

UNIVERZA V NOVI GORICI
POSLOVNO-TEHNIŠKA FAKULTETA

**PROGRAMSKA NADGRADNJA PRIPRAVE POTNIH
NALOGOV V PODJETJU**

DIPLOMSKO DELO

Rok Boltar

Mentor: doc. dr. Bogdan Filipič

Nova Gorica, 2008

ZAHVALA

Zahvaljujem se mentorju doc. dr. Bogdanu Filipiču za nasvete in pomoč pri izdelavi diplomske naloge ter Jožetu Ilcu in kolektivu sektorja Organizacije in informatike v Iskri Avtoelektrika d.d. za sprejem in pomoč pri izdelavi diplomskega dela. Zahvaliti se želim tudi svoji družini, ki mi je stala ob strani in mi pomagala z nasveti.

IZVLEČEK

Elektronska izmenjava podatkov v podjetjih in med njimi je danes zelo pomembna, saj prihrani dragoceni čas in denar. V diplomskem delu obravnavamo programsko nadgradnjo priprave potnih nalogov v podjetju Iskra Avtoelektrika d.d. Predstavimo nekatere vidike razvoja tehnologije XML, saj v podjetju uporabljajo intranet, ki temelji na tej tehnologiji. Da bi povečali preglednost potnih nalogov in odpravili težave pri njihovem tiskanju na različnih tiskalnikih, so se odločili, da njihovo pripravo programsko nadgradijo. Pri nadgradnji so bili ključnega pomena tehnologija XML ter programa Oracle BI Publisher in Oracle JDeveloper 10g. Rezultat dela je programska oprema, ki omogoča pripravo potnih nalogov s programi Microsoft Office in v obliki PDF. Programska oprema se že redno uporablja v podjetju.

ABSTRACT

Electronic exchange of data within companies and among them is very important nowadays because it saves valuable time and money. This diploma thesis deals with the software upgrade of travel commission processing in the Iskra Avtoelektrika company. Some aspects of the XML technology are presented as the company uses intranet based on this technology. To increase the clarity of travel commissions and eliminate the difficulties with printing them on different printers, it was decided to upgrade their preparation. Of crucial importance in upgrading were the XML technology, and the programs Oracle BI Publisher and Oracle JDeveloper 10g. The work resulted in a software that makes it possible to prepare travel commissions with the Microsoft Office programs and in the PDF format. The software is in regular use in the company.

KLJUČNE BESEDE

elektronska izmenjava podatkov, spletna storitev, potni nalog, XML, XSL-FO, java

KEY WORDS

electronic data exchange, web service, travel comission, XML, XSL-FO, java

KAZALO

| | | |
|-------|--|----|
| 1 | UVOD | 1 |
| 2 | ELEKTRONSKA IZMENJAVA PODATKOV IN DOKUMENTOV | 3 |
| 3 | PRIPRAVA POTNIH NALOGOV V ISKRI AVTOELEKTRIKI | 6 |
| 3.1 | O podjetju..... | 6 |
| 3.2 | Računalniška priprava potnih nalogov | 7 |
| 3.3 | Potrebe po preglednejšem prikazovanju potnih nalogov..... | 8 |
| 4 | UPORABLJENI JEZIKI IN PROGRAMSKA ORODJA | 9 |
| 4.1 | Označevalni jezik XML..... | 9 |
| 4.1.1 | Nastanek in zgodovina jezika XML..... | 9 |
| 4.1.2 | Specifikacija jezika XML..... | 11 |
| 4.1.3 | Uporaba jezika XML..... | 11 |
| 4.2 | Označevalni jezik XSL | 12 |
| 4.3 | Programski jezik java | 14 |
| 4.4 | Souporaba jave in označevalnega jezika XML..... | 14 |
| 4.5 | Program za grafično oblikovanje Oracle BI Publisher | 15 |
| 4.6 | Okolje za programiranje v javi Oracle JDeveloper 10g | 18 |
| 5 | PROGRAMSKA NADGRADNJA PRIPRAVE POTNIH NALOGOV | 21 |
| 5.1 | Razvoj programske nadgradnje..... | 23 |
| 5.2 | Priprava grafičnih elementov potnih nalogov..... | 26 |
| 5.3 | Povezava v obstoječi sistem s programom Oracle JDeveloper 10g..... | 33 |

| | | |
|---|------------------------------------|----|
| 6 | OPRAVLJENO DELO IN REZULTATI | 36 |
| 7 | ZAKLJUČEK | 38 |
| 8 | LITERATURA | 40 |

KAZALO SLIK

| | |
|---|----|
| Slika 1: Shema omrežja v podjetju..... | 1 |
| Slika 2: Transformacija XSLT | 13 |
| Slika 3: Program za grafično oblikovanje Oracle BI Publisher | 17 |
| Slika 4: Razvojni program Oracle JDeveloper 10g..... | 20 |
| Slika 5: Strežnik za upravljanje z internetnimi aplikacijami Apache Tomcat..... | 21 |
| Slika 6: Shema dosedanje in nove aplikacije za pripravo potnih nalogov | 22 |
| Slika 7: Programiranje z razvojnim programom Oracle JDeveloper 10g | 24 |
| Slika 8: Uporabe funkcije <i>Insert</i> , ki omogoča prikaz podatkov | 26 |
| Slika 9: Prikaz uporabe funkcije <i>Table/Form</i> | 27 |
| Slika 10: Potni nalog, oblikovan s programom Oracle BI Publisher | 28 |
| Slika 11: Prikaz potnega naloga v formatu PDF..... | 29 |
| Slika 12: Prikaz potnega naloga v obliki HTML | 30 |
| Slika 13: Prikaz potnega naloga v programu Microsoft Excel | 31 |
| Slika 14: Pregled potnega naloga v programu Microsoft PowerPoint | 32 |
| Slika 15: Vnašanje nadgrajenih datotek v javin strežnik..... | 34 |
| Slika 16: Shranjevanje datotek z datotečnim poslovođnikom | 35 |
| Slika 17: Prikaz potnega naloga v nadgrajeni obliki..... | 37 |

1 UVOD

S prodorom računalniške in telekomunikacijske tehnologije v večino dejavnosti sodobne družbe postaja vse pomembnejša možnost preproste in učinkovite izmenjave podatkov med računalniškimi sistemi. Pri tem igrajo bistveno vlogo standardi, saj edino ti, če so dovolj široko sprejeti, lahko zagotovijo preprosto komunikacijo in izmenjavo podatkov med sistemi z različno strojno in programsko opremo.

V tej diplomski nalogi obravnavamo primer izboljšave računalniške izmenjave podatkov med uporabniki v organizaciji. Naloga je bila opravljena v podjetju Iskra Avtoelektrika d.d. v sektorju Organizacija in informatika. Obsegal je programsko nadgradnjo priprave potnih nalogov v podjetju. Potni nalogi so se pred tem prikazovali le v spletni obliki in so bili zato nepregledni. Tudi tiskanje ni bilo v enotni obliki. Uporabniki so želeli, da bi bili potni nalogi dostopni v obliki PDF, saj bi tako vsi tiskalniki v podjetju tiskali dokumente v enaki obliki. S tem bi dosegli urejenost in preglednost obravnave potnih nalogov.

Iskra Avtoelektrika d.d. ima za podporo internega poslovanja razvit strežnik z lastno programsko opremo, ki deluje na osnovi tehnologije označevalnega jezika XML (angl. Extensible Markup Language). Izvorna koda programov na strežniku je zapisana v označevalnem jeziku XML. S pomočjo pretvorbe med označevalnim jezikom XML in standardom XSLT dobimo podatke za izhodni prikaz v jeziku CML. CML (angl. Concept Markup Language) je zelo preprost jezik za opisovanje spletnih dokumentov. Poenostavlja pošiljanje informacij, predstavljenih s simboli, preko interneta. Pridobljeni CML dokument nato z dodatnimi pretvorbami pripravimo za prikaz v spletnem brskalniku.

Za potrebe zahtevnejših prikazov in izpisov potnih nalogov so želeli v podjetju uvesti označevalni jezik XSL-FO (angl. Extensible Stylesheet Language Formatting Objects). Program za grafično oblikovanje Oracle BI Publisher je rešitev, ki omogoča izdelavo datotek v jeziku XSL-FO z urejevalnikom besedil Microsoft Word. S tem enostavno rešimo izdelavo razmeroma zahtevnih XSL-FO dokumentov. Naloga, ki jo je bilo treba opraviti, je nadgradnja obstoječega strežnika s tehnologijo XSL-FO.

Naloga je zahtevala seznanitev z označevalnima jezikoma XML in XSL-FO, s funkcionalnostjo programa Oracle BI Publisher in delovanjem Iskrinega spletnega strežnika. Nadalje smo morali poiskati možnosti združitve z novo tehnologijo in ustrezno predelati obstoječi strežnik. V ta namen smo si pomagali s spletno stranjo konzorcija W3C (angl. World Wide Web Consortium) (W3Schools, 2008).

S programsko nadgradnjo priprave potnih nalogov smo pridobili dragoceno znanje in izkušnje o standardih jezikih in razvojnih orodjih za oblikovanje in elektronsko izmenjavo dokumentov. To znanje in izkušnje bomo lahko uporabili pri prenovi drugih programov in razvoju novih programov za potrebe podjetja.

V naslednjih poglavjih opisujemo elektronsko izmenjavo podatkov in dokumentov. Predstavljamo podjetje Iskra Avtoelektrika d.d., njihovo računalniško pripravo potnih nalogov in potrebe po preglednejšem prikazovanju. Razvidne so uporabljeni jeziki in programska orodja, kot so označevalna jezika XML in XSL-FO, programski jezik java, program za grafično oblikovanje Oracle BI Publisher in okolje za programiranje v javi Oracle JDeveloper 10g. Podrobneje opisujemo programsko nadgradnjo priprave potnih nalogov, ki zajema razvojni del s programom Oracle JDeveloper 10g in grafični del z urejevalnikom besedil Microsoft Word in programom Oracle BI Publisher, ter povezavo v obstoječi strežnik s programom Oracle JDeveloper 10g. Na koncu povzemamo opravljeno delo in dosežene rezultate.

2 ELEKTRONSKA IZMENJAVA PODATKOV IN DOKUMENTOV

Po Gradišarju (2003) elektronsko oziroma računalniško izmenjavo podatkov (angl. Electronic Data Interchange) najenostavneje opredelimo kot elektronsko izmenjavanje posebej oblikovanih podatkov oziroma standardnih poslovnih dokumentov. Poteka na izvajalnem nivoju obdelave podatkov, kjer se oblikuje baza podatkov organizacije, ki igra pomembno vlogo v komunikaciji znotraj organizacije in z drugimi organizacijami. Elektronska izmenjava podatkov omogoča, da komunikacija poteka lažje in hitreje. Podjetje Iskra Avtoelektrika ima veliko hčerinskih družb po svetu in mu elektronska izmenjava podatkov veliko pomeni.

Organizacije si pri klasičnem poslovanju izmenjujejo dokumente v tiskani obliki. Te dokumente mora prejemnik ponovno vnašati v računalnik. To pomeni podvajanje dela in vir dodatnih napak. Pri elektronskem poslovanju, ki je usklajeno med obema organizacijama, je možno računalniško izmenjavanje dokumentov. Z dogovorom kako bodo podjetja šifrirala materiale, izdelke, storitve, tipe dokumentov itd. in kako bodo ti oblikovani ter se prenašali, se to lahko doseže. Če organizacija posluje z več partnerji, sklene dogovor z vsakim posebej. Rezultat je lahko množica oblikovno različnih, čeprav vsebinsko enakih dokumentov. Da bi se izognili zamudnemu, dragemu delu in viru novih napak, so se na ožjih področjih industrije in bančništva izoblikovali posebni standardi pri oblikovanju poslovnih dokumentov. Tako je možno, da se pred pošiljanjem s posebnim prevajalnikom dokument pretvori v standardno obliko. S podobnim prevajalnikom se avtomatsko prevede v dokument, ki je razumljiv prejemniku in njegovemu informacijskemu sistemu ter podatke vpiše v bazo. V razvitem svetu se vlagajo veliki napor v oblikovanje standardov za zapis dokumentov na področjih, kot je avtomobilska industrija. Mnogo programskih rešitev in paketov, ki omogočajo lažjo in cenejšo računalniško izmenjavo podatkov, je dostopnih tudi manjšim organizacijam.

Elektronska izmenjava podatkov ima številne prednosti, ki se kažejo v neposrednih in posrednih koristih.

Neposredne koristi so:

- zmanjšanje stroškov pri pripravi in pošiljanju dokumentov,
- hitrejše komuniciranje,
- povečanje zanesljivosti prenosa podatkov.

Posredne koristi pa so:

- krajšanje poslovnih ciklov,
- zmanjševanje zalog in nanje vezanih obratnih sredstev,
- lažja izvedba koncepta ravno ob pravem času (angl. just in time),
- večja konkurenčnost pri pridobivanju poslov itd.

S povezovanjem računalnikov dobimo računalniška omrežja. V omrežja se lahko vključujejo računalniki različnih proizvajalcev in z različnimi operacijskimi sistemi. V omrežje pa se lahko povezujejo tudi druge naprave. Računalniška omrežja delimo na dve vrsti: lokalna ali LAN (angl. Local Area Network) in prostrana ali WAN (angl. Wide Area Network). Posameznim uporabnikom omogoča lokalno omrežje uporabo in dostop do strojnih virov in podatkov. To je posebno aktualno v velikih poslovnih in drugih sistemih.

V Iskri Avtoelektriki imajo obe računalniški mreži, tako lokalno kot prostrano. Njihov intranet, ki omogoča tudi pripravo potnih nalogov, deluje samo na lokalni mreži. Na intranetu dobijo zaposleni vse potrebne informacije v zvezi s kadrovanjem, z internim imenikom, internim časopisom itd. Veliko dela v Iskri Avtoelektriki opravijo hitro in poceni z intranetom.

Na elektronsko izmenjavo podatkov, še posebej pa na elektronsko trgovanje, je močno vplival označevalni jezik XML. V njem zapisani dokumenti omogočajo mnogo enostavnejšo izmenjavo podatkov, hkrati pa zaradi enotne infrastrukture lažje procesiranje. Izboljšava je predvsem tam, kjer se uporabljajo podatki z zapletenimi strukturami. XML postavlja skupni imenovalec za vse takšne formate, ki so bili v

preteklosti vedno specifični in s tem za mnoge razvijalce programske opreme zapleteni. Z razvojem elektronskega trgovanja pričakujemo pospešeno rabo XML-ja v elektronskih transakcijah, ki prenašajo podatke o nakupih med različnimi sistemi (Gradišar in drugi, 2005).

3 PRIPRAVA POTNIH NALOGOV V ISKRI AVTOELEKTRIKI

3.1 O podjetju

Začetki delovanja Iskre Avtoelektrike segajo v leto 1960, ko je bila ustanovljena poslovna enota Avtoelektrike s sedežem v Šempetru pri Gorici, vanjo pa prenesena proizvodnja izdelkov avtoelektrike. V svoji preteklosti so doživljali hiter razvoj, oblikovali so jih izzivi domačega in tujih trgov, na katere so bili usmerjeni od vsega začetka. Ob proizvodnji so se kasneje razvile spremljajoče dejavnosti, posledično danes v Iskri Avtoelektriki v celoti in samostojno obvladujejo vse poslovne procese.

Z letom 1990 je Iskra Avtoelektrika začela poslovati kot družbeno podjetje, od leta 1991 je registrirana kot delniška družba. Leta 1997 je bilo zaključeno lastninsko preoblikovanje podjetja. V letu 2004 je Iskra Avtoelektrika svoje delnice uvrstila na organiziran trg vrednostnih papirjev na Ljubljanski borzi.

Iskra Avtoelektrika je globalna dobaviteljica zaganjalnikov in generatorjev za motorje z notranjim zgorevanjem, električnih pogonskih in mehatronskih sistemov ter delov zanje. V podjetju je zaposlenih preko 2.500 ljudi, kar jih ob ostalih kazalnikih uvršča med največje slovenske industrijske družbe. Danes opravijo preko 94% prodaje na razvitih trgih Evropske unije, Severne Amerike ter ostalih razvitih in stabilnih trgih sveta. To jih uvršča med največje slovenske izvoznike. Z lastno prodajno mrežo in podjetji doma in v tujini podjetje posluje čedalje bolj globalno.

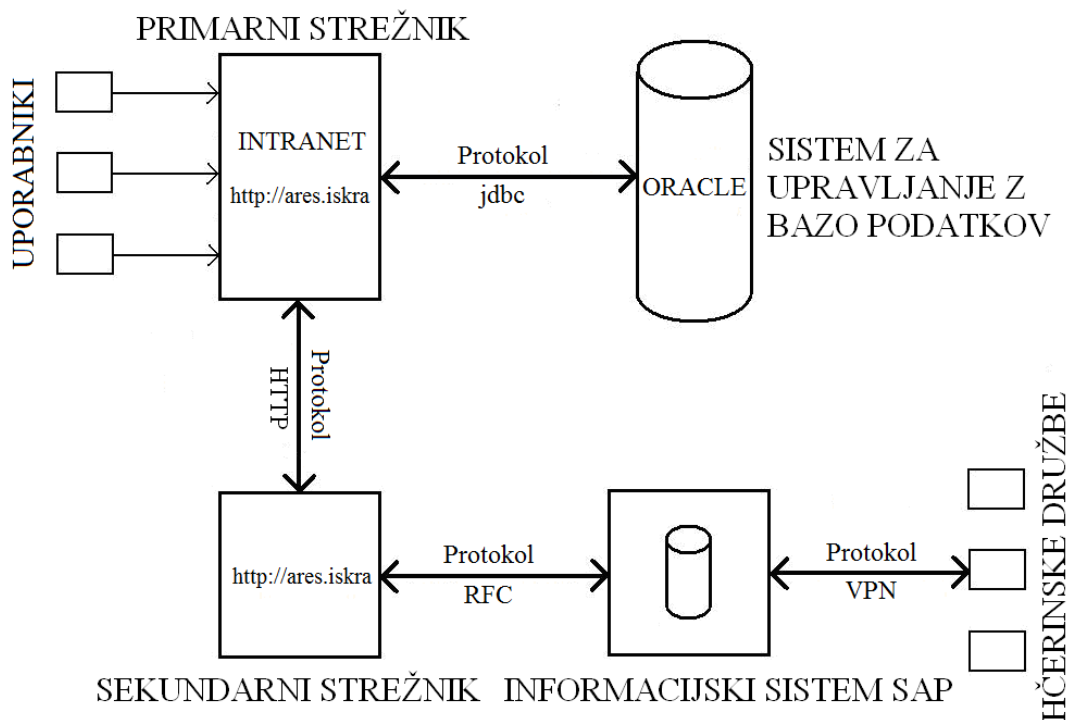
Iskra Avtoelektrika razvija, proizvaja in trži globalno z lastno proizvodnjo in prodajno-distribucijsko mrežo, ki poleg podpore industrijskim odjemalcem trži tudi širok izbor proizvodov. Iskra Avtoelektrika je prepoznavna po inovativnosti, trajnostnem razvoju, kakovosti proizvodov in procesov, poslovni odličnosti ter veliki tržni in razvojni podpori svojim odjemalcem. Prepoznavnost temelji na strokovno usposobljenih ljudeh in prožnih poslovnih sistemih.

V podjetju je pet organiziranih strateških poslovnih enot (SPE): Avtoelektrika, Pogonski sistemi, Mehatronika, Sestavni deli in Trading. Poleg tega podjetje sestavlja še devet direktij. Ena od teh je direktija Organizacija in informatika, v kateri je bil opravljen praktični del diplomske naloge.

Iskra Avtoelektrika ima po svetu več hčerinskih družb, s katerimi neposredno posluje. Te so v Braziliji, na Japonskem, v Iranu, Italiji, Nemčiji, Franciji ter Bosni in Hercegovini (Iskra Avtoelektrika, 2008).

3.2 Računalniška priprava potnih nalogov

Iskra Avtoelektrika ima obsežno računalniško omrežje. Imajo primarni spletni strežnik, ki uporablja njihovo bazo podatkov, in sekundarni spletni strežnik, ki predstavlja povezavo med primarnim spletnim strežnikom in poslovnim informacijskim sistemom SAP. Sekundarni spletni strežnik komunicira s sistemom SAP, ki je glavni dejavnik pri povezavi in izmenjavi podatkov med podjetjem in hčerinskimi družbami. Na sliki 1 je prikazana shema omrežja v podjetju.



Slika 1: Shema omrežja v podjetju

V podjetju potne naloge že nekaj let pripravljajo v elektronski obliki. Vsi potni nalogi so dostopni na intranetu družbe. Pred leti so potne naloge pripravljali še ročno, kar je pomenilo večjo porabo časa in denarja. Z inovativno računalniško podporo so ta problem odpravili. Računalništvo, internet in elektronsko poslovanje so prinesli v

podjetje izjemno novost. Delo je postalo lažje, hitrejše in z manj napakami, kar je izjemno pomembno tudi pri delu z dokumenti, kot so potni nalogi.

3.3 Potrebe po preglednejšem prikazovanju potnih nalogov

V Iskri Avtoelektriki potne naloge in različne druge dokumente urejajo na intranetu. Predlog potnega naloga vnese delavec v računalnik in ga pošlje nadrejenemu. Nadrejeni predlog pregleda in ga odobri ali zavrne. Če predlog zavrne, ga mora delavec po njegovih navodilih dopolniti. Ko je predlog potrjen, gre lahko delavec na službeno potovanje. Po vrnitvi s službenega potovanja delavec v potni nalog vnese obračunske podatke, ga natisne, priloži dokazila o stroških in vse skupaj odda v računovodstvo. Tam potni nalog obračunajo in arhivirajo. Stroške, ki so nastali pri službenem potovanju, povrnejo delavcu pri plači.

Na osnovi pripomb uporabnikov, da se potni nalogi ne tiskajo pregledno, so se v podjetju odločili, da bodo pripravo potnih nalogov nadgradili. Težava je nastala pri tiskalnikih, saj imajo v podjetju veliko različnih modelov, ki tiskajo vsak v svoji obliki. Nesmiselno bi bilo vse potne naloge na novo pisati in jih nato vnašati na intranetno stran. To bi bila izguba časa in denarja. Odločili smo se za enotno predstavitev nalogov v formatu PDF, ki ga zaznava večina njihovih tiskalnikov. Na voljo je bilo več programov, s katerimi bi težavo lahko rešili. Izbrali smo programa Oracle BI Publisher in Oracle JDeveloper 10g, saj se je izkazalo, da bo z njima delo lažje in hitreje opravljeno. Tako smo s programom Oracle BI Publisher in urejevalnikom besedil Microsoft Word oblikovali potni nalog. Razvojni program Oracle JDeveloper 10g nam je služil pri pripravi programskih datotek, ki so ključnega pomena pri pripravi potnih nalogov in njihovem vnosu v obstoječi aplikacijski strežnik. S tema dvema programoma smo tako na enostaven način omogočili pregledno pripravo in enotno izpisovanje potnih nalogov.

4 UPORABLJENI JEZIKI IN PROGRAMSKA ORODJA

V nadaljevanju predstavljamo jezike in programska orodja, ki smo jih uporabljali pri programski nadgradnji potnih nalogov. Opisani so označevalni jezik XML, njegov nastanek in zgodovina, lastnosti ter uporabnost. Opisujemo tudi označevalni jezik XSL in programski jezik java. Predstavljeni sta programska orodja, ki smo ju uporabljali. To sta program za grafično oblikovanje Oracle BI Publisher in razvojni program Oracle JDeveloper 10g.

4.1 Označevalni jezik XML

Označevalni jezik XML (angl. Extensible Markup Language) je nov jezik, ki je bil razvit zato, da so z njim informacije pregledneje predstavljene in bolj razumljive. V začetku leta 1998 ga je W3C (angl. World Wide Web Consortium) dal v končno uporabo. Jezik se je zelo hitro razširil v znanosti in industriji. W3C upa, da bo XML rešil nekaj največjih problemov svetovnega spleta, med njimi sta tudi počasnost nalaganja spletnih strani in neučinkovito iskanje informacij.

Označevalni jezik XML je nastal, ker je pri tradicionalnem jeziku za opis spletnih dokumentov HTML prišlo do težav. Jezik HTML opisuje le, kako naj brskalnik uredi besedilo in slike, zato je zastarel. Ko podjetja želijo sprejemati naročila, nastane težava pri pošiljanju zapisov, saj HTML ni bil razvit za takšne namene in ne omogoča izmenjave podatkov.

Težave nastanejo tudi, ko pri iskanju na spletu na osnovi določene ključne besede dobimo preveč informacij, ki so nekoristne. Vzrok za to je, da večina iskalnikov na svetovnem spletu deluje na osnovi obsežnih indeksov. Z jezikom XML je dana možnost, da se indeksirajo samo pomembne oznake v besedilu, ne pa celotna besedila. V jeziku HTML razen naslova namreč ni drugih oznak, ki bi določale pomen besedila (Curk in Budimir, 1999).

4.1.1 Nastanek in zgodovina jezika XML

Leta 1996 je konzorcij W3C začel iskati rešitev za uvedbo novega jezika. Kot osnovo je vzel metajezik SGML (angl. Standard Generalized Markup Language), s katerim lahko opišemo drug jezik. SGML je razvila mednarodna organizacija ISO

(angl. International Organization for Standardization) leta 1986, pod oznako ISO 8879. Kot osnovo so vzeli jezik GML (angl. Generalized Markup Language), ki so ga že konec šestdesetih let razvijali pri IBM.

Slabosti jezika SGML so, da je presplošen in preobsežen. Uporabljajo ga v velikih poslovnih sistemih in vladnih organizacijah. V nekaterih pogledih velja, da XML v kompleksnih sistemih, na primer v letalski industriji, jezika SGML ne bo mogel nadomestiti. Uporabnikom jezika SGML (npr. v knjižnično-informacijskih sistemih), ki bodo želeli preiti na XML, to ne bo težko, saj je XML podmnožica jezika SGML. HTML so razvili v Ženevi leta 1990 iz jezika SGML. HTML ni metajezik, temveč le aplikacija jezika SGML. HTML je zelo enostaven, dostopen in najbolj uporabljan jezik za opisovanje dokumentov doslej. V svetovnem spletu pa je prišlo do novih potreb in povpraševanj, zato HTML ne zadošča več zahtevam.

Skupina strokovnjakov se je leta 1996 zbrala v W3C s ciljem, da pripravijo SGML za uporabo na svetovnem spletu. Februarja leta 1998 se je to tudi zgodilo, saj se je dobesedno rodil XML. V ta projekt so vključeni največji proizvajalci programske opreme kot so Microsoft, Netscape, Adobe, IBM, Lotus in Sun.

XML je razširjen označevalni jezik oziroma metajezik. Z njim je mogoče definirati nove označevalne jezike. Uporablja se predvsem tam, kjer HTML odpove. V jeziku HTML ne moremo opisovati matematičnih enačb, kemijskih formul ipd., razen če jih ne vstavljamo v dokumente kot slike. Slike se na spletu prepočasno nalagajo. Podatke v jeziku XML lahko primerjamo tudi z metapodatki v klasičnih bazah podatkov. Tam nam metapodatki povedo, kaj pomenijo podatki.

Za ustvarjanje preprostih, osebnih spletnih strani HTML zadošča in ga XML ne bo zamenjal. Vendar v poslovnih aplikacijah, kjer se morajo informacije prelivati od strežnikov do odjemalcev v velikih količinah v vse smeri, pride v poštev prav XML (Curk in Budimir, 1999).

4.1.2 Specifikacija jezika XML

Označevalni jezik XML je enostaven standardiziran jezik, ki ga lahko bereta človek in računalnik. Napisan je v obliki teksta, kar zagotavlja dokumentom dolgo življenjsko dobo, za razliko od dokumentov, pripravljenih npr. z urejevalnikom besedil Microsoft Word, kjer dokumenti zaradi spreminjanja verzij programa s časom zastarijo in jih ni več moč uporabiti.

XML uporablja elemente za opis vsebine dokumentov. Vsak element je opremljen z začetno oznako (angl. start tag) in končno oznako (angl. end tag), ki oklepata vsebino npr. `<ime>Rok</ime>`. Izjeme so prazni elementi, ki nimajo vsebine npr. `<referenčna-številka></referenčna-številka>`. Elementi imajo lahko tudi attribute, ki natančneje določajo njihov pomen npr. `<naslov vrsta="stalni">`.

V specifikaciji jezika XML obstajajo enostavna, toda striktna pravila. Tako oznake, razen pri praznem elementu, nastopajo v parih in oklepajo vsebino elementa. Naslednje pravilo je, da so lahko ti pari vgnezdjeni do poljubnega nivoja, prekrivanje kot npr. `<ime>Ciril<priimek>Krajnc</ime></priimek>`, pa ni dovoljeno. Dokument mora vsebovati tudi izhodiščni element (angl. root), ki zajema vse preostale elemente, npr. `<naročniki> ... </naročniki>`. Tako dobimo t.i. drevesno strukturo dokumenta, ki jo je mogoče enostavno implementirati z računalniškimi programi.

Oznake elementov, njihova vsebina in atributi so lahko v kakršnikoli pisavi, ki jo podpira standard *Unicode*, ki vključuje večino pisav sveta. V dokumentih XML se da pisave brez težav menjavati. Moramo pa biti pozorni na velike in male črke, saj npr. `<ime>` ni isto kot `<lme>` (Curk in Budimir, 1999).

4.1.3 Uporaba jezika XML

Označevalni jezik XML omogoča mednarodno elektronsko založništvo, neodvisno od medija. Industrijam omogoča definiranje od opreme neodvisnih protokolov za izmenjavo podatkov, posebno pri elektronskem poslovanju. Razpošilja informacije v obliki, primerni za takojšnje procesiranje. Poenostavlja in ceni obdelave podatkov. Ljudem omogoča prikazovanje informacij po njihovih željah. Omogoča, da podatke opišemo z metapodatki, kar nam je v pomoč pri iskanju informacij.

Tehnologija XML omogoča, da se v enem dokumentu lahko prepleta več jezikov XML. Med seboj jih je treba ločiti, da lahko programi procesirajo le njim namenjene oznake, drugih pa ne upoštevajo. S tem se ukvarja standard *Namespaces*. Z njegovo pomočjo lahko enostavno razlikujemo imena elementov in atributov v dokumentih XML, ne glede na to, od kod prihajajo.

Tehnologija XML je v svetovnem spletu hitro zaživela in se začela uporabljati v raznih aplikacijah. K temu so veliko pripomogli novejši programi za delo z jezikom XML. Vsi ti programi so razviti v javi in večinoma brezplačni. Največje zasluge za to imata IBM in fundacija Apache. Tehnologiji XML in java sta zelo pomembni in uporabni prav zaradi svoje komplementarnosti.

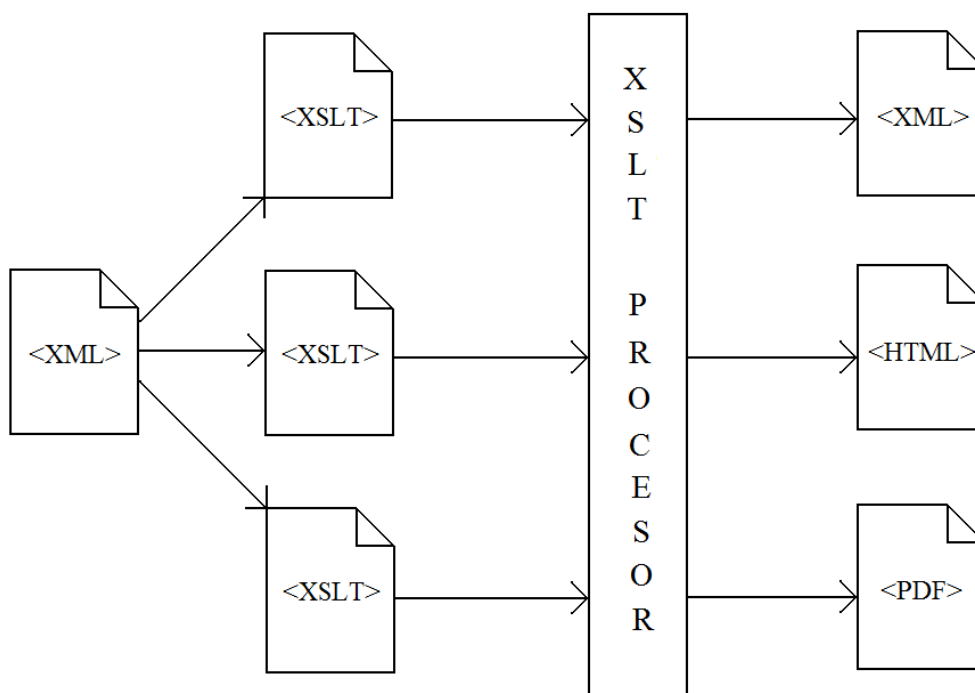
Jezik XML se največkrat uporablja pri pretvorbi dokumentov v drug format ali drugo datoteko XML. Pretvorbo dokumenta XML v drug dokument XML določimo v ločeni datoteki, kjer opišemo transformacijo elementov XML po specifikaciji XSLT (Curk in Budimir, 1999).

4.2 Označevalni jezik XSL

Označevalni jezik XSL (angl. Extensible Stylesheet Language) je jezik za oblikovanje in pretvorbo dokumentov XML, ki je sestavljen iz treh delov: XSLT, XPath in XSL-FO. XSLT (angl. Extensible Stylesheet Language Transformations) je deklarativni programski jezik, namenjen pretvorbi dokumenta XML v drug dokument XML ali v enega od predstavitvenih jezikov, kot so HTML, PDF, RTF. Izhaja iz standarda XSL, ki je bil namenjen tako pretvorbi XML dokumentov za izmenjavo kot oblikovanju podatkov iz dokumentov za prikaz. Ker gre pri tem za dve dokaj različni stvari, je bil kasneje sprejet standard XSLT, ki naj bi služil samo za pretvorbo podatkov za njihovo izmenjavo. Kljub temu ga je še vedno moč uporabiti tudi za oblikovanje vsebine za prikaz (XSL Transformations, 2008). Za naslavljanje delov dokumentov XML in enostavno povpraševanje po dokumentih XML služi jezik XPath (angl. XML Path Language) (XML Path Language, 2008). XSL se po novem imenuje XSL Formatting Objects (XSL-FO), da ga lažje ločimo od standarda XSLT. XSL-FO (angl. Extensible Stylesheet Language Formatting Objects) služi za določanje semantike oblikovanja (sloga) dokumentov XML.

Označevalni jezik XSL-FO prikaže izgled dokumenta na zaslonu ali v tiskanem dokumentu (Extensible Stylesheet Language, 2008).

Pri transformacijah XSLT sodelujejo trije dejavniki: vhodni dokument XML, transformacija XSLT in procesor XSLT. Vhodni dokument XML vsebuje podatke, ki jih želimo pretvoriti v neko drugo obliko ali prikazati na zaslonu. Transformacija XSLT je predpis, ki določa, kako naj se vhodni dokument preoblikuje. Procesor XSLT je program, ki izvede proces. Izhod iz procesa je nov dokument XML, ki vsebuje podatke vhodnega dokumenta, oblikovane na način, ki ga predpisuje transformacija XSLT. Postopek je shematično prikazan na sliki 2.



Slika 2: Transformacija XSLT

XSLT temelji na jeziku XML, kar pomeni, da ukaze predstavljajo XML elementi. Za izraze uporablja enostaven, a zelo zmogljiv jezik XPath. Čeprav jezik XSLT vsebuje le nekaj deset ukazov, je v kombinaciji z jezikom XPath izjemno zmogljiv. Velika prednost jezika XSLT pri generiranju XML dokumentov je ravno v tem, da je tudi sam XSLT v obliki XML. To pomeni, da sintaktično pravilna XSLT transformacija vedno generira pravilno oblikovan XML, kar močno zmanjša možnost napak (Microsoft XML Core Services, 2008).

4.3 Programski jezik java

Java (Mesojedec in Fabjan, 2004) je predmetno usmerjen programski jezik, ki ne pozna samostojnih podprogramov in funkcij. Java je zelo enostavna in podobna programskima jezikom C in C++. Je tudi robustna in varna. Nima kazalcev, ima avtomatsko upravljanje s pomnilnikom in varnostne mehanizme. Java API so skupine razredov (paketov), ki vsebujejo podporo za večnitnost, grafiko ter delo z mrežo in podatkovnimi bazami. Java je eden redkih jezikov, v katerih lahko pišemo programe za aktivne spletne strani. Te lahko v obliki programov v javi (*Applet*) vključujemo v spletne strani (Mrhar, 2002).

4.4 Souporaba jave in označevalnega jezika XML

Od nastanka programskega jezika java je XML najpomembnejša novost v aplikacijah svetovnega spleta, v katerih sta pomembna elektronska izmenjava in procesiranje podatkov preko omrežja, kot je npr. elektronsko poslovanje.

Jezika sta komplementarna. XML omogoča prenosljivost podatkov, java pa prenosljivost programske kode. Vsebina spletnih strani postaja z jezikom XML neodvisna od spletnih brskalnikov, z javo pa postaja neodvisna od strojne opreme tudi programska oprema.

Jezika pa nista samo komplementarna, ampak tudi podpirata drug drugega, da bi tako čim bolj izkoristila svoje potenciale. Oba podpirata standard *Unicode*, kar omogoča procesiranje mednarodnih dokumentov v jeziku XML. S podporo standarda tudi lažje izdelamo večjezične aplikacije.

Java je tako prava izbira za razvoj aplikacij v jeziku XML. Mnogi programi za procesiranje dokumentov v jeziku XML so napisani v jeziku java, kar razvijalcem ne zagotavlja samo predmetne orientiranosti in prenosljivosti kode, temveč tudi porazdeljenost njihovega dela in hitre rešitve.

Temeljna značilnost programiranja v javi je hierarhija javinih objektov. XML ponuja podoben hierarhični prikaz podatkov. Tako sta predstavitev struktur XML v javi in javinih objektov v strukturah XML združljivi, kar je uporabno pri povezovanju javinega programskega okolja z drugimi nejavinimi programskimi okolji.

Strukturirani podatki XML dobro predstavijo stanje javnih objektov, te pa lahko s pomočjo tehnologije XML uporabimo v drugih sistemih. Če imamo v nekem sistemu formatirane podatke XML, jih lahko s pomočjo tehnologije XML prenesemo v javino platformo. XML na ta način v veliki meri podpira interoperabilnost podatkov med različno strojno opremo in različnimi programi, kar je v današnjih heterogenih računalniških sistemih velikokrat pomembno.

V programih, ki uporabljajo tehnologiji XML in java, se lahko več funkcionalnosti prenese neposredno na odjemalca, kar povečuje število ukazov, ki jih lahko končni uporabnik izvede nad podatki, in hkrati zmanjšuje prenos in procesiranje podatkov prek omrežja (Curk in Budimir, 1999).

4.5 Program za grafično oblikovanje Oracle BI Publisher

Program za grafično oblikovanje Oracle BI Publisher v celoti temelji na programskem jeziku java. Je zelo enostaven za uporabo. Zgrajen je na standardnih tehnologijah, kot je XML, in omogoča uporabnikom prednost pri programskih orodjih, kot sta program za urejanje besedil Microsoft Word in program za pregledovanje dokumentov Adobe Acrobat.

Oracle BI Publisher sestavljajo:

- knjižnice API, ki so zbirke programskih komponent s številnimi funkcijami, kot so na primer enoten grafični vmesnik, mrežne povezave itd.,
- integrirana arhitektura s sistemom za upravljanje baz podatkov Oracle E-Business Suite,
- orodja za pripravo poročil, ki nudijo veliko izhodnih možnosti.

Klasičen pristop k izdelavi formatiranih dokumentov je ta, da združimo vse elemente dokumenta v entiteto, ki vsebuje podatkovne definicije in interpretacije. Tak dokument se v spletu prikazuje počasi, zato razvijalci tega pristopa ne svetujejo. Njegova slabost je tudi, da zelo poveča vzdrževanje dokumenta.

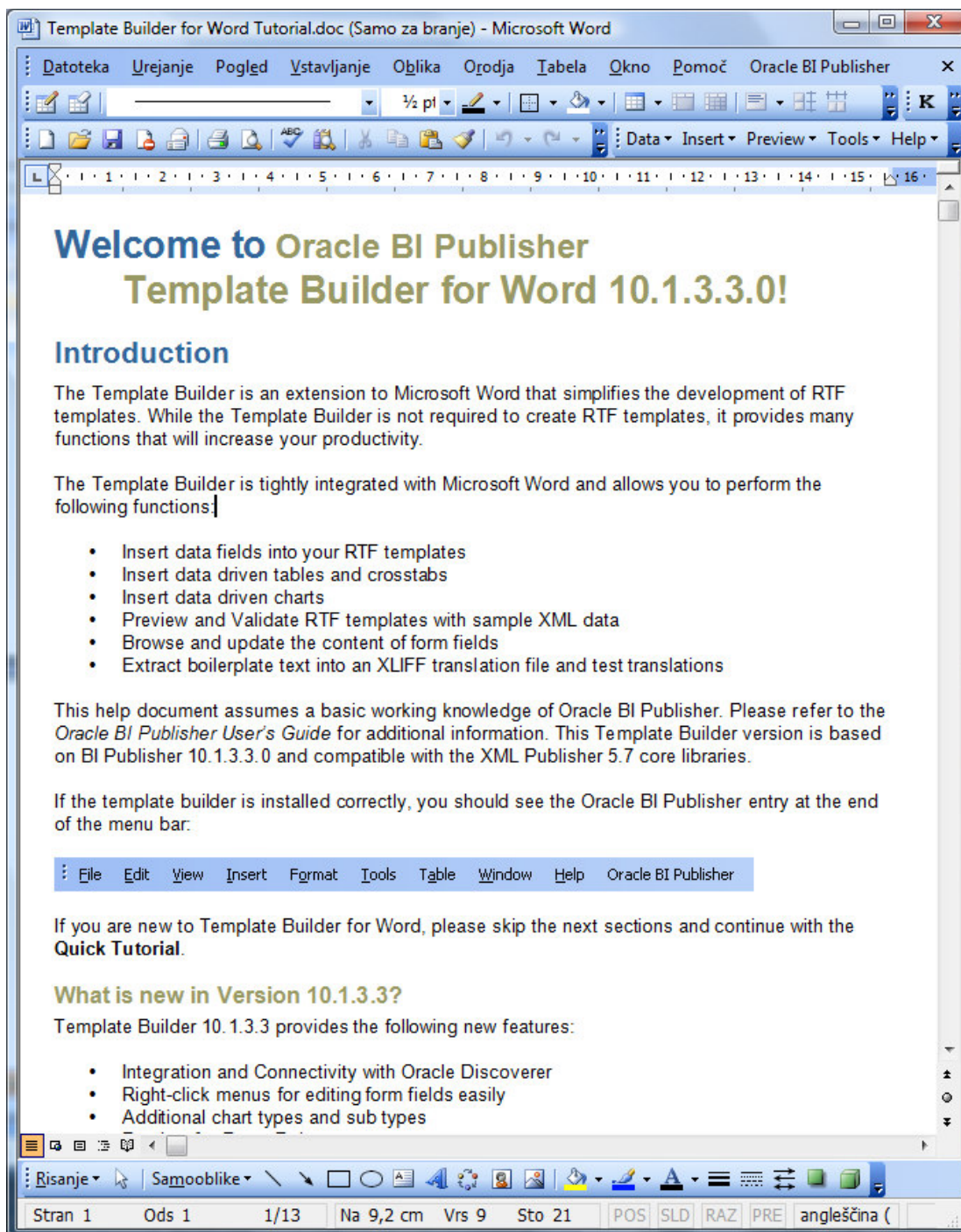
Program vsebuje tudi izdelek Oracle Reports, s katerim pripravljamo poročila. Oracle BI Publisher posebej loči komponente in jih obravnava ločeno. Ob zagonu se

komponente združijo in s programom Oracle BI Publisher ustvarijo končno obliko. Imamo podatkovno definicijo, ki podpira večino izhodnih formatov in privede do zmanjšanja stroškov vzdrževanja. Takšen pristop je zato bolj priporočljiv.

Oracle BI Publisher ima možnost objavljaja različnih dokumentov z uporabo istih podatkov. Na primer, podjetja ustvarijo vse potrebne dokumente za plačilne naloge v enem samem programu za pripravo podatkov. S tem programom pripravljajo:

- plačilne naloge za objavo na podjetniškem spletnem portalu,
- datoteke za vnos računovodskih podatkov v preglednice ali baze podatkov,
- elektronska nakazila za komunikacijo z banko,
- poročila, napisana v formatu PDF.

Z uporabo različnih izhodnih oblik je Oracle BI Publisher poenostavil zgradbo poročil za uporabnike. Program podatke lahko obdela v več korakih, dokler ne pride do ustrezne rešitve problema (Davis in drugi, 2007). Na sliki 3 je prikazan program za grafično oblikovanje Oracle BI Publisher. Na prvi strani je opis programa in njegovega delovanja. Opisane so novosti in spremembe, ki so dodane najnovejši različici programa.



Slika 3: Program za grafično oblikovanje Oracle BI Publisher (zaslonska slika z osebnega računalnika)

4.6 Okolje za programiranje v javi Oracle JDeveloper 10g

Oracle JDeveloper 10g je razvojno okolje (angl. Integrated Development Environment, IDE) za razvoj programov in internetne podpore z uporabo zadnjih industrijskih standardov, kot so programski jezik java, označevalni jezik XML in programski jezik SQL.

Razvoj in inovativna tehnologija Oracle ADF (angl. Oracle Application Development Framework) delata skupaj, z namenom poenostaviti razvoj programske opreme in skrajšati za to potreben čas. Oracle ADF je tehnologija, ki v celoti temelji na javi. S pomočjo razvitih objektov in orodij za delo z njimi nam olajša in skrajša tudi razvoj spletne programske opreme.

Oracle JDeveloper 10g podpira celoten razvojni krog z vgrajeno podporo za oblikovanje, razvoj, testiranje, usklajevanje delov programov različnih razvijalcev in povezovanje programov. Vse te naloge opravi sam integrirani razvojni program – IDE z uporabo zaporedja integriranih potez. Oracle JDeveloper 10g temelji na javinih razvojnih aplikacijah in uporablja javina programska okolja J2EE (angl. Java 2 Enterprise Edition), J2SE (angl. Java 2 Standard Edition), J2ME (angl. Java 2 Micro Edition). Z dodatkom omogoča tudi uporabo označevalnega jezika XML. Kot celovita razvojna orodjarna Oracle JDeveloper 10g poskrbi za celotno razvojno in modelno okolje, potrebno za gradnjo podatkovnih baz ter podporo podatkovnih skladišč.

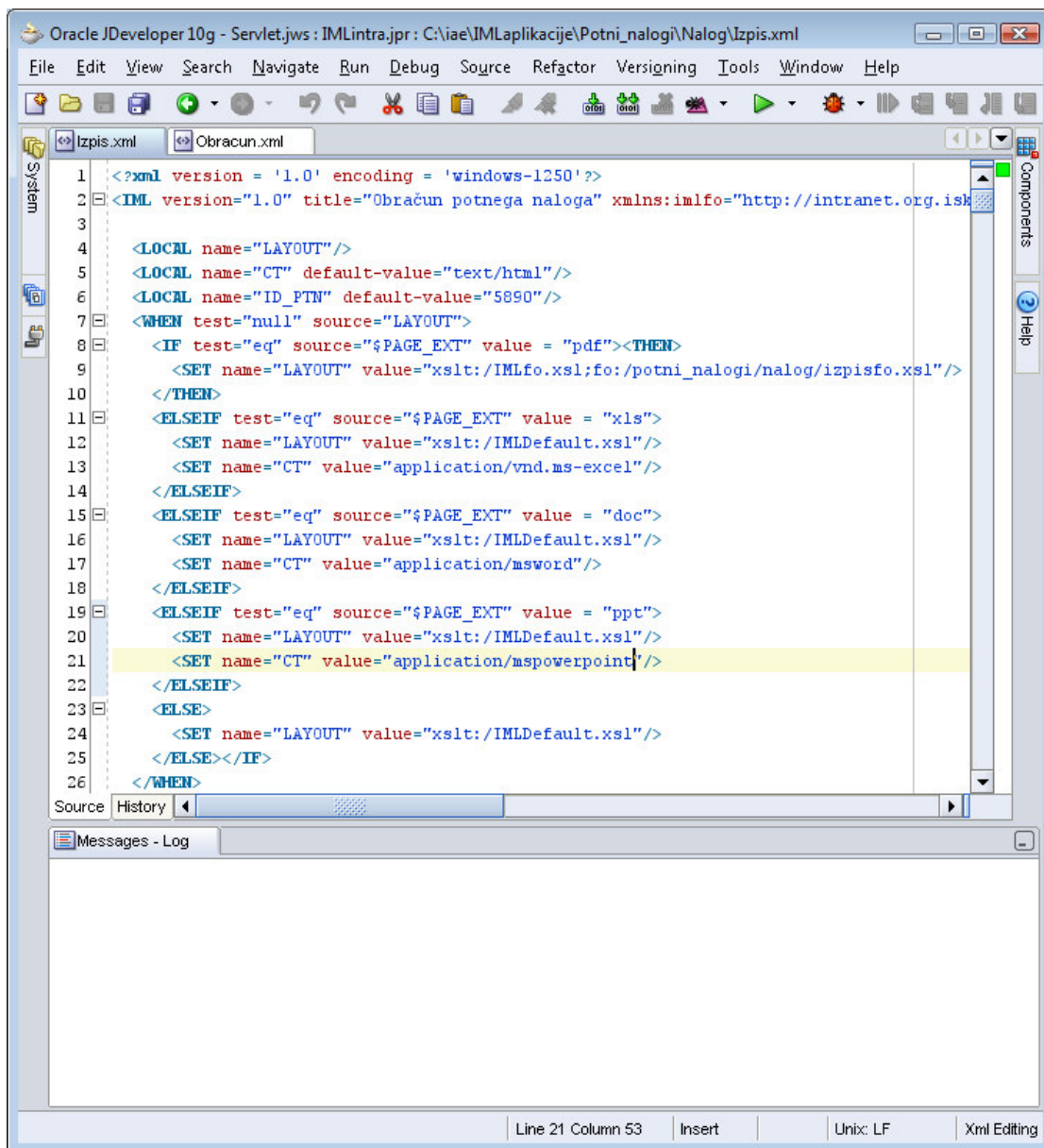
Namen programa Oracle JDeveloper 10g je narediti razvijanje v programskem okolju J2EE enostavnejše in bolj dostopno. Za doseg tega cilja se je Oracle JDeveloper 10g osredotočil na grafično in klasično programiranje v programskem okolju J2EE. Dodatne poenostavitve so zagotovljene z Oracle ADF, kar je razvojno delovno okno okolja J2EE.

Razvijalci so različno izkušeni in imajo vsak svoj najljubši razvojni pristop. Oracle JDeveloper 10g nudi veliko izbiro razvojnih pristopov. Ti vključujejo model MDA (angl. Model Driven Architecture), deklarativni razvoj in ročni razvoj. Razvijalci lahko menjajo pristop po svojih željah in potrebah. Razvoj s programom Oracle

JDeveloper 10g je možen tudi brez podatkov in lahko poteka v vsakem strojnem in programskem okolju *Java 2 Enterprise Edition*.

Oracle JDeveloper 10g je razvit v programskem jeziku java, ki deluje v operacijskih sistemih Windows, Unix in Linux. S tem zagotavlja izbiro različnih razvojnih programskih okolij. Oracle JDeveloper 10g omogoča razvijalcem uporabo zadnjih industrijskih standardov za razvoj aplikacij, ki delujejo v večini strojnih in programskih okolij.

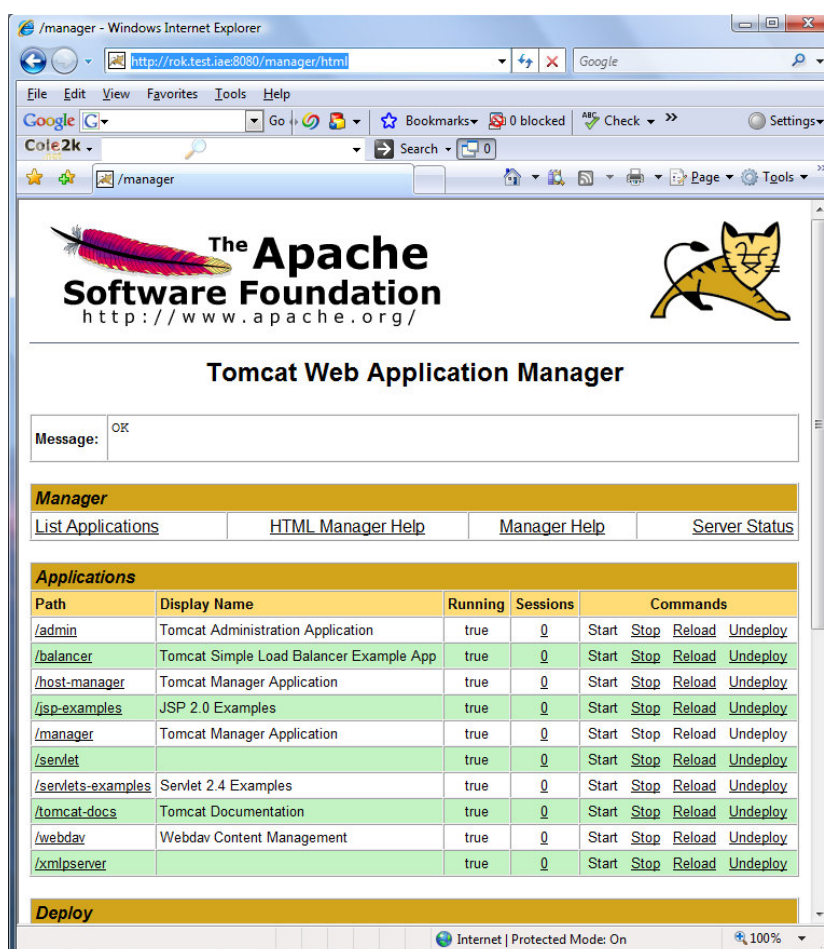
Oracle JDeveloper 10g nudi razširitev SDK (angl. Software Development Kit), ki predstavlja javino razvojno okolje in nudi razvijalcem veliko možnosti. Oracle JDeveloper 10g je zgrajen kot skupina razširitev na vrhu jedra okolja IDE. Razvijalci lahko poljubno vključijo ali izključijo razširitve po njihovi potrebi. API, ki ga uporablja skupina JDeveloper za razvoj programov, je na voljo razvijalcem in uporabnikom, kot so podjetja, ki se zanimajo za povezovanje in širitev programa Oracle JDeveloper 10g (Oracle JDeveloper 10g Reviewers Guide, 2004). Na sliki 4 je prikazan razvojni program Oracle JDeveloper 10g. Na sliki se vidi razvojno okno, v katerem je napisana koda. Nad oknom imamo raznovrstne ukaze, ki so predstavljeni kot ikone in mape. Pod razvojnim oknom imamo dodatno okno, v katerem program izpiše napake, ki jih zazna v kodi.



Slika 4: Razvojni program Oracle JDeveloper 10g (zaslonska slika z osebnega računalnika)

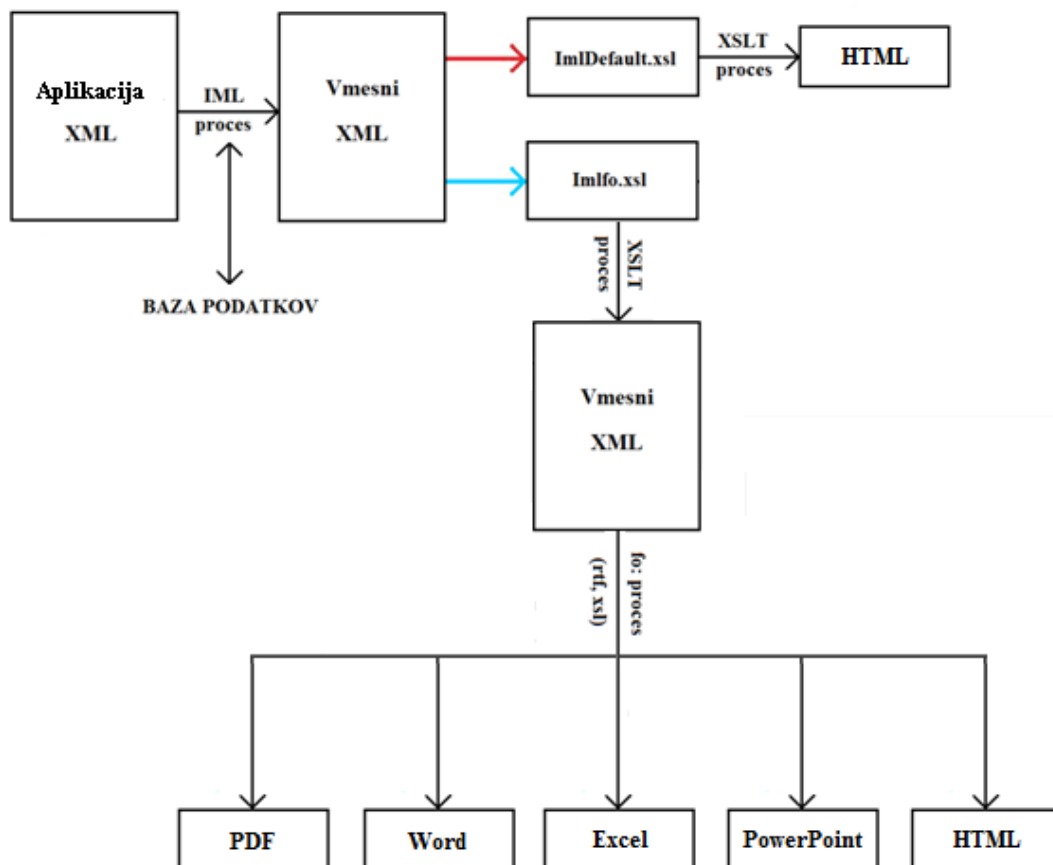
5 PROGRAMSKA NADGRADNJA PRIPRAVE POTNIH NALOGOV

Delo na programski nadgradnji priprave potnih nalogov smo začeli z nalaganjem podatkov in programov, potrebnih za izvedbo prikaza potnih nalogov v obliki PDF. Programerji iz sektorja Organizacija in informatika so na delovni računalnik, namenjen za razvoj programske nadgradnje, namestili celotno bazo podatkov z intraneta. To smo potrebovali za testiranje programske opreme med razvojem. Namestili so tudi potrebne programe, kot sta Oracle JDeveloper 10g in Oracle BI Publisher. Namestili in aktivirali so tudi strežnik Apache Tomcat. To je strežnik za upravljanje z internetnimi aplikacijami in omogoča nalaganje sprememb programske opreme na intranet. Brez tega strežnika intranet ne deluje. Na sliki 5 je prikazan strežnik Apache Tomcat v vlogi upravljalca intranetnih aplikacij, ki bo služil tudi pri nalaganju nadgradnje priprave potnih nalogov.



Slika 5: Strežnik za upravljanje z internetnimi aplikacijami Apache Tomcat (zaslonska slika z osebnega računalnika)

Naša naloga je bila nadgraditi prikaz potnih nalogov. Že ta naslov pove, da bomo obstoječo programsko podporo nadgradili in ne zamenjali, kar je velika razlika. Torej bomo dosedanje stanje nadgradili tako, da bo tudi to še vedno delovalo. Na sliki 6 je prikazana shema dosedanjega in novega stanja intranetne aplikacije za prikaz potnih nalogov. Njen osnovni gradnik je aplikacija XML, ki preko procesa IML s podatki iz baze podatkov tvori datoteko XML. IML je javin proces, ki prebere vhodni dokument XML, pridobi potrebne podatke iz baze podatkov in tvori vmesni dokument XML. Rdeča puščica prikazuje dosedanje procesiranje podatkov in izhod v edini obliki HTML. Modra puščica označuje dodatek, s katerim dobimo dodatno datoteko XML, ki s pomočjo datoteke RTF ali XSL daje kot izhod datoteke v oblikah PDF, Microsoft Word, Microsoft Excel, Microsoft PowerPoint in HTML.



Slika 6: Shema dosedanje in nove aplikacije za pripravo potnih nalogov

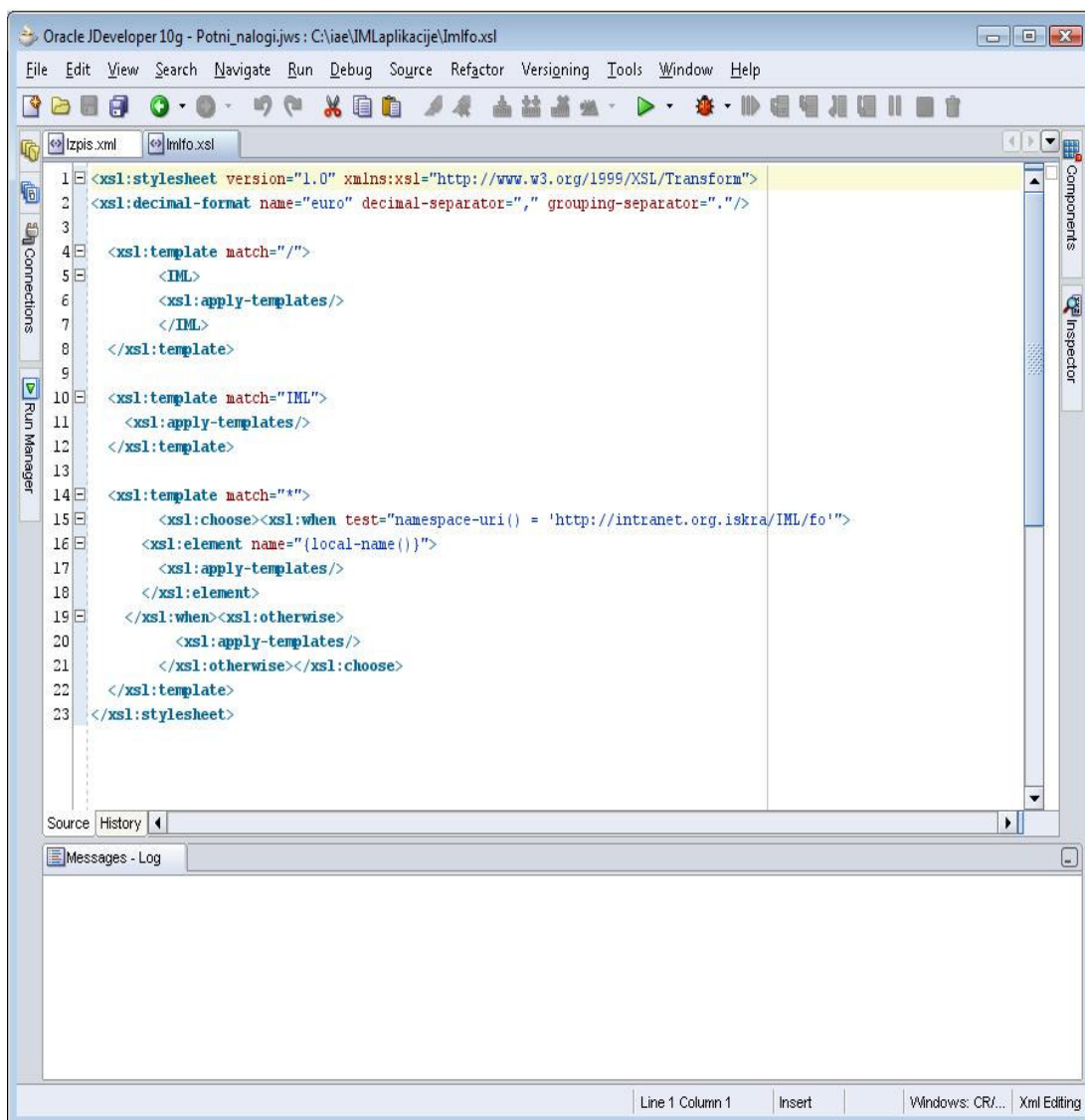
5.1 Razvoj programske nadgradnje

Obsežno področje, kjer označevalni jezik XML igra pomembno vlogo, je integracija poslovnih aplikacij. XML je dovolj zmogljiv jezik za predstavitev kakršnikoli struktur, zato lahko vse podatke izvozimo v jeziku XML in jih nato uvozimo v drugo aplikacijo. Lahko pa pred uvozom podatke pretvorimo v ustrežnejšo obliko, npr. z uporabo XSL. Enako lahko od novejših podatkovnih baz pričakujemo uvoz podatkov v obliki XML in s tem mnogo lažji prenos podatkov med podatkovnimi bazami.

Za pridobitev podatkov za potne naloge v jeziku XML smo napisali datoteko v formatu XSL. Napisana datoteka izvozi samo potrebne podatke iz datoteke *izpis.xml*, grafične vmesnike pa izloči. Grafični vmesniki so tabele, okna, slike itd. Z dodajanjem ukazov `imlfo:_____` v datoteki *izpis.xml* smo izločili vse grafične vmesnike in pridobili samo glavne podatke, ki so služili pri nadaljnjem delu. V nadaljevanju prikazujemo uporabljeni kodo za datoteko *imlfo.xsl*. Z rdečo barvo so označeni glavni stavki, ki izvozijo potrebne podatke, ki jih potrebujemo pri potnih nalogih. Na sliki 7 je prikazan razvojni program Oracle JDeveloper 10g. Na njej se vidi razvojno okno, v katerem je napisana koda. Pod razvojnim oknom imamo dodatno okno, v katerem se ob zagonu kode prikažejo sporočila o napakah.

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:decimal-format name="euro" decimal-separator="," grouping-separator="."/>

<xsl:template match="/">
  <IML>
    <xsl:apply-templates/>
  </IML>
</xsl:template>
<xsl:template match="IML">
  <xsl:apply-templates/>
</xsl:template>
<xsl:template match="*">
  <xsl:choose>
    <xsl:when test="namespace-uri() = http://intranet.org.iskra/IML/fo">
      <xsl:element name="{local-name()}">
        <xsl:apply-templates/>
      </xsl:element>
    </xsl:when>
    <xsl:otherwise>
      <xsl:apply-templates/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
</xsl:stylesheet>
```



Slika 7: Programiranje z razvojnim programom Oracle JDeveloper 10g (zaslonska slika z osebnega računalnika)

Nadalje predstavljamo del datoteke *izpis.xml*, v katerem so dodani elementi zapisani v obliki `<imlfo:_____>`. Imenski prostori (angl. namespaces) so definirani z *imlfo*, naredili smo pa tudi izhod za krmiljenje. Imenski prostor je logična zbirka razredov, struktur, vmesnikov, ki so smiselno in hierarhično urejeni. Gre za programske elemente, namenjene za pomoč pri organizaciji aplikacije. Z modro so označeni elementi *tag*, rdeči podatki pa so tisti, ki jih potrebujemo za pripravo potnega naloga.

```

<imlfo:delavec>
<tr>
  <td align="right">Delavec</td>
  <td class="value" align="right">
    <imlfo:ost>
      <VALUE source="OST"/>
    </imlfo:ost>
  </td>
  <td colspan="2" class="value">
    <imlfo:potuje>
      <VALUE source="POTUJE"/>
    </imlfo:potuje>
  </td>
</tr>
</imlfo:delavec>

```

Prikazujemo tudi dodatno izhodno kodo *Layout*, ki smo jo dodali v datoteko *izpis.xml*. S tem dodatkom smo vzpostavili povezavo med datotekama *izpis.xml* in *izpisfo.xml*. Datoteka *izpisfo.xml* je tako izvozila potrebne podatke iz datoteke *izpis.xml*.

```

<LOCAL name="LAYOUT"/>
<LOCAL name="ID_PTN" default-value="5890"/>
<WHEN test="null" source="LAYOUT">
  <IF test="eq" source="$PAGE_EXT" value = "pdf"><THEN>
    <SET name="LAYOUT" value="xslt:/IMLfo.xml;fo:/potni_nalogi/nalog/izpisfo.xml"/>
  </THEN><ELSE>
    <SET name="LAYOUT" value="xslt:/IMLDefault.xml"/>
  </ELSE></IF>
</WHEN>

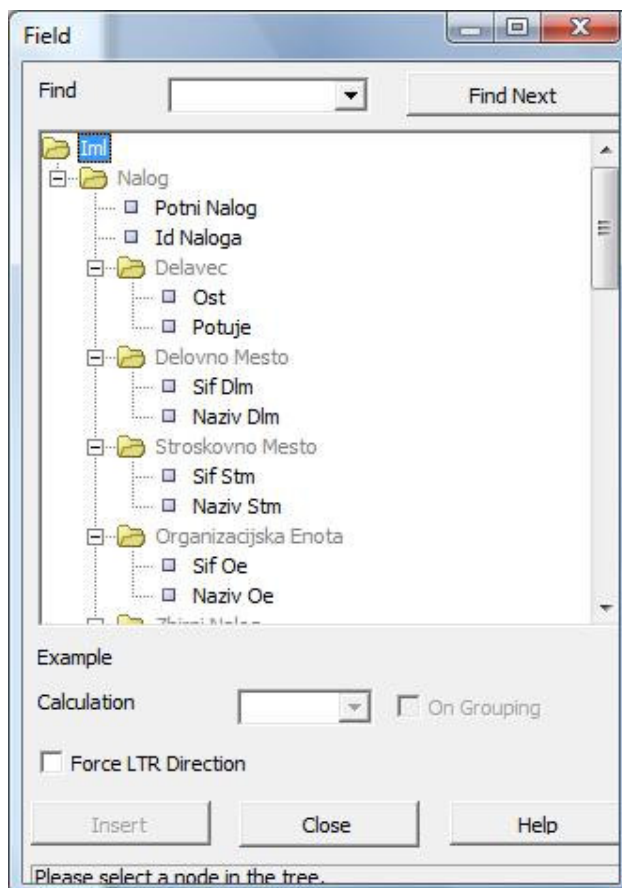
```

Ko smo končali z dodajanjem vseh opisov podatkov, elementov imlfo, smo nadgrajene datoteke, ki služijo potnim nalogom, shranili v izbrano mapo. Ko Oracle JDeveloper 10g pri shranjevanju ni opozoril na nobeno napako, smo si lahko ogledali, kako potni nalog izgleda na intranetu podjetja, kjer smo dobili prikaz kode XML. Prikazano kodo smo nato shranili v izbrano datoteko na disku kot *izpis.xml*.

Po končanem delu dela s programom Oracle JDeveloper 10g smo prešli na grafično sestavljanje potnih nalogov v urejevalniku besedil Microsoft Word s pomočjo nameščenega programa za grafično oblikovanje Oracle BI Publisher. Ta nam pomaga pri nalaganju datoteke XML *izpis.xml*, ki smo jo pravkar pridobili.

5.2 Priprava grafičnih elementov potnih nalogov

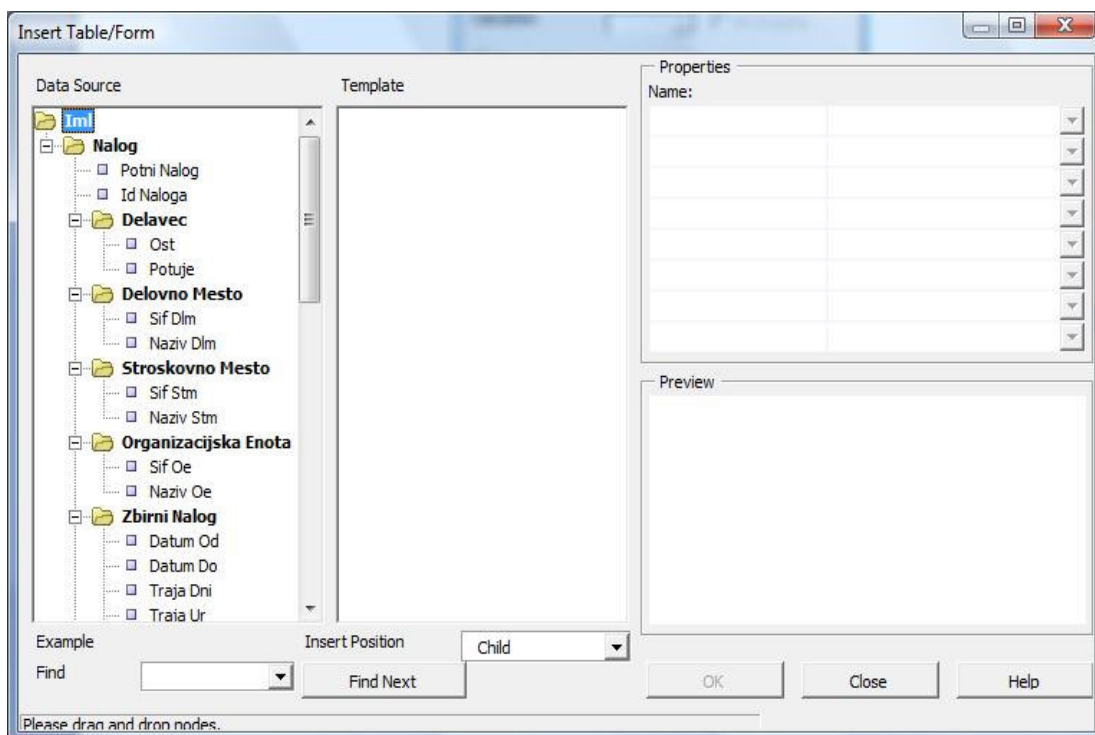
V urejevalniku besedil Microsoft Word smo vnesli podatke iz datoteke *izpis.xml*, ki smo jo pravkar preuredili. Če smo podatke pravilno naložili, se nam pokaže okence, v katerem piše *Data Loaded Successfully*. Nato lahko uporabimo funkcijo *Insert*, prikazano na sliki 8, in začnemo z oblikovanjem grafičnih elementov potnih nalogov.



Slika 8: Uporabe funkcije *Insert*, ki omogoča prikaz podatkov (zaslonska slika z osebnega računalnika)

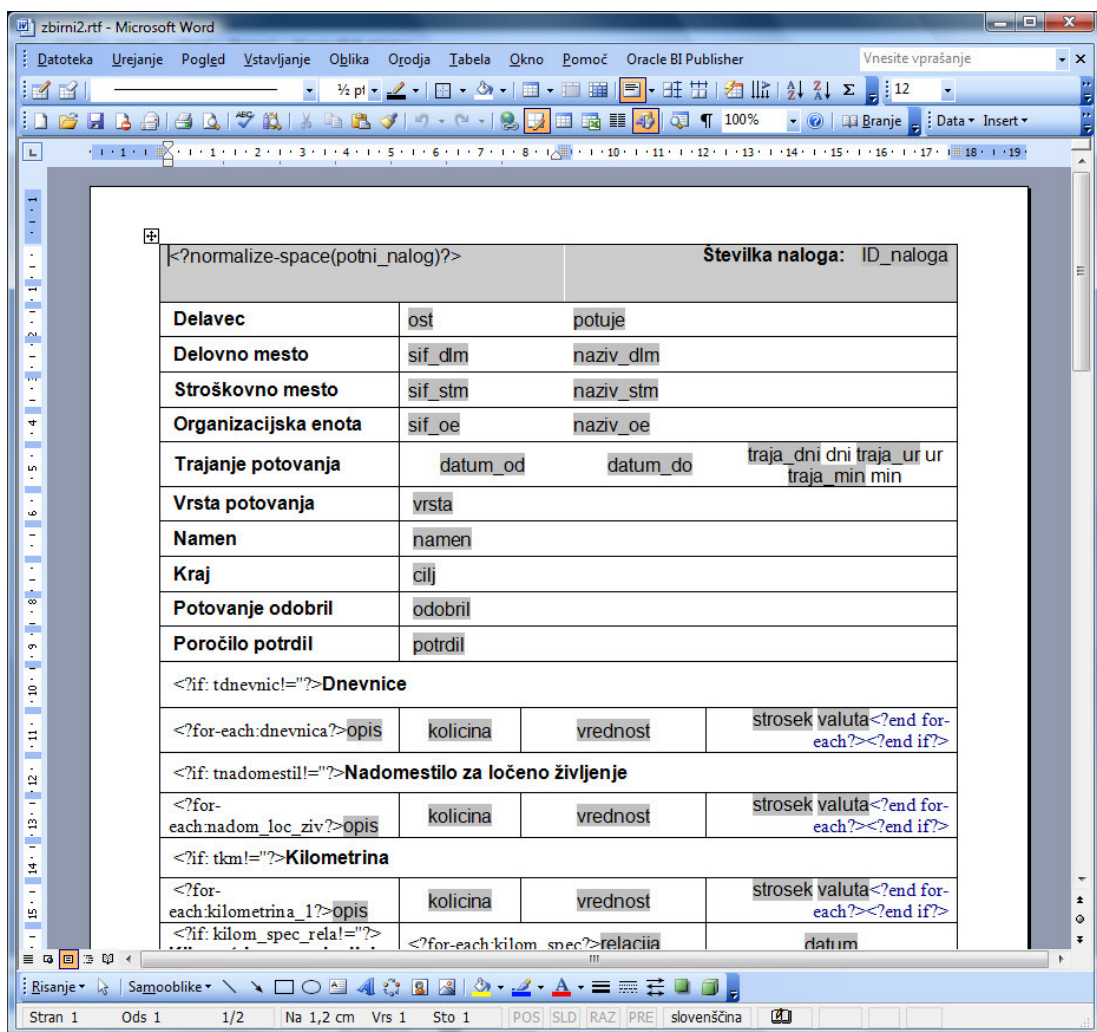
Podatke lahko vnašamo posamezno ali v sklopih. Na primer, pri podatkih o delavcu lahko vnesemo samo naziv delavca ali pa celoten sklop podatkov, ki zajema ime in priimek delavca in njegovo šifro.

Pri vnašanju podatkov s funkcijo *Insert* imamo tudi možnost vnašanja in ustvarjanja tabel, ki vsebujejo izbrane podatke. Ti dve funkciji se imenujeta *Table Wizard* in *Table/Form*, z obema souporabljammo okno *Template*. Na sliki 9 vidimo okno *Template*, v katerem se med delom prikaže tabela.



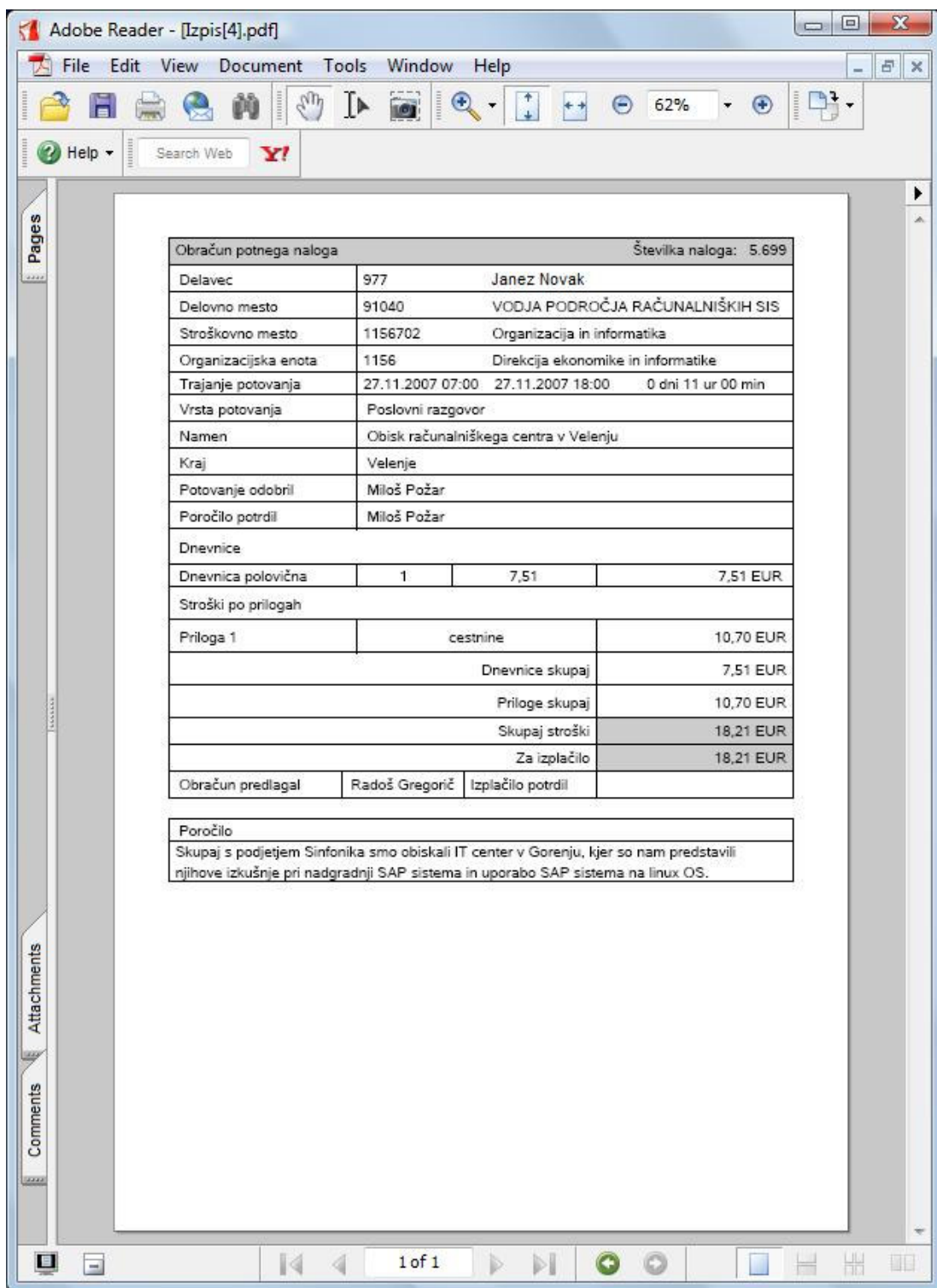
Slika 9: Prikaz uporabe funkcije *Table/Form* (zaslonska slika z osebnega računalnika)

Rezultate dosedanjega dela lahko vidimo v funkciji *Preview*, kjer imamo na voljo pet pregledov in sicer v formatih PDF, HTML, RTF, v programu za delo s preglednicami Microsoft Excel in programu za računalniške predstavitve Microsoft PowerPoint. Prednost programa Oracle BI Publisher je prav v tem, da ga podpirajo skoraj vsi programi iz zbirke Microsoft Office. Delo z njimi je zelo enostavno, tako da tudi delo s programom za grafično oblikovanje Oracle BI Publisher postane prijetno in lahko. Delo poteka v urejevalniku besedil Microsoft Word. Pri vnašanju podatkov s programom Oracle BI Publisher lahko uporabljamo tudi orodja urejevalnika besedil Microsoft Word, na primer oblikovanje tabel, spreminjanje pisave, grafične objekte, kot so slike ipd. Na sliki 10 je prikazan rezultat dela s programom Oracle BI Publisher v urejevalniku besedil Microsoft Word.



Slika 10: Potni nalog, oblikovan s programom Oracle BI Publisher (zaslonska slika z osebnega računalnika)

Ta rezultat omogoča, da prikaz deluje za vse potne naloge. Obrazcu smo dodelili ukaz <?for-each:dnevnic?>. Ta ukaz v potni nalog vpiše vse dnevnic ne glede na to, koliko jih je. Isto velja za kilometrino, priloge in poročila. Obrazcu smo dodali tudi ukaz <?if: tdnevnic!="?">, ki vpiše dnevnic, če dnevnic ni bilo, pa ostane polje prazno. Enake ukaze smo uporabili za kilometrino, priloge in poročila. Na sliki 11 je prikaz potnega naloga v formatu PDF, na sliki 12 v obliki HTML, na sliki 13 v programu za delo s preglednicami Microsoft Excel in na sliki 14 v programu za računalniške predstavitve Microsoft PowerPoint. Prikaz v programu za računalniške predstavitve Microsoft PowerPoint smo dodali samo le kot primer uporabnosti in delovanja potnega naloga tudi v tem programu, čeprav zanj ne vidimo posebne uporabne vrednosti.



Slika 11: Prikaz potnega naloga v formatu PDF (zaslonska slika z osebnega računalnika)

| Obračun potnega naloga 5.524 | | | |
|-------------------------------------|--|------------------------------------|----------------------|
| Delavec | 4767 | Janez Novak | |
| Delovno mesto | 91305 | PROJEKTANT INFORMATIKE | |
| Stroškovno mesto | 1156702 | Organizacija in informatika | |
| Organizacijska enota | 1156 | Direkcija ekonomike in informatike | |
| Trajanje potovanja | 22.10.2007 07:00 | 24.10.2007 19:15 | 2 dni 12 ur 15 minut |
| Vrsta potovanja | Izobraževanje (seminarji, tečaji, delavnice, ogledi sejmov ...) | | |
| Namen | Izobraževanje - Oracle | | |
| Kraj | Ljubljana | | |
| Način prevoza | | | |
| Potovanje odobril | Miloš Požar | Poročilo potrdil | Miloš Požar |
| Dnevnice | | | |
| Dnevnica cela | 3 | 15,02 | 45,06 EUR |
| Skupaj dnevnic | | | 45,06 EUR |
| Stroški po prilogah | | | |
| 1. | Parkirnina | | 5,20 EUR |
| 2. | Parkirnina | | 4,80 EUR |
| 3. | Parkirnina | | 4,40 EUR |
| Ostalo | | | 14,40 EUR |
| Skupaj stroški po prilogah | | | 14,40 EUR |
| Skupaj stroški | | | 59,46 EUR |

Slika 12: Prikaz potnega naloga v obliki HTML (zaslonska slika z osebnega računalnika)

| | A | B | C | D |
|----|---|---------------------------|-----------------------------|----------------------|
| 1 | Obračun potnega naloga | | | |
| 2 | 5.680 | | | |
| 3 | Delavec | 4779 | Janez Novak | |
| 4 | Delovno mesto | 23125 | PREDELAVEC | |
| 5 | Stroškovno mesto | 1113017 | ODE-8 Planetna gred | |
| 6 | Organizacijska enota | 1113 | SPE sestavni deli in orodja | |
| 7 | Trajanje potovanja | 22.11.2007 6:45 | 22.11.2007 19:45 | 0 dni 13 ur 00 minut |
| 8 | Vrsta potovanja | Drugo | | |
| 9 | Namen | predprevzem cnc stružnice | | |
| 10 | Kraj | Ljubljana-BTS | | |
| 11 | Način prevoza | | | |
| 12 | Potovanje odobril | Jože Štrukelj | Poročilo potrdil | Jože Štrukelj |
| 13 | | | | |
| 14 | Dnevnice | | | |
| 15 | Dnevnica cela | 1 | 15,02 | 15,02 EUR |
| 16 | Skupaj dnevnic | | | 15,02 EUR |
| 17 | | | | |
| 18 | Skupaj stroški | | | 15,02 EUR |
| 19 | Stroški naloga bremenijo stroškovno mesto ODE-8 Planetna gred | | | 1113017 |
| 20 | | | | |
| 21 | Za izplačilo | | | 15,02 EUR |
| 22 | | | | |
| 23 | Obračun predlagal | Aleš Kovic | Izplačilo odobril | |
| 24 | Poročilo o opravljenem potovanju | | | |
| 25 | | | | |
| 26 | predprevzem CNC stružnice | | | |
| 27 | | | | |
| 28 | | | | |
| 29 | | | | |
| 30 | | | | |
| 31 | | | | |
| 32 | | | | |
| 33 | | | | |

Slika 13: Prikaz potnega naloga v programu Microsoft Excel (zaslonska slika z osebnega računalnika)

Microsoft PowerPoint - [Izpis[3].ppt [Samo za branje]]

Datoteka Urejanje Pogled Vstavljanje Oblika Orodja Diaprojekcija Okno Pomoč

Arial 18 Načrtovanje Nov diapozitiv

| Obračun potnega naloga 5.680 | | | |
|---|---------------------------|-----------------------------|----------------------|
| Delavec | 4779 | Janez Novak | |
| Delovno mesto | 23125 | PREDELAVEC | |
| Stroškovno mesto | 1113017 | ODE-8 Planetna gred | |
| Organizacijska enota | 1113 | SPE sestavni deli in orodja | |
| Trajanje potovanja | 22.11.2007 06:45 | 22.11.2007 19:45 | 0 dni 13 ur 00 minut |
| Vrsta potovanja | Drugo | | |
| Namen | predprevzem CNC stružnice | | |
| Kraj | Ljubljana-BTS | | |
| Način prevoza | | | |
| Potovanje odobril | Jože Štrukelj | Poročilo potrdil | Jože Štrukelj |
| Dnevnice | | | |
| Dnevnic celotna | 1 | 15,02 | 15,02 EUR |
| Skupaj dnevnic | | | 15,02 EUR |
| Skupaj stroški | | | 15,02 EUR |
| Stroški naloga bremenijo stroškovno mesto ODE-8 Planetna gred | | | 1113017 |
| Za izplačilo | | | 15,02 EUR |
| Obračun predlagal | Aleš Kovic | Izplačilo odobril | |
| Poročilo o opravljenem potovanju predprevzem CNC stružnice | | | |

Risanje Samooblike Privzeti načrt slovenščina

Diapozitiv 1 od 1

Slika 14: Pregled potnega naloga v programu Microsoft PowerPoint (zaslonska slika z osebnega računalnika)

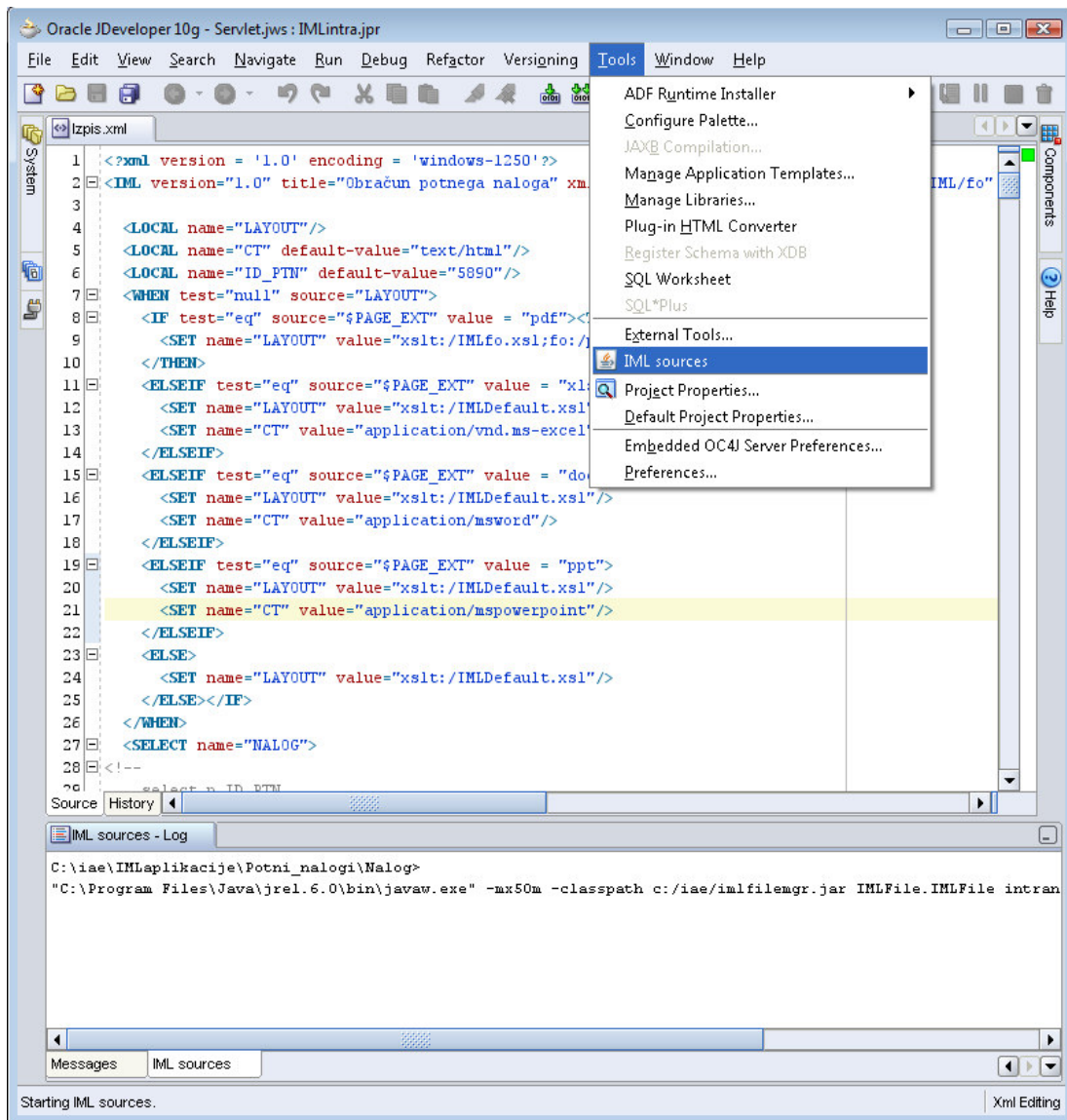
5.3 Povezava v obstoječi sistem s programom Oracle JDeveloper 10g

Po pripravi grafičnih elementov potnih nalogov smo začeli s povezovanjem nadgrajenih datotek v obstoječi sistem. Ustvarili smo več datotek XML, ki so nam služile pri pregledovanju našega dosedanjega dela. Preuredili smo datoteko *izpis.xml* in napisali datoteko *imlfo.xsl*, s katerima smo dobili potrebne podatke za izvedbo datoteke *zbirni2.rtf*. Z njo smo izpopolnili obliko potnih nalogov. Naštete datoteke imajo skupno velikost 168 KB. Sledilo je testiranje in odpravljanje napak pri prikazovanju potnih nalogov. V nadaljevanju je opisano povezovanje programske nadgradnje v obstoječi strežnik.

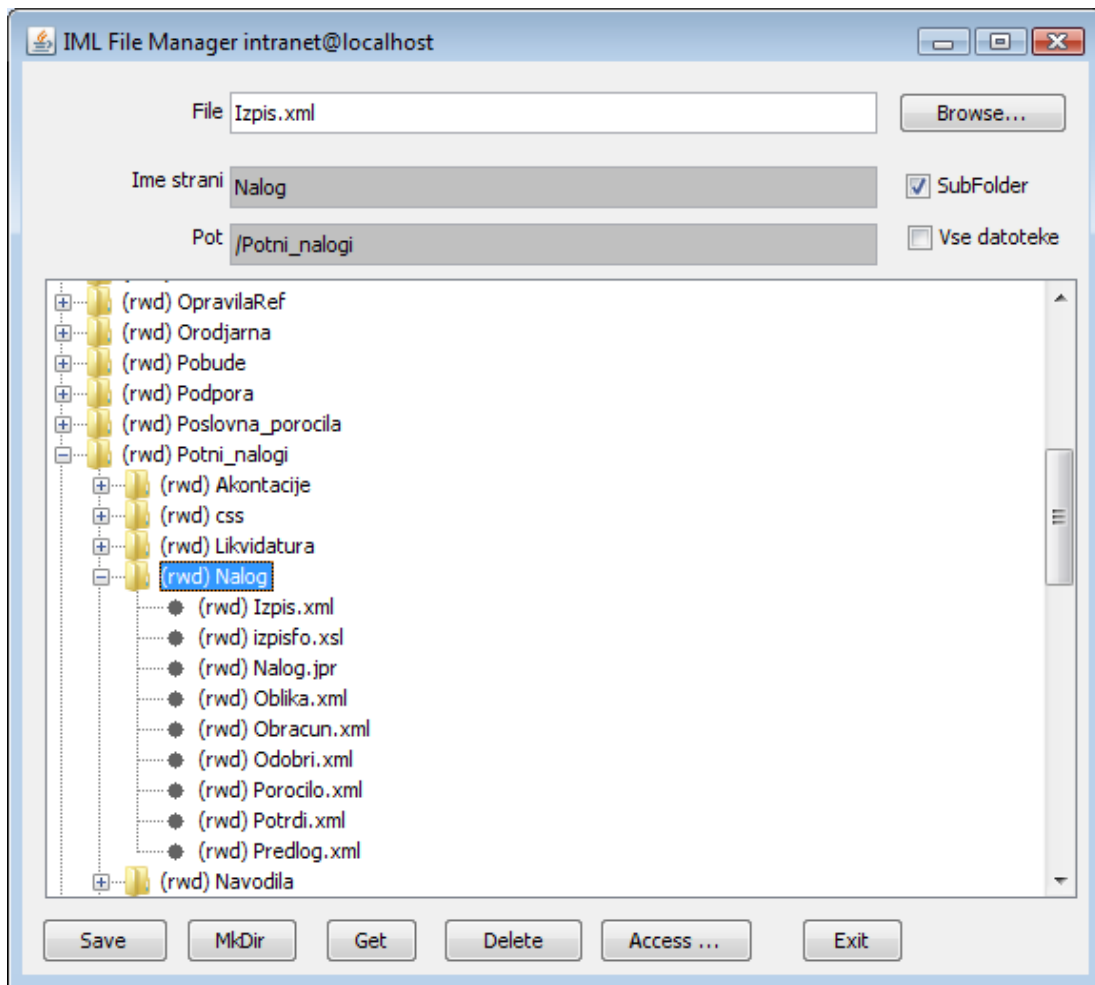
Ta faza dela se ukvarja z javnim strežnikom (angl. java-servlet) in javnimi knjižnicami (angl. java API). Nadgrajene in spremenjene datoteke smo naložili v javni strežnik s pomočjo programa Oracle JDeveloper 10g. Na sliki 15 je prikazan potek nalaganja datotek. V ukazni vrstici je okno z imenom Orodja (angl. Tools), kjer imamo ukaz označen z modro barvo z imenom *IML source*. Pri izbiri tega ukaza se nam odpre okno z imenom *IML File Manager intranet@localhost*. Tam izberemo mapo za shranjevanje datotek. To je prikazano na sliki 16.

Pri strežniških datotekah *IMLParserContext.java*, *IMLServletContext.java*, *TagIML.java* smo dodali naslove URL, s katerimi aplikacija zazna obliko prikaza, ko uporabnik na primer izbere pregled v obliki HTML, formatu PDF ali drugih oblikah. Ob zagonu datoteke *iaservlet.war* ni bilo nobenih napak. Ko smo zatem naložili datoteko *servlet.war*, smo ugotovili, da deluje pravilno. Sledilo je nalaganje potrebnih knjižnic.

Potni nalogi so na intranetu močno zaščiteni. Do njih nima dostopa vsakdo. Vsak uporabnik ima svoje uporabniško ime in geslo. Lahko si ogleda le svoje potne naloge, do drugih nalogov pa nima dostopa. Na primer, ko računovodstvo zahteva poročilo o stroških, ki so nastali pri potnih nalogih, dobi le finančne podatke, ne dobi pa ostalih podatkov, npr. kdo je potoval, kam je potoval itd. Poln dostop do potnih nalogov imajo le zaposleni z ustreznimi pooblastili.



Slika 15: Vnašanje nadgrajenih datotek v javin strežnik (zaslonska slika z osebnega računalnika)



Slika 16: Shranjevanje datotek z datotečnim poslovdnikom (zaslonska slika z osebnega računalnika)

Po končanem nalaganju datotek s programom Oracle JDeveloper 10g smo naložili datoteko *servlet.war*. Datoteka *servlet.war* je pomembna pri potnih nalogih, saj veže vse nadgrajene in druge datoteke, ki služijo za pripravo potnih nalogov. Pri nalaganju datoteke *servlet.war* smo uporabili strežnik Apache Tomcat, ki je namenjen upravljanju z internetnimi aplikacijami.

6 OPRAVLJENO DELO IN REZULTATI

Delo, ki smo ga opravili v okviru diplomske naloge, je ključnega pomena za prikazovanje in obdelavo potnih nalogov v podjetju Iskra Avtoelektrika. Naredili smo nov pregled potnih nalogov in sicer v obliki PDF, dodali pa smo še preglede v programih Microsoft Word, Microsoft Excel in Microsoft PowerPoint.

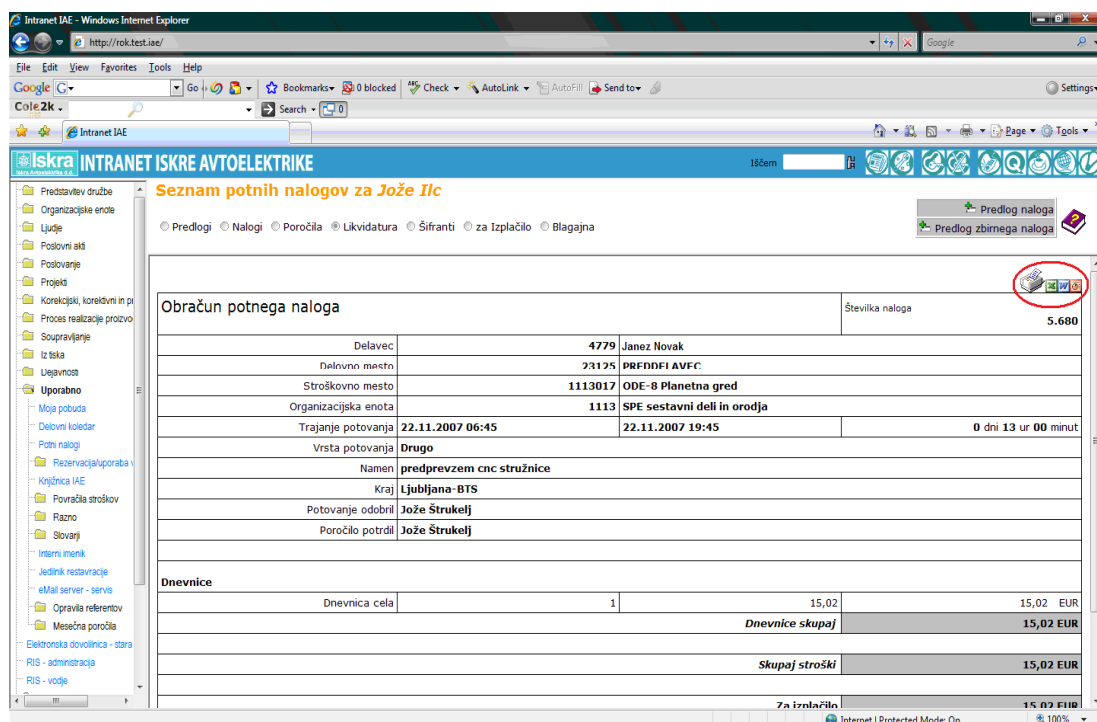
Uporabljene jezike in razvojno programsko opremo smo spoznavali mesec in pol. V tem času smo na delovni prenosni računalnik naložili celotno bazo podatkov z intraneta. Baza podatkov nam je služila pri preverjanju nastajajoče programske nadgradnje za pripravo potnih nalogov. Nato smo na prenosni računalnik naložili še vse potrebne programe za delo: Oracle JDeveloper, Oracle BI Publisher in strežnik Apache Tomcat.

Za nadgradnjo smo porabili tri mesece in pol. Najprej je bilo potrebno pridobiti podatke iz obrazca potnih nalogov. Napisali smo novo datoteko XSL, ki je izpisala potrebne podatke iz datoteke XML. Ti podatki so služili pri nadaljnjem delu s programom za grafično oblikovanje Oracle BI Publisher in urejevalnikom besedil Microsoft Word. Ko smo imeli pripravljeno grafično datoteko potnega naloga, smo lahko že prikazali potni nalog v obliki PDF. Delo smo nadaljevali s povezavo pripravljene nadgradnje v obstoječi sistem s programom Oracle JDeveloper 10g.

Na sliki 17 je prikazan končni rezultat, prikaz potnega naloga v nadgrajeni obliki. Na njej v desnem zgornjem kotu vidimo štiri rdeče obkrožene ikone. Prva ikona predstavlja tiskalnik. Če kliknemo nanjo, se prikaže potni nalog v obliki PDF. Druga ikona nam odpre potni nalog v programu za delo s preglednicami Microsoft Excel, tretja v urejevalniku besedil Microsoft Word, četrta pa v programu za računalniške predstavitve Microsoft PowerPoint.

Prvotna programska oprema za potne naloge v podjetju Iskra Avtoelektrika deluje od 1. marca 2003. Naša programska nadgradnja pa je v uporabi od 1. aprila 2008. Uporabljajo jo vsi, ki v podjetju potrebujejo in obdelujejo potne naloge. S tem smo dosegli lažjo in preglednejšo pripravo in tiskanje dokumentov, ki so v podjetju zelo pomembni. Z nadgradnjo priprave potnih nalogov smo pripomogli tudi k

enostavnejšemu arhiviranju dokumentov, saj se je z enotnim tiskanjem tudi arhiviranje poenostavilo.



Slika 17: Prikaz potnega naloga v nadgrajeni obliki (zaslonska slika z osebnega računalnika)

Vpeljevanje uporabnikov v delo z nadgrajeno programsko opremo ni bilo potrebno, saj se je oblikovno le malo spremenila, njena uporabnost pa povečala. Mesečno se z njo pripravi in obdela od 50 do 100 potnih nalogov.

Praktični del diplomske naloge je bil opravljen v sektorju Organizacija in informatika podjetja Iskra Avtoelektrika. Uporabljali smo razvojni program Oracle JDeveloper 10g in program za grafično oblikovanje Oracle BI Publisher. Delali smo z Iskrinim intranetom. Obseg dela z intranetom v podjetju je obsežen, ker je potrebno programe in podatke na njem dopolnjevati, spreminjati, nadgrajevati. Eden od programov na intranetu omogoča delo s potnimi nalogi. Potni nalogi so z nadgradnjo dobili nov videz in uporabnost. Sčasoma pa bo njihovo pripravo zaradi sprememb v poslovnem okolju verjetno potrebno ponovno nadgraditi.

7 ZAKLJUČEK

Elektronska izmenjava in obdelava podatkov je v današnjem času pomembna za vsako podjetje, saj lahko bistveno pripomore k učinkovitosti poslovanja. Podjetje Iskra Avtoelektrika se tega zaveda in ima že veliko poslovnih funkcij računalniško podprtih. K preglednejšemu in učinkovitejšemu izvajanju ene od njih, t.j. priprave potnih nalogov, pa je pripomoglo tudi to diplomsko delo.

Za nastajanje tega dela je bilo zelo koristno konstruktivno vzdušje med zaposlenimi v sektorju Organizacija in informatika. Delo v informatiki se spreminja in od razvijalca zahteva stalno izobraževanje in izpopolnjevanje. Tehnologija hitro napreduje, zato je tudi delo dinamično. Zaposleni na tem delovnem mestu mora slediti razvoju in novostim, ki jih nudi tržišče. Dobro mora poznati internet in ga stalno uporabljati pri svojem delu, saj na njem dobi veliko koristnih informacij.

V podjetju so dobro računalniško opremljeni in tudi njihovo omrežje je zelo dobro razvito. Intranet Iskre Avtoelektrika je za delavce zelo uporabna informacijska tehnologija. Delavcem omogoča hiter dostop do internih informacij, hitro komunikacijo in vpogled v dokumente. S tem prihranijo veliko dragocenega časa, ki bi ga v nasprotnem primeru porabili za obveščanje in pridobivanje informacij.

Diplomska naloga je uspešno zaključena. Z opravljenim delom so zadovoljni vsi uporabniki potnih nalogov. Potne naloge sprejemajo in tiskajo vsi tiskalniki na enak način. Izboljšana je preglednost in učinkovitost priprave potnih nalogov. Tudi njihova obdelava in arhiviranje sta postala enostavnejša. Koristen in pri nadaljnjem delu uporaben rezultat diplomskega dela pa je tudi pridobljeno znanje o jezikih za opis dokumentov za elektronsko izmenjavo in programskih orodjih za pripravo dokumentov.

Nadgrajeno programsko opremo bo odslej potrebno vzdrževati in jo po potrebi dopolnjevati v skladu z zahtevami uporabnikov in morebitnimi spremembami predpisov, ki zadevajo pripravo in obračun potnih nalogov. Za nadaljnje delo pa bi lahko izbrali še marsikatero komponento programske opreme v podjetju in jo nadgradili podobno, kot smo programsko opremo za potne naloge. Tak primer je program *Moja pobuda*, s katerim zaposleni vnašajo nove ideje, kritike, mnenja in

pohvale v zvezi z delom v Iskri Avtoelektriki. Uporablja ga večina zaposlenih, ki dnevno vnesejo do pet pobud. Teh se sčasoma nabere veliko, tako da bi bilo potrebno izboljšati preglednost njihovih prikazov, vnašalcem pobud pa omogočiti tudi urejanje vnosov. Od uporabnikov pa lahko pričakujemo tudi predloge za dopolnitve drugih programov in razvoj novih.

8 LITERATURA

Curk, L., Budimir, G. (1999). XML – Nov jezik na svetovnem spletu. Maribor: Institut informacijskih znanosti.

Davis, L., Johnson, D., Sawyer, B., Shekhar, K. (2007). Oracle BI Publisher 10g Fundamentals. California: Oracle University.

Extensible Stylesheet Language (XSL) Version 1.1. Pridobljeno 1.6.2008 s svetovnega spleta: <http://www.w3c.org/TR/xsl>

Gradišar, M. (2003). Uvod v informatiko. Ljubljana: Ekonomska fakulteta.

Gradišar, M., Jaklič, J., Talib, D., Baloh, P. (2005). Osnove poslovne informatike. Ljubljana: Ekonomska fakulteta.

Iskra Avtoelektrika (2008). Vodnik po Iskri Avtoelektrika d.d. Šempeter pri Gorici: Skupina Iskra Avtoelektrika.

Mesojedec, U., Fabjan, B. (2004). Java 2: temelji programiranja. Ljubljana: Pasadena.

Microsoft XML Core Services. Pridobljeno 1.6.2008 s svetovnega spleta: <http://kb.cert.org/vuls/id/585137>

Mrhar, P. (2002). Java 2: prvi korak. Šempeter pri Gorici: Flamingo.

Oracle JDeveloper 10g Reviewers Guide (2004). California: Oracle Corporation World Headquarters.

XSL Transformations (XSLT) Version 1.0. Pridobljeno 1.6.2008 s svetovnega spleta: <http://www.w3c.org/TR/xslt>

XML Path Language (XPath) Version 1.0. Pridobljeno 1.6.2008 s svetovnega spleta: <http://www.w3c.org/TR/xpath>

W3Schools - Web Developer's Site. Pridobljeno 2.2.2008 s svetovnega spleta:
<http://www.w3schools.com/>