije lahko podjetje v celoti uporabi obstoječe sheme in s tem prihrani veliko časa za razvoj lastnega slovarja. V primeru da obstoječa shema ne zadovoljuje vseh potreb uporabnika, ampak njihov večji del, jo lahko vzame za osnovo svojega slovarja. Pri tem se lahko podvoji obstoječa shema in se nadgradi. Druga možnost pa je kombinirana uporaba obstoječega in lastnega slovarja z uporabo imenskega prostora. Le malo slovarjev je globalnih, ampak so večinoma regionalno omejeni, kar pomeni, da upoštevajo zgolj lokalni jezik, davčno politiko, sistem zaščite osebnih podatkov, poslovno in pravno ureditev. Težava, ki se pojavlja pri množici slovarjev, ki jih razvijajo različne ustanove in organizacije, je v tem, da obstajajo različne aplikacije XML-a za iste panoge oziroma za podobne ali enake tipe elektronske komunikacije.

O XML-u v elektronskem poslovanju se pojavlja mnogo napačnih predstav o njegovi vlogi in pomenu. Pogoste razlage napeljujejo na to, da je za jezik XML v elektronskem poslovanju bistvenega pomena njegova enostavnost, možnosti prenosa preko omrežja in prikazovanja v brskalniku. Prav tako je zmotno prepričanje, da je po dogovoru dveh strank glede sheme možno izmenjavati katere koli dokumente, ki zadoščajo tej shemi. V resnici je med partnerjema, ki bosta izmenjevala podatke, potrebnih še veliko drugih dogovorov, ki opredeljujejo izmenjavo glede na njun odnos in način poslovanja. Te dodatne zahteve pogosto niso del standarda, ampak so specifična nadgradnja, ki omogoča točno določeno komunikacijo med konkretnima partnerjema. Seveda pa je v tem primeru pomembno, da je XML standard, ki je razširljiv in omogoča take nadgradnje (Jamnik, 2001, str. 66).

Pri razvoju v XML se srečujemo z dvema vmesnikoma – XML SAX in DOM. Vmesnik objektnega modela dokumenta je zelo enostaven, zato mnogi razvijalci v XML avtomatično predpostavljajo, da je najbolj primeren za delo z dokumenti XML v elektronskem poslovanju. Pri načrtovanju tovrstnih rešitev

moramo upoštevati velikost dokumentov, ki se izmenjujejo. Čeprav je DOM primeren za kratka sporočila in dokumente, pa lahko pri velikih dokumentih pomeni hudo obremenitev za sistemske vire, saj naloži celotno vsebino dokumenta v spomin. V takih primerih pa je bolj smotrno uporabiti enostavni vmesnik za XML SAX.

Standardni razčlenjevalniki preverjajo veljavnost dokumentov, ki se izmenjujejo med aplikacijami. Veljavnost dokumentov pa še ne zagotavlja integritete sistema, kar pomeni, da ni nujno, da se bo sporočilo, ki je prispelo v aplikacijo, uspešno procesiralo. Torej, veljavnost dokumenta XML zagotavlja, da je dokument dobro oblikovan in da so podatki strukturirani v skladu z definicijo tipa dokumenta ali shemo, kar pomeni, da sta slovar in slovnica korektna in da dokument vsebuje zahtevane elemente. Prav tako pa mora aplikacija preveriti veljavnost oziroma ustreznost podatkov v elementih glede na poslovna pravila, ki pripadajo vsebini. Mnogokrat se pripeti, da je treba sporočilo prevesti še v drugo obliko, ki jo razume aplikacija, ki bo sporočilo obravnavala. V takih primerih se pojavi dodatna potreba po preverjanju ustreznosti podatkov. Pri preverjanju pa naletimo na problem, saj zahtev, ki jih narekuje poslovna logika, ne morejo preverjati standardni razčlenjevalniki, saj so te zahteve specifične in odvisne od narave poslovnih partnerjev, ki sodelujejo. Rešitev tega problema je preverjanje ustreznosti sporočil, ki se izvaja na aplikativnem nivoju oz. lahko tudi na vmesnem nivoju, ki popolnoma preveri ustreznost prihajajočih sporočil in obravnava še preverjanje z vidika varnosti ter jih posreduje aplikaciji, če se ugotovi, da dokument ustreza vsem določenim zahtevam.

3.6. *Izmenjavanje dokumentov XML*

Tehnologija podatkovnih baz je temelj standarda XML. Prednost jezika XML, kot standarda za izmenjavo podatkov na internetu, je razvidna predvsem takrat, ko je možno generirati dokumente XML neposredno iz baz podatkov, izmenjevati informacije iz različnih podatkovnih virov ter ustvarjati dinamične dokumente XML.

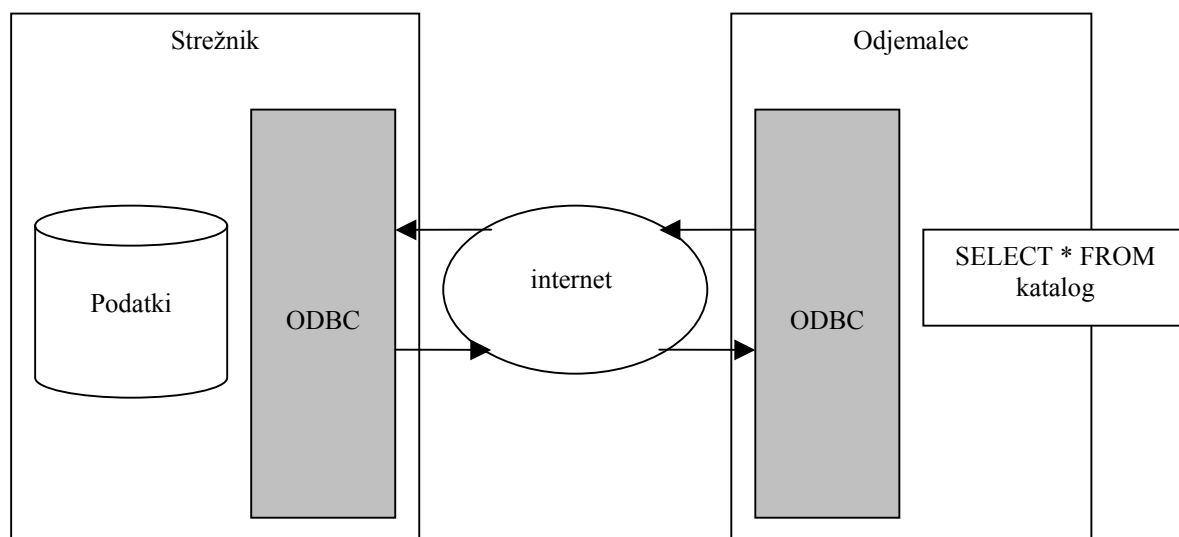
Jezik XML omogoča poizvedovanje po informacijah iz različnih sistemov, prinaša spremembe v komunikaciji med odjemalci in strežniki v dvo- ali

večnivojski tehnologiji sistemov in predstavlja standard za izmenjavo informacij med različnimi sistemi.

Večina podatkov je shranjena v takih in drugačnih sistemih v tej ali oni obliki, zato morajo obstajati načini, kako te podatke predstaviti kot XML. Poleg tega pa dokument XML ni nujno statičen, ampak se zgradi dinamično takrat, ko se potrebuje, zlasti ko gre za spletne aplikacije (Pucihar, 1999, str. 9).

Dokument XML lahko tvorimo tako, da podatke iz relacijske ali objektne podatkovne baze programsko pretvorimo v želeno obliko. Tak dokument lahko prenesemo v kateri koli drug sistem, ki podpira XML. Izmenjava podatkov med sistemi je v preteklosti potekala s pomočjo različnih tehnologij, kot je na primer vmesnik za odprto povezovanje podatkovnih baz ODBC – Open Database Connectivity, prikazan na Sliki 3.

Slika 3: Shema izmenjave podatkov med sistemi z uporabo ODBC

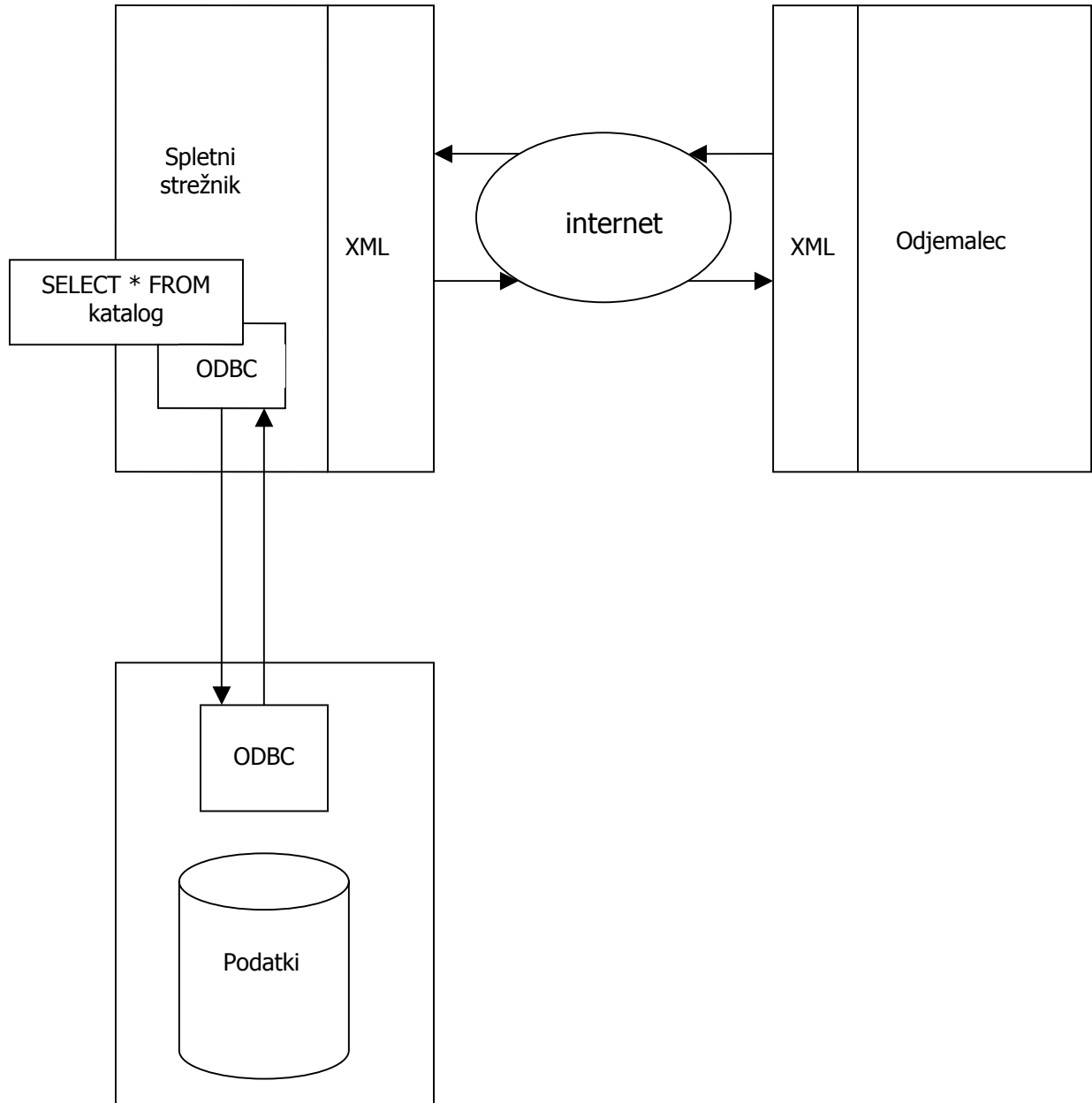


Vir: Anderson, 2000, str. 449.

Pomanjkljivosti tega načina izmenjave informacij so: izmenjava podatkov ne tovrsten način lahko poteka le med sistemi, ki podpirajo in razumejo ODBC, transakcije ODBC so občutljive zaradi možnosti nepooblaščenih vdorov, zato jih mnogi požarni zidovi ne prepuščajo. Z uvedbo XML-a pa lahko zagotovimo, da se podatki prenašajo preko vrat 80, ki jih uporablja standardni internetni protokol za prenos hiperteksta HTTP – Hypertext Transfer Protocol in se s tem izognemo blokiranju komunikacije s strani požarnih zidov. Z uporabo

XML-a dosežemo, da odjemalec ni več odvisen od ODBC in s tem povečamo fleksibilnost in uporabnost sistema (Slika 4).

Slika 4: Shema izmenjave podatkov med sistemi z uporabo XML-a



Vir: Anderson, 2000, str. 450.

Za zajemanje podatkov iz podatkovne baze obstajata dva načina. Celotne podatke v dani bazi lahko pretvorimo v XML in nato poizvedujemo po pretvorjenem dokumentu XML. Lahko pa naredimo poizvedbo že v bazi sami in pretvorimo v XML le tiste podatke, ki jih potrebujemo. Z uporabo druge

metode lahko zmanjšamo promet po omrežju in čas za odgovor, vendar je ta način poizvedovanja še vedno odvisen od konkretnega sistema za upravljanje podatkovnih baz. Objektni model dokumenta DOM, ki je opisan v razdelku 3.4.1 lahko uporabimo za prenos podatkov iz relacijske baze v XML. DOM namreč poskrbi za dobro oblikovanost novo nastalega dokumenta. Poizvedbe po prebranih podatkih v objektnem modelu, lahko izvajamo tudi s pomočjo poti XML in transformacij. Vendar je tudi ta način odpove pri večjih količinah podatkov, zato je bolje preslikati izraze iz specifikacije pot XML v poizvedbe, zapisane v strukturiranem poizvedovalnem jeziku SQL (Structured Query Language). Ta jezik se uporablja v relacijskih bazah podatkov in na ta način je možno omejiti množico vozlišč, ki jih zajamemo v objektni model dokumenta (Harold, 1999, str 356).

3.7. XML in varnost podatkov

Poslovni svet uporablja na različnih področjih tehnologije varnosti in zaščite. S šifriranjem elektronskih sporočil, prometa po omrežju ali konferenčnih sej varujemo podatke pred prisluškovanjem ali vdorom nepooblaščenih oseb. S posebnimi metodami lahko tudi preprečimo, da bi kupec zanikal svoje elektronsko naročilo ali pa podpisujemo elektronske pogodbe. Identifikacija in avtorizacija uporabnikov ali obiskovalcev spletnih strani omogoča posameznikom dostop do izbranih vsebin, hkrati pa lahko strankam omogočimo, da preverijo verodostojnost naše strani in podatkov na njej.

Po napovedih strokovnjakov bo XML v prihodnosti zagotovo postal zelo razširjen standard za izmenjavo podatkov med sistemi na internetu in v elektronskem poslovanju, zato je vidik varnosti in zaščite podatkov v XML-u zelo pomemben. Metode varovanja virov so zelo različne in so odvisne od zahtevane stopnje zaščite in strukture sistema. Tako lahko posamezne dele sistemov varujemo s fizičnimi ukrepi, medtem ko v poslovanju med različnimi omrežji najpogosteje uporabljamo kriptografijo in iz nje izpeljane storitve zaščite, kot je na primer digitalni podpis. Varnostne storitve v elektronskem poslovanju obsegajo overjanje identitete, nadzor dostopa, preprečevanje zanikanja sporočil in aktivnosti, omogočanje stalne razpoložljivosti ter zagotavljanje zaupnosti in neokrnjenosti informacij (Jerman Blažič, 2001, str. 101).

3.7.1. Javni ključi in šifriranje podatkov

Šifriranje ali kriptografija je veda o zakrivanju sporočil. Šifrirni algoritmi transformirajo berljive podatke v tajne, pri obratnem procesu pa tajne podatke pretvorijo v čitljive. Šifrirni algoritmi uporabljajo šifrirne ključe, ki določajo njihovo delovanje. Če je ključ za šifriranje in dešifriranje enak, imenujemo postopek šifriranja simetrični kriptosistem. Stranke, ki želijo izmenjavati podatke s pomočjo simetrične metode, si morajo pred uporabo varno izmenjati ključ, kar pa v javnih omrežjih postane problem, saj pogosto komuniciramo z osebami, ki so fizično zelo oddaljene. Druga težava simetrične kriptografije je veliko število ključev, saj potrebujemo po en ključ za vsak subjekt, s katerim komuniciramo (Jerman Blažič, 2001, str. 102).

Asimetrični kriptosistem je drugi pristop, pri katerem sta ključa za šifriranje in dešifriranje različna. Ključ za šifriranje je javni ključ, in ga ne moremo uporabiti za dešifriranje. Drug ključ, ki ga uporabimo za dešifriranje, imenujemo privatni ali zasebni ključ, s katerim pretvorimo šifrirano sporočilo nazaj v berljivo besedilo. Privatni in javni ključ povezujejo matematični algoritmi, seveda pa potrebujemo še dodatne informacije, da lahko iz enega ključa določimo drugega. Z objavo javnega ključa omogočimo drugim, da nam pošiljajo šifrirana sporočila, ta pa jih lahko preberemo samo sami, saj imamo samo mi sami zasebni dešifrirni ključ. Asimetrični kriptosistemi imajo pomanjkljivosti predvsem pri overjanju javnih ključev ter hitrosti šifriranja in dešifriranja. Asimetričnim sistem kriptiranja zahteva veliko tehnologij, ki jih moramo uporabljati, da zagotovimo varnost in zaščito podatkov v elektronskem poslovanju. V ta sklop sodijo porazdeljeno preverjanje avtentičnosti, dogovor o tajnem ključu preko javnega ključa, šifriranje podatkov brez predhodne izmenjave tajnih podatkov in digitalni podpis.

Za preverjanje identifikacije podpisnika in integriteto sporočila, to pomeni, ali je sporočilo na poti ostalo nespremenjeno uporabljamo digitalni podpis. To je informacija, ki jo dobimo z matematično transformacijo na podatkih, ki jih želimo podpisati, s pomočjo privatnega ključa pošiljatelja. Digitalni podpis lahko kreira samo oseba, ki poseduje zasebni ključ. Vsakdo, ki ima njen javni ključ, lahko preveri digitalni podpis in ugotovi njegovo pristnost. Vsaka sprememba podatkov, ki jih podpišemo, spremeni tudi digitalni podpis, ki tako postane neveljaven.

Porazdeljeno preverjanje avtentičnosti s pomočjo javnih ključev zagotavlja, da je pošiljatelj podatkov res tisti, za katerega prejemnik misli, da je. Preverjanje poteka na naslednji način. Oseba A želi preveriti, ali je podatke, ki jih je dobila od osebe B, res poslala oseba B. Zato pošlje osebi B tako imenovani poziv, ki je šifriran z javnim ključem osebe B. Oseba B dešifrira poziv in ga pošlje nazaj osebi A in s tem dokaže, da poseduje privatni ključ, ki ustreza javnemu ključu, s katerim je bil poziv šifriran. Oseba A lahko pošlje osebi B tudi nešifriran poziv, ki ga potem oseba B poveže z drugimi informacijami, ki jih podpiše z digitalnim podpisom. Oseba A nato uporabi javni ključ osebe B, da preveri njen podpis in dokaže, da oseba B res poseduje ustrezni privatni ključ (Jamnik, 2001, str. 55).

3.8. Primerjava XML-a in RIP-a

RIP in XML imata v elektronskem poslovanju enako vlogo, vendar ima XML v primerjavi s standardom RIP kar nekaj prednosti, ki so predstavljene v Tabeli 2.

Jezik XML in rešitve, ki temeljijo na njegovih osnovah, so dostopne tudi manjšim podjetjem, saj za elektronsko poslovanje ne potrebujejo omrežij z dodano vrednostjo in specifične programske opreme, kot v primeru uporabe standarda RIP, ampak potrebujejo le internet in osebni računalnik z aplikacijo, ki procesira XML sintakso.

Slovarji v XML-u podpirajo enako obliko uporabe podatkov tako za interno predstavitev podatkov kot tudi za izmenjavo z zunanjimi partnerji. Nasprotno pa je treba pri standardu RIP interno obliko podatkov prevesti v obliko, ki bo primerna za izmenjavo z zunanjimi partnerji.

XML sintaksa je enostavna in omogoča večji nadzor nad aplikacijami. Sintaksa pri standardu XML pa je izjemno zapletena.

Razvoj rešitev v RIP in v XML zahteva določen čas in investicije, vendar pa je razvoj v XML bistveno cenejši zaradi bistveno večje razširjenosti in možnosti uporabe znotraj in izven podjetja. Razvijalci in proizvajalci programske opreme so ugotovili, da dandanes lahko z uporabo XML-a svojim naročnikom omogočijo povezljivost sistemov in možnost sodelovanja med različnimi

sistemi. Trg uporabnikov standarda XML je velik, zato je na trgu tudi veliko število aplikacij in orodij, ki uporabnikom omogočajo bolj enostaven in hitrejši razvoj končnih rešitev.

Tabela 2: Primerjava XML-a in RIP-a

	XML	RIP
<i>dostopnost</i>	dostopen tudi majhnim podjetjem	dostopen samo velikim podjetjem in ustanovam
<i>predstavitev podatkov</i>	enaka oblika predstavitve podatkov za zunanjo komunikacijo in interno uporabo	prevajanje interne predstavitve podatkov v predstavitev za izmenjavo s partnerji
<i>sintaksa</i>	enostavna sintaksa	zapletena sintaksa
<i>berljivost</i>	lahko berljiv	nečitljiv
<i>uporabnost</i>	široka uporaba	ozka uporaba
<i>nadgradnja</i>	enostavnost pri razširitvi	dolgotrajno in zapleteno nadgrajevanje
<i>odvisnost od informacijske tehnologije</i>	neodvisen od strojne in programske opreme	odvisen od strojne in programske opreme
<i>programerji</i>	veliko programerjev za podporo	poznavanje omejeno na ozek krog programerjev
<i>vmesniki</i>	podpora v aplikacijah samih	različni vmesniki za komunikacijo s posameznimi partnerji
<i>omrežje</i>	internet	omrežja z dodano vrednostjo (VAN)
<i>novi standardi</i>	razmeroma enostavna nadgradnja	dolgotrajen in drag proces nadgradnje

Vir: Jamnik, 2001, str. 69.

XML vsebuje podporo za standardne varnostne mehanizme, ki zagotavljajo avtentifikacijo sporočil, digitalni podpis in integriteto.

Tehnologijo je obvladoval ozek krog specialistov. Pri tehnologiji XML to ni več tako, saj je krog strokovnjakov, ki se ukvarja s tem, iz dneva v dan večji, to pa predvsem zato, ker je tehnologija enostavna, lahko berljiva in ne preveč kompleksna. O XML jeziku in rešitvah, ki temeljijo na njegovih osnovah, je napisanih veliko knjig, veliko gradiva je na internetu in je tema mnogih simpozijev in seminarjev. Podjetje, ki se odloči za to novo tehnologijo, ima na

voljo dovolj strokovnjakov, ki imajo znanja in sposobnost za nadgradnjo in vzdrževanje sistema elektronskega poslovanja.

3.9. Nadgradnja standarda XML

Razvoj standardov za elektronsko poslovanje je pripeljal do razvoja standarda ebXML ali Electronic Business XML, ki je ena izmed najobetavnejših rešitev za elektronsko poslovanje. Standard ebXML definira uporabo XML za namene elektronskega poslovanja. Vendar XML ni edini standard, ki ga XML zavzema, saj je sestavljen iz več standardov in protokolov, ki jih med seboj dopolnjuje in povezuje, naj omenim le kombinacijo protokolov za prenos HTTP (Hyper Text Transfer Protocol), SMTP (Simple Mail Transfer Protocol), ki ju srečujemo vsak dan na internetu, in XML za vsebino dokumentov (Mivšek, 2003, str. 165).

Poleg tehnoloških prednosti je prisoten tudi vsebinski korak, saj ebXML omogoča povezovanje poslovnih procesov. EbXML je prvi standard, ki omogoča opis in izvajanje poslovnih procesov na elektronski način, neodvisno od tehnologije, ki se uporablja.

Standard ebXML je sestavljen iz več delov, ki zajemajo naslednja področja (Mivšek, 2003, str. 166):

- poslovna sporočila,
- poslovni procesi,
- sporazum o sodelovanju,
- profil sodelovanja,
- register,
- sporočilni sistem.

Poslovna sporočila so v sistemu ebXML običajno v XML obliki, čeprav je standard ebXML narejen za pošiljanje katere koli vrste sporočil. Poslovne procese definiramo z XML jezikom, s katerim opišemo poslovni proces kot množico stanj, prehode med njimi ter poslovne akcije pa definiramo kot posledice poslovnih dohodkov. V profilu sodelovanja so natančno določeni skupni poslovni procesi, parametri sporočilnega sistema in formati dokumentov. V profilu sodelovanja podjetje zapiše podatke o svojih poslovnih procesih, kakšne parametre ima sporočilni sistem in katere formate podpira.

Register podjetja je neke vrste oglas, v katerem podjetje objavi svoje profile sodelovanja in do katerega lahko dostopajo druga podjetja in tako začnejo sodelovati. Sporočilni sistem je namenjen za elektronski prenos poslovnih sporočil med poslovnimi partnerji.

4. POSLOVNI SISTEM MERCATOR D.D.

4.1. Predstavitev

Skupina Mercator je ena največjih in najuspešnejših gospodarskih družb v JV Evropi, ki deluje že 54 let. Skupino Mercator sestavlja 19 gospodarskih družb, in sicer 12 trgovskih družb, od tega 4 izven Slovenije, 4 živilskopredelovalne, 1 hotelsko-gostinska ter po ena kmetijska in inženiring družba. Mercatorjeve družbe zaposlujejo skupaj več kot 14.000 ljudi. Skupina Mercator je vodilna trgovska veriga na slovenskem trgu, poleg tega pa postaja tretji največji trgovec s prehrabnenimi izdelki na hrvaškem trgu in pomemben trgovec na tržišču BiH, v Srbiji in Črni gori.

4.2. Dejavnost

Najpomembnejša in najobsežnejša gospodarska dejavnost Skupine Mercator je trgovina na debelo in drobno z izdelki široke potrošnje. V svojem sestavu ima 1.054 maloprodajnih enot različnih tipov, ki zajemajo 542.474 kvadratnih metrov skupnih površin, od nakupovalnih centrov, hipermarketov, supermarketov, blagovnic, do samopostrežnih in specializiranih prodajaln. Mercatorjev tržni delež v slovenski trgovini znaša 39,8 odstotkov brez franšiznih prodajaln in mu zagotavlja prvo mesto v dejavnosti trgovine z živili. Povprečno v Mercatorjevih prodajalnah mesečno nakupuje 10 milijonov kupcev.

V Sloveniji je skupina Mercator največji kupec, kar se izraža v odnosih s končnimi kupci, dobavitelji in proizvajalci. Za zagotovitev boljšega poslovanja in boljših poslovnih rezultatov je treba skupaj s poslovnimi partnerji optimizirati procese v celotni oskrbovalni verigi. Eno izmed orodij, ki se lahko uporabijo za doseg tega cilja, je računalniška izmenjava podatkov RIP, ki v

tesnem sodelovanju s poslovnimi partnerji omogoča zniževanje stroškov v celotni oskrbovalni verigi. Posledica tega je učinkovito poslovanje s številnimi ugodnostmi za končne kupce v Mercatorju.

4.3. Uvedba RIP-a

Januarja leta 2000 so bila izoblikovana osnovna izhodišča za uvedbo RIP-a v podjetju Mercator. Izbrali so standard EANCOM, ki temelji na mednarodno uveljavljenem standardu UN/EDIFACT. Z vpeljevanjem mednarodnih standardov, sodobnih telekomunikacijskih sredstev in novih načinov poslovanja so s pomočjo RIP-a vplivali na takratni že obstoječi proces nabave blaga.

Začetek RIP-a v Mercatorju predstavlja vzpostavitev RIP sistema za izmenjavo dokumenta »naročilo« med poslovnim partnerjem Orbico, ki je dobavitelj za izdelke proizvajalca Proctor & Gamble ter distribucijskim centrom (DC) Ljubljana. Osnovni elementi dokumenta »naročilo« so EAN kode artiklov in EAN lokacijske številke pošiljatelja ter prejemnika, opisi artiklov in vključenih partnerjev pa so uporabljeni le za možnost kontrole. Sprva je bila izvedena povezava RIP sistema z rešitvijo za blagovno poslovanje v DC Ljubljana, kasneje pa je bila še izdelana rešitev za prevajanje in prenos podatkov.

Posredovani podatki poslovnemu partnerju o prodaji omogočajo pravočasno in učinkovito planiranje proizvodnje, zato je bil v nadaljevanju projekta s proizvajalcem Proctor & Gamble vpeljan tudi dokument »poročilo o zalogah«, ki vsebuje dnevno stanje zalog in podatke o prodaji proizvajalčevih artiklov v opazovanem distribucijskem centru.

Postopoma so bili v sistem RIP vključeni vsi distribucijski centri skupine Mercator in 16 dobaviteljev. Izmenjani dokumenti predstavljajo približno desetino vseh naročil, posredovanih dobaviteljem. Prav tako pa je bil RIP sistem prirejen na način, ki omogoča tudi sprejemanje in potrjevanje naročil kupcev.

Uvedba RIP-a ima tako prednosti kot slabosti. Treba je predvideti stroške programske opreme, vzdrževanja, dela, druge opreme in samega delovanja RIP sistema. Prednost uporabe RIP-a je v tem, da za operativno naročanje

niso več potrebni obiski trgovskih potnikov dobaviteljev. Naročila se hkrati izpisujejo in prenašajo v prevzemnem oddelku distribucijskega centra k poslovnemu partnerju. To omogoča krajši cikel od naročila blaga do dobave naročenega blaga in s tem tudi hitrejši odziv dobavitelja. Možnost napak pri obeh poslovnih partnerjih pa odpravljajo usklajene EAN kode artiklov in vzpostavljen postopek rednega obveščanja ter osveževanja podatkov o artiklih.

Naloge, ki so bile predvidene v nadaljevanju projekta, lahko razdelimo na projektne in operativne naloge. Projektna naloga je uvedba novih dokumentov. Med operativne naloge štejemo: pridobivanje novih dobaviteljev za naročanje preko RIP-a, uvajanje naročanja s pomočjo RIP-a v maloprodajnih enotah za dobavitelje, ki dobavljajo neposredno in operativno vzdrževanje sistema.

Naslednji uvedeni dokument je bil dokument »potrditev naročila«. Z vpeljavo tega dokumenta v elektronsko poslovanje se je nabavnemu referentu omogočilo sproten nadzor nad naslednjimi stvarmi: odstopanja od naročenih količin, naročanje nedobavljenega blaga pri alternativnem dobavitelju in kontrola nad predvidenim časom dobave. Dokument »potrditev naročila« omogoča tudi vnaprejšnje planiranje aktivnosti na prevzemu blaga, kar pa pomeni tudi odpravo konic v procesu prevzema in s tem zmanjšanje čakalnih vrst.

Mercator sodeluje v okviru projekta e-SLOG, ki poteka na Gospodarski zbornici Slovenije. Skupaj z ostalimi podjetji, ki sodelujejo v projektu, bodo skušali definirati elektronski dokument »račun«. Vsebina dokumenta bo zasnovana na osnovi EANCOM standarda, izmenjavanje dokumentov pa bo mogoče tako v standardni obliki XML kot tudi EANCOM.

Uporaba dokumenta »račun« omogoča kreiranje zapisov v finančno računovodskem informacijskem sistemu ter avtomatsko primerjavo podatkov, vsebovanih na računu, s podatki, ki so bili zabeleženi ob prevzemu blaga.

Ažurna podatkovna baza o artiklih, ki so predmet poslovanja, je osnova za izmenjavanje navedenih dokumentov. Tovrstni podatki se trenutno usklajujejo z vsakim dobaviteljem posebej, kar predstavlja problem. Izgradnja e-kataloga artiklov na nacionalni bazi bi rešila ta problem. Katalog artiklov pa bi lahko povezali s tovrstnim katalogom artiklov, ki obstajajo v svetu, na primer češki

ali nemški katalog artiklov. Združenje Global Commerce Initiative – GCI proučuje povezovanje nacionalnih katalogov. GCI pa prav tako pripravlja klasifikacijo potrebnih podatkov in standardizacijo postopkov prenosa podatkov ter kontrole le-teh.

Pri izgradnji in uvajanju RIP sistema v Mercatorju sta bila ključnega pomena aktivno sodelovanje dobaviteljev in podpora vodstva podjetja. V podjetju Mercator je vzpostavljeni RIP sistem zanesljiv in primeren za veliko število poslovnih partnerjev, sprejemljiv pa je tudi za manjše poslovne partnerje, ki imajo manj človeških in materialnih virov.

EAN Slovenija je institucija, ki skrbi za globalno sodelovanje na osnovi mednarodno sprejetih vodilnih standardov v svetu. Zagotavlja uspešnost omenjenega sodelovanja poslovnih partnerjev na nacionalnem nivoju. Uvajanje standardiziranih dokumentov, ki so prirejeni za posamezno panogo ali nacionalni nivo, zagotavlja hitrejše in poenostavljeno vpeljevanje računalniške izmenjave podatkov.

V Sloveniji omenjeno povezovanje poslovnih partnerjev z računalniškim izmenjavanjem podatkov še ni dovolj razširjeno. To predstavlja priložnost in izziv podjetjem za pospešeno uvedbo tovrstnih rešitev in izkoriščanje prednosti, ki jih tesno sodelovanje partnerjev omogoča. Ena izmed rešitev je tudi sistem RIP na podlagi XML standarda in nadgradnja v standard, kot je ebXML.

4.4. Smotrnost uvedbe XML-a V PS Mercator d.d.

Slovenija je iz dneva v dan bliže evropskim integracijam, z vstopom v Evropsko unijo pa bomo člani močne evropske zveze. Prav tako pa mora tudi slovensko gospodarstvo spremeniti mišljenje in se pripraviti na vstop v Evropsko unijo. Začetna faza vstopa bo za nepripravljene kaj lahko imela zelo neugodne posledice. Poslovni sistem Mercator pa ima možnost, da se ob vstopu v Evropsko unijo približa sorodnim gospodarskim subjektom in se jim tudi konkurenčno postavi po robu. Ena izmed stvari, ki jih Mercator lahko naredi v tej smeri, je sodobno elektronsko poslovanje oz. novejši standard za računalniško izmenjavo podatkov.

Kot sem že omenil v predstavitvi obstoječega stanja elektronskega poslovanja v PS Mercator, ugotavljamo, da trenutno uporabljajo standard za RIP EANCOM, ki temelji na mednarodno uveljavljenem standardu UN/EDIFACT. Osredotočil se bom na poslovni proces veleprodajnega poslovanja in analiziral, ali so kakršne koli prednosti, ki bi jih prinesla uvedba standarda XML. Prav tako pa si bomo ogledali tudi morebitne slabosti, na koncu pa bom podal skupno oceno o smotrnosti uvedbe standarda XML v PS Mercator d.d..

4.4.1. Podproces prihod blaga v skladišče

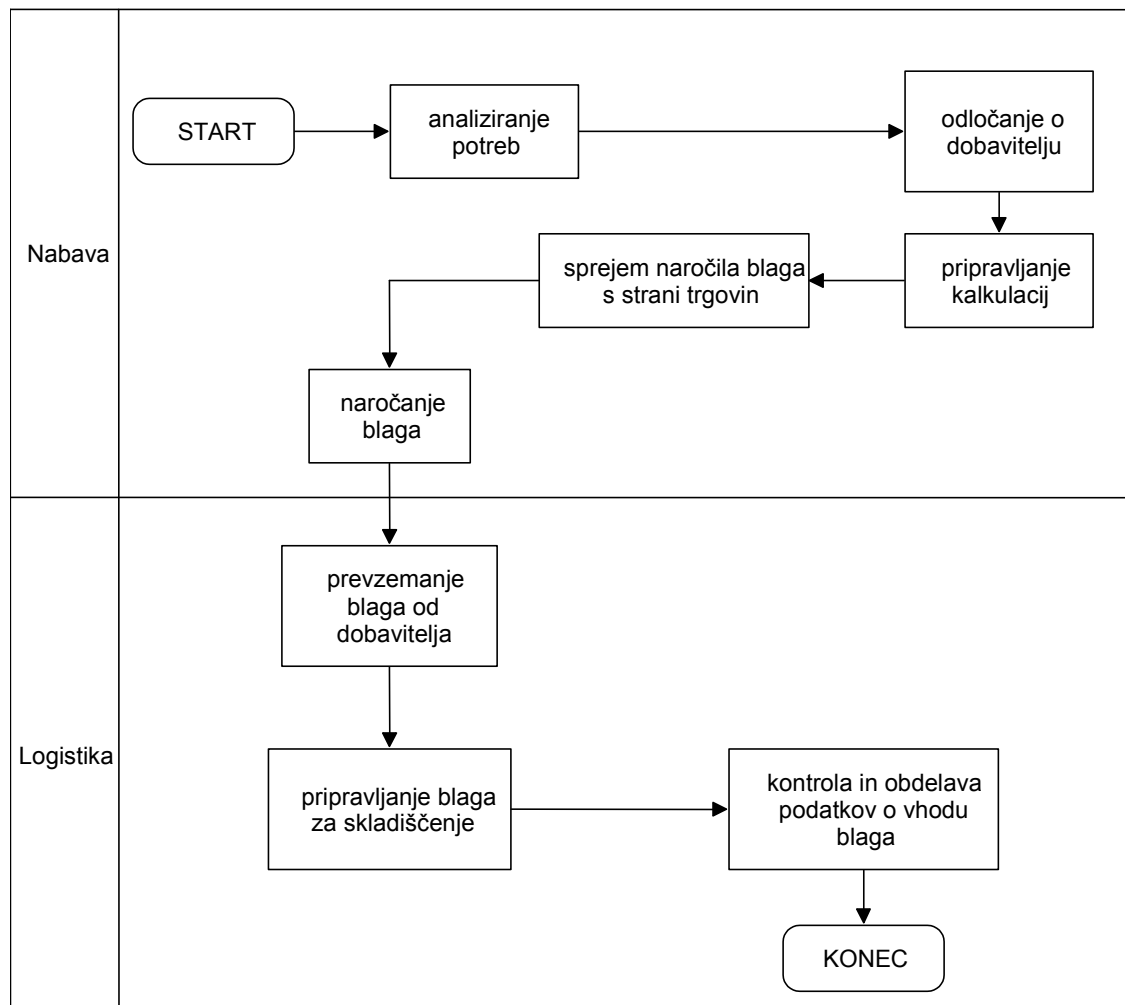
Podproces prihod blaga v skladišče je prvi del veleprodajnega procesa v Mercatorju. S prihodom blaga v skladišče zagotavljajo nemoteno preskrbo lastnih trgovin, ki se zalagajo z artikli iz centralnega distribucijskega centra v Ljubljani.

Kako poteka podproces prihod blaga v skladišče lahko vidimo na Sliki 5. Model sem narisal na podlagi interne dokumentacije Mercatorja in pogovorov z zaposlenimi, ki so zadolženi za ta proces.

Proces sestavlja 8 aktivnosti, ki si sledijo v naslednjem vrstnem redu: sprejem naročila blaga s strani trgovin, analiziranje potreb, odločanje o dobavitelju, pripravljane kalkulacij, naročanje blaga, prevzemanje blaga od dobavitelja, priprava blaga za skladiščenje ter kontrola in obdelava podatkov o vходу blaga.

Mercator ima politiko, da od določenega dobavitelja dobavlja samo določene artikle. Z dobavitelji imajo sklenjene pogodbe o tem, katere izdelke bodo kupovali. Odločanje o dobavitelju pomeni pregled pogodb in na podlagi le-teh izločitev dobaviteljev, ki dobavljajo izdelek, ki ga trenutno potrebujejo. Naslednji korak je izdelava kalkulacij, kjer s pomočjo aplikacije naredijo kalkulacijo pri katerem dobavitelju bodo naročili blago. Proces naročanja se prične, ko v distribucijski center iz trgovin sporočijo svoje potrebe po novih artiklih. Vsa ta naročila se zberejo in naredi se analiza potreb, glede na to kakšna je trenutna zaloga v skladišču. V primeru da v nabavnem oddelku ugotovijo, da trenutno nekega izdelka ni na zalogi, sprožijo postopek nabave blaga od dobavitelja.

Slika 5: Prihod blaga v skladišče



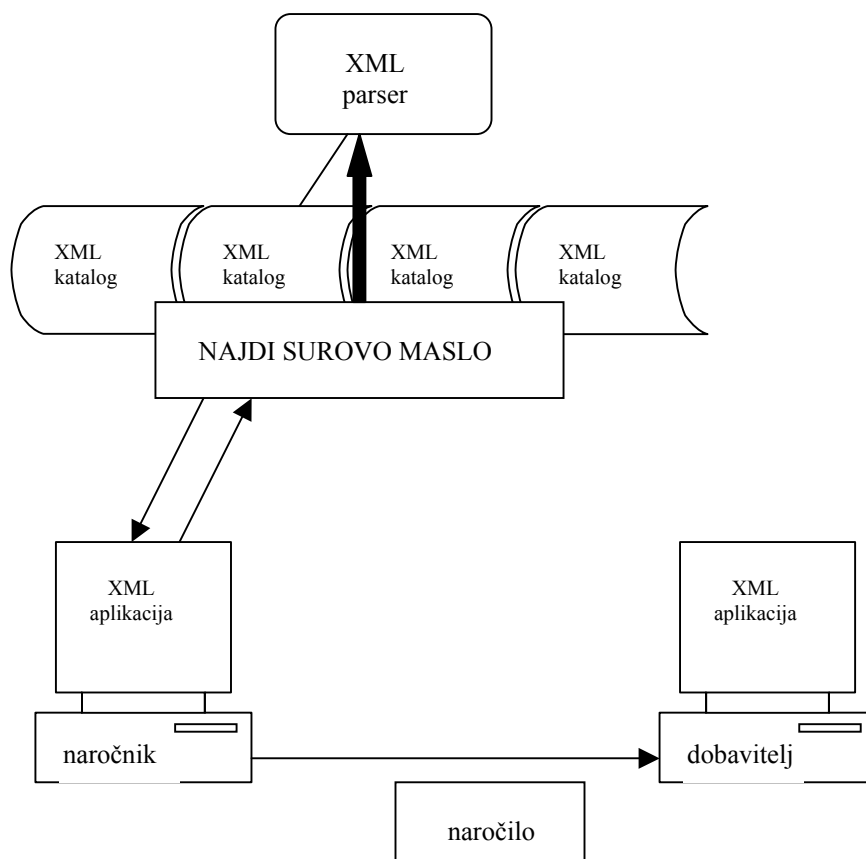
Izvede se postopek naročanja blaga. Po potrditvi naročila dobavitelj dobavi blago, ki se pripelje v distribucijski center in se pripravi na skladiščenje. Kontrola blaga je obsežnejši postopek, ki se izvede ročno v skladišču in ni toliko vezan na delo z računalniki, saj morajo kontrolo izvršiti skladiščniki – prevzemniki fizično v skladišču. Naj omenim, da se kontrolira kakovost blaga, količina blaga in cena blaga. Ko so podatki na dobavnica in naročilnicah pregledani, ter ko je blago fizično preverjeno se lahko blago uskladišči in podatki o prihodu blaga se vnesejo v računalnik.

Vidimo, da v procesu sodelujeta dva poslovna oddelka: logistika in nabava. Pri obstoječi uporabi standarda UN/EDIFACT bi pri tem procesu potekala izmenjava sporočil z dobaviteljem samo pri aktivnosti naročanje blaga. Poleg tega pa je treba tudi predvideti, da tudi dobavitelj uporablja standard

UN/EDIFACT, za katerega smo ugotovili, da je mnogo dražji od standarda XML.

Kako bi prihod blaga v skladišče potekal s standardom XML, si oglejmo na naslednjem enostavnem primeru: naročanje artiklov preko svetovnega spleta. Sedimo v udobnem direktorskem naslonjaču in naročimo svoji aplikaciji nakup surovega masla. Aplikacija, ki zna obvladovati dokumente XML-a, preko svetovnega spleta poišče ustrezne ponudnike surovega masla in dostopa do njihovih katalogov. Ker so katalogi zapisani v jeziku XML, lahko aplikacija preko analizatorja XML (XML parser) izloči iz dokumenta XML s celotnim katalogom samo tiste pomembne podatke, ki jih za to naročilo potrebuje. Iz podatkov o kakovosti, času dobave, količini zalog, cen in ostalih informacij izbere najugodnejšega ponudnika in avtomatično sproži naročilo. Celoten postopek je prikazan na Sliki 6.

Slika 6: Postopek naročanja blaga z aplikacij XML-a preko interneta



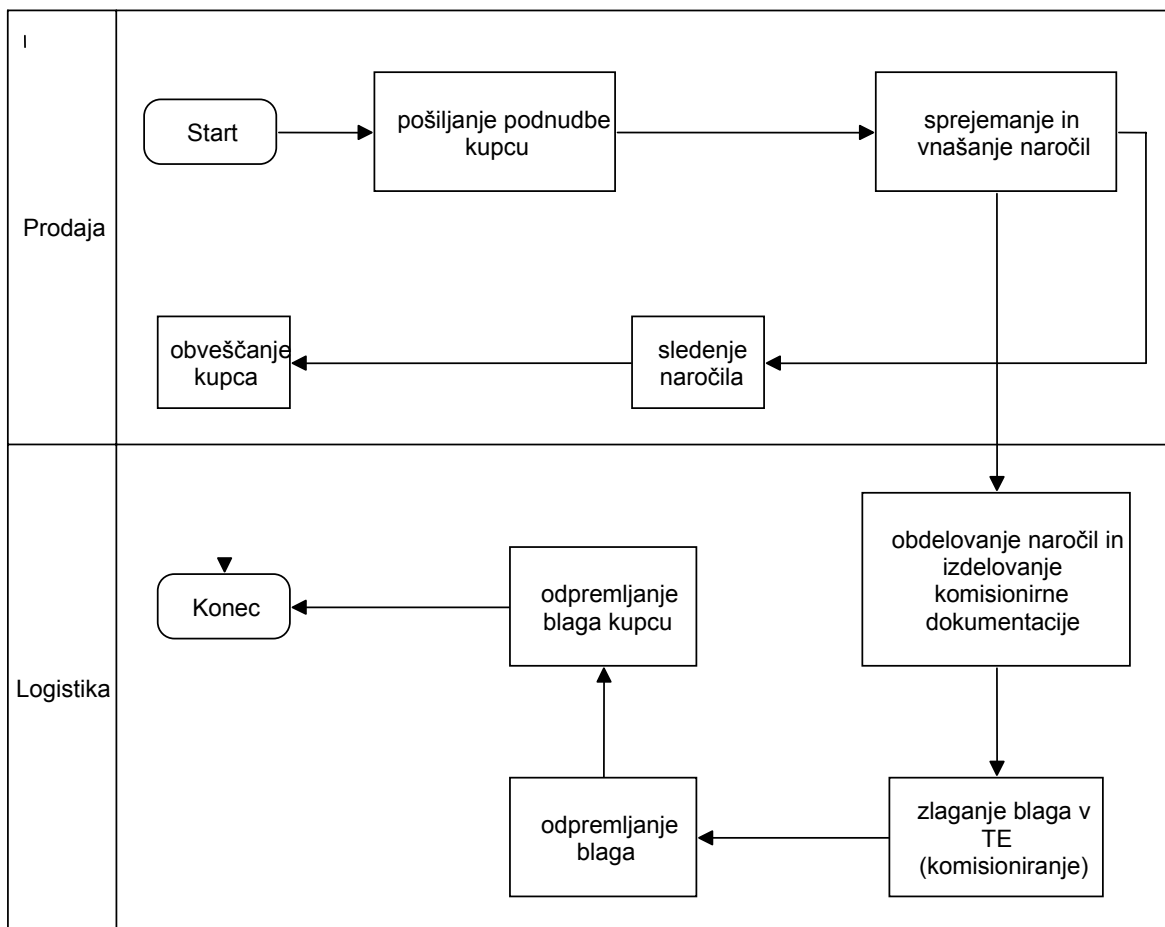
Tu je predvsem pomembno poudariti vpliv katalogov, analizatorjev in splošne razširjenosti jezika XML. Kot sem že omenil obstajajo v Evropi nacionalni katalogi prodajnih artiklov za določeno panogo. Prednost analizatorjev je v tem, da jih ni treba imeti v fizični obliki v svojem podjetju. Do njih lahko dostopajo majhni podjetniki z enim ali več zaposlenimi kakor tudi večja podjetja, preko interneta na strani, kjer so ti analizatorji javno dostopni. Tretja, najpomembnejša stvar, ki nam jo omogoča XML, pa je velika izbira dobaviteljev. Računalnik, dostop do interneta in aplikacijo za procesiranje XML-a si lahko privoščijo tudi manjše mlekarne in zato se bodo tudi njihovi izdelki pojavili v XML katalogih. Standard UN/EDIFACT pa si lahko privoščijo le večja podjetja – Mercator na ta način sodeluje le s 16 dobavitelji. Z uporabo XML bi si lahko razširili krog dobaviteljev, poleg tega pa po vstopu v Evropsko unijo ne bo več pomembno ali blago prihaja s Poljske ali iz Avstrije. Trenutno je zaenkrat še razlika v carinah in zato tudi v cenah. Z vstopom v Evropsko unijo bo trg konkurenčnih dobaviteljev za Mercator čez noč dosegljiv po enakih pogojih.

Kupci pa lahko s pomočjo svoje aplikacije ustvarimo svoj dokument XML s podatki o tem, kaj želimo, kakšno količino, kvaliteto in za kakšno ceno. Nato ta dokument naš računalnik pošlje prodajalcem. Zaznamki v dokumentu točno določajo strukturo dokumenta in identificirajo posamezne elemente v dokumentu, ki predstavljajo naše naročilo. Aplikacija na strani prodajalca brez človeškega posega sprejme in analizira dokument XML, preveri, če je ustrezna surovina na zalogi, če ustreza zahtevani kvaliteti, če je ponujena cena sprejemljiva, in pošlje ponudbo. Pri tem postopku je zelo pomemben čim večji krog dobaviteljev, s katerimi lahko poslujemo.

4.4.2. Podproces odhod blaga iz skladišča

Drugi podproces veleprodajnega procesa je odhod blaga iz skladišča. V prejšnjem primeru smo si ogledali, kako poteka prihod blaga v skladišče, sedaj pa si bomo ogledali, kako poteka odhod blaga iz skladišča. V primerjavi s prejšnjim procesom vidimo, da smo zamenjali funkcijo kupca in dobavitelja. Prej smo bili mi kupec, sedaj pa smo dobavitelj. Model poteka aktivnosti sem tudi tokrat narisal glede na interno gradivo Mercatorja in po pogovoru z zaposlenimi.

Slika 7: Odhod blaga iz skladišča



Slika 7 predstavlja proces odhod blaga iz skladišča, ki ga sestavlja 8 aktivnosti: pošiljanje ponudbe kupcu, sprejemanje in vnašanje naročil, obdelovanje in izdelovanje komisionirnih dokumentov, zlaganje blaga v transportne enote (komisioniranje), odpremljanje blaga, odpremljanje blaga kupcu, sledenje naročila in obveščanje kupca. Tudi pri tem procesu sodelujeta dva poslovna oddelka: prodaja in logistika.

Odprema blaga iz skladišča je obratni pojav od sprejema blaga v skladišče. Pri naročanju blaga z naše strani smo z aplikacijo poiskali v XML registrih dobavitelje in izdelke, sedaj pa se sami vpišemo v določen XML register in tako kupci dostopajo do nas. Naročilo z njihove strani se bo obdelalo avtomatsko – s pomočjo XML aplikacije. Kupci, ki uporabljajo XML kot standard elektronskega poslovanja, ne potrebujejo novih zaposlenih. Sporočila, izmenjana preko XML-a bodo zavedena v bazi podatkov in kasneje

procesirana na isti način in skupaj kot sporočila, ki niso bila prenesena preko standarda XML.

5. SKLEP

Glede na želje Mercatorja ostati prvi trgovec v Sloveniji in postati vodilni prodajalec prehrane na trgih bivše Jugoslavije je uvedba novega, cenejšega standarda za elektronsko poslovanje nujna. Cenejše elektronsko poslovanje omogoča sodelovanje z več dobavitelji. V prilogi prilagam seznam aplikacij za delo z XML standardom (od programskih orodij za pisanje programov, do razčlenjevalnikov in drugih zelo uporabnih aplikacij). Njihova številčnost dokazuje, kako je to področje razvito in koliko različnih možnosti ima podjetje, da izbere orodje, ki bo za podjetje najbolj primerno.

Mercator se za uvedbo standarda XML zaenkrat še ne bo odločil. Trenutno sodelovanje z Gospodarsko zbornico prinaša določene rezultate, a le v obliki uvajanja določenih dokumentov in ne v celostni uvedbi standarda v podjetje. Odločitev Mercatorja, za prehod na XML standard bi imelo posledice v večjem delu gospodarstva, ki pa na to korak še ni popolnoma pripravljen. Uvedba standarda XML je zato predstavljena v obdobje, ko bo na to pripravljeno slovensko gospodarstvo. Nov uveden standard za elektronsko poslovanje pa najverjetneje ne bo XML, ampak ebXML.

Kot smo videli skozi predstavitev XML-a in sorodnih standardov, vidimo, da je standard XML povzročil določeno revolucijo na področju elektronskega poslovanja. Odgovor na vprašanje, zakaj ta revolucija ni vidna tudi pri nas, je predvsem v tem, da je Slovenija majhna država, ki se prilagaja drugim. V času, ko se je v Sloveniji vse vrtelo okrog standarda UNI/EDIFACT, se je v svetu že počasi začel uveljavljati XML.

Čas hiti mimo nas in tudi XML standard trenutno že ni več najbolj idealna rešitev elektronskega poslovanja. V svetu se že uveljavlja nadgradnja standarda XML, in sicer ebXML. Poleg vseh prednosti, ki nam jih prinese XML, pa ima ta vseeno eno zelo veliko slabost. Zaradi velikosti trga uporabnikov standarda XML se pojavljajo vedno nove izboljšave, dodatki in razno razni pripomočki za še lažje delo s standardom XML. Uporabnik ima težave s sledenjem vseh teh novitet in zato za uspešno delovanje včasih potrebujemo celo skupino strokovnjakov, ki se ukvarjajo prav s tem.

Ne glede na vse povedano, podjetja se morajo sama odločiti, kako bodo v prihodnje poslovala, ali se bodo odločila za elektronsko poslovanje ali ne. Informacijska tehnologija je področje, na katerem se vsak dan pripeti nekaj novega, strokovnjaki izumijo novo stvar, že obstoječa tehnologija zastara ali pa se pripeti nekaj povsem nepričakovanega. Podjetja morajo zato imeti strokovnjake, ki se razumejo na te stvari in ki gredo s časom naprej, kakor gre sama informacijska tehnologija.

6. LITERATURA

1. Adler Sharon et al.: Extensible Stylesheet Language (XSL) Version 1.0. [URL: <http://www.w3.org/TR/xsl>], W3C, 21.11.2000.
2. Anderson, Richard et al.: Professional XML. Birmingham, Chicago : Wrox Press, 2000. 1169 str.
3. Bartel Mark et al.: XML-Signature Syntax and Processing. [URL: <http://www.w3.org/TR/2001/CR-xmldsig-core-20010419/>], W3C, 19.4.2001.
4. Berger Sandy: B2B, B2C, and C2B. Who Needs a Translation?. [URL: <http://www.compukiss.com/np/2000/b2bb2c/b2bb2c.html>], Compu-KISS, 2000.
5. Blanc Miloš, Hozjan Martina: Računalniško izmenjavanje podatkov (RIP) v skupini Mercator d.d.. Organizacija, Kranj, 35 (2002), 7, str. 455 – 456.
6. Bos Bert et al.: Cascading Style Sheets, level 2. [URL: <http://www.w3.org/TR/REC-CSS2>], W3C, 12.5.1998.
7. Bos Bert, Lie Hakon Wium: Cascading Style Sheets, level 1. [URL: <http://www.w3.org/TR/REC-CSS1>], W3C, 11.1.1999.
8. Bračko Andrej: Ocena priložnosti računalniškega izmenjavanja podatkov in elektronskega poslovanja v medorganizacijskih povezavah ministrstva za notranje zadeve. Magistrsko delo. Ljubljana : Ekonomska fakulteta, 1997. 83 str.
9. Brain Marshall: How E-commerce works. [URL: <http://www.howstuffworks.com/ecommerce.htm>], 20.7.2001.
10. Bray Tim et al.: Extensible Markup Language (XML) 1,0 (Second Edition) [URL: <http://www.w3.org/TR/2000/REC-xml-20001006>], W3C, 6.10.2000.
11. Clark James: XSL Transformations (XSLT) Version 1.0. [URL: <http://www.w3.org/TR/xslt>], W3C, 16.11.1999.
12. Cowan John, Tobin Richard: XML Information Set. [URL: <http://www.w3.org/TR/xml-infoset>], W3C, 14.5.2001.
13. Damjan Maruša: VIoga urada za UN/EDIFACT pri uvajanju svetovnega standarda za RIP. Informatika v državnih organih, zbornik referatov. 1993. str. 55 - 59.
14. Dečman Mitja: Elektronsko poslovanje in XML. Uporabna informatika, Ljubljana, 8 (2000), 1, str. 51-56.
15. Flisi Maximilian: E-distributors poised to lead B2B. [URL: <http://www.itworld.com/Tech/3478/CI0010415newsletter>], 15.4.2001.

16. Harold Elliotte Rusty: XML Bible. Foster City, CA : IDG Books Worldwide, 1999. 974 str.
17. Hiroshi, Schaad Jim, Simon Ed: XML Encryption Syntax and Processing. [URL: http://lists.w3.org/Archives/Public/xml-encryption/2000Dec/att-0024/01-XMLEncryption_v01.html], W3C, 15.12.2000.
18. Huemer Christian: XML vs. UN/EDIFACT or Flexibility vs. Standardisation. Bled :13th International Electronic Commerce Conference, 2000, str. 680 - 692.
19. Jamnik Anka: Razširljivi označevalni jezik XML in njegove razširitve ter pomen pri razvoju poslovnih Internetnih storitev. Magistrsko delo. Ljubljana : Ekonomska fakulteta, 2001. 98 str.
20. Jerman Blažič Borca et al.: Elektronsko poslovanje na Internetu. Ljubljana : GV Založba, 2001. 206 str.
21. Košir Franc: Elektronsko poslovanje v blagovno-logističnih procesih. Organizacija, Kranj, 31 (1998), 3, str. 130 – 132.
22. Mohorič Tomaž, Krisper Marjan: O razširljivem označevalnem jeziku XML in njegovi uporabi. Uporabna informatika, Ljubljana, 9(2001), 2, str. 55 – 59.
23. North Ken: Understanding ODBC 3.0 Standards and OLE DB. [URL: <http://www.dbmsmag.com/9604d53.html>], DBMS OnLine, april 1996.
24. Outershout K. John. [URL: <http://home.pacbell.net/ouster/>], 20.05.2002.
25. Pardi J. William: XML in Action. Microsoft Press, 1999. 329 str.
26. Pucihar Andreja: Priložnosti in težave elektronskega poslovanja med organizacijami v Sloveniji. Uporabna informatika, Ljubljana, 7(1999), 4, str. 7-13.
27. Simon Ed, Madsen Paul, Adams Carlisle: An Introduction to XML Digital Signatures. [URL: <http://www.xml.com/pub/a/2001/08/08/xmlsig.html>], XML.com, 8.8.2001.
28. Slana Lidija: Elektronsko poslovanje - uvajanje mednarodnega standarda UN/EDIFACT v poslovno in bančno okolje. Uporabna informatika, Ljubljana, 7(1999), 4, str. 25 – 31.
29. Sol Selena: History of XML. [URL: <http://wdvl.internet.com/Authoring/Languages/XML/Tutorials/Intro/history.html>], Web Developer's, 8.3.1999.
30. Sturm Jake: Developing XML Solutions. Microsoft Press, 2000. 415 str.

- 31.Škedelj Miran: Uporaba sodobnih telekomunikacijskih storitev v poslovanju vedrogerije. Magistrsko delo. Ljubljana : Ekonomska fakulteta, 1999. 97 str.
- 32.Toplišek Janez: Elektronsko poslovanje. Prva izdaja. Ljubljana : Atlantis, 1998. 336 str.
- 33.Travis E. Brian: XML and SOAP programming for Biztalk servers. Microsoft Press, 2000. 428 str.
- 34.Vavan Ilijana: B2C End-to-End Solution (URL: <http://www.microsoft.si/6ntk/PPTpredstavitev/Tor/241.ppt>), 21.7.2001
- 35.Walsh Norman: What is XML?.
[URL: <http://www.xml.com/pub/a/98/10/guide1.html#AEN58>], 3.10.1998.
- 36.Weisman Jon: The Making of E-Commerce: 10 Key Moments.
[URL: <http://www.ecommercetimes.com/perl/story/4085.html>], 22.8.2000.
- 37.Zupan Neja, Gričar Jože: Priložnosti in težave izmenjavanja podatkov preko Interneta med večjimi in manjšimi podjetji v Sloveniji. Zbornik posvetovanja. 1999, str. 278 – 286.
- 38.Zupan Neja: Nove internet tehnologije za elektronsko poslovanje. Organizacija, Kranj, 33 (2000), 4, str. 274 – 280.
- 39.Zupančič Boris, Sedej Marjan: Elektronsko poslovanje v oskrbovalnih verigah. Organizacija, Kranj, 33 (2000), 3, str. 194 – 199.

7. VIRI

1. About the World Wide Web Consortium (W3C). [URL: <http://www.w3.org/Consortium>], W3C, januar2001.
2. Anketa podjetij, Elektronsko poslovanje z državo RIS 2000/2001. [URL: <http://www.ris.org/id/elekLposLdrzava.htm>], julij 2001.
3. Bokler: Bokler Software's Cryptographic Resources Page. [URL: http://spider.bokler.com/bokler/resrcs_1.html], Bokler Software Corp., 2001.
4. Certifikati javnih ključev. [URL: <http://www.gov.si/tecaj/kripto/kr-cert.htm>], Center vlade RS za informatiko, december 2000.
5. ebXML - Enabling A Global Electronic Market -Specifications. [URL: <http://www.ebxml.org/specs/index.htm>], oktober2001.
6. ebXML FAQ. [URL: <http://www.ebxml.org/faq.htm>], ebXML.org, oktober2001.
7. Extensible Markup Language (XML) 1.0 (Second Edition). [URL: <http://www.w3.org/TR/2000/REC-xml-20001006>], 6.10.2000.
8. iPier.com: E-Commerce 101: E-Commerce Overview. [URL: <http://www.ipier.com/ecom101/ecom1.html>], 20.7.2001.
9. Kubler Jean: Electronic commerce on the Web - background information: From paper to EDI to electronic forms using XML?. [URL: <http://www.unece.org/cefact/xml/press/eforms.pdf>], United Nations Centre for Trade Facilitation and Electronic Business, 7.12.1999.
10. RIS: Anketa podjetij, Elektronsko poslovanje RIS 2000/2001. [URL: http://www.ris.org/publikacije/j_eposlovnj.htm], julij 2001. RiS:
11. SAX 2.0: The Simple API for XML. [URL: <http://www.megginson.com/SAX/index.html>], Megginson Technologies Ltd., 5.5.2000.
12. SAX2 Developer's Guide. [URL: <http://www.msdn.microsoft.com/library/default.asp?URL=/library/psdk/xmlsdk/sax20jef.ht>], Microsoft, 17.11.2000.
13. Šalamon Brane: Internet pojmovnik. Druga izdaja. Izola : Desk d.o.o., 1998. 287 str.
14. W3C: Extensible Markup Language (XML) 1.0 (second edition). [URL: <http://www.w3.org/TR/REC-xml>], W3C, 6.10.2000.
15. World Wide Web Consortium (W3C). [URL: <http://www.w3.org>], W3C, august 2001.

16.XML DOM User Guide.

[URL:<http://msdn.microsoft.com/library/default.asp?URL=/library/psdk/xmlsdk/xmlp2lt1.htm>], Microsoft, 17.11.2000.

17.Zakon o elektronskem poslovanju in elektronskem podpisu. (Uradni list RS, št. 57/2000).

PRILOGE

Priloga 1: Pojasnilo kratic in slovar angleških izrazov

	Java	Programski jezik podjetja Sun Microsystems
	Multi-tier	Večnivojski
	SQL Server	Microsoftov sistem za upravljanje baz podatkov
	Tag	Oznaka
	TeX	Jezik za oblikovanje matematičnih besedil
	Unicode	Univerzalna kodna tabela
	Valid XML document	Veljaven dokument-XML
	XML Encryption	Šifriranje XML
	Well-formed XML document	Dobro oblikovan dokument XML
	Windows	Microsoftov operacijski sistem
API	Application Protocol Interface	Vmesnik uporabniškega programa
B2B	Business to Business	Poslovanje med podjetji
B2C	Business to Customer	Poslovanje med podjetji in strankami
CSS	Cascading Style Sheets	Kaskadne stilne predloge
DOM	Document Object Model	Objektni model dokumenta
DTD	Document Type Definition	Definicija tipa dokumenta
EDI	Electronic Data Interchange	Elektronska izmenjava podatkov
HTML	Hyper Text Markup Language	Jezik za označevanje hiperteksta
HTTP	Hyper Text Transfer Protocol	Protokol za prenos hiperteksta
IEC	International Electrotechnical Commission	Mednarodna komisija za elektrotehniko
ISO	International Organization for Standardization	Mednarodna organizacija za standardizacijo
ODBC	Open Database Connectivity	Vmesnik za odprto povezovanje podatkovnih baz
PDF	Portable Document Format	Prenosljiv format dokumenta
PKI	Public Key Infrastructure	Infrastruktura javnega ključa
RTF	Rich Text Format	Format z obogatenim besedilom
SAX	Simple API for XML	Enostavni vmesnik uporabniškega programa za XML

SGML	Standard Generalized Markup Language	Standardni splošni označevalni jezik
SQL	Structured Query Language	Strukturiran poizvedovalni jezik
SSL	Secure Socket Layer	Sloj varnih vtičnic
TLS	Transport Layer Security	Transportni nivo varnosti - .
URL	Uniform Resource Locator	Enolični lokator vira
VAN	Value Added Networks	Omrežja z dodano vrednostjo
W3C	World Wide Web Consortium	Konzorcij za svetovni splet.
WAP	Wireless Application Protocol	Protokol za brezžične aplikacije
WML	Wireless Markup Language	Brezžični označevalni jezik
XML	Extensible Markup Language	Razširljivi označevalni jezik
XSL	Extensible Stylesheet Language	Razširljivi jezik za stil
XSLF	XSL Formatting Object	Oblikovani objekt XML
XSLT	Extensible Stylesheet Language Transformation	Transformacije razširljivega jezika za stil

Priloga 2: Programi, orodja in aplikacije za delo z XML

CSS urejevalniki

Izdelek	Podjetje	Platforma
<u>css-mode</u>	Lars Marius Garshol	Emacs
<u>HTML-Kit</u>	Chami.com	Win32

DTD oblikovalci dokumentov

Izdelek	Podjetje	Platforma
<u>dtddoc</u>	Lars Marius Garshol	Python
<u>DTDParse</u>	Norman Walsh	Perl
<u>LiveDTD</u>	Robert Stayton	Perl
<u>perISGML</u>	Earl Hood	Perl

DTD urejevalniki

Izdelek	Podjetje	Platforma
<u>ezDTD</u>	Duncan Chen	Win32
<u>tdtd</u>	Tony Graham	Emacs

DTD generatorji

Izdelek	Podjetje	Platforma
<u>Data Descriptors by Example</u>	IBM alphaWorks	Java
<u>FirstSTEP EXML</u>	Product Data Integration Technologies (PDIT)	Win32
<u>Rhythmyx XSpLit</u>	Percussion Software	Win32
<u>SAXON</u>	Michael H. Kay	Java 1.4
<u>XMI Toolkit</u>	IBM alphaWorks	Java 1.2
<u>xml2ddml</u>	Rick Jelliffe	OmniMark

DTD prevajalniki

Izdelek	Podjetje	Platforma
<u>CL-XML</u>	James Anderson, Benno Biewer	Common Lisp
<u>DTD Parser</u>	Ron Bourret	Java
<u>DTDParse</u>	Norman Walsh	Perl
<u>DTDParser</u>	Mark Wutka	Java
<u>PXP</u>	Gerd Stolpmann	Objective Caml 3.00
<u>xmlproc</u>	Lars Marius Garshol	Python 1.5

Integrirana razvijalska okolja

Izdelek	Podjetje	Platforma
<u>XML and Web Services DE</u>	IBM alphaWorks	Win32

Prevajalniki shem

Izdelek	Podjetje	Platforma
<u>DTD2RELAX</u>	Murata Makoto, Nanba Ryosuke	Java
<u>dtd2xs</u>	Ralf Schweiger	Java

XSL skladišča

Izdelek	Podjetje	Platforma
<u>XSL Lint</u>	Norman Walsh	Perl
<u>XSL Trace</u>	IBM alphaWorks	Java

XSL prevajalniki

Izdelek	Podjetje	Platforma
<u>XSL to XSLT Converter</u>	Microsoft	Win32

XSLT urejevalniki

Izdelek	Podjetje	Platforma
<u>FOA</u>	Fabio Giannetti	Java
<u>HTML-Kit</u>	Chami.com	Win32
<u>XPath Tester</u>	FiveSight	Java 1.2
<u>XSL Editor</u>	IBM alphaWorks	Java
<u>XSL Tester</u>	VBXML.COM	Win32
<u>xslide</u>	Mulberry Technologies	Emacs
<u>XSLT-process</u>	Ovidiu Predescu	Emacs

XSLT generatorji

Izdelek	Podjetje	Platforma
<u>Rhythmyx XSpLit</u>	Percussion Software	Win32
<u>WH2FO</u>	Fabio Giannetti	Java

Splošni n-prevajalniki: prevajalniki za prevajanje iz drugih oblik v obliko XML

Izdelek	Podjetje	Platforma
<u>DB2XML</u>	Volker Turau	Java 1.1
<u>Java HTML Tidy</u>	Andy Quick	Java
<u>JEDI</u>	Gerald Huck, Peter Fankhauser	Java
<u>Majix</u>	TetraSix	Java
<u>Some2XML</u>	Paul Tchistopolskii	Perl
<u>Tidy</u>	Dave Raggett	Win32, Linux, MacOS, BeOS, SunOS, Solaris, IRIX, HP-UX, AIX, Amiga, Atari, MS-DOS and C source.
<u>XML Lightweight Extractors</u>	IBM alphaWorks	AIX, OS/390, Win32 and Unix, written in Java
<u>XML::DT</u>	J.Joao Almeida, J.Carlos Ramalho, Jorge G. Rocha	Perl
<u>xmlizer</u>	e-freak web agency	Java

Prevajalniki v obliko za objavo

Izdelek	Podjetje	Platforma
<u>JXRTE</u>	JWay	Java
<u>TeXML</u>	IBM alphaWorks	Java

General S-converters

Izdelek	Podjetje	Platforma
<u>Ace</u>	RMIT Multimedia Database Systems Group	Solaris and Win32
<u>Fxt</u>	Alexandru Berlea	Standard ML
<u>MetaMorphosis</u>	Ovidius	Linux (and commercially available on other platforms)
<u>SwiX</u>	Griffin Brown Digital Publishing	Java
<u>TransformXML</u>	Frank Boumphrey	Win32
<u>X-IT</u>	IBM alphaWorks	Java
<u>X-Tract</u>	DecisionSoft	Win32 and Linux
<u>XML Translator Generator</u>	IBM alphaWorks	Java
<u>XTAL</u>	Oliver Zeigermann	Java
<u>xtr2any</u>	Sema Group	Linux and Win32

Orodja za shranjevanje dokumentov XML

Izdelek	Podjetje	Platforma
<u>4Suite</u>	Fourthought	Python
<u>dbXML</u>	dbXML Group	Java
<u>eXist</u>	The eXist team	Java 1.3
<u>infozone</u>	The Ozone Project	Java
<u>RDFDB</u>	R.V. Guha	Linux and C source, also accessible from Perl
<u>Redfoot</u>	The Redfoot Team	Python
<u>XDBM</u>	XDBM Project	Win32 and Linux
<u>XML-DBMS</u>	Ron Bourret	Java and Perl

Posebni n-prevajalniki

Izdelek	Podjetje	Platforma
<u>RTF2XML</u>	Rick Geimer	OmniMark v3r1a or later
<u>XML::Edifact</u>	Michael Koehne	Perl

Orodja za upravljanje XML dokumentov

Izdelek	Podjetje	Platforma
<u>XMill</u>	Hartmut Liefke, Dan Suciu	Portable C++ source
<u>XML Diff and Merge Tool</u>	IBM alphaWorks	AIX, Win32
<u>XML TreeDiff</u>	IBM alphaWorks	Java
<u>xmldiff</u>	Alexandre Fayolle	
<u>XMLZip</u>	XML Solutions	Win32, Linux and Solaris

XML iskalniki

Izdelek	Podjetje	Platforma
<u>Fxgrep</u>	Andreas Neumann	Standard ML
<u>GMD-IPSI XQL Engine</u>	Gerald Huck, Ingo Macherius	Java
<u>QuiP</u>	Software AG	Win32
<u>sgrep</u>	Jani Jaakkola, Pekka Kilpeläinen	Win32, HP-UX, Linux, OSF/1 Unix and Solaris
<u>XML::QL</u>	Matt Sergeant	Perl
<u>XSet</u>	Ben Zhao	Java
<u>Xtract</u>	Malcolm Wallace	Haskell

Orodja za objavljanje na spletu

Izdelek	Podjetje	Platforma
<u>Cocoon</u>	The Java Apache Project	Java
<u>maki</u>	Sam Brauer	Python
<u>mod_xslt</u>	UserActive	C++ Apache module
<u>PHP</u>	The PHP development team	Win32, Unix and C source.
<u>PXSLServlet</u>	Paul Tchistopolskii	Java
<u>Rich Men's Document Management System</u>	rm -d ms team	Java
<u>XSL ISAPI Extension</u>	Microsoft	Win32

XML urejevalniki

Izdelek	Podjetje	Platforma
<u>Amaya</u>	World Wide Web Consortium	Win32, Linux, Sun/Solaris, AIX, OSF/1 Unix and source for other platforms.
<u>Cooktop</u>	The Cooktop Team	Windows
<u>debit</u>	interati	Win32
<u>Emilé</u>	Media Design in*Progress	MacOS
<u>eXchaNGeR</u>	Edwin Dankert	Java
<u>GenDiapo</u>	Alexandre Flament, Christophe Quay-Thevenon, Cecile Guillox	Java 1.2
<u>HTML-Kit</u>	Chami.com	Win32
<u>PECEL</u>	ATL Systems	Java 1.2.1
<u>Protégé</u>	The Protégé group	Java
<u>PSGML</u>	Lennart Staflin	Emacs
<u>S-Link-S Editor</u>	Openly Informatics	Java 1.1
<u>Sixpack</u>	simple/CHAOS	MacOS
<u>sxml-mode</u>	Philippe Le Hégaret	Emacs
<u>Visual XML</u>	Pierre Morel	Java 1.1 with JFC
<u>XED</u>	Henry Thompson	Win32, Linux, FreeBSD and Solaris
<u>Xeena</u>	IBM alphaWorks	Win32 and Unix
<u>xmloperator</u>	Didier Demany	Java 1.2
<u>xmltools</u>	Alexandre Fayolle	Python
<u>XmlTree</u>	Alberto Bellina	Win32

XML brskalniki

Izdelek	Podjetje	Platforma
<u>Amaya</u>	World Wide Web Consortium	Win32, Linux, Sun/Solaris, AIX, OSF/1 Unix and source for other platforma.
<u>eXchaNGeR</u>	Edwin Dankert	Java
<u>IBM techexplorer</u>	IBM	Win32, Linux, MacOS, AIX, Solaris and IRIX
<u>Microsoft Internet Explorer</u>	Microsoft	Win32, HP-UX and MacOS
<u>Mozilla</u>	The Mozilla team	Win32, MacOS, Linux, OpenVMS, FreeBSD and Unix source
<u>X-Smiles</u>	Telecommunications Software and Multimedia Laboratory at Helsinki University of Technology	Java
<u>XML Viewer</u>	IBM alphaWorks	Java
<u>Xplorer</u>	IBM alphaWorks	Java

Orodja za izgradnjo arhitekturne oblike

Izdelek	Podjetje	Platforma
<u>XAF</u>	Megginson Technologies, JXML	Java 1.1.
<u>xmlarch</u>	Geir Ove Grønmo	Python

DSSSL orodja

Izdelek	Podjetje	Platforma
<u>DSC</u>	Henry Thompson	SunOS
<u>Jade</u>	James Clark	Win32, C++ source for other platforma.
<u>OpenJade</u>	The OpenJade team	Portable C++ code

Implementacije vmesnika DOM

Izdelek	Podjetje	Platforma
<u>4Suite</u>	Fourthought	Python
<u>Docuverse DOM SDK</u>	Docuverse	Java
<u>domc</u>	Michael B. Allen	C
<u>Gdome</u>	Raph Levien	Linux
<u>GNU JAXP</u>	The GNU Project	Java
<u>libxml-eno</u>	Enno Derksen	Perl
<u>TclDOM</u>	Steve Ball	tcl 8.x
<u>tDOM</u>	Jochen Loewer	C, to be used from tcl
<u>XDOM</u>	Open XML	Delphi 3 and higher, perhaps also 2

RDF pretvorniki

Izdelek	Podjetje	Platforma
<u>4Suite</u>	Fourthought	Python
<u>ARP</u>	Jeremy Carroll	Java
<u>DATAx</u>	Megginson Technologies	Java 1.2
<u>ICS-FORTH Validating RDF Parser</u>	ICS-FORTH	Java 1.2
<u>Jena</u>	Brian McBride	Java
<u>Raptor</u>	Dave Beckett	C
<u>RDF Filter</u>	Megginson Technologies	Java
<u>RDF-Parser</u>	Profium Ltd.	Perl
<u>Redfoot</u>	The Redfoot Team	Python
<u>repat</u>	Jason Diamond	C
<u>SiRPAC</u>	World Wide Web Consortium	Java

SGML/XML pretvorniki

Izdelek	Podjetje	Platforma
<u>OpenSP</u>	The OpenJade team	Win32, MS-DOS, Solaris 2.3 SPARC and 2.5 i386, Linux i386, OSF/1 Unix DEC Alpha V3.2 with C++ source for other platforma.
<u>pysp</u>	Lars Marius Garshol	Python
<u>SGMLSpM</u>	David Megginson	Perl
<u>SP</u>	James Clark	Win32, MS-DOS, Solaris 2.3 SPARC and 2.5 i386, Linux i386, OSF/1 Unix DEC Alpha V3.2 with C++ source for other platforma.
<u>SPIN_py</u>	Graham Swallow	Python
<u>SPIN_tcl</u>	Graham Swallow	tcl

Topic map programi

Izdelek	Podjetje	Platforma
<u>GNEWSYS</u>	GNEWSYS	Python
<u>GWTK</u>	Jan Algermissen	C and Python
<u>Perl XTM</u>	Robert Barta	Perl
<u>SemanText</u>	Eric Freese	Python
<u>TM4J</u>	Kal Ahmed	Java 1.2
<u>tmproc</u>	Geir Ove Grønmo	Python
<u>TMTab</u>	Kal Ahmed	Java

XLink/XPointer aplikacije

Izdelek	Podjetje	Platforma
<u>Jaxen</u>	James Strachan	Java
<u>Pathan</u>	DecisionSoft	C++
<u>SAXPath</u>	James Strachan	Java
<u>XML::XPath</u>	Matt Sergeant	Perl
<u>XPath4XT</u>	Takuki Kamiya	Java

XML vmesniki

Izdelek	Podjetje	Platforma
<u>BML</u>	Anders W. Tell	Java
<u>CGI::XMLForm</u>	Matt Sergeant	Perl
<u>Cost</u>	Joe English	tcl
<u>CSS2 Parser</u>	Steady State Software	Java 1.1
<u>DatabaseDom</u>	IBM alphaWorks	Java
<u>dom4j</u>	James Strachan	Java 1.2
<u>DOMParser</u>	John Cowan	Java
<u>DT4DTD</u>	Extensibility	Java
<u>easyXML</u>	IBM alphaWorks	Java 1.1
<u>ExCost</u>	JWay	tcl
<u>Flute</u>	World Wide Web Consortium	Java and C
<u>GPS</u>	Geir Ove Grønmo	Python
<u>Java/XML Quick</u>	The Quick project	Java 1.2
<u>JDOM</u>	The JDOM Project	Java 1.1
<u>LT PyXML</u>	Edinburgh Language Technology Group	Python
<u>MDSAX</u>	JXML	Java 1.2
<u>Parser Filters</u>	John Cowan	Java
<u>PatML</u>	IBM alphaWorks	Java
<u>Protégé</u>	The Protégé group	Java
<u>Python XML package</u>	The Python XML-SIG	Python 2.x
<u>Pyxie</u>	Sean McGrath	Python
<u>PyXPath</u>	Dieter Maurer	Python
<u>Relaxer</u>	Asami Tomoharu	Java
<u>SAX</u>	Megginson Technologies	Java
<u>SAX drivers for Lark and MSXML</u>	Megginson Technologies	Java
<u>SAX in C++</u>	Jez Higgins	C++
<u>SAX2</u>	David Megginson	Java
<u>SAXON</u>	Michael H. Kay	Java 1.4
<u>Wraf</u>	The Wraf Team	Perl
<u>XFA Scripting System</u>	XML For All	Win32 and

		Linux.
<u>XIncluder</u>	Elliotte Rusty Harold	Java
<u>XML Bean Suite</u>	IBM alphaWorks	Java
<u>XML Entity and URI Resolvers</u>	Sun Microsystems	Java
<u>XML Generator</u>	IBM alphaWorks	Java
<u>XML Master</u>	IBM alphaWorks	Java
<u>XML Security Suite</u>	IBM alphaWorks	Java
<u>XML Testbed</u>	Steve Withall	Java 1.1
<u>Xml2Beans</u>	Robbie Schäfer	Java
<u>XML::Grove</u>	Ken MacLeod	Perl
<u>XML::Twig</u>	Michel Rodriguez	Perl
<u>XML::Writer</u>	David Megginson	Perl
<u>xmlBlaster</u>	The xmlBlaster team	Java
<u>XOM</u>	Elliotte Rusty Harold	Java

XML pretvorniki

Izdelek	Podjetje	Platforma
<u>AdvXMLParser</u>	Sebastien Andrivet	Portable C++ source
<u>CenterPoint/XML</u>	CenterPoint	C++
<u>CL-XML</u>	James Anderson, Benno Biewer	Common Lisp
<u>eXML</u>	Andreas Leitner	Eiffel
<u>expat</u>	James Clark	Platform-independent C source
<u>Expat Module for Ruby</u>	Yoshida Masato	Ruby
<u>Expat REALbasic plugin</u>	Doug Holton	REALbasic
<u>fxp</u>	Andreas Neumann	Standard ML
<u>GNU JAXP</u>	The GNU Project	Java
<u>HaXml</u>	Malcolm Wallace, Colin Runciman	Haskell
<u>HXML</u>	Joe English	HXML
<u>Java Project X</u>	Sun Microsystems	Java 1.1

<u>Lark</u>	Tim Bray	Java
<u>libxml</u>	Daniel Veillard	C
<u>LTXML</u>	Edinburgh Language Technology Group	Win32 binary, and C source for Unix platforma.
<u>Microsoft XML Parser</u>	Microsoft	Win32 with MSIE 4.01 SP1
<u>Nenie XML</u>	Frank Arnaud	Eiffel
<u>Perl libxml</u>	Ken MacLeod	Perl
<u>PXP</u>	Gerd Stolpmann	Objective Caml 3.00
<u>RXP</u>	Richard Tobin	C source has been compiled under Win32, Solaris and FreeBSD.
<u>sgmlop</u>	PythonWare	Python
<u>SXP</u>	Patrice Bonhomme	Java 1.1
<u>TclExpat</u>	Steve Ball	tcl 8.0 and 8.1a2, tested on MacOS, Windows, Solaris 2.5.1, Linux RedHat 5.0 and HP-UX 10.10.
<u>TclXML</u>	Steve Ball	tcl 8.1
<u>TinyXML</u>	Tom Gibara	Java
<u>Tony</u>	Christian Lindig	Objective Caml
<u>UncommonXML</u>	onShore	Common Lisp
<u>Whisper</u>	Jesse Jones	MacOS and Win32, written in C++.
<u>Windows Foundation Classes</u>	Sam Blackburn	C++ source for Win32 and Unix
<u>XDOM</u>	Open XML	Delphi 3 and higher, perhaps also 2
<u>Xerces C++</u>	The Apache XML Project	Portable C++ source, tested on Linux, Win32 Solaris, AIX, HP-UX, OS/390, AS/400, IRIX, MacOS and OS/2.
<u>Xerces Java</u>	The Apache XML Project	Java
<u>Xerces Perl</u>	The Apache XML Project	Perl
<u>XML for C++</u>	IBM alphaWorks	Portable C++ source, known to work on AIX, Linux, Solaris,

		Win32, HP-UX, OS/2, AS/400 and OS/390.
<u>XML Interface for RPG</u>	IBM alphaWorks	AS/400
<u>XML Parser Component for Delphi</u>	ICOM Datenverarbeitungs GmbH	Delphi 4
<u>XML Parser for Java</u>	IBM alphaWorks	Java 1.1
<u>XML Pull Parser</u>	Aleksander Slominski	Java and C++
<u>XML Tools Scripting Addition</u>	Late Night Software	AppleScript 1.3 on PowerPC
<u>xml.expats</u>	Kaelin Colclasure	Common Lisp
<u>xml.lisp</u>	The CLOCC team	Common Lisp
<u>XML::Parser</u>	Clark Cooper	Perl
<u>XMLIO</u>	Paul Miller	Portable C++ source
<u>xmlparse</u>	Morus Walter	Linux and other Unixes.
<u>xmlproc</u>	Lars Marius Garshol	Python 1.5
<u>xmlutils</u>	Franz, Inc.	Common Lisp Allegro 6.0
<u>XP</u>	James Clark	Java 1.1
<u>Ælfred</u>	Jun Fujisawa	Java kvm

XSL orodja

Izdelek	Podjetje	Platforma
<u>FOP</u>	The Apache XML Project	Java 1.2
<u>jfor</u>	Bertrand Delacrétaiz	Java
<u>PassiveTeX</u>	Sebastian Rahtz	TeX
<u>REXP</u>	DIBE at University of Genoa	Java
<u>Unicorn Formatting Objects</u>	Unicorn Enterprises	Windows

XSLT orodja

Izdelek	Podjetje	Platforma
<u>4Suite</u>	Fourthought	Python
<u>jd.xslt</u>	Johannes Döbler	Java 1.1.
<u>libxslt</u>	Daniel Veillard	C
<u>LotusXSL</u>	IBM alphaWorks	Java
<u>MDC-XSL</u>	Minoru Development Corporation	C++ source, known to be running on Linux.
<u>Microsoft XML Parser</u>	Microsoft	Win32 with MSIE 4.01 SP1
<u>Pyana</u>	Brian Quinlan	Python
<u>Sablotron</u>	Ginger Alliance	Portable C++ source for Linux, OpenBSD and Win32, also interfaces to Perl, Ruby and Python
<u>SAXON</u>	Michael H. Kay	Java 1.4
<u>Unicorn XSLT Processor</u>	Unicorn Enterprises	Win32
<u>Xalan-C++</u>	The Apache XML Project	C++
<u>Xalan-Java</u>	The Apache XML Project	Java
<u>XML::LibXSLT</u>	Matt Sergeant	Perl
<u>xsl:p</u>	Keith Visco	Java
<u>xslt-parser</u>	Geert Josten, Egon Willighagen	Perl
<u>XSLTC</u>	Olivier Gerardin	C++
<u>XT</u>	James Clark	Java

XML validatorji

Izdelek	Podjetje	Platforma
<u>DSD Processor</u>	Anders Møller, Michael I. Schwartzbach	C source and binaries for Unix and Win32
<u>Jing</u>	James Clark	Java
<u>PyTREX</u>	James Tauber	Python
<u>RELAX Verifier for Java</u>	Swift Inc.	Java
<u>Sample TREX implementation</u>	James Clark	Java
<u>Schematron</u>	Rick Jelliffe	XSLT, Python, and Perl
<u>VBRELAXNG</u>	YONEKURA Koji	Visual Basic 6.0
<u>XML Schema Validator</u>	Richard Tobin, Henry Thompson	Python

Vir: URL: http://www.garshol.priv.no/download/xmltools/cat_ix.html, 2003