

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Damjan Kmet

IZDELAVA VMESNIKA UPRAVITELJA PREDLOG
ZA DOKUMENTNE SISTEME

DIPLOMSKO DELO NA
VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Ljubljana, 2010



Št. naloge: 00513/2010

Datum: 05.04.2010

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **DAMJAN KMET**

Naslov: **IZDELAVA VMESNIKA UPRAVITELJA PREDLOG ZA DOKUMENTNE
SISTEME**

**IMPLEMENTATION OF A TEMPLATE MANAGEMENT INTERFACE
FOR DOCUMENT SYSTEMS**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

Dokumentni sistemi omogočajo podjetjem zbiranje, shranjevanje in dostopanje do večjega števila dokumentov iz najrazličnejših virov. Za večjo uporabnost so dokumentni sistemi pogosto povezani s pisarniškimi paketi, ki omogočajo večjo avtomatizacijo sistema, lažje delo z njim in poenotenje dokumentov.

V okviru diplomske naloge razvijte modul za dokumentni sistem v okolju Lotus Domino Notes, ki bo omogočal povezovanje s paketi MS Office in OpenOffice.org in bo vseboval možnost sestavljanja in upravljanja s predlogami dokumentov za te pakete.

Mentor:


doc. dr. Janez Demšar



Dekan:


prof. dr. Franc Solina

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Damjan Kmet

IZDELAVA VMESNIKA UPRAVITELJA PREDLOG
ZA DOKUMENTNE SISTEME

DIPLOMSKO DELO NA
VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: doc. dr. Janez Demšar

Ljubljana, 2010

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Damjan Kmet,
z vpisno številko 63020234,

sem avtor diplomskega dela z naslovom:

Izdelava vmesnika upravitelja predlog za dokumentne sisteme

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Janeza Demšarja
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 12.5.2010

Podpis avtorja:

ZAHVALA

Za pomoč pri izdelavi diplomske naloge se zahvaljujem mentorju doc. dr. Janezu Demšarju. Zahvaljujem se moji družini, puncu Kristýni, lektorici Barbari in vsem prijateljem za spodbudne besede in pomoč pri krašjanju študentskih dni. Za strokovno pomoč pri izdelavi praktičnega dela diplomske naloge se zahvaljujem sodelavcem iz podjetja 3K IT.

Kazalo

POVZETEK.....	1
ABSTRACT.....	3
1 UVOD.....	5
2 DOKUMENTNI SISTEM.....	6
2.1. Lastnosti dokumentnega sistema.....	7
3 OKOLJE LOTUS DOMINO/NOTES.....	10
3.1. Zgodovina.....	10
3.2. Struktura odjemalec-strežnik.....	10
3.3. Podatkovna baza Lotus Notes.....	12
3.3.1. Razdelitev dokumentov v zbirki podatkov.....	13
3.3.2. Programski jeziki.....	16
3.3.3. Domino Objektni Model (DOM)	17
3.3.4. COM/OLE.....	19
3.3.5. Repliciranje podatkovnih baz.....	19
3.4. Varnost v okolju Lotus Domino/Notes.....	20
3.5. Primernost okolja za dokumentne sisteme.....	21
4 PISARNIŠKI PAKETI.....	22
4.1. Microsoft Office.....	22
4.1.1. Programiranje.....	22
4.2. OpenOffice.org.....	24
4.2.1. Programiranje.....	26
5 UPRAVITELJ PREDLOG PISARNIŠKIH PAKETOV.....	30
5.1. Predloga pisarniškega paketa.....	30
5.2. Upravitelj predlog za dokumentni sistem 3K Document Cycle.....	31
5.2.1. Grafični vmesnik nastavitvev.....	32
5.2.2. Implementacija v dokumentni sistem 3K Document Cycle.....	35
6 SKLEPNE UGOTOVITVE.....	37

KRATICE

MS	Microsoft
DMS	Document Management System; Dokumentni sistem
EAN	European Article Number; Europsko označevanje artiklov
OCR	Optical character recognition; Optična prepoznavna znakov
ZVDAGA	Zakon o varstvu dokumentarnega in arhivskega gradiva ter arhivih
ISO	International Organization for Standardization; Mednarodna organizacija za standardizacijo
ERP	Enterprise Resource Planning; Sistem za upravljanje z viri
HRM	Human Resource Management; Sistem za upravljanje s človeškimi viri
DSS	Decision Support System; Odločitveni sistem
XML	Extensible Markup Language; Razširljiv označevalni jezik
CSV	Comma-separated Values; Vrednosti ločene z vejicami
SQL	Structured Query Language; Strukturirani povpraševalni jezik
PDF	Portable Document Format; Prenosni format dokumentov
MB	Megabajt; Merska enota
LAN	Local area network; Lokalno krajevno omrežje
WAN	Wide area network; Širokorazsežno omrežje
UI	User Interface; Uporabniški vmesnik
VBA	Visual Basicu for Applications; Programski jezik
JDK	Java Development Kit; Razvojni komplet Java
DOM	Domino Object Model; Objektni model Domino
COM	Component Object Model; Komponentni objektni model
OLE	Object Linking and Embedding; Povezovanje objektov in vgradnja
ACL	Access Control List; Kontrolna lista dostopa
GNU	General Public License; Splošna javna licenca

POVZETEK

Pri razvoju dokumentnih sistemov se pogosto srečujemo z implementacijo pisarniških paketov v dokumentni sistem. Pri tem se srečujemo s podobnimi zahtevami glede implementacije predlog, zato lahko zahteve združimo in izdelamo vmesnik za upravljanje predlog v dokumentnem sistemu. Vmesnik nam skrajša potreben čas za izdelavo in vzdrževanje predlog pisarniških paketov v dokumentnem sistemu. V okviru diplomske naloge smo izdelali programski vmesnik za implementacijo pisarniških paketov v dokumentne sisteme, ki delujejo v tehnologiji Lotus Domino/Notes. V prvem delu diplomske naloge smo predstavili zgradbo, lastnosti in namen dokumentnih sistemov. Pri tem smo spoznali grobo razdelitev dokumentov v podjetjih, delovne procese ter zakonodajo s področja arhiviranja dokumentov v elektronski obliki. Drugi del diplomske naloge je namenjen predstavitvi tehnologije Lotus Domino/Notes, ki je primerna za razvoj dokumentnih sistemov. Pri predstavitvi tehnologije smo spoznali zgodovino razvoja, dokumentno bazo podatkov, oblikovne dokumente in programske jezike podprte v omenjeni tehnologiji. Sledi opis pisarniške programske opreme na trgu s poudarkoma na pisarniških paketih Microsoft Office in OpenOffice.org. Oba pisarniška paketa smo spoznali s strani uporabnika in razvijalca. Ob tem smo navedli prednosti in slabosti obeh omenjenih paketov. Ključni del diplomske naloge je izdelava in predstavitev vmesnika upravitelja predlog za dokumentni sistem 3K Document Cycle. Vmesnik omogoča nastavitve preslikav preko grafičnega vmesnika, kar nam omogoča vključitev predlog v dokumentni sistem brez pisanja programske kode. Upravitelj predlog skrajša čas in zmanjša finančna sredstva pri implementaciji pisarniškega paketa v dokumentni sistem.

Ključne besede: Dokumentni sistem, Lotus Domino/Notes, Microsoft Office, OpenOffice.org, upravitelj predlog.

ABSTRACT

In the development of document management systems we are often confronted with the implementation of office packages into the document management system. By doing this, we face similar concerns regarding the implementation of the template, therefore we can put the requirements together and create an interface to manage the template into the document management system. The interface can shorten the time required for the construction and maintenance of templates of the office packages in the document management system. Within this diploma thesis we made an programme interface for the implementation of office packages in the document management systems operating in the Lotus Domino/Notes technology. In the first part of this thesis we presented the structure, properties and purpose of the document management systems. In so doing, we got to know the general distribution of documents within companies, work processes and legislation of the preservation of documents in electronic form. The second part of this thesis is to present Lotus Domino/Notes technology, which is suitable for developing document systems. In the presentation of the technology we have met history of development, document database, design documents and programme languages supported in this technology. Next is description of the office programme equipment offered on the market with a focus on office packages Microsoft Office and OpenOffice.org. Both of the office packages we have met from the view of the user and developer. Furthermore, we have listed the advantages and disadvantages of both these packages. A key part of this thesis is composition and presentation of the template to the administrator interface document management system 3K Document Cycle. The interface allows the maps setting through the graphical interface, which allows us to include the template into the document management system without writing code. A template manager reduces time and financial means in the implementation of the office package into the document management system.

Key words: Document Management System, Lotus Domino/Notes, Microsoft Office, OpenOffice.org, template manager.

1 UVOD

V uspešnih podjetjih se zavedajo pomembnosti obvladovanja velikega števila informacij v zelo kratkem času. Zaradi težnje po čim večjem izkoristku zbranih informacij stalno izboljšujejo svoj informacijski sistem, prilagajajo ga svojim potrebam, povezujejo različne informacijske podsisteme ter vgrajujejo ostalo programsko opremo v informacijski sistem.

Pomembno vlogo v poslovanju podjetja predstavlja dokumentni sistem. Dokumentni sistem je računalniški sistem za shranjevanje dokumentov v elektronski obliki. Med te dokumente sodijo vhodna pošta, izhodna pošta in notranji dokumenti podjetja. Že ob hitrem pregledu izhodne pošte ugotovimo slogovne, oblikovne in tudi vsebinske podobnosti posameznih dokumentov. Omenjene značilnosti dajo dodatne informacije prejemniku pošte o urejenosti, enotnosti podjetja. Da bi podjetja zagotovila ugled v družbi, se nagibajo k poenotenju svojih dopisov. Poenotenju pogosto sledi avtomatizacija in to sta dve glavni dejstvi, zaradi katerih se v dokumentne sisteme vključuje pisarniške pakete.

Na trgu programske opreme je dosegljivih več pisarniških paketov, med njimi izstopata predvsem dva. Prvi je Microsoft Office, v nadaljevanju MS Office, in drugi je OpenOffice.org. Oba paketa pokrijeta večino potreb uporabnika in imata podobne komponente. Med njima so tudi razlike. MS Office je plačljiv, naprednejši, z lepšo obliko. Njegov tekmeč je brezplačen, za poznavalce lažje prilagodljiv, najpomembnejša prednost je njegova odprta koda. Oba omogočata snemanje, poganjanje, pisanje programov, tako imenovanih makrov. Do omenjenih ukazov in spremenljivk dokumentov je možen dostop iz drugih programov in to je poglavitno za povezavo pisarniškega paketa z dokumentnim sistemom.

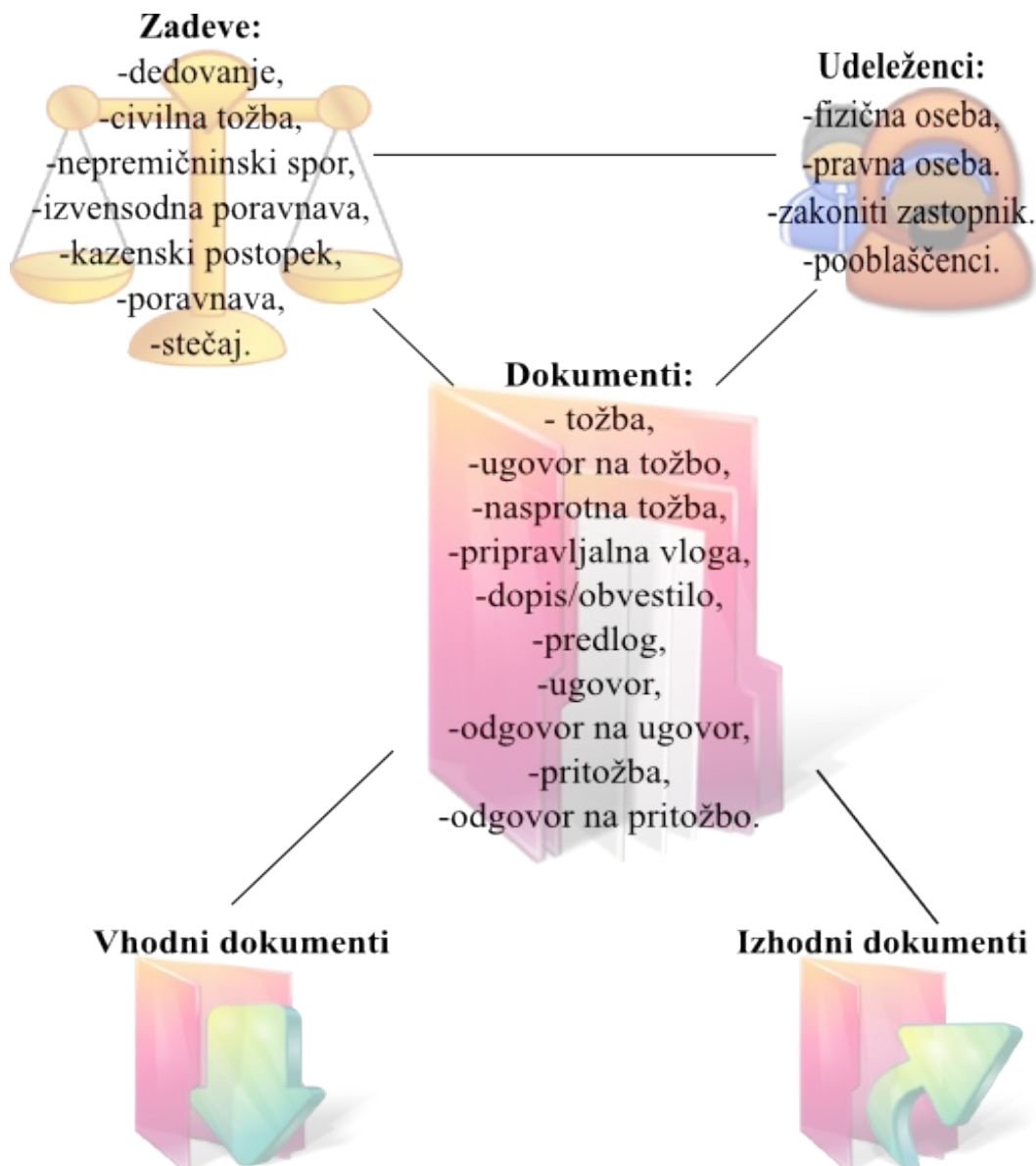
V okviru diplomske naloge smo razvili dokumentni sistem 3K Document Cycle, ki deluje v okolju Lotus Domino/Notes. V omenjenem sistemu že obstaja možnost izdelave Microsoft Word dokumentov s pomočjo MS Word predlog. Zastarel modul za izdelavo predlog se imenuje »Office Integrator«. Namen naše diplomske naloge je prenoviti obstoječi modul za izdelavo dokumentov MS Word, mu dodati novo funkcionalnost za izdelavo dokumentov MS Excel, OpenOffice.org Writer in OpenOffice.org Calc. MS Word in OpenOffice.org Writer sta primerna za pisanje izhodnih dopisov, ISO dokumentacije in pravilnikov v podjetjih. MS Excel in OpenOffice.org Calc sta s stališča dokumentnega sistema pomembna za hranjenje specifičnih podatkov, izvoz podatkov iz sistema za naknadno obdelavo, posredovanje drugim osebam ali uvoze v druge sisteme. V tem delu bomo predstavili dokumentni sistem, oba pisarniška paketa in okolje Lotus Domino/Notes. Vse troje bomo združili v celovit dokumentni sistem primeren za sodobna podjetja, da se le-ta lahko bolje posvetijo svojim poglavitnim dejavnostim.

2 DOKUMENTNI SISTEM

Dokumentni sistem (DMS) je računalniški sistem (ali nabor programske opreme), ki se uporablja za urejanje in shranjevanje elektronskih dokumentov in/ali digitalnih preslikav papirnatih dokumentov.[1]

Že samo ime dokumentni sistem nam pove, da govorimo o sistemu za zajem, izdelavo, hranjenje in preprosto iskanje dokumentov. V povezavi z besedo „računalniški“ dobimo elektronski sistem za celovito skrb dokumentov v elektronski obliki.

Dokumentni sistemi se pojavljajo v velikih, srednjih in tudi v manjših podjetjih, v katerih je potreba po obvladovanju velikega števila dokumentov. Naj kot primer manjšega podjetja navedemo odvetniško pisarno, ki zaradi narave svojega dela obvladuje veliko število dokumentov (glej sliko 1).



Slika 1: Povezava med dokumenti, zadevami in udeleženci v pravni pisarni.

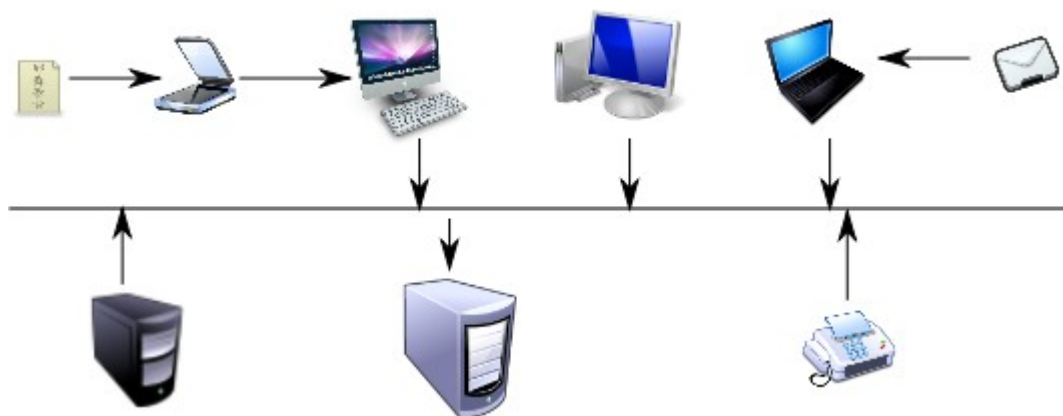
2.1. Lastnosti dokumentnega sistema

Napreden dokumentni sistem mora zagotavljati naslednje lastnosti [2]:

- vnos dokumentov,
- pregled in iskanje dokumentov,
- elektronski arhiv,
- delovni proces,
- povezovanje dokumentov,
- varnost,
- integracijo z drugimi sistemi,
- povezovanje z uporabniško programsko opremo.

Dokumentni sistem brez vsebine nima nobene uporabne vrednosti, zato se uporaba začne pri vnosu. Pri vnosu dokumenta uporabnik določi vrsto dokumenta, doda potrebne podatke, datoteke, skenirane dokumente, elektronsko pošto (e-mail), določi prejemnike in jih tudi ustrezno obvesti (glej sliko 2). Na sliki je prikazan tudi avtomatski vnos dokumentov iz elektronskega faksa in masovni prenos iz drugih strežnikov.

V podjetjih obstajajo delovna mesta s skrbjo za vnos vhodne pošte v dokumentni sistem tako imenovane skenirne postaje oziroma vložišča. Pri enem izmed slovenskih ponudnikov dostopa do spleta smo uvedli dokumentni sistem, ki je zanimiv zaradi vnosa podpisanih naročniških pogodb v sistem. Operater v podjetju izdela pogodbo, jo natisne in pošlje potencialnemu naročniku. Naročnik podpisano pogodbo vrne podjetju, ki sedaj pristane v vložišču. Omenjena pogodba v papirnati obliki vsebuje v desnem gornjem kotu črtno kodo (EAN) [3], preko katere dokumentni sistem prepozna številko pogodbe in jo samodejno pripne, v elektronski obliki, ustrezni že obstoječi pogodbi v sistemu.



Slika 2: Vnos dokumentov v dokumentni sistem.

Imeti ustrezen dokument v pravem trenutku na pravem mestu je eden izmed ključnih vzrokov vpeljave dokumentnega sistema v podjetje. Dokumentni sistem mora zagotoviti pregledno ureditev dokumentov po vrsti, prejemniku, naslovniku, pošiljatelju, statusu življenjske dobe dokumenta, datumih. Vse dokumente, preko pogledov, je težko najti, zato je občasno hitrejše iskanje po ključnih podatkih. Ključen podatek je vsebovana beseda v elektronskem dokumentu, potrebna za razpoznavo in opis dokumenta. Napredni dokumentni sistemi

omogočajo optično prepoznavo znakov (OCR) [4], pri katerih pride do pretvorbe slike v elektronski obliki v besedilo v elektronski obliki, primerno za urejanje ali kot ključni podatek pri iskanju po dokumentih.

Množica dokumentov zahteva odziven, prostoren, trajen in stabilen arhiv. O pomembnosti pravilnega arhiviranja govori dejstvo obstoja Zakona o varstvu dokumentarnega in arhivskega gradiva ter arhivih (ZVDAGA) [5].

Zakon predpisuje za elektronske arhive:

- ustrezno fizično varovanje,
- ustrezne klimatizirane prostore,
- politiko dostopa do arhiva,
- ustrezne pogoje in postopke uničevanja izvornega dokumenta,
- zajem dokumentov,
- sledljivost dostopa,
- reprodukcijo dokumentov.

Ob dokazanem pravilnem izvajanju arhiviranja postane elektronski arhiv pravnomočen dokument, izenačen izvornemu dokumentu.

Arhiv Republike Slovenije, glede na izpolnjene zakonske kriterije, podeli akreditacijo določenemu arhivu, s katerim pridobi le ta zaupanje potencialnih uporabnikov.

V podjetjih obstajajo dokumenti, kateri nastanejo po vnaprej določenem postopku oz. delovnem procesu (workflow). Delovni proces se uporablja za dopolnjevanje, potrjevanje in za obvezen pregled dokumentov. Najpogosteje ga srečamo pri izdelavi in potrjevanju vhodnih računov, potnih nalogov, zahtevkov za naročila, ISO dokumentaciji, ISO presojah, ukrepah, obvestilih, pogodbah.

Glavni pojmi:

- **delovni proces** je sestavljen potek dela, v katerem ljudje delujemo kot udeleženci v določenem stanju - statusu;
- **status** je stanje dokumenta v nekem trenutku;
- **začetni status** je prvi status delovnega procesa dokumenta;
- **končni status** je zadnji status v delovnem procesu. Ta status nima nobene možne akcije, niti nima aktivnega udeleženca;
- **procesna akcija** je aktivnost, pri kateri se zamenja status dokumenta;
- **aktivni udeleženec** izvaja akcije za spremembo statusov.

Nekateri dokumenti so med sabo v odvisni zvezi npr. aneks k pogodbi ne more obstajati brez prej podpisane pogodbe, spet drugi so med seboj tesno povezani npr. vhodni račun je posledica zahtevka za naročilo. Med seboj odvisne, povezane dokumente povezujemo zaradi preglednosti in hitrejšega iskanja.

V dokumentnem sistemu, kakor tudi v računalništvu na splošno, ne smemo pozabiti na varnost skozi celoten sistem. Naj gre za dostop do posameznih modulov sistema, do določenega dokumenta, urejanje določenega dokumenta ali dodajanje novega dokumenta, vedno moramo vedeti, kdo lahko kaj počne. Pri tem ne smemo pozabiti na varen prenos sporočil po medmrežju in uspešno avtorizacijo uporabnikov.

Današnji informacijski sistem je skupek sistemov delujočih na različnih tehnologijah. Med seboj se tesno prepletajo zaradi poenotenih šifrantov, enotnih podatkov, ne ponavljajočega vnosa, povezovanja s sistemi poslovnih partnerjev, lažje administracije in preprostejše uporabe. Dokumentni sistemi se tako najpogosteje povezujejo s sistemi za upravljanje s sredstvi (ERP) [6], sistemi za upravljanje s človeškimi viri (HRM) [7] in odločitvenimi sistemi (DSS) [8].

Med sabo niso povezani samo sistemi z različnim poglavitnim namenom, temveč delujejo na različnih tehnologijah. Tako se pri povezovanju srečujemo z uporabo:

- XML - razširljivega označevalnega jezika (Extensible Markup Language),
- CSV - vrednosti ločenih z vejicami (Comma-separated Values),
- s poizvedbami SQL v druge sisteme,
- spletnimi storitvami (Web service),
- izvozi in uvozi MS Excel in OpenOffice.org Calc preglednic.

Tehnologija, v kateri je razvit dokumentni sistem, velikokrat ne dopušča izdelavo oziroma obdelavo datotek, oziroma bi bila realizacija potrebnega v omenjenih tehnologiji cenovno in časovno preveč potratna, zato se v dokumentne sisteme vpeljuje tudi uporabniška programska oprema oziroma namizne aplikacije. Najpogosteje se v dokumentne sisteme implementirajo pisarniški paketi, programi za izdelavo PDF dokumentov (Portable Document Format) in obdelavo slik.

Za primer implementacije uporabniškega programa bomo podali program za zmanjšanje bitnih slik v modul za pomoč uporabnikom v podjetju. Pri vnosu novih zahtevkov uporabniki velikokrat priložijo sliko računalniškega zaslona z napako. Pri računalniškem zaslonu, z resolucijo 1680x1050, slika preseže velikost 5MB podatkov, kar pri zelo velikem številu zahtevkov upočasni delovanje elektronskega arhiva. Z zmanjševanjem slik na zaključenih zahtevkih za podporo se je velikost arhiva dokumentov računalniške podpore zmanjšala za več kot 5-krat.

Dokumente zaradi večje preglednosti in lažjega upravljanja dokumentov na grobo delimo v tri skupine:

- **vhodna pošta**, nastane v izvorni obliki izven podjetja, v katerem deluje dokumentni sistem;
- **izhodna pošta**, potuje iz podjetja proti zunanji fizični oziroma pravni osebi;
- **dokumenti podjetja**, so namenjeni uporabi osebam znotraj podjetja.

3 OKOLJE LOTUS DOMINO/NOTES

Vsak dokumentni sistem je razvit v okolju in s pomočjo tehnologije, katero podpira okolje. Eden izmed primernih okoljih za razvoj dokumentnega sistema je okolje Lotus Domino/Notes, poznano s svojo dolgo zgodovino v informacijski tehnologiji.

3.1. Zgodovina

Začetki razvoja okolja Lotus Domino/Notes segajo v leto 1973, ko je David Wolley razvil sistem računalniške uporabniške podpore imenovan Notes. Takratni program Notes je bila prva več uporabniška aplikacija, ki je omogočala delo v povezanem načinu in pogovor preko odgovora na odgovor in je delovala na računalniškem sistemu PLATO. V letu 1974 je bil programu dodan modul Osebna beležka (Personal Notes), kar je spremenilo program tudi v odjemalec elektronske pošte, ki je omogočal nitno odgovarjanje pošte .

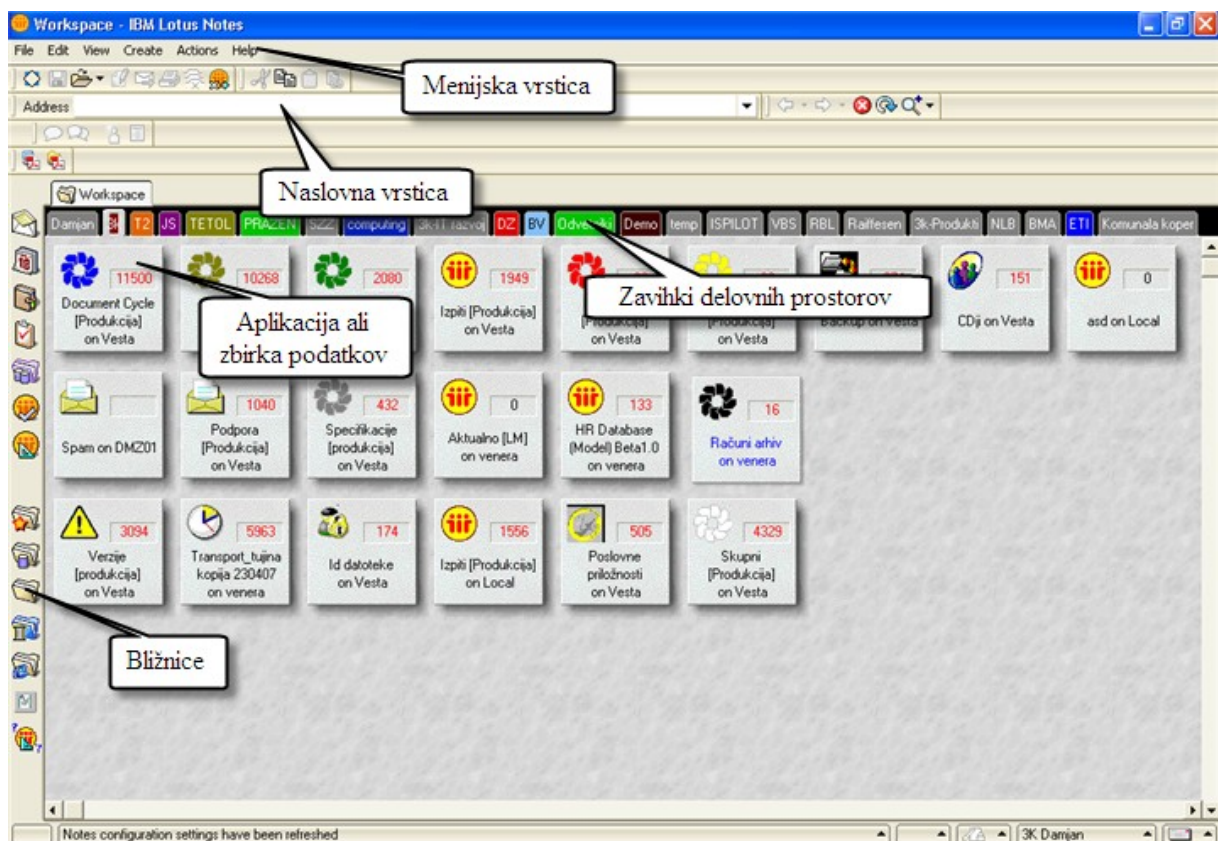
Ime Lotus Notes se pojavi v letu 1989, ko je začel računalniški program podpirati koncept objektnega modela zapisa baze, omogočal kreiranje in upravljanje nedoločene podatkovne sheme, obogateno besedilo, gradnjo robustne podatkovne baze v uporabniku prijaznem programskem vmesniku, podprtim konceptom prenosne baze in oddaljenega dostopa do baze podatkov. Skozi različne izdaje je dobival dodatne razširitve in lastnosti, ki jih poznamo danes. [9]

Prej omenjene lastnosti so program Lotus Notes uvrstile v enega izmed najuspešnejših poslovnih programov v zgodovini in število prodanih licenc je doseglo 145 milijonov. [10]

3.2. Struktura odjemalec-strežnik

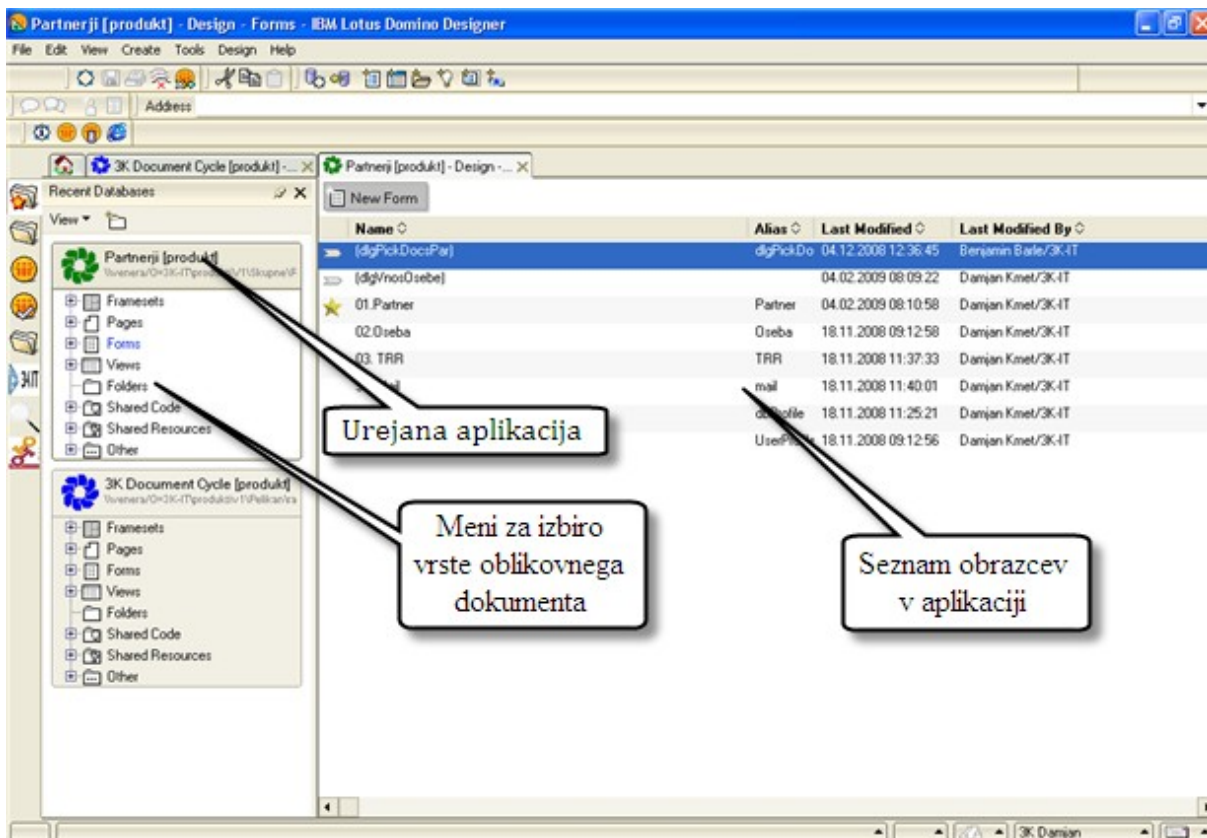
V podjetjih zaradi več uporabniškega dostopa do podatkov težijo k centralizaciji podatkovnih baz na centralnih strežnikih. Arhitektura Lotus Domino/Notes deluje po principu strežnika in odjemalca in tako zagotavlja dostop več uporabnikov do istega podatka v želenem trenutku. Vlogo strežnika ima Lotus Domino, vlogo glavnega odjemalca ima Lotus Notes. Kriptirana sporočila med strežnikom Lotus Domino in uporabljenim odjemalcem se izmenjujejo preko lokalnega krajevnega omrežja (LAN), širokorazsežnega omrežja (WAN) ali svetovnega spleta.

Do aplikacij Lotus Notes je možen dostop preko več različnih uporabniških vmesnikov glede na zelene potrebe. Najbolj razširjen odjemalec do zbirk podatkov Lotus Notes je odjemalec **Lotus Notes** (slika 3). Uporabnik ima v omenjenem odjemalcu izbrane aplikacije, ki jih potrebuje. Aplikacije so razvrščene po delovnih prostorih (Workspace). Poleg uporabniških aplikacij lahko dostopa do svojega poštnega predala in svetovnega spleta.



Slika 3: Odjemalec Lotus Notes

Razvijalci uporabljajo za razvoj aplikacij Lotus Notes odjemalec **Domino Designer** (slika 4). Oblikovno je vmesnik podoben odjemalecu Lotus Notes, le da pri njemu ne dodajamo, iščemo podatkovnih dokumentov, temveč dodajamo in urejamo oblikovne dokumente.



Slika 4: Domino Designer

Za upravljanje strežnika Lotus Domino uporabljamo odjemalec **Domino Administrator**. Vmesnik uporabljamo za upravljanje uporabnikov, uporabniških skupin, certifikatov, baz Lotus Notes, repliciranja podatkovnih baz, analiziranja delovanja strežnika, upravljanje poštnega strežnika, podpisovanja aplikacij in drugih operacij za celovito upravljanje infrastrukture Lotus Domino/Notes. Za dostop do aplikacij preko pametnih telefonov se uporablja odjemalec **Lotus Notes Traveler**. Poleg vsega naštetega lahko uporabniki do aplikacij dostopajo preko spletnega brskalnika.

Lotus Domino/Notes deluje neodvisno na različnih operacijskih sistemih. Strežnik Lotus Domino deluje na operacijskih sistemih: Microsoft Windows, Linux, Sun Solaris in IBM AIX. Odjemalci Lotus Notes, Lotus Domino Administrator, Lotus Domino Designer delujejo na operacijskih sistemih: Microsoft Windows, Linux in Mac OS. Lotus Notes Traveler je podprt z velikim naborom pametnih telefonov [11]. Operacijski sistemi, v omrežju strežnikov in odjemalcev, so lahko mešano razporejeni. Tako lahko en strežnik Lotus Domino deluje na operacijskem sistemu Sun Solaris, drugi na Microsoft Windows, do njiju lahko dostopamo preko vseh prej omenjenih operacijskih sistemov.

3.3. Podatkovna baza Lotus Notes

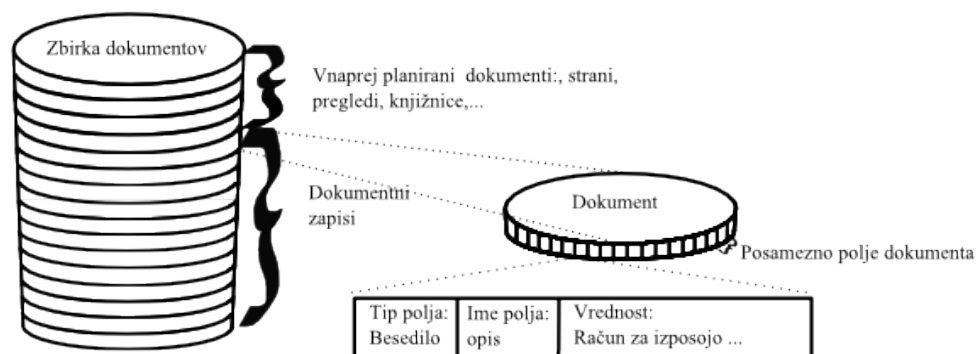
Vsaka aplikacija v tehnologiji Lotus Notes vsebuje eno ali več podatkovnih baz oziroma zbirk zapisov. Zbirka zapisov Lotus Notes ima končnico .nsf (Notes Storage Facility) na datotečnem sistemu. Če je glavna funkcionalnost relacijske podatkovne baze v hrambi velike količine podatkov vnaprej določene oblike potem podatkovna zbirka Lotus Notes vsebuje še

uporabniške vmesnike, ostale oblikovne elemente in programsko kodo. Omenjene lastnosti naredijo razvoj podatkovnih baz Lotus Notes edinstven in težko razumljiv celo razvijalcem aplikacij, ki ne poznajo razvoja v Lotus Notes.

3.3.1. Razdelitev dokumentov v zbirki podatkov

Zaradi edinstvene, prej omenjene zbirke podatkov delimo zapise oziroma dokumente Lotus Notes v dve glavni skupini (slika 5) [12] :

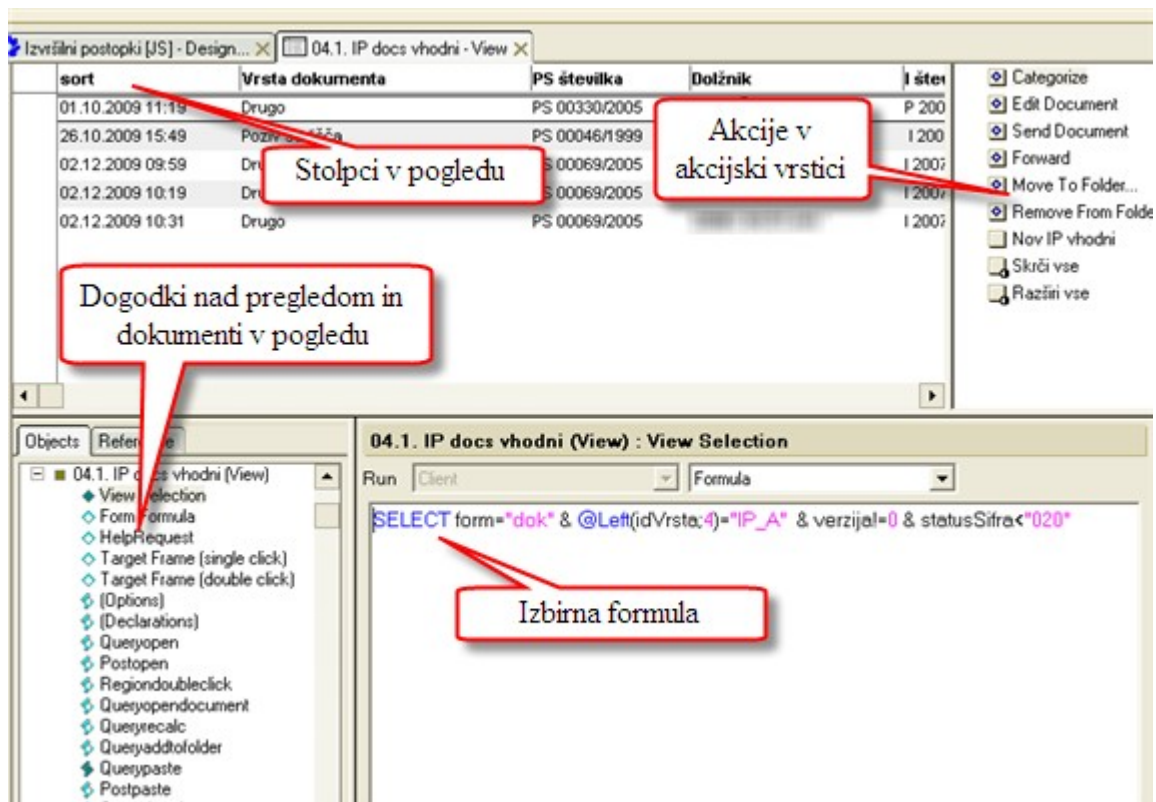
- zbirka oblikovnih dokumentov,
- zbirka podatkovni dokumentov.



Slika 5: Lotus Domino podatkovna baza. [12]

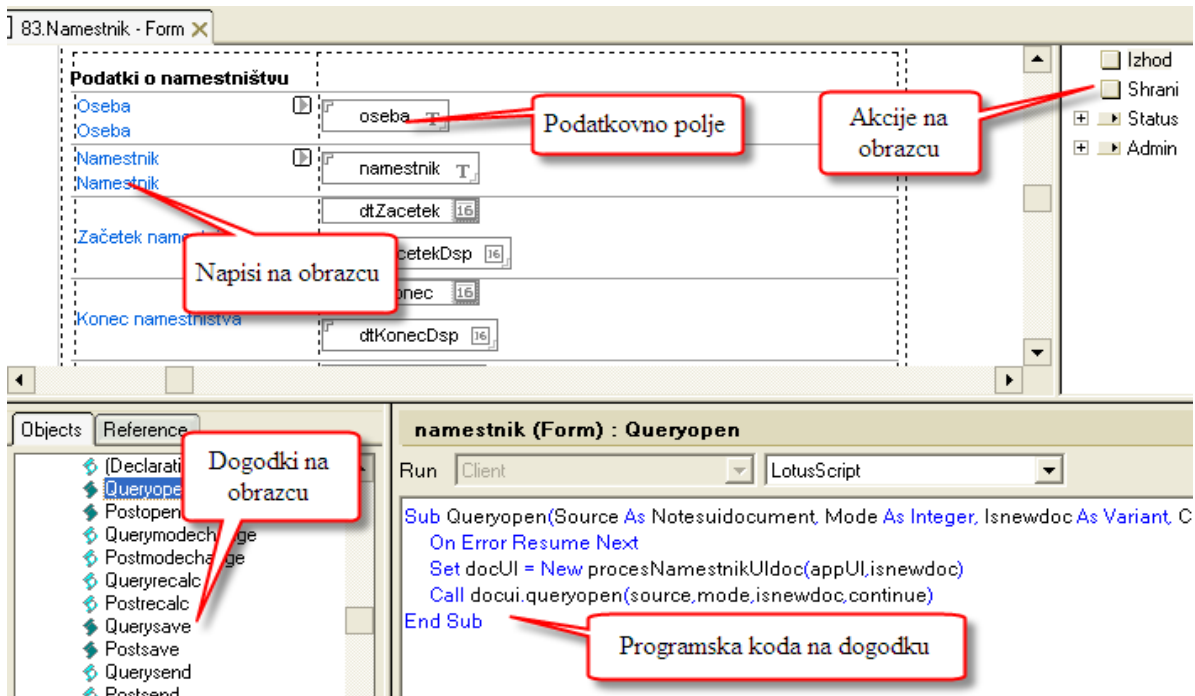
Oblikovni dokumenti v zbirki podatkov služijo prikazu, urejanju, dodajanju podatkovnih dokumentov in se ne spreminjajo oziroma njihov nabor ostane enak skozi uporabo aplikacije.

Najpomembnejši oblikovni element za prikazovanje zbirke dokumentov je pogled (View), prikazan na sliki 6. V njem so prikazani dokumenti, izbrani z izbirno formulo v obliki stolpcev in vrstic.



Slika 6: Oblikovni dokument pogled kot ga vidi razvijalec.

Za dodajanje, urejanje in pregled posameznega dokumenta se uporablja obrazec (Form) (slika 7).



Slika 7: Oblikovni dokument obrazec kot ga vidi razvijalec.

Navigacija po bazi poteka preko navigatorja (Outline). Preko navigacije lahko izbiramo poglede, poglede iz drugih baz, odpiramo obrazce, strani, poganjamo agente, odpiramo druge navigatorje. Navigacija poteka s pomočjo okna razdeljenega v okvirje (Frame).

Eno izmed najmočnejših orodij v tehnologiji Lotus Domino/Notes so agenti. Agenti so programi, makri za izvajanje operacij nad dokumentom, zbirko dokumentov, nad celotno bazo ali dokumentih v drugih bazah. Delujejo lahko na odjemalcu ali strežniku. Sprožijo se lahko ob vsakem novem dodanem dokumentu v bazo, novi elektronski pošti, uporabniškem proženju, ob določenem času in dnevu ali po vnaprej določeni časovni periodi [9].

Posameznemu dokumentu v podatkovni bazi Lotus Notes ustreza objektni zapis v podatkovni bazi. Vrednosti na dokumentu so shranjene v poljih, kateremu pripada tudi tip zapisa in ime polja. Podatkovna baza Lotus Notes, za razliko od relacijskih podatkovnih baz, ne vsebuje vnaprej točno določene oblike zapisov. Z nedoločeno obliko zapisov je razvoj aplikacij hitrejši, saj razvijalcu ni potrebno za novo vrsto podatka popravljati podatkovno shemo baze.

Za lažjo razumevanje dokumentne baze in princip dela z dokumenti si lahko predstavljamo zbirko prejetih papirjev v fasciklu, urejenih po abecedi pošiljatelja. Vsak papir je v barvno-presojni plastični ovojnici, katere barva predstavlja vrsto dokumenta npr. račun, zavarovalno polico... Poleg barve lahko označimo datum prejema z jezičkom na ovojnici. Tako označene dokumente lahko hitro in enostavno poiščemo po vrsti dokumenta, po datumu prejema ali pošiljatelju. Dokument lahko pogledamo preko presoje mape. Če želimo na dokument dopisati, ga vzamemo iz ovojnice, popravimo in vrnemo. Ko se nam v mapi nabere veliko število računov, lahko le-te ločimo v drugo mapo, namenjeno računom.

Pregled (slika 8) si lahko zamislimo kot zbirko vseh ali določenih (računi v ločeni mapi) dokumentov, urejenih po ključnih podatkih dokumentov. V primeru mape s papirnatimi dokumenti (lahko iščemo) imamo pogled po pošiljatelju, vrsti (barvi ovojnice) ali datumu prejema (jeziček ovojnice) dokumenta. Znotraj posameznega pogleda lahko eno ali več stopenjsko kategoriziramo podatke npr. prva kategorija je lahko po letu vnosa, druga po vnašalcu dokumenta. Podatke v stolpcih lahko preprosto sortiramo s klikom na stolpec.

Dokumente lahko z iskalnikom poiščemo v pregledu, glede na sortiran stolpec in začetne znake iskalnega niza. Drugi način iskanja je po vsebini prikazanih dokumentov, ne glede na to, ali je prikazan iskani podatek v pogledu.

Številka	Podjetje	Vrsta	Datum upisa	Opis
2004				
2005				
2006				
1009/2006				Vabilo na mednarodni simpozij o telekomunikacijah
1007/2006				Otrestni list
1007/2006		Bančni izpisek	11.10.2006	Specifikacija plačanih nadomestil
1006/2006		Dopis	11.10.2006	Ponudba predlog aktualnih pogodb
1005/2006		Pogodba	11.10.2006	3K03/2006: Pogodba o razvoju programske opreme
		Račun	11.10.2006	0185-2006: Dob. št.: 69/06
		Račun	11.10.2006	0184-2006: Dob. št.: 0454046618
		Dopis	11.10.2006	Ponudba za računovodske storitve
1001/2006		Dopis		1770758
1000/2006		Račun		770758
0999/2006		Račun		0394
0998/2006		Specifikacija		Specifikacija za razvoj programske opreme, kamera, nalog
0997/2006		Račun	10.10.2006	0122-2006: Storitve prosis za MOL
0996/2006		Specifikacija	10.10.2006	Capex
0995/2006		Specifikacija	10.10.2006	GPI
0994/2006		Pogodba	10.10.2006	PROSIS_0603: POGODBO O SODELOVANJU PRI PI
0993/2006		Ponudba	10.10.2006	0060-2006: Pelikan
0992/2006		Dopis	09.10.2006	vabilo na F-SECURE ROADSHOW
0991/2006		Tiskovina	09.10.2006	Glas gospodarstva -revija
0990/2006		Račun	09.10.2006	0181-2006: Dob. št.: 06-360-000219
0989/2006		Račun	06.10.2006	0121-2006: september 2006

Slika 8: Pregled dokumentov kot ga vidi uporabnik.

Posamezen dokument najpogosteje odpremo z dvojnimi klikom na izbrani dokument in se nam prikaže v prej omenjenem obrazcu. Za urejanje besedila je potrebno dokument odpreti v urejevalnem načinu. Po spreminjanju lahko dokument shranimo, zapremo in spremembe so končane. Podobno je s papirjem v prosojni mapi: ne moremo ga popraviti, lahko ga vzamemo iz mape in popravimo. Najpogostejši način dodajanja novega dokumenta je z odpiranjem praznega obrazca in shranitvijo le-tega. Ta način dodajanja dokumenta poteka preko uporabniškega vmesnika (UI). Drugi način dodajanja ali spreminjanja dokumenta se izvaja programsko v ozadju (Back-End).

3.3.2. Programski jeziki

Tekom razvoja Lotus Notes so se razvili in dodajali programski jeziki. Programske jezike v Lotus Notes obravnavamo pod poglavjem podatkovna baza, ker so del podatkovne baze dostopne preko odjemalca Lotus Notes Designer.

Jezik formul (Formula) je prvi, nepogrešljivi, najhitrejši programski jezik v Lotus Notes. Dejstvo je, da ne obstaja aplikacija Lotus Notes brez jezika formul. Najdemo ga na skoraj vseh oblikovnih dokumentih. Samo ta jezik je v uporabi za začetne vrednosti polj, preračunavanje prikaznih polj, za naslov obrazca, za izbirni stavek pogledov, za določitev vrednosti na pogledih in še za veliko drugega. Poleg tega ga lahko uporabimo namesto drugih novejših programskih jezikov. O njegovi hitrosti in uporabnosti nas prepriča dejstvo, da ga velikokrat vključimo v programsko kodo drugega jezika, da pospešimo izvajanje aplikacije ali zaradi prezahtevnega pisanja programske kode v drugem jeziku. Primer izbirne formule na pogledu se nahaja na spodnji sliki (slika 9).


```
SELECT form="dok" & @Contains(idTip;"\vhodni račun") & !@IsResponseDoc & verzija=0
```

Slika 9: Primer izbirne formule na pogledu.

Najpopularnejši jezik v sodobnih aplikacijah Lotus Notes je LotusScript. Semantično je jezik podoben "Visual Basicu for Applications" (VBA) podjetja Microsoft. Poleg tega imata oba skupno lastnost gostovanja v drugem okolju. Tako VBA gostuje v pisarniškem paketu Microsoft Office in posledično na operacijskih sistemih Windows in v omejeni različici na Mac OS. LotusScript gostuje v okolju Lotus Domino/Notes in tako deluje na sistemih Windows, Mac OS, Linux, OS/2, OS/400, AIX in Solaris. LotusScript je objektno usmerjeni programski jezik, zaradi tega lahko kreiramo lastne objekte, jih dedujemo in še veliko več.[9]

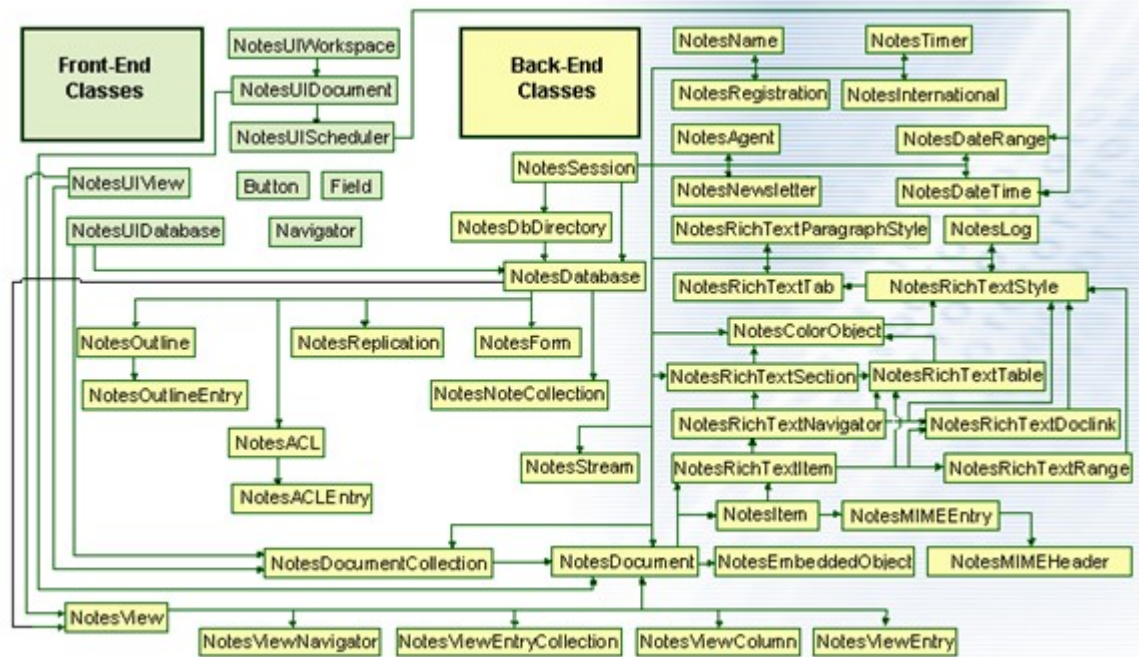
Pri razvoju aplikacij v Lotus Notes se srečamo tudi z nam dobro poznanim programskim jezikom Java. Kodo, ki je pisana v omenjenem programskem jeziku, lahko shranjujemo v knjižnice za programski jezik Java in v agente. Javanske knjižnice vsebujejo razrede JDK (Java Development Kit) in dodatne podrazrede "lotus.domino" s podporo za podatkovne baze Lotus Notes in dokumente v njej. [9] Neposredna uporaba javanskih funkcij na dogodkih ostalih oblikovnih dokumentov ni mogoča, lahko jo uporabimo le preko klicev iz ostalih podprtih programskih jezikov. Do Lotus Domino/Notes verzije 7 je bila Java edini programski jezik, preko katere je bilo mogoče izdelati spletne servise in prenašati datoteke z enega strežnika na drugega brez shranitve datoteke v podatkovno bazo Lotus Notes.

Moč Jave v Lotus Notes se je izkazala na primer pri dekodiranju zaslonske slike, shranjene v podatkovni bazi Lotus Notes, v obliki zapisa "base64". Zaradi počasnosti jezika LotusScript nad obdelavo velikih nizov znakov, je dekodiranje slike velikosti 1280x1024 potekalo približno 30 minut. Z vpeljavo funkcije, napisane v programskem jeziku Java, je bil isti učinek dosežen v približno dveh sekundah. Vzrok za dolg čas preračunavanja v LotusScriptu je v počasnosti skriptnih jezikov.

Za namen dostopa do aplikacij Lotus Notes preko spletnega brskalnika je bil dodan skriptni jezik JavaScript. JavaScript se uporablja na dogodkih nad oblikovnimi dokumenti in v agentih. Programska koda je lahko shranjena na uporabljenem oblikovnem dokumentu ali v skriptni knjižnici.

3.3.3. Domino Objektni Model (DOM)

Za razumevanje aplikacij Lotus Notes moramo razumeti objektni model Domino (DOM). V objektnem modelu Domino delimo objekte na tako imenovane razrede "Front-End" in "Back-End". Razredi "Front-End" so na sliki 10 obarvani z zeleno barvo in delujejo na uporabniškem vmesniku tako rekoč na vidnih pogledih, obrazcih, delovnem prostoru (NotesUIWorkspace). Objekti "Back-End" so na sliki obarvani z rumeno barvo in se uporabljajo za obdelavo posameznega dokumenta ali zbirke dokumentov v ozadju. Objekti "Front-End" vsebujejo objekte "Back-End", obratno pa ne velja.



Slika 10: Prikaz "Front-End" in "Back-End" razredov na zaslonski sliki odjemalca Domino Designer.

Za lažje razumevanje DOM-a bomo pogledali primer spreminjanja vrednosti polja na dokumentu s pomočjo izbire vrednosti polja z drugega dokumenta preko uporabniškega vmesnika - obrazca. Uporabnik sproži akcijo na obrazcu, pokaže se mu pogled z dokumenti, izbere dokument, klikne gumb "Potrdi" in tako spremeni vrednost na dokumentu. Programska koda iz omenjenega gumba se nahaja na sliki 11, kjer so komentarji programske kode pisani z zeleno barvo.

```

Sub Click(Source As Button)
'Deklariram Front-End razrede.
  Dim workspace As NotesUIWorkspace
  Dim uiDB As NotesUIDatabase
  Dim uiDok As NotesUIDocument

'Deklariram Back-End razrede.
  Dim db As NotesDatabase
  Dim dok As NotesDocument
  Dim zbiruDok As NotesDocumentCollection
  Dim izbraniDok As NotesDocument

'Nastavim Front-End razrede.
  Set workspace =New NotesUIWorkspace
  Set uiDok = workspace.CurrentDocument
  Set uiDB =workspace.CurrentDatabase

'Nastavim Back-End razrede.
  Set db=uidb.Database
  Set dok=uiDok.Document

'izvedem izbiro preko Front-End razreda workspace.
'Funkcija pickListCollection mi vrne Back-End zbirko dokumentov.
'Prvi dokument v zbirki je izbrani Back-End dokument.
'Nastavimo vrednost na trenutnem Back-End dokumentu.
'Posodobimo Front-End dokument in prikažemo vrednost iz izbranega dokumenta na obrazcu.
  Set zbiruDok=workspace.PickListCollection(3, False,db.Server,db.FilePath,"(pogledZalzbiri)","Izbira dokumenta","Izbira")
  If zbiruDok.Count>0 Then
    Set izbraniDok=zbiruDok.GetFirstDocument
    dok.poljeNaObrazcu=izbraniDok.izbranoPolje(0)
    Call uiDok.Reload
  End If
End Sub

```

Slika 11: Programska koda za izbiro dokumentov na "Front-End" razredu.

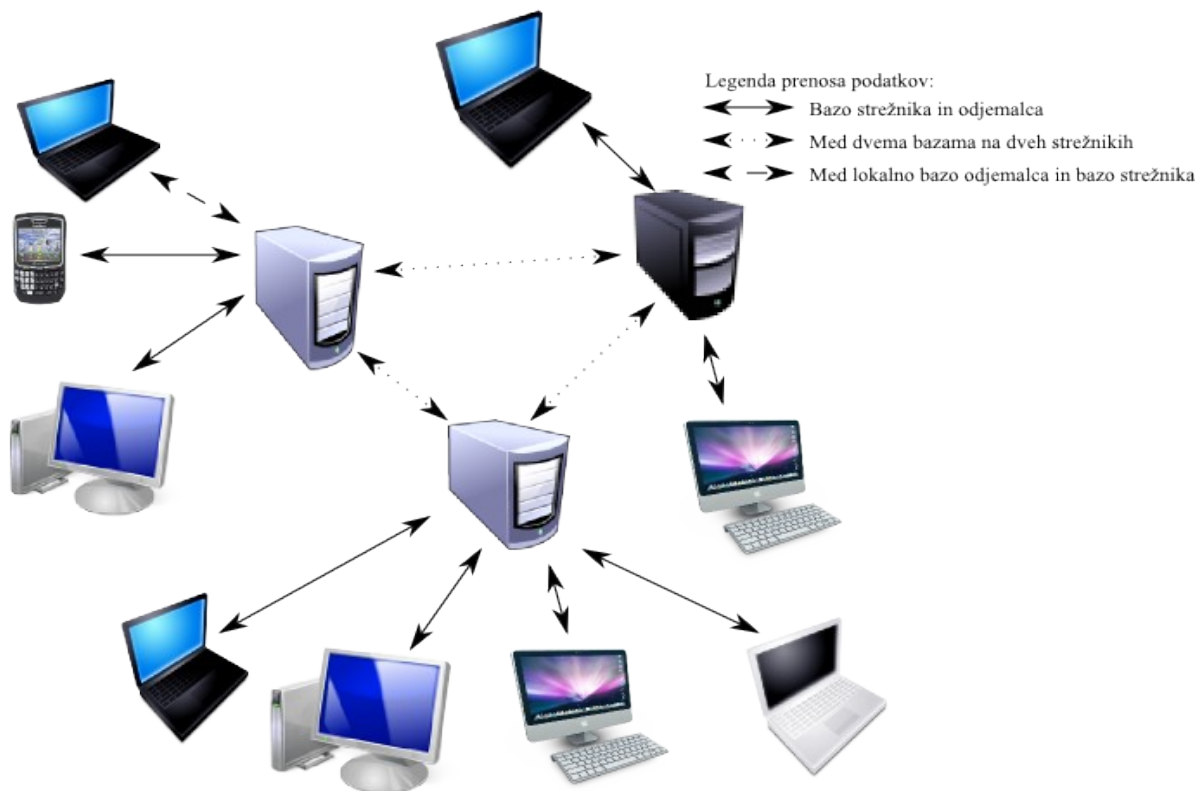
3.3.4. COM/OLE

Glavni cilj te diplomske naloge je integracija pisarniških paketov v dokumentni sistem, ki deluje v okolju Lotus Domino/Notes. Za povezavo okolja Lotus Notes z ostalimi namiznimi aplikacijami se uporablja objektno komponentni model COM (Component Object Model) in povezovanje objektov in vgradnja OLE (Object Linking and Embedding) [9]. Obe tehnologiji delujeta na operacijskem sistemu Microsoft Windows. Za delovanje tehnologije OLE na operacijskem sistemu MAC OS je potrebno na omenjen sistem namestiti podporo za tehnologijo OLE. Pri programiranju kreiramo objekt OLE, kateremu podamo ime objekta aplikacije. Tako kreirani objekt vsebuje povezavo na novo nastali proces programa. Preko povezave OLE lahko uporabljamo funkcije in lastnosti programa. [13]

3.3.5. Repliciranje podatkovnih baz

Posebnost arhitekture Lotus Domino/Notes je možnost repliciranja podatkovnih baz na različnih strežnikih Domino in tudi pri odjemalcih Lotus Notes. Ta posebnost je prikazana na sliki 12, kjer vidimo s pikčastimi puščicami repliciranje podatkovnih baz med dvema strežnikoma. Neprekinjene puščice kažejo direkten dostop odjemalca do strežnika, kjer si odjemalec prenese le želene podatke. S prekinjeno puščico je označeno repliciranje podatkovnih baz med odjemalcem in strežnikom. Pri tem uporabnik dela na lokalni bazi

podatkov na svojem računalniškem sistemu, po končanem delu pa se podatki sinhronizirajo med bazama strežnika in odjemalca. Pri sinhronizaciji se posodobijo le spremenjeni podatki v podatkovni bazi Lotus Notes. Sinhronizacijo lahko uporabniki sprožijo ročno ali pa se izvede avtomatično po vnaprej določeni časovni periodi. Prednost repliciranih podatkovnih baz je v hitrejšem delovanju aplikacij na različnih geografskih lokacijah, možnost dostopa in urejanje podatkov v primeru nemožnosti dostopa uporabnika do strežnika. Podvojeno podatkovno bazo je možno hitro vključiti v sistem, kadar so nastali problemi z glavnim strežnikom.



Slika 12: Prikaz poti podatkov med strežniki in odjemalci.

Za primer bomo prikazali uporabo repliciranja baz v mednarodnem podjetju za prodajo in servisiranje težke strojne mehanizacije. Podjetje s sedežem v Švici ima svoja hčerinska podjetja za področje skupne bivše države Jugoslavije razmeščena v enem izmed večjih mest države v omenjeni regiji. Zaradi do pred kratkim slabih internetnih povezav so postavili strežnike Lotus Domino v vsakem hčerinskemu podjetju. Podatkovne baze se med sabo redno samodejno sinhronizirajo. Z repliciranjem baz so rešili slabo internetno povezavo med posameznimi podjetji in tako pospešili dostop uporabnikov do zelenih podatkov. Zaradi načina dela so serviserji in prodajalci velikokrat primorani delati izven možnosti dostopa do interneta oziroma je hitrost prenosa podatkov zelo počasna. V takem primeru si uporabniki sinhronizirajo svojo kopijo podatkovne baze na svojem računalniku in jo lahko uporabljajo tudi v primeru dela na terenu.

3.4. Varnost v okolju Lotus Domino/Notes

Varnost v poslovnih aplikacijah je področje, na katerega razvijalci informacijskih sistemov ne smemo nikoli pozabiti. Bodisi gre za dostop do podatkovnih baz, dostop do določenih

podatkov, identifikacije uporabnika ali šifriranega prenosa informacij med strežnikom in odjemalcem.

Identifikacija uporabnika poteka preko uporabnikove identifikacijske datoteke, preko katere se uporabnik identificira v sistem. Strežnik Lotus Domino določa pravice uporabnika glede nivoja dostopa do strežnika in strežnikovih lokalnih map.

Posameznemu ali skupini uporabnikov lahko urejamo dostop do podatkovne baze in zapisov preko kontrolne liste dostopa (ACL). Uporabnikom in skupinam uporabnikov v kontrolni listi dostopa določamo nivo dostopa do dokumentov in celotne podatkovne baze, posamezne pravice za delo z bazo in vloge v podatkovni bazi.

Uporabniku lahko na nivoju podatkovne baze omejimo branje, spreminjanje, dodajanje, urejanje oblikovnih dokumentov, kreiranje privatnih agentov, map, pregledov, spreminjanje programske kode, repliciranje, kopiranje dokumentov.

Na nivoju dokumenta lahko uporabnikom omejujemo dostop v posebnih poljih na dokumentu tipa avtor oziroma bralec. Z vpisom vrednosti v omenjena tipa polja določimo pravico urejanja ali branja posameznega dokumenta. Uporabnike lahko vpišemo poimensko, po imenu skupine v katero pripada, oziroma dodeljene vloge uporabniku ali skupini. Nad posameznim poljem v dokumentu lahko izvedemo kriptiranje.

3.5. Primernost okolja za dokumentne sisteme

Na trgu informacijske tehnologije je možno zaslediti veliko dokumentnih sistemov, ki so razviti v okolju Lotus Domino/Notes. Glavni dejavniki za primernost omenjene tehnologije so: struktura dokumentne baze podatkov, urejen nadzor varnosti, podprtost na različnih operacijskih sistemih, delovanje s stalno ali občasno povezavo do strežnika, možnosti povezovanja v druge sisteme v drugih tehnologijah in implementacija uporabniške programske opreme.

Pri informacijskih sistemih razvitih na relacijskih podatkovnih bazah mora razvijalec za vsako želeno novo vrsto podatka spremeniti shemo podatkovne baze, vnosni obrazec, stavke SQL in iskalnik po zapisih. Pri dodajanju nove vrste podatka v podatkovno bazo Lotus Notes je potrebno razvijalcu dodati le novo polje na oblikovni dokument obrazec. Zaradi objektnega zapisa v podatkovno bazo tako razvijalcu ni potrebna skrb nad zapisi v zbirko podatkov, izdelavo iskalnika po vsebini.

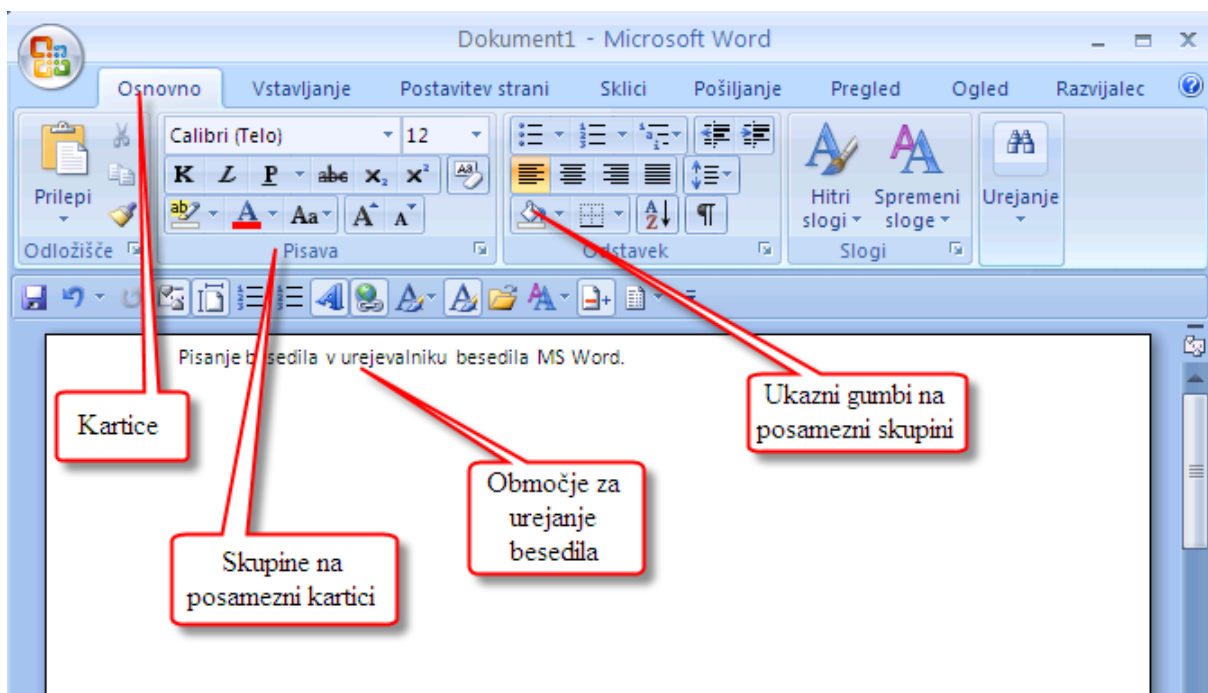
Hiter razvoj aplikacij v okolju Lotus Domino/Notes omogoča prilagodljive informacijske rešitve podjetjem, da se le-te lahko bolje posvečajo poglavitnim ciljem delovanja.

4 PISARNIŠKI PAKETI

Brez uporabniške programske opreme oziroma namiznih aplikacij bi bili računalniki neuporabni. Med najbolj uporabne programe spadajo pisarniški paketi. V osnovi so pisarniški paketi namenjeni pisanju, obdelovanju, pregledovanju besedil, upravljanju e-pošte, preglednic in predstavitev. Poleg omenjenih funkcij vsebujejo tudi ostale dodatne funkcionalnosti, po katerih veliko povprečnih uporabnikov nikoli ne poseže.

4.1. Microsoft Office

Večina ob besedni zvezi pisarniška programska oprema pomisli na pisarniški paket Microsoft Office, v nadaljevanju MS Office. Začetki programja MS Office segajo v leto 1989, ko se pojavi prva izdaja vizualnega urejevalnika besedila "Word" za računalnike Macintosh. Leto kasneje se pojavi za operacijski sistem Windows. Danes smo priče, če izvzamemo prihajajočo verzijo, deseti izdaji omenjenega pisarniškega paketa. To je izdaja MS Office 2007 za Microsoft Windows uporabnike in MS Office 2008 za uporabnike operacijskega sistema Mac OS. Urejevalniku besedila Word so dodajali dodatne pisarniške programe, tako da sedanja polna verzija pisarniškega paketa vsebuje naslednje programe: Word, Excel, PowerPoint, Outlook + Business Contact Manager, Accounting Express, Publisher, Access, InfoPath, Grove, OneNote, Communicator, Enterprise Content Manager, Electronic Form, Information Rights Manager and Policy Capabilities. [14]



Slika 13: Grafični vmesnik pisarniškega programa Word.

4.1.1. Programiranje

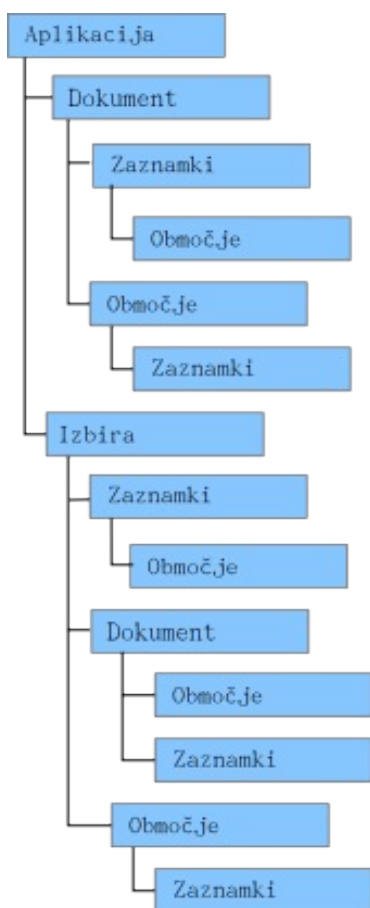
Večini uporabnikov zadostuje grafični vmesnik za delo z besedili, preglednicami, predstavitvami, odjemalcem elektronske pošte. Za zahtevnejše uporabnike, opravila in ponavljajoče naloge imajo programi v pisarniški zbirki možnost pisanja programov, tako

imenovanih makrov. Makri v pisarniški zbirki MS Office so programi pisani v programskem jeziku VBA, dostopni preko urejevalnika programske kode Microsoft Visual Basic. Za zelo uporabno funkcionalnost se obnese snemalnik makrov. Uporabnik si vključi snemalnik, izvede spremembe na dokumentu, izklopi snemalnik in že ima napisano programsko kodo, ki ob poganjanju da enak rezultat, kot ga je določil uporabnik preko grafičnega vmesnika. Posnet program lahko uporabnik popravi in uporabi pri podobnih operacijah.

Do programskih funkcij programov pisarniškega paketa je mogoč dostop iz drugih programov in tehnologij. Ta pomembna lastnost nam omogoča povezovanje informacijskih sistemov s pisarniškim paketom MS Office. Povezovanje okolja Lotus Notes/Domino poteka preko objektov OLE.

Najbolj prepoznaven program v pisarniški zbirki je urejevalnik besedil MS Word. Uporablja se za izdelavo preprostih ali obogatenih besedil s slikami, tabelami, grafi in drugimi elementi.

Urejevalnik besedil je za razvijalce programja dostopen preko objekta „Word.Application“ in vsebuje povezane objekte, ki so prikazani v drevesu objektov (slika 14). Vsak posamezen objekt ima svoje metode in lastnosti dostopne preko programov makrov kot tudi iz drugih aplikacij. [15]



Slika 14: Objekti v MS Word [15]

Med najbolj zmogljive programe v pisarniški zbirki Microsoft Office spada urejevalnik preglednic MS Excel. Odlično se obnese pri preračunavanju velikega števila ponavljajočih se,

preprostih izračunov, hrambi in prikazu večjega števila podatkov različnega tipa, statistični analizi vpisanih podatkov, prikazu podatkov v obliki grafov ter izvozu in prenosu podatkov med dvema različnima informacijskima sistemoma.

V prid moči tega pisarniškega progama podajamo primer mednarodne korporacije, ki ima svoje poslovne enote po celem svetu. Posamezne poslovne enote so združene v regijske centre. V zvezi s poročanjem o kadrih v posameznih podjetjih v Vzhodni in Srednji Evropi smo skupaj z njimi izdelali sistem poročil v dokumentnem sistemu. Kadrovnik v posamezni enoti izdelava mesečno poročilo o novih zaposlitvah, odhodih iz podjetja, bolniških izostankih, izobraževanjih in ostalih pomembnih dogodkih v podjetju. Celotno poročilo zavzema več listov s podatki, samodejnimi preračuni podatkov in vnaprej izdelanih grafov. Na nivoju regije pooblaščen oseba avtomatično kreira mesečno poročilo, kjer se združijo poročila posameznih poslovalnic. Brez vsakega truda ima sedaj nadrejena oseba mesečno poročilo v primerni obliki za vse podrejene poslovne enote v eni datoteki, ki je primerna za nadaljnjo analizo in obdelavo. Ta postopek združevanja poročil gre od posamezne poslovalnice do najvišjih nivojev odločanja v omenjenem mednarodnem podjetju. Ker so podatki vnešeni v preglednico, je kasnejša obdelava teh podatkov enostavna in podprta z močnim orodjem za obdelavo podatkov.

V zgornjem odstavku smo podali primer povezovanja programa MS Excel z dokumentnim sistemom. Pri tem so se poročila avtomatično združevala in shranjevala v dokumentni sistem.

Za razliko od objektov v MS Word-u, kjer večinoma programsko dostopamo in spreminjamo vsebino preko zaznamkov in polj na besedilnem dokumentu, je dostop do podatkov na preglednicah mogoč preko posameznih celic na listih in območij večjega števila celic. Večje število celic se nahaja na posameznem listu in en ali več listov se nahaja v zbirki delovnih listov. Za dostop do programskih funkcij programa MS Excel uporabljamo objekt "Excel.Application".

4.2. OpenOffice.org

Ponudba pisarniške programske opreme se ne začne in konča pri podjetju Microsoft. Med najpomembnejšo alternativo, prej opisanem programskem paketu, spada brezplačni pisarniški paket OpenOffice.org. Odprto kodni pisarniški paket OpenOffice.org je zaščiten pod licencama GNU Lesser General Public License (LGPL) in Sun Industry Standards Source License (SISSL). Licenca LGPL spada v družino licenc GPL in ima vse lastnosti licence GPL, razen ene. Če za svoje programe uporabite dele programske kode zaščitene pod licenco LGPL, vam končnih izdelkov ni potrebno izdati pod licenco GPL. Tako vam ni potrebno izdati izvorne kode vašega programa. Ta licenca omogoča uporabo izvorne kode v lastniških programih. Licenca SISSL uporabnikom omogoča spreminjanje, razširjanje izvorne kode, zagotovljena pa mora biti združljivost. Dodatek pogodbi določa standarde, ki se jih je potrebno držati. [16]. Možnost, ki jo omogoča licenca LGPL, je izkoristilo podjetje IBM in izdalo svoj lastniški pisarniški paket imenovan IBM Lotus Symphony. Namestitev programa Symphony si lahko uporabnik izbere pri namestitvi najnovejše izdaje odjemalca Lotus Notes, oziroma si lahko pisarniški paket brezplačno pridobi s spletne strani podjetja IBM. [17]

Pri razvoju pisarniškega paketa OpenOffice.org sta imeli pomembno vlogo podjetji StarDivision in Sun Microsystems. Podjetje StarDivision je leta 1984 izdalo urejevalnik besedil StarWriter, kateremu so kasneje dodali ostale pisarniške programe in izdali pisarniški paket pod blagovno znamko StarOffice. Prelomno leto v razvoju pisarniškega paketa je bilo 1999, ko je podjetje Sun Microsystems kupilo podjetje StarDivision. Sun Microsystems je z

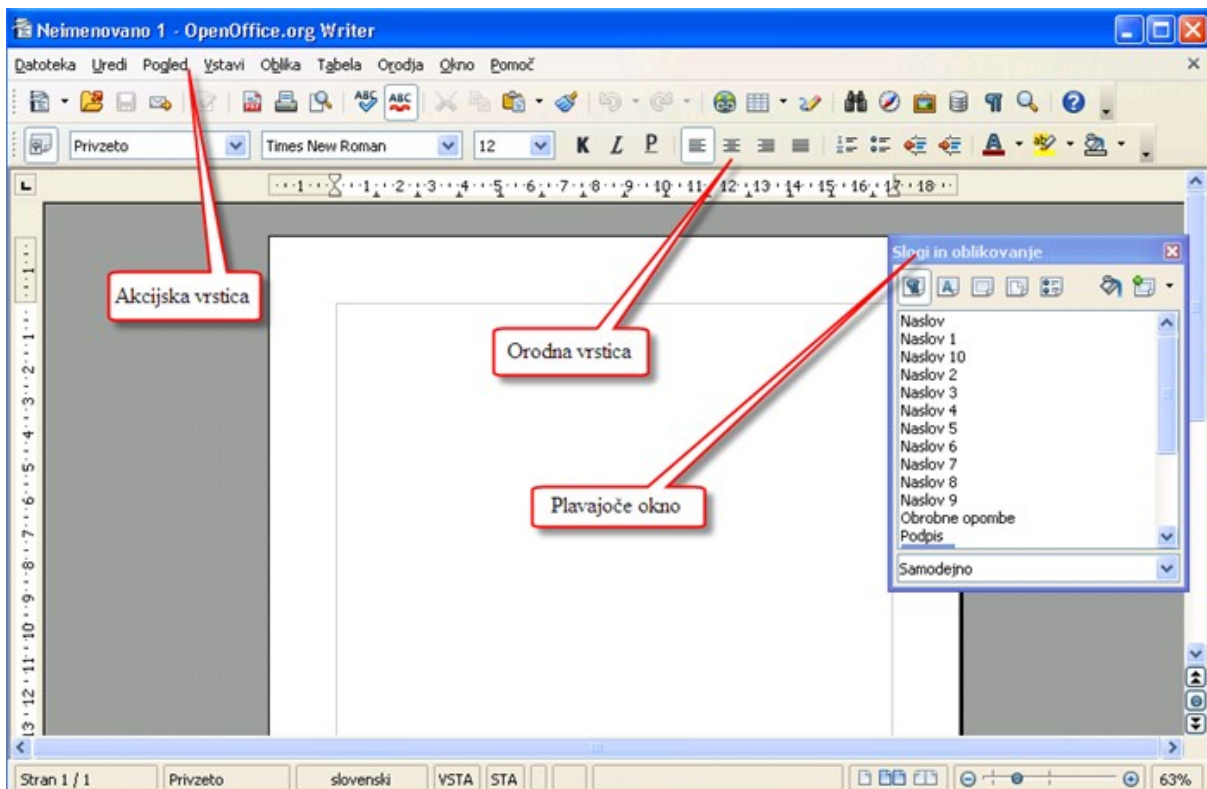
nakupom podjetja pridobil blagovno znamko StarOffice in vse ostale pravice podjetja. V letu 2000 je podjetje Sun Microsystems objavil izvorno kodo pisarniškega paketa StarOffice pod prej omenjenimi licencami in blagovno znamko OpenOffice.org [18]. Sedanje izdaje pisarniškega paketa StarOffice vsebujejo pisarniški paket OpenOffice.org in dodatno programje. Tako z nakupom pisarniškega paketa StarOffice pridobimo dodatne razširitve, odjemalca elektronske pošte, koledar, popravke, razširitve in tri brezplačne klice na podporo podjetja Sun Microsystems [18]. Trenutno največ v razvoj pisarniškega paketa OpenOffice.org prispeva podjetje Oracle, ki je nedavno kupilo podjetje Sun Microsystems. V nadaljevanju bomo omenjali le pisarniški paket OpenOffice.org, oba ostala temeljita na omenjenemu paketu in se bistveno vsebinsko ne razlikujeta.

Pisarniški paket OpenOffice.org vsebuje urejevalnik besedil Writer, urejevalnik preglednic Calc, urejevalnik grafik Draw, program za predstavitve Impress, program za izračun enačb Math in podatkovno bazo Base.

Vsi trije pisarniški paketi, omenjeni v tem poglavju, so med sabo popolnoma kompatibilni in uporabljajo isti format zapisa datotek. Poleg tega vsi trije znajo izdelovati, brati in spreminjati dokumente pisarniškega paketa MS Office.

Pri industrijskem združenju OASIS (Organizacija za napredek standardov za strukturirane informacije) so razvili odprti datotečni format OpenDocument (ODT). Format temelji na datotečnem formatu na osnovi strukture XML, katerega so razvili pri OpenOffice.org. Standard je prosto dostopen, kar pomeni, da ga lahko implementira vsakdo, brez kakršnekoli omejitve. Najpogostejše končnice v formatu OpenDocument so: .odt za besedilne dokumente, .ods za preglednice, .odp za predstavitve in .odg za grafiko. Format ODT so podprli vsi vodilni proizvajalci pisarniških programov vključno s podjetjem Microsoft, ki pogosto zagovarja svoje zaprte standarde. Tako pisarniški paket MS Office 2007 že podpira format zapisa ODT. [19]

Pisarniški paket OpenOffice.org deluje na vseh bolj poznanih operacijskih sistemih. Srečamo ga na operacijskih sistemih: Microsoft Windows, Mac OS, Linux, Solaris in FreeBSD. Odprto kodna skupnost poskuša zagotavljati jezikovno podporo v čim večjem možnem številu jezikov, tako je program razširjen po celem svetu. Grafični vmesnik OpenOffice.org je podoben vmesnikom ostalih pisarniških paketov.



Slika 15: Grafični vmesnik urejevalnika besedila Writer.

4.2.1. Programiranje

V pisarniškem paketu OpenOffice.org je možno pisati programe makre, podobno kot v pisarniškem paketu MS Office. Pišemo jih lahko v programskih jezikih OpenOffice.org Basic, Python, BeanShell in JavaScript. V jeziku OpenOffice.org Basic je mogoče posneti tudi programe makre, za razliko od makrov posnetih v pisarniškem paketu MS Office so težje berljivi. Zaradi tega je neizkušen razvijalec primoran uporabiti več iznajdljivosti pri programiranju in uporabiti več iskanja po svetovnem spletu.

Skupnost OpenOffice.org je razvila tehnologijo UNO (Universal Network Objects). UNO je sestavni del modela za OpenOffice.org, omogoča sodelovanje med programskimi jeziki, programi v pisarniškem paketu in strojno opremo. Tehnologija omogoča komunikacijo med posameznimi programi v pisarniški zbirki na lokalnem računalniku, lokalni mreži ali preko svetovnega spleta [20]. Objekte UNO najlažje opazimo, če pogledamo posneti makro v urejevalniku besedil ali preglednic (slika 16).

```

sub Pisanje
rem -----
rem define variables
dim document as object
dim dispatcher as object
rem -----
rem get access to the document
document = ThisComponent.CurrentController.Frame
dispatcher = createUnoService("com.sun.star.frame.DispatchHelper")

rem -----
dim arg1(0) as new com.sun.star.beans.PropertyValue
arg1(0).Name = "Text"
arg1(0).Value = "Pisanje besedila"

dispatcher.executeDispatch(document, ".uno:InsertText", "", 0, arg1())
end sub

```

Slika 16: Posnet makro ob pisanju besedila.

V okolju Windows lahko do pisarniške zbirke OpenOffice.org dostopamo s pomočjo tehnologije OLE/COM. Z omenjeno tehnologijo lahko izdelamo programski vmesnik, ki zunaj pisarniškega paketa upravlja s funkcijami pisarniškega paketa. Za razvoj takega vmesnika potrebujemo programski jezik, ki podpira omenjeno tehnologijo. Na seznamu teh jezikov so: Visual C++, Visual Basic, Delphi, VBScript, JScript in LotusScript [21].

Pred klici funkcij pisarniškega paketa OpenOffice.org moramo ustvariti ustrezne objekte. Programska koda z razlago se nahaja na sliki 17.

```

Dim oServiceManager As Variant 'Objekt upravitelja storitev
Dim mProperties(0) As Variant 'Spremenljivka za podajanje parametrov
Dim fileName As String 'Pot do datoteke
Dim oDesktop As Variant 'Objekt namizja
Dim officeApp As Variant 'Objekt urejevalnika

'Nastavim pot do datoteke z besedilom.
fileName="file:///C:/besedilo.odt"

'Nastavim upravitelja storitev.
Set oServiceManager = CreateObject("com.sun.star.ServiceManager")

'Nastavim lastnost "Hidden" na "True".
Set mProperties(0) = oServiceManager.Bridge_GetStruct("com.sun.star.beans.PropertyValue")
mProperties(0).Name = "Hidden"
mProperties(0).Value = True

'Kreiram objekt namizja.
Set oDesktop = oServiceManager.createInstance("com.sun.star.frame.Desktop")

'Odprem datoteko besedila in objekt z datoteko shranim v spremenljivko officeApp.
Set officeApp= oDesktop.loadComponentFromURL(fileName, "_blank", 0, mProperties)

```

Slika 17: Kreiranje objektov za delo z dokumenti.

Kot vidimo na zgornji sliki, smo kreirali objekt upravitelja storitev, preko katerega smo pridobili namizje in na njem odprli dokument. Na namizju na novo kreiramo prazen oziroma odpremo obstoječ dokument programa v pisarniški zbirki.

Interakcija med programi poteka preko storitev na vmesniku za programiranje aplikacij (API). Seznam podprtih storitev v vmesniku API za OpenOffice.org je dolg, zato bomo omenili le najpomembnejše:

- **com.sun.star.chart** vsebuje zbirko storitev za diagrame,
- **com.sun.star.drawing** vsebuje zbirko storitev za risanje črt, krogov, pravokotnikov...
- **com.sun.frame** vsebuje zbirko storitev namizja, preko katerega odpiramo obstoječi dokument ali kreiramo novega,
- **com.sun.presentation** vsebuje storitve za delo s predstavitvami,
- **com.sun.sheet** vsebuje storitve za delo s preglednicami,
- **com.sun.table** vsebuje storitve za delo s tabelami,
- **com.sun.text** vsebuje storitve za delo s tekstovnimi dokumenti.

Prednost dela s storitvami je v uporabi storitev ne glede na vrsto obdelovanega dokumenta. Tako lahko za delo s predstavitvami uporabimo storitve za delo s tekstovnimi dokumenti.

Elementi na dokumentu so razvrščeni v oštevilčenih zbirkah elementov, ločenih po tipu elementov. Tako lahko upravljamo s tekstovnim poljem s pomočjo funkcije "getTextField().createEnumeration()", ki nam vrne zbirko tekstovnih polj. Do posameznih listov v dokumentu preglednic lahko dostopamo preko zbirke, ki jo pridobimo s pomočjo funkcije "getSheets().getByIndex(1)", kateri podamo zaporedno številko lista preglednice.

Vsem elementom na dokumentu pripada identična zaporedna številka glede na tip elementa in ne identično ime elementa. [22]

V podrobnosti programiranja in programskih funkcij se ne bomo spuščali. Več informacij o programiranju v pisarniškem paketu OpenOffice.org je mogoče najti v literaturi za pisarniški paket StarOffice.

Izkušnje kažejo, da večini uporabnikov zadošča pisarniški paket OpenOffice.org, kljub temu pa se le-ta ne pojavlja pogosto v pisarnah.

5 UPRAVITELJ PREDLOG PISARNIŠKIH PAKETOV

Delo v pisarnah pogosto zahteva izdelavo slogovno, grafično in vsebinsko podobnih dokumentov. Gre za dopise poslovnim partnerjem, strankam, za dokumente znotraj podjetja ali le izpis množice podatkov za kasnejšo obdelavo. Integracija pisarniškega paketa v dokumentni sistem omogoča hitro, poenoteno izdelavo dokumentov.

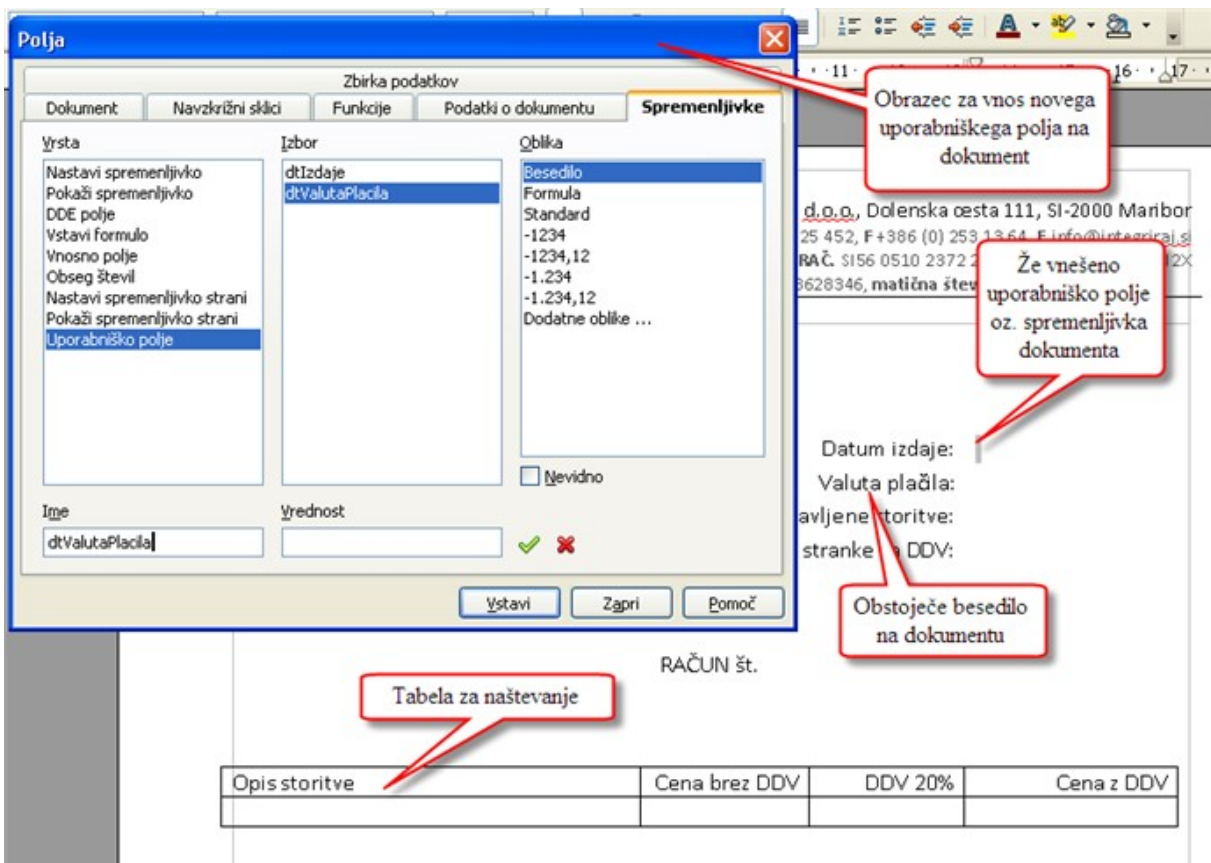
Za uspešno implementacijo pisarniškega paketa v informacijski sistem igra glavno vlogo razvit sistem za implementacijo pisarniškega paketa v dokumentni sistem, tako imenovan upravitelj predlog pisarniških programov, v nadaljevanju upravitelj predlog. Z razvitim upraviteljem predlog skrajšamo čas pri implementaciji predloge v dokumentni sistem, spreminjanju, administraciji in zmanjšamo dodatne stroške stalnega razvoja dokumentnega sistema. Dober programski vmesnik upravitelja predlog omogoča nastavitvev predloge preko grafičnega vmesnika s čim manj programiranja in podvajanja programske kode.

5.1. Predloga pisarniškega paketa

Za razumevanje programskega spreminjana vrednosti na dokumentu moramo vedeti, kaj bomo spremenili oziroma na katerem mestu dokumenta bomo nastavili želeno vrednost. S tem smo določili glavne zahteve, ki so potrebne za nastavitvev spreminjajočih vrednostih na predlogi, shranjeni v elektronskem arhivu dokumentnega sistema. Predloga dokumenta služi za izdelavo novega dokumenta, pri katerem so nespremenljivi podatki shranjeni že na sami predlogi in programsko spremenimo le del vrednosti. Uporabnik lahko po sami sprožitvi preslikav poljubno spreminja vrednosti na novo nastalem dokumentu.

Polja spremenljivk na dokumentu besedila so gradniki, do katerih dostopamo s pomočjo njihovega imena. Polja spremenljivk se uporabljajo za podatke, ki so na primer odvisni od posameznega prejemnika (naziv, naslov, poštna številka prejemnika) in se prenesejo iz dokumentnega sistema v pisarniški program. Vrednosti, kot so glava, noga, naziv zadeve in del vsebine, so predhodno nastavljen na predlogi dokumenta in se pri preslikavi ne spremenijo. Uporabnik pri izdelavi dopisa dopiše le vsebino dokumenta oziroma mu lahko dokumentni sistem izdela že dokončen dopis. Drug pomemben element pri integraciji urejevalnika besedila v dokumentni sistem je tabela vrednosti, do katere dostopamo preko zaznamka. Tabela se uspešno uporablja pri naštevajočih vrednostih, kot so artikli in cene na izhodnem računu.

Na sliki 18 je prikazan vnos nove spremenljivke oziroma uporabniškega polja "dtValutaPlacila" na predlogu izhodnega računa, izdelanega s pisarniškim paketom OpenOffice.org. Prikazana je tudi tabela, ki se tekom polnjenja dinamično povečuje in je zaradi tega primerna za naštevajoče vrednosti. Pri programu MS Word pisarniškega paketa MS Office je način vnosa podoben.



Slika 18: Dodajanje novega uporabniškega polja na besedilni dokument v pisarniškem paketu OpenOffice.org.

Preglednice so močno orodje za obdelavo podatkov. Posamezen dokument je lahko sestavljen iz večjega števila preglednic, na katerih se nahaja velika množica celic določene oblike. Do posamezne celice je možen dostop z naslavljanjem preglednice in celice ali območja celic. Pomembna lastnost pri preglednicah v obeh omenjenih pisarniških paketih je kopiranje celic, vrstic ali stolpcev, pri katerih se ohrani oblika, vrednost in formula celice. Poleg omenjenega se na novo določijo formule, katere se relativno sklicujejo na kopirane celice. To nam omogoča izdelavo predloge, na katero dodamo ustrezen graf, ki se sklicuje na mesto masovnega vnosa podatkov. Graf na izdelanem dokumentu nam predstavi izpisane podatke v nam bolj prilagojenem načinu.

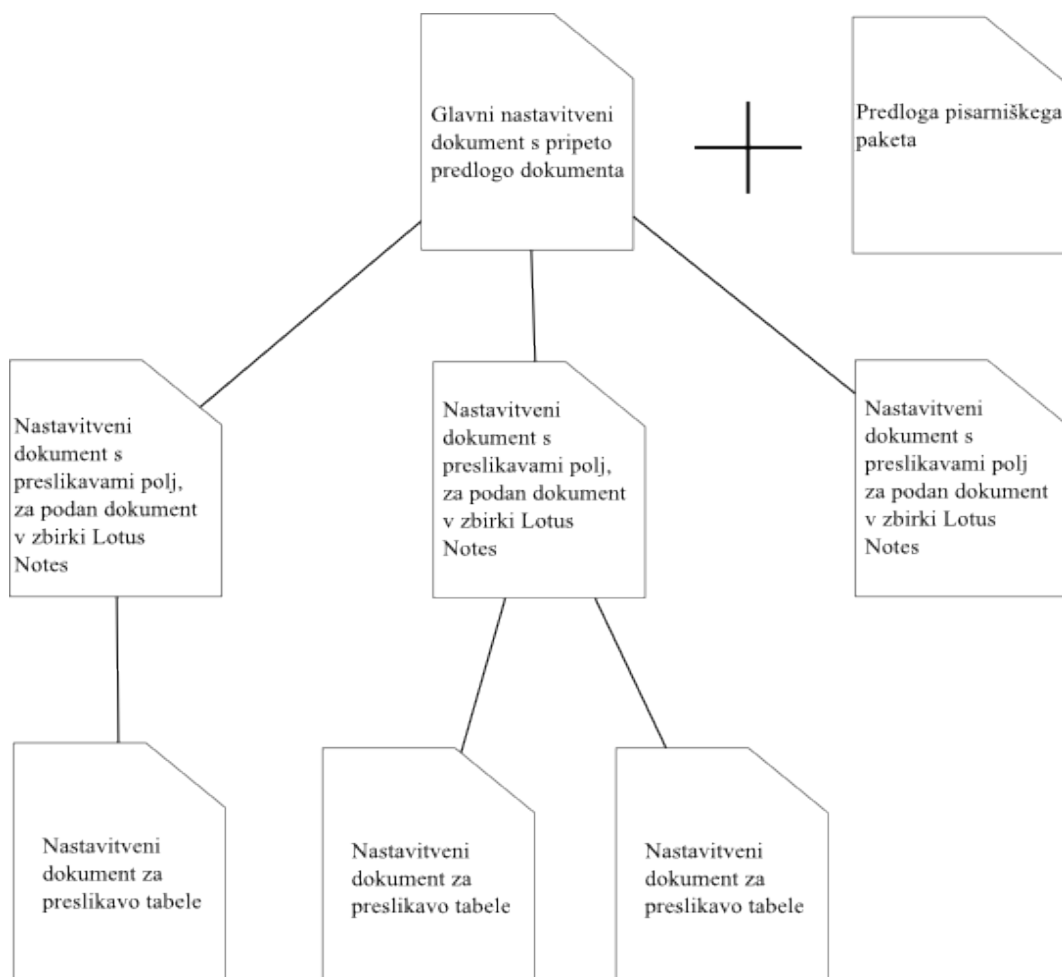
5.2. Upravitelj predlog za dokumentni sistem 3K Document Cycle

V okviru diplomskega dela smo razvili dokumentni sistem 3K Document Cycle, namenjen obvladovanju poslovne dokumentacije in poslovnih procesov. Omenjeni sistem je že imel okrnjen vmesnik upravitelja predlog za izdelavo predlog v pisarniškem programu MS Word, ki je del pisarniškega paketa MS Office. Vmesnik je omogočal izdelavo besedilne pisarniške predloge s preslikavami polj in tabel na dokumentu. Imel je številne pomanjkljivosti, kot so: ne možnost določanja posebne akcije za urejanje predloge, določitev predloge glede na formulo. Pri preslikavah tabel ni bilo možnosti iskanja dokumentov po bazi s pomočjo iskalne formule in ni bilo sortiranja podatkov. Poleg tega ni deloval za preglednice in pisarniški paket OpenOffice.org.

Najtežja naloga pri samem razvoju vmesnika upravitelja predlog je zagotovitev nemotenega prestopa iz starejše izdaje vmesnika na novejšo izdajo pri delujočih dokumentih sistemih pri strankah. Posamezne stranke imajo različne potrebe glede uporabe pisarniških programov in temu je sledil tudi razvoj posamezne inštalacije dokumentnega sistema. Tako smo velikokrat uporabili, dedovali in spremenili programske funkcije vmesnika in si s tem otežili izdajo novih izdaj. Omenjeno je izvajalo tudi partnersko podjetje, ki ponuja omenjeni dokumentni sistem na trgu.

5.2.1. Grafični vmesnik nastavitvev

Na sliki 19 so na diagramu prikazana razmerja med nastavitvenimi dokumenti upravitelja predlog. V prvi vrsti sta prikazana dva dokumenta. Prvi je glavni nastavitveni dokument Lotus Notes, na katerega je pripeta predloga dokumenta izdelanega v enem izmed programov pisarniškega paketa. Na vsakem glavnem nastavitvenem dokumentu se lahko pojavi en ali več nastavitvenih dokumentov s preslikovalnimi podatki. Preslikujemo lahko enega ali več produkcijskih dokumentov Lotus Notes v spremenljivke oziroma celice preglednice na dokumentu, ki je izdelan v pisarniškem paketu. Vsakemu nastavitvenemu dokumentu s preslikavami lahko dodamo nastavitvene dokumente za masovno polnjenje podatkov v tabelo urejevalnika besedil ali preglednico.



Slika 19: Prikaz nastavitvenih dokumentov.

Na sliki 20 je prikazan glavni nastavitveni dokument. Na njem nastavimo tip predloge, pisarniški paket, predlogo in po potrebi določimo akcijo, ki je vidna iz dokumentnega sistema. Možno je nastaviti dodatne parametre, ki se upoštevajo po izvedbi preslikave. Tako nam lahko predloga služi samo za tiskanje dokumentov Lotus Notes. V spodnjem delu obrazca se nahaja pogled z nastavitvami preslikav za podane dokumente.

The screenshot shows the 'Izhodni račun' configuration page. It is divided into several sections: 'Integrator', 'Predloga', 'Akcije', 'Dodatne možnosti', and 'Preslikave'. Red callout boxes point to specific settings:

- Osnovne nastavitve integratorja:** Points to the 'Integrator' section, specifically the 'Šifra' (IR2) and 'Naziv' (Izhodni račun) fields.
- Izbira ustrezne predloge glede na formulo:** Points to the 'Predloga' section, specifically the 'Privzeta predloga' (IR2.doc) and the 'jezik="ANG"' filter.
- Akcije na gumbu:** Points to the 'Akcije' section, specifically the 'Gumb akcije' checkbox and the 'Naziv akcije' (Račun) field.
- Dodatne možnosti po izvedbi integracije:** Points to the 'Dodatne možnosti' section, specifically the 'Zakleni datoteko' and 'Ne prikaži datoteke' checkboxes.
- Pogled z dokumenti preslikav:** Points to the 'Preslikave' section, specifically the 'Nova preslikava' button and the table below.

Šifra	Naziv
P01	Izhodni račun

Slika 20: Glavni nastavitveni dokument upravitelja predlog.

Za vsak podani tip dokumenta iz podane zbirke dokumentov za preslikavo je potrebno določiti nov nastavitveni dokument, prikazan na sliki 21. Na tem nastavitvenem dokumentu določimo vir podatka, najpogosteje polje Lotus Notes ali formulo za izračun vrednosti iz dokumenta Lotus Notes in ciljni element na dokumentu predloge. Pod ciljni element mislimo polje na besedilnem dokumentu, celico preglednice ali naziv preglednice.

Preslikava	Integrator																		
Šifra P01	Šifra IR2																		
Naziv Izhodni račun	Tip predloge Besedilo Preglednica																		
Zaznamki	Vklop preslikave zaznamkov																		
Lotus Notes polje	Besedilno polje																		
<pre> zapSTSub podjetje kontaktNarocnik naslov TXPosta + " " + kraj dtRac dtValuta dtStoritev @text(znesekBrezDDV;"F,2") @text(DDV;"F,2") @text(znesek;"F,2") str := @Subset(@DbName;1);potNastavi := @GetProfileField("dbprofile";"kljucSif");nacin := @DbLookup("";"nocache";str.potNastavi;"modul";"PART";"lokalniNacin");@If(@IsError(nacin);@Return(""));strPart:=@If(str=""&nacin="1";@DbLookup("";"nocache";str.potNastavi;"m odul";"PART";"streznikRep");@DbLookup("";"nocache";str.potNastavi;"modul";"PART";"stre znik");potPart:=@If(str=""&nacin="1";@DbLookup("";"nocache";str.potNastavi;"modul";"PA RT";"potRep");@DbLookup("";"nocache";str.potNastavi;"modul";"PART";"pot");id:=@DbLoo kup("";"nocache";strPart.potPart;"PartnerjiSifra";idPartner;"identifikajska");@If(@IsError(id);@Return(""));id); stNarocPog </pre>	<pre> zapSt podjetje kontakt naslov posta datum dtRok dtopravi brezDDV DDV znesek idStevilka stNarocPog </pre>																		
Tabele																			
Nova tabela																			
<table border="1"> <thead> <tr> <th>Tabela</th> <th>Stolpec1</th> <th>Stolpec2</th> <th>Stolpec3</th> <th>Stolpec4</th> <th>Stolpec5</th> <th>Stolpec6</th> <th>Stolpec7</th> <th>Stolpec8</th> </tr> </thead> <tbody> <tr> <td>t1</td> <td>tmpNaziv</td> <td>tmpKolicina</td> <td>tmpCena</td> <td>tmpDavek</td> <td>tmpPopust</td> <td>tmpVrednost</td> <td></td> <td></td> </tr> </tbody> </table>	Tabela	Stolpec1	Stolpec2	Stolpec3	Stolpec4	Stolpec5	Stolpec6	Stolpec7	Stolpec8	t1	tmpNaziv	tmpKolicina	tmpCena	tmpDavek	tmpPopust	tmpVrednost			
Tabela	Stolpec1	Stolpec2	Stolpec3	Stolpec4	Stolpec5	Stolpec6	Stolpec7	Stolpec8											
t1	tmpNaziv	tmpKolicina	tmpCena	tmpDavek	tmpPopust	tmpVrednost													

Slika 21: Nastavitveni dokument preslikav.

Tretji nastavitveni dokument, prikazan na sliki 22 je namenjen definicijam za masovno polnjenje tabele besedilne datoteke in preglednice. Na levi strani obrazca imamo nastavitve vira preslikave, začetne pozicije preslikave in pogoj za neizvajanje preslikave tabele. Desni gornji del je namenjen informacijam o preslikavi, pod njim se nahajajo dodatne možnosti za delo s tabelo. Na spodnjem delu je prikazana tabela preslikav v stolpce. Glede na vir podatkov in pripadajočo formulo za omenjeni stolpec se podatki preslikajo v tabelo.

Preslikava tabele	Integrator										
<input checked="" type="checkbox"/> Vklop preslikave tabele	Tip predloge Besedilo Preglednica										
Številna polja: Število stolpcev 5, Začetna vrstica 5, Začetni stolpec 2	Šifra integratorja TEST_EXCEL, Šifra preslikave P01, Šifra tabele T01										
Način pridobitve podatkov: DB search, Programsko preko dbSearchFormulasList	Dodatni parametri: Kopiranje vrstic Ne Da Brisanje dodatne, Avtomatska razširitev stolpcev Ne Da Z predhodno, Sortiranje Narasčujoče Padajoče Brez sortiranja										
Modul za iskanje PELIKAN, Formula za iskanje form="dok"	Formula za sortiranje, Stolpec za sortiranje 4										
<table border="1"> <thead> <tr> <th>Stolpec</th> <th>Polje oz. formula</th> </tr> </thead> <tbody> <tr> <td>Stolpec 1</td> <td>leto</td> </tr> <tr> <td>Stolpec 2</td> <td>sub</td> </tr> <tr> <td>Stolpec 3</td> <td>vrsta</td> </tr> <tr> <td>Stolpec 4</td> <td>@TextToNumber(zapSt)</td> </tr> </tbody> </table>	Stolpec	Polje oz. formula	Stolpec 1	leto	Stolpec 2	sub	Stolpec 3	vrsta	Stolpec 4	@TextToNumber(zapSt)	
Stolpec	Polje oz. formula										
Stolpec 1	leto										
Stolpec 2	sub										
Stolpec 3	vrsta										
Stolpec 4	@TextToNumber(zapSt)										

Slika 22: Preslikava tabele

5.2.2. Implementacija v dokumentni sistem 3K Document Cycle

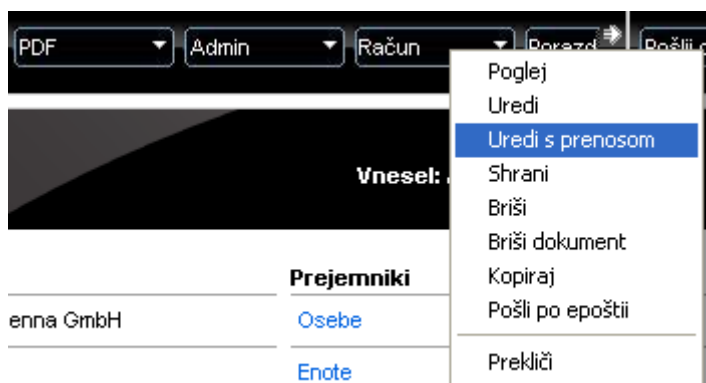
Zgradba dokumentnega sistema 3K Document Cycle temelji na prilagodljivosti strankinim potrebam, hitri postavitvi in enostavni administraciji. Temu sodi tudi učinkovita implementacija modula upravitelja predlog v dokumentni sistem. Dokumenti v glavni dokumentni bazi imajo skupen glaven obrazec in poljubne pod-obrazce glede na ustrezno vrsto dokumenta. Zaradi tega je glavni obrazec najprimernejšo mesto za postavitev akcijskega gumba za upravljanje predlog. Poleg akcijskega gumba mora imeti baza s produkcijskimi dokumenti tudi programsko knjižnico upravitelja predlog in omogočen dostop do elektronskega arhiva in nastavitvene baze vmesnika upravitelja predlog.

Pri ustvarjanju novega dokumenta v dokumentnem sistemu se iz nastavitvenega dokumenta izbrane vrste dokumenta prenesejo ključni podatki za določitev pod-obrazca, pravice uporabnikov nad dokumentom, podatki za delovanje delovnega procesa in seveda tudi šifra predloge. Glede na šifro predloge se ob kliku na akcijski gumb preračunajo možne akcije za upravljanje predloge in seveda se ob izbiri ustrezne akcije le-ta tudi izvede. Na sliki 23 je prikazano ustvarjanje novega računa s pomočjo predloge. Naziv same akcije "Račun" se je prenesel iz glavnega nastavitvenega dokumenta upravitelja predlog.



Slika 23: Ustvarjanje računa s pomočjo akcijskega gumba.

Na sliki 24 so prikazane akcije, določene na glavnem nastavitvenem dokumentu predloge.



Slika 24: Akcije na že ustvarjenem računu.

Zgornji primer prikazuje najpogostejšo uporabo upravitelja predlog v dokumentnem sistemu 3K Document Cycle.

Izkušnje kažejo, da je nemogoče združiti vse uporabniške zahteve v nastavitvene dokumente oziroma bi bil razvoj ustreznega modula časovno in finančno preobsežen. Za take primere je omogočena uporaba programskih objektov in funkcij upravitelja predlog. Na sliki 25 je prikazana uporaba programskih objektov in funkcij za preprost izvoz dokumentov Lotus Notes, izdelanih v letu 2010. Objekt "OfficeIntegrator" predstavlja programski objekt upravitelja predlog.

```

'Deklaracija objekta OfficeIntegrator.
  Dim office As officeIntegrator

'Po potrebi kreiramo splošne objekte.
If sUi Is Nothing Then
  Set sui = New myuisession
  Set appui = New pelikanuiapp(sui, "")
End If

'Creiranje objekta OfficeIntegrator s parametri:
'baza nastavitvev, šifra integratorja, morebitni trenutni dokument in pot do morebitne obstoječe predloge.
Set office = New officeIntegrator(appui.app.sifpelikanApp, "POGLED_IZVOZ", Nothing, "")

'Nastavitev modula in iskalne formule za določeno preslikavo tabele.
office.dbSearchModul("POGLED_IZVOZ#P01#T01") = "SIFPELIKAN"
office.dbSearchFormulaList("POGLED_IZVOZ#P01#T01") = {@year(@created)=2010}

'Izvedba preslikav in prikaz kreiranega dokumenta v ustreznem pisarniškem paketu.
Call office.IntegrateTo()
Call office.officeSetVisible(True)

```

Slika 25: Uporaba objektov in funkcij upravitelja predlog.

Z malo več ustvarjalnosti in programske kode je možna izdelava močnega orodja za izvoze podatkov iz podatkovnih baz Lotus Notes v preglednice. Pri tem izkoristimo zmogljivost pisarniških paketov in obogatimo dragocene podatke z grafikoni.

6 SKLEPNE UGOTOVITVE

Namen diplomske naloge je bil v izdelavi vmesnika upravitelja predlog za preslikavo dokumentov Lotus Notes v dokumente pisarniških paketov. Vmesnik skrajša razvoj dokumentnega sistema, skrajša čas za izdelavo dokumentov s pomočjo predlog pisarniškega paketa in poenostavi vzdrževanje dokumentnega sistema. O kakršnikoli številčni oceni o uspešnosti razvoja vmesnika je težko govoriti, saj ni namenjen neposredni prodaji na trgu, temveč pomoči razvijalcem pri implementaciji programov pisarniških paketov v dokumentni sistem 3K Document Cycle. Uporabniki omenjenega dokumentnega sistema se še niso srečali z novim vmesnikom upravitelja predlog zaradi strankinih specifičnih zahtev v posamezni inštalaciji dokumentnega sistema. Te zahteve so nam dale nove ideje o samih temeljih dokumentnega sistema, katere morajo biti realizirane pred implementacijo novega vmesnika upravitelja predlog v obstoječe in nove inštalacije dokumentnih sistemov. Pri samem razvoju vmesnika smo poglobili svoje znanje o dokumentnih sistemih, konkurenčnih sistemih na trgu, pisarniških paketih in zakonodaji o arhiviranju dokumentov v elektronski obliki.

Pred izdelavo te diplomske naloge sem v praksi poznal le pisarniški paket Microsoft Office in sem bil zmotno prepričan o njegovi nenadomestljivosti v primerih zahtevnejših pisarniških opravil. Svojo zmoto sem spoznal ob izdelavi tega diplomskega dela v pisarniškem paketu OpenOffice.org. Odprto kodni pisarniški paket sem spoznal s strani uporabnika, kot pisec diplomskega dela in razvijalec pri implementaciji pisarniškega paketa v dokumentni sistem. Pri izdelavi diplomskega dela sem spoznal, da je pisarniški paket OpenOffice.org dobra alternativa plačljivemu lastniškemu paketu MS Office. Grafična vmesnika in način dela v obeh pisarniških paketih je zelo podoben, zato naj stroški izobraževanja pri morebitnem prehodu podjetij na odprto kodni pisarniški paket ne bi dosegali visokih zneskov. Podjetja bi morala večji del finančnih sredstev nameniti za spremembo dokumentnih predlog in za spremembo morebitnih implementacij programov pisarniškega paketa v svojih informacijskih sistemih. Ob tem je potrebno poudariti, da bi se omenjeni vloženi strošek na dolgi rok povrnil podjetjem zaradi brezplačnosti pisarniškega paketa OpenOffice.org. Edino slabost v odprto kodnem pisarniškem paketu vidim v slabi podpori za razvijalce. Ta slabost bi se dokaj hitro odpravila v primeru, da bi se podjetja in javne ustanove bolj pogosto odločale za prehod na odprto kodni pisarniški paket. Povpraševanje po znanju programiranja za pisarniški paket OpenOffice.org bi spodbudilo dodaten razvoj podpore za razvijalce in tudi samega pisarniškega paketa. S tem bi podjetja in javne ustanove podrla domača informacijska podjetja, namesto da kupujejo programske licence od podjetij iz tujine. Odprto kodni pisarniški paket me je navdušil nad odprto kodo, tako da sem sredi izdelave diplomske naloge prešel na domačem računalniku na odprto kodni operacijski sistem.

SLIKE

Slika 1: Povezava med dokumenti, zadevami in udeleženci v pravni pisarni.....	6
Slika 2: Vnos dokumentov v dokumentni sistem.....	7
Slika 3: Odjemalec Lotus Notes.....	11
Slika 4: Domino Designer.....	12
Slika 5: Lotus Domino podatkovna baza.....	13
Slika 6: Oblikovni dokument pogled kot ga vidi razvijalec.....	14
Slika 7: Oblikovni dokument obrazec kot ga vidi razvijalec.....	14
Slika 8: Pregled dokumentov kot ga vidi uporabnik.....	16
Slika 9: Primer izbirne formule na pogledu.....	17
Slika 10: Prikaz "Front-End" in "Back-End" razredov na zaslonski sliki odjemalca Domino Designer.....	18
Slika 11: Programska koda za izbiro dokumentov na "Front-End" razredu.....	19
Slika 12: Prikaz poti podatkov med strežniki in odjemalci.....	20
Slika 13: Grafični vmesnik pisarniškega programa Word.....	22
Slika 14: Objekti v MS Word.....	23
Slika 15: Grafični vmesnik urejevalnika besedila Writer.....	26
Slika 16: Posnet makro ob pisanju besedila.....	27
Slika 17: Kreiranje objektov za delo z dokumenti.....	28
Slika 18: Dodajanje novega uporabniškega polja na besedilni dokument v pisarniškem paketu OpenOffice.org.....	31
Slika 19: Prikaz nastavitvenih dokumentov.....	32
Slika 20: Glavni nastavitveni dokument upravitelja predlog.....	33
Slika 21: Nastavitveni dokument preslikav.....	34
Slika 22: Preslikava tabele.....	34
Slika 23: Ustvarjanje računa s pomočjo akcijskega gumba.....	35
Slika 24: Akcije na že ustvarjenem računu.....	35
Slika 25: Uporaba objektov in funkcij upravitelja predlog.....	36

Viri

- [1] (2010) Document Management system. Dostopno na: http://en.wikipedia.org/wiki/Document_management_system
- [2] (2010) Upravljanje z dokumenti. Dostopno na: http://www.3k-it.si/produkti/3k_doc_pregled.html
- [3] (2010) European Article Number. Dostopno na: http://en.wikipedia.org/wiki/European_Article_Number
- [4] (2010) Optical character recognition. Dostopno na: http://en.wikipedia.org/wiki/Optical_character_recognition
- [5] (2010) Zakon o varstvu dokumentarnega in arhivskega gradiva ter arhivih (ZVDAGA). Dostopno na: <http://www.arhiv.gov.si/fileadmin/arhiv.gov.si/pageuploads/zakonodaja/zvdaga.pdf>
- [6] (2010) Enterprise resource planning. Dostopno na: http://en.wikipedia.org/wiki/Enterprise_resource_planning
- [7] (2010) Human resource management. Dostopno na: http://en.wikipedia.org/wiki/Human_resource_management
- [8] (2010) Decision support system. Dostopno na: http://en.wikipedia.org/wiki/Decision_support_system
- [9] Brian Benz, Rocky Oliver: Lotus Notes and Domino 6 Programming Bible, 2003, poglavje 1, 18, 23, 29-32.
- [10] (2010) IBM Lotus Notes. Dostopno na: http://en.wikipedia.org/wiki/IBM_Lotus_Notes
- [11] (2010) Detailed system requirements for Lotus Notes, Lotus Domino, Lotus Domino Administrator, Lotus Domino Designer, and Lotus Notes Traveler. Dostopno na: <http://www-01.ibm.com/support/docview.wss?rs=463&uid=swg27007909>
- [12] (2005) Application Development with Lotus Domino Designer. Dostopno na: <http://www.ibm.com/developerworks/lotus/documentation/dominodesigner/70x.html>
- [13] (2010) Using OLE. Dostopno na: http://www-12.lotus.com/ldd/doc/domino_notes/Rnext/help6_designer.nsf/f4b82fbb75e942a6852566ac0037f284/d6f5f235d1b4f25985256c54004c50da?OpenDocument
- [14] (2010) History of Microsoft Office. Dostopno na: http://en.wikipedia.org/wiki/History_of_Microsoft_Office
- [15] (2010) Word Object Model. Dostopno na: <http://msdn.microsoft.com/en-us/library/kw65a0we%28VS.80%29.aspx>
- [16] (2010) Licence - pravni vidiki. Dostopno na: <http://sl.openoffice.org/licence.html>
- [17] (2010) IBM Lotus Symphony. Dostopno na: <http://symphony.lotus.com/software/lotus/symphony/home.nsf/home>
- [18] (2010) StarOffice. Dostopno na: <http://en.wikipedia.org/wiki/StarOffice>
- [19] (2010) OpenDocument. Dostopno na: <http://sl.wikipedia.org/wiki/OpenDocument>

- [20] (2010) udk: UNO Development Kit Project. Dostopno na: <http://udk.openoffice.org/>
- [21] (2010) Automation OpenOffice.org. Dostopno na:
http://udk.openoffice.org/common/man/tutorial/office_automation.html
- [22] (2000) StarOffice Programmer's Tutorial. Dostopno na:
<http://api.openoffice.org/basic/man/tutorial/tutorial.pdf>