

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Roman Ivančič

**VPELJAVA PODPORE KREDITIRANJA V DOKUMENTNI
SISTEM BANKE**

DIPLOMSKO DELO
NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: dr. Igor Rožanc

Ljubljana, 2009



Št. naloge: 00438/2009

Datum: 05.04.2009

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

Kandidat: **ROMAN IVANČIČ**

Naslov: **VPELJAVA PODPORE KREDITIRANJA V DOKUMENTNI SISTEM
BANKE**
**ENHANCING BANKING ELECTRONIC DOCUMENT MANAGEMENT
SYSTEM WITH CREDIT SUPPORT**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

Sodobni dokumentni sistemi omogočajo zbiranje, evidentiranje, razdeljevanje in uporabo elektronskih dokumentov in so kot taki ključen dejavnik uspešnega poslovanja številnih organizacij. V diplomski nalogi predstavite vpeljavo podpore procesa izdelave kreditnih map v obstoječi bančni dokumentni sistem.

V nalogi predstavite področje elektronskih dokumentnih sistemov s poudarkom na sistemu Lotus Notes. V nadaljevanju predstavite obstoječi bančni dokumentni sistem ter njegovo uporabo. Na podlagi opisa postopka potrjevanja kreditov ustrezno zasnujete in izvedite postopek nadgradnje dokumentnega sistema. Nalogo zaključite s prikazom uporabe novega dela dokumentnega sistema.

Mentor:

viš. pred. dr. Igor Rožanc



Dekan:

prof. dr. Franc Solina

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Roman Ivančič,

z vpisno številko 63010307,

sem avtor diplomskega dela z naslovom:

Vpeljava podpore kreditiranja v dokumentni sistem banke

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom (naziv, ime in priimek)
dr. Igorja Rožanca
- 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 02.07.2009

Podpis avtorja:

ZAHVALA

Iskreno se zahvaljujem mentorju na Fakulteti za računalništvo in informatiko, dr. Igorju Rožancu, ter podjetju IS d.o.o., ki mi je omogočilo izdelavo te diplomske naloge. Še posebej bi se zahvalil mag. Mitji Nikoliću za strokovno pomoč in pojasnila.

Prav tako bi se zahvalil ženi Nini in mojim domačim, ki so mi stali ob strani in me spodbujali skozi celoten študij.

KAZALO

| | |
|---|----|
| POVZETEK | 1 |
| ABSTRACT | 2 |
| 1 UVOD..... | 3 |
| 2 DOKUMENTNI SISTEMI..... | 4 |
| 2.1 Vloga dokumentnih sistemov | 4 |
| 2.2 Osnovni gradniki dokumentnih sistemov | 5 |
| 2.2.1 Podsystem za upravljanje z dokumenti | 6 |
| 2.2.2 Podsystem za računalniško skeniranje dokumentov | 7 |
| 2.2.3 Podsystem za upravljanje zapisov | 7 |
| 2.2.4 Podsystem za krmiljenje delovnih procesov | 7 |
| 2.2.5 Logični arhiv in fizični arhiv | 7 |
| 2.2.6 Sporočilni sistem ali podpora sodelovanju | 8 |
| 2.2.7 Povezovanje s transakcijskim sistemom | 8 |
| 2.3 Arhiviranje dokumentov | 8 |
| 2.4 Računalniško skeniranje dokumentov | 9 |
| 2.4.1 Zgoščevanje | 11 |
| 2.4.2 Pravna veljavnost skeniranih dokumentov | 11 |
| 2.5 Krmiljenje delovnih procesov | 11 |
| 2.6 Problematika vnosa evidenčnih podatkov | 12 |
| 3 LOTUS NOTES / DOMINO R7 | 13 |
| 3.1 Primernost Domino okolja za podporo obravnavanega procesa | 13 |
| 3.2 Odjemalci | 14 |
| 3.2.1 Lotus Notes R7 | 14 |
| 3.2.2 Domino designer R7 | 14 |
| 3.2.3 Domino Administration odjemalec | 15 |
| 3.3 Struktura zbirke dokumentov | 15 |
| 3.3.1 Repliciranje..... | 16 |
| 3.3.2 Elementi dokumentne zbirke | 17 |
| 3.3.2.1 Obrazci | 17 |

| | | |
|-----------|--|----|
| 3.3.2.2 | Pregledi..... | 21 |
| 3.3.3 | Domino imenik..... | 23 |
| 3.3.4 | Zbirka razgovorov | 24 |
| 3.4 | Uporaba programskih jezikov v Domino Notes okolju..... | 24 |
| 3.4.1 | Jezik formul | 24 |
| 3.4.2 | Skriptni jezik JavaScript..... | 26 |
| 3.4.3 | LotusScript | 26 |
| 3.4.4 | Java..... | 26 |
| 3.5 | Varnost | 27 |
| 3.5.1 | Preverjanje pristnosti in nadzor dostopa..... | 27 |
| 3.5.2 | Preverjanje pristnosti v Dominu..... | 28 |
| 3.5.3 | Nadzor dostopa v Domino okolju..... | 29 |
| 3.5.3.1 | Zaščita na nivoju strežnika..... | 29 |
| 3.5.3.2 | Zaščita na nivoju aplikacije..... | 30 |
| 3.5.3.2.1 | Skrivanje aplikacije (zbirke)..... | 30 |
| 3.5.3.2.2 | Aplikacijski ACL..... | 30 |
| 3.5.3.2.3 | Vloge..... | 32 |
| 3.5.3.3 | Zaščita na nivoju dokumenta | 32 |
| 3.5.3.4 | Zaščita na nivoju polja | 33 |
| 4 | OPIS DOKUMENTNEGA SISTEMA | 34 |
| 4.1 | Upravljanje z dokumenti | 34 |
| 4.2 | Upravljanje poslovnih procesov | 37 |
| 5 | VPELJAVA PROCESA POTRJEVANJA KREDITNIH MAP V DOKUMENTNI SISTEM V BANKI VOLKSBANK D.D. LJUBLJANA | 38 |
| 5.1 | Opis sistema potrjevanja kreditnih map | 40 |
| 5.1.1 | Kratek opis procesa potrjevanja kreditnih map | 40 |
| 5.1.2 | Podrobnejši opis sistema in kreiranja kreditov..... | 41 |
| 5.1.2.1 | Struktura/Hierarhija razredov – Objektni model..... | 42 |
| 5.1.2.2 | Ponudba..... | 43 |
| 5.1.2.3 | Kreditni predlog | 43 |
| 5.1.2.4 | Kreditna mapa | 45 |

| | | |
|---------|--|----|
| 5.1.3 | Uporaba elektronskega potrjevanja kreditnih map »v praksi« | 46 |
| 5.1.3.1 | Prijava v dokumentni sistem preko Lotus Notes odjemalca | 46 |
| 5.1.3.2 | Mapa komitenta..... | 46 |
| 5.1.3.3 | Struktura/hierarhija dokumentov..... | 47 |
| 5.1.3.4 | Statusna in bilančna dokumentacija | 49 |
| 5.1.3.5 | Zabeležke v mapi komitenta | 49 |
| 5.1.3.6 | Priprava ponudbe..... | 50 |
| 5.1.3.7 | Kreditni predlog (priprava dokumentacije komitenta)..... | 50 |
| 5.1.3.8 | Kreditna mapa | 52 |
| 6 | ZAKLJUČEK..... | 53 |
| | VIRI..... | 54 |

SEZNAM UPORABLJENIH KRATIC IN SIMBOLOV

- ACL (ang. Access Control List) - sistem nadzora dostopa oziroma lista pristopnih pravic (npr. za določen dokument)
- API (ang. Application Programming Interface) - aplikacijski programski vmesnik.
- ASCII (ang. American Standard Code for Information Interchange) - ameriški standardni nabor za izmenjavo informacij, je 7-bitni nabor znakov.
- CAL (ang. Compose Access List) - lista uporabnikov
- COLD (ang. Computer output to Laser Disc) - sistem za iskanje in pregledovanje izpisanih faktur
- CORBA (ang. Common Object Request Broker Architecture) - objektno usmerjena komunikacija med porazdeljenimi aplikacijami.
- DMS (ang. Document Management Systems); sistem za upravljanje z dokumenti
- EDMS (ang. Electronic Document Management Systems) - sistem za elektronsko upravljanje z dokumenti
- ERM (ang. Enterprise Records Management) - novejši sistem za iskanje in pregledovanje izpisanih faktur
- HSM (ang. Hierarchical Storage Management) - sistemi, ki skrbijo za optimalnoo zasedenost pomnilnih medijev.
- JVM (ang. Java Virtual Machine) - javanski navidezni stroj.
- NRPC (ang. Notes Remote Procedure Calls) - kombinacija Notes Name Service in DNS za spremembo strežniških imen v omrežne naslove.
- ODBC (ang. Open Database Conectivity) - tehnologija, ki se uporablja za premikanje podatkov iz ene vrste baze podatkov v drugo.
- ODMA (ang. Open Document Management API) - API ki poenostavlja komunikacijo namizne aplikacije z dokumentnim sistemom.
- SUPB Sistem za upravljanje podatkovnih baz
- TCP/IP (ang. Transmission Control Protocol/Interneet Protocol) - protokol nadzora prenosa in internetni protokol.
- URL (ang. Uniform Resource Locators) - naslov spletnih strani v spletu.
- VBA (ang. Visual Basic for Applications) - microsoftov programski jezik.

KAZALO SLIK

| | |
|---|----|
| Slika 1: Princip dostopa na enem mestu..... | 4 |
| Slika 2: Koncept dokumentnega sistema..... | 6 |
| Slika 3: Življenski cikel dokumenta [8]..... | 10 |
| Slika 4: Lotus Notes »Workspace« | 15 |
| Slika 5: Struktura baze dokumentov na Domino strežniku [10] | 16 |
| Slika 6: Primer replika ID..... | 17 |
| Slika 7: Primer formule za skrivanje polja..... | 19 |
| Slika 8: Primer pregleda v Domino zbirki..... | 21 |
| Slika 9: Prikaz pregledov v Lotus Domino Designer-ju s sinonimi..... | 22 |
| Slika 10: Strežniški ACL..... | 29 |
| Slika 11: Aplikacijski ACL | 30 |
| Slika 12: Nivoji dostopa | 31 |
| Slika 13: Vloge | 32 |
| Slika 14: Shema back-end razredov in funkcij..... | 42 |
| Slika 15: Shema front-end razredov in funkcij..... | 42 |
| Slika 16: Shema procesa kreditnega predloga..... | 45 |
| Slika 17: Prijava v dokumentni sistem | 46 |
| Slika 18: Mapa komitenta..... | 47 |
| Slika 19: Struktura dokumentov komitenta in dokumentov kredita..... | 48 |
| Slika 20: Nastavljanje varnostne sheme nad dokumentom | 48 |
| Slika 21: Statusna in bilančna dokumentacija | 49 |
| Slika 22: Zabeležka v mapi komitenta | 49 |
| Slika 23: Ponudba..... | 50 |
| Slika 24: Ustvarjanje novega kreditnega predloga v mapi komitenta..... | 51 |
| Slika 25: Kreditni predlog | 51 |
| Slika 26: Tabela za prikaz obstoječih kreditov..... | 51 |
| Slika 27: Povezava kreditnega predloga s kreditno mapo preko povezovalnega dokumenta..... | 52 |
| Slika 28: Kreiranje nakazila v kreditni mapi | 52 |

POVZETEK

Diplomska naloga je namenjena predstavitvi nadgradnje delujočega dokumentnega sistema, natančneje vpeljavi potrjevanja kreditov v obstoječi dokumentni sistem banke. Namen tega sistema je omogočiti Banki Volksbank d.d. Ljubljana elektronsko potrjevanje kreditov in s tem tudi elektronsko arhiviranje pripadajoče dokumentacije.

V nalogi je najprej predstavljen dokumentni sistem kot ključni del pri sodobnih poslovnih informacijskih sistemov. Dokumentni sistemi, ki se danes razvijajo z bliskovito hitrostjo, namreč omogočajo več kot zgolj varno hranjenje podatkov. Zahteve so danes veliko večje, saj današnji dokumentni sistemi nadgrajujejo celoten poslovni proces v podjetju ali družbi. To pa daje družbi prednost pred konkurenco in povečuje njeno učinkovitost.

Postopek potrjevanja kreditov v banki pred tem ni bil elektronsko podprt, kar je zaradi obilice dokumentov povzročalo nepreglednost, slab nadzor in s tem tudi počasnost kreditnega poslovanja družbe. To je privedlo do zahteve po izdelavi programske rešitve »Potrjevanje kreditov v bančnem dokumentnem sistemu«, ki je tema mojega diplomskega dela.

Obstoječi dokumentni sistem, ki se že uporablja v delovnem okolju družbe, smo tako razširili še na proces potrjevanja kreditov. Uporabili smo orodje Lotus Notes/Domino, ki je primerno za obvladovanje poslovnih procesov in dokumentacije. Deluje na principu odjemalec – strežnik, ki omogoča dostop do informacij (dokumentov), njihovo razvrščanje ter upravljanje v omrežju. Dostop do dokumentov je zelo enostaven in se izvaja preko pregledov ali vpetih pregledov na obrazcu. S kategoriziranim prikazom in dostopom do dokumentov kredita iz enega mesta smo olajšali delo bančnim uslužbencem, saj bi drugače dokumentacija potovala iz oddelka na oddelek.

Cilj naloge je bil dosežen: razvit je bil delujoč del dokumentnega sistema, ki ustreza vsem zahtevam. Kredit je mogoče spremljati skozi celoten proces, torej od poslance ponudbe do nakazila oziroma izplačila zneska. Tako je zagotovljena varnost, sledljivost in sprotno obveščanje odgovornih oseb o stanju kreditnih dokumentov.

Ključne besede:

dokumentni sistem, elektronsko potrjevanje, elektronsko arhiviranje, Lotus Notes, Lotus Domino

ABSTRACT

The purpose of this diploma work is to present upgrading of an operating document system, more precisely introducing credit confirmation into the existent banking document management system. The intention of this system is to enable Banka Volksbank d.d. Ljubljana to use electronic credit confirmation as well as electronic archiving of the related documentation.

First, a document system as a key part of up-to-date business information systems is described. Document systems, which nowadays develop extremely rapidly, enable more than merely secure data saving. Today requirements are much higher, since current document systems upgrade the entire business process of a company, which consequently puts the company ahead of competition and increases its effectiveness.

In the past credit confirmation procedure at the bank was not electronically supported, which due to the abundance of documents caused lack of clarity, bad control and hence, slow credit operating of the company. This led to the requirement for creation of a program solution "Credit confirmation in banking document management system," which is the topic of my diploma work.

The existing document system, being already used in the working environment of the company, has been expanded to the process of credit confirmation. The tool Lotus Notes/Domino, suitable for business process and documentation management, has been used. It works on the principle client – server, which enables access to information (documents), its classification and management in the network. The access to documents is very simple and is performed via views or embedded views in the form. By categorized display and access to credit documents from one site, the work of bank clerks has been facilitated, as otherwise the documentation would move from one department to another.

The aim of this paper has been achieved: an operating part of a document system that meets all the requirements has been developed. A credit can be monitored through the entire process, from sending an offer to transferring the money. In this way, safety, tracking and prompt informing of accountable staff about the state of credit documents is ensured.

Key words:

document system, electronic confirmation, electronic archiving, Lotus Notes, Lotus Domino

1 UVOD

Družbe, ki želijo v času hitre rasti podatkov in informacij na vseh področjih, predvsem poslovnem, preživeti in se razvijati, morajo nenehno izboljševati svoje poslovanje, organizacijo in način upravljanja družbe. V času elektronskega poslovanja se zmeraj bolj pojavlja zahteva po splošnem dostopu do dokumentov in skupinskem načinu dela z dokumenti. Vsak dan se v številnih delih družbe zapiše ogromno sporočil, ki jih je potrebno zajeti, shraniti in narediti dostopne uporabnikom. S temi zahtevami pridemo do potrebe po vpeljavi dokumentnega sistema.

Dokumentni sistem je ključni del prenove poslovnih informacijskih sistemov. Osnova sistema je dokument v elektronski in ne v papirnati obliki. Obstoječi dokumentni sistem »3K Document Cycle«, ki ga je razvilo podjetje 3K-IT d.o.o., nam vse to omogoča. Razvit je bil v Domino okolju, ki je narejen za izdelavo dokumentnih sistemov. Omogoča nam enostavno izdelavo organizacijske strukture družbe (v našem primeru Banke Volksbank d.d. Ljubljana), kamor vpišemo vse zaposlene osebe združbe v pravilno hierarhično strukturo. Vsaka zaposlena oseba spada v določeno enoto, ki je sestavljena iz vodje in članov enote. Hierarhija organizacije družbe je zelo pomembna, saj jo uporabljamo pri zaščiti (določanju varnostne sheme) na nivoju aplikacije in na nivoju dokumenta. Imamo tudi možnost določanja zaščite nad samim delovnim procesom. Omogoča nam gradnjo strukture dokumentov, ki nam določa strukturirane podatke in poslovno logiko dokumenta (slika 17). Razvijalec se opira na hierarhijo dokumentov skozi celoten razvoj, saj lahko za vsak dokument posebej definiramo svoj delovni proces in svojo varnostno shemo na nivoju dokumenta. Delovni proces vsebuje več povezljivih faz (statusov), v katerih se izvajajo določene akcije. Sami dokumenti se med delovnim procesom spreminjajo, zato je tu omogočena kontrola pretoka dokumentov.

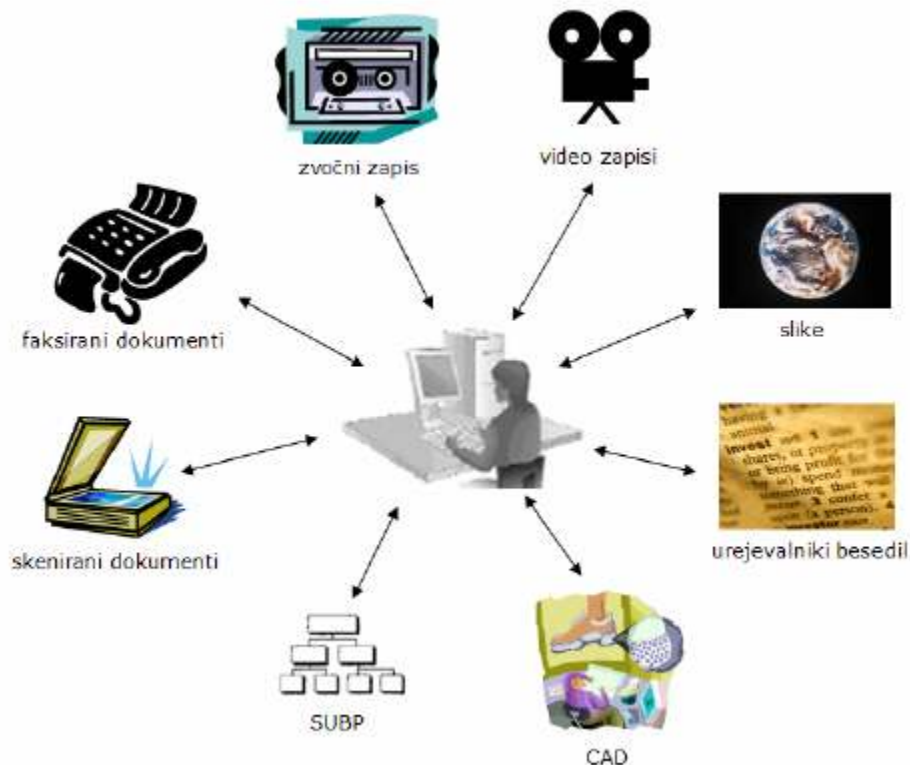
Z nekaterimi že vpeljanimi delovnimi procesi v obstoječem dokumentnem sistemu se je v Banki Volksbank d.d. Ljubljana pojavila zahteva po vpeljavi elektronskega potrjevanja kreditnih map. Dejstvo, da postopek do sedaj ni bil elektronsko podprt, je botrovalo temu, da je potrjevanje kreditnih map potekalo brez večjega nadzora in precej počasneje kot bi lahko. Glavne prednosti, ki jih je naročnik pričakoval z vpeljavo elektronskega potrjevanja kreditnih map, so bile sledljivost dokumentov, večji nadzor nad potekom delovnega procesa in elektronski arhiv. Vse to nam je tudi uspelo zagotoviti. Za razvoj aplikacije smo uporabili odjemalec Lotus Domino Designer, ki omogoča izdelavo aplikacij za Domino strežnik.

2 DOKUMENTNI SISTEMI

Sistemi za elektronsko upravljanje dokumentov ali dokumentni sistemi (ang. Electronic Document Management Systems - EDMS) so danes ključni del prenove poslovnih informacijskih sistemov. Hiter razvoj tako strojne kot programske opreme je omogočil ekspanzijo dokumentnih sistemov. Omogočajo zbiranje, sestavljanje, evidentiranje, elektronsko distribucijo, indeksiranje za potrebe iskanja, arhiviranje in hitro iskanje informacij. Osnova sistema je dokument v elektronski in ne v papirnati obliki.

2.1 Vloga dokumentnih sistemov

Razvoj koncepta informacijskih sistemov lahko predstavimo v treh obdobjih. Prvo označuje Von Neumann-ov model procesiranja. Gre za linearni proces vnosa podatkov, ki so rezultat določenega procesa. Nato je prišlo obdobje relacijskih podatkovnih baz in gostiteljskih (ang. Host) sistemov. Gre za tradicionalne sisteme za upravljanje baz podatkov (SUPB). Danes koncept centralno shranjenih in zgolj relacijsko povezanih podatkov izginja, saj uporabnika ne zanima, od kod prihaja informacija. Pomembno je, da je ta na voljo ne glede na obliko (glej sliko 1): podatkovna baza, faksirani dokument, skenirani dokument, video zapis, datoteka urejevalnika besedil, slika, zvočni zapis ali kaj drugega [6].



Slika 1: Princip dostopa na enem mestu

Uporabnik naj bi imel dostop do dokumentov na enem mestu, ne glede na izvor ali obliko dokumenta. Tak koncept imenujemo koncept izgubljene škatle (ang. the missing box) in je osnova sodobnih dokumentnih sistemov. Za razumevanje DMS se je potrebno znebiti razmišljanja o podatku kot polju v neki podatkovni bazi. Na podatke je treba gledati preko nove osnove – dokumenta. Sistemi za upravljanje podatkovnih baz (SUPB) služijo kot spodnji nivo dokumentnega sistema, kot zbirka referenčnih podatkov za opis dokumenta, torej kot zbirka ključnih podatkov za iskanje dokumenta. Usmerjenost k sistemom, kjer je v središču uporabnik in kjer je osnova dokument, je pripeljala do razvoja tehnologij za krmiljenje poslovnih procesov (ang. workflow) in podpora skupinskemu delu (ang. groupware, collaboration). Prav tako je bistvena razlika, da se informacija filtrira ob poizvedbi in ne ob nastanku, kot je običajno pri sistemih s strukturiranimi informacijami [4].

Če zadevo posplošimo, pridemo do spoznanja, da nam je na voljo celo preveč podatkov in informacij. Pridemo do problema, ko so nam informacije na voljo, potrebno jih je le primerno izluščiti in uporabiti. Iskanje informacij lahko opišemo prek treh različnih pristopov [4]:

- **Iskanje po ključnih besedah in podatkih:** Ključne besede se vnesejo ob nastanku oziroma vstopu dokumenta v sistem in se nahajajo v tradicionalnih podatkovnih bazah. Pod ključnimi podatki pojmuje avtorja, datum ustvarjanja in morda še nekaj podobnih atributov, ki so odvisni od profila dokumenta. Tu igra pomembno vlogo tehnologija prepoznavanja obrazcev.
- **Iskanje po polnem besedilu (ang. Full text search):** Vsebina dokumenta se nahaja v vzporedni podatkovni strukturi in služi kot indeks originalnemu dokumentu. Tu pridejo do izraza sistemi optičnega prepoznavanja znakov znotraj sistemov za elektronsko upodabljanje oziroma skeniranje dokumentov.
- **Princip združevanja (ang. Clustering):** Dokumenti se združujejo v skupine po podobnosti. Poizvedbe vrnejo množico dokumentov, ki ustrezajo iskalnim pogojem, kar zahteva primerno organizacijo dokumentov.

Izbira primernega orodja za iskanje informacij v sistemu je ključnega pomena. Če informacij ne moremo poiskati, potem je morda bolje, da sistema sploh ni.

2.2 Osnovni gradniki dokumentnih sistemov

Osnovne gradnike dokumentnega sistema najboljše ponazarja slika 2. Koncept dokumentnega sistema sestavljajo naslednje komponente [9]:

- podsistem za upravljanje z dokumenti,
- podsistem za krmiljenje delovnih procesov (ang Workflow),
- podsistem za računalniško upodabljanje oziroma skeniranje dokumentov (ang. Imaging),
- podsistem za upravljanje zapisov,
- logični arhiv in
- fizični arhiv.



Slika 2: Koncept dokumentnega sistema

Komponente medsebojno povežemo s funkcijami, ki jih nudi sporočilni sistem, del sistema v smislu izmenjave podatkov je tudi transakcijski sistem. Seveda v posameznih primerih implementacije niso vedno zajeti vsi gradniki. Brez sistema za računalniško upodabljanje si dokumentnega sistema z vhodnimi dokumenti v papirni obliki ni mogoče zamisliti, saj bi v tem primeru imeli le evidenčni sistem, torej hranjenje informacije o obstoju dokumenta, ne pa tudi njegove vsebine. Na tak način izgubimo osnovni namen iskanja informacije ne glede na njen izvor.

2.2.1 Podsystem za upravljanje z dokumenti

V okviru podsistema za upravljanje dokumentov zagotavljamo naslednjo funkcionalnost:

- enoten vhod in izhod za dokumente (enotni postopki evidentiranja za vse vhodne in vse izhodne dokumente),
- indeksiranje dokumentov,
- povezovanje dokumentov,
- pripravo izhodnih dokumentov (predloge, rezervacijo¹, vračanje, verzije) in
- iskanje dokumentov

¹ Rezervacija in vračanje dokumentov (ang. check-out in check-in) je pomembna lastnost dokumentnih sistemov, ki omogoča kontrolo nad konflikti spreminjanja in shranjevanja dokumentov. V primeru rezervacije se dokument v sistemu zaklene in je na voljo le za branje. Spreminjamo ga lahko šele, ko ga uporabnik, ki ga je rezerviral, vrne v sistem.

V okviru tega sistema zagotavljamo tudi varnost (avtentikacija in avtorizacija) ter podporo postopkom, ki so povezani z morebitnim digitalnim podpisovanjem dokumentov.

2.2.2 Podsystem za računalniško skeniranje dokumentov

V okviru podsistema za računalniško skeniranje dokumentov (ang. imaging) izvajamo zajem dokumentov v elektronski obliki ter tudi zajem elektronskih dokumentov (npr. faks). Podrobneje bom opisal v poglavju 2.4.

2.2.3 Podsystem za upravljanje zapisov

Pri množičnih izpisih (računi, obračuni, itd.), lahko vsebino, ki jo pošiljamo na tiskalnik, obenem pošljemo tudi v sistem za upravljanje zapisov COLD² ali novejši ERM³. V COLD sistemih shranjujemo ASCII⁴ zapis dokumenta, indeksne podatke in podatke o obrazcu za izpis. Dokumenta ne hranimo v celoti, temveč ob zahtevi dokument sestavimo in prikažemo na zaslonu. COLD torej ni namenjen arhiviranju.

2.2.4 Podsystem za krmiljenje delovnih procesov

V okviru sistema za krmiljenje delovnih procesov lahko definiramo, pripravimo in izvajamo krmiljenje delovnih procesov ter nadziramo njihovo izvajanje. Več sem opisal v poglavju 2.5.

2.2.5 Logični arhiv in fizični arhiv

V logičnem arhivu hranimo evidenčne podatke o dokumentih, fizično pa se posamezni objekti (datoteke) nahajajo v fizičnem arhivu. V logičnem delu se nahajajo le kazalci na objekte v fizičnem arhivu. Fizični arhiv skrbi tudi za migracijo starejših objektov ter zavaruje dokumente pred nepooblaščenim dostopom. Fizični arhiv ni le arhiv dokumentov, temveč tudi aplikativna rešitev, ki zagotavlja varnost in sledljivost dokumentov. Več o arhiviranju dokumentov je opisano v poglavju 2.3.

² COLD = Computer Output to Laser Disc

³ ERM = Enterprise Records Management

⁴ ASCII = American Standard Code for Information Interchange

2.2.6 Sporočilni sistem ali podpora sodelovanju

S pomočjo sporočilnega sistema znotraj dokumentnih sistemov uporabnikom omogočamo pomembne funkcionalnosti, kot npr.:

- obveščanje o nastanku ali spremembe dokumenta v sistemu,
- obveščanje o spremembi faze v delovnem procesu oziroma delovnem toku in morebitni potrebni akciji,
- nastavitve alarmov za posamezna opravila,
- ustvarjanje opravil v elektronskem koledarju ter
- enostavna izmenjava sporočil, kjer je še posebej potrebno poudariti uporabnost tehnologij sodelovanja v realnem času (ang. Instant Messaging).

2.2.7 Povezovanje s transakcijskim sistemom

Vseh podatkov ne moremo hraniti v dokumentnem sistemu, saj je namenjen zgolj delu z dokumenti. Da se izognemo dvojnemu vnosu ter dvojnimi zapisom podatkov, mora biti dokumentni sistem povezan s transakcijskim sistemom, pri čemer je lahko povezava zelo enostavna (dostop do šifrantov) ali pa kompleksnejša (pisanje novih zapisov v transakcijske sisteme, spremembe podatkov v enem ali drugem sistemu, vpliv na delovni tok v dokumentnem sistemu, itd.). Pomembno je, da ločimo namembnost sistemov. Znotraj transakcijskega sistema ne moremo reševati problema dokumentov, dokumentni sistem pa ni namenjen informatizaciji kompleksnih podatkovnih relacij in transakcij.

2.3 Arhiviranje dokumentov

Pri velikem številu dokumentov, s katerimi se srečujemo z dneva v dan, igra najpomembnejšo vlogo sistem za shranjevanje oziroma arhiviranje dokumentov, saj je dokumentom nemogoče slediti ročno ali preko datotečnega sistema. Tovrstno početje je vprašljivo tudi z vidika varnosti podatkov, ne le za shranjevanje skeniranih dokumentov, temveč vseh dokumentov v sistemu za elektronsko upravljanje z dokumenti. Zato je pred načrtovanjem sistema za arhiviranje potrebna podrobna analiza predvidene količine dokumentov v določenem časovnem intervalu.

Zaradi že omenjenih večjih potreb po diskovnih kapacitetah, so že davno pred pojavom osebnih računalnikov razvili in uporabljali posebne programske produkte, ki so omogočali hierarhično arhiviranje dokumentov. To so HSM⁵ sistemi, ki skrbijo za optimalno zasedenost pomnilnih medijev in za čim ustrežnejšo razpoložljivost datotek. Večnivojski sistem arhiviranja lahko organiziramo tako, da določimo hierarhični vrstni red pomnilniških medijev za posamezne

⁵ HSM = Hierarchical Storage Management

skupine dokumentov (izbrane glede na profil). Na osnovi tega potem HSM sistem razvršča, arhivira in prenaša dokumente glede na stopnjo uporabe oziroma glede na frekvenco dostopa. Pri tem je najpomembnejše, da uporabnikom oziroma nadzornikom sistema ni potrebno skrbeti na katerem mediju se posamezni dokument nahaja. HSM jih sam predstavlja po nivojih (trdi disk, zunanje enote). S tem je onemogočeno pogosto izgubljanje in nepotrebno kopiranje papirnih dokumentov, ko isti dokument hkrati potrebuje več uporabnikov. Z uporabo različnih medijev zgradimo elektronski arhiv, v katerem dokumenti kontrolirano prehajajo z medija na medij. Prehajajo na osnovi predhodno definiranih pravil oziroma postopkov znotraj profilnih dokumentov [7].

2.4 Računalniško skeniranje dokumentov

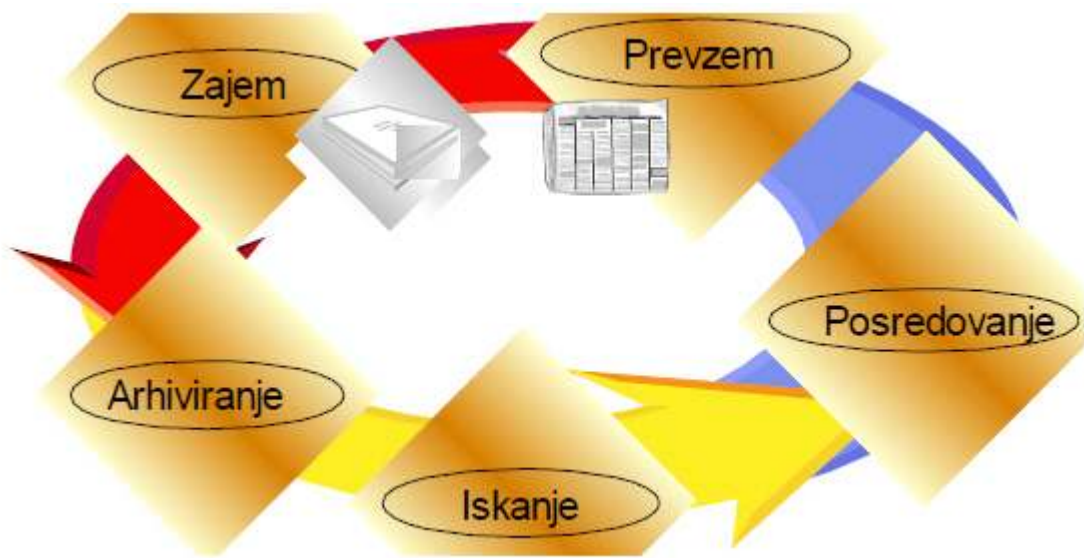
Sistemi za računalniško skeniranje dokumentov predstavljajo vhodni del sistemov za upravljanje z dokumenti, kjer je izvorni dokument v papirni obliki. Brez učinkovitega sistema zajema in arhiviranja papirnih dokumentov imamo opravka zgolj z nekakšnim evidenčnim sistemom, ki omogoča evidenco papirnih dokumentov. Skeniranje dokumentov je torej nadgradnja za dokumentne (zgolj evidenčne) sisteme, ki slonijo na papirni obliki dokumenta.

Življenski cikel skeniranega dokumenta je proces ponavljajočih se faz in aktivnosti (glej sliko 3):

- zajem in vnos podatkov v sistem za upravljanje z dokumenti,
- indeksiranje podatkov,
- arhiviranje,
- procesiranje podatkov iz dokumentov,
- distribuiranje in
- iskanje, pregledovanje ali popravljanje

Na osnovi teh ugotovitev lahko predvidimo temeljne funkcije, ki jih mora imeti ustrezna programska oprema [1]:

- kontrola, nadzor delovanja čitalca (skenerja),
- kontrola kakovosti oziroma vidnosti dokumenta (osvetljenost, kontrast, povečava),
- izboljšanje vidljivosti (čiščenje ozadja),
- vpis dodatnih opomb in komentarjev (kot sestavni del skeniranega dokumenta),
- optično prepoznavanje znakov in priprava podatkov za iskanje po polnem besedilu (ang. Full text search) in
- pregledovanje in arhiviranje dokumentov



Slika 3: Življenski cikel dokumenta [8]

Pri skeniranju znotraj dokumentnih sistemov ločimo različne načine skeniranja. Ti so odvisni predvsem od izvora, količine in oblike dokumentov:

- **Posamično skeniranje:** ob vpisu evidenčnih podatkov in določanju profila dokumentov
- **Skeniranje v paketu (ang. Batch scanning):** več dokumentov položimo na podajalec. Sistem vsakič, ko poskenira dokument, ponudi uporabniku pogovorno okno za vpis indeksnih podatkov. Pri tem obstajajo različni načini določanja ločilca (ang. separator), kdaj se konča en in kdaj prične drug dokument. Ločilec je lahko prazna stran, črtna koda (zelo priporočljiv in najzaneslivejši pristop) ali pa enostavno skeniramo vse dokumente z v naprej določenim številom strani. V določenih primerih okno za vpis indeksnih podatkov niti ni potrebno, če s pomočjo črtna kode lahko pridobimo ustrezen podatek.
- **Skeniranje v paketu s tehnologijo prepoznavanja obrazcev:** Je v bistvu skeniranje v paketu, vendar uporabnik ne vpisuje podatkov, pač pa jih skuša sistem prepoznati sam. Ta način je uporaben, če skeniramo enake, v naprej pripravljene obrazce.
- **Skeniranje kot zunanja storitev (ang. Outsourcing):** Skeniranje izvaja specializirano podjetje, ki prevzame dokumente ter vrne medij s skeniranimi objekti ter opisi atributov. Cena storitve je določena na dokument in je odvisna predvsem od števila dokumentov. Tovrstna storitev je velikokrat primerna ob vpeljavi dokumentnega sistema, ko zagotovimo sprotno skeniranje aktualnih dokumentov, medtem ko skeniranje dokumentov iz preteklega obdobja prepustimo zunanjemu izvajalcu.

S skeniranjem pridobimo dokument v elektronskem grafičnem formatu in ga lahko opremimo z dodatnimi podatki, predvsem v obliki anotacij in obarvanjem različnih delov dokumenta. Tu se že pojavlja kot dodatek modul optičnega prepoznavanja znakov, ki lahko razpozna cel dokument ali le posamezen del. Poleg funkcij skeniranja in prepoznavanja, omenimo še možnost shranjevanja

v različnih slikovnih formatih in podpora različnim algoritmom zgoščevanja (ang. compression) slikovne datoteke.

2.4.1 Zgoščevanje

Arhiviranje skeniranih dokumentov zahteva veliko pomnilniškega prostora, zato je uporaba primernih algoritmov zgoščevanja nujno potrebna. V preteklosti se je zgoščevanje izvajalo s pomočjo posebnih strojnih vmesnikov, zdaj pa so na voljo učinkovite programske rešitve. Tehnike in algoritmi zmanjšajo velikost datoteke z odstranitvijo bitov, ki predstavljajo neuporabljeni bel prostor na dokumentu ali večje črne površine.

2.4.2 Pravna veljavnost skeniranih dokumentov

Gre za lastno laično mnenje glede pravne veljavnosti skeniranih dokumentov, saj nikjer nismo zasledili uradnega pravnega mnenja na to temo. Pravna veljavnost dokumenta je pomembna le tedaj, ko je dokument ali podatek iz dokumenta sporen. V tem primeru se glede na trenutno zakonodajo ne upošteva skeniran dokument, ki je v originalu v papirni obliki. Če tudi je elektronsko podpisan, to nikoli ne bo pomenilo, da je enak originalu, saj je nastal v drugi obliki in je bil lahko pred, med ali po elektronski pretvorbi spremenjen. Papirna dokumentacija se mora v Sloveniji hraniti v papirni obliki, če želi služiti kot dokaz v sporu. Eno od podjetij, ki mu je naša firma uvedla dokumentni sistem, je zahtevalo pravno mnenje glede pravne veljavnosti skeniranih dokumentov. Po uspešni implementaciji sistema so želeli skenirati celotno poslovno dokumentacijo za nazaj in papirni arhiv uničiti. Izkazalo se je, da morajo zakonsko zahtevane dokumente za primere sporov hraniti v papirni obliki.

2.5 Krmiljenje delovnih procesov

Krmiljenje delovnih procesov je eden ključnih delov sistema za elektronsko upravljanje dokumentov. Primerjamo ga lahko s konceptom avtomatizacije proizvodnje s tekočim trakom. Omogoča avtomatizacijo informacijskih nalog in aktivnosti z definicijo pravil, vlog in usmerjanj (ang. rules, roles & routing). Je proces, ki povezuje ljudi in postopke v poslovnem procesu, da bolj učinkovito izvršujejo naloge in si med seboj lažje izmenjujejo informacije. Osnovna predpostavka je sprejeti pisarniško okolje kot proizvajalca dokumentov skozi več povezljivih faz. V tej arhitekturi nastopa aplikacija kot povezovalc teh nalog in aktivnosti, ki nepotrebne izloči, ostale pa avtomatizira. Za definiranje delovnih postopkov je potrebna predhodna analiza sistema in temu primerno kasnejše načrtovanje [4].

Dejavnosti se znotraj postopka lahko izvajajo zaporedno, vzporedno ali pogojno (dinamično). Vsak del postopka ima predvidenega izvajalca, ki je lahko človek, skupina ljudi ali pa sam informacijski sistem. Vključen mora biti tudi sistem obveščanja, ki se aktivira na prehodu iz ene v drugo fazo, ob spremembi dokumenta ali pa ob preteku predvidenega časa za določeno dejavnost. Ravno zaradi tega je nujna povezava s sporočilnimi sistemi.

Sami dokumenti se med delovnim postopkom lahko spreminjajo, zato je prav tako pomembna sledljivost dokumentom (ang. access logging) in kontrola različic (ang. version control). Aplikacija mora omogočati varnost dostopa do dokumenta (ang. access control), ki je lahko odvisna tudi od trenutne faze procesa. V povezavi z nadzorom dokumenta je poleg sledljivosti in kontrole različic potrebno omeniti še mehanizme kot so vrnitev dokumenta v sistem (ang. check-in) ter rezervacija dokumenta v sistemu (ang. check-out).

V sistemu za upravljanje z dokumenti ima krmiljenje delovnih procesov pomembno vlogo, saj omogoča kontrolo pretoka dokumentov, ki podpirajo poslovne procese.

2.6 Problematika vnosa evidenčnih podatkov

Ob skeniranju in vnosu dokumenta v dokumentni sistem je potrebno dokument do neke mere opisati oziroma mu pripeti evidenčne podatke in ključne besede, predvsem zaradi kasnejšega iskanja dokumenta. V večini primerov to opravijo operaterji. Delo je zamudno in predstavlja ozko grlo v delovnem toku dokumenta. Dejansko je pomembno optimizirati proces zajema dokumenta na način, da delovno mesto s skenerjem vnaša čim manj evidenčnih podatkov. Bolje je, da se vstavlja v procesu na drugih delovnih mestih.

Včasih sama narava delovnega procesa omogoča avtomatsko vstavljanje podatkov, preko vmeščanja dokumenta iz faks strežnika v dokumentni sistem glede na številko pošiljatelja. Prav tako današnje programske rešitve skeniranja s pomočjo dodatnih orodij za obdelavo dokumenta (ang. Image Processing – IP) lahko iščejo in preberejo vrednosti črtnih kod na dokumentu v fazi skeniranja. Vrednost črtne kode lahko predstavlja ključ za umestitev dokumenta v sistem in pridobitev nekaterih evidenčnih podatkov.

3 LOTUS NOTES / DOMINO R7

Lotus Domino (v nadaljevanju Domino) je sistem z dvonivojsko arhitekturo tipa odjemalec strežnik. Oba, strežniški in odjemalski del lahko uporabljamo na različnih operacijskih sistemih. Odjemalec na vseh platformah uporablja enak uporabniški vmesnik.

Domino je pisan na kožo projektom, kjer potrebujemo hiter razvoj, predstavitev dinamične vsebine, nestrukturirane podatke, delovne tokove in skupinsko delo, črpanje podatkov iz različnih in nezdružljivih virov, dostop do podatkov brez povezave z omrežjem ter prilagodljivost ciljnim odjemalcem. Neprimeren je v sistemih, kjer pričakujemo visoke obremenitve v transakcijskih sistemih.

Domino shranjuje informacije v dokumentne baze, ki vsebujejo objekte, imenovane dokumenti in elemente za načrtovanje (ang. design elements). Dokument je objekt, ki vsebuje grafiko, tekst, video, avdio ali ostale vrste tekstovno bogatih podatkov. Uporabniki dostopajo do informacij preko aplikacij, ki obsegajo eno ali več dokumentnih baz.

3.1 Primernost Domino okolja za podporo obravnavanega procesa

Prednosti:

- Domino z lahkoto obvladuje večjo količino besedil in drugih mešanih podatkovnih tipov.
- Domino dokumentna baza se nahaja na Domino strežniku in ne na uporabnikovem računalniku. Zato je vsaka informacija, ki jo vnese uporabnik, takoj na voljo drugim uporabnikom v omrežju.
- Uporabniki lahko izmenjujejo dokumente po elektronski pošti ali preko Domino dokumentne baze.

Slabosti:

- Domino sicer omogoča izvajanje matematičnih operacij, vendar ne vsebuje številnih finančnih in statističnih funkcij za finančno modeliranje.
- Domino ne zaklepa dokumentov v bralnem ali popravljalnem načinu, zato transakcijske obdelave za Domino niso primerne ali pa jih lahko izvajamo le v omejenem obsegu.
- Domino ne omogoča takojšnje izmenjave podatkov z lokacijsko razpršenimi enotami, ki uporabljajo ločene Domino strežnike.

3.2 Odjemalci

Čeprav je Domino z različnimi verzijami strežniške programske opreme prikazal možnost različnih odjemalcev, ponuja v svojem sklopu tudi več svojih odjemalcev. Omenili bomo tri najpomembnejše:

3.2.1 Lotus Notes R7

Lotus Notes (v nadaljevanju Notes) ni samo odjemalec Domino strežniških storitev, ampak tudi odjemalec drugih internetnih strežnikov. Je polno opremljen »Fat client«⁶, ki omogoča delo z aplikacijami in nudi polno varnost dostopa prek NRPC⁷ in Domino certifikatov. Notes je grafični vmesnik odjemalca Domino strežnika (glej sliko 4). V samem Notesu so klici funkcij urejeni tako, da se lahko izvedejo z večih mest. Programska oprema je tipična Windows aplikacija, saj ima vse razpoznavne znake teh aplikacij. Tako ima menu vrstico in ikone za proženje določenih funkcij. Z Notes kot odjemalcem lahko uporabljamo določene funkcije kot so:

- možnost neposrednega dostopa do Domino uporabniških rešitev,
- možnost uporabe Domina kot poštnega strežnika,
- možnost uporabe Domina za časovno načrtovanje dela skupin (ang. group scheduling),
- možnost uskladitve baz podatkov na strežniku s tistimi na osebem računalniku in obratno (ang. replication),
- večja raven varnosti, vključno s kodiranjem vsebine na ravni polja,
- iskanje (ang. full-text search) po vseh listinah, shranjenih v indeksirani bazi podatkov,
- povezovanje z drugimi zbirkami podatkov, kot so na primer uvoz/izvoz podatkov in ODBC⁸,
- povezava z mrežnim okoljem, ki ni zasnovano na standardu TCP/IP⁹

3.2.2 Domino designer R7

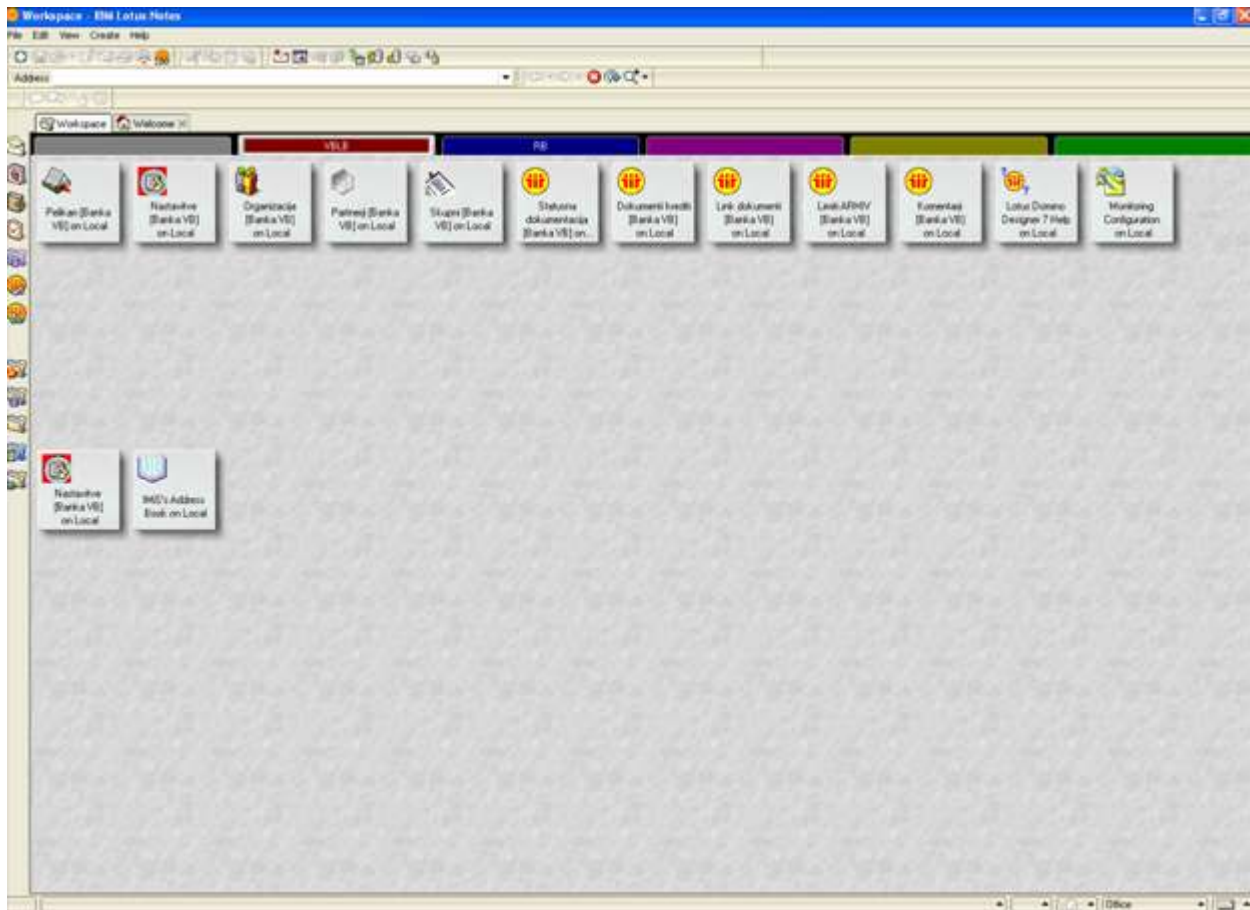
Domino Designer je integrirano aplikacijsko razvojno okolje, ki omogoča razvijalcem in oblikovalcem spletnih strani načrtovanje, upravljanje in uvajanje varnih, interaktivnih aplikacij za Domino strežnik.

⁶ Fat client = Odjemalec v omrežni arhitekturi odjemalec-strežnik, ki ima tipično bogato opremljeno funkcionalnost, ki ni odvisna od glavnega strežnika.

⁷ NRPC = Notes Remote Procedure Calls (kombinacija Notes Name Service in DNS za spremembo strežniških imen v omrežne naslove)

⁸ ODBC = Open Database Conectivity

⁹ TCP/IP = Transmission Control Protocol/Internet Protocol (protokol nadzora prenosa)



Slika 4: Lotus Notes »Workspace«

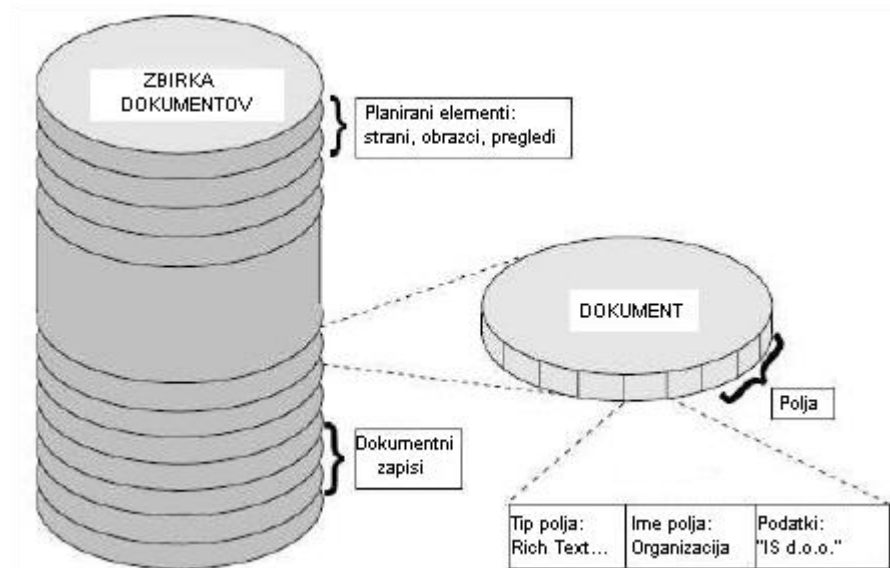
3.2.3 Domino Administration odjemalec

Je poseben odjemalec, ki omogoča polno administriranje celotnega Domino sistema. Čeprav Domino ponuja tudi Web Administration segment, le Domino administrator omogoča polni nadzor celotnega Domino sistema. Domino administrator omogoča enostavno administriranje in funkcionalnost najbolj pogostih administratorjevih nalog kot so kreiranje, brisanje, premikanje uporabnikov, upravljanje z okoljem in drugega z enega mesta.

3.3 Struktura zbirke dokumentov

Domino baza (glej sliko 5) je skupek medsebojno povezanih dokumentov in oblikovalskih orodij shranjenih v eni datoteki imenovani Notes Storage Facility (NSF). Velika posebnost te baze je v njeni nestrukturiranosti (privzeto je, da ne vsebuje nikakršne strukture). Po svoje je zelo podobna datotečnemu sistemu, saj tudi ta nima prednastavljene strukture. Na datotečnem sistemu je viden kot običajna datoteka, ki jo lahko normalno kopiramo ali zapisujemo na CD oziroma druge

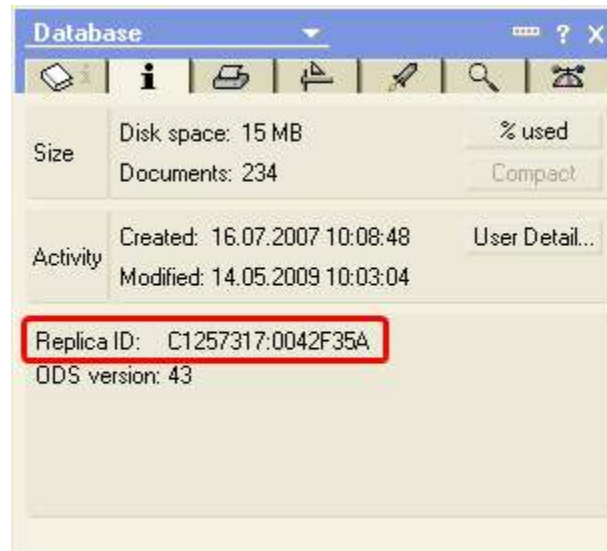
optične medije. Posebna vrsta baze so datoteke tipa .NTF(Notes template File), ki predstavljajo predlogo (ang. template) za izdelavo nove baze. Vsaka generacija Domino sistema prinese še eno novost, to je način zapisa baze na disk. Imenuje se ODS (On Disk Structure) in je v primeru R7 enak kot pri prejšnji verziji R6, in sicer 43:0. To v praksi pomeni, da so baze vsaj s tehničnega vidika 100-odstotno prenosljive med obema verzijama Domino sistemov. To še ne zagotavlja, da bo ta baza delovala, zato je treba vse aplikacije pred nadgradnjo strežnika stestirati v testnem okolju.



Slika 5: Struktura baze dokumentov na Domino strežniku [10]

3.3.1 Repliciranje

Replikacija je proces izmenjave sprememb med posebnimi kopijami dokumentnih baz. Te posebne kopije baz imenujemo replike. Ker se dokumentne baze med seboj na različnih strežnikih lahko ločijo po datotečnem imenu, Domino opredeljuje posebno oznako, na podlagi katere določa, katere baze se med seboj lahko replicirajo. Ta oznaka se imenuje replika ID in je unikatna številka, ki določa Domino bazo in vse njene replike. Ker si replike delijo isto ID-repliko, si lahko izmenjujejo informacije prek replikacije ne glede na to, kakšno je dejansko ime datoteke. Repliko ID najdemo v oknu lastnosti baze, ki je prikazana na sliki 6.



Slika 6: Primer replika ID

Razlogi za izdelavo replike [5]:

- izboljšanje strežniških storitev ob veliki uporabi določene dokumentne zbirke,
- razporeditev omrežnega prometa,
- zagotovitev razpoložljivosti zbirke ob padcu enega strežnika,
- zagotovitev razpoložljivosti zbirke oddaljenim uporabnikom na oddaljenih lokacijah, kjer bi bil dostop do centralne zbirke časovno potraten zaradi slabih omrežnih povezav ter
- izdelava delne replike za delovno skupino zaradi narave in varnosti dela.

3.3.2 Elementi dokumentne zbirke

Dokumenti vsebujejo podatke kot so tekst, grafika in podobno. Ko sestavljamo dokument, se izbrani obrazec prikaže na ekranu. Pri tem si lahko polja predstavljamo kot luknje, skozi katera vpisujemo podatke v dokument, ki leži pod obrazcem. Podatki se shranijo v dokument, medtem ko se obrazec ne shrani skupaj z dokumentom (razen če tega posebej ne zahtevamo), saj bi tako zasedli preveč prostora.

Programski elementi oziroma elementi za načrtovanje so naslednji:

3.3.2.1 Obrazci

V Dominu je obrazec (ang. form) osnova:

- za vnos podatkov (s tem nastane dokument),
- za prikaz podatkov in
- za izpis dokumentov v izbrani obliki

V eni Domino zbirki lahko uporabljamo več različnih obrazcev, z vsakim od njih pa sestavljamo določen tip dokumentov. Domino zbirka vsebuje dokumente, ki so sestavljeni iz enega ali več obrazcev.

Obrazec lahko vsebuje:

- polja, ki vsebujejo podatke.
- besedilo (statična besedila), ki opisuje polje oziroma nudi navodilo za izpolnjevanje polj.
- podobrazce, ki so zbirke elementov obrazca, ki se pojavljajo večkrat v različnih obrazcih. Podobrazec nam omogoča shranjevanje polj in drugih stičnih elementov na enem mestu. Podoben je poljem v skupni rabi, ki jih ravno tako lahko uporabljamo na več obrazcih.
- grafiko, ki daje obrazcu lažjo razumljivost
- tabele, ki povzemajo oziroma organizirajo informacije
- akcije in gumbе, ki izvajajo določene funkcije
- pripete dokumente, URL¹⁰ in druge povezave
- ozadje in grafiko, ki dajo obrazcu izgled

Ko določimo osnovno strukturo obrazca, lahko dodajamo polja na obrazec. Polja na obrazcu predstavljajo poimenovano področje, ki vsebuje določen tip podatkov. Polja so elementi obrazca za shranjevanje podatkov, ki določajo, kakšne podatke lahko posamezen obrazec (tekst, številke, datum, uporabnikovo ime) vsebuje. Uporabnik lahko v polja vnaša podatke ali pa se podatki vnesejo avtomatsko na podlagi formule.

Polje je lahko definirano za uporabo na enem obrazcu ali pa je dano na razpolago večjemu številu obrazcev (ang. shared fields).

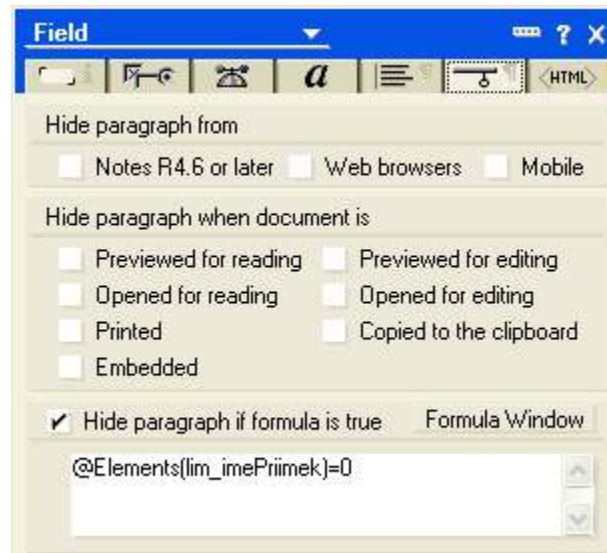
Na obrazcu lahko uporabimo dve vrsti polj:

- **Polje v enostavni rabi** je tisto polje, ki je definirano za uporabo v enem samem obrazcu. Če uporabimo enako definirano polje v drugem obrazcu znotraj iste zbirke, sta taki polji medsebojno neodvisni.
- **Polje v skupni rabi** lahko uporabljamo na različnih obrazcih v isti zbirki. Kadar spremenimo definicijo takega polja, sprememba velja v vseh pojavljanjih tega polja na različnih obrazcih.

Uporaba polja v skupni rabi pomeni prihranek časa za razvijalca aplikacije, ker definira polje samo na enem mestu.

Domino omogoča tudi skrivanje delov obrazca v posameznih načinih dela z »Hide-When« metodo. Skrivanje delov obrazca omogoča ustvarjanje skritih polj (glej sliko 7). Ta polja uporabniku v običajnem načinu dela niso vidna. Uporabljamo jih kot običajna polja.

¹⁰ URL = Uniform Resource Locators (Naslov spletnih strani v spletu)



Slika 7: Primer formule za skrivanje polja

Polju določimo tudi tip, kar določa vrsto podatkov za vnos v to polje. Poznamo več vrst tipov polj:

- **Besedilo (ang. text):** Tekstovno polje dovoljuje vnos podatkov kot zaporedja črk ali drugih znakov. Maksimalna dolžina tekstovnega polja je 15 KB. Slog besedila je vnaprej določen s slogom polja, zato ga uporabnik ne more spreminjati. Števil, ki so vnesena v tekstovno polje ne moremo uporabljati v matematičnih operacijah.
- **Obogateno besedilo (ang. richtext):** Polje za obogateno besedilo sprejme poljubno kombinacijo besedila, ki ga lahko oblikujemo s pomočjo grafičnih elementov, povezav z drugimi dokumenti, prilog, itd. Obseg polja za obogateno besedilo je praktično neomejen. Ima dve omejitvi:
 - ne moremo ga uporabljati v pregledih in
 - lahko ga uporabljamo kot argument v funkcijah, vendar pa ga ne moremo uporabljati v formulah v kombinaciji z drugimi polji.
- **Število (ang. number):** Številčno polje je namenjeno vnosu številčnih vrednosti, ki jih je treba matematično prikazati ali z njimi računati.
- **Datum in čas(ang. date and time):** Datumsko – časovno polje dovoljuje vnos zaporedja znakov v predpisanih oblikah za predstavitev datuma in časa, ki je odvisno tudi od operacijskega sistema.
- **Ključne besede (ang. Dialog list, Checkbox, Radio button, Listbox, Combobox):** Polja s ključnimi besedami omogočajo izbiranje vrednosti iz seznama vnaprej določenih podatkov. V taka polja lahko sami vnesemo seznam ali pa napišemo formulo, na osnovi katere Domino sam sestavi seznam. Če seznam ključnih besed ni popoln, lahko uporabniku dovolimo, da vpiše svoje ključne besede.
- **Ime avtorja (ang. authors):** Avtorsko polje omogoča vnos seznama uporabnikov, ki lahko spreminjajo dokument, v katerem se nahaja tako polje. Če obrazec vsebuje vsaj eno

avtorsko polje, potem po shranitvi lahko spreminjajo dokument le tisti uporabniki, ki so navedeni v tem polju.

Z izbranim obrazcem lahko sestavlja nove dokumente vsak uporabnik, ki ima pristopni nivo »avtor« v listi pristopnih pravic ACL¹¹ in je naveden tudi v listi uporabnikov CAL¹², ki lahko sestavljajo dokumente s tem obrazcem. Če dovolimo vnos podatkov v tako polje, lahko dejanski avtor dokumenta sam izbira še druge avtorje dokumenta.

Če ostane polje tipa »avtor« v dokumentu prazno, potem tudi sestavljalec dokumenta ne more spreminjati dokumenta, če nima v listi pristopnih pravic vsaj nivoja »urednik« (ang. manager).

- **Ime bralca (ang. readers):** Bralsko polje omogoča vnos seznama uporabnikov, ki lahko berejo dokument, v katerem se nahaja tako polje. Razumljivo je, da uporabniki, ki so navedeni v avtorskem polju lahko tudi berejo dokument, zato jih ni treba še enkrat navesti v polju bralca. Avtorji dokumenta praviloma lahko tudi dodajajo in brišejo imena uporabnikov, ki smejo brati dokument. Uporabniki, ki niso navedeni v bralskem polju (ali avtorskem polju), ne morejo videti dokumenta v nobenem pregledu, zato tudi ne morejo dokumenta brati.
- **Ime uporabnika (ang. names):** Imensko polje uporabljamo za prikaz imen uporabnikov, kadar sta avtorsko polje ali polje bralca neprimerna, ker ne želimo omejevati pravic uporabnikov.

Podatki navedeni v avtorskih in poljih bralcev ne prevladajo nad pristopnimi pravicami, ki so navedene v ACL, ampak le dodatno omejijo pravice uporabnikov.

Na obrazcih lahko definiramo polja, na katerih vrednost uporabnika lahko vpliva in taka na katera ne more vplivati neposredno. S tega stališča v Notes-u ločimo:

- **Spremenljivo polje (ang. editable):** Uporabnik lahko spremeni vrednost polja.
- **Izračunano polje (ang. computed):** Vredost polja se izračuna na osnovi danih podatkov, uporabnik pa ga ne more spremeniti.
- **Izračunano polje za prikaz podatka (ang. computed for display):** Isto kot za izračunano polje, vendar se vrednost polja vsakič izračuna le za prikaz na zaslonu, v dokument pa se ne shrani.
- **Izračunano polje ob sestavljanju dokumenta (ang. computed when composed):** Isto kot za izračunano polje, vendar se vrednost polja izračuna le enkrat in sicer ob sestavljanju dokumenta.

¹¹ ACL = Access Control List

¹² CAL = Compose Access List

Notes ima posebno možnost, da v polju navedemo oziroma naštejemo več vrednosti. To velja v vseh podatkovnih tipih polj razen obogatenega tekstovnega polja, ki že samo po sebi nudi skoraj poljuben obseg podatkov.

3.3.2.2 Pregledi

Pregled (ang. view) je vrstično urejen seznam dela dokumentov ali vseh dokumentov iz baze (glej sliko 8). Dobro načrtovan pregled omogoča uporabniku, da hitro najde dokument, ki je zanj zanimiv. Vrstica v pregledu predstavlja en dokument ali eno kategorijo in je razdeljena v več stolpcev. V stolpcu je prikazana vsebina enega polja, kombinacija več polj iz dokumenta ali rezultat formule. Na osnovi podatkov v stolpcih so dokumenti tudi razvrščeni.

Glede na to, kako so pregledi ustvarjeni, lahko izberemo, sortiramo ali kategoriziramo dokumente na različne načine.

| | Številka | Podjetje | Vrsta | Datum vpisa | Opis |
|------|-------------|----------|---------------------------|-------------|--|
| 2009 | | | | | |
| | 000866/2009 | | Ažuriranje limita | 04.06.2009 | Odobr. lim. št: 041/2 |
| | 000865/2009 | | Ažuriranje limita | 04.06.2009 | Odobr. lim. št: 041/2 |
| | 000863/2009 | | Ponudba | 02.06.2009 | 22.000,00 EUR, Obnova trgovine |
| | 000862/2009 | | Kreditni predlog | 02.06.2009 | 12.000,00 EUR, Nakup stanovanja |
| | 000848/2009 | | Bankart | 29.05.2009 | testiranje uvrsti |
| | 000846/2009 | | Račun | 28.05.2009 | 000003-2009: Dob. št.: 13123 |
| | 000845/2009 | | Akreditivi in garancije | 28.05.2009 | test |
| | 000843/2009 | | Potrditev in vpis rating: | 26.05.2009 | Vpis ratinga za |
| | 000842/2009 | | Ažuriranje limita | 26.05.2009 | Odobr. lim. št: 041/1 |
| | 000840/2009 | | Kreditni predlog | 15.05.2009 | 7.300,00 EUR, Obnova poslovnih prostorov |
| | 000830/2009 | | Potrditev in vpis rating: | 11.05.2009 | Vpis ratinga za |
| | 000829/2009 | | Ažuriranje limita | 11.05.2009 | Odobr. lim. št: 006/1 |
| | 000828/2009 | | Ažuriranje limita | 11.05.2009 | Odobr. lim. št: 006/1 |
| | 000820/2009 | | Predlog odobritve limite: | 11.05.2009 | Odobr. lim. št: 041/2 |
| | 000802/2009 | | Kreditni predlog | 08.05.2009 | 2.300,00 EUR, testiranje integratorja |
| | 000800/2009 | | Kreditni predlog | 08.05.2009 | 0,00 EUR, |
| | 000799/2009 | | Ponudba | 08.05.2009 | 2.300,00 EUR, testiranje integratorja |
| | 000798/2009 | | Ponudba | 07.05.2009 | 12.000,00 EUR, Nakup stanovanja |
| | 000797/2009 | | Predlog odobritve limite: | 07.05.2009 | Odobr. lim. št: 041/1 |
| | 000790/2009 | | Kratkoročni kredit | 07.05.2009 | 22.000,00 EUR, test11 |

Slika 8: Primer pregleda v Domino zbirki

Za vsak pregled razvijalec zapiše formulo izbire, ki določi, kateri dokumenti bodo prikazani v pregledu. Formula lahko zajame vse dokumente v bazi ali le tiste, ki izpolnjujejo določen pogoj.

Uporabnik lahko izvede kar nekaj opravil v pregledu. Lahko odpre dokument (podatki se prikažejo preko obrazca dokumenta), se med dokumenti premika, izbere dokument za izvajanje

določenih akcij (npr. tiska, izvaža, osveži polja itd.), izbriše dokument, kopira dokument in osveži pregled.

Pregledi so lahko skupni ali privatni. Skupni pregledi so na voljo vsem uporabnikom Domino zbirke, razen v primeru, ko so napisane določene omejitve v seznamu dostopa, ki se nanašajo na ta pregled. Privatni pregled lahko vidi le oseba, ki ga je tvorila. Za ustvarjanje privatnega pregleda moramo imeti vsaj bralni dostop do zbirke. Privatni pregled uporabnika lahko prikaže le tiste dokumente do katerih ima uporabnik dostop. Zaščiteni podatki in dokumenti, za katere uporabnik nima pravice branja, se ne prikažejo v pregledu.

Pregled lahko pripravimo tudi na obrazec ali stran. Tak način prikazovanja nudi razvijalcem več nadzora nad uporabnikovo uporabo dokumentov v zbirki. Take preglede imenujemo vpeti pregledi (ang. embedded views). Vpeti pregledi uporabljajo polja na obrazcu za interakcijo s pregledom in tako omogočajo bolj dinamično interakcijo z uporabnikom.

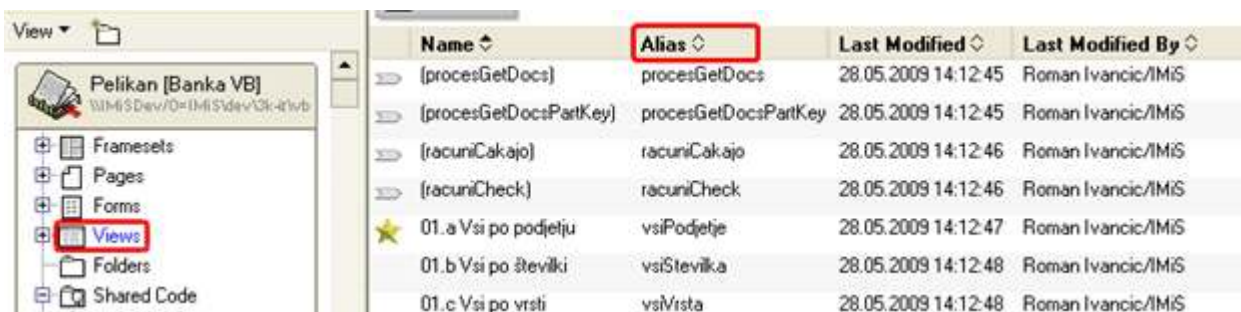
V pregledih, ki jih uporabljamo v formulah, LotusScriptu ali agentih moramo paziti, ko shranjujemo dokument v back-endu. V takem primeru moramo pregled osvežiti, da se nam ta dokument prikaže v pregledu. To lahko rešimo programsko s klicem funkcije »refresh« ali pa ročno s klikom na opozorilni znak, ki nam pove, da je prišlo do spremembe pregleda.

Skrivanje pregleda

Če želimo pregled skriti pred uporabniki podatkovne zbirke, da se ta ne prikaže v meniju, moramo zapisati ime pregleda v oklepaje (npr. »(vsiDokumentiPoPodjetju)«). Skriti pregledi so namenjeni uporabi v formulah, LotusScriptu in agentih.

Uporaba sinonimov (ang. alias)

Sinonimi nam omogočajo, da spremenimo ime pregleda v meniju »View« (glej sliko 9). Pri tem nam ni treba prepisovati formul, ki vsebujejo sklic na originalno ime.



| Name | Alias | Last Modified | Last Modified By |
|------------------------|----------------------|---------------------|--------------------|
| [procesGetDocs] | procesGetDocs | 28.05.2009 14:12:45 | Roman Ivancic/IMIS |
| [procesGetDocsPartKey] | procesGetDocsPartKey | 28.05.2009 14:12:45 | Roman Ivancic/IMIS |
| [racuniCakajo] | racuniCakajo | 28.05.2009 14:12:46 | Roman Ivancic/IMIS |
| [racuniCheck] | racuniCheck | 28.05.2009 14:12:46 | Roman Ivancic/IMIS |
| 01.a Vsi po podjetju | vsiPodjetje | 28.05.2009 14:12:47 | Roman Ivancic/IMIS |
| 01.b Vsi po številki | vsiStevilka | 28.05.2009 14:12:48 | Roman Ivancic/IMIS |
| 01.c Vsi po vrsti | vsiVrsta | 28.05.2009 14:12:48 | Roman Ivancic/IMIS |

Slika 9: Prikaz pregledov v Lotus Domino Designer-ju s sinonimi

Indeksi

Vsakemu pregledu pripada indeks, s katerim Domino nadzira zaporedje dokumentov v pregledu. Kadarkoli uporabnik doda ali spremeni dokument v zbirki ali pa ga iz zbirke odstrani, mora Domino sestaviti nov indeks, da lahko v pregledu prikaže nov razpored dokumentov. Indeks, ki pripada pregledu, je neodvisen od indeksa za iskanje po polnem besedilu. Če hočemo izboljšati učinkovitost sistema, moramo v nastavitvah pregleda določiti, na kakšen način in kako pogosto naj se indeks osveži.

Načini osveževanja indeksa:

- Display indicator – v levem zgornjem kotu pregleda se prikaže opozorilni znak, da je prišlo do spremembe pregleda
- Refresh display – pregled se avtomatsko osveži ob vsaki spremembi
- Refresh display from top row – pregled se avtomatsko osveži ob vsaki spremembi. To možnost izberemo, ko je pregled kronološko urejen tako, da uporabniki iščejo dokumente na začetku pregleda.
- Refresh display from bottom row – pregled se avtomatsko osveži ob vsaki spremembi. To možnost izberemo, ko je pregled kronološko urejen tako, da uporabniki iščejo dokumente na koncu pregleda.

Nastavitev pogostosti osveževanja indeksa:

- Auto, after first use – Indeks se osveži šele ob prvi uporabi zbirke po prvi spremembi
- Automatic – indeks se osveži avtomatsko po izvedeni spremembi, ne glede na uporabo zbirke.
- Manual – v tem primeru mora uporabnik sam poskrbeti, da se indeks osveži.(opcija je uporabna za velike zbirke, saj se zbirka hitreje odpre, če ni potrebno sprotno osveževanje)

3.3.3 Domino imenik

Domino imenik (ang. Domino Address Book) je prav posebna dokumentna zbirka oziroma zbirka dokumentov, ki omogoča normalno delovanje Domina in Notesa. Zato je tudi najpomembnejša zbirka v Domino sistemu, saj brez nje Domino strežnik ne more delovati. Ta zbirka ima privzeto ime `names.nsf`.

Domino imenik vsebuje naslednje informacije:

- kako bo dostavljena pošta uporabniku,
- kako je postavljen vsak strežnik v domeni,
- kako poteka določeno opravilo(task) v Domino sistemu,
- kako poteka komunikacija med strežniki,
- na koliko časa se izvaja replikacija in usmerjanje pošte,
- kateri zunanji elementi lahko dostopajo do strežnika,
- kakšne so varnostne nastavitve v Domino sistemu

Vse potrebne spremembe v Domino imeniku izvajamo prek odjemalca Domino Administrator, saj ima samo ta potrebna orodja za polno dosegljivost določenih funkcij v sistemu Domino. V bistvu je Domino Administrator odjemalec zbirke `names.nsf`. Seveda ima tudi druge dostope do nastavitvenih zbirk, vendar je ta najpomembnejši.

3.3.4 Zbirka razgovorov

Zbirka razgovorov (ang. Discussion) je namenjena izmenjavi mnenj in idej. Uporablja jo lahko inženirska skupina, ki razpravlja o produktu, ki ga razvija, raziskovalna skupina za pogovor o raziskovalni nalogi itn. V zbirki je prikazana zgodovina razprav tako, da lahko novi člani, ki bi se radi seznanili s temami dosedanjih pogovorov in nasploh z delovanjem skupine, to hitro in učinkovito storijo.

3.4 Uporaba programskih jezikov v Domino Notes okolju

3.4.1 Jezik formul

Jezik formul (ang. formula language) je preprost ukazni jezik. Z njim krmilimo obnašanje odjemalca, kličemo akcije, itd., ki pa za daljše procedure kmalu postane nepregleden. Čeprav se mnogi otepajo programiranja s formulami, pa je njihova odlika hitrost, saj je izvedba neke akcije s formulami lahko tudi do petkrat hitrejša kot z ostalimi jeziki [2].

V Dominu uporabljamo formule v številnih situacijah:

- pri kontroli podatkov, ki jih vnesemo v polje,
- za določitev vrednosti v izračunanem polju,
- pri določanju vsebine stolpcev v pregledu,
- za izbiranje dokumentov, ki jih prikažemo v pregledu,
- za spreminjanje podatkov v dokumentih preko agentov in
- pri izbiranju obrazca, ki naj se uporabi pri prikazu dokumenta v različih situacijah.

Domino formula je kombinacija naslednjih elementov:

- spremenljivk
- konstant
- funkcij
- operatorjev
- rezerviranih besed

Te elemente povežemo med seboj po določenih pravilih, ki jim pravimo sintaksa formul. Najpogostejša vrsta spremenljivke v Dominu je polje, ki ga definiramo na obrazcu. Če želimo

polje uporabljati kot spremenljivko, v formulo enostavno vpišemo njegovo ime. Poleg tega lahko definiramo začasna polja, ki dejansko obstajajo le znotraj formul. Začasna polja uporabljamo za izračun vmesnih vrednosti med izračunavanjem formul. Začasno polje preneha obstajati, ko se izračuna vrednost formule. Pri izbiri imen začasnih ali stalnih polj moramo paziti, da ne uporabimo imen, ki so namenjena rezerviranim besedam (npr. FIELD, REM, SELECT, DEFAULT, ...).

Funkcije so nadomestilo za zahtevne formule in omogočajo izračun rezultatov na osnovi začetnih vrednosti po določenem postopku. Na razpolago imamo preko 100 funkcij, ki večinoma operirajo z besedili, saj je Domino tekstovno orientiran program.

Pri uporabi funkcij veljajo posebna sintaktična pravila:

- vse funkcije se prično z znakom @, sledi pa jim predpisano ime funkcije (npr. @Trim(niz), @Length(niz), @Left(niz;n), @Now, ...)
- če funkcija potrebuje argumente ali konstante za izračun rezultata, le te zapišemo med okrogle oklepaje, medsebojno pa jih ločimo s podpičjem
- če funkcija ne potrebuje argumentov, oklepajev ne pišemo
- če funkcija potrebuje kot argument listo, podatke ločimo z dvopičjem
- s funkcijo je predpisano, kakšen podatkovni tip argumenta je dovoljen, pa tudi kakšnega tipa je rezultat funkcije. To moramo paziti predvsem v formulah pri kombinaciji večih spremenljivk, zato moramo po potrebi pretvoriti podatkovne tipe na isti nivo.

V Domini uporabljamo tri vrste izračunanih polj (ang. Computed, Computed for display, Computed when composed), ki sem jih opisal v poglavju 3.2.1.1, v katerih s formulo predpišemo izračun njihove vsebine. Te vsebine uporabnik ne more več spreminjati. Taka polja uporabljamo, ko je njihova vsebina odvisna od drugih polj v dokumentu.

Pri delu z zbirko se določena opravila ponavljajo, kot na primer:

- ustvarjanje novega dokumenta: @Command([Compose]; "dok")
- shranjevanje dokumenta: @Command([FileSave])
- dokument dam v urejevalen način: @Command([EditDocument])

V zbirkah dokumentov za ponavljiva opravila znotraj pregledov ali dokumentov ustvarimo ustrezno akcijo, v katero vključimo Domino menijske ukaze. Za to v Domino obstaja posebna kategorija funkcij @Command, ki omogočajo večino Domino menijskih ukazov. Take funkcije uporabljamo v formulah za akcije¹³, na gumbih, aktivnih mestih in nekaterih agentih. Primeri vnaprej pripravljenih sistemskih akcij: kategoriziranje, odpiranje dokumenta v urejevalnem načinu, pošiljanje dokumenta, ..

¹³ Akcija je postopek, ki ga lahko izvedemo na izbranem obrazcu ali pregledu.

3.4.2 Skriptni jezik JavaScript

JavaScript skriptni jezik lahko uporabljamo za programiranje tako spletnega kot tudi Notes odjemalca. Ta jezik je bil dodan predvsem za izboljšanje funkcionalnosti spletnega uporabniškega vmesnika. V Notes odjemalca pa je bil dodan zato, da lahko isto kodo uporabimo v obeh odjemalcih.

3.4.3 LotusScript

LotusScript [3] je programski jezik, ki razširja razvojne zmožnosti Domino aplikacij. Zelo je podoben Microsoftovemu VBA¹⁴ in je objektno orientiran jezik. Imena spremenljivk so lahko dolga največ štirideset znakov in niso znakovno občutljiva (ang. case sensitive). Ne razlikuje med malimi in velikimi črkami abecede in prvi znak mora biti črka. Ta pravila veljajo tudi za imena konstant, tipov, funkcij in drugih gradnikov jezika.

LotusScript podpira tudi splošne podatkovne strukture, kot so polja in sezname. Podpira tudi objektno orientirane konstrukte, kreiramo lahko svoje razrede, metode in lastnosti.

3.4.4 Java

Javo lahko uporabljamo znotraj Domino Notes aplikacije tako, da pišemo razrede v programske elemente. Domino ima vključeno svojo različico JVM-ja¹⁵. Iz samostojne java aplikacije je omogočen dostop do podatkov v Domino zbirki preko CORBA¹⁶ standarda. CORBA je premoščanje platform in jezikov za komunikacijo med aplikacijskimi komponentami. Kličejo se lahko metode objektov in tudi prenašajo se celotni objekti s serializacijo.

Domino objektni model je API¹⁷ plast do Domino zbirke. Zgrajen je kot vmesnik do hierarhične strukture objektov, s katerimi lahko manipuliramo dokumente in strukturne elemente zbirke. Vsi zgoraj naštetih jeziki imajo dostop do API vmesnika, vendar le v LotusScript jeziku lahko uporabljamo celotno funkcionalnost. Razrede v Domino objektnem modelu razdelimo v dve skupini:

- »**Front end**« razredi manipulirajo uporabniški vmesnik Notes odjemalca
- »**Back end**« razredi, pa manipulirajo podatkovni model (dokumente)

¹⁴ VBA = Visual Basic for Applications

¹⁵ JVM = Java Virtual Machine (Javanski navidezni stroj)

¹⁶ CORBA = Common Object Request Broker Architecture

¹⁷ API = Application Programming Interface (Aplikacijski programski vmesnik)

Agenti so majhni programi, ki opravijo neko nalogo. Agenti se sprožijo na več načinov. Lahko je to uporabniška akcija, lahko je določeno po urniku, lahko pa je nek drug dogodek, ki se zgodi v sistemu (prihod novega poštnega sporočila, sprememba obstoječega dokumenta, itd.).

Knjižnice so strukturni elementi, ki so namenjeni shranjevanju programske kode za ponovno uporabo. Primerne so za pisanje procedur in razredov, ki jih uporabimo v drugih strukturnih elementih in se tako izognemo podvajanju kode. Domino Notes podpira JavaScript, LotusScript in Java knjižnice¹⁸.

3.5 Varnost

Domino je že od samega začetka namenil veliko pozornosti varnosti podatkov [10]. Svoj produkt Lotus Notes je opremljal z najnovejšimi idejnimi in tehnološkimi dosežki na področju varovanja podatkov. S prihodom Domina je bil prisiljen svoj model varovanja podatkov razširiti na internet in ga dopolniti z novimi tehnološkimi dosežki zaščite podatkov na internetu. Predstavil bom osnovna načela za varovanje podatkov v okolju Domino.

3.5.1 Preverjanje pristnosti in nadzor dostopa

Preverjanje pristnosti (ang. Authentication) je proces identifikacije uporabnikov – odjemalcev. Uporabnikova identiteta se preverja pri vsakem poskusu dostopa do strežnika. Nadzor dostopa vključuje omejevanje pravic in funkcionalnosti uporabnikom potem, ko je strežnik že preveril njihovo pristnost. Preverjanje pristnosti in nadzor dostopa sta osnovna koncepta, ki jih mora razumeti vsak administrator, preden lahko učinkovito implementira varnostni model.

Temelj varnostnega modela je preverjanje pristnosti. Če pristnosti uporabnika ne preverimo sto odstotno, nam ne pomaga še tako učinkovit sistem nadzora dostopa (ang. Access Control). V primeru, da se nekdo predstavi kot druga oseba, nam ne pomaga omejevanje dostopa glede na identiteto uporabnika. Pogosti tehniki za preverjanje pristnosti sta geslo in ID značka. Kljub temu, da postane preverjanje pristnosti lahko zelo kompleksen problem, pa vsi sistemi spadajo v eno od naslednjih kategorij:

- sistemi, ki se zanašajo na nekaj, kar veš (npr. geslo),
- sistemi, ki se zanašajo na nekaj, kar imaš (npr. ID značka) ali
- sistemi, ki se zanašajo na to, kar si (npr. prstni odtis ali DNA).

¹⁸ Javanske knjižnice so bile podprte z verzijo Notes Domino 6

Večina sistemov se je do sedaj zanašala na »nekaj, kar več«, kot je na primer geslo. Pri tem pristopu je veliko problemov in je zato stopnja zaščite vprašljiva. Sistemi, ki uporabljajo geslo, so bili zgrajeni na ta način, ker na področju zaščite ni bilo veliko izbire.

Domino je bil eden prvih velikih programskih paketov, ki je integriral tak model preverjanja pristnosti direktno v produkt. Domino uporablja pri preverjanju pristnosti imena, šifrirne ključe in potrdila.

Seznami nadzora dostopa (ang. Access Control List – ACL) so najbolj običajna pot za omejevanje dostopa do računalniških sistemskih virov. Seznami nadzora dostopa so enostavno spiski imen. Vsako ime ima določena dovoljenja (ang. permissions), ki dajejo imenovanemu uporabniku določene pravice. Uporabnik, ki ga ni na seznamu dostopa, ne more dostopiti do želenega vira, ki je zaščiten s seznamom. Domino uporablja ACL za zaščito dokumentov, aplikacij in strežnikov.

3.5.2 Preverjanje pristnosti v Dominu

Notes omogoča dvosmerno preverjanje pristnosti: strežnik preveri identiteto uporabnika in uporabnik preveri identiteto strežnika. Drugi korak je pomemben pri komunikaciji preko javnega omrežja (npr. interneta).

Ko uporabnik prvič dostopi do strežnika, si odjemalec in strežnik izmenjata potrdila (ang. certificates). Obe strani morata imeti potrdilo, ki mu druga stran zaupa. Pri Dominu se potrdilo zaupa samo, če je kopija potrdila vsebovana v lokalni ID datoteki.

Ni nujno, da ID datoteke vsebujejo enako potrdilo, pomembno je le, da imata obe potrdili enakega prednika. Strežniki in odjemalci, ki imajo v ID datotekah potrdila s skupnimi predniki, lahko komunicirajo brez omejitev. ID datoteka lahko vsebuje več kot eno potrdilo. Ta situacija se najpogosteje pojavi, ko želita medsebojno komunikacijo vzpostaviti strežnika iz različnih organizacij. Pred začetkom prve komunikacije morata ta dva strežnika izmenjati potrdila. Hierarhično potrdilo, prejeto od druge organizacije, se imenuje navzkrižno potrdilo (ang. cross certificate). Navzkrižna potrdila je potrebno izmenjevati vedno na najnižjem možnem nivoju. Na ta način se zagotovi najvišjo stopnjo varnosti.

Kljub temu, da je bistvo potrdil povezovanje potrdil in ključev, pa Domino ne preverja, ali je javni ključ shranjen v imeniku enak tistemu v ID datoteki uporabnika, ki dostopa do sistema. Takšno preverjanje upočasnjuje sistem in je nepotrebno. Domino zato uporablja tehniko izziva in odgovora. Ko uporabnik (ali strežnik) vzpostavi zvezo s strežnikom, mu pošlje naključno številko. Strežnik zašifrira to številko z uporabo svojega privatnega ključa in jo šifrirano vrne

uporabniku. Uporabnik šifrirano sporočilo dešifrira z uporabo javnega ključa strežnika, ki je dostopen v javnem imeniku. Če se dešifrirana številka ujema s poslano, je lahko uporabnik prepričan, da je dostopil do zelenega strežnika. Proces se potem ponovi v obratni smeri, tako da strežnik izzove uporabnika. Po končanem postopku izziva in odgovora se preveri potrdila za validacijo identitet, ki so povezane s ključi.

3.5.3 Nadzor dostopa v Domino okolju

Domino uporablja več različnih seznamov nadzora dostopa. Ločeni seznammi nadzirajo dostop do strežnikov, aplikacij in dokumentov. Večina ACL sprejema navadna imena in polna hierarhična imena. Pri uporabi polnih hierarhičnih imen dosežemo večjo stopnjo varnosti, zato je tak pristop priporočljiv. V nadaljevanju bom opisal uporabo ACL v Domino okolju.

3.5.3.1 Zaščita na nivoju strežnika

Dostop do strežnika je kontroliran z ACL, ki je prikazan na sliki 10. Vsak strežnik ima več ACL-ov. Vsi ti seznammi so del strežniškega dokumenta v javnem imeniku.

Seznammi vključujejo:

- »dovoli – dostop« seznam (ang. allow - access list)
- »prepreči – dostop« seznam (ang. deny - access list)
- seznam ljudi, ki lahko ustvarjajo aplikacije
- seznam ljudi, ki lahko ustvarjajo dvojnike (ang. replication)
- seznam ljudi, ki lahko uporabljajo Domino prepustni strežnik
- seznam strežnikov, ki so dosegljivi preko prepustnega strežnika
- seznam administratorjev za ta strežnik

| Server Access | Who can - | Pass thru Use | Who can - |
|--------------------------------|-------------|-----------------------|----------------|
| Access server: | */IMIS | Access this server: | Pass thruUsers |
| Not access server: | DeniedUsers | Route through: | |
| Create databases & templates: | DominoAdmin | Cause calling: | |
| Create new replicas: | | Destinations allowed: | |
| Create master templates: | | | |
| Allowed to use monitors: | | | |
| Not allowed to use monitors: * | | | |
| Trusted servers: | | | |

Slika 10: Strežniški ACL

Mnogi od teh seznamov so specializirani. Kontrolirajo seznam do dela Domino okolja, ACL-i pa kontrolirajo dostop do sistemskih Domino virov.

Administrator lahko dovoli dostop brez preverjanja pristnosti. V tem primeru se Domino izogne preverjanju pristnosti in sprejme predlagano ime s strani uporabnika ali strežnika. Domino sledi vsem uporabnikom, katerim pristnost ni bila preverjena. Vse takšne uporabnike poimenuje »anonimnež« (ang. anonymous) in omogoči, da tudi takšnim uporabnikom dodelimo nivo dostopa. Tudi če dovolimo dostop uporabnikom z nepreverjeno pristnostjo, sistema ne pustimo popolnoma brez obrambe. Omogočanje take vrste dostopa je zelo uporabno pri javnih strežnikih in javnih aplikacijah.

3.5.3.2 Zaščita na nivoju aplikacije

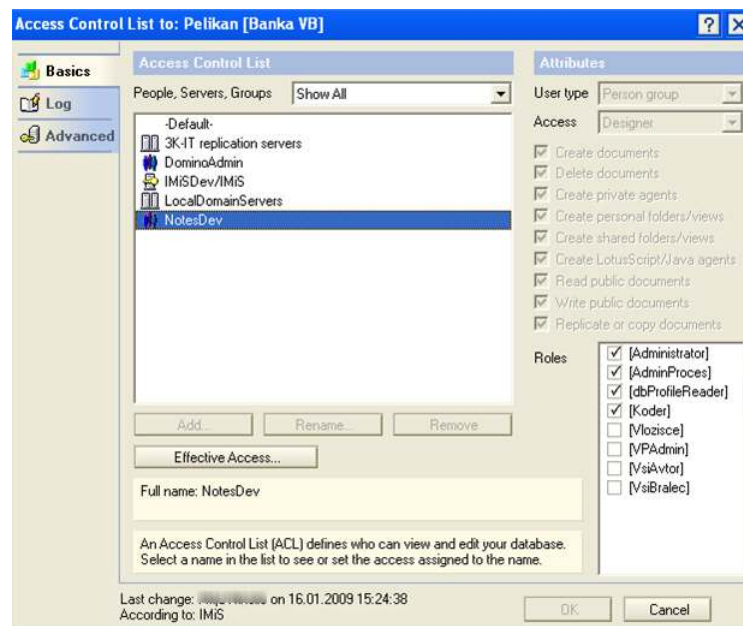
Aplikacijo lahko zaščitimo na več različnih načinov. V Domino okolju so uporabljene naslednje tehnike:

3.5.3.2.1 Skrivanje aplikacije (zbirke)

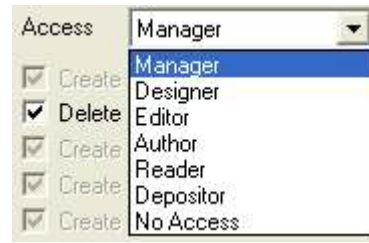
Aplikacijo ali posamezno zbirko (kot njen sestavni del) lahko skrijemo tako, da je ne navedemo v nobenem katalogu ali v pogovornem oknu za odpiranje zbirk. Ta metoda ni zanesljiva, je pa učinkovita.

3.5.3.2.2 Aplikacijski ACL

Primarna metoda za nadzor dostopa do aplikacije (zbirke) je ACL. Vsako ime na seznamu ima določen nivo dostopa. Vsaka Domino zbirka ima svoj ACL (glej sliko 11).



Slika 11: Aplikacijski ACL



Slika 12: Nivoji dostopa

ACL omogoča naslednje nivoje dostopa do baze (glej sliko 12):

Upravitelj (ang. Manager Access)

Kdorkoli z upraviteljskim dostopom lahko bere ali spreminja vse nešifrirane podatke, razen dokumentov, ki so zaščiteni s poljem »Imena bralcev«. Uporabnik z upraviteljskim dostopom ima polna pooblastila pri spreminjanju načrta zbirke (obrazci, pregledi, agenti, ...). Najpomembneje pa je, da lahko uporabnik z upraviteljskim dostopom spreminja ACL. To je tudi edini tehtni razlog, da nekomu dodelimo ta nivo dostopa. Upravitelji so ponavadi administratorji zbirke in drugi strežniki v organizaciji. Vse Domino zbirke morajo imeti vsaj enega upravitelja.

Načrtovalec (ang. Designer Access)

Uporabnik s tem nivojem dostopa ne more spreminjati ACL-a, opravlja pa lahko vse ostale funkcije, ki jih sme opravljati upravitelj.

Urednik (ang. Editor Access)

Uporabnik s tem nivojem dostopa lahko bere, spreminja in briše vse nešifrirane dokumente. Ne more spreminjati načrta baze ali ACL-a.

Avtor (ang. Author Access)

Uporabnik s tem nivojem dostopa ne more popravljati nobenega dokumenta, tudi tistega ne, ki ga je ustvaril sam. Edina izjema so tisti dokumenti, v katerih je uporabnik naveden v poljih »Imena avtorjev«. Tak uporabnik lahko ustvari nove dokumente in bere vse ostale dokumente, ki niso šifrirani ali zaščiteni s poljem »Imena bralcev«.

Bralec (ang. Reader Access)

Uporabnik s tem nivojem dostopa lahko bere vse dokumente, ki niso drugače zaščiteni. Ne more pa ustvariti, brisati ali popravljati dokumentov.

Dodajalec (ang. Depositor Access)

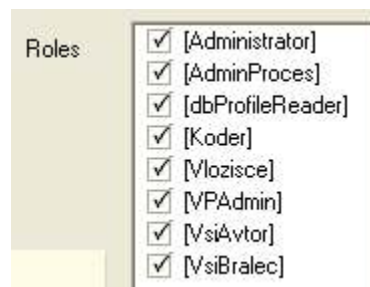
Uporabnik s tem nivojem dostopa lahko ustvari dokumente, ne more pa jih videti. Dodajalec ne more spreminjati ali brisati dokumentov iz zbirke. Nivo dostopa »Dodajalec« je največkrat uporabljen v primeru usmerjanja in pošiljanja elektronske pošte na zbirki v kateri se stekajo vsa sporočila (ang. mailbox).

Brez dostopa (ang. No Access)

Tak nivo dostopa popolnoma prepreči dostop do zbirke. Navadno se tak nivo dostopa uporablja kot privzet nivo dostopa do zbirke. V ACL-u vedno obstaja ime »Privzeto« (ang. default). Dodelitev nivoja dostopa »Privzeto«, pomeni, da bodo vsi uporabniki, ki jih ni v ACL dobili ta nivo dostopa. Dodatno lahko dodamo v ACL ime »Anonimnež« (ang. Anonymous) in s tem opredelimo nivo dostopa vsem uporabnikom, katerih pristnost ni bila preverjena.

3.5.3.2.3 Vloge

Vloge (ang. Roles) ščitijo dostop do specifičnih obrazcev, pregledov ali polj. V preteklosti se vlog ni uporabljalo pogosto. Enak efekt kot z vlogami je možno doseči z uporabo skupin, ki so specifične za posamezne aplikacije. Vloge so našteje v ACL-u, tako kot je prikazano na sliki 13. Njihova izvedba je enaka izvedbi skupin. Bistvena razlika med obema je v njunem nastanku. Skupine je potrebno definirati v javnem imeniku in so uporabne v vseh aplikacijah, vloge pa definiramo v posameznih aplikacijah. Vlog ne moremo uporabiti v različnih aplikacijah.



Slika 13: Vloge

Prednosti uporabe vlog so:

- vloge ne vplivajo na transparentnost javnega imenika,
- za dodeljevanje vlog administratorji ne potrebujejo pravic za spreminjanje javnega imenika,
- uporaba vlog znotraj uporabe načrta omogoči enostavnejše vzdrževanje celotne aplikacije. Spremembo naredimo le enkrat v vlogi, ne pa povsod, kjer se določeno ime uporablja,
- z uporabo vlog nazorno pokažemo, da v aplikaciji obstajajo procesi, ki zahtevajo akcijo točno določene skupine ljudi.

3.5.3.3 Zaščita na nivoju dokumenta

Ta nivo zaščite se ukvarja s pravicami uporabnikov do spreminjanja, brisanja in branja posameznih dokumentov. Takšno zaščito lahko uporabimo za dodeljevanje pravic splošno po vrstah dokumentov ali pa direktno v posameznih dokumentih.

Zaščita na nivoju dokumenta (ang. Document – Level Security) uporablja za nadzor nad dostopom naslednje elemente:

- Polje »Imena bralcev«: V tem polju so navedeni vsi, ki imajo pravice ta dokument brati – pregledovati, ne pa popravljati ali brisati. Pri uporabnikih, ki niso navedeni v polju, se dokument ne prikaže v nobenem od pregledov, tako da se ti uporabniki ne zavedajo, da dokument sploh obstaja.
- Polje »Imena avtorjev«: Spisek v tem polju vsebuje imena uporabnikov, ki lahko dokument pregledujejo in popravljajo, če pa jim ACL to dovoljuje, lahko tudi brišejo.

Zaščita na nivoju dokumenta se podreja aplikacijskemu ACL-u. Uporabnik recimo nikoli ne more popravljati dokumentov, če aplikacijski ACL tega ne dovoljuje. Uporabnik, ki je v ACL-u naveden kot urednik, pa lahko popravlja tudi dokumente, v katerih ni naveden v polju »Imena avtorjev«.

3.5.3.4 Zaščita na nivoju polja

Obstajajo trije načini nadzora dostopa do informacij na nivoju polja:

- kodiranje (ang. encryption),
- skrivanje polja (ang. field hiding) in
- zaščiteni odseki (ang. protected sections)

Edino šifriranje resnično zaščiti podatke v polju. Drugi dve metodi sta uporabni pri izdelovanju kvalitetnih uporabniških vmesnikov in pri preprečevanju uporabniških napak.

Posebnost razbijanja dokumenta v sekcije je v tem, da pri tej opciji lahko uporabimo zelo uporabne elektronske podpise. Elektronski podpis se avtomatično generira z uporabo odjemalčeve ID datoteke.

4 OPIS DOKUMENTNEGA SISTEMA

Dokumentni sistem »3K Document Cycle« je programska rešitev, ki je namenjena obvladovanju poslovnih procesov in dokumentacije. Poslovni dokumenti podjetja so shranjeni v različnih mapah in fasciklih, z vpeljavo dokumentnega sistema pa te dokumente lahko vključimo v informacijski sistem. Dokument v sistemu vsebuje eno ali več priponk (nestrukturirani podatki) in opisne podatke (strukturirani podatki). V dokument se lahko shranjuje priponke poljubnega formata: npr. skenirane dokumente (.tiff) - (najbolj pogosto), MS Office dokumente (.doc, .xls), slike, filme, ... Dokumentni sistem se integrira v obstoječe informacijske sisteme in jih nadgrajuje.

4.1 Upravljanje z dokumenti

Organizacija dokumentov

Za čim bolj učinkovito spremljanje dokumentacije so v aplikaciji definirane vrste dokumentov, ki so organizirane v drevesni strukturi. Struktura je prilagodljiva, zato da rešitev čim bolj ustreza potrebam podjetja. Vrsta dokumenta določa strukturirane podatke in poslovno logiko dokumenta. Vsaka vrsta dokumenta nosi tudi informacijo o pravicah dostopa in morebitnem poslovnem procesu. Tipične vrste dokumentov so račun, pogodba, tehnična dokumentacija, ...

Vnos dokumentov

Glavna načina za dodajanje pripetih datotek v aplikacijo sta skeniranje in zajem dokumentacije, ki sta že v elektronski obliki. Dokumentacijo, ki prispe po navadni pošti, zajemamo v aplikaciji s skeniranjem. Postopek zajema (skeniranja) poteka direktno iz aplikacije z odjemalcem IMiS/Scan, ki ga je razvilo podjetje IS d.o.o. kjer sem zaposlen.

IMiS/Scan produkt [8] je namenjen uporabi na delovnih postajah z optičnimi čitalniki. Dokument v slikovni obliki lahko opremimo z dodatnimi podatki na različne načine. Posamezne dele lahko poljubno obarvamo, dopišemo ali narišemo opombe ali dodatna pojasnila in vnesemo attribute oziroma ključne besede, ki nam kasneje olajšajo iskanje določenega dokumenta. Ko je dokument shranjen, je že na voljo vsem sodelavcem v omrežju. IMiS/Scan lahko shranjuje dokumente na štiri načine:

- ODMA priponke v Domino.Doc (Notes) dokumente,
- objekte na IMiS Arc strežnik,
- priponke v Notes dokumente(.tif, .pdf) in
- povezavo (ang. link) do datoteke, ki je shranjena kjerkoli na datotečnem strežniku

Ena od pomembnih lastnosti celotne IMiS arhitekture je v tem, da za hranjenje informacij o upodobljenem dokumentu, potrebuje največ 128 byte-ov prostora v Notes dokumentu. Povezava

IMiS-ovih modulov z Lotus Notes zbirko dokumentov je izredno preprosta. Del komponent so tudi Lotus Notes oblikovni elementi, namenjeni povezavi z Lotus Notes programskimi rešitvami. Te strukturne elemente razvijalec programske rešitve enostavno prepíše med ostale oblikovne elemente dokumentne zbirke in jih nato vključi v svoje obrazce kot podobrazce ali pa kliče funkcije iz knjižnic v agentih (akcijah), ki rokujejo z IMiS objekti. Za dostop do IMiS objektov uporabnik potrebuje IMiS/Scan ali IMiS/View odjemalca ali pa spletnega odjemalca.

V dokumentni sistem lahko dodajamo datoteke iz različnih pomnilniških medijev. Sistem glede na opredeljeno vrsto poskrbi za pravilno razvrstitev, pravice dostopa in morebitni začetek delovnega procesa.

Iskanje dokumentov

Z uporabo dokumentnega sistema število dokumentov v sistemu narašča, tako da je zelo pomembno tudi iskanje dokumentov.

Imamo dve možnosti iskanja:

- iskanje po pregledih/kategorijah: je iskanje po opisnih podatkih, npr. nazivu podjetja, datumu plačila,... ter
- iskanje po vsebini dokumentov: je podobno iskanju preko spletnih iskalnikov. Vpišemo iskano besedo ali več besed, sistem pa nam ponudi ovrednoten seznam zadetkov.

Povezovanje dokumentov

Sistem omogoča vzpostavitev povezav med dokumenti in povezovanje dokumentov v mape. Povezavo med dvema dokumentoma vzpostavimo, če sta dokumenta med sabo odvisna oziroma vsebinsko povezana. Medsebojno povezovanje dokumentov olajšuje spremljanje poslovnih procesov. Povezavo med dvema dokumentoma si lahko predstavljamo kot povezavo na kateri od spletnih strani. Vsak dokument je lahko povezan s poljubnim številom drugih dokumentov. Povezovanje v mape pomeni združevanje dokumentov po skupnem kriteriju (npr. vsi dokumenti, ki so del kreditne mape).

Varnost in sledljivost

Dokumenti v sistemu imajo poslovno vsebino, zato je zelo pomembno kdo lahko dostopa do posameznega dokumenta in kdo ga lahko spreminja. S pomočjo organizacijske sheme, definiranih vlog in nastavljenih pravil sistem avtomatično nastavlja pravice dostopa. Varnostno shemo lahko določimo tudi za vsako vrsto dokumenta posebej. Tu določimo pravice uporabnikom, ki lahko urejajo, berejo oziroma vidijo dokument.

Za vsak dokument je vedno zapisana celotna sled, kaj se je s tem dokumentom dogajalo skozi celoten poslovni proces.

Elektronski arhiv

Sistem omogoča centralno elektronsko hrambo dokumentov različnih formatov (doc, pdf, tif, ...). Za shranjevanje na daljši časovni rok in zagotavljanje zakonskih zahtev elektronskega arhiva je sistem možno povezati z drugimi arhivskimi sistemi. Eden od takih sistemov je tudi IMiS/ARChive, ki ga je razvilo podjetje IS d.o.o..

IMiS/ARChive [8] je aplikacijski strežnik, ki omogoča hierarhično arhiviranje dokumentov (ang. Hierarchical Storage Management - HSM). Deluje na Linux operacijskem sistemu, ki zagotavlja stabilno in zanesljivo okolje. To pomeni, da skrbi za optimalno zasedenost pomnilnih medijev in za čimbolj ustrezno razpoložljivost datotek oziroma dokumentov. Hierarhično arhiviranje dokumentov je obvezni podsistem vsakega sistema za računalniško upodabljanje in arhiviranje dokumentov. HSM sistem v bistvu simulira klasične metode arhiviranja, pri katerih imajo uporabniki na svojih mizah najbolj aktualne dokumente, medtem ko se starejši nahajajo v omarah ali kletih. Razlika med ročnim in avtomatičnim sistemom je v tem, da uporabniku ni treba skrbeti za dostavo in iskanje dokumentov (iz arhiva). Na ta način preprečimo sicer pogosto izgubljanje dokumentov ali njihovo nepotrebno fotokopiranje. Z uporabo različnih medijev zgradimo neke vrste elektronski arhiv v katerem dokumenti »potujejo« iz medija na medij – odvisno od pogostosti uporabe.

Integracija z drugimi sistemi

Dokumentni sistem je del nekega informacijskega sistema organizacije, zato se na več ravneh povezuje z njegovimi ostalimi deli:

- povezava z ERP sistemi (SAP, Pantheon, ...),
- povezava z elektronsko pošto – zajem dokumenta, avtomatično obveščanje itd.,
- povezava z MS Office orodjem ali
- povezava z ostalimi arhivskimi sistemi.

4.2 Upravljanje poslovnih procesov

Definicije poslovnih procesov

Dokumentni sistem 3K Document Cycle omogoča dinamično nastavljanje poslovnih procesov. Vsak proces je sestavljen iz statusov (korakov), ki določajo katere akcije (prehodi med statusi) so na voljo in kateri uporabniki ali skupine uporabnikov so potrjevalci v tem statusu. Določitev oseb in omejitev posameznih akcij je mogoče s pogoji in pravili zelo natančno opredeliti (npr. pri odobritvi limita je v primeru zneska nad določeno vsoto potrebna še potrditev direktorja). Možne so tudi nastavitve avtomatičnih obveščanj in avtomatičnih prehodov med statusi.

Izvajanje poslovnih procesov

Vsak dokument, za katerega je določen poslovni proces, sistem upravlja in nadzoruje v skladu s pravili, ki so bila določena pri definiciji. Uporabnik na skupnem mestu dobi vse informacije, na katerih dokumentih mora izvesti ustrezne akcije. O tem je lahko obveščen tudi po elektronski pošti.

Sistem namestništva omogoča, da pooblašteni namestniki lahko opravijo ustrezne akcije v določenih procesih. Namestništvo si lahko uredi vsak, tako da določi nadomestno osebo za potrjevalca na procesu. Torej gre za nadomeščanje na nivoju procesa, ne pa tudi za varnostno shemo, ki se definira na nivoju vrste dokumenta. Namestništvo obsega izključno pravice iz naslova procesa.

Zagotovljena je celotna sled izvajanja procesa, saj se ob vsaki izvedeni akciji na dokument zapiše akcija, ki se je izvedla, potrjevalec in čas izvedbe akcije. Za vsak dokument je vedno na voljo informacija, v katerem statusu in pri katerem uporabniku se nahaja.

Spremljanje procesov

Sistem preko različnih pregledov in poročil omogoča vpogled v dejansko izvajanje procesov, kot so čas trajanja procesa, čas pri posameznih udeležencih v procesu, število procesov, Sistem z določenimi pravili omogoča odkrivanje ozkih grl v procesu (npr. računi, ki se odobravajo več kot 10 dni v procesu).

5 VPELJAVA PROCESA POTRJEVANJA KREDITNIH MAP V DOKUMENTNI SISTEM V BANKI VOLKSBANK D.D. LJUBLJANA

Banka Volksbank d.d. Ljubljana se je odločila za vpeljavo elektronskega potrjevanja kreditnih map na podlagi pozitivnih izkušenj z obstoječim dokumentnim sistemom, saj ima v svojem delovnem okolju že nekaj vpeljanih poslovnih procesov (vhodna pošta, izhodna pošta, odobravanje limitov in kartic fizičnim osebam, ...). S tem bo banki omogočena večja produktivnost in hitrost izvajanja poslovnega procesa, predvsem pa olajšano iskanje in arhiviranje kreditne dokumentacije. Z vpeljavo povečamo tudi nadzor nad dokumentom v procesu oziroma sledljivost dokumenta skozi celoten poslovni proces.

Stanje procesa potrjevanja kreditnih map pred uvedbo elektronskega potrjevanja ni primerljivo z učinkovitostjo elektronskega poslovanja. Ravno zaradi velike količine papirne dokumentacije, ki se podvaja ali po nepotrebnem večkrat razmnožuje, podjetja želijo uvesti poslovne procese v elektronski obliki. Ni pa to edini razlog, saj je pri pripravi predlogov brez programske opreme večja možnost napak in nepravilnih stanj v procesu. Kreditna dokumentacija tako vedno potuje po banki od pisarne do pisarne, s čemer se poveča možnost izgube dokumentacije kredita, manjkajoča dokumentacija se potem ugotovi šele pri pripravi pogodbe. Obstaja tudi več možnosti za napake v sami pogodbi in pri najavi izplačil zaradi ročnega vnosa podatkov. Vse to poveča nepreglednost poslovnih procesov, saj ne vemo kje v procesu se kreditna dokumentacija nahaja.

Z željo po vpeljavi poslovnega procesa potrjevanja kreditnih map v dokumentni sistem so se pojavile različne zahteve, ki naj bi prinesle izboljšanje učinkovitosti poslovanja na področju kreditiranja:

- izboljšanje nadzora nad dokumenti,
- pohiترitev postopka potrjevanja kreditov,
- boljša preglednost nad poslovnimi procesi,
- hitrejše iskanje posameznih dokumentov kredita,
- manj možnosti za napake pri vpisovanju podatkov,
- manjše število ročnih evidenc in
- določitev strukture celotne organizacije podjetja

Do zahtev je prišlo predvsem zaradi pozitivnih izkušenj z elektronskimi poslovnimi procesi potrjevanja računov ter odobravanja limitov in kartic fizičnih oseb, ki se že nekaj časa uporabljajo v delovnem okolju podjetja.

Vsak elektronski poslovni proces se uvede v dokumentni sistem zaradi prednosti, ki jih sistem prinaša:

- zajem podatkov v kreditni predlog brez neposredne uporabe transakcijskega sistema HIBIS¹⁹
- enostaven pregled, kje v procesu se kreditni predlog nahaja,
- ob kreiranju novega predloga je na voljo že vsa obstoječa dokumentacija,
- enostavno predajanje odprtih poslov v primeru odsotnosti ali zamenjave sodelavca,
- graditev mape o stranki in vodenje zabeležk,
- nepotrebnost (fizičnega) pregledovanja arhivirane dokumentacije,
- kontrola podatkov o obstoječem poslovanju stranke z banko ni potrebna (črpanje podatkov iz transakcijskega sistema HIBIS),
- manj napak pri pripravi kreditnih predlogov (delni zajem podatkov iz HIBIS-a),
- možnost spremljave poslovnega procesa kreditnih map (sledljivost korakov),
- razpoložljivost celotne kreditne dokumentacije,
- možnost priprave poročil (npr. seznam vseh odobrenih kreditov),
- hitrejša priprava kreditnih map (zajem podatkov iz sistema HIBIS in iz kreditnega predloga)
- kontrola nad vsebino kreditne mape
- avtomatizacija opozoril (npr. opozorilo o približevanju zapadlosti kredita),
- enostavno dodajanje novih dokumentov,
- elektronska podpora izvajanja nakazil kreditov in
- izposoja kreditne dokumentacije ni potrebna

S prednostmi, ki jih prinaša elektronsko potrjevanje kreditnih map, podjetje ustvari nadzor in arhiviranje dokumentacije kredita. Iskanje dokumentacije kredita postane enostavno in časovno nepotratno. Posledično se znižajo tudi stroški iskanja predvsem pa stroški arhiviranja.

¹⁹ HIBIS = Računalniška programska oprema za podporo bančnemu informacijskemu sistemu (transakcijski sistem).

5.1 Opis sistema potrjevanja kreditnih map

Potrjevanje kreditnih map je dodano v dokumentni sistem z že obstoječimi poslovnimi procesi (vhodna pošta, potrditev vhodnih računov, odobritev limita, ...). Uporablja se v procesu priprave in potrjevanja kreditnih map. Povezan je z IMiS/Scan produktom, ki omogoča skeniranje dokumentov. Dokumentni sistem omogoča povezavo s transakcijskim sistemom HIBIS preko spletnih storitev (ang. Web services).

Z vpeljavo kreditnih map v dokumentni sistem je prišlo tudi do razlikovanja med konceptom dokumenta in konceptom posla. Kot primer koncepta dokumenta bi navedel potrjevanje računa. Tu imamo samo en dokument, ki ga spremljamo skozi celoten proces in ni treba paziti na druge dokumente.

Pri konceptu posla pa je potrebno spremljati več dokumentov, ki so povezani med seboj, lahko tudi odvisni eden od drugega. Nekateri dokumenti posla morajo biti v določenem statusu v procesu, se pravi brez ustreznosti teh dokumentov se proces ne more nadaljevati. Koncept posla je lahko kar kreditna mapa, ki vsebuje več različnih dokumentov (statusni dokumenti, bilančni dokumenti, zavarovanja, nakazila). Dokumenti so povezani s kreditno mapo, zato si na vse te dokumente zapišemo številko kreditne mape, da lahko vzpostavim povezavo med njimi.

5.1.1 Kratek opis procesa potrjevanja kreditnih map

Kredit lahko sklepa pravna ali fizična oseba. Oseba v pogovoru z komercialistom izbere vrsto kredita, ki bi ga rada sklenila. Komercialist na podlagi vrste kredita zahteva določeno dokumentacijo, ki jo mora pridobiti pravna oz. fizična oseba.

Komercialist ima možnost podati **ponudbo** stranki, ki je namenjena predvsem zato, da se stranka lažje odloči in premisli preden sklene pogodbo za najem kredita. Zaenkrat je možno kreirati ponudbe samo za podjetja. Če stranka sprejme ponudbo, potem se iz ponudbe lahko ustvari **kreditni predlog**.

Kreditni predlog pa lahko kreiramo kar brez ponudbe in predstavlja pravzaprav proces zbiranja in urejanja dokumentacije. Vsa ta dokumentacija je potrebna za sklenitev kredita. Ko je pridobljena, potrjena in pregledana vsa potrebna dokumentacija, se iz kreditnega predloga tvori **kreditna mapa**. Ko je kreditna mapa arhivirana, se kreira **nakazilo** za kredit, ki je bil potrjen in odobren s strani vseh potrjevalcev (odgovornih oseb).

5.1.2 Podrobnejši opis sistema in kreiranja kreditov

Mapa komitenta je središče, kjer vidimo vse podatke o komitentu, ki je lahko fizična ali pravna oseba. Namenjena je predvsem pregledu vseh dokumentov in kreditnih map, ki so vezani na komitenta ter za ustvarjanje novih kreditnih map ali drugih dokumentov z enega mesta.

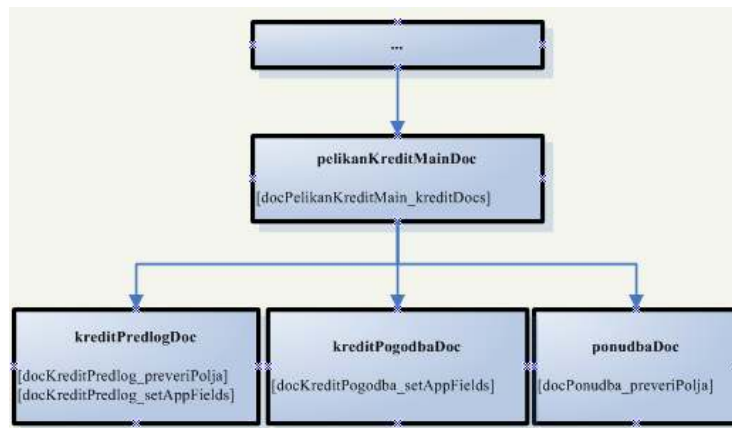
Najprej je bilo potrebno narediti tabelo z zavihki, kjer so prikazane različne vrste dokumentov komitenta. Dokumenti, ki so prikazani v pregledih, so v različnih zbirkah. Statusni in bilančni dokumenti se ustvarjajo in shranjujejo v svoji zbirki, prav tako so tudi dokumenti kredita (splošni dokumenti, nakazila, zavarovanja) v posebni zbirki. Ti dokumenti so v posebnih zbirkah zaradi razbremenitve glavne zbirke, ki bi postala sčasoma prevelika, s tem pa bi bila odzivnost in hitrost izvedbe različnih akcij občutno počasnejša. Največji problem pri razdelitvi glavne baze na več različnih je bil v izvajanju procesov v ostalih bazah. Če noben dokument ne bi bil vezan na proces, ne bi bilo težav. V našem primeru pa imamo kar nekaj takih dokumentov, ki so vezani na proces (bilančni dokument; analiza poslovanja in dokumenti kredita; vsa nakazila). Za ta način razdelitve dokumentov po različnih bazah smo se odločili predvsem na podlagi izkušenj iz prejšnjih projektov. Ta problem smo rešili s sodelovanjem s podjetjem 3K IT d.o.o., ki je s popravki dokumentnega sistema 3K Document Cycle podprla izvajanje procesa tudi v drugih zbirkah.

Vsi pregledi, ki jih prikazujemo v tabeli z zavihki v mapi komitenta, so kategorizirani po šifri komitenta. Ko dodamo vpeti pregled v tabelo, mu določimo, po kateri vrednosti naj se pregled kategorizira (ang. show single category). Ponavadi je to kar vrednost nekega polja, v našem primeru vpetih pregledov v mapi komitenta se vsi kategorizirajo po šifri komitenta.

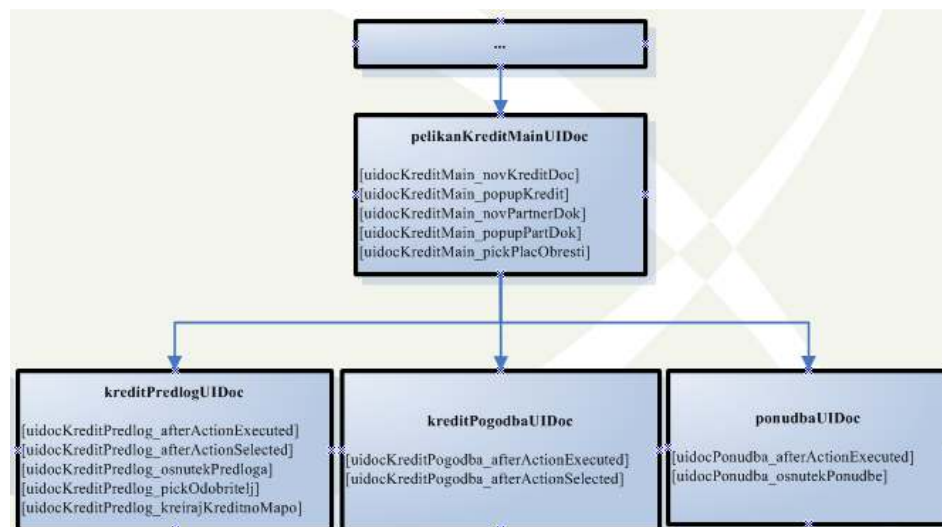
Za vsako ustvarjanje novega dokumenta potrebujemo nek glavni obrazec, ki se ne spreminja. Na njem so definirana polja za osnovne podatke o komitentu, skrita polja kot so avtorska in bralska polja kamor se zapišejo avtorji in bralci dokumenta (potrebujemo jih zaradi varnosti dostopa do dokumenta). Na tem obrazcu so definirane tudi akcija za shranjevanje, urejevanje, proženje procesnih akcij (procesna akcija je vidna samo, če je dokument vezan na proces) dokumenta, izračunani podobrazci kamor se zapisuje zgodovina dokumenta in trenutno stanje procesa ter izračunan podobrazec, ki prikazuje podatke na podlagi vrste dokumenta.

5.1.2.1 Struktura/Hierarhija razredov – Objektni model

V Lotus Domino objektnem modelu razrede razdelimo v dve skupini: **back-end razredi** (razredi, ki se uporabljajo za delo z dokumenti oziroma podatkovnim modelom) in **front-end razredi** (razredi, ki se uporabljajo za delo z uporabniškim vmesnikom Notes odjemalca). Na slikah 14 in 15 je predstavljena struktura na novo definiranih razredov in funkcij v glavni bazi dokumentnega sistema.



Slika 14: Shema back-end razredov in funkcij



Slika 15: Shema front-end razredov in funkcij

5.1.2.2 Ponudba

Kot smo opisali že v poglavju 5.1.1, komercialist lahko najprej pripravi ponudbo. Za ponudbo je bilo potrebno narediti nov podobrazec (ang. subform), kjer so definirana polja za vse podatke o ponudbi. Na podobrazcu za ponudbo imamo tudi akciji »Osnutek« in »Kreditni predlog«.

Ko se ustvari ponudba, se nanjo avtomatsko vpiše številka komitenta in na podlagi te številke kategoriziramo pregled po komitentu. Pri vnosu podatkov v ponudbo se preverijo vsa polja (funkcija »preveriPolja«), da niso prazna oziroma so v pravem formatu. Če polje ni pravilno izpolnjeno, potem s funkcijo `errorGotoField` vrnemo napako in postavimo kurzor v polje, kjer so napačni podatki.

Z že vgrajeno funkcionalnostjo v dokumentnem sistemu smo si pomagali ustvariti osnutek ponudbe. Polja iz podobrazca lahko preslikamo v .doc format (MS Word dokument). Funkciji za preslikavo podamo kot parametre Notes dokument (v našem primeru dokument ponudbe) in šifro integratorja²⁰, ki smo jo določili že v nastavitvah dokumenta ponudbe. Šifra integratorja se avtomatsko zapiše na dokument ponudbe ob nastanku dokumenta.

Za pregledovanje dokumentov ponudb ustvarimo kategoriziran pregled, ki nam prikaže ponudbe samo za določenega komitenta. V vsakem ustvarjenem pregledu moramo napisati izbirno formulo (ang. selection formula), ki prikaže samo ponudbe.

Iz sprejete ponudbe lahko preko akcije »Kreditni predlog« (akcija je vidna samo, če se ponudba nahaja v statusu »Sprejeta«) ustvarimo kreditni predlog, kamor se zapišejo tudi nekateri podatki iz ponudbe.

5.1.2.3 Kreditni predlog

Za kreditni predlog smo naredili nov podobrazec, kjer so definirana vsa potrebna polja za prikaz podatkov, skrita polja in akcija za ustvarjanje osnutka kreditnega predloga (podobna akcija kot pri ponudbi).

Kreditne predloge prikazujemo v zavihku tabele »Kreditni predlogi« v mapi komitenta. V tem zavihku je pripet vpeti pregled, kjer so prikazani vsi kreditni predlogi komitenta. Na tem pregledu je dodana akcija, s katero ustvarimo nov kreditni predlog ter akcija v skupni rabi »Pregled«, ki nam prikaže objekt kreditnega predloga, če seveda obstaja.

²⁰ Integrator = nastavitveni dokument kjer imamo definirane preslikave med Lotus Notes polji in MS Office Word polji ter prilepljen MS Office Word dokument.

Kreditni predlog ustvarimo iz sprejete ponudbe ali iz mape komitenta. Ob prvi shranitvi kreditnega predloga, ki je bil narejen iz ponudbe, se ustvari povezava med njima. Nastaneta povezovalna dokumenta (ang. link document), ki povezujeta kreditni predlog in ponudbo (povezava ponudba → kreditni predlog in kreditni predlog → ponudba). Oba dokumenta se prikažeta v pregledu v zavihku tabele »Povezani dokumenti«. Povezovalni dokumenti se shranjujejo v posebni bazi.

Vrsto kreditnega predloga izberemo iz menija, ki se dinamično kreira, odvisno od števila vrst kreditov katere so definirane v nastavitveni bazi. Ravno zato je zelo pomembno, kako sestavimo strukturo dokumentov, saj potem nimamo problemov pri izbiri določene vrste dokumentov iz strukture. Ko se dokument kreditnega predloga prvič shrani, se avtomatsko nanj povežejo vsi obstoječi statusni in bilančni dokumenti, ki so v mapi komitenta. Bilančni dokumenti se ne povežejo vsi, ampak samo trenutno aktualni (npr. dokument »Bilanca stanja in uspeha« je aktualen samo za obdobje enega leta). Povezava se vzpostavi tako, da se na vse statusne in bilančne dokumente v večvrednostno polje vpiše številka kreditnega predloga. Tako vemo kateri dokumenti spadajo v kreditni predlog. To nam pomaga pri prikazovanju statusnih in bilančnih dokumentov v vpetem pregledu v kreditnem predlogu. Če pa dodamo bilančne dokumente pozneje v mapo komitenta, ti dokumenti ne bodo vidni v zavihku tabele »Bilančni dokumenti« v pregledu kreditnega predloga, dokler ne osvežimo tega pregleda z akcijo »Osveži«. Ob kliku na akcijo se v vse nove bilančne dokumente, ki so označeni kot aktualni, zapiše številka kreditnega predloga.

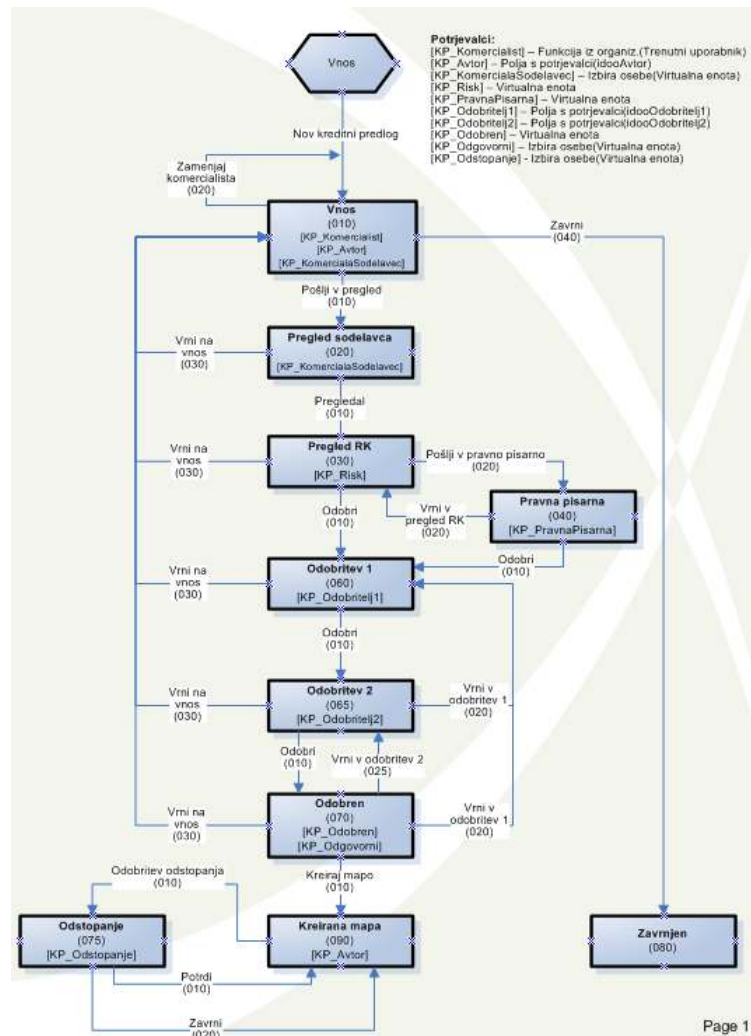
Pri sestavi pregleda najprej zapišemo izbirno formulo za pregled bilančnih dokumentov v kreditnem predlogu, ki izgleda takole:

```
SELECT form = "dok" & verzija != 0 & idPartner != "" & aktualno = "1" & @Matches(@Left(idVrsta; 3); "D{12}2")
```

Potem definiramo stolpce in polja, ki naj se v stolpcih prikažejo. V nastavitvah določimo še kategorizacijo pregleda. V tem primeru je kategorija kar številka kreditnega predloga, saj so bilančni dokumenti vezani nanj.

Iz kreditnega predloga se avtomatsko ustvari kreditna mapa, ko v statusu »Odobren« sprožimo akcijo »Kreiraj mapo«. Na kreditno mapo se zapišejo tudi nekateri podatki iz kreditnega predloga. Preveri se, če je v kreditnem predlogu že določen porok ter ga zapiše v skrito polje na kreditno mapo.

Na sliki 16 je lepo razvidno skozi katere statusne potrjevanja mora iti dokument kreditni predlog, preden se iz njega avtomatsko ustvari kreditna mapa.



Slika 16: Shema procesa kreditnega predloga

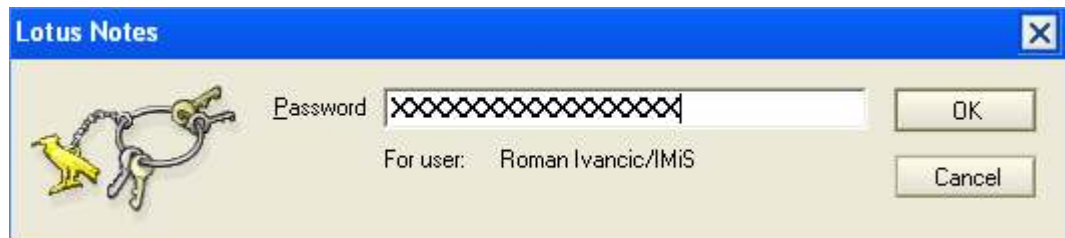
5.1.2.4 Kreditna mapa

Posebnost v kreditni mapi so obveznosti nekaterih dokumentov. To pomeni, da je dokument v nekem statusu obvezen za pregled. Programsko se preveri, če ima kreditni dokument v določenem statusu poskeniran objekt, sicer vrne napako oziroma opozorilo. Obveznost določimo že v nastavitvah dokumenta za katerikoli status v procesu. Obveznost preverjamo tako, da naredimo pregled kjer so prikazani vsi dokumenti, ki so obvezni v določenem statusu. Nato se programsko sprehodimo čez celoten pregled in preverimo obveznost vseh dokumentov. Če obvezni dokumenti nimajo poskeniranega objekta potem vrnemo opozorilo kjer se izpiše spisek vseh teh dokumentov.

5.1.3 Uporaba elektronskega potrjevanja kreditnih map »v praksi«

5.1.3.1 Prijava v dokumentni sistem preko Lotus Notes odjemalca

Prijava (slika 17) je možna le z veljavnim uporabniškim geslom, ki ga dodeli skrbnik sistema in ID značko, ki jo ima vsak uporabnik.



Slika 17: Prijava v dokumentni sistem

5.1.3.2 Mapa komitenta

V sklopu šifranta partnerjev/komitentov se dokument s podatki komitenta razširi v mapo komitenta (slika 18). V njej so na enem mestu zbrane povezave do zabeležk o partnerju, statusne in bilančne dokumentacije, kreditnih predlogov, kreditnih map, limitov, kartic ter poštno korespondence. Vse te povezave so vzpostavljene z vpetimi pregledi, ki so dodani v tabelo na mapi komitenta.

V mapi komitenta imamo statusne in bilančne dokumente, ki se nahajajo v posebni bazi, ponudbe, kreditni predlogi in kreditne mape pa se nahajajo v glavni bazi.

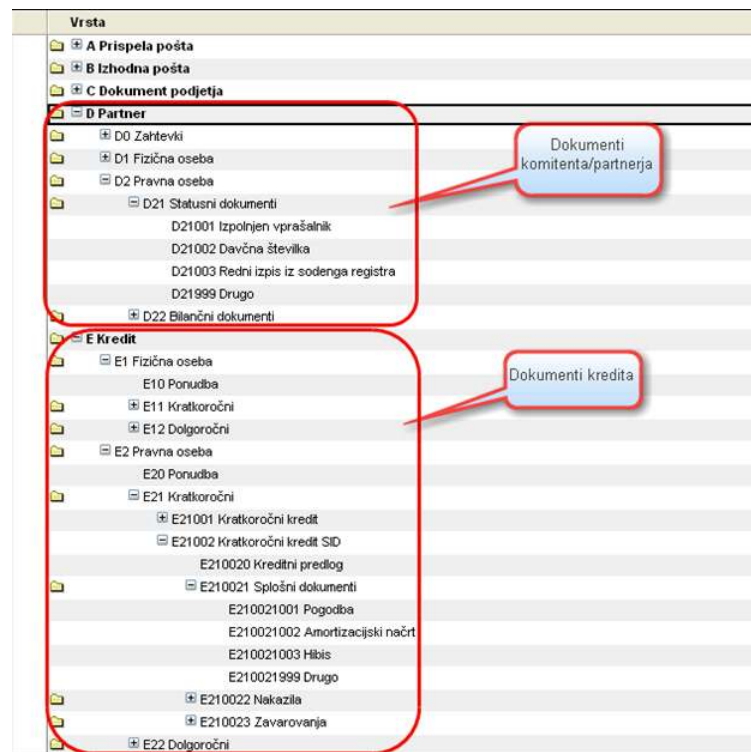
| Status | Številka | Tip kredita | Ime kredita | Namen | Znesek (Val.) | Obrestna mera | Potrjevalci |
|---------------------|-------------|-------------|------------------------|-------------------------|---------------|---------------|---------------|
| Vnos | 000840/2009 | Dolgoročni | Dolgoročni kredit | testiranje integratorja | 2.300,00 EUR | 5,1 | Roman Ivancic |
| Ⓜ Pregled sodelavca | 000802/2009 | Dolgoročni | Dolgoročni kredit | testiranje integratorja | 2.300,00 EUR | 5,1 | Alma CORALIC |
| Vnos | 000800/2009 | Dolgoročni | Dolgoročni kredit | | 0,00 EUR | | Roman Ivancic |
| Ⓜ Kreirana mapa | 000788/2009 | Kratkoročni | Kratkoročni kredit | test1 | 22.000,00 EUR | 5,1 | Roman Ivancic |
| Ⓜ Kreirana mapa | 000739/2009 | Kratkoročni | Kratkoročni kredit SID | test1 | 20.000,00 EUR | 5,1 | Roman Ivancic |

Slika 18: Mapa komitenta

Pri izdelavi izpostavljenosti (akcija »Oblig«) sistem sam prepozna ali gre za pravno ali fizično osebo. Na obrazcu se nam prikažejo vsi obstoječi krediti, kartice in limiti komitenta. Te podatke pridobimo iz transakcijskega sistema preko shranjenih procedur (ang. stored procedure). Izpostavljenost potem lahko uvozimo v mapo komitenta v bilančno dokumentacijo, kjer se ustvari dokument »Izpostavljenost«. Na dokument »Izpostavljenost« se pripne objekt v formatu .tif, kjer je prikazana izpostavljenost komitenta.

5.1.3.3 Struktura/hierarhija dokumentov

V dokumentnem sistemu je zelo pomembna drevesna porazdelitev dokumentov (glej sliko 19). Zato se zmeraj ob sestavljanju strukture dokumentov pojavi dilema, kako najboljše sestaviti strukturo, da ne bi imeli težav pri kodiranju, določanju izbirnih formul (ang. selection formula) v pregledih, skrivanju akcij za določene vrste dokumentov na obrazcih in podobno.



Slika 19: Struktura dokumentov komitenta in dokumentov kredita

Dokumenti, ki so vezani na komitenta (pravna ali fizična oseba), so razdeljeni na statusne in bilančne, krediti pa so v osnovi razdeljeni na pravne in fizične osebe. Vsak kredit ima potem svojo definicijo in vrsto kreditnega predloga ter kreditno dokumentacijo.

Vsakemu dokumentu v strukturi lahko določimo tudi varnostno shemo ali pa pustimo privzete nastavitve, da imajo vsi uporabniki pravico za izračun avtorjev, za izračun bralcev in za vnos podatkov (slika 20).

| Pravila za avtorje | | Pravila za bralce | | Pravice za vnos | | |
|------------------------------------|-------------|-----------------------------|---|-----------------|-----------------|---------------------------|
| Pravila za izračun avtorjev | | | | | | |
| Vsi uporabniki | | <input type="checkbox"/> Da | | | | |
| Formula za vse uporabnike | | | | | | |
| Zap.št. | Oseba Enota | Polje/Vrednost | Ime | Vloge v enoti | Nadrejene enote | Vloge v nadrejenih enotah |
| 1. | Oseba | Polje v Pelikanu | Prejemniki | | Nimajo pravic | |
| 2. | Enota | Konstanta | Kontroling in poročanje, Računovodstvo, Računovodstvo in kontroling, Uprava, Uprava - Poslovanje I, Uprava - Poslovanje II, Uprava - Tveganje | Vodja, član | Nimajo pravic | |
| 3. | Oseba | Polje v Pelikanu | ProcesZgoOdobri | | Nimajo pravic | |

Slika 20: Nastavljanje varnostne sheme nad dokumentom

5.1.3.4 Statusna in bilančna dokumentacija

Statusna in bilančna dokumentacija se hranita le na enem mestu in sicer v mapi komitenta. Ob vsakem poslovnem dogodku, ki povzroči kreiranje kreditnega predloga ali kreditne mape, se statusna in bilančna dokumentacija povežeta na kreditni predlog ali kreditno mapo. Bilančni dokument se poveže na kreditni predlog ali kreditno mapo samo v primeru, ko je označen kot aktualen (slika 21).

Gumb "Pregled" prikaže objekt, ki ga vsebuje dokument(.xls, .doc, .pdf, .tiff)

Določimo aktualnost ali neaktualnost dokumenta

Ta ikona pomeni, da je dokument še v procesu potrjevanja

V tem stolpcu se prikaže ikona, če dokument vsebuje kakšen objekt

Ikoni prikazujeta aktualnost oz. neaktualnost bilančnega dokumenta

| | Številka | Vrsta | Datum vnosa | Opis |
|---|-------------|--------------------------|-------------|---|
| ☒ | 000852/2009 | Bilanca stanja in uspeha | 29.05.2009 | test uvrsti |
| ☒ | 000837/2009 | Izpostavljenost | 14.05.2009 | Kreiranje izpostavljenosti iz obliga. |
| ☒ | 000787/2009 | Analiza poslovanja | 07.05.2009 | test1 |
| ☒ | 000718/2009 | Izpostavljenost | 23.04.2009 | Kreiranje izpostavljenosti ob uvozu v mapo. |

Slika 21: Statusna in bilančna dokumentacija

Za podjetje je dokument »Bilanca stanja in uspeha« aktualen samo za obdobje enega leta, naslednje leto ima podjetje novo bilanco stanja in uspeha. V tem primeru postane bilanca preteklega leta neaktualna.

5.1.3.5 Zabeležke v mapi komitenta

Funkcionalnost zabeležke je enakovredna komentarjem na posameznem dokumentu v dokumentnem sistemu. Zabeleži se avtor in čas zabeležke ter seveda komentar, kot je prikazano na sliki 22. Z zabeležko si pridobimo predvsem znanje o komitentu.

| Datum in čas vnosa | Vnašalec | Komentar |
|--------------------|---------------|--|
| 29.05.2009 21:46 | Roman Ivancic | Stranka želi financiranje poslovne stavbe v vrednosti 500.000 €. |
| 05.02.2009 10:00 | Roman Ivancic | abcdef |
| 09.01.2009 09:18 | Roman Ivancic | pa še en |

Slika 22: Zabeležka v mapi komitenta

5.1.3.6 Priprava ponudbe

Možnost kreiranja ponudbe je trenutno namenjena samo pravnim osebam. Ko komercialist izpolni obrazec za ponudbo (glej sliko 23), najprej izdelava osnutek ponudbe s klikom na gumb »Osnutek« in ga pošlje podjetju po elektronski pošti. Če podjetje sprejme ponudbo, potem komercialist lahko direktno iz ponudbe ustvari kreditni predlog. Gumb za ustvarjanje kreditnega predloga se prikaže samo ob sprejetju ponudbe.

Ponudba 000798/2009
07.05.2009 15:05
Vnesel: Roman Ivancic
Verzija št. 1

Osnovni podatki Šifra KMT: 4206 **Prejemniki**

| | | | |
|-----------------|---------------------------|-------|--|
| Podjetje | | Osebe | |
| Naslov | | Enote | |
| Kraj | | | |
| Vrsta dokumenta | KreditPravna osebaPonudba | | |

Podatki o ponudbi

| | | | |
|-------------------------------------|-------------------------------------|---|--|
| Namen | Nakup stanovanja | Vrsta kredita | Dolgoročni kredit |
| Črpanje | črpanje | Znesek | 12.000,00 EUR |
| Datum črpanja | do 28.05.2009 | Doba kreditiranja | 12 mesecev |
| Obrestna mera | 1 2% | Plačevanje obresti | Mesečno |
| Odplačilo glavnice | odplačilo | Nadomestilo za pripravo kreditnega predloga | 0,10 % od zneska odobrenega kredita |
| Nadomestilo za rezervacijo sredstev | 0,30 % od zneska odobrenega kredita | Nadomestilo za nečrpan del | 0,20 % od zneska nečrpanega dela kredita |
| Strošek vođenja kredita | 0,10 % od zneska odobrenega kredita | Nadomestilo za pregled dokumentacije | 100,00 EUR |

Povezani dokumenti

| Številka | Podjetje | Vrsta | Opis | Avtor povezave | Datum povezave |
|-------------|----------|------------------|--------------------------------------|----------------|----------------|
| 000862/2009 | | Kreditni predlog | Kreditni predlog - Dolgoročni kredit | Roman Ivancic | 02.06.2009 |

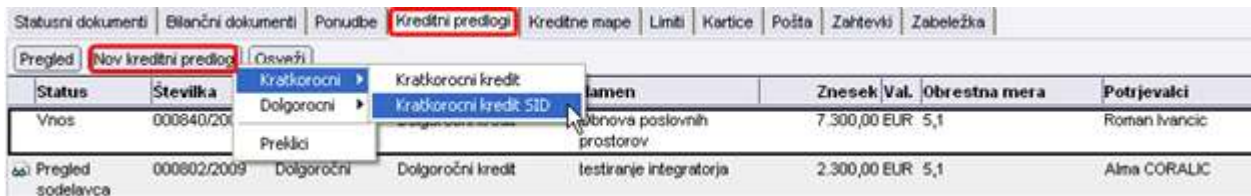
Slika 23: Ponudba

Ob kreiranju kreditnega predloga iz ponudbe se dokumenta avtomatsko povežeta med seboj tako, da se ustvari povezovalen dokument. Prikaže pa se nam v tabeli »Povezani dokumenti«.

5.1.3.7 Kreditni predlog (priprava dokumentacije komitenta)

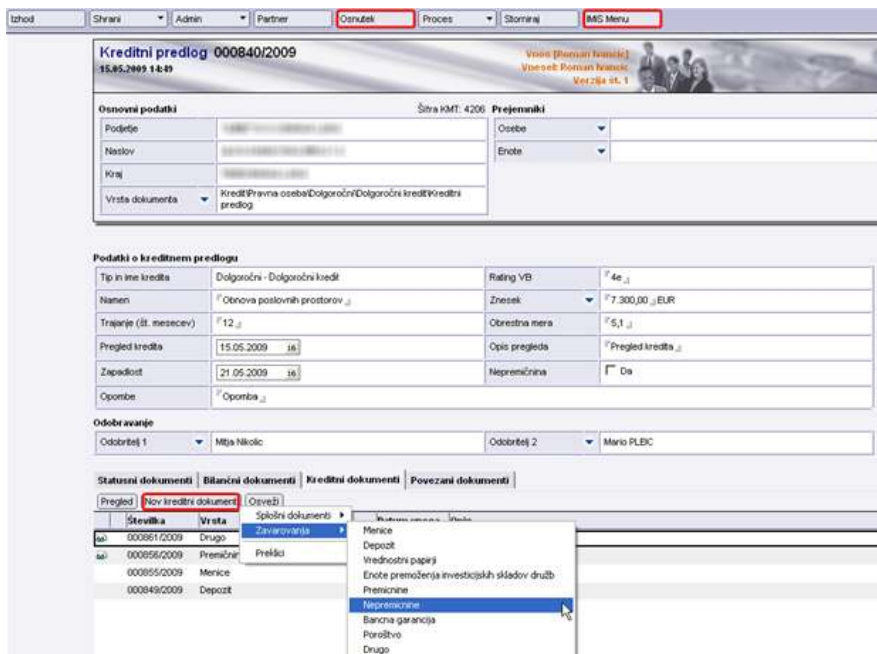
Kot je razvidno iz slike 16 ima vsaka vrsta kredita svoj kreditni predlog. To pomeni, da za vsak kreditni predlog lahko definiramo svojo vnosno masko in svoj proces. Kreditni predlogi posameznih vrst kreditov so tako popolnoma neodvisni, kar je zelo pomembno za fleksibilnost sistema.

Do kreditnih predlogov enostavno dostopamo prek zavihka Kreditni predlogi v vsaki mapi komitenta (glej sliko 24). Prav tako je enostavno tudi ustvarjanje novega kreditnega predloga. Lahko ga ustvarimo preko akcije na pregledu »Nov kreditni predlog« ali pa iz ponudbe, če je bila ta sprejeta s strani kreditojemalca.



Slika 24: Ustvarjanje novega kreditnega predloga v mapi komitenta

Vsak kreditni predlog ima svojo vnosno masko (glej sliko 25) in svoj proces odobritve glede na vrsto kredita. Glede na vnosno masko lahko za vsak kreditni predlog definiramo svoj obrazec, saj komercialist s pomočjo gumba »Osnutek« ustvari privzeti izgled kreditnega predloga glede na vnešene podatke. S tem dosežemo enotnost izgleda kreditnih predlogov.



Slika 25: Kreditni predlog

Komercialist ima v kreditnem predlogu tudi možnost pregleda obstoječih kreditov komitenta, ki sklepa kredit. Če klikne na gumb »Obstoječi krediti«, se mu v tabeli na dokumentu prikažejo vsi obstoječi krediti s podatki o kreditu, kot je razvidno na sliki 26.

| VPO | Partija | Obrok | Nepl. obveznosti | Ilačin odplačila | Saldo kredita | Izpostava |
|---------------------|---------|------------|------------------|------------------|----------------|------------------|
| 123 | 1234567 | 150,04 EUR | 160,06 EUR | | 1.500,00 EUR | 1.500,07 EUR |
| 48 | 3232334 | 120,00 EUR | 150,03 CHF | | 12.000,00 CHF | 8.000.000,00 EUR |
| 153 | 3202302 | 100,03 EUR | 2.000,00 EUR | | 120.000,07 EUR | 15.000,04 EUR |
| 54 | 8912345 | 122,50 EUR | 3.000,00 EUR | | 121.212,05 EUR | 112,02 EUR |

Slika 26: Tabela za prikaz obstoječih kreditov

5.1.3.8 Kreditna mapa

Kreditna mapa nastane ob uspešnem zaključku procesa potrditve kreditnega predloga. Ob nastanku kreditne mape se ustvari povezan dokument, ki ustvari povezavo kreditne mape s kreditnim predlogom (slika 27).

| Statusni dokumenti | | Bilančni dokumenti | | Kreditni dokumenti | | Nakazila | | Povezani dokumenti | |
|--------------------|----------|--------------------|-----------------------|--------------------|--|----------------|----------------|--------------------|--|
| Številka | Podjetje | Vrsta | Opis | | | Avtor povezave | Datum povezave | | |
| 000788/2009 | | Kreditni predlog | 22.000,00 EUR, test11 | | | Roman Ivancic | 07.05.2009 | | |

Slika 27: Povezava kreditnega predloga s kreditno mapo preko povezovalnega dokumenta

Vsaka kreditna mapa ima lahko svojo vnosno masko (glej sliko 28), proces in vzorec pogodbe. Na kreditno mapo lahko dodajamo različne dokumente zavarovanja, splošne dokumente mape ali naloge za črpanje (nakazila).

Prednost takega pristopa je, da lahko na vsakem od teh dokumentov definiramo svoj proces ali posebnosti glede meta podatkov. Tako imajo nakazila za vsako vrsto kredita definiran proces odobritve pred plačilom. Bistvena prednost je v tem, da ima vsak potrjevalec nakazila v vsakem trenutku dostop do celotne dokumentacije v kreditni mapi, kreditnem predlogu in mapi komitenta. Enako lahko dostopa tudi do zgodovine odobritve vseh procesov vezanih na kredit. Nakazila lahko ustvarimo šele takrat, ko je kreditna mapa v statusu »Arhivirana«.

| Podatki o kreditni mapi | | | |
|-------------------------|------------|---------------|---------------|
| Namen | test11 | Znesek | 22.000,00 EUR |
| Trajanje (št. mesecev) | 24 | Komercialist | Roman Ivancic |
| Številka pogodbe | 1234-4321 | Obrestna mera | 5,1 |
| Zapadlost | 29.05.2009 | | |

| Statusni dokumenti | | Bilančni dokumenti | | Kreditni dokumenti | | Nakazila | | Povezani dokumenti | |
|--------------------|---------------|--------------------|--|--------------------|------------|-----------|------|--------------------|--|
| Pregled | Novo nakazilo | Osveži | | Status | Datum | Znesek | Opis | | |
| | | | | Vnos | 02.06.2009 | 22.000,00 | test | | |

Slika 28: Kreiranje nakazila v kreditni mapi

Celoten proces potrjevanja kreditne mape poteka od vnosa podatkov o komitentju (pravna, fizična oseba) do zaključka potrjevanja kreditne mape in nakazil sredstev na kreditojemalčev račun. Iz uporabniškega stališča se proces zelo poenostavi, saj ni več potrebno prenašati kreditne dokumentacije z oddelka na oddelek.

Glede na izkušnje prejšnjih projektov, uporabniki ne potrebujejo veliko časa za uvajanje na nov sistem dela, saj so v večini primerov zelo motivirani in bi radi čim prej uvedli elektronsko potrjevanje.

6 ZAKLJUČEK

V Banki Volksbank d.d. Ljubljana skušajo povečati poslovno uspešnost in učinkovitost poslovanja. Del teh procesov predstavlja tudi potrjevanje kreditnih map, ki je prikazano v okviru tega diplomskega dela. Razvoj programske rešitve je potekal s pomočjo prototipov, ki so pripeljali do končne rešitve.

Z vpeljavo elektronskega potrjevanja smo poenostavili delovni proces potrjevanja kreditnih map. Programska rešitev trenutno še ni v uporabi v delovnem okolju, vseeno pa lahko rečemo, da so prednosti novega načina dela pričakovane in se v celoti uresničujejo. Poleg lažjega iskanja dokumentov in urejenega elektronskega arhiva kreditnih map in njene dokumentacije, je izpostavljena še prednost nadzora nad celotnim potekom potrjevanja kreditnih map. Lotus Notes/Domino je že v osnovi varno okolje, ki ga z uporabo različnih možnosti lahko še nadgradimo. To smo dosegli z obvezno sestavo organizacijske sheme družbe, ki je sestavljena iz vseh oddelkov kateri sodelujejo pri potrjevanju kreditov. Izboljšali smo zajem podatkov v različne dokumente brez neposredne uporabe transakcijskega sistema. Z enkratnim zajemom podatkov iz transakcijskega sistema praktično ni potrebna kontrola obstoječih podatkov o poslovanju stranke. S tem smo pridobili veliko dragocenega časa, ki ga komercialist v nasprotnem primeru porabi za preverjanje podatkov.

Porajajo pa se že ideje po izboljšavah. Z večanjem števila dokumentov v zbirki se slabšajo odzivni časi. Vzrok je indeksiranje kazal v zbirkah, ki zaseda precej pomnilniškega prostora, zato bo čez čas potrebno dokumente, ki niso povezani s tekočimi dogodki, umikati v arhivske zbirke. V nadaljevanju razvoja te programske rešitve bi bilo potrebno nakazilo oziroma izplačilo kredita povezati s transakcijskim sistemom v sodelovanju s podjetjem, ki je razvilo ta sistem.

VIRI

- [1] Avedon Don M.: Introduction to Electronic Imaging. AIIM Catalog No. C125, 1995.
- [2] Benz Brian, Oliver Rocky and Schwartz Richard: Lotus Notes and Domino 6 Programming Bible, USA, 2003.
- [3] Bregar A., B. Lahovnik, R. T. Leskovar, V. Lešnik: Priročnik za uporabo okolja Lotus Notes, Maribor: Inštitut za informatiko, 2000
- [4] Koulopoulos Thomas, Frappaolo Carl: Electronic Document Management Systems: a portable consultant. McGraw-Hill, 1995.
- [5] Lazar Janez: Domino sedmica: sistemska administracija Domino, Ljubljana: SRC.SI, 2007
- [6] Mantelman Lee: Mantelman's Imaging buyers Guide. Flatiron Publishing, 1995.
- [7] Žerko Bine. (2003). Imaging & archiving v teoriji in praksi. Dostopno na: <http://www.imis.si/Web/IMiSSLO.nsf/e49e2d9e0dcaba38c1256b1200365294/ee36251e0c355962c1256d270029afa4?OpenDocument>
- [8] Interna dokumentacija podjetja IS d.o.o., Ljubljana, 2009
- [9] (2005) Koncept dokumentnih sistemov. Dostopno na: http://www.mju.gov.si/fileadmin/mju.gov.si/pageuploads/sodobnajavnauprava/ppt/smaragdna_B/OGRINC.ppt
- [10] (December, 2005) Application Development with Lotus Domino Designer 7. IBM Corp. Dostopno na: <http://www.ibm.com/developerworks/lotus/documentation/dominodesigner/70x.html>