

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matej Pintar

**INTEGRACIJA ELEKTRONSKEGA DOKUMENTNEGA
SISTEMA Z OSTALIMI SISTEMI V PODJETJU**

MAGISTRSKO DELO

Mentor: prof. dr. Denis Trček

Ljubljana, 2016



Št.: 149-MAG-ISO/2016

Datum: 29. 02. 2016

Matej PINTAR, univ. dipl. inž. rač. in inf.

L j u b l j a n a

Fakulteta za računalništvo in informatiko Univerze v Ljubljani izdaja naslednjo magistrsko nalogo

Naslov naloge: **Integracija elektronskega dokumentnega sistema z ostalimi sistemi v podjetju**

Integration of electronic document management system with other systems in a company

Tematika naloge:

Večje ko je podjetje, večje je število dokumentov s katerimi se vsakodnevno srečuje. Prav zato je za uspešno poslovanje podjetja izjemno pomembno učinkovito upravljanje s temi dokumenti.

V magistrskem delu predstavite konkreten model integracije elektronskega dokumentnega sistema z ostalimi informacijskimi sistemi v podjetju. Natančneje, predstavite elektronski dokumentni sistem EBA DMS in opišite, katere načine integracije sistem ponuja. V nadaljevanju predstavite ostale sisteme v podjetju in podajte konkretne načine integracije le teh z dokumentnim sistemom EBA DMS. V zadnjem delu napravite načrt za nadaljnji razvoj in izboljšanje sistema za poslovanje z elektronskimi dokumenti.

Mentor:

prof. dr. Denis Trček



Dekan:

prof. dr. Nikolaj Zimic

Rezultati magistrskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov magistrskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Zahvala

Zahvaljujem se prof. dr. Denisu Trčku za mentorstvo pri izdelavi magistrskega dela.

Rad bi se zahvalil tudi vsem domačim, ki so mi skozi vsa leta šolanja stali ob strani in me podpirali. Prav posebna zahvala gre moji puncu Poloni, ki mi je stala ob strani in me neprestano spodbujala pri končanju študija in izdelavi magistrskega dela.

Vsem skupaj še enkrat hvala za soustvarjanje okolja, v katerem je bila izdelava magistrskega dela mogoča.

Kazalo vsebine

Povzetek	1
Abstract	3
1 Uvod.....	5
2 Integracija računalniških sistemov	9
2.1 Poslovno-informacijska arhitektura	9
2.1.1 Stopnje poslovno-informacijske arhitekture	10
2.1.1.1 Poslovni silosi.....	10
2.1.1.2 Poenotenje tehnologij	11
2.1.1.3 Optimizacija jedra podjetja	12
2.1.1.4 Poslovna modularnost.....	12
2.1.2 Primerjava stopenj poslovno-informacijske arhitekture	13
2.2 Integracija aplikacij	13
2.2.1 Integracijska arhitektura.....	14
2.2.1.1 Integracija od točke do točke	15
2.2.1.2 Integracija strežnik–odjemalec.....	16
2.2.1.3 Sporočilno usmerjen vmesni sloj.....	17
2.2.1.4 Storitveno vodilo	18
2.2.2 Nivoji integracije aplikacij.....	19
2.2.2.1 Integracija na podatkovnem nivoju	20
2.2.2.2 Integracija na aplikacijskem nivoju	21

2.2.2.3	Integracija na nivoju metod.....	21
2.2.2.4	Integracija na nivoju uporabniškega vmesnika	21
2.3	Povzetek.....	22
3	Elektronski dokumentni sistem EBA DMS.....	23
3.1	Programski paket EBA DMS	23
3.1.1	Odjemalec EBA.....	24
3.1.2	Tiskalnik EBA	25
3.1.3	Nadzorni center storitev EBA	26
3.2	Dokumenti in njihov zajem v sistem EBA DMS	27
3.3	Elektronska izmenjava dokumentov.....	28
3.3.1	EBA Exchange	28
3.3.2	Elektronska pošta.....	29
3.3.3	Bank Exchange	29
3.3.4	UJP Exchange.....	30
3.4	Načini povezovanja s poslovno informacijskimi sistemi	32
3.4.1	Vmesnik EBAXCOM	32
3.4.2	Razvojno okolje EBA Developer	32
3.4.2.1	Predloga projekta za navaden skriptni projekt	33
3.4.2.2	Predloga projekta za razvoj vtičnikov	34
3.4.2.3	Predloga projekta za generiranje obrazcev	35
3.4.2.4	Objava projekta v sistem EBA DMS	35

3.5	Povzetek.....	37
4	Integracija Dynamics NAV s sistemom EBA DMS	39
4.1	Sistem za upravljanje z dokumenti v sistemu Dynamics NAV	39
4.1.1	Vrste dokumentov DMS	40
4.1.2	Dovoljenja vrst dokumentov DMS.....	42
4.1.3	Gradnik za predogled dokumentov iz sistema EBA DMS	43
4.1.4	Opravilo za izmenjavo dokumentov med EBA DMS in Dynamics NAV.....	44
4.1.4.1	Uvoz novih dokumentov iz sistema EBA DMS.....	45
4.1.4.2	Izvoz novih dokumentov v sistem EBA DMS	46
4.2	Modul NAV v sistemu EBA DMS.....	48
4.3	Primer - Prejeti računi in likvidacija	52
4.4	Sistem opozarjanja uporabnikov z alarmi.....	54
4.5	Povzetek.....	55
5	Integracija aplikacij BIS in ŽCO s sistemom EBA DMS	57
5.1	Uporaba spletnih storitev EBA DMS	57
5.2	Program za arhiviranje dokumentov iz aplikacij BIS in ŽCO	58
5.2.1	Elektronsko pošiljanje izdanih računov iz aplikacije BIS	60
5.2.2	Elektronsko pošiljanje izdanih zbirnih računov iz aplikacije BIS.....	62
5.3	Spletna storitev za masovno pripravo dokumentov	65
5.3.1	Predstavitev problema.....	65
5.3.2	Predstavitev rešitve	66

5.4	Povzetek.....	71
6	Integracija spletne aplikacije e-Križanka s sistemom EBA DMS	73
6.1	Opis integracije	75
6.2	Spletna aplikacija e-Križanka in spletna storitev Križanka v sistemu EBA DMS v postopku priprave in shranjevanja dokumenta.....	76
6.3	Uporabniški modul Križanka v sistemu EBA DMS in spletna storitev e-Križanka v postopku parafiranja	77
6.4	Orodje za nadzor nad dokumenti spletne aplikacije e-Križanka.....	81
6.5	Nadomeščanje	82
6.5.1	Splošno nadomeščanje	82
6.5.2	Področno nadomeščanje.....	83
6.5.2.1	Vnos nastavitvev za področno nadomeščanje	84
6.5.2.2	Modul NAV v sistemu EBA DMS	85
6.5.2.3	Opravilo za uveljavitev sprememb glede na nastavitve področnega nadomeščanja	86
6.6	Integracija e-Križanke in strežnika Microsoft Exchange	87
6.7	Povzetek.....	90
7	Razvoj upravljanja z elektronskimi dokumenti v prihodnje	91
8	Zaključek	93
9	Priloge	95
9.1	WSDL spletne storitve EBAIntegration v sistemu Dynamics NAV.....	95
9.2	WSDL spletne storitve GeneralDocuments v sistemu EBA DMS	97

9.3	WSDL spletne storitve za masovno pripravo dokumentov	98
9.4	WSDL spletne storitve Križanka v sistemu EBA DMS	106
9.5	WSDL spletne storitve e-Križanka spletne aplikacije e-Križanka.....	109
	Literatura in viri	111

Kazalo slik

Slika 1: Blokovna shema informacijskega sistema predstavljenega v magistrskem delu	6
Slika 2: Prikaz integracijske arhitekture od točke do točke	16
Slika 3: Prikaz integracijske arhitekture s centralnim sistemom.....	17
Slika 4: Prikaz integracijske arhitekture s sporočilno usmerjenim vmesnim slojem	18
Slika 5: Prikaz integracijske arhitekture s storitvenim vodilom.....	19
Slika 6: Nivoji integracije aplikacij [7].....	20
Slika 7: Uporabniški vmesnik elektronskega dokumentnega sistema EBA DMS	24
Slika 8: Razvojno okolje EBA Developer.....	25
Slika 9: Shema delovanja navideznega tiskalnika	25
Slika 10: Uporabniški vmesnik Nadzornega centra storitev EBA.....	26
Slika 11: e-Prijava in e-Povratnica, povezava medbančnega sveta z EBA DMS in interno aplikacijo BIS	30
Slika 12: Primerjava pošiljanja e-računa med protokolom Bank Exchange in UJP Exchange	31
Slika 13: Primer projekta v razvojnem okolju EBA Developer	33
Slika 14: Primer uporabe obrazcev (pretvorba dokumenta e-odjava iz oblike XML v obliko PDF).....	35
Slika 15: Nastavitve projekta v razvojnem okolju EBA Developer	36
Slika 16: Modul za upravljanje s projekti razvitimi v programskem okolju EBA Developer .	37
Slika 17: Prikaz strani za urejanje vrst dokumentov DMS	41
Slika 18: Prikaz strani za urejanje dovoljenj vrst dokumentov DMS.....	42

Slika 19: Sistem za upravljanje z dokumenti v Dynamics NAV z gradnikom za predogled dokumentov iz sistema EBA DMS	43
Slika 20: Kartica dokumenta z gradnikom za predogled dokumentov iz sistema EBA DMS .	44
Slika 21: Opravilo za izmenjavo dokumentov med EBA DMS in Dynamics NAV	45
Slika 22: Tabela večnivojskega parafiranja pred začetkom parafiranja.....	49
Slika 23: Tabela večnivojskega parafiranja po končanem parafiranju	51
Slika 24: Proces parafiranja dokumenta iz sistema Dynamics NAV v sistemu EBA DMS	51
Slika 25: Poslovni proces obdelave vhodnih računov od prejema do knjiženja skozi vidik integracije med sistemoma EBA DMS in Dynamics NAV	52
Slika 26: Primer elektronskega sporočila, ki ga dobi uporabnik kot alarm zaradi neizvedenega parafiranja	55
Slika 27: Struktura datotečnega sistema za arhiviranje dokumentov BIS in ŽCO	58
Slika 28: Prikaz povezanosti komponent spletne aplikacije e-Križanka s sistemom EBA DMS in obratno.....	75
Slika 29: Vstavljanje dokumenta iz aplikacije e-Križanka v sistem EBA DMS.....	77
Slika 30: Izvajanje akcij nad dokumenti e-Križanke v sistemu EBA DMS in nadziranje postopka s strani spletne storitve e-Križanka	80
Slika 31: Orodje za nadzor nad dokumenti spletne aplikacije e-Križanka	81
Slika 32: Modul za nastavljanje splošnega nadomeščanja preko vseh sistemov v e-Križanki	83
Slika 33: Modul za nastavljanje področnega nadomeščanja preko vseh sistemov v e-Križanki	85
Slika 34: Obdelava opravila za uveljavitev sprememb glede na nastavitve področnega nadomeščanja v spletni aplikaciji e-Križanka	87

Slika 35: Uporabniški vmesnik za izbiro dogodkov iz koledarja za uvoz v spletno aplikacijo e-Križanka	89
Slika 36: Uporabniški vmesnik za urejanje podrobnosti o posameznih dogodkih iz koledarja za uvoz v spletno aplikacijo e-Križanka.....	90

Kazalo tabel

Tabela 1: Stopnje poslovno-informacijske arhitekture [2].....	13
Tabela 2: Protokoli za elektronsko izmenjavo dokumentov	28

Seznam kratic

kratica	angleško	slovensko
AS2	Applicability Statement 2	specifikacija za varen in zanesljiv prenos podatkov preko interneta
API	Application Programming Interface	programski vmesnik
COM	Component Object Model	Microsoftova tehnologija
CRM	Customer Relationship Management	sistem za upravljanje odnosov s strankami
CSV	Comma Separated Values	format za besedilno datoteko, ki vsebuje z vejico ločene vrednosti
DMS	Document Management System	sistem za upravljanje z dokumenti
ERP	Enterprise Resource Planning	celovit informacijski sistem
EA	Enterprise Architecture	poslovno-informacijska arhitektura
EAI	Enterprise Application Integration	integracija aplikacij
ESB	Enterprise Service Bus	storitveno vodilo
IMAP	Internet Message Access protocol	internetni protokol, ki se uporablja za prejemanje elektronske pošte iz oddaljenega strežnika
JPG, JPEG	Joint Photographic Experts Group	format zapisa bitne slike
PDF	Portable Document Format	format datoteke, ki je neodvisen od računalniškega okolja in medoperacijsko prenosljiv
QoS	Quality of Service	kakovost storitve – lastnost, ki opredeljuje doseženo stopnjo storitve v primerjavi s pričakovano
RPC	Remote Procedure Call	klic oddaljene procedure
SMTP	Simple Mail Transport Protocol	internetni protokol za pošiljanje elektronske pošte
SOA	Service Oriented Architecture	storitveno usmerjena arhitektura

SOAP	Simple Object Access Protocol	protokol za spletne storitve, ki temelji na XML
TIFF	Tagged Image File Format	format zapisa bitne slike
TXT	Text	format zapisa besedila
UI	User Interface	uporabniški vmesnik
UJP		Uprava Republike Slovenije za javna plačila
WSDL	Web Services Description Language	jezik za opis spletnih storitev, ki temelji na XML
XML	Extensible Markup Language	razširljivi označevalni jezik

Povzetek

Magistrsko delo opisuje konkreten model integracije elektronskega dokumentnega sistema z ostalimi sistemi v srednje velikem podjetju v Sloveniji. V začetku dela so predstavljene teoretične osnove raziskovalnega področja, ki je integracija računalniških sistemov. V nadaljevanju je s stališča razvijalca podrobno predstavljen konkreten elektronski dokumentni sistem EBA DMS. V osrednjem delu magistrskega dela so predstavljeni načini integracije elektronskega dokumentnega sistema EBA DMS z ostalimi sistemi v podjetju. Delo se zaključuje s predstavitvijo nadaljnjega dela na področju poslovanja z elektronskimi dokumenti v podjetju.

Ključne besede: integracija, povezovanje, elektronski dokumentni sistem, informacijski sistem, aplikacija, COM vmesnik, spletna storitev

Abstract

This master's thesis presents specific model of integration between electronic document management system and other systems in a medium-sized company in Slovenia. The introduction part of the thesis begins with theoretical basis of the research, which is the integration of computer systems. The introduction part is followed by presentation of a concrete electronic document management system EBA DMS. Central part of the master's thesis presents different methods of integration of electronic document system EBA DMS with other systems in company. Thesis is concluded with the presentation of further work in the field of electronic document management in the company.

Keywords: integration, electronic document management system, information system, application, COM interface, web service

1 Uvod

Večje kot je podjetje, večje je število dokumentov, s katerimi se podjetje vsakodnevno srečuje. Prav zato je za uspešno poslovanje podjetja izjemno pomembno učinkovito upravljanje s temi dokumenti.

V podjetju, v katerem delam in za katerega želim, da ostane anonimno, je v zadnjih letih potekala celovita prenova informacijskega sistema. V prvem delu prenove sta bili razviti dve namizni aplikaciji, ki pokrivata osnovno dejavnost podjetja. Aplikacija Življenjski cikel odjemalca (v nadaljevanju ŽCO) omogoča informacijsko podporo za postopek priključevanja odjemalcev na elektrodistribucijsko omrežje. Druga aplikacija pa podpira poslovni proces obračuna omrežnine (v nadaljevanju BIS). V drugem delu prenove smo prenovili preostale dele našega starega informacijskega sistema. Prešli smo na novo celovito informacijsko rešitev (angl. Enterprise Resource Planning, ERP) in sicer na rešitev MS Dynamics NAV. Predmet prenove so bila področja finančnega poslovanja, plač, kadrovskega poslovanja, materialno-skladišnega poslovanja, nabave, prodaje, baze tehničnih podatkov, osnovnih sredstev in drugih.

Med prenovo informacijskega sistema so bili prenovljeni poslovni procesi v podjetju, staro papirno poslovanje je izpodrinilo modernejše brezpapirno in vse zgoraj naštetе rešitve so bile tesno integrirane s sicer ločenim elektronskim dokumentnim sistemom.

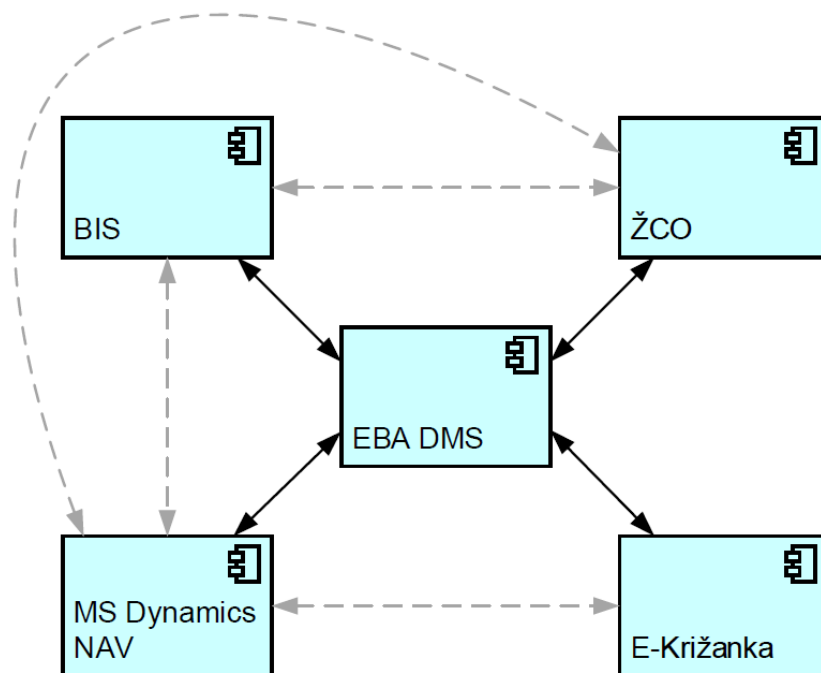
Raziskovalno področje magistrskega dela je področje poslovanja z elektronskimi dokumenti v podjetju. V podjetjih, ki ne poslujejo samo v okviru enega centralnega informacijskega sistema, ampak uporabljajo več specializiranih sistemov, je za uspešno poslovanje podjetja ključnega pomena prav tesna integracija med temi, sicer ločenimi sistemi. Eden izmed teh ključnih ločenih sistemov je elektronski dokumentni sistem, saj je navadno prav ta povezan z vsemi ostalimi sistemi.

V podjetju uporabljamo elektronski dokumentni sistem EBA DMS, ki je celovita rešitev za brezpapirno poslovanje v podjetjih in organizacijah [14]. Rešitev vključuje funkcionalnosti kot so zajemanje in izdajanje dokumentov, elektronska izmenjava pravno veljavnih dokumentov, napredno prepoznavanje dokumentov, enostavno iskanje dokumentov, podpora delovnim

tokovom dokumentov, nadzor nad dokumenti, avtorizacija in dostop do dokumentov, klasifikacija, varnost, povezljivost s poslovno informacijskimi sistemi, obveščanje itn.

V sistemu EBA DMS imamo na voljo več različnih načinov integracije z ostalimi sistemi in aplikacijami. Integracija je najpogosteje izvedena prek programske knjižnice EBAXCOM, ki je objektna programska knjižnica, napisana v tehnologiji COM. Knjižnica omogoča, da se informacijski sistem in dokumentni sistem zlijeta v eno. Drugi način povezovanja je orodje EBA Developer, ki je lastno razvojno orodje dokumentnega sistema EBA DMS. Z orodjem lahko sami v programskem jeziku JavaScript razvijamo dodatne module in specifične funkcionalnosti, ki jih želimo vgraditi v sistem EBA DMS.

V magistrskem delu bodo predstavljeni načini integracije sistema EBA DMS s sistemom Dynamics NAV, z namiznima aplikacijama ŽCO in BIS ter s spletno aplikacijo e-Križanka.



Slika 1: Blokovna shema informacijskega sistema predstavljenega v magistrskem delu

Med omenjenimi sistemi in aplikacijami obstajajo še druge integracijske povezave, ki pa niso predmet tega magistrskega dela in so na zgornji blokovni shemi označene s sivo črtkanimi povezavami.

Informacijski sistem Dynamics NAV je v osnovi s sistemom EBA DMS povezan preko programske knjižnice EBAXCOM. Drugi del te povezanosti zagotavlja interno razvit modul v

orodju EBA Developer, ki je implementiran v sistem EBA DMS. Natančneje bo predstavljeno obravnavanje prejetih računov in likvidacija le-teh, obravnava izdanih računov ter večnivojsko potrjevanje drugih dokumentov sistema Dynamics NAV v sistemu EBA DMS.

Aplikaciji ŽCO in BIS generirata račune, obvestila, pogodbe in druge dokumente v obliki PDF in jih odložita na dogovorjeno omrežno mesto, kjer čakajo, da jih aplikacija za arhiviranje ustrezno obdelata. Aplikacija za arhiviranje pripravljene dokumente vstavi v sistem EBA DMS preko programske knjižnice EBAXCOM, v izvorni aplikaciji pa shrani povratne informacije, potrebne za kasnejše odpiranje posameznih dokumentov neposredno iz aplikacije.

Spletna aplikacija e-Križanka se med drugim uporablja za vodenje evidence opravljenega dela zaposlenih. To je za napovedovanje nadur, dopustov, drugih odsotnosti, službenih poti, izobraževanj, za pripravo obračunov stroškov na službenih poteh itd. V ozadju vseh teh procesov nastajajo elektronski dokumenti, ki se vstavlja v elektronski dokumentni sistem EBA DMS in morajo biti potrjeni bodisi s strani delavca, njegovih nadrejenih, tajnice, direktorja oz. predsednika. Povezava med spletno aplikacijo e-Križanka in sistemom EBA DMS temelji na komunikaciji preko spletnih storitev. Na strani aplikacije e-Križanka sta tako aplikacija kot tudi spletna storitev napisani v tehnologiji ASP.NET. Na strani sistema EBA DMS pa sta v orodju EBA Developer razvita dva modula. Prvi predstavlja spletno storitev na strani sistema EBA DMS, drugi je pa implementiran v sistem EBA DMS in razširja funkcionalnosti osnovnega uporabniškega vmesnika sistema EBA DMS.

Magistrsko delo se zaključi s predstavitvijo nadaljnjega razvoja na področju elektronskega poslovanja z dokumenti v podjetju.

2 Integracija računalniških sistemov

Dandanes podjetje za uspešno poslovanje nujno potrebuje učinkovit in celovit informacijski sistem. Skozi čas, ko je tehnologija napredovala, so se v podjetjih razvijale ločene aplikacije oz. ločeni informacijski (pod)sistemi, predvsem z namenom, da bi poenostavili in/ali avtomatizirali nekatere aktivnosti, ki so se pred tem izvajale »ročno«. Vsaj nekaj od teh starih aplikacij oz. informacijskih (pod)sistemov se je v veliko podjetjih ohranilo vse do danes.

Še več, kljub dejstvu, da stremimo k čim bolj celovitemu informacijskemu sistemu, se vse več podjetij namesto za en osrednji informacijski sistem še vedno pogosto odloča za nakup in uvedbo več ločenih, namenskih in specializiranih aplikacij in/ali informacijskih sistemov. Vzroki za take odločitve podjetij so na eni strani finančne narave, saj so manjši produkti navadno cenejši, po drugi strani pa je podjetje z več manjšimi aplikacijami in/ali informacijskimi sistemi tudi bolj in hitreje prilagodljivo in se lahko v krajšem času prilagodi na spremembe, ki so v današnjem svetu za uspešno poslovanje podjetja nujno potrebne. Kot negativno stran uvedbe več ločenih, namenskih in specializiranih aplikacij in/ali informacijskih sistemov pa moramo v zakup vzeti dodatno delo in pozornost, ki ju moramo posvetiti integraciji teh aplikacij in/ali sistemov med seboj. Le z dobro integriranimi aplikacijami in/ali informacijskimi sistemi si namreč lahko zagotovimo enakovredno informacijsko podporo, kot bi jo sicer imeli pri enem samem celovitem informacijskem sistemu.

V nadaljevanju poglavja bo predstavljen pojem poslovno-informacijske arhitekture in stopnje, v katerih se lahko poslovno-informacijska arhitektura nahaja. Sledila bo predstavitev različnih integracijskih arhitektur in različnih nivojev, na katerih je lahko izvedena integracija med aplikacijami.

2.1 Poslovno-informacijska arhitektura

Arhitektura na nivoju poslovnega sistema oziroma skupine poslovnih sistemov z množico skupnih ciljev se imenuje poslovno-informacijska arhitektura (ang. enterprise architecture) [8].

Lankhorst in ostali poslovno-informacijsko arhitekturo definirajo, kot sledi: Poslovno-informacijska arhitektura je skladna celota načel, metod in modelov, ki se uporabljajo pri

načrtovanju in uresničevanju organizacijske strukture, poslovnih procesov, informacijskih sistemov in infrastrukture poslovnega sistema. Z njo tako lahko opišemo obstoječe in prihodnje stanje poslovnega sistema ter pripravimo plan prehoda iz obstoječega v željeno ciljno stanje [8] [4].

Bijata in Piotrkowki definirata poslovno-informacijsko arhitekturo kot formalni opis strukture in funkcij sestavnih delov podjetja (vključno z ljudmi, poslovnimi procesi, znanjem in tehnologijami v podjetju), odvisnosti med temi deli podjetja in načela ter smernice, ki vodijo njihovo ustvarjanje in razvoj čez čas. Poslovno-informacijska arhitektura je koncept, ki se stalno razvija. Podjetje začne na dnu in napreduje preko več stopenj, pri tem pa mnoga podjetja nikoli ne dosežejo vrha [2].

Arhitektom temelj za izgradnjo oziroma predstavitev poslovno-informacijske arhitekture predstavljajo ogrodja poslovno-informacijske arhitekture. Primer takih ogrodij so npr. Zachmanovo ogrodje, TOGAF (angl. The Open Group Architecture Framework), DoDAF (angl. Department of Defense Architecture Framework) in druga.

2.1.1 Stopnje poslovno-informacijske arhitekture

Obstajajo štiri stopnje poslovno-informacijske arhitekture. To so poslovni silosi, poenotenje tehnologij, optimizacija jedra podjetja in poslovna modularnost. Na vsaki stopnji ima podjetje določene lastnosti, kot so modularnost, kohezija, integriranost in avtomatizacija. Te skupaj določajo kako napredno oz. zrelo je podjetje. Gruman pravi [18], da podjetje v povprečju potrebuje okoli pet let, da napreduje na naslednjo stopnjo poslovno-informacijske arhitekture, odvisno od predanosti podjetja k temu cilju in števila napak, ki jih podjetje napravi na tej poti. Večina podjetij pa na zadnjo, četrto stopnjo sploh nikoli ne pride.

2.1.1.1 Poslovni silosi

Prva stopnja poslovno-informacijske arhitekture je poslovni silos. Vsaka poslovna aplikacija, poslovni proces in poslovna enota v podjetju deluje neodvisno in nudi zelo nizko stopnjo integracije različnih poslovnih procesov in funkcij med različnimi poslovnimi enotami v podjetju. Vsako področje v podjetju se osredotoča zgolj na izboljšanje poslovanja lastnega področja in ne na izboljšanje poslovanja podjetja kot celote [1].

Na tej stopnji je arhitektura v podjetju sestavljena iz neodvisnih sistemov, ki nimajo skupne oziroma standardizirane tehnologije. Bijata and Pitrkowski pravita: »Osebe, ki vodijo investicije v IT, so vodje posameznih področij in njihove investicije se osredotočajo zgolj na izpolnjevanje lokalnih potreb na njihovem ozkem področju« [2].

Glavne lastnosti arhitekture poslovnih silosov [19]:

- enkratne rešitve,
- po principu od spodaj navzgor,
- ni integracije z ostalimi informacijskimi sistemi,
- ni povezave s strežniki,
- ni deljenja podatkov.

2.1.1.2 Poenotenje tehnologij

Naslednja stopnja v razvoju poslovno-informacijske arhitekture je poenotenje tehnologij znotraj podjetja. Na tej stopnji je podjetje sprejelo in implementiralo standardno tehnološko platformo na vseh področjih v podjetju. Poenotenje tehnologij prinese povečanje zanesljivosti in zmanjšanje števila tehnoloških platform, ki jih je potrebno upravljati. Manj platform sicer pomeni nižje stroške, na drugi strani pa pomeni tudi manjšo izbiro produktov, ki ustrezajo izbrani platformi. Vendar so podjetja pripravljena sprejeti ta kompromis.

Kljub poenotenju tehnologij pa na tem nivoju posamezna področja med seboj še vedno niso povezana in tako ne morejo nemoteno izmenjevati informacij. Podatki in aplikacije se tako implementirajo in nadzirajo še vedno na lokalnem nivoju, namesto na nivoju celotnega podjetja. Cilji in strategija podjetja še vedno temeljijo zgolj na izboljšanju poslovanja znotraj posameznih področij v podjetju.

Glavne lastnosti arhitekture poenotenje tehnologij [19]:

- racionalizacija, standardizacija in konsolidacija IT infrastrukture,
- vzpostavitev zanesljive in stroškovno učinkovite IT infrastrukture,
- še vedno se osredotoča na hitre zmage oz. enkratne rešitve.

2.1.1.3 Optimizacija jedra podjetja

Tretja stopnja v razvoju poslovno-informacijske arhitekture je optimizacija jedra podjetja. Na tej stopnji se v podjetju uvede globalna infrastruktura, stremi pa se k popolni poslovni integraciji. Pri optimizaciji jedra se podjetje osredotoča na skupne podatke in programe ter integrirane informacijske sisteme. Na tej stopnji se poslovna strategija in cilji ne planirajo več lokalno oz. na nivoju posameznih področij, ampak s strani širšega in predvsem višjega vodstva podjetja, z ozirom na poslovanje podjetja kot celote. Prav zaradi tega lahko posamezni vodje področij to stopnjo zavračajo, saj jim je odvzeta moč, ki so jo na prejšnjih dveh stopnjah poslovno-informacijske arhitekture še imeli.

Glavne lastnosti arhitekture optimizacija jedra podjetja [19]:

- poenotenje ključnih poslovnih procesov,
- vodje področij izgubijo vpliv pri odločanju,
- združevanje odvečnih aplikacij v enoten informacijski sistem (npr. ERP in CRM),
- podatki in aplikacije se implementirajo na nivoju podjetja,
- po principu od zgoraj navzdol. Poslovni procesi in investicije v IT se izvedejo v centralnem IT oddelku.

2.1.1.4 Poslovna modularnost

Poslovna modularnost je četrti in zadnji nivo poslovno-informacijske arhitekture. Na tem nivoju se na poslovne procese gleda kot na modularne enote, ki so dobro definirane in učinkovite. Te modularne enote so lahko implementirane in oblikovane za opravljanje različnih nalog v podjetju. Večkratno uporabljivi moduli ponujajo ogrodje, s katerim lahko povečamo hitrost in izboljšamo razvoj novih poslovnih procesov v podjetju. Podjetja na tej stopnji lahko tako uporabnikom ponudijo hiter razvoj novih idej oz. prilagodljivost obstoječih.

Poslovna modularnost zagotavlja najboljšo rešitev za podjetje, tako z globalnega nivoja kot tudi z lokalnih nivojev.

Glavna lastnost arhitekture poslovna modularnost je, da lokalna prilagodljivost preide v globalno prilagodljivost [19].

2.1.2 Primerjava stopenj poslovno-informacijske arhitekture

Vsaka stopnja poslovno-informacijske arhitekture:

- določa, kako zrelo je podjetje,
- določa, kako so določeni poslovni procesi v podjetju,
- opisuje, kdo in kakšne vrste odločitve sprejema v podjetju.

V naslednji tabeli je prikazan razvoj poslovno-informacijske arhitekture po stopnjah. Lokalno pomeni, da so odločitve in implementacije izvedene na nivojih posameznih področij. Globalno pomeni, da so le-te izvedene na nivoju podjetja oz. vrhovnega vodstva podjetja. Modularno pa pomeni lahko oboje, tako lokalno kot tudi globalno.

	Poslovni silosi	Poenotenje tehnologij	Optimizacija jedra podjetja	Modulacija poslovanja
Infrastruktura	lokalno	globalno	globalno	globalno
Podatki	lokalno	lokalno	globalno	globalno
Aplikacije	lokalno	lokalno	globalno	modularno
Kdo odloča	vodje področij	vodje področij in vodja IT	vrhovno vodstvo (uprava)	vodja IT, vodje področij

Tabela 1: Stopnje poslovno-informacijske arhitekture [2]

2.2 Integracija aplikacij

Po definiciji se integracija aplikacij (angl. Enterprise Application Integration, EAI) nanaša na proces integracije več aplikacij, ki so bile sicer razvite ločeno, neodvisno ena od druge, lahko tudi z uporabo različnih tehnologij, in jih lahko po integraciji še vedno samostojno upravljamo [6].

Integracija aplikacij (EAI) vključuje metode, tehnike in orodja za povezovanje in usklajevanje programov in podatkov v podjetju za zagotavljanje večje združljivosti.

David S. Linthicum definira integracijo aplikacij kot neomejeno deljenje podatkov in poslovnih procesov med vsemi povezanimi aplikacijami in podatkovnimi viri v podjetju [3].

Poslovni sistemi podjetij so velikokrat sestavljeni iz več ločenih sistemov in programov. Od njih vsak posebej podjetju ponuja različne storitve, od katerih je dnevno odvisna uspešnost poslovanja podjetja. Ločeni sistemi oz. programi so lahko bodisi razviti v podjetju ali pa so kupljeni od zunanjih ponudnikov. Taka modulacija je v podjetjih navadno zaželeno. V teoriji taka modulacija sistema omogoča lažjo in hitrejšo implementacijo tako tehnoloških novosti na posameznem področju, kot tudi prilagoditev na hitro se spreminjajoče poslovne potrebe.

Podjetje lahko v celoti izkoristi prednosti takega modularnega sistema šele, ko se zaveda in z ustreznimi tehnologijami obravnava tudi probleme, ki jih taka arhitektura prinaša [17].

- Interoperabilnost: različni deli sistema lahko uporabljajo različne operacijske sisteme, oblike podatkov, programske jezike, kar pa preprečuje povezovanje sistemov s standardnimi vmesniki.
- Integracija podatkov: za funkcionalnost modularnega sistema je zelo pomemben standarden način obravnave toka podatkov med sistemi in aplikacijami. Le tak način obravnave podatkov bo zagotovil skladnost podatkov v podatkovnih bazah.
- Robustnost, stabilnost in skalabilnost: metode, tehnike in orodja za povezovanje in usklajevanje programov in podatkov so za tak modularni sistem kot lepilo, ki ta sistem drži skupaj. Zato je zelo pomembno, da so robustne, stabilne in skalabilne.

Integracija aplikacij (EAI) je proces povezovanja različnih sistemov, programov in podatkov s ciljem zagotoviti čim bolj učinkovit poslovni sistem za podjetje. Namen integracije aplikacij je povezovanje vseh ločenih sistemov v podjetju za zagotovitev nekega poslovnega cilja.

Za zagotovitev integracije aplikacij moramo obravnavati in zagotoviti fizično integracijo, aplikacijsko integracijo ter tudi poslovno integracijo. Fizično integracijo zagotovimo že s povezovanjem vseh naprav, ki se uporabljajo v poslovnem okolju. Aplikacijska integracija je proces povezovanja različnih sistemov oz. programov v podjetju. Poslovna integracija pa vključuje tako razumevanje kot tudi optimizacijo samih poslovnih procesov v podjetju.

2.2.1 Integracijska arhitektura

Za razumevanje integracijske arhitekture je nujno potrebno poznavanje in razumevanje integracijskih topologij, iz katerih so izpeljane posamezne integracijske arhitekture [3].

Osnovne integracijske arhitekture so:

- integracija od točke do točke,
- integracija strežnik–odjemalec,
- sporočilno usmerjen vmesni sloj,
- storitveno vodilo.

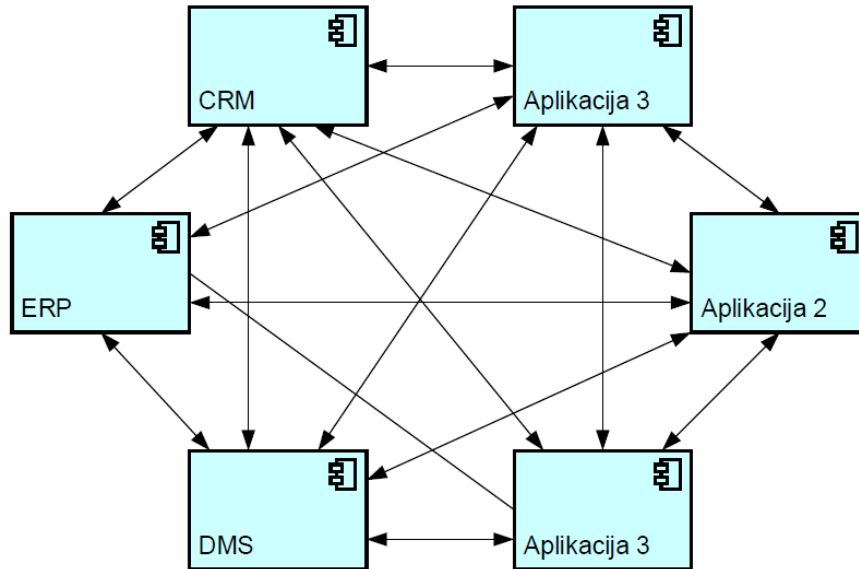
V nadaljevanju poglavja bodo podrobneje predstavljene vse štiri zgoraj navedene integracijske arhitekture.

2.2.1.1 Integracija od točke do točke

Največkrat se je v preteklosti za integracijo aplikacij uporabila prav integracijska arhitektura od točke do točke [3].

Pri integraciji od točke do točke se za vsak par aplikacij oz. sistemov definira svoja in od drugih ločena integracijska rešitev. Ta arhitektura je izmed vseh integracijskih arhitektur najlažja. Prednost te arhitekture pred ostalimi je, da lahko z njo dosežemo zelo tesno integracijo, saj se obe točki, ki nastopata v konkretni integraciji, neposredno zavedata ena druge. Za to arhitekturo se največkrat odločamo prav zaradi enostavnosti in hitrosti implementacije. Poudariti je potrebno, da se učinkovitost te arhitekture s povečevanjem števila sistemov, ki jih želimo integrirati, hitro zmanjšuje. Če imamo v integraciji vsak sistem povezan z vsakim sistemom, moramo namreč v implementaciji rešitve poskrbeti za število povezav $n*(n-1)/2$. To je število povezav, ki nastopajo v polnem grafu. Največja slabost te arhitekture je prav zaradi tega skalabilnost.

Naslednja shema prikazuje integracijsko arhitekturo od točke do točke.



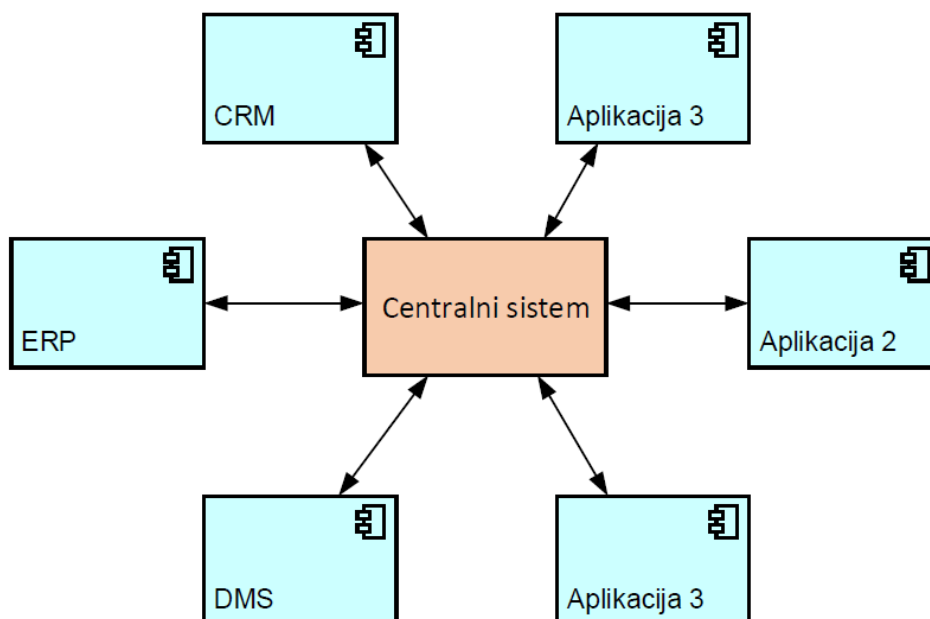
Slika 2: Prikaz integracijske arhitekture od točke do točke

2.2.1.2 Integracija strežnik–odjemalec

Glavna ideja pri načinu integracije strežnik–odjemalec (angl. Hub-and-Spoke) je, da zmanjšamo število povezav, ki s povečevanjem števila sistemov v integraciji pri načinu od točke do točke zelo hitro narašča. Pri tej arhitekturi mrežo povezav iz integracijske arhitekture od točke do točke nadomestimo z nečim, kar spominja na kolo. V sredini je centralni sistem, špice kolesa pa predstavljajo povezave na integrirane sisteme. V taki topologiji je število povezav enako številu sistemov v integraciji, kar pomeni, da je rast števila povezav linearna.

Največja slabost te arhitekture je dejstvo, da so vsi sistemi preko povezav povezani z enim samim centralnim sistemom, ki v primeru nedelovanja povzroči nedelovanje celotnega integriranega sistema.

Tudi če se odločimo za integracijo na način strežnik–odjemalec, se rešitev z dodajanjem nekaj dodatnih povezav od točke do točke, ki obidejo strežnik, lahko izkaže kot zelo učinkovita. Z razvojem in uporabo integriranega sistema večino arhitektur tako ali tako konča kot nek hibrid več integracijskih arhitektur [16].



Slika 3: Prikaz integracijske arhitekture s centralnim sistemom

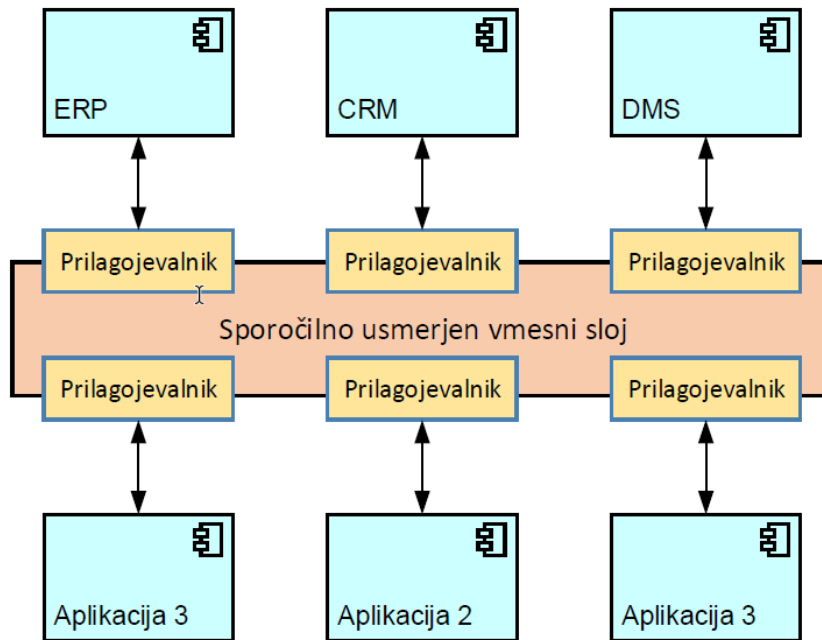
2.2.1.3 Sporočilno usmerjen vmesni sloj

Sporočilno usmerjen vmesni sloj je kombinacija skupnega podatkovnega modela, skupnega nabora ukazov in sporočilne infrastrukture, ki omogoča različnim sistemom in aplikacijam, da komunicirajo med seboj preko skupnega nabora vmesnikov [5].

Sporočilno usmerjen vmesni sloj lahko primerjamo s komunikacijskim vodilom v računalniškem sistemu, ki služi kot osrednja linija za komunikacijo med procesorjem, glavnim pomnilnikom in perifernimi napravami. Prav tako kot v strojni analogiji, obstaja tudi v primeru sporočilno usmerjenega vmesnega sloja več sestavnih delov, ki skupaj tvorijo sporočilno usmerjen vmesni sloj.

Sestavni deli, ki sestavljajo sporočilno usmerjen vmesni sloj so:

- skupna komunikacijska struktura (skrbi za pravilno usmerjanje sporočil med sistemi in aplikacijami),
- prilagojevalniki (angl. adapters) (omogočajo, da lahko različni sistemi in aplikacije komunicirajo s sporočilno usmerjenim vmesnim slojem),
- skupna struktura ukazov (nabor skupnih ukazov, ki jih razumejo vsi sistemi, ki so povezani na sporočilno usmerjen vmesni sloj).



Slika 4: Prikaz integracijske arhitekture s sporočilno usmerjenim vmesnim slojem

2.2.1.4 Storitveno vodilo

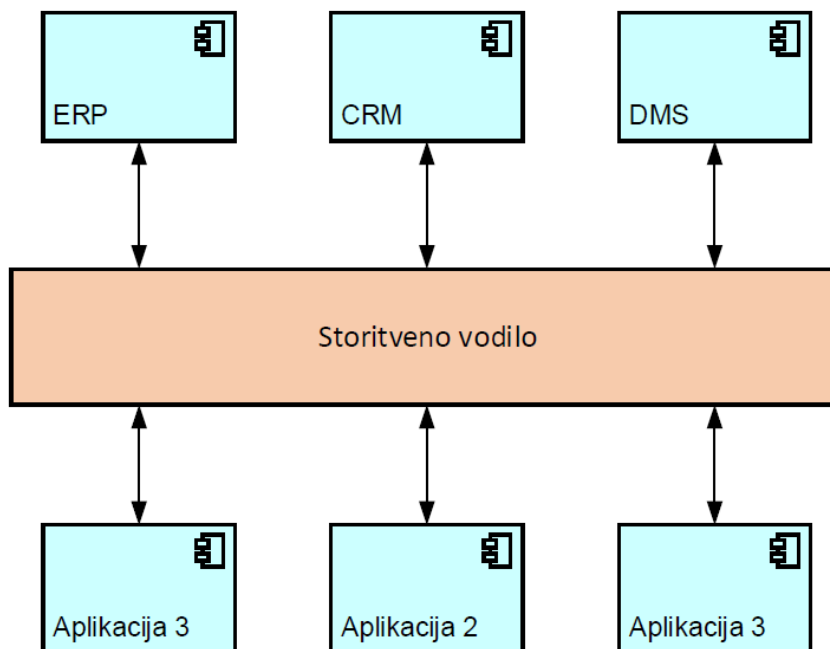
Pri storitvenem vodilu (angl. Enterprise Service Bus, ESB) različne aplikacije in sistemi med seboj ne komunicirajo neposredno, ampak preko vmesne storitveno usmerjene arhitekture (angl. Service Oriented Architecture, SOA). Večina rešitev, ki so zgrajene po principu storitvenega vodila, temelji na spletnih storitvah, ki za prenos in preoblikovanje sporočil uporabljajo jezik XML [3].

Gartner [9] in drugi [11] navajajo, da je storitveno vodilo arhitektura, ki z uporabo spletnih storitev sporočilno usmerjenega vmesnega nivoja, inteligentnega usmerjanja in možnostjo izvajanja transformacij predstavlja integracijsko osnovo za vodenje toka in upravljanja programskih storitev in aplikacijskih komponent.

Povzeto po [3] in [17] so glavne značilnosti storitvenega vodila naslednje:

- inteligentno usmerjanje sporočil,
- mediacija sporočil: storitveno vodilo skrbi za premoščanje razlik v protokolu, formatu in varnosti med pošiljatelji in prejemniki sporočil,
- upravljanje procesov,

- kvaliteta storitev (angl. Quality of Service, QoS): storitveno vodilo mora zagotavljati zanesljivo dostavo sporočil in transakcijsko obnašanje,
- varnost,
- spremljanje aktivnosti: revizija, beleženje in merjenje.



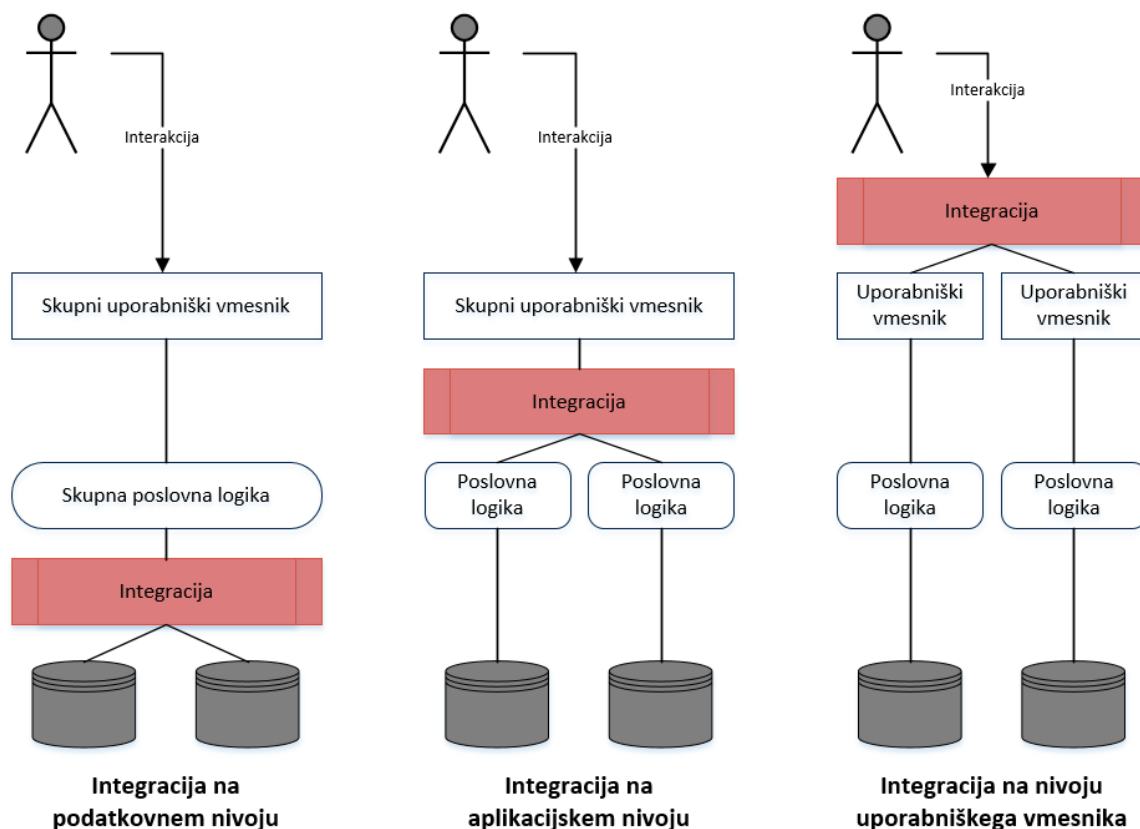
Slika 5: Prikaz integracijske arhitekture s storitvenim vodilom

Tako kot pri stopnjah poslovno-informacijskih arhitektur tudi pri integracijskih arhitekturah velja, da se je integracije ne glede na izbrano arhitekturo treba lotiti premišljeno in postopoma ter z discipliniranim, nadzorovanim in sistematičnim pristopom [3].

2.2.2 Nivoji integracije aplikacij

Integracija aplikacij (EAI) je lahko izvedena na različnih integracijskih nivojih. Poznamo naslednje [10]:

- nivo podatkov,
- nivo vmesnikov do aplikacije,
- nivo metod,
- nivo uporabniških vmesnikov.



Slika 6: Nivoji integracije aplikacij [7]

2.2.2.1 Integracija na podatkovnem nivoju

Pri integraciji na podatkovnem nivoju (angl. data-level integration) so med seboj integrirani viri podatkov različnih aplikacij, kar omogoča pretok podatkov med njimi. Povedano drugače, podatki, pridobljeni iz ene baze podatkov, se obdelajo, če je to potrebno, in nato vstavijo v bazo podatkov druge aplikacije.

Integracija tega tipa je lahko izvedena na dva načina, in sicer lahko temelji bodisi na principu potiska (angl. push) bodisi na principu potega (angl. pull). Pri principu potiska ena aplikacija izvaja ukaze SQL na tabelah druge aplikacije, s čimer potiska svoje podatke v podatkovno bazo druge aplikacije bodisi preko povezav med bazami podatkov bodisi preko baznih procedur (angl. stored procedure). Nasprotno pa gre pri principu potega za uporabo prožilcev (angl. triggers) in povpraševanja (angl. polling). Prožilci zaznavajo spremembe v podatkih in zapišejo identifikacijske podatke v neko vmesno tabelo. Prilagojevalniki (angl. adaptors) pa s periodičnim povpraševanjem vmesne tabele, preko identifikacijskih podatkov, ki so v njej,

pridobijo ustrezne podatke v izvornih tabelah in izvedejo ustrezne spremembe v podatkovni bazi ciljne aplikacije [15].

Prednost te metode pred drugimi je, da pri tej metodi ni treba posegati v programsko kodo aplikacij, ki so med seboj integrirane. Vseeno pa se je dobro držati načela, da to metodo uporabljamo le, ko nam aplikacija, s katero se želimo integrirati, ne ponuja zadosti programskih vmesnikov (angl. Application Programming Interface, API) za integracijo [10] [15].

2.2.2.2 Integracija na aplikacijskem nivoju

Integracija na aplikacijskem nivoju (angl. application-level integration) je najverjetneje najboljši način integriranja aplikacij. Pri tem načinu za integracijo z neko aplikacijo uporabimo ogrodja in programske vmesnike (angl. Application Programming Interface, API), ki nam jih ta aplikacija ponuja za integracijo. Programski vmesniki aplikacij omogočajo in zagotavljajo, da se vedno sklicujemo neposredno na poslovno logiko, ki velja v aplikacijah. Prav to pa nam v vsakem trenutku integracije zagotavlja ohranjanje celovitosti podatkov v aplikaciji [15].

2.2.2.3 Integracija na nivoju metod

Integracija na nivoju metod (angl. method-level integration) je nadgrajena oblika integracije na nivoju programskih vmesnikov, ki se ne uporablja tako pogosto. Pri integraciji na nivoju metod združimo skupne funkcije sicer več aplikacij v eno aplikacijo, ki jo v nadaljevanju obravnavamo kot »glavno« aplikacijo za izvajanje teh funkcij. Slabost te metode je, da spremembe v programskem vmesniku integrirane aplikacije povzroči nedelovanje »glavne« aplikacije in s tem tudi nedelovanje vseh aplikacij, ki se zanašajo nanjo [15].

2.2.2.4 Integracija na nivoju uporabniškega vmesnika

Integracija na nivoju uporabniškega vmesnika (angl. user interface (UI)-level integration) se navadno uporabi, ko odpovejo vse ostale metode integracije. To je takrat, ko ne moremo dostopati do podatkovne baze aplikacije, ko nimamo na voljo programskih vmesnikov ali ko je logika aplikacije vdelana neposredno v uporabniški vmesnik. Takrat je uporabniški vmesnik edini mehanizem, preko katerega lahko dostopamo do poslovne logike in podatkov aplikacije.

S tem je to tudi edini način, preko katerega imamo zagotovljeno ohranjanje celovitosti podatkov v aplikaciji [15].

Integracija na nivoju uporabniškega vmesnika združi aplikacije s ponovno uporabo uporabniških vmesnikov teh aplikacij v nov, skupen uporabniški vmesnik. To pomeni, da je predstavitveni nivo sestavljene aplikacije sestavljen, vsaj v nekaterih delih, iz predstavitvenih nivojev posameznih komponent osnovnih aplikacij.

2.3 Povzetek

Integracija informacij je stara toliko, kot je stara informacijska tehnologija [3]. Izvajamo jo lahko na različne načine in z uporabo različnih tehnologij. Najbolj je pomembno, da nam integrirane rešitve ob koncu dneva rešijo več obstoječih problemov, kot nam ustvarijo novih. Prav zato sta poznavanje in izbira pravilne integracijske arhitekture ter ustreznega nivoja, na katerem je integracija aplikacij izvedena, ključnega pomena pri zagotovitvi učinkovitega in celovitega informacijskega sistema.

V naslednjem poglavju bodo podrobneje predstavljeni elektronski dokumentni sistem EBA DMS, moduli, ki ga sestavljajo, načini, ki jih ponuja za elektronsko izmenjavo dokumentov in načini povezovanja z drugimi poslovno informacijskimi sistemi.

3 Elektronski dokumentni sistem EBA DMS

Elektronski dokumentni sistem EBA DMS je celovita rešitev za brezpapirno poslovanje v podjetjih in organizacijah [14]. Rešitev omogoča zajemanje in izdajanje dokumentov, elektronsko izmenjavo pravno veljavnih dokumentov, napredno prepoznavanje dokumentov, enostavno iskanje dokumentov, podporo delovnim tokovom dokumentov, nadzor nad dokumenti, avtorizacijo in dostop do dokumentov, klasifikacijo dokumentov, varnost, obveščanje uporabnikov, popolno elektronsko hrambo vseh poslovnih dokumentov v podjetju, povezljivost s poljubnimi poslovno-informacijskimi sistemi itn.

EBA DMS je zelo odprt in prilagodljiv dokumentni sistem [14]. Pooblaščen uporabnik lahko že samo skozi uporabniške vmesnike sistema EBA DMS v veliki meri prilagajajo in nadzirajo delovanje sistema EBA DMS. Uporabniki lahko sami določajo nove tipe dokumentov in upravljajo z obstoječimi, sami lahko predpisujejo attribute tipov dokumentov, določajo pooblastila uporabnikov v sistemu, določajo tokove dokumentov in pravila nad dokumenti.

Poleg bogatih možnosti za pooblaščen uporabnik pa tudi za razvijalce obstajajo zelo široke možnosti za nadgrajevanje in prilagajanje sistema EBA DMS.

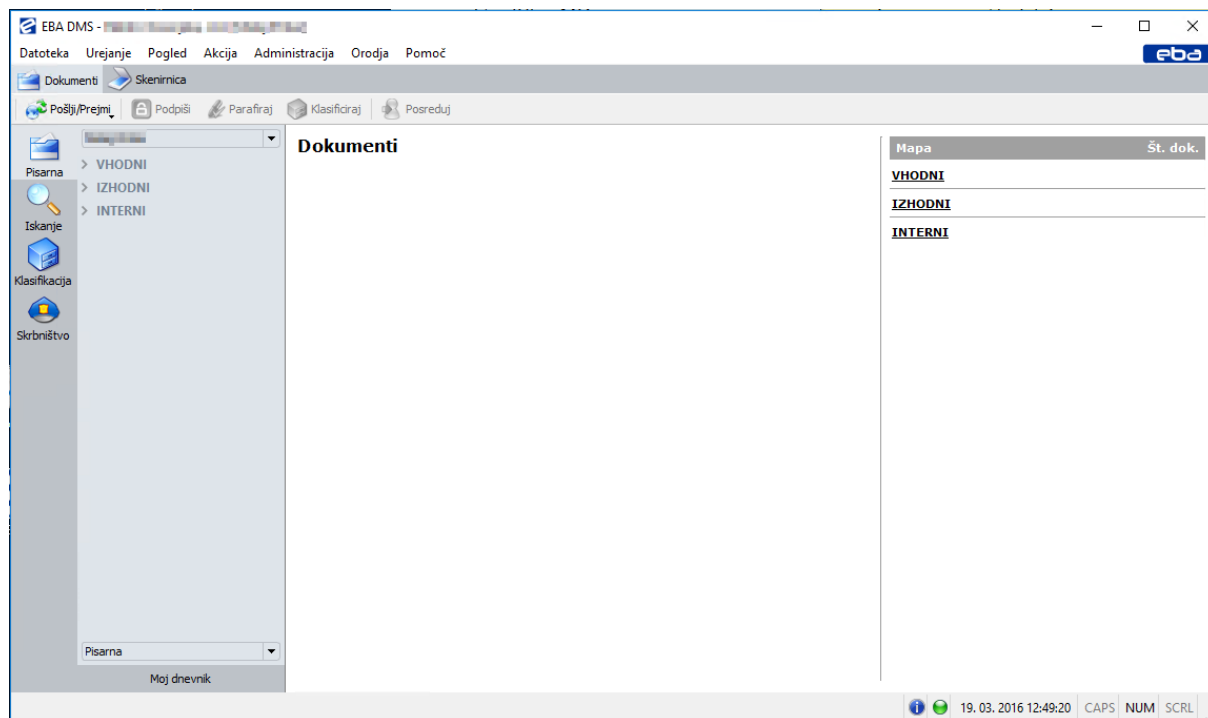
3.1 Programski paket EBA DMS

Prvi korak pri pripravi okolja za elektronski dokumentni sistem EBA DMS zajema izbiro in namestitvev podatkovne baze. Izbiramo lahko med podatkovnimi bazami PostgreSQL, Oracle in MS SQL.

Elektronski dokumentni sistem EBA DMS vsebuje več namestitvenih paketov oz. t. i. modulov. Nekateri moduli se namestijo na strežnik, drugi na računalnike končnih uporabnikov, nekateri pa tudi na obe vrsti naprave.

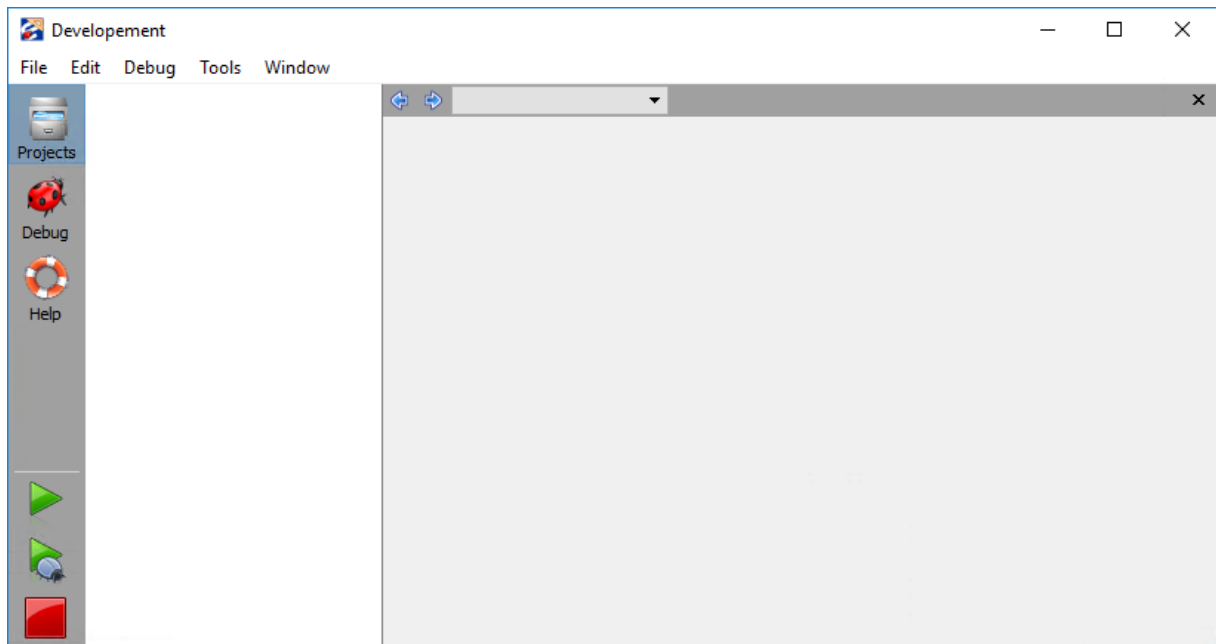
3.1.1 Odjemalec EBA

Odjemalec EBA predstavlja uporabniški vmesnik za končne uporabnike elektronskega dokumentnega sistema EBA DMS. Namestimo ga na računalnike končnih uporabnikov, lahko pa tudi na strežnik.



Slika 7: Uporabniški vmesnik elektronskega dokumentnega sistema EBA DMS

Drugi del namestitvenega paketa Odjemalec EBA je namestitveni paket, ki vsebuje razna orodja EBA. Orodja EBA so namenjena izključno razvijalcem. Eno izmed orodij je tudi razvojno okolje EBA Developer, ki bo natančneje predstavljeno v nadaljevanju.

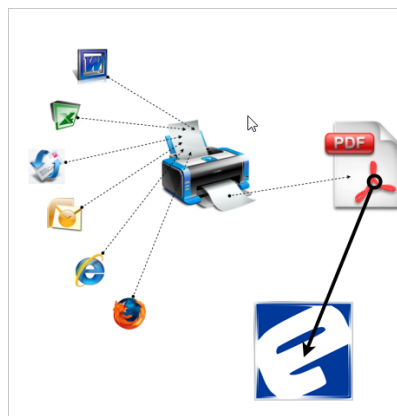


Slika 8: Razvojno okolje EBA Developer

3.1.2 Tiskalnik EBA

Tiskalnik EBA je program, ki deluje kot navidezni tiskalnik in se uporablja za zajemanje elektronskih dokumentov v elektronski dokumentni sistem EBA DMS. Namestimo ga samo na računalnikih končnih uporabnikov. V okolju Windows je navidezni tiskalnik EBA viden kot vsak drug tiskalnik, ki je nameščen v okolje.

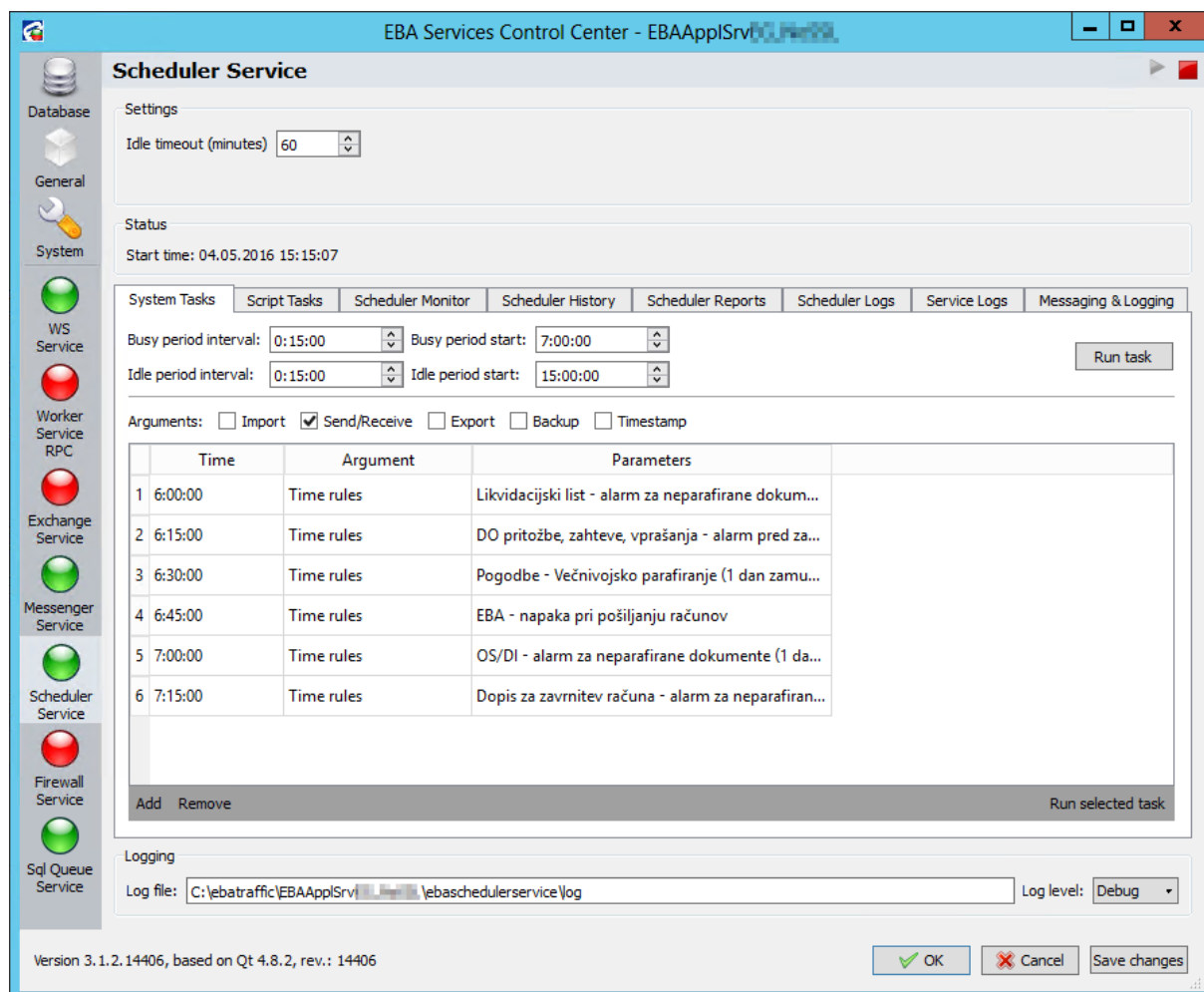
Preko navideznega tiskalnika lahko uporabniki iz vsakega programa, ki omogoča tiskanje na tiskalnik, z izbiro tiskalnika EBA natisnejo dokument v elektronsko obliko v elektronski dokumentni sistem EBA DMS.



Slika 9: Shema delovanja navideznega tiskalnika

3.1.3 Nadzorni center storitev EBA

Nadzorni center storitev EBA (angl. EBA Services Control Center) je samostojen program, ki ga namestimo samo na strežniku. Preko programa nadzorujemo in upravljamo skupino neodvisnih storitev, ki so del strežniškega programa.



Slika 10: Uporabniški vmesnik Nadzornega centra storitev EBA

Najpomembnejše izmed storitev, ki jih uporabljamo preko nadzornega centra storitev EBA, so:

- storitev za gostovanje spletnih storitev (WS Service): skrbi za gostovanje spletnih storitev sistema EBA DMS,
- storitev za izvajanje oddaljenih klicev (Worker Service RPC): nekatere funkcije, ki jih sicer uporabnik izvede na strani uporabniškega vmesnika Klient EBA, se dejansko izvedejo na strežniku. Za njihovo izvedbo pa poskrbi storitev za izvajanje oddaljenih klicev (angl. Remote Procedure Call),

- storitev za zagotavljanje obveščanja (Messenger Service): storitev skrbi za obveščanje uporabnikov tako preko elektronskih sporočil, kot tudi preko sporočilnih oken v Klientu EBA,
- storitev za izvajanje opravil (Scheduler Service): skrbi za izvajanje ponavljajočih se opravil, kot so Pošlji/Prejmi, časovno žigosanje dokumentov, izvajanje drugih časovnih opravil, lahko tudi samo funkcij itd.

3.2 Dokumenti in njihov zajem v sistem EBA DMS

V elektronskem dokumentnem sistemu EBA DMS dokumente najprej ločimo glede na smer. Poznamo tri vrste dokumentov glede na smer:

- vhodni dokumenti,
- izhodni dokumenti,
- interni dokumenti.

Vhodni so dokumenti, ki jih podjetje prejme od zunanjih poslovnih partnerjev. Izhodni so tisti, ki jih podjetje pošlje zunanjim poslovnim partnerjem. Interni dokumenti pa nastanejo v podjetju in se ne pošiljajo izven podjetja.

Dokumente ločimo tudi glede na tipe dokumentov. Razdelitev na tipe dokumentov temelji na vsebini dokumentov. Tipi dokumentov so npr. računi, dopisi, zahtevnice, naročilnice, vabila, soglasja, pritožbe itn.

Dokumente lahko v sistem EBA DMS zajamemo na naslednje načine:

- skeniranje z optičnim čitalnikom,
- uvoz dokumentov preko uporabniškega vmesnika EBA DMS,
- tiskanje preko navideznega tiskalnika EBA,
- uvoz preko spletnih storitev,
- uvoz neposredno iz integriranega informacijskega sistema,
- zajem preko enega izmed načinov elektronske izmenjave dokumentov.

3.3 Elektronska izmenjava dokumentov

Elektronska izmenjava dokumentov se izvaja samo nad vhodnimi in izhodnimi dokumenti elektronskega dokumentnega sistema EBA DMS. Poleg podatkov o smeri in tipu dokumenta se na vhodnih in izhodnih dokumentih v sistemu EBA DMS hranita še informaciji o protokolu in naslovu protokola elektronske izmenjave dokumentov.

V spodnji tabeli so prikazani protokoli, ki se v podjetju uporabljajo za elektronsko izmenjavo dokumentov.

Protokol	Naslov protokola	Opomba
EBA Exchange	ID poslovnega partnerja na agenciji EBA	
IMAP Email Exchange	Elektronski naslov pošiljatelja	Samo prejetanje
SMTP Email Exchange	Elektronski naslov prejemnika	Samo pošiljanje
UJP Exchange	BIC banke in TRR prejemnika	
Bank Exchange	BIC banke in TRR prejemnika	

Tabela 2: Protokoli za elektronsko izmenjavo dokumentov

Elektronsko izmenjavo dokumentov lahko pooblaščen uporabnik kadarkoli zažene ročno, preko Odjemalca EBA, s klikom na gumb Pošlji/Prejmi. Običajno se ta funkcija proži avtomatsko na strežniku preko storitve za izvajanje opravil, v nastavljenem časovnem intervalu.

Funkcija Pošlji/Prejmi zaporedno izvede prejetanje in pošiljanje dokumentov po vseh protokolih, ki so v sistemu EBA DMS nastavljeni za uporabo.

V nadaljevanju bodo podrobneje predstavljeni posamezni protokoli elektronske izmenjave dokumentov.

3.3.1 EBA Exchange

Protokol izmenjave EBA Exchange je interni protokol in je del elektronskega dokumentnega sistema EBA DMS. Protokol omogoča neposredno elektronsko izmenjavo dokumentov samo med podjetji, ki uporabljajo elektronski dokumentni sistem EBA DMS.

Izmenjava dokumentov poteka na relaciji med sistemi EBA DMS posameznih podjetij in distribucijskim centrom EBA. Pri pošiljanju dokumentov po protokolu EBA Exchange se dokumenti iz našega sistema EBA DMS pošljejo v distribucijski center EBA in tam čakajo, da jih sistem EBA DMS zunanjega poslovnega partnerja od tam prenese v svoj sistem. Pri prejemanju dokumentov po tem protokolu pa je ravno obratno. Naš sistem EBA DMS preveri, ali v distribucijskem centru EBA obstajajo neprevzeti dokumenti, ki so jih tja poslali naši zunanji partnerji, ki imajo tudi elektronski dokumentni sistem EBA DMS. Če neprevzeti dokumenti obstajajo, se le-ti prenesejo v naš sistem EBA DMS.

3.3.2 Elektronska pošta

Za elektronsko izmenjavo dokumentov preko elektronske pošte imamo na voljo dva protokola, in sicer enega za prejemanje in enega za pošiljanje dokumentov. V obeh primerih se sistem EBA DMS samo poveže na zunanji poštni strežnik. Prek protokola IMAP Email Exchange se izvaja prejemanje dokumentov, preko protokola SMTP Email Exchange pa pošiljanje dokumentov.

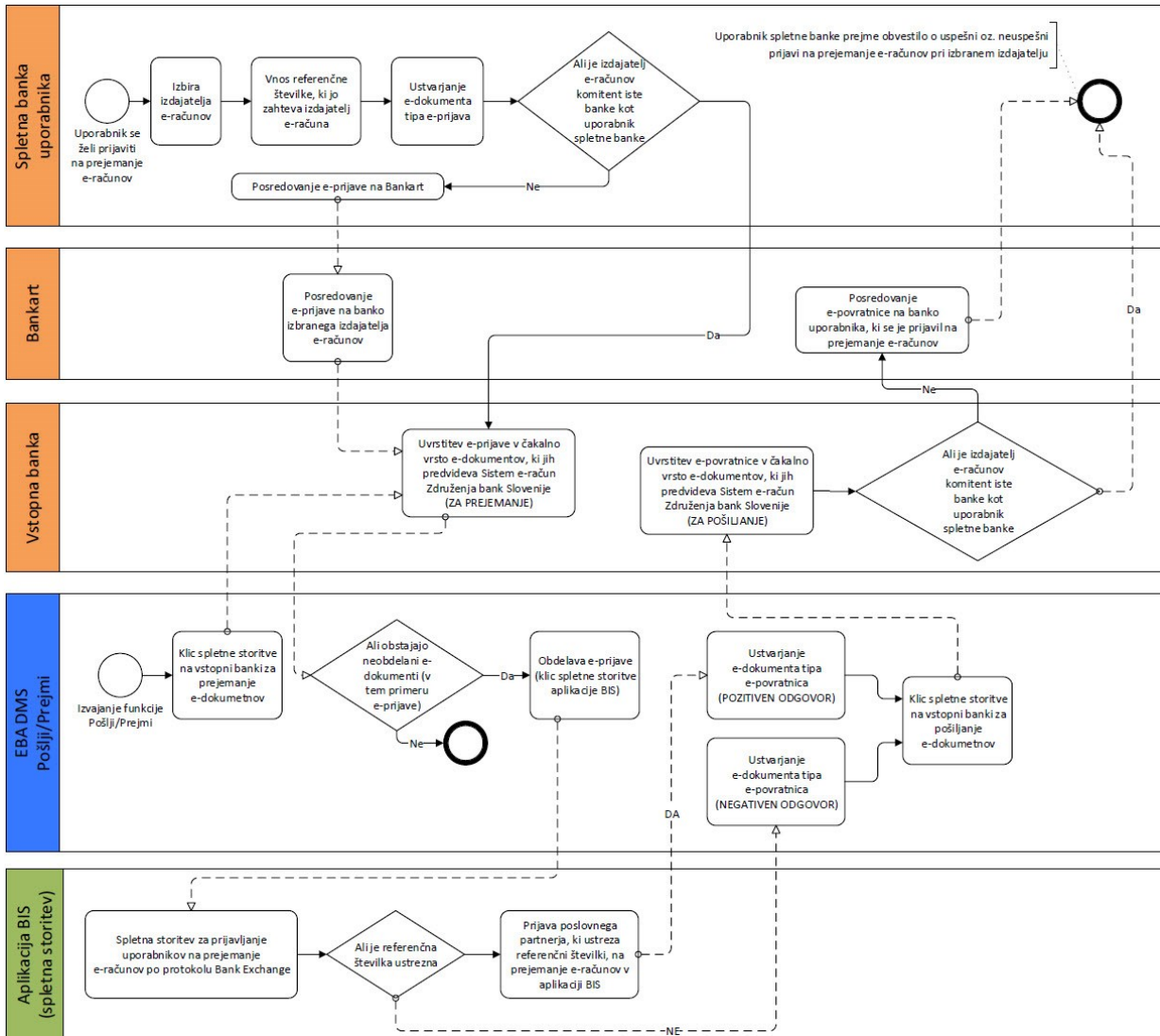
Vhodne dokumente lahko zajemamo iz poljubno mnogo elektronskih naslovov, ki so definirani na zunanjem poštnem strežniku, medtem ko lahko izhodne dokumente pošiljamo s samo enega naslova, ki smo ga sami določili za pošiljanje.

3.3.3 Bank Exchange

Preko protokola Bank Exchange v podjetju poteka elektronska izmenjava e-računov. Sistem e-račun Združenja bank Slovenije poleg e-računov predvideva še več tipov e-dokumentov, ki se izmenjujejo med pošiljatelji, vmesnimi člani prenosnega sistema in prejemniki e-računov. Tipi e-dokumentov, ki se izmenjujejo, so:

- ovojnica e-računa,
- e-račun,
- splošna ovojnica,
- e-prijava in e-odjava o prejemanju e-računov,
- e-povratnica,
- dostavnica.

Protokol Bank Exchange zagotavlja dodatno razvit vtičnik, ki je vgrajen v sistem EBA DMS. Ta preko spletnih storitev izmenjuje vse zgoraj navedene datoteke z eno izmed slovenskih bank, preko katere je naprej zagotovljena pot do vseh ostalih bank preko Bankarta.



Slika 11: e-Prijava in e-Povratnica, povezava medbančnega sveta z EBA DMS in interno aplikacijo BIS

3.3.4 UJP Exchange

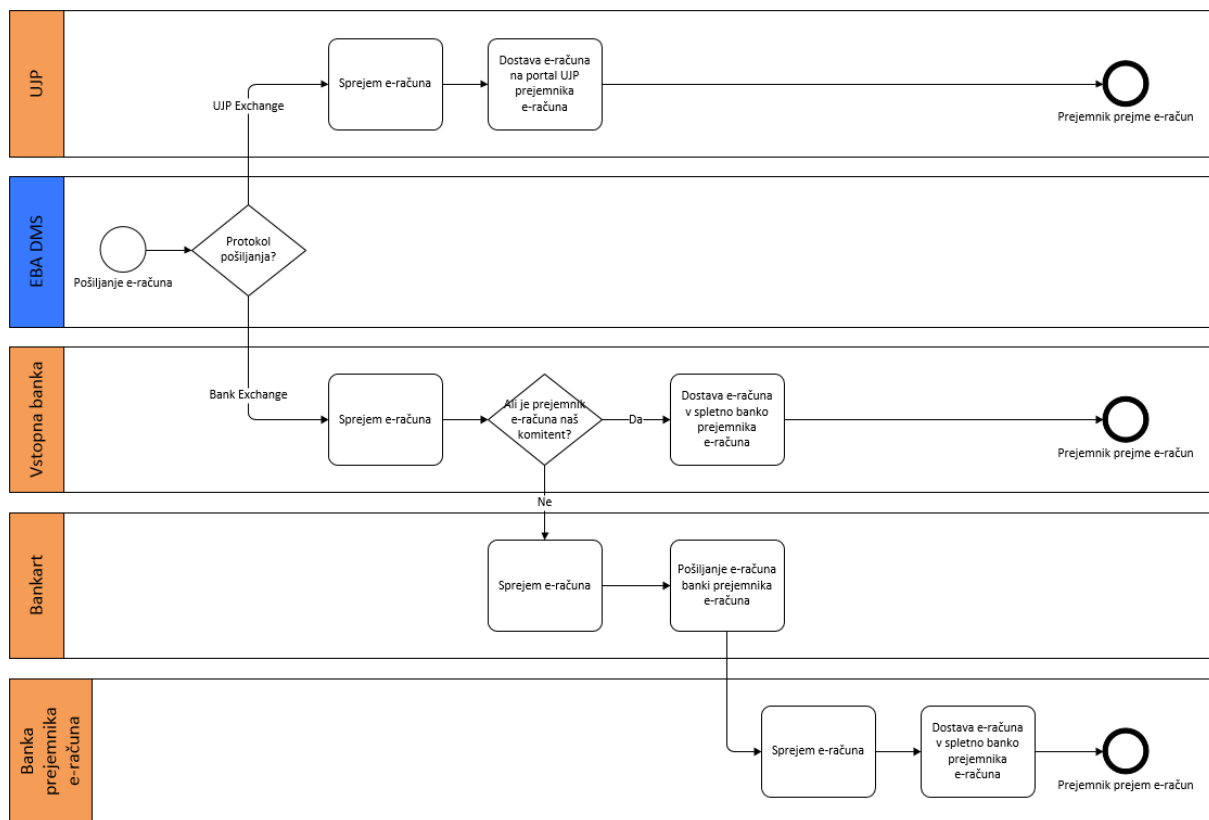
Za izmenjavo e-računov in ostalih dokumentov samo in neposredno s proračunskimi uporabniki (z Upravo Republike Slovenije za javna plačila ali kratko UJP) imamo v sistemu EBA DMS na voljo dodaten vtičnik, ki nam zagotavlja protokol UJP Exchange. Prednost tega protokola pred protokolom Bank Exchange je samo v tem, da izmenjava podatkov oz. datotek poteka neposredno med sistemom EBA DMS in sistemom UJP. Neposredna izmenjava pomeni, da

praktično ni časovnega zamika, med trenutkom, ko EBA DMS podatke oz. dokumente pošlje, in trenutkom, ko jih na drugi strani nek proračunski uporabnik prejme.

Za razliko od neposredne izmenjave na strani protokola UJP Exchange pa tega ne moremo trditi za protokol Bank Exchange, kjer je časovni zamik kar precejšen.

Pot, ki jo opravi e-račun od pošiljatelja iz sistema EBA DMS do prejemnika v spletno banko, je naslednja. Če je prejemnik e-računa komitent naše vstopne banke, potem je e-račun v spletni banki prejemnika razmeroma hitro. V nasprotnem primeru je e-račun iz naše vstopne banke najprej poslan na Bankart, od tam pa naprej na banko prejemnika e-računa. Šele v naslednjem koraku pa e-račun prejme njegov prejemnik. Potrditev o uspešni dostavi e-računa mora prepotovati ravno obratno pot, kot jo je opravil e-račun.

Velik časovni zamik od pošiljanja e-računa do prejema potrditve o uspešni dostavi nastane predvsem zaradi dejstva, da se obdelave na bankah in Bankartu izvajajo samo nekajkrat dnevno.



Slika 12: Primerjava pošiljanja e-računa med protokolom Bank Exchange in UJP Exchange

3.4 Načini povezovanja s poslovno informacijskimi sistemi

V elektronskem dokumentnem sistemu EBA DMS imamo na voljo več različnih načinov integracije z ostalimi sistemi in aplikacijami.

Integracija je najpogosteje izvedena prek programske knjižnice EBAXCOM, ki je objektna programska knjižnica, napisana v tehnologiji COM. Knjižnica omogoča, da se informacijski sistem in dokumentni sistem zlijeta v eno.

Drugi način povezovanja je orodje EBA Developer, ki je lastno razvojno orodje dokumentnega sistema EBA DMS. Z orodjem lahko sami, v programskem jeziku JavaScript, razvijamo dodatne module in specifične funkcionalnosti, ki jih želimo vgraditi v sistem EBA DMS.

Tretji način povezovanja je neposredna povezava na podatkovne baze zunanjih sistemov, npr. izbira vrednosti nekega šifrant, ki je sicer shranjen v podatkovni bazi zunanjega sistema.

3.4.1 Vmesnik EBAXCOM

EBAXCOM je programska knjižnica, napisana v tehnologiji COM, in nam omogoča, da v zunanje sisteme vgradimo funkcionalnosti, ki jih sicer ponuja elektronski dokumentni sistem EBA. V programski knjižnici so nam na voljo vse funkcije za delo z dokumenti, kot so sicer na voljo v razvojnem okolju EBA Developer.

3.4.2 Razvojno okolje EBA Developer

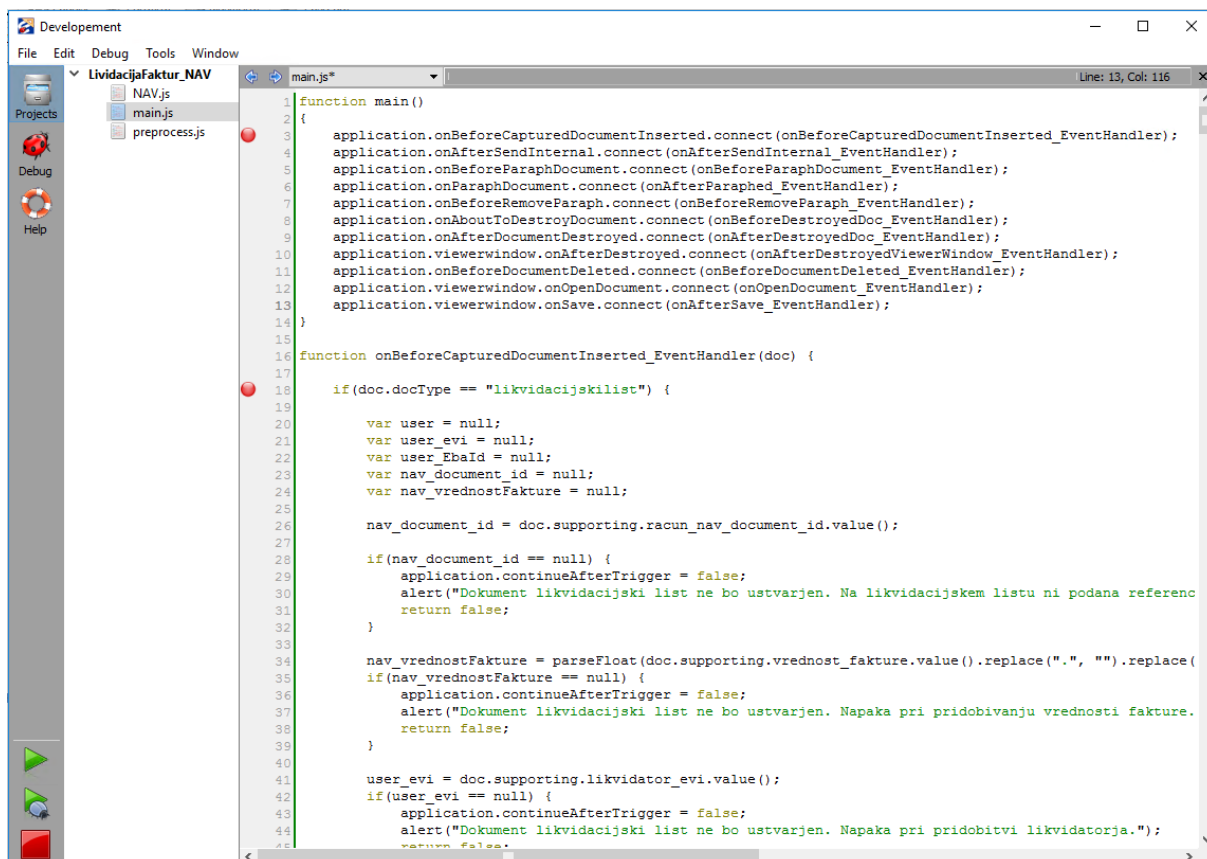
Razvojno okolje EBA Developer je lastno razvojno okolje dokumentnega sistema EBA DMS. Omogoča lasten razvoj kompleksnih projektov, kot so integracije z drugimi informacijskimi sistemi in razširitve sistema EBA DMS. S projektom, razvitim v razvojnem okolju EBA Developer, lahko sistem EBA DMS razširimo z lastnimi specifičnimi funkcionalnostmi, ki jih potrebujemo za podporo poslovnim procesom. Razvite projekte uvozimo neposredno v sistem EBA DMS in tako postanejo neločljiv del le-tega. Sam razvoj projektov v okolju EBA Developer poteka v programskem jeziku JavaScript.

Pri razvoju imamo na voljo velik nabor knjižnic, preko katerih je omogočeno delo z bazami podatkov, delo s spletom in mrežnimi protokoli, delo z grafičnimi gradniki in pripravo grafičnih

vmesnikov, delo z datotekami XML, delo s spletnimi storitvami (SOAP), delo z internimi EBA DMS objekti in dokumenti [14].

Izbiramo lahko med tremi glavnimi tipi predlog projektov:

- predloga projekta za navaden skriptni projekt,
- predloga projekta za razvoj vtičnikov,
- predloga projekta za generiranje obrazcev.



Slika 13: Primer projekta v razvojnem okolju EBA Developer

3.4.2.1 Predloga projekta za navaden skriptni projekt

Navaden skriptni projekt nam omogoča najširši nabor možnosti za razvoj lastnih specifičnih funkcionalnosti sistema EBA DMS.

Sistem EBA DMS lahko nadgradimo s popolnoma novimi elementi kot so:

- novi moduli uporabniškega vmesnika in pripadajočo programsko kodo,

- spletne storitve,
- poljubne funkcije (ki jih na primer storitev za izvajanje opravil na strežniku zaganja po nekem definiranem časovnem urniku).

Druga zelo uporabna možnost je, da lahko z dodatnimi funkcionalnostmi razširjamo popolnoma osnovne funkcionalnosti sistema EBA DMS. Na voljo imamo zelo velik nabor signalov, na katere se lahko povežemo in v sistem EBA DMS vgradimo nam lastno logiko. V splošnem imamo na voljo signale, ki se prožijo pred ali pa po dogodkih, ki se zgodijo v sistemu EBA DMS.

Za primer si oglejmo dva signala, ki sta vezana na dogodek parafiranja dokumenta. Signal `onBeforeParaphDocument()`, ki se sproži tik preden se izvede dodajanje parafa na dokument in signal `onParaphDocument()`, ki se izvede tik za izvedenim parafiranjem dokumenta. Pri prvem signalu lahko npr. dodamo logiko preverjanja ali je uporabnik, ki izvaja akcijo parafiranja, sploh na vrsti za prafiranje. Če je, pustimo, da se postopek parafiranja izvaja naprej, če pa ni, pa lahko izvajanje postopka parafiranja prekinemo in uporabniku javimo opozorilo z navedeno napako. Na drugem signalu, ki se izvede po že dodanem parafu na dokumentu, pa npr. dodamo logiko za preverjanje ali je dokument že parafiran s strani vseh oseb, ki so bile določene za parafiranje dokumenta. V primeru, da je, lahko npr. izvedemo akcijo posodabljanja nekega statusa v zunanjem informacijskem sistemu bodisi preko spletne storitve ali pa kar neposredno preko podatkovne baze.

3.4.2.2 Predloga projekta za razvoj vtičnikov

Projekt za razvoj vtičnikov nam omogoča integracijo elektronskega dokumentnega sistema EBA DMS s poljubnim zunanjim sistemom, ki komunicira po poljubnem protokolu za izmenjavo podatkov.

Preko vtičnikov je v sistemu EBA DMS omogočeno komuniciranje s sistemi, ki podpirajo protokole kot so AS2, IMAP in SMTP.

V celoti se preko dveh vtičnikov v sistemu EBA DMS izvaja tudi izmenjava e-računov. Eden izmed vtičnikov skrbi za izmenjavo e-računov s sistemom UJP (vtičnik UJP Exchange Plugin), drugi pa skrbi za izmenjavo e-računov med izdajatelji in prejemniki prek t. i. (med)bančne infrastrukture (vtičnik Bank Exchange Plugin).

3.4.2.3 Predloga projekta za generiranje obrazcev

Projekt za generiranje obrazcev nam omogoča, da v grafičnem vmesniku oblikujemo poljuben obrazec za vnos in/ali predstavitev podatkov. Obrazec nam v sistemu EBA DMS služi kot ena izmed možnih predlog za generiranje novih dokumentov. Ob shranjevanju obrazca se obrazec pretvori v obliko PDF, ki se uporabi za prikaz dokumenta.

V primeru vtičnika »Bank Exchange Plugin« se npr. preko obrazcev prejeti oz. poslani dokumenti, ki se sicer pošiljajo v obliki XML (e-prijava, e-odjava, e-povratnica) v sistemu EBA DMS shranijo oz. predstavijo v uporabniku prijaznejši obliki.

```

<?xml version="1.0" encoding="UTF-8"?>
- <InvoiceSubscription xmlns="hal:icl:01" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <MsgID>
    <DocID>54663</DocID>
    <DocTime>2016-02-11T08:43:35.705125+01:00</DocTime>
    <Version>1.0</Version>
  </MsgID>
  - <RcpDtls>
    <RcpIBAN>[redacted]</RcpIBAN>
    <RcpTaxID>[redacted]</RcpTaxID>
    <RcpType>FO</RcpType>
    <RcpName>[redacted]</RcpName>
    <RcpAddr>[redacted]</RcpAddr>
    <RcpCity>[redacted]</RcpCity>
    <RcpEmail>[redacted]</RcpEmail>
    <RcpTel>[redacted]</RcpTel>
  </RcpDtls>
  - <IssDtls>
    <IssIBAN>[redacted]</IssIBAN>
    <IssTaxID>[redacted]</IssTaxID>
    <IssName>[redacted]</IssName>
    <IssAddr>[redacted]</IssAddr>
    <IssCity>[redacted]</IssCity>
  </IssDtls>
  - <SubDtls>
    <InvRefID>256928-151867728</InvRefID>
    <SubType>unsubscribe</SubType>
    <BankTrms>Niso na voljo</BankTrms>
    <IssuerTrms>[redacted]</IssuerTrms>
  </SubDtls>
</InvoiceSubscription>

```

e-Prijava / e-Odjava

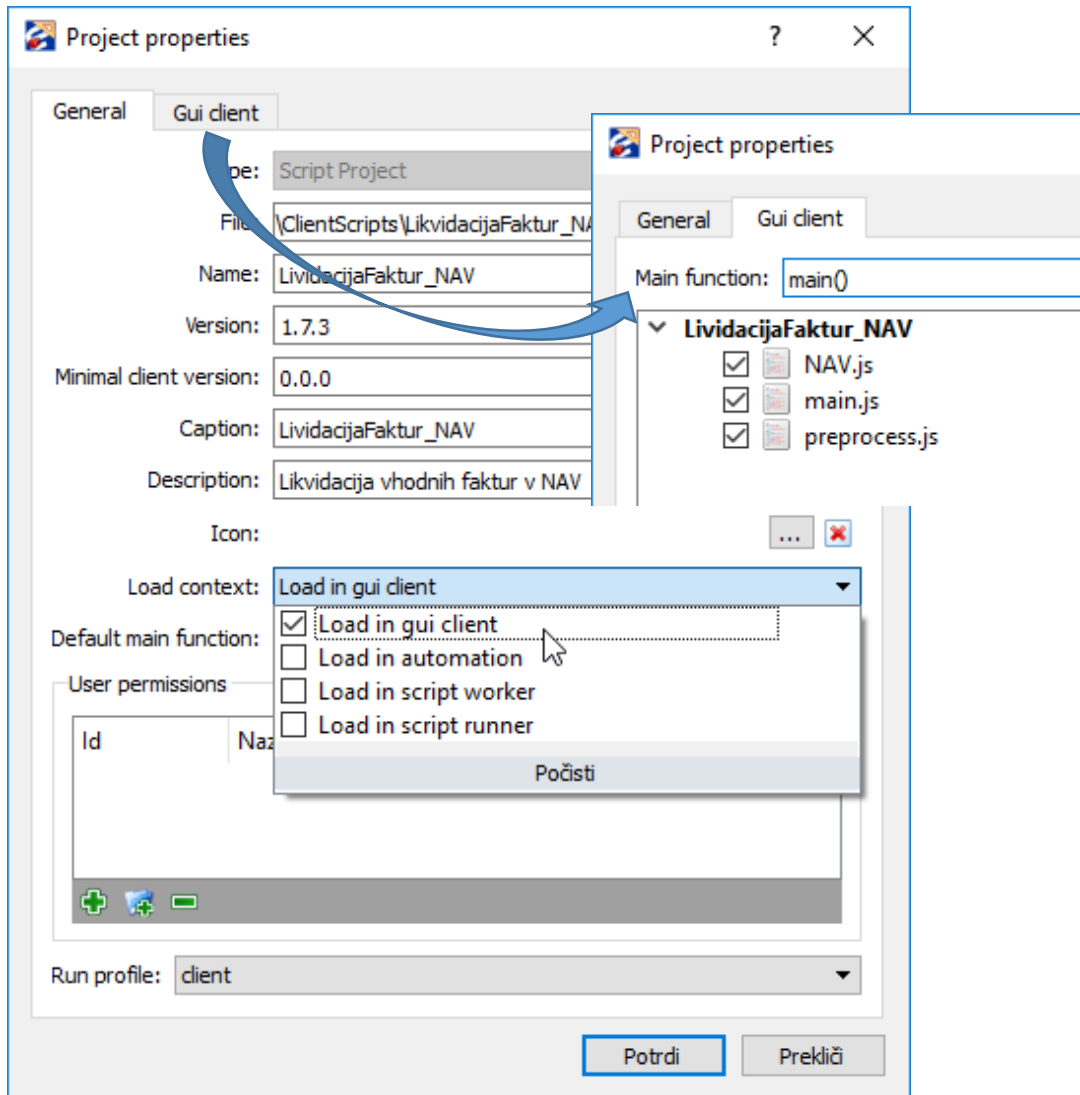
Tip zahteve:	unsubscribe
Datum in čas:	2016-02-11T08:43:35.705125+01:00
BIC banke:	[redacted]
TRR:	[redacted]
Referenčna oznaka	256928-151867728

Slika 14: Primer uporabe obrazcev (pretvorba dokumenta e-odjava iz oblike XML v obliko PDF)

3.4.2.4 Objava projekta v sistem EBA DMS

Po zaključenem razvoju projekta je potrebno za uspešno objavo projekta v sistemu EBA DMS projekt pravilno pripraviti za objavo. Izbrati je potrebno okolja, v katerih se bo projekt ob

zagonu okolja sploh naložil in bo kasneje na voljo tudi za izvajanje. Za vsako od izbranih okolij potem v dodatnem zavihku, ki se prikaže po izbiri, izberemo datoteke iz projekta, katere želimo, da so na voljo za izvajanje v izbranem okolju.

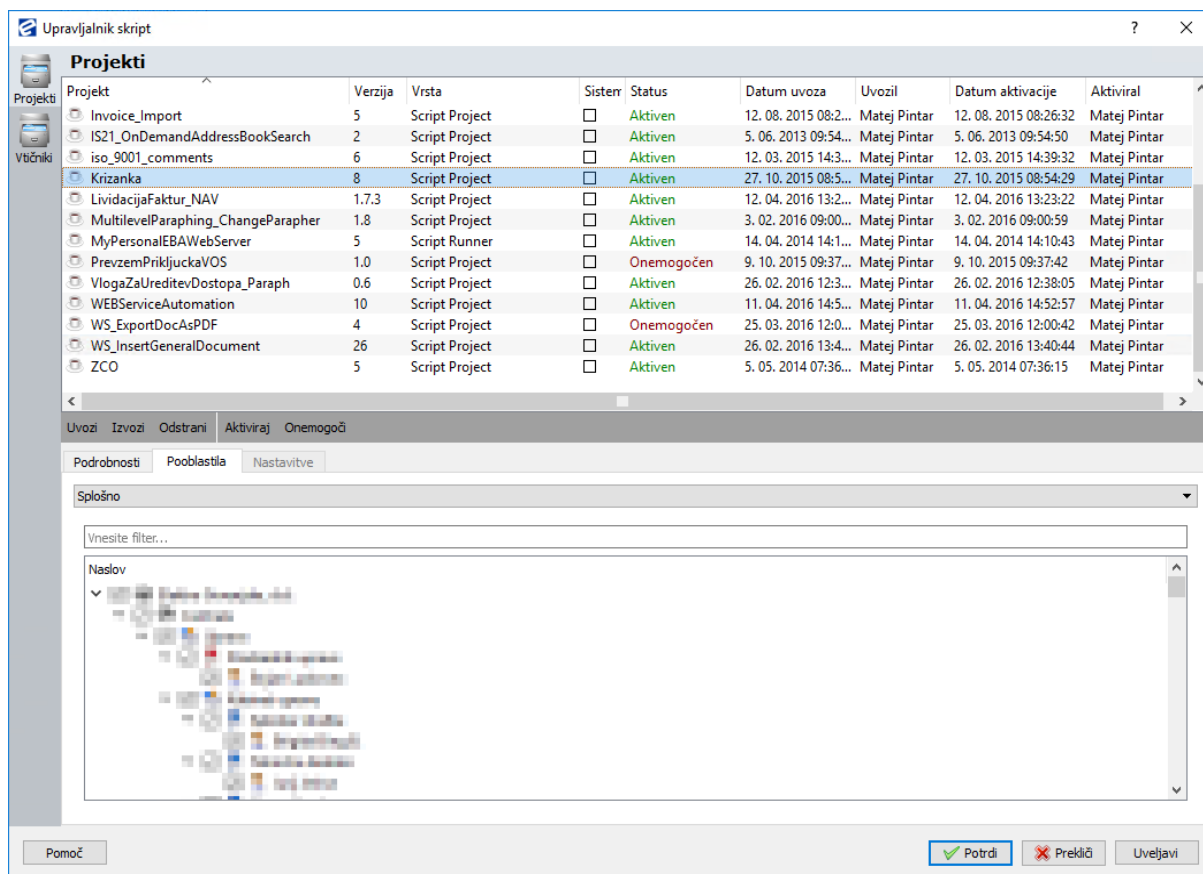


Slika 15: Nastavitve projekta v razvojnem okolju EBA Developer

Za izvajanje projektov imamo na voljo naslednja okolja:

- odjemalec EBA,
- strežniški del za gostovanje spletnih storitev,
- strežniški del za izvajanje opravil,
- poljubno izvajanje programske kode (npr. kar iz ukazne vrstice).

Ko je projekt pripravljen za objavo v sistemu EBA DMS, preko orodja za objavo projekta pripravimo namestitveni paket. Programsko kodo projekta lahko kriptiramo, namestitveni paket pa še podpišemo z digitalnim potrdilom. Ko je namestitveni paket pripravljen, le-tega uvozimo v sistem EBA DMS preko modula za upravljanje s projekti v klientu EBA.



Slika 16: Modul za upravljanje s projekti razvitimi v programskem okolju EBA Developer

Ob uvozu moramo za vsak projekt določiti tudi pooblastila za izvajanje projektov. Le-ta določajo, katerim uporabnikom se bo projekt ob zagonu okolja sploh naložil in bo kasneje na voljo za izvajanje. Projekt dokončno objavimo in je na voljo za izvajanje, ko ga aktiviramo. V posameznem okolju, ki je bil na projektu izbran za izvajanje, pa bo projekt na voljo za izvajanje šele, ko bomo izbrano okolje ponovno zagnali, saj se projekti nalagajo le ob zagonu okolij.

3.5 Povzetek

V predhodnih poglavjih je bilo prikazano, da je elektronski dokumentni sistem EBA DMS res celovita rešitev za brezpapirno poslovanje v podjetjih in organizacijah ter da je v vseh pogledih

zelo odprt in prilagodljiv sistem. Od tega, da pooblašteni uporabniki lahko sami prilagajajo sistem in njegovo delovanje z npr. določanjem novih tipov dokumentov, atributov tipov dokumentov, pravil nad dokumenti, do tega, da sistem EBA DMS razvijalcem ponuja širok nabor različnih možnosti za integriranje sistema z drugimi informacijskimi sistemi in aplikacijami.

Predstavljeni so bili tudi različni načini elektronske izmenjave dokumentov, ki jih sistem ponuja. Kot smo pa videli, pa lahko tudi sami in kadarkoli z razvojem novega vtičnika v sistemu EBA DMS podpremo nov način elektronske izmenjave dokumentov.

V naslednjih treh poglavjih bodo predstavljeni konkretni načini integracije različnih informacijskih sistemov in aplikacij z elektronskim dokumentnim sistemom EBA

4 Integracija Dynamics NAV s sistemom EBA DMS

Microsoft Dynamics NAV je celovita programska rešitev (angl. Enterprise Resource Planning, ERP), ki jo v podjetju uporabljamo kot glavni informacijski sistem.

Informacijski sistem Dynamics NAV v osnovi ni integriran z elektronskim dokumentnim sistemom EBA DMS, zato smo morali v podjetju ob vpeljavi sistema Dynamics NAV poskrbeti za medsebojno integracijo obeh sistemov.

Na strani sistema Dynamics NAV je bila za integracijo s sistemom EBA DMS uporabljena programska knjižnica EBAXCOM. Za integracijo z nasprotne strani, torej s strani sistema EBA DMS, pa je bilo poskrbljeno z razvojem modula NAV v orodju EBA Developer, ki je implementiran v sistem EBA DMS.

V nadaljevanju bosta najprej predstavljena oba načina integracije, celovito pa bo integracija obeh sistemov prikazana na primeru poslovnega procesa likvidacije vhodnih računov. Predstavljena bo obravnava prejetega računa vse od prejema računa v podjetje pa vse do knjiženja tega računa oz. do izvedbe postopka za zavrnitev računa izdajatelju.

4.1 Sistem za upravljanje z dokumenti v sistemu Dynamics NAV

Privzeti del produkta Microsoft Dynamics NAV je tudi že sistem za upravljanje z dokumenti (angl. Document Management System, DMS). Vendar pa ta integrirani sistem za upravljanje z dokumenti ne nudi vseh funkcionalnosti, ki jih nudi sodobni sistem za upravljanje z dokumenti. Zato smo se v podjetju odločili, da bomo obdržali naš obstoječ elektronski dokumentni sistem EBA DMS. Pogoj za uspešno vpeljavo novega informacijskega sistema Dynamics NAV pa je s tem postala tesna integracija s sistemom EBA DMS.

Kljub odločitvi za integracijo z zunanjim elektronskim dokumentnim sistemom EBA DMS internega sistema za upravljanje z dokumenti, ki je del sistema Dynamics NAV, nismo zavrgli. Prav nasprotno, vse njegove funkcionalnosti so bile ohranjene. V primerih, kjer se sistem Dynamics NAV povezuje s sistemom EBA DMS, je prav ta interni sistem za upravljanje z dokumenti uporabljen kot vezni člen med obema sistemoma. V obeh sistemih je bilo uvedeno novo polje, ki se uporablja v integraciji. Na strani sistema Dynamics NAV je bilo dodano polje

ID dokumenta EBA, ki predstavlja ID dokumenta iz sistema EBA DMS, na strani sistema EBA DMS pa je bilo na dokumentu dodano polje Številka dokumenta NAV, ki predstavlja ID dokumenta v sistemu za upravljanje z dokumenti v sistemu Dynamics NAV.

V zunanji elektronski dokumentni sistem EBA DMS so bile iz sistema Dynamics NAV predstavljene naslednje funkcije:

- potrjevanje dokumentov,
- pošiljanje e-računov,
- arhiviranje dokumentov,
- avtomatizacija zajema vhodnih dokumentov.

Dokumenti so v internem sistemu za upravljanje z dokumenti opremljeni s širokim naborom metapodatkov. Za integracijo s sistemom EBA DMS so pomembni zgolj trije:

- številka postavke (ID dokumenta),
- šifra vrste dokumenta DMS,
- status odobritve.

4.1.1 Vrste dokumentov DMS

Sistem Dynamics NAV dokumente razlikuje in različno obravnava glede na vrste dokumentov DMS. Za vsako vrsto dokumentov DMS obstajajo nastavitve, ki določajo, kako bo sistem obravnaval dokumente, ki so uvrščeni v to vrsto.

Izmed vseh nastavitvev na posameznem tipu dokumenta so za integracijo z zunanjim dokumentnim sistemom EBA DMS najpomembnejše naslednje:

- Vrsta pomnilnika za prilogo,
- Tip dokumenta v integraciji,
- Zahteva odobritev EBA.

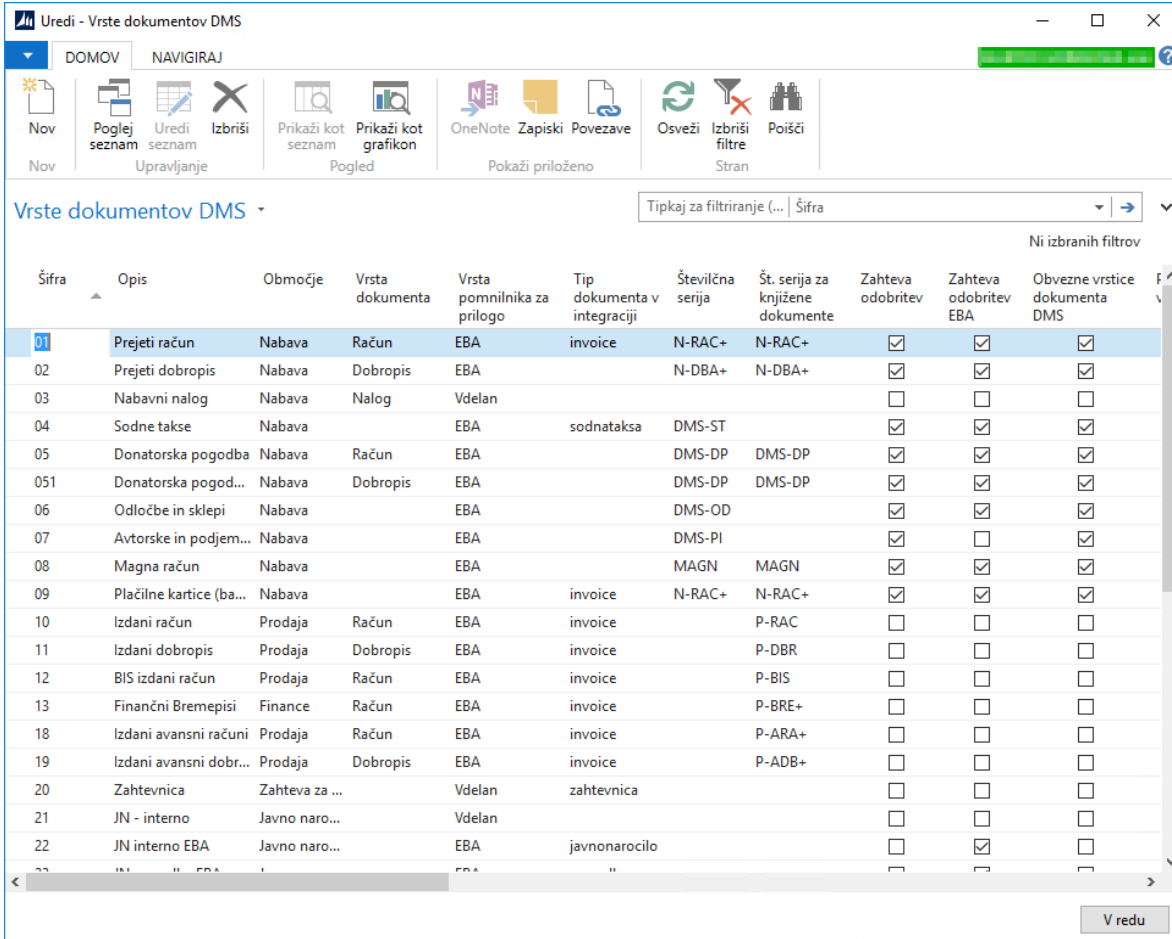
Polje Vrsta pomnilnika za prilogo določa mesto, kjer bodo dokumenti posamezne vrste dokumentov DMS shranjeni. Če je za posamezno vrsto dokumentov DMS vrednost nastavljena na EBA pomeni, da se bodo dokumenti te vrste hranili v zunanjem dokumentnem sistemu EBA DMS. V primeru, ko je pa vrednost enaka Vdelan, pa to pomeni, da se dokument ne bo izvozil

v zunanji dokumentni sistem, ampak bo shranjen v internem dokumentnem sistemu sistema Dynamics NAV.

Polje Tip dokumenta v integraciji je povezovalno polje med sistemom Dynamics NAV in sistemom EBA DMS. Vrednost polja za posamezno vrsto dokumentov DMS določa, v kateri tip dokumenta v sistemu EBA DMS se bodo dokumenti te vrste iz sistema Dynamics NAV preslikali.

Polje Zahteva za odobritev EBA določa, ali bo nad dokumenti te vrste dokumentov DMS potrebno izvesti postopek potrjevanja v sistemu EBA DMS.

Vsak dokument, ki nastane v sistemu Dynamics NAV, se najprej shrani v internem sistemu za upravljanje z dokumenti DMS sistema Dynamics NAV. Šele nato se v enem izmed naslednjih korakov, če je nastavitev Vrsta pomnilnika za prilogo za vrsto dokumenta DMS, v katero je dokument uvrščen, nastavljena na EBA, prenese v zunanji dokumentni sistem EBA DMS.



Šifra	Opis	Območje	Vrsta dokumenta	Vrsta pomnilnika za prilogo	Tip dokumenta v integraciji	Številčna serija	Št. serija za knjižene dokumente	Zahteva odobritev	Zahteva odobritev EBA	Obvezne vrstice dokumenta DMS
01	Prejeti račun	Nabava	Račun	EBA	invoice	N-RAC+	N-RAC+	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
02	Prejeti dobropis	Nabava	Dobropis	EBA		N-DBA+	N-DBA+	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
03	Nabavni nalog	Nabava	Nalog	Vdelan				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
04	Sodne takse	Nabava		EBA	sodnataksa	DMS-ST		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
05	Donatorska pogodba	Nabava	Račun	EBA		DMS-DP	DMS-DP	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
051	Donatorska pogod...	Nabava	Dobropis	EBA		DMS-DP	DMS-DP	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
06	Odločbe in sklepi	Nabava		EBA		DMS-OD		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
07	Avtorske in podjem...	Nabava		EBA		DMS-PI		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
08	Magna račun	Nabava		EBA		MAGN	MAGN	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
09	Plačilne kartice (ba...	Nabava		EBA	invoice	N-RAC+	N-RAC+	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	Izdani račun	Prodaja	Račun	EBA	invoice		P-RAC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11	Izdani dobropis	Prodaja	Dobropis	EBA	invoice		P-DBR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12	BIS izdani račun	Prodaja	Račun	EBA	invoice		P-BIS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13	Finančni Bremepisi	Finance	Račun	EBA	invoice		P-BRE+	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
18	Izdani avansni računi	Prodaja	Račun	EBA	invoice		P-ARA+	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19	Izdani avansni dobr...	Prodaja	Dobropis	EBA	invoice		P-ADB+	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20	Zahtevnica	Zahteva za ...		Vdelan	zahtevnica			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
21	JN - interno	Javno naro...		Vdelan				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
22	JN interno EBA	Javno naro...		EBA	javnonarocilo			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Slika 17: Prikaz strani za urejanje vrst dokumentov DMS

4.1.2 Dovoljenja vrst dokumentov DMS

Za avtomatsko dodajanje dovoljenj na dokumente, ki so shranjeni v sistemu EBA DMS, je v prvi vrsti poskrbljeno v sistemu EBA DMS preko pravil. Vendar pa pravila lahko definira in z njimi upravlja samo administrator sistema EBA DMS.

Z modulom Dovoljenja, ki je del sistema za upravljanje z dokumenti v sistemu Dynamics NAV pa pravila, ki veljajo tudi v sistemu EBA DMS, definiramo že na strani Dynamics NAV.

V modulu Dovoljenja lahko za vsako od vrst dokumentov DMS definiramo dovoljenja iz sistema EBA DMS, ki se avtomatsko dodajajo na dokumente te vrste v sistemu EBA DMS. Definirana dovoljenja se dodajo tako na dokumente, ki jih sistem Dynamics NAV na novo ustvari v sistemu EBA DMS, kot tudi na dokumente, ki jih Dynamics NAV uvozi iz sistema EBA DMS v svoj interni sistem za upravljanje z dokumenti.

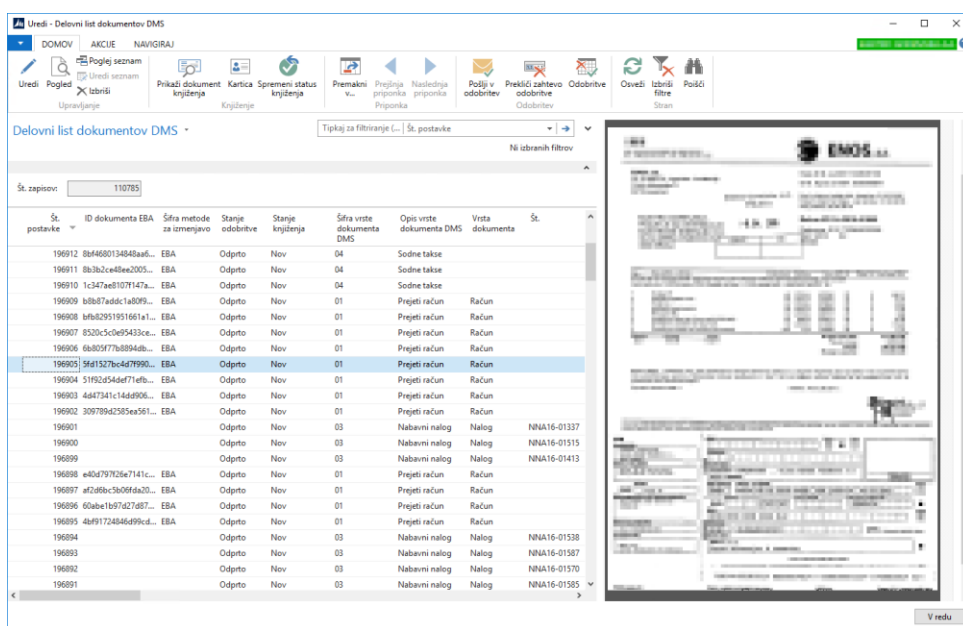
Prednost drugega načina je, da uporabniku, ki upravlja z dovoljenji na strani sistema Dynamics NAV, ni potrebno biti administrator v sistemu EBA DMS.

Šifra vr...	Ime	Ime dovoljenja
01	arhivvhodnihračunov	1014
04	sodnataksa	1021
10	arhivizhodnihračunov	974
11	arhivizhodnihračunov	974
35	arhivvhodnihračunov	1014
50	arhivodpisovosdi	1044
51	arhivodpisovosdi	1044
52	arhivodpisovosdi	1044
53	arhivodpisovosdi	1044
54	arhivodpisovosdi	1044
55	arhivodpisovosdi	1044
60	pogodbazaizplacilo	1022
62	prevzempriljuckavos	1048
70	pogodbazaizplacilo	1022

Slika 18: Prikaz strani za urejanje dovoljenj vrst dokumentov DMS

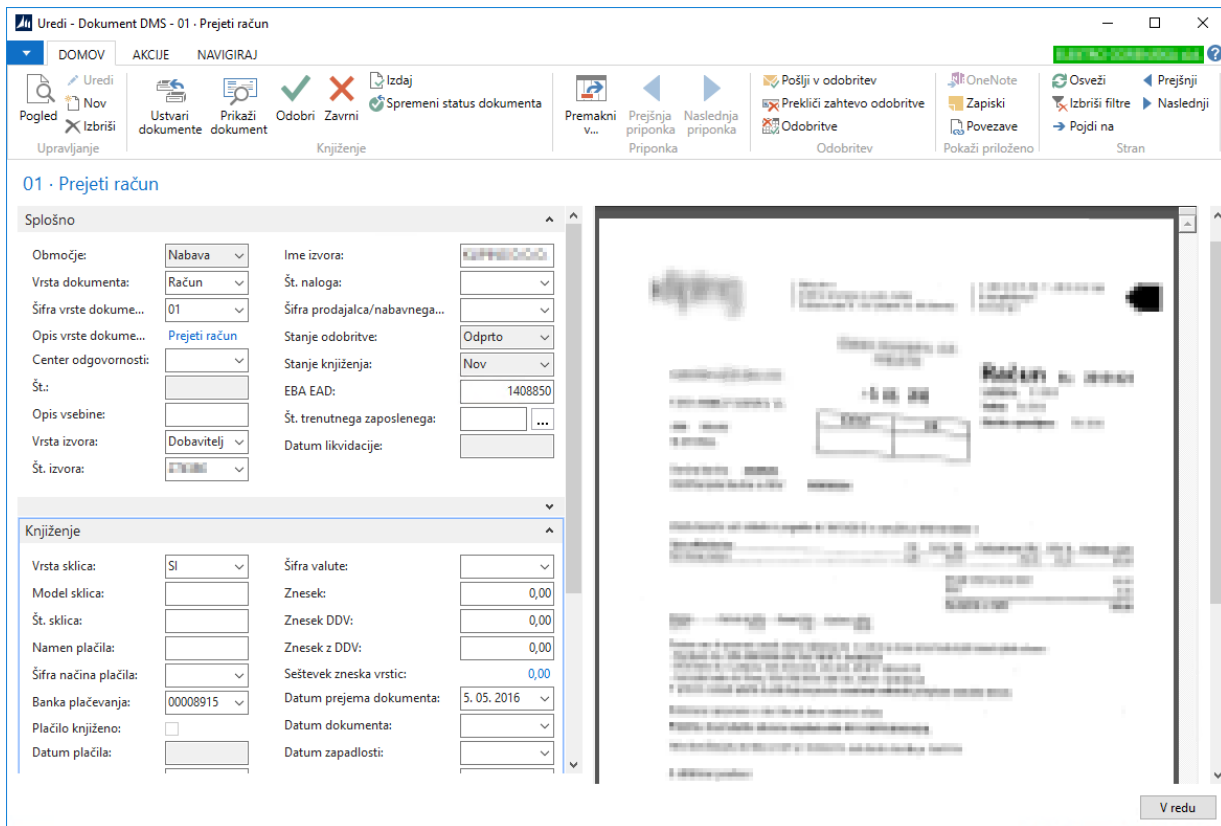
4.1.3 Gradnik za pregled dokumentov iz sistema EBA DMS

V želji po čim bolj tesni integraciji obeh sistemov, smo v sistemu Dynamics NAV interni sistem za upravljanje z dokumenti preko programske knjižnice EBAXCOM razširili z novim gradnikom, ki prikazuje dokumente, shranjene v sistemu EBA DMS, v uporabniškem vmesniku sistema Dynamics NAV. Gradnik za pregled dokumentov iz sistema EBA DMS za delovanje zahteva, da je uporabnik v ozadju prijavljen tudi v sistem EBA DMS. Gradnik uporabnikom bistveno olajša delo, saj jim za učinkovito delo ni potrebno menjati pogleda med dvema sistemoma.



Slika 19: Sistem za upravljanje z dokumenti v Dynamics NAV z gradnikom za pregled dokumentov iz sistema EBA DMS

V glavnem oknu internega sistema za upravljanje z dokumenti je v osnovi prikazan samo seznam dokumentov, uporabniki pa si lahko glavno okno prilagodijo tako, da si na desno stran okna dodajo gradnik za pregled dokumentov. Sistem Dynamics NAV ob omogočenem gradniku uporabniku v njem prikazuje tisti dokument, ki je trenutno izbran v seznamu dokumentov na levi strani. Prav tako si lahko uporabniki dodajo gradnik za pregled dokumenta na okno za urejanje dokumenta.



Slika 20: Kartica dokumenta z gradnikom za pregled dokumentov iz sistema EBA DMS

4.1.4 Opravilo za izmenjavo dokumentov med EBA DMS in Dynamics NAV

EBA Management je eno izmed opravil sistema Dynamics NAV in skrbi za dvoje:

1. uvoz novih neprenesenih vhodnih dokumentov iz sistema EBA DMS v sistem Dynamics NAV,
2. izvoz novih neprenesenih dokumentov iz sistema Dynamics NAV v sistem EBA DMS.

Slika 21: Opravilo za izmenjavo dokumentov med EBA DMS in Dynamics NAV

Z nastavitvami vrst dokumentov DMS določamo, katere vrste oz. tipe dokumentov bo opravilo EBA Management prenašalo in katerih ne.

V nadaljevanju bosta podrobneje predstavljeni obe smeri prenosa dokumentov.

4.1.4.1 Uvoz novih dokumentov iz sistema EBA DMS

V tem primeru gre največkrat za dokumente finančne narave, ki morajo biti obravnavani v sistemu Dynamics NAV. Ne glede na to, kako so bili dokumenti vloženi v sistem EBA DMS (ne glede na protokol), sistem EBA DMS s funkcionalnostjo avtomatske prepoznave podatkov omogoča, da so nekateri podatki iz dokumentov avtomatsko prepoznani in shranjeni že ob zajemu v sistem EBA DMS. Kasneje, ko opravilo EBA Management v sistemu Dynamics NAV izvaja uvoz novih dokumentov iz sistema EBA DMS, poleg osnovnih podatkov (datum prejema, številka poslovnega partnerja ...), opravilo prenese tudi vse avtomatsko prepoznane

podatke. Če je bila prepoznavna pomanjkljiva ali neuspešna, je potrebno podatke za nadaljnjo obdelavo dokumenta dopolniti samo v sistemu Dynamics NAV.

Uvoz dokumentov poteka zaporedno, po en dokument naenkrat. Opravilo EBA Management med uvoženimi in neuvoženimi dokumenti iz sistema EBA DMS razlikuje po izpoljenosti polja Številka dokumenta NAV na dokumentih v sistemu EBA DMS. Vedno se uvozijo vsi tisti dokumenti, ki imajo polje Številka dokumenta NAV prazno. Uspešen uvoz ima v sistemu Dynamics NAV za posledico nov dokument v internem sistemu za upravljanje z dokumenti. Poleg vsebinskih podatkov se iz sistema EBA DMS prenese še podatek ID dokumenta EBA, ki je ID dokumenta v sistemu EBA DMS. Preko tega podatka se kasneje izvaja prikazovanje dokumenta v gradniku za prikazovanje dokumentov v sistemu Dynamics NAV iz sistema EBA DMS in pa za odpiranje dokumentov iz sistema Dynamics NAV neposredno v klientu EBA. Uvoz posameznega dokumenta v sistem Dynamics NAV se zaključi, ko opravilo EBA Management v sistemu EBA DMS na dokument v polje Številka dokumenta NAV zapiše ustrezno številko dokumenta iz internega sistema za upravljanje z dokumenti v sistemu Dynamics NAV. S tem podatkom je v prvi vrsti zagotovljeno, da se dokument v enem od naslednjih zagonov opravila EBA Management ne bo ponovno uvozil v sistem Dynamics NAV. V nadaljevanju se ta podatek v sistemu EBA DMS v Modulu NAV uporablja tudi za pridobivanje podatkov iz sistema Dynamics NAV in posodabljanje statusov ter drugih podatkov v sistemu Dynamics NAV. Na primer, po končani likvidaciji vhodnega računa v sistemu EBA DMS se status dokumenta na strani Dynamics NAV spremeni s klicem spletne storitve iz sistema EBA DMS, glavni parameter klica pa je prav podatek Številka dokumenta NAV.

4.1.4.2 Izvoz novih dokumentov v sistem EBA DMS

V tem primeru gre za izvoz vseh še neprenesenih dokumentov iz internega sistema za upravljanje z dokumenti v sistemu Dynamics NAV v sistem EBA DMS. V osnovi sistem Dynamics NAV samo pripravi dokumente v internem sistemu za upravljanje z dokumenti, opravilo EBA Management pa ob naslednjem zagonu vse neprenesene dokumente ustrezno obdela.

Izvoz dokumentov poteka zaporedno, po en dokument naenkrat. Opravilo EBA Management med izvoženimi in neizvoženimi dokumenti iz sistema Dynamics NAV razlikuje po izpoljenosti polja ID dokumenta EBA v internem sistemu za upravljanje z dokumenti v

sistemu Dynamics NAV. Vedno se izvozijo vsi tisti dokumenti, ki imajo polje ID dokumenta EBA prazno. Izvoz posameznega dokumenta se zaključi, ko opravilo EBA Management v sistemu Dynamics NAV na dokumentu v internem sistemu za upravljanje z dokumenti v polje ID dokumenta EBA zapiše ustrezno številko dokumenta iz zunanjega sistema EBA DMS.

Opravilo EBA Management izvaža dokumente iz sistema Dynamics NAV v sistem EBA DMS na dva različna načina:

1. samo za potrebe arhiviranja dokumentov,
2. za pošiljanje e-računov.

V prvem načinu se z uvozom dokumenta v sistem EBA DMS pot dokumenta zaključi. Dokument je arhiviran in kakršnekoli akcije na dokumentu so onemogočene. Na ta način se arhivirajo npr. tisti izhodni računi, ki se pošiljajo po navadni pošti in so bili natisnjeni že preko vmesnika Dynamics NAV.

V drugem primeru, ko imamo opravka z dokumentom, ki predstavlja e-račun, pa opravilo EBA Management poskrbi, da se dokument izvozi v sistem EBA DMS na način, da bo v nadaljevanju sistem EBA DMS sam, neodvisno od sistema Dynamics NAV, poskrbel za pošiljanje e-računa.

Za avtomatsko pošiljanje dokumenta, kot je npr. e-račun, mora opravilo EBA Management poleg shranjevanja dokumenta v sistem EBA DMS izvesti še naslednje korake:

1. nastaviti protokol za pošiljanje e-računa (če je prejemnik e-računa proračunski uporabnik, se izbere protokol UJP Exchange sicer pa Bank Exchange),
2. nastaviti naslov za pošiljanje (BIC banke in TRR prejemnika),
3. nastaviti podatke XML, ki jih predpisuje shema e-Slog (ker je v sistemu Dynamics NAV datoteka XML po shemi e-Slog že pripravljena, se le-ta samo še pripne na dokument v sistemu EBA DMS),
4. prestaviti dokument, ki predstavlja e-račun, v odpremo.

Dokument, ki je e-račun in je v statusu V odpremi, bo poslan ob naslednjem izvajanju funkcije Pošlji/Prejmi. Bodisi jo bo zagnal uporabnik ali pa bo zagnana s strani strežniške storitve za izvajanje opravil.

4.2 Modul NAV v sistemu EBA DMS

Modul NAV v sistemu EBA DMS je skriptni projekt, razvit v razvojnem okolju EBA Developer. Modul NAV sistem EBA DMS razširja s sistemom za definiranje, izvajanje in nadziranje poteka večnivojskega parafiranja dokumentov iz sistema Dynamics NAV v sistemu EBA DMS.

V prejšnjem poglavju opisano opravilo EBA Management skrbi le za izvoz tistih dokumentov iz sistema Dynamics NAV v sistem EBA DMS, kjer nadaljnje posredovanje uporabnikov ni več potrebno. Ostali dokumenti iz sistema Dynamics NAV pa so v sistem EBA DMS poslani interaktivno s strani uporabnikov. Navadno uporabniki, ki kreirajo take dokumente, v sistemu Dynamics NAV vnašajo podatke, na podlagi katerih se kasneje na njihovo zahtevo pripravijo dokumenti, ki jih tudi parafirajo.

Na zgoraj opisan način se v trenutnem sistemu pripravljajo naslednji dokumenti:

- likvidacijski listi (gre za dokument, ki nastane z namenom potrjevanja vhodnega dokumenta, ki ima za posledico finančno izplačilo, npr. za potrjevanje vhodnih računov, pogodb za izplačilo, sodnih taks ...),
- dokumenti, povezani s prenosi in odpisi osnovnih sredstev in drobnega inventarja,
- dopisi za zavrnitev vhodnih računov.

Ko uporabnik vnese vse zahtevane podatke preko ukaznega gumba Potrdi ali pa Pošlji v potrditev, sproži pripravo dokumenta in avtomatsko vstavitvev le-tega v sistem EBA DMS preko programske knjižnice EBAXCOM. Predpogoj za vstavljanje dokumentov v sistem EBA DMS na ta način je, da je uporabnik že pred vstavljanjem dokumenta prijavljen v Odjemalca EBA, saj se vstavljanje dokumentov na ta način izvede lokalno, na računalniku uporabnika. Modul NAV v sistemu EBA DMS, ki teče tudi v okviru Odjemalca EBA, pa zahtevo prestreže, še preden se dokument sploh shrani v sistem EBA DMS.

Obdelavo zahteve in nadaljnji proces v sistemu EBA DMS lahko razdelimo v dve fazi:

1. definiranje tabele za večnivojsko parafiranje,
2. izvedba parafiranja.

Glede na tip dokumenta obstajata dva načina izvedbe prve faze. V obeh primerih je rezultat izvedbe definirana tabela, ki določa potek oz. izvedbo večnivojskega parafiranja oz. druge faze po prej navedeni delitvi.

1. faza: definiranje tabele za večnivojsko parafiranje

Prvi način definiranja tabele večnivojskega parafiranja je, da se zaporedje potrjevalcev določi glede na organizacijsko strukturo zaposlenih. Prvi v zaporedju je uporabnik, ki je dokument pripravil, sledi mu nadrejeni, temu njegov nadrejeni itd., vse do vrha organizacijske strukture zaposlenih v podjetju. Pri tem načinu imamo lahko na posameznem nivoju samo enega paraferja.

Drugi način definiranja tabele večnivojskega parafiranja pa temelji na nizu znakov, ki ga sistem Dynamics NAV ob vstavljanju dokumenta zapiše v polje Seznam podpisnikov NAV. Za razliko od prvega načina definiranja tabele večnivojskega parafiranja ta način omogoča tudi možnost, da lahko na posamezen nivo parafiranja uvrstimo tudi več paraferjev hkrati. Seznam podpisnikov NAV vsebuje vse podatke, ki so potrebni, za definiranje tabele večnivojskega parafiranja. Niz znakov je sestavljen iz poljubno mnogo podenot n=dddd, ki so med seboj ločene z vejico. Število n predstavlja nivo, na katerem se pojavi delavec s šifro dddd. Primer takega niza je 1=1354,2=1589,3=1486,3=1156,4=1824,5=1568.

Večnivojsko parafiranje

Podpisnik	Nivo	Parafirano	Datum	Zap. st. parafa	Dostop	Opomba
Alejo Žumer	1					
Boris Hudič	2					
Agnes Štuler	3					
Andrej Štrukelj	3					
Harald Zupanc	4					
Edvard Koželj	5					
Večnivojsko parafiranje končano						
Opomba (vzrok uničenja)						
Seznam podpisnikov NAV						
1=0178,2=0098,3=0203,3=0502,4=0111,5=0000						

Slika 22: Tabela večnivojskega parafiranja pred začetkom parafiranja

Če je pri pripravi tabele za izvedbo večnivojskega parafiranja prišlo do kakršnihkoli težav, se dokument v sistemu EBA DMS ne shrani. Uporabnik je o napaki obveščen in z delom nadaljuje v sistemu Dynamics NAV, kjer skuša odpraviti vzroke, zaradi katerih je prišlo do težav.

Če je bil postopek priprave tabele za izvedbo večnivojskega parafiranja uspešen, se dokument v sistemu EBA DMS shrani. Hkrati se v sistemu Dynamics NAV v internem sistemu za upravljanje z dokumenti spremeni tudi status odobritve iz Nerešena odobritev v Nerešen podpis. Sprememba statusa v Nerešen podpis izvajanje poslovnega procesa nad dokumentom sistemu Dynamics NAV zamrzne. Onemogočeno je kakršnokoli spreminjanje podatkov in izvajanje akcij nad dokumentom. Proces nad dokumentom v sistemu Dynamics NAV steče naprej šele, ko iz sistema EBA DMS pride informacija o spremembi statusa na dokumentu bodisi ob končanem parafiranju, ko se status spremeni v Odobreno bodisi ob uničenju dokumenta, ko se status spremeni nazaj v Nerešena odobritev.

Vse spremembe statusov Modul NAV v sistemu EBA DMS sporoča v sistem Dynamics NAV preko klicev spletne storitve EBAIntegration na strani sistema Dynamics NAV. Opis vmesnika spletne storitve EBAIntegration v jeziku WSDL je naveden v prilogi 9.1.

2. faza: izvedba parafiranja

Ko je tabela za izvedbo večnivojskega parafiranja pripravljena, se začne druga faza, postopek izvedbe parafiranja. V vsakem koraku druge faze je enolično določen trenutni nivo parafiranja, ki je najnižji še ne v celoti parafiran nivo parafiranja. V vsakem koraku se dokument pošlje v parafiranje vsem paraferjem, ki so na trenutnem nivoju in še niso parafirali dokumenta. Po vsaki spremembi oz. napredku v postopku parafiranja se trenutno stanje parafiranja preko klica spletne storitve EBAIntegration sistema Dynamics NAV sporoči tudi v sistem Dynamics NAV. Na ta način lahko uporabniki, ki niso del procesa potrjevanja dokumenta v sistemu EBA DMS, spremljajo potek potrjevanja tudi skozi sistem Dynamics NAV. Ko so enkrat v celoti parafirani vsi nivoji, je večnivojsko parafiranje končano. Takrat sistem EBA DMS preko klica spletne storitve EBAIntegration v sistem Dynamics NAV sporoči spremembo statusa iz Nerešen podpis v Odobreno.

Večnivojsko parafiranje

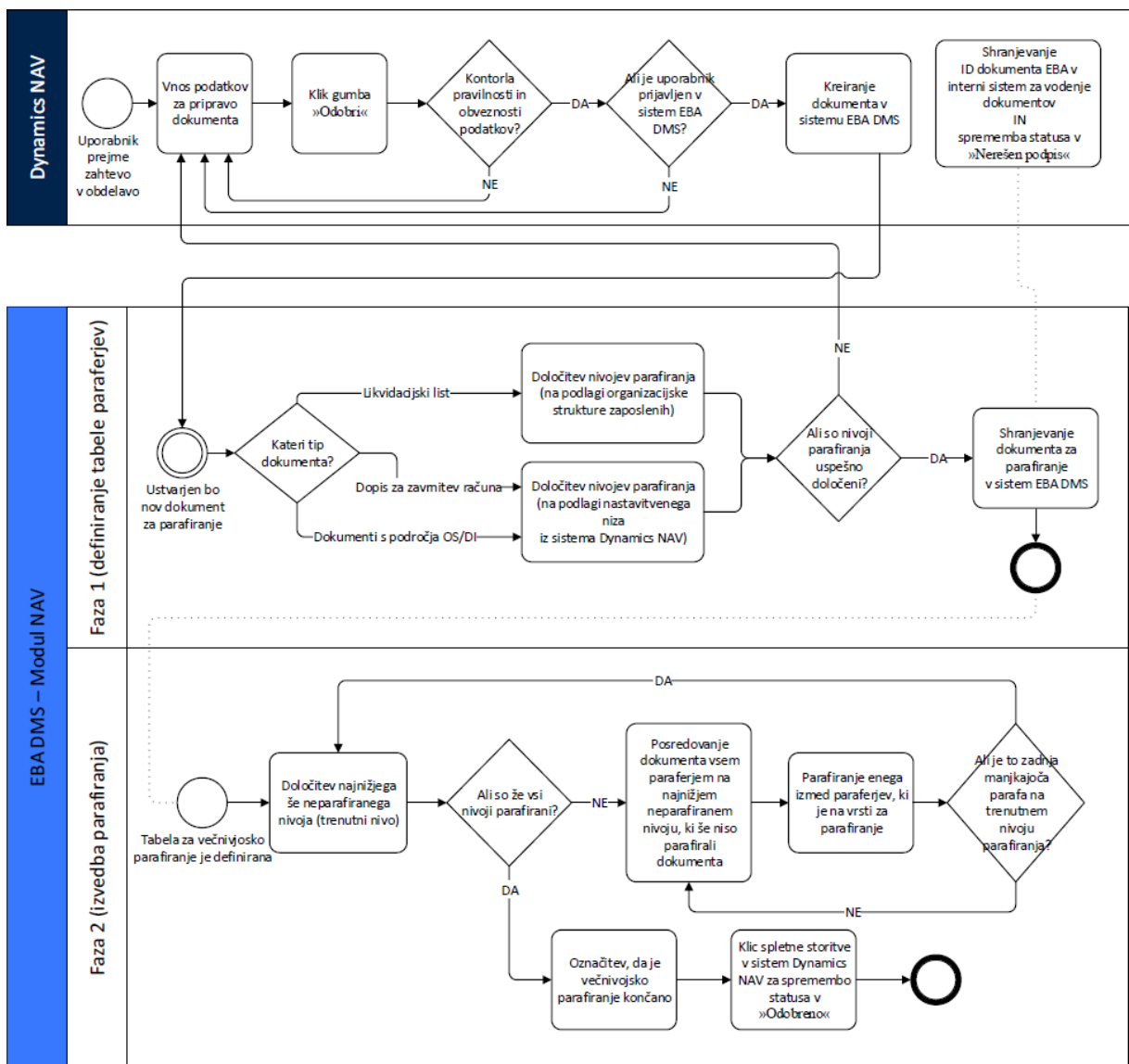
Podpisnik	Nivo	Parafirano	Datum	Zap. st. parafa	Dostop	Opomba
Aljoša Čumer	1	Da	10.12.2015	1	Da	
Boris Kubič	2	Da	15.12.2015	2	Da	
Agata Štuler	3	Da	15.12.2015	3	Da	
Andrej Štrukelj	3	Da	15.12.2015	4	Da	Nastavljeno namestništvo.
Marija Županc	4	Da	16.12.2015	5	Da	Nastavljeno namestništvo.
Edvard Koželj	5	Da	16.12.2015	6	Da	

Večnivojsko parafiranje končano **Da**

Opomba (vzrok uničenja)

Seznam podpisnikov NAV
 1=0178,2=0098,3=0193,3=0502,4=0131,5=0000

Slika 23: Tabela večnivojskega parafiranja po končanem parafiranju

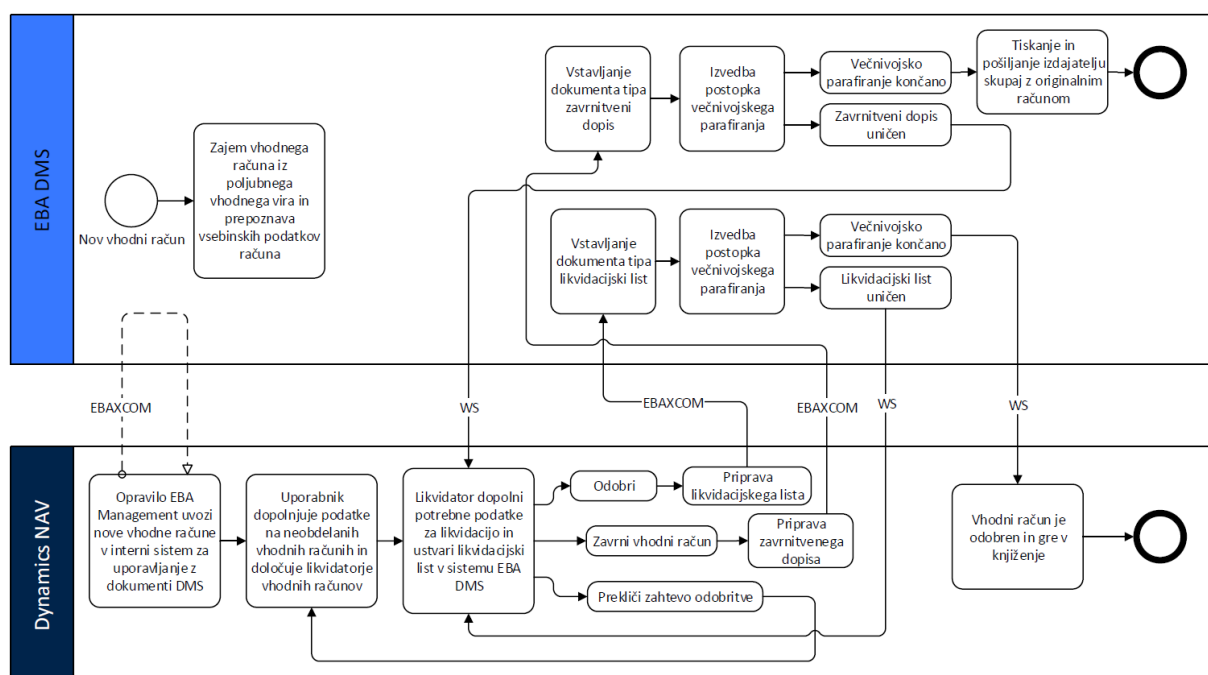


Slika 24: Proces parafiranja dokumenta iz sistema Dynamics NAV v sistemu EBA DMS

4.3 Primer - Prejeti računi in likvidacija

Celovita integracija med sistemoma EBA DMS in Dynamics NAV je najlepše vidna na procesu obdelave vhodnih računov. Vse od prejema vhodnega računa v podjetje, uvoza v sistem Dynamics NAV, prek njegove likvidacije pa vse do postopka knjiženja le-tega, če do knjiženja sploh pride, se poslovni proces izmenjuje izvaja vsaj štirikrat v drugem sistemu. Lahko pa je teh izmenjav tudi več.

V nadaljevanju bo podrobno predstavljen poslovni proces obdelave vhodnih faktur od prejema do knjiženja skozi vidik integracije med sistemoma EBA DMS in Dynamics NAV.



Slika 25: Poslovni proces obdelave vhodnih računov od prejema do knjiženja skozi vidik integracije med sistemoma EBA DMS in Dynamics NAV

Vsi vhodni računi, ki pridejo v podjetje, so bodisi vloženi v sistem EBA DMS na vložišču preko optičnega čitalca bodisi so v sistem EBA DMS vstavljeni avtomatsko, ker so bili prejeti na enega izmed elektronskih načinov prejemanja dokumentov. Avtomatska prepoznavna vsebinskih podatkov sistema EBA DMS nam pomaga, da nekatere vsebinske podatke iz slike dokumenta sistem zazna že avtomatsko ob zajemu dokumenta v sistem EBA DMS.

Opravi EBA Management, ki je eno izmed opravil sistema Dynamics NAV in ki se izvaja avtomatsko po zanj določenem časovnem urniku, ob naslednjem izvajanju vse nove oz. še

neprenesene vhodne račune prenese iz sistema EBA DMS v sistem Dynamics NAV v interni sistem za upravljanje z dokumenti (podrobnosti so bile predstavljene v poglavju 4.1.4). Iz sistema EBA DMS se poleg dokumenta prenesejo tudi vsi vsebinski podatki, ki so bili iz dokumenta razbrani že ob shranjevanju vhodnega računa v sistem EBA DMS.

Uporabnik iz finančne službe, ki je zadolžen za opremljanje vhodnih računov v internem sistemu za upravljanje z dokumenti v sistemu Dynamics NAV, vsak račun dopolni z vsemi podatki, ki še manjkajo in so v tej fazi zahtevani. Eden izmed zahtevanih podatkov je tudi določen likvidator računa. Ko je uporabnik vnesel vse podatke, vključno z likvidatorjem, temu likvidatorju vhodni račun pošlje v odobritev.

Naloga likvidatorjev je, da na vhodnem računu dopolnijo podatke, ki so potrebni za uspešno likvidacijo vhodnega računa. Po vnosu vseh podatkov vhodni račun odobrijo, kar v ozadju sproži najprej ustvarjanje in nato še vstavljanje dokumenta likvidacijski list v sistem EBA DMS. Modul NAV sistema EBA DMS prestreže vstavljanje dokumenta in izvede definiranje tabele za večnivojsko parafiranje, na podlagi katere je kasneje izvedeno večnivojsko parafiranje (podrobnosti so bile predstavljene v poglavju 4.2).

V katerikoli fazi večnivojskega parafiranja lahko kdorkoli, ki ima dostop do likvidacijskega lista, izrazi nestrinjanje z dokumentom in ta dokument tudi uniči. Z uničenjem se dokument iz sistema EBA DMS ne izbriše, postane samo neveljaven, čez sliko dokumenta pa se doda vodni žig Uničeno. Modul NAV v sistemu EBA DMS uporabniku dovoli uničenje samo v primeru, ko je uporabnik v za to namenjeno polje na dokumentu likvidacijskega lista vnesel vzrok uničenja. Ta podatek se sporoči v sistem Dynamics NAV hkrati s klicem spletne storitve EBAIntegration takoj po uničenju, ki spremeni status na dokumentu v internem sistemu za upravljanje z dokumenti v sistemu Dynamics NAV. Na podlagi podatka o vzroku uničenja likvidator v sistemu Dynamics NAV bodisi ustrezno popravi podatke in ponovno sproži ustvarjanje likvidacijskega lista bodisi zavrne celoten vhodni račun izdajatelju računa.

V primeru zavrnitve računa izdajatelju se izvede podoben postopek kot za likvidacijski list.

V primeru, da je bilo večnivojsko parafiranje likvidacijskega lista uspešno zaključeno, informacija o tem preko klica spletne storitve EBAIntegration pride nazaj do sistema Dynamics NAV, ki vhodni račun omogoči za knjiženje.

4.4 Sistem opozarjanja uporabnikov z alarmi

Pred uvedbo likvidacije v elektronski obliki, kot je bila predstavljena v prejšnjem poglavju, so se likvidacijski listi podpisovali še v fizični obliki. Predvsem fizična razpršenost podpisnikov in potreba po prenašanju likvidacijskega lista iz pisarne v pisarno sta vzroka, ki sta v največji meri podaljševala čas, ki je bil potreben za likvidacijo.

Z uvedbo elektronskega parafiranja likvidacijskih listov so bili vsi zgoraj navedeni problemi odpravljeni. Posledično se je zmanjšal tudi čas, potreben za likvidacijo vhodnih računov. Uporabnik, ki se ga čaka, da parafira dokument, lahko namreč sedaj dokument parafira tudi, če ni na svojem delovnem mestu, potrebuje le dostop do svojega računalnika.

Še dodatno pa se je čas, potreben za likvidacijo, zmanjšal po uvedbi alarmov za neparafirane dokumente v sistemu EBA DMS. Definirana sta dva nivoja alarmov:

- prvi je nastavljen tako, da opozarja vsakega uporabnika, ki ima dokument, ki ga mora parafirati, v pisarni že dlje kot 24 ur, a ga še ni parafiral,
- drugi je nastavljen tako, da v primeru, ko ima nek uporabnik dokument v svoji pisarni že več kot 5 dni in ga še ni parafiral, o tem obvesti odgovorno osebo. V primeru likvidacijskih listov je to vodja finančne službe.

Oba nivoja alarmov se sprožita vsak delovni dan zjutraj. S tem zagotovimo, da uporabnike, ki zadržujejo nadaljevanje poteka poslovnega procesa, čaka opozorilo o neparafiranem dokumentu v poštnem nabiralniku elektronske pošte že ob prihodu v službo.

Število alarmov: 2

Zadeva: Opozorilo - 'Likvidacijski list - **SPREJETA - OBLASTI**'

Dokument 'Likvidacijski list - **SPREJETA - OBLASTI**' je v pisarni **Primož Kramar** že '2 dneva, 17 ur, 42 minut' in čaka na parafiranje.

Prosimo, če ga čim prej parafirate.

[Odpri](#)

Zadeva: Opozorilo - 'Likvidacijski list - **ELJEKTRONSKIŠTIPELIK, PODJETJE ZA AKOVIČIČI**'

Dokument 'Likvidacijski list - **ELJEKTRONSKIŠTIPELIK, PODJETJE ZA AKOVIČIČI**' je v pisarni **Primož Kramar** že '2 dneva, 14 ur, 52 minut' in čaka na parafiranje.

Prosimo, če ga čim prej parafirate.

[Odpri](#)

Slika 26: Primer elektronskega sporočila, ki ga dobi uporabnik kot alarm zaradi neizvedenega parafiranja

4.5 Povzetek

V poglavju, ki je za nami, smo najprej podrobneje spoznali sistem za upravljanje z dokumenti, ki je že del produkta Microsoft Dynamics NAV. Sledila je predstavitev prilagoditev in nadgradenj sistema Dynamics NAV, ki nam zagotavljajo integracijo sistema Dynamics NAV z elektronskim dokumentnim sistemom EBA DMS. Opis integracije med obema sistemoma se je zaključil z opisom skriptnega projekta Modul NAV, ki, kot je bilo že rečeno, sistem EBA DMS razširja s sistemom za definiranje, izvajanje in nadziranje poteka večnivojskega parafiranja dokumentov iz sistema Dynamics NAV v sistemu EBA DMS.

Celovito je bila integracija med sistemoma EBA DMS in Dynamics NAV najlepše prikazana na koncu poglavja na primeru poslovnega procesa obdelave vhodnih računov.

V naslednjem poglavju sledi predstavitev načina integracije elektronskega dokumentnega sistema EBA DMS z aplikacijama BIS in ŽCO.

5 Integracija aplikacij BIS in ŽCO s sistemom EBA DMS

Aplikaciji BIS in ŽCO sta specialni aplikaciji in podpirata poslovne procese, ki so del naše osnovne dejavnosti v podjetju. V omenjenih aplikacijah se generirajo računi, obvestila, opomini, pogodbe in drugi dokumenti. Vsi dokumenti so generirani v obliki PDF in so v končni fazi arhivirani v elektronskem dokumentnem sistemu EBA DMS.

Poti, po katerih dokumenti končajo v elektronskem dokumentnem sistemu EBA DMS, so različne. Aplikaciji BIS in ŽCO imata za kreiranje in arhiviranje dokumentov naslednje možnosti:

- neposredno vstavljanje dokumentov iz aplikacije BIS in ŽCO v sistem EBA DMS preko spletnih storitev,
- odlaganje dokumentov v obliki datotek PDF na datotečnem sistemu,
- prelaganje dela na storitev za masovno pripravo izhodnih dokumentov.

5.1 Uporaba spletnih storitev EBA DMS

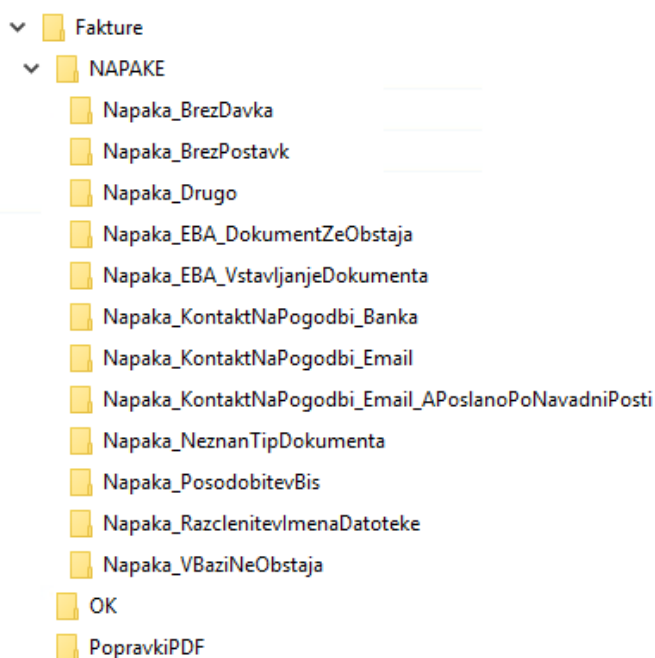
Način arhiviranja dokumentov prek spletnih storitev sistema EBA DMS se uporablja v primerih, ko uporabniki delajo na posameznih primerih oz. posameznih dokumentih. Primer take uporabe je kreiranje pogodb. Uporabnik iz predloge v programu Word pripravi končno obliko pogodbe in jo shrani v aplikacijo. S klikom v aplikaciji pa potem to datoteko preko spletne storitve GeneralDocuments arhivira v sistem EBA DMS. Kot povratno informacijo spletna storitev sistema EBA DMS vrne ID dokumenta, ki je bil pravkar vstavljen v sistem EBA DMS. Aplikaciji BIS in ŽCO ta ID dokumenta shranita in ga kasneje uporabita za omogočanje prikaza dokumenta iz aplikacije v sistemu EBA DMS. Opis vmesnika spletne storitve GeneralDocuments v jeziku WSDL je naveden v prilogi 9.2.

Prednost tega načina pred ostalimi načini, ki bodo predstavljeni v nadaljevanju, je v tem, da je dokument arhiviran takoj. V ostalih načinih bomo videli, da so dokumenti arhivirani šele ob naslednjem zagonu opravil, ki skrbijo za arhiviranje dokumentov.

5.2 Program za arhiviranje dokumentov iz aplikacij BIS in ŽCO

Pri drugem načinu za arhiviranje dokumentov v sistem EBA DMS uporabniki aplikacij BIS in ŽCO ne igrajo ključne vloge pri ustvarjanju dokumentov. Kreiranje dokumentov navadno izvajajo opravila, ki so del samih programov BIS in ŽCO. Za razliko od načina, opisanega v prejšnjem poglavju, gre v tem primeru tudi za generiranje večjega števila dokumentov. Opravila kreirajo dokumente tipa PDF in jih odlagajo na dogovorjenem omrežnem mestu, kjer čakajo, da jih program za arhiviranje ustrezno obdela.

Na dogovorjenem omrežnem mestu je definiranih več skupkov podmap. Posamezen skupek podmap se navadno uporablja zgolj za en tip dokumentov. Spodaj je kot primer prikazan skupek podmap, ki se uporablja za arhiviranje računov oz. faktur.



Slika 27: Struktura datotečnega sistema za arhiviranje dokumentov BIS in ŽCO

Program za arhiviranje dokumentov BIS in ŽCO teče na posebnem strežniku, izvaja pa se kot opravilo v sistemu Windows Server. Razporejevalnik opravi program za arhiviranje dokumentov BIS in ŽCO avtomatsko zažene enkrat na vsako uro.

Postopek arhiviranja

Ob vsakem izvajanju program za arhiviranje dokumentov BIS in ŽCO pregleduje strukturo map na omrežnem disku in obdeluje neobdelane oz. nearhivirane datoteke PDF. Neobdelane datoteke se v drevesu map nahajajo v korenski mapi posameznega skupka podmap. V prej navedenem primeru je to mapa Fakture. Program se v zanki premika po neobdelanih dokumentih in obdeluje po en dokument PDF naenkrat. Vsaka datoteka PDF je poimenovana po točno določenem, vnaprej dogovorjenem vzorcu. Navadno tako, da vsebuje podatke za določitev tipa dokumenta in pa ključ, preko katerega lahko program za arhiviranje pridobi za arhiviranje pomembne podatke iz podatkovne baze programa, iz katerega je bila datoteka ustvarjena.

Program za arhiviranje dokumentov BIS in ŽCO nato preko programske knjižnice EBAXCOM glede na pridobljene podatke ustrezno vstavi dokument v sistem EBA DMS.* Sledi zapis podatkov o pravkar ustvarjenem dokumentu v sistemu EBA DMS v podatkovno bazo izvorne aplikacije. Preko teh podatkov je v izvorni aplikaciji kasneje omogočen dostop do dokumenta, ki je sicer shranjen v sistemu EBA DMS. Če v celotnem zgoraj opisanem postopku arhiviranja ni prišlo do nobene napake, sledi še zadnji korak, ki je premik datoteke PDF v podmapo OK, kjer se nahajajo samo uspešno arhivirane datoteke PDF. S tem je obravnava ene datoteke PDF končana in program za arhiviranje dokumentov BIS in ŽCO nadaljuje z arhiviranjem naslednje datoteke PDF v korenski mapi.

Obravnava napak pri arhiviranju

V primeru, da pri obdelovanju datoteke PDF pride do kakršnekoli napake, program za arhiviranje dokumentov datoteko PDF glede na vzrok napake, premakne v ustrezno podmapo podmape NAPAKE. Imena podmap mape NAPAKE so poimenovana na način, da že iz imena podmape lahko sklepamo na vzrok, zaradi katerega se datoteka PDF ni uspešno obdelala. Program za arhiviranje dokumentov določeno osebo, ki je nadzornik programa za arhiviranje dokumentov, o podrobnostih nastale napake obvesti preko elektronske pošte. S tem je obravnava nastale napake končana in program za arhiviranje dokumentov BIS in ŽCO nadaljuje z arhiviranjem naslednje datoteke PDF v korenski mapi.

Naloga nadzornika programa za arhiviranje dokumentov v nadaljevanju je, da skrbi za naslednje:

- da spremlja elektronska sporočila o napakah,

- da poskrbi za odpravo posameznih vzrokov, ki so povzročili napake pri obdelavi,
- da po odpravljenem vzroku nastale napake premakne ustrezno datoteko PDF nazaj v korensko mapo. Tam ta datoteka PDF med ostalimi neobdelanimi čaka na naslednjo obdelavo še neobdelanih oz. nearhiviranih datotek PDF.

V korenski mapi se nahaja še ena podmapa, in sicer PopravkiPDF. V to podmapo programa BIS in ŽCO odlagata datoteke PDF, ki so že bile arhivirane in za katere želimo, da jih zamenja nova verzija dokumenta. Program za arhiviranje dokumentov posamezno datoteko PDF v podmapi PopravkiPDF obdela tako, da v sistemu EBA DMS, ravno tako preko programske knjižnice EBAXCOM, najprej izbriše že arhiviran dokument. Za tem izbriše sledi o arhiviranem dokumentu še v izvorni aplikaciji, postopek pa konča s premikom datoteke PDF v korensko mapo, kjer datoteka PDF čaka na naslednjo obdelavo še neobdelanih oz. nearhiviranih datotek PDF.

5.2.1 Elektronsko pošiljanje izdanih računov iz aplikacije BIS

Postopek obdelave datoteke PDF izdanega računa za pripravo na elektronsko pošiljanje se povsem ujema s postopkom, ki je bil podan v prejšnjem poglavju. Dodanih je le še nekaj vmesnih korakov pred vstavljanjem dokumenta v sistem EBA DMS, ki kasneje v tem sistemu zagotovijo avtomatsko pošiljanje dokumenta kot e-račun.

Program za arhiviranje dokumentov za vsak dokument PDF, ki je tipa račun, pridobi tudi podatek, ki definira, na kakšen način bo račun dostavljen prejemniku. Z načinom dostave računa je enolično določen protokol v sistemu EBA DMS, po katerem bo račun v elektronski obliki poslan prejemniku, protokol pa določa obliko naslova, ki jo sistem EBA DMS potrebuje za uspešno pošiljanje računa.

V aplikaciji ŽCO in BIS poznamo naslednje načine dostave računov:

- po navadni pošti,
Račun je natisnjen, vložen v ovojnico in oddan na navadno pošto.
- račun PDF po elektronski pošti,
Račun je prejemniku poslan po elektronski pošti z dvema priponkama (račun v obliki PDF in račun v obliki XML v e-Slogu).

- po navadni pošti in dokument PDF po elektronski pošti.
Združena sta prejšnja načina.
- e-račun.
Vrsta protokola, po katerem se e-račun pošlje, se ugotovi iz naslova protokola glede na BIC banke (bodisi je to Bank Exchange ali pa UJP Exchange).
- e-račun z obvestilom o računu po elektronski pošti.
Način enak tistemu v prejšnji alineji, dodatno pa program za arhiviranje dokumentov prejemniku e-računa pošlje še elektronsko sporočilo z vsebino: »V najkrajšem možnem času lahko v svoji spletni banki pričakujete nov e-račun ...«
- EBA Exchange.
Izmenjava po internem protokolu sistema EBA DMS (podrobnosti so bile predstavljene v poglavju 3.3.1).

Postopek obdelave programa za arhiviranje dokumentov iz BIS in ŽCO

V primeru računa, ki ima določen način dostave po navadni pošti, gre v bistvu samo za arhiviranje računa in je postopek popolnoma enak postopku, ki je bil opisan v prejšnjem poglavju.

V primeru računa, ko je način dostave računa različen od pošiljanja preko navadne pošte, pa gre za vstavljanje računa, ki bo kasneje poslan prejemniku kot e-račun na enega izmed elektronskih načinov izmenjave dokumentov. Za uspešno pošiljanje takega računa kot e-račun je dokument še pred vstavljanjem v sistem EBA DMS potrebno dopolniti z vsemi podatki, ki so navedeni na dokumentu PDF in ki so zahtevani s strani sheme e-Slog. Gre za naslednje podatke:

- množica različnih vrst zneskov,
- postavke računa,
- tabela davkov,
- klavzule na računu,
- subjekti oz. partnerji računa (kupec, prejemnik, plačnik in izdajatelj),
- itd.

Ko so v dokumentu dopolnjeni vsi zahtevani podatki in se nastavita še protokol in naslov protokola v sistemu EBA DMS (glej poglavje 3.3), je dokument pripravljen na vstavljanje v sistem EBA DMS. Če se je vstavljanje izvedlo brez napak, program za arhiviranje dokument premakne še v odpremo in s tem so izpolnjeni vsi predpogoji, da naslednje izvajanje akcije Pošlji/Prejmi izvede pošiljanje dokumenta kot e-račun.

Nadaljevanje postopka je enako kot je opisano v predhodnem poglavju 5.2 od znaka * naprej.

5.2.2 Elektronsko pošiljanje izdanih zbirnih računov iz aplikacije BIS

Redni posamezni računi se strankam navadno pošiljajo ob koncu meseca. V primeru, ko ima stranka več rednih mesečnih računov, se lahko prijavi tudi na prejemanje zbirnega računa, ki se izdaja enkrat mesečno. Zbirni račun je v bistvu umetno ustvarjen račun, na katerem so navedeni vsi elementi navadnega računa. Kot postavke so na zbirnem računu navedeni posamezni računi, ki so vključeni na zbirnem računu.

Pred uvedbo elektronskega načina pošiljanja računov so se vsi računi brez izjeme natisnili in pošiljali posamezno po navadni pošti. Enako je veljalo tudi za zbirne račune, le da so se ti pošiljali v kuverti skupaj s še vsemi posameznimi računi, ki so bili vključeni na zbirnem računu.

Zaradi zahteve po pošiljanju posameznih računov skupaj z zbirnimi računi, smo pri uvedbi pošiljanja zbirnega računa kot e-računa naleteli na več omejitev. V nadaljevanju bodo predstavljene omejitve zunanjih sistemov, ki smo jih morali upoštevati pri vzpostavljanju rešitve za pošiljanje zbirnih računov kot e-računov (povzeto po [13] in [12]).

Sistem e-račun Združenja bank Slovenije v svojem modelu ne predvideva pošiljanja zbirnikov kot e-računov. Predvideva pa različno število prilog in tudi različne tipe prilog. Omejitve se razlikujejo tudi glede na to, ali je račun namenjen pravni ali fizični osebi. Za pošiljanje pravnim osebam so dovoljeni tipi priponk XML, TIFF, JPG, PDF, TXT, CSV. Število prilog je praviloma omejeno, saj je omejena skupna velikost prilog. Vse priloge, skupaj stisnjene v *.zip, so lahko velike največ 2 MB, vključno z e-računom v formatu XML eSlog. Za pošiljanje fizičnim osebam so omejitve strožje. Prva priponka mora biti v formatu XML eSlog, dovoljena je le še ena priponka, ki je lahko le *.pdf. Ostali tipi priponk ali dodatne priponke niso dovoljene. Omejitev velikosti je enaka – največ 2 MB.

V okviru sistema UJP (med proračunskimi uporabniki) je mogoča izmenjava večjih datotek oziroma e-računov, in sicer v velikosti do 20 MB.

V našem sistemu lahko zbirne račune elektronsko pošiljamo na naslednja dva načina:

- *zbirni račun PDF po elektronski pošti.*

Zbirni račun je prejemniku poslan po elektronski pošti z dvema priponkama (zbirni račun v obliki PDF in pripadajoča datoteka XML zbirnega računa v e-Slogu),

- posamezni računi so poslani po elektronski pošti.

- *e-račun zbirnega računa.*

Vrsta protokola, po katerem se e-račun pošlje, se ugotovi iz naslova protokola glede na BIC banke (bodisi je to Bank Exchange ali pa UJP Exchange),

- *protokol Bank Exchange:*

posamezni računi so poslani po elektronski pošti,

- *protokol UJP Exchange:*

način pošiljanja posameznih računov je odvisen od skupne velikosti posameznih računov. Če je velikost manjša od 20 MB, se posamezni računi pošljejo kot priponke k e-računu, v nasprotnem primeru se posamezni računi pošljejo po elektronski pošti.

Postopek obdelave programa za arhiviranje dokumentov iz BIS in ŽCO

V primeru zbirnega računa imamo na definiranem omrežnem mestu odloženo tako datoteko PDF zbirnega računa kot tudi vse datoteke PDF posameznih računov. Vrstni red obdelave je vedno tak, da se najprej obdelajo vse datoteke PDF posameznih računov, zatem pa sledi šele obdelava datoteke PDF zbirnega računa.

Vsaka datoteka PDF posameznega računa je obdelana samostojno po postopku, ki je bil opisan v poglavju 5.2.1, z načinom dostave računa po navadni pošti. To pomeni, da je račun v sistem EBA DMS samo arhiviran in ne bo elektronsko poslan s strani sistema EBA DMS. Seveda bi lahko v primerih, ko so posamezni računi stranki poslani preko elektronske pošte, te poslali tudi preko sistema EBA DMS, vendar bi bil v tem primeru vsak posamezni račun poslan stranki v ločenem elektronskem sporočilu. S tem bi v primeru, ko je na zbirnem računu vključeno večje število posameznih računov, stranka prejela tudi enako število elektronskih sporočil s

posameznimi računi, kar pa ni prijazno do strank. Po končani obdelavi posameznega računa se datoteka PDF tega računa ne premakne v mapo OK, kot se to zgodi za vse ostale datoteke, ampak v mapo OK/[številka zbirnega računa].

V trenutku obdelovanja datoteke PDF zbirnega računa se v podmapi OK/[številka zbirnega računa] že nahajajo vse datoteke PDF posameznih računov. Program za arhiviranje po uspešnem vstavljanju zbirnega računa v sistem EBA DMS preveri skupno velikost datotek PDF posameznih računov, ki se nahajajo v mapi OK/[številka zbirnega računa].

V primeru, da je protokol pošiljanja trenutnega zbirnega računa UJP Exchange in da je skupna velikost datotek PDF manjša od 20 MB, se vsaka datoteka PDF iz mape OK/[številka zbirnega računa] doda kot priloga na dokument v sistemu EBA DMS. Protokol UJP Exchange v sistemu EBA DMS deluje tako, da se ob pošiljanju dokumenta po tem protokolu k e-računu kot priponke avtomatsko dodajo tudi vse priponke, ki so sicer dodane na dokument v sistemu EBA DMS.

V vseh ostalih primerih pa se vse datoteke PDF iz mape OK/[številka zbirnega računa] pošljejo stranki ločeno po elektronski pošti. Želja je, da bi lahko poslali vse datoteke z enim elektronskim sporočilom. Ker pa stranke uporabljajo za elektronsko pošto različne ponudnike storitev in ker imajo ti ponudniki določene različne omejitve glede maksimalne velikosti elektronskega sporočila, smo tudi na tem mestu morali uvesti dodatne ukrepe. Pri generiranju elektronskega sporočila, kateremu se prilagajo datoteke PDF iz mape OK/[številka zbirnega računa] kot priponke, se spremlja skupna velikost trenutnega sporočila. Ko velikost sporočila doseže mejno vrednost 10 MB, se trenutno elektronsko sporočilo pošlje, preostale datoteke PDF pa se prilagajo k naslednjemu elektronskemu sporočilu. Postopek se ponavlja toliko časa, dokler niso poslane vse datoteke PDF posameznih računov konkretnega zbirnega računa.

Program za arhiviranje dokumentov BIS in ŽCO obdelavo datoteke PDF zbirnega računa zaključi s premikom dokumenta v sistemu EBA DMS v odpremo. S tem so izpolnjeni vsi predpogoji, da naslednje izvajanje akcije Pošlji/Prejmi izvede pošiljanje zbirnega računa kot e-račun oz. po elektronski pošti, odvisno od nastavljenega protokola za pošiljanje.

Nadaljevanje postopka je enako kot je opisano v poglavju 5.2 od znaka * naprej.

5.3 Spletna storitev za masovno pripravo dokumentov

Tretja pot, po kateri lahko dokumenti, tokrat zgolj samo iz aplikacije ŽCO, končajo v dokumentnem sistemu EBA DMS, pa je preko spletne storitve za masovno pripravo dokumentov. Opis vmesnika spletne storitve za masovno pripravo dokumentov v jeziku WSDL je naveden v prilogi 9.3.

5.3.1 Predstavitev problema

V aplikaciji ŽCO se pripravljajo razna obvestila in opomini za stranke. Količina teh dokumentov je velika, včasih tudi po več tisoč dokumentov naenkrat. Dokumenti se generirajo iz predlog programa Word, njihova vsebina pa se v dokument vstavlja dinamično, glede na podatke iz podatkovne baze aplikacije ŽCO.

Pred uvedbo spletne storitve za masovno pripravo dokumentov je uporabnik v aplikaciji ŽCO v uporabniškem vmesniku določil nabor zahtev oz. primerov, za katere je potem sprožil pripravo dokumentov. Celoten postopek priprave dokumentov se je izvajal lokalno na računalniku uporabnika, ki je pripravo zagnal. Priprava dokumentov se je zaradi velikega števila dokumentov izvajala zelo dolgo časa, uporabnik pa je imel zaradi narave delovanja aplikacije ŽCO onemogočeno vsakršno delo v aplikaciji. Počakati je moral, da se je priprava dokumentov izvedla do konca in šele takrat je lahko nadaljeval z delom v aplikaciji. Aplikacija ŽCO je pripravljene dokumente odlagala na definiranem omrežnem mestu. Tam so čakali, da jih je program za arhiviranje dokumentov, ki je bil podrobneje predstavljen v prejšnjem poglavju, ustrezno obdelal in arhiviral v dokumentni sistem EBA DMS.

Z uporabniškega stališča se je z uvedbo spletne storitve za masovno pripravo izhodnih dokumentov spremenilo samo mesto pripravljanja dokumentov iz uporabnikovega računalnika na strežnik. Uporabnik lahko zaradi te spremembe takoj po oddaji zahteve za generiranje dokumentov nemoteno nadaljuje s svojim delom v aplikaciji ŽCO. Če ga zanima status obdelave, le-tega lahko preveri v vsakem trenutku, preko ene izmed funkcij, ki jih ponuja spletna storitev.

Na strani strežnika se je obdelava priprave dokumentov v primerjavi s staro obdelavo, ki je potekala na računalniku uporabnika, spremenila najbolj v tem, da pripravljanje dokumentov

poteka večnitno. S testiranjem je bilo ugotovljeno, da na trenutnem strežniškem okolju najboljše čase za pripravo dokumentov dosežemo z uporabo osmih niti. Čas, ki je potreben za pripravo nekega paketa dokumentov smo z uvedbo spletne storitve za masovno pripravo dokumentov zmanjšali kar za faktor 20.

5.3.2 Predstavitev rešitve

Spletna storitev za masovno pripravo dokumentov temelji na definiranih nastavitvah tipov dokumentov in na paketih, nad katerimi se izvajajo operacije. Vsak paket je sestavljen iz postavk, pri čemer ena postavka predstavlja en dokument.

Posamezen tip dokumenta, za katerega želimo pripravljati dokumente preko spletne storitve za masovno pripravo dokumentov, definiramo s naslednjimi nastavitvami:

- ODN_ID
ID tipa dokumentov.
- ODN_NAZIV
Naziv tipa dokumentov.
- ODN_NAZIV_KRATEK
Kratek naziv tipa dokumentov, ki se uporabi pri avtomatskem generiranju imen generiranih datotek.
- ODN_POT_DO_PREDLOGE
Omrežno mesto kjer je nameščena predloga za generiranje dokumentov tega tipa.
- ODN_POT_DO_PAKETOV_ZA_TISK
Omrežno mesto, kamor se kopirajo pripravljene datoteke za tiskanje. Če je na konkretnem paketu navedeno drugo omrežno mesto, se datoteke kopirajo na mesto, ki je definirano na konkretnem paketu.
- ODN_POT_EBA_ARHIVIRANJE
Omrežno mesto, kamor se kopirajo datoteke PDF in kjer potem čakajo, da jih obdelata program za arhiviranje dokumentov iz aplikacij BIS in ŽCO.
- ODN_VELIKOST_PAKETA_ZA_TISK
Zaradi velikega števila dokumentov je za vsak tip dokumenta nastavljeno maksimalno število dokumentov, ki so vsebovani v vsakem od paketov za tisk. Če je v osnovnem paketu 1700 dokumentov in je velikost paketa za tiskanje nastavljena na 500, bo spletna

storitev ustvarila 4 pakete za tiskanje, tri s po 500 dokumenti in četrtega z 200 dokumenti.

- ODN_VELJAVNO

Status, ki pove, ali je uporaba tipa dokumentov omogočena.

- ODN_PARAM1_OPIS,

Vrednost je naziv parametra, na katerega se nanaša vrednost parametra ODG_PARAM1, ki je naveden na glavi posameznega paketa.

- ODN_PARAM2_OPIS

Vrednost je naziv parametra, na katerega se nanaša vrednost parametra ODG_PARAM2, ki je naveden na glavi posameznega paketa.

- ODN_PARAM3_OPIS

Vrednost je naziv parametra, na katerega se nanaša vrednost parametra ODG_PARAM3, ki je naveden na glavi posameznega paketa.

- ODN_PARAM4_OPIS

Vrednost je naziv parametra, na katerega se nanaša vrednost parametra ODG_PARAM4, ki je naveden na glavi posameznega paketa.

- ODN_EMAIL_NASLOVI_ZA_OBVESCANJE

Seznam elektronskih naslovov, ločenih s podpičjem, ki se uporabi za obveščanje o končanih obdelavah in pripravljenih paketih za tiskanje. Navadno so navedeni elektronski naslovi tistih uporabnikov, ki nadaljujejo delo s pripravljenimi dokumenti.

- ODN_DATUMVPISA

Datum in čas vpisa nastavitve za nov tip dokumentov.

- ODN_SYSUSER

- ODN_SYSDATE

Vsak paket opisuje glava paketa, v kateri so definirane vse podrobnosti, ki določajo, na kakšen način se bodo postavke paketa obdelale in pa več statusov, ki opisujejo trenutno stanje paketa.

Glava paketa je opisana z naslednjimi parametri:

- ODG_UID

ID paketa.

- ODG_TIP

Tip paketa, ki določa tip dokumenta.

- ODG_ZUNANJI_ID
ID paketa v zunanji aplikaciji.
- ODG_PARAM1
Vrednost parametra ODN_PARAM1_OPIS, ki je definiran v nastavitvah tipa dokumentov.
- ODG_PARAM2
Vrednost parametra ODN_PARAM2_OPIS, ki je definiran v nastavitvah tipa dokumentov.
- ODG_PARAM3
Vrednost parametra ODN_PARAM3_OPIS, ki je definiran v nastavitvah tipa dokumentov.
- ODG_PARAM4
Vrednost parametra ODN_PARAM4_OPIS, ki je definiran v nastavitvah tipa dokumentov.
- ODG_ZAKLJUCEN
Vrednosti 0 ali 1. Status, ki pove, ali je paket zaključen. Paket lahko obdelamo šele, ko je le-ta zaključen. Ko je odprt, lahko dodajamo nove postavke.
- ODG_OBDELAN
Vrednost 0 ali 1. Status, ki pove, ali je paket v celoti obdelan. Za paket štejemo, da je v celoti obdelan, ko so pripravljeni vsi dokumenti v obliki docx in so izvedene še vse dodatno zahtevane akcije na glavi paketa. Na paketu je možno dodatno zahtevati še pripravo dokumentov v obliki PDF (parameter ODG_IZDELAJ_PDF), kopiranje dokumentov v obliki PDF na omrežno mesto, kjer potem čakajo, da jih obdela program za arhiviranje dokumentov iz aplikacij BIS in ŽCO (parameter ODG_IZDELAJ_EBA) in pripravo paketov za tiskanje (parameter ODG_IZDELAJ_PAKETE_ZA_TISK). Uspešna izvedba prve dodatne akcije je predpogoj za izvedbo druge in tretje. Druga in tretja dodatna akcija pa sta med seboj neodvisni.
- ODG_PDF_OBDELANO
Vrednost 0 ali 1. Status, ki pove, ali je priprava datotek PDF končana.
- ODG_NAPAKA_V_OBDELAVI
Vrednost 0 ali 1. Status, ki pove, ali je v obdelavi prišlo do kakšne napake.
- ODG_NAPAKA_OPIS
Podrobnosti o nastali napaki.

- ODG_DATUMVPISA
Datum in čas, ko je bil paket shranjen v sistem.
- ODG_DATUMOBDELAVE
Datum in čas, ko je bil paket obdelan.
- ODG_IZDELAJ_PDF
Parameter za vklop priprave dokumentov v obliki PDF.
- ODG_IZDELAJ_EBA
Parameter za vklop kopiranja dokumentov v obliki PDF na omrežno mesto, kjer datoteke obdelata program za arhiviranje dokumentov iz aplikacij BIS in ŽCO.
- ODG_IZDELAJ_PAKETE_ZA_TISK
Parameter za vklop priprave paketov za tiskanje.
- ODG_POT_DO_PAKETOV_ZA_TISK
Omrežno mesto, kamor se kopirajo pripravljene datoteke za tiskanje.
- ODG_SYSUSER
- ODG_SYSDATE

Postavka paketa je definirana kot:

- ODP_UID
ID postavke.
- ODP_ODG_UID
ID paketa.
- ODP_POSTAVKA_ID
ID postavke v zunanji aplikaciji.
- ODP_OBDELANO
Vrednost 0 ali 1. Status, ki pove, ali je postavka v celoti obdelana. Prvi korak obdelave postavke je vedno priprava dokumenta v obliki docx. Drugi korak obdelave postavke pa je priprava dokumenta v obliki PDF. Izvedba drugega koraka je pogojena z vrednostjo parametra ODG_IZDELAJ_PDF na glavi paketa. Izvede se samo v primeru, ko je vrednost parametra ODG_IZDELAJ_PDF enaka 1.
Za postavko štejemo, da je v celoti obdelana, ko sta uspešno izvedena oba koraka oz. v primeru, ko je ODG_IZDELAJ_PDF enak 0, ko je izveden prvi korak.
- ODP_NAPAKA
Vrednost 0 ali 1. Status, ki pove ali je prišlo pri obdelavi postavke do napake.

Sama rešitev spletne storitve za masovno pripravo dokumentov je bila razvita s pomislekom na čim bolj splošno rešitev. Vseeno pa je za uvedbo novega tipa dokumenta, za katerega želimo pripravljati dokumente preko spletne storitve za masovno pripravo dokumentov, potrebnega nekaj dodatnega dela.

Za zagotovitev podpore novemu tipu dokumentov moramo spletno storitev za masovno pripravo dokumentov razširiti z:

- nastavitvami za nov tip dokumenta,
- novo funkcijo za pridobivanje podatkov iz izvorne aplikacije,
- novo funkcijo za vstavljanje podatkov v dokument predloge,
- novo funkcijo za posodabljanje podatkov v izvorni aplikaciji.

5.4 Povzetek

V tem poglavju smo si podrobneje pogledali več različnih poti oz. načinov, po katerih lahko dokumenti iz zunanjih aplikacij končajo v elektronskem dokumentnem sistemu EBA DMS. Od spletnih storitev, programa za arhiviranje dokumentov iz aplikacij BIS in ŽCO, do spletne storitve za masovno pripravo dokumentov. Izbira ustreznega načina pa je odvisna od konkretnih potreb oz. narave poslovnega procesa, ki ga podpiramo.

V naslednjem poglavju sledi zadnja predstavitev konkretnega načina integracije elektronskega dokumentnega sistema EBA DMS, in sicer s spletno aplikacijo e-Križanka.

6 Integracija spletne aplikacije e-Križanka s sistemom EBA DMS

E-Križanka je interno razvita spletna aplikacija in je v podjetju v prvi vrsti namenjena evidentiranju opravljenega dela zaposlenih. Sem štejemo predvsem:

- vpisovanje rednih ur,
- dopusta,
- dežurstva,
- nadur,
- drugih odsotnosti.

Poleg glavnega namena pa glede na uporabniški profil uporabnikom spletna aplikacija e-Križanka nudi še naslednje funkcije:

- kreiranje in upravljanje s projekti (Dynamics NAV),
- odpiranje delovnih nalogov (Dynamics NAV),
- delegiranje delovnih nalog,
- upravljanje z delovnimi nalogami,
- planiranje dela,
- planiranje dežurstva,
- določanju delovnega programa,
- rezervacije vozil,
- vpisovanje prevoženih kilometrov,
- vpisovanje službenih poti,
- obračun stroškov na službeni poti (Dynamics NAV),
- vpisovanje izobraževanj (Dynamics NAV),
- orodja za pregled nad dokumenti in opravljenim delom,
- orodja za vodje in potrjevanje,
- orodja za tajnice (urejanje relacij za regres za prevoz za uporabnike, regres za malico, regres za prevoz, zaključevanje meseca (Dynamics NAV)),
- orodja za administratorje (urejanje uporabnikov, profili)
- poročila.

Kjer je pri zgoraj navedenih postavkah v oklepaju navedeno Dynamics NAV, je izvedena tesna integracija spletne aplikacije e-Križanka s sistemom Dynamics NAV. V večini primerov je v spletni aplikaciji pripravljen le uporabniški vmesnik, podatki pa se izmenjujejo bodisi preko spletnih storitev bodisi preko poizvedb neposredno na podatkovno bazo sistema Dynamics NAV. Spletne storitve se uporabljajo predvsem za vpisovanje podatkov v sistem Dynamics NAV. Na ta način je zagotovljena konsistenca podatkov, saj se izvedejo tudi vsa preverjanja pravilnosti podatkov, ki jih sicer nudi vmesnik v sistemu Dynamics NAV. Drugi način izmenjevanja podatkov, torej način z dostopanjem neposredno do podatkovne baze sistema Dynamics NAV, pa se uporablja zgolj za branje podatkov iz sistema Dynamics NAV.

Med izvajanjem zgoraj naštetih funkcij, ki so del poslovnih procesov, nastajajo elektronski dokumenti različnih tipov.

Tipi dokumentov spletne aplikacije e-Križanka so:

- odreditev dopusta,
- odreditev nadurnega dela,
- nalog za nadurno delo,
- mesečno poročilo o opravljenem delu,
- dežurna lista,
- zahteva po spremembi registracijskega časa,
- nalog za službeno potovanje,
- obračun potnih stroškov na službeni poti po domovini,
- obračun potnih stroškov na službeni poti po tujini,
- 5 tipov dokumentov za delo pod napetostjo,
- prijava na usposabljanje.

Vsi zgoraj navedeni dokumenti se takoj po nastanku interaktivno vstavijo v elektronski dokumentni sistem EBA DMS, kjer se nad njimi izvede postopek parafiranja oz. potrjevanja.

Postopek parafiranja vedno začne oseba, ki je dokument ustvarila, sledijo pa lahko še parafe nadrejenega, tajnice, vodja področja, direktorja ali predsednika. Postopek parafiranja, tj. vrstni red in število paraf, za nek dokument v večini primerov enolično določa tip tega dokumenta. V nekaterih primerih pa se določi tudi v povezavi z nekaterimi podatki iz dokumenta. Npr.

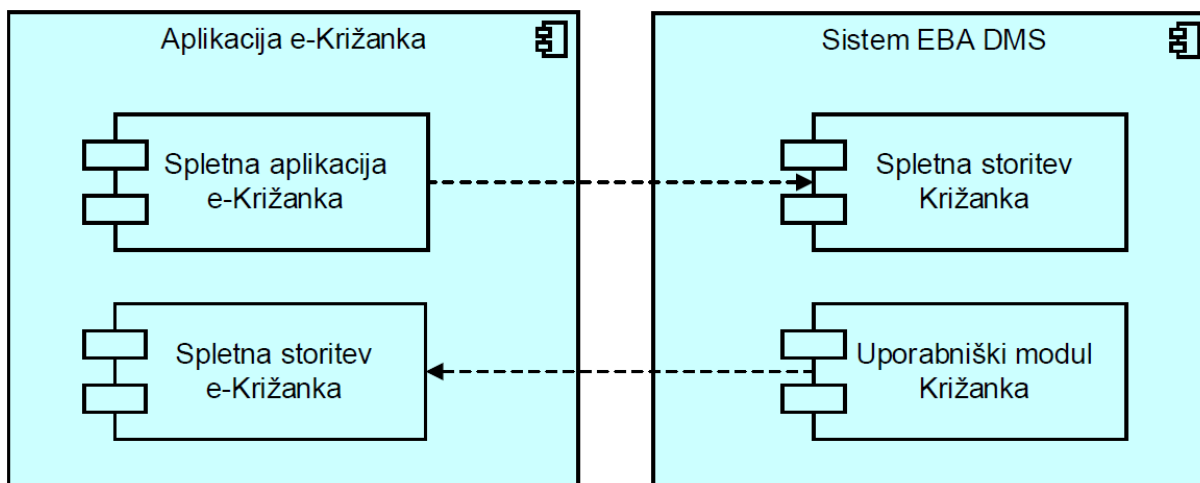
predsednik podpiše nalog za službeno potovanje samo v primeru, ko gre za službeno pot po tujini, sicer pa njegova parafa ni potrebna.

6.1 Opis integracije

Za integracijo spletne aplikacije e-Križanka in elektronskega dokumentnega sistema EBA DMS skrbijo štiri komponente, dve na strani spletne aplikacije e-Križanka in dve na strani sistema EBA DMS.

e-Križanka je v celoti interno razvita aplikacija, sestavljena iz dveh komponent. Osnovna komponenta je sama spletna aplikacija e-Križanka, druga komponenta, ki je prav tako pomembna, pa je spletna storitev e-Križanka, ki skrbi za izvajanje postopka parafiranja v sistemu EBA DMS. Obe komponenti aplikacije e-Križanka sta razviti v tehnologiji ASP .NET.

Na strani sistema EBA DMS imamo prav tako dve komponenti. Prva je spletna storitev Križanka, preko katere spletna aplikacija e-Križanka vstavlja dokumente v sistem EBA DMS. Druga je uporabniški modul Križanka, ki razširja uporabniški vmesnik s funkcionalnostmi za podporo potrjevanja dokumentov iz spletne aplikacije e-Križanka. Obe komponenti sistema EBA DMS sta razviti v razvojnem okolju EBA Developer v jeziku Javascript.



Slika 28: Prikaz povezanosti komponent spletne aplikacije e-Križanka s sistemom EBA DMS in obratno

Komunikacija poteka samo med dvema paroma komponent in izključno prek spletnih storitev. V naslednjih dveh poglavjih bosta predstavljena oba para komponent in interakcije med komponentami v paru.

6.2 Spletna aplikacija e-Križanka in spletna storitev Križanka v sistemu EBA DMS v postopku priprave in shranjevanja dokumenta

Spletna aplikacija e-Križanka predstavlja uporabniški vmesnik, preko katerega uporabniki aplikacije vnašajo podatke, na podlagi katerih se kasneje pripravijo različni dokumenti. Uporabnik sproži proces priprave in pošiljanja dokumenta v sistem EBA DMS s klikom na gumb. Priprava vsakega dokumenta se začne s preverjanjem, ali so vsi zahtevani podatki vnešeni in ali so podatki vrednostno ustrezni. V primeru, da preverjanje ne uspe, je postopek priprave dokumenta prekinjen, uporabnik pa je o napakah obveščen z opozorilom. V nasprotnem primeru, ko preverjanje uspe, se sproži priprava dokumenta, ki ji sledi pošiljanje dokumenta v sistem EBA DMS preko spletne storitve Križanka na strani sistema EBA DMS. Opis vmesnika spletne storitve Križanka v sistemu EBA DMS v jeziku WSDL je naveden v prilogi 9.4.

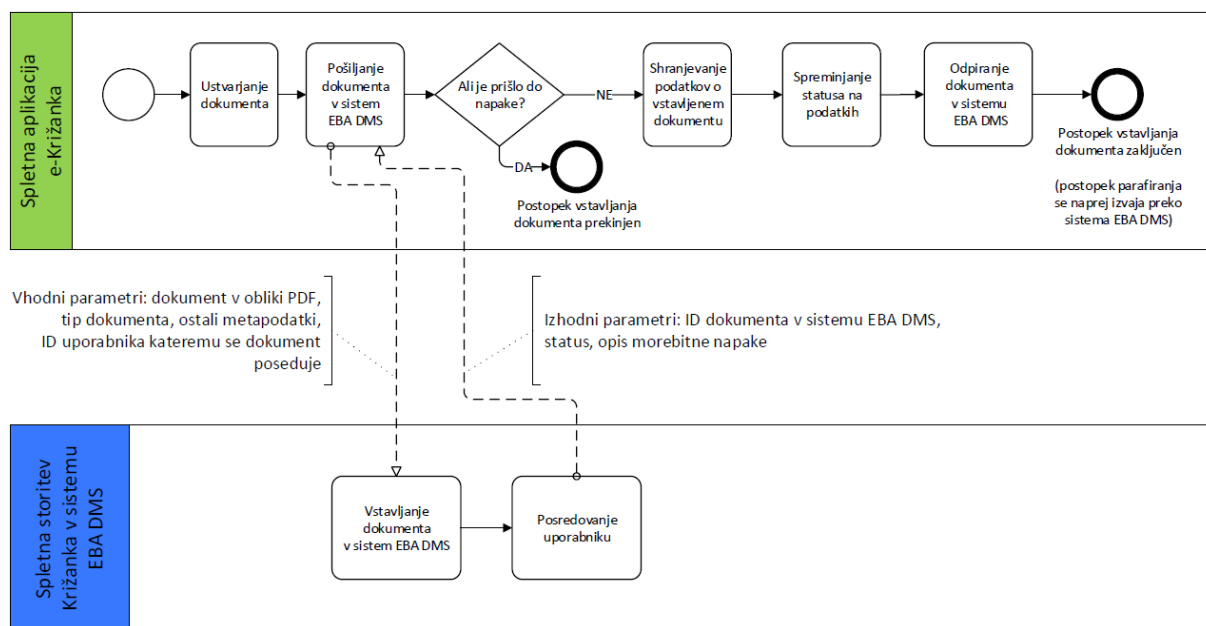
Spletna storitev pridobi vse potrebne podatke za uspešno shranjevanje dokumenta v sistem EBA DMS preko vhodnih parametrov klica spletne storitve. Ko je dokument shranjen v sistem EBA DMS, ga spletna storitev posreduje v pisarno uporabniku, ki je bil naveden v vhodnem parametru klica spletne storitve, da ga bo lahko kot prvi parafiral. Pri vseh tipih dokumentov e-Križanke je dokument vedno poslan najprej uporabniku, ki je dokument pripravil. (Namen tega je, da uporabnik, ki kreira dokument, naprej sam preveri pravilnost podatkov na dokumentu in svoje strinjanje z dokumentom potrdi s parafo. Vendar je to že del naslednjega poglavja.)

Obdelava vhodne zahteve spletne storitve se zaključi z odgovorom na vhodno zahtevo. Če v postopku ni prišlo do nobenih težav, se kot odgovor sporoči ID vstavljenega dokumenta v sistemu EBA DMS in status, da je vse v redu. Če pa je prišlo do težav in dokument ni bil uspešno shranjen, se vrne negativni status in opis napake, ki ga vmesnik spletne aplikacije e-Križanka prikaže uporabniku.

Po uspešnem shranjevanju dokumenta v sistem EBA DMS si spletna aplikacija e-Križanka shrani podatke, ki jih je prejela kot odgovor na klic spletne storitve v svojo podatkovno bazo. Shrani si ID dokumenta in lokacijo dokumenta v sistemu EBA DMS ter interni status dokumenta. Interni status dokumenta je takoj po vstavljanju dokumenta vedno enak Vpisan. Po uspešnem shranjevanju dokumenta se spremeni tudi status osnovnih podatkov, na podlagi katerih je bil pripravljen dokument. S spremembo statusa na podatkih lahko zagotovimo, da so

podatki z obstojem dokumenta zaklenjeni in jih uporabnik ne more spreminjati. Če želi uporabnik ponovno urejati dokumente, mora najprej veljavni dokument v sistemu EBA DMS uničiti.

Ker mora uporabnik, ki je ustvaril dokument, kot prvi dokument tudi parafirati, mu ga za konec spletna aplikacija e-Križanka še avtomatsko odpre v sistemu EBA DMS.



Slika 29: Vstavljanje dokumenta iz aplikacije e-Križanka v sistem EBA DMS

Nadaljevanje postopka, izvedba postopka parafiranja, je že del komunikacije med drugim parom štirih komponent, ki so del integracije spletne aplikacije e-Križanka in sistema EBA DMS. Ta del bo predstavljen v naslednjem poglavju.

6.3 Uporabniški modul Križanka v sistemu EBA DMS in spletna storitev e-Križanka v postopku parafiranja

Postopek parafiranja dokumentov, ki izvirajo iz spletne aplikacije e-Križanka, za uporabnika navidezno v celoti poteka v okviru Klienta EBA, vendar temu ni tako. Uporabniški modul Križanka v sistemu EBA DMS tudi samega uporabniškega vmesnika Klienta EBA prav nič ne spreminja, vsaj z uporabnikovega stališča ne. V ozadju izvajanja samo razširja dve izmed akcij, ki ju uporabnik lahko izvede nad katerimkoli dokumentom. Ti dve razširjeni akciji sta

parafiranje in uničevanje dokumenta. V celoti pa izvajanje parafiranja nadzira in usmerja spletna storitev e-Križanka. Opis vmesnika spletne storitve e-Križanka v jeziku WSDL je naveden v prilogi 9.5.

Odpiranje dokumenta

Prvi uporabnik, ki mora parafirati vsak dokument tipa Križanka, je vedno uporabnik, ki je dokument ustvaril in shranil v sistem EBA DMS. Temu uporabniku se po uspešnem shranjevanju dokumenta v sistemu EBA DMS dokument tudi že avtomatsko odpre v Klientu EBA.

Vsak naslednji uporabnik, ki v sistemu EBA DMS prejme dokument tipa Križanka v parafiranje, je o tem obveščen preko elektronske pošte. Sporočilni sistem sistema EBA DMS sicer omogoča avtomatsko obveščanje o posredovanih dokumentih, vendar se za dokumente tipa Križanka obveščanje v sistemu EBA DMS ne uporablja. Namesto sistemskih elektronskih sporočil se v okviru spletne storitve e-Križanka ob posredovanju dokumentov uporabnikom pošiljajo tudi vsebinsko bolj bogata elektronska sporočila, kot bi jih sicer poslal sistem EBA DMS. V elektronskem sporočilu je navedena tudi povezava, preko katere uporabniki odprejo dokument v Klientu EBA.

Izvedba akcije

Naloga vsakega uporabnika, ki mu je dodeljen dokument v parafiranje, je, da preveri njegovo pravilnost in to potrdi s parafiranjem dokumenta. V primeru, ko uporabnik ugotovi, da dokument ni pravilen v uporabniškem vmesniku izbere akcijo Uniči, v primeru, ko pa je dokument pravilen pa dokument potrdi z izbiro akcije Parafiraj. V obeh primerih se v ozadju izvede klic spletne storitve e-Križanka s parametri ID dokumenta, tip akcije in ID uporabnika.

Obdelava klica spletne storitve

Spletna storitev e-Križanka po sprejemu vhodne zahteve najprej izvede zaporedje kontrol o pravilnosti podatkov in upravičenosti uporabnika, da izvede akcijo. Kontrole so npr.:

- ali dokument obstaja v spletni aplikaciji e-Križanki,
- ali uporabnik obstaja v spletni aplikaciji e-Križanka,

- ali je zahtevana akcija dovoljena glede na status (dokument je lahko npr. že v celoti potrjen),
- ali je uporabnik na vrsti za izvedbo akcije (dostopa do dokumenta nima samo trenutni uporabnik, parafiranje pa je dovoljeno samo njemu),
- ali uporabnik lahko izvede akcijo zaradi nastavljenega nadomeščanja (uporabnik lahko sicer ni trenutni uporabnik, ki bi moral parafirati, lahko pa nadomešča trenutnega uporabnika in mu moramo dovoliti parafiranje).

Sledi določitev pravila za obdelavo vhodne zahteve. Pravila so definirana glede na tip dokumenta, akcijo in status dokumenta natančno. Če imamo 11 osnovnih tipov dokumentov, dve možni akciji in v povprečju za vsako akcijo še 3 statuse, se torej vsaka vhodna zahteva v spletni storitvi e-Križanka obdela po enem od 66 pravil.

Izvedba posameznega pravila lahko vsebuje naslednje korake:

- kdo je naslednji, ki je na vrsti za parafiranje dokumenta (izhodni parameter: ID uporabnika),
- nastavitev statusa osnovnih podatkov v spletni aplikaciji e-Križanka,
- nastavitev statusa in lokacije dokumenta v spletni aplikaciji e-Križanka,
- pošiljanje elektronskega sporočila naslednjemu, ki je na vrsti za parafiranje dokumenta,
- če je parafiranje s to akcijo zaključeno, se v sistemu EBA DMS zahteva še izvedba arhiviranja dokumenta (izhodni parameter: arhiviranje),
- ali je v obdelavi prišlo do napake (izhodna parametra: status in opis napake).

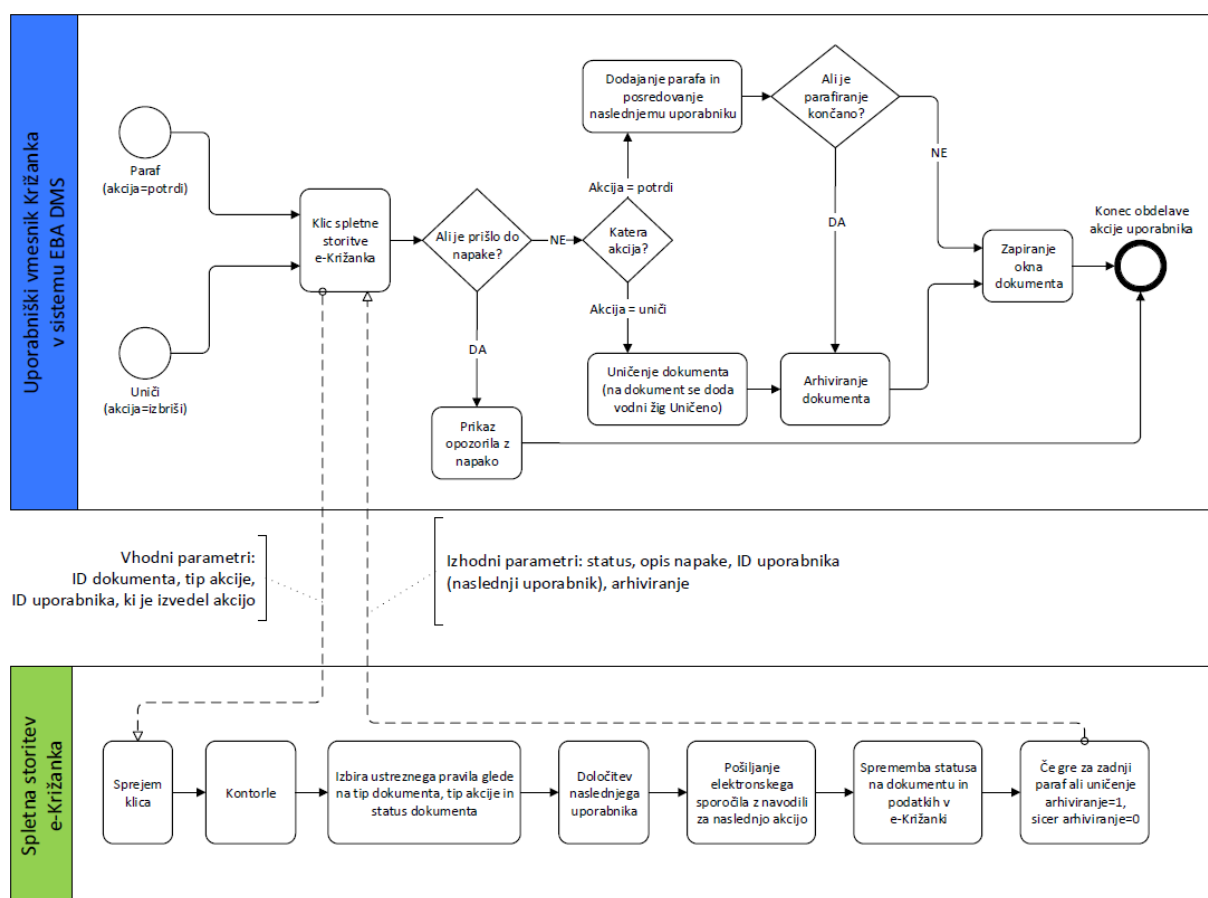
Obdelava vhodne zahteve se zaključi s podajanjem odgovora, ki vsebuje podatke o naslednjem uporabniku, ki mora parafirati dokument, parameter arhiviranje, podatek o statusu obdelave zahteve in pa opis napake, če je pri obdelavi vhodne zahteve do nje prišlo.

Obdelava odgovora na klic spletne storitve oz. nadaljevanje obdelave akcije uporabnika

Modul Križanka v sistemu EBA DMS nadaljuje z obdelavo akcije uporabnika, ko prejme odgovor na klic spletne storitve e-Križanka. Če je pri obdelavi prišlo do kakršnekoli napake, uporabniku prikaže obvestilo z opisom napake in prekine izvedbo akcije, ki je bila zahtevana s strani uporabnika.

Če pa pri obdelavi klica spletne storitve ni prišlo do napake, Klient EBA izvede zahtevano akcijo do konca. V primeru, ko je šlo za akcijo Uniči, se dokumentu v sistemu EBA DMS spremeni status v Uničeno, čez sliko dokumenta se doda vodni žig Uničeno, dokument se arhivira, odprto okno dokumenta pa se uporabniku avtomatsko zapre. V primeru, ko gre za akcijo Parafiraj, pa se na dokument doda parafa trenutnega uporabnika, dokument se posreduje naslednjemu uporabniku, če je dokument v celoti potrjen, se ga arhivira, odprto okno dokumenta pa se uporabniku avtomatsko zapre.

Postopek, opisan v tem poglavju, se ponavlja toliko časa, dokler dokument v spletni aplikaciji e-Križanka ni v končnem statusu.



Slika 30: Izvajanje akcij nad dokumenti e-Križanke v sistemu EBA DMS in nadziranje postopka s strani spletne storitve e-Križanka

6.4 Orodje za nadzor nad dokumenti spletne aplikacije e-Križanka

Uporabniki lahko v tekočem obračunskem mesecu ustvarijo veliko dokumentov. Vsi se nahajajo v sistemu EBA DMS in prav vsi se tudi po večkrat posredujejo od enega uporabnika k drugemu. Sledenje dokumentom je zaradi zgoraj naštetih dejstev preko Klienta EBA za uporabnike pretežko. Prav zato je bilo v spletni aplikaciji e-Križanka razvito orodje za nadzor nad dokumenti spletne aplikacije e-Križanka.

The screenshot shows the 'Orodje - Pisarna' interface. At the top, it displays the user 'Pintar Matej' and search filters for 'Obdobje' (01.04.2016 to 15.05.2016), 'Teden' (18. teden), and 'Mesec' (4. mesec). Below these are tabs for 'Dokumenti', 'Redno delo', 'Opravljenega dela', and 'Moja pisarna'. A 'Tip dokumenta' section includes checkboxes for 'Dopusti', 'Odreditve nadurnega dela', 'Nalogi nadurnega dela', 'Poročila opravljenega dela', 'Dežurna lista', 'Spremembe reg. časa', 'Službene poti', and 'Obračuni potnih stroškov'. A 'Status dokumenta' section includes checkboxes for 'Vpisan', 'Potrdil vpisovalec', 'Potrdil vodja', 'Potrdil direktor', and 'Potrjena celota'. The main area is a table with columns: Tip dokumenta, EBA ID, Ustvarjeno, Status, and Pisarna. A tooltip for 'Dopusti' shows the time '5.5.2016 (7:00 - 15:00) Testni vpis dopusta'. The table contains 11 rows of document entries, with the last row showing 'Prikaz opravljenega dela' by 'Pintar Matej'.

Tip dokumenta	EBA ID	Ustvarjeno	Status	Pisarna
Dopusti	43d6df2c6b236ef07c18e358af741bff	4.4.2016	Potrjena celota	Pintar Matej (ARHIV)
Sprememba registracijskega časa	3549e661b3a187c2f71a5f6d683b1f0e	6.4.2016	Potrjena celota	Pintar Matej (ARHIV)
Sprememba registracijskega časa	fc930630da950d3709d0503a87933e24	7.4.2016	Potrjena celota	Pintar Matej (ARHIV)
Dopusti	7ace9414c31067b66c71e44039655051	12.4.2016	Potrjena celota	Pintar Matej (ARHIV)
Sprememba registracijskega časa	c60745ddb48f2da8aa7b776d63e1e7b4	12.4.2016	Potrjena celota	Pintar Matej (ARHIV)
Nalog za nadurno delo	41ce0b36fa68d90931e60649ba25f8d8	13.4.2016	Potrjena celota	Pintar Matej (ARHIV)
Dopusti	255177394fa2f4964e45743b8897a943	18.4.2016	Potrjena celota	Pintar Matej (ARHIV)
Odreditve nadurnega dela	f553d3ab803107d99f5bce4c349297ca	27.4.2016	Potrjena celota	Pintar Matej (ARHIV)
	c88e65bf2d9af7463782a229b33936b	27.4.2016	Potrjena celota	Pintar Matej (ARHIV)
	51bd3f77d728b8eae76bf5f8e8f64356	28.4.2016	Potrjena celota	Pintar Matej (ARHIV)
	615b6d12cd39353eb027d30280b6da4	2.5.2016	Potrjena celota	Pintar Matej (ARHIV)
Dopusti	095382f618edaf8924339c340ad4311	3.5.2016	Vpisan	Pintar Matej
Prikaz opravljenega dela	1a4d8e6cff314ce83256b25e44c3ccd7	3.5.2016	Potrdil vodja	Pintar Matej

Slika 31: Orodje za nadzor nad dokumenti spletne aplikacije e-Križanka

Orodje uporabniku v prvi vrsti ponuja veliko možnosti za filtriranje seznama prikazanih dokumentov. Glede na datum nastanka dokumentov lahko filtrira na tri načine. Filtrira pa lahko tudi na tip dokumenta in status dokumenta.

Uporabnik lahko iz seznama prikazanih dokumentov za posamezen dokument razbere veliko informacij. Na voljo so mu informacije o tipu dokumenta, o datumu nastanka dokumenta, o trenutnem statusu dokumenta in o trenutni lokaciji dokumenta v sistemu EBA DMS. Tudi o sami vsebini dokumenta lahko uporabnik sklepa že brez da bi dokument sploh odprl v sistemu EBA DMS. S postavitvijo miškega kazalca nad naziv tipa dokumenta se uporabniku namreč prikaže majhno pojavno okno s kratkim opisom vsebine dokumenta.

Če pa uporabnik iz samega seznama vseeno ne uspe pridobiti zadostnih informacij, ima še vedno na voljo odpiranje dokumenta v sistemu EBA DMS. Vsak dokument v seznamu prikazanih dokumentov ima na začetku vrstice gumb, s katerim se uporabniku v Klientu EBA odpre dokument.

6.5 Nadomeščanje

Nadomeščanje je funkcionalnost, ki se uporablja v času odsotnosti zaposlenih. Glavni namen nadomeščanja je, da se poslovni procesi v podjetju vedno izvajajo nemoteno. Tudi takrat, ko zaposlenih, ki bi sicer morali aktivno sodelovati v procesu, ni prisotnih. Funkcionalnost nadomeščanja deluje na način, da se tistemu, ki nadomešča, omogoči vse akcije, ki bi jih sicer lahko oz. moral opraviti tisti, ki je nadomeščan.

V primeru nastopa odsotnosti je naloga tistega, ki bo odsoten, da v vseh sistemih, ki podpirajo nastavljanje nadomeščanja, nadomeščanje tudi nastavi. Če odsotnost ni bila planirana in je nastopila nenadno, lahko nadomeščanje vedno nastavijo tudi administratorji sistemov.

Sistema EBA DMS in Dynamics NAV ter spletna aplikacija e-Križanka so pred tesno integracijo vsak zase že podpirali funkcionalnost nadomeščanja. Slabost starega stanja je bila v tem, da je uporabnik moral nastaviti nadomeščanje na treh mestih. S tem se je zelo povečala možnost za napake. Bodisi so bili vnosi oz. nastavitve nadomeščanja med seboj neuskklajene bodisi je uporabnik celo pozabil nastaviti nadomeščanje v kakšnem od sistemov.

6.5.1 Splošno nadomeščanje


Prav zaradi zgoraj opisanih slabosti je bil v spletni aplikaciji e-Križanka na strani Moje nastavitve dodan nov modul za nastavljanje nadomeščanja preko vseh sistemov z enim samim vnosom.


Uporabnik mora za uspešen vnos nadomeščanja iz seznama zaposlenih izbrati namestnika in določiti obdobje v katerem ga bo namestnik nadomeščal. Vnos potrdi s klikom na potrditveni gumb. V ozadju spletna aplikacija e-Križanka najprej z neposrednima posegoma v podatkovni bazi sistema EBA DMS in Dynamics NAV vnese enake nastavitve, kot so bile vnešene na strani


e-Križanke, še v povezana sistema. Konča pa s shranjevanjem podatkov o nadomeščanju v spletni aplikaciji e-Križanka.



Po preteku obdobja, za katero je bilo nadomeščanje nastavljeno, se v vseh sistemih nadomeščanje avtomatsko onemogoči. V primeru, ko uporabnik želi zaradi kakršnegakoli razloga sam preklicati nadomeščanje, podobno kot za nastavljanje nadomeščanja v spletni aplikaciji e-Križanka na strani Moje nastavitve, preko gumba za preklic nadomeščanja prekliče nadomeščanje v vseh sistemih. V ozadju se prav tako kot za nastavljanje nadomeščanja spremembe vnesejo še v sistema EBA DMS in Dynamics NAV z neposrednima posegoma v podatkovni bazi obeh sistemov. Nazadnje se podatki o preklicu nadomeščanja zabeležijo še v spletni aplikaciji e-Križanka.

Nastavljanje nadomeščanja preko eKrižanke avtomatsko zagotovi nadomeščanje tudi v okolju Dynamics NAV in EBA.

Namestnik 

Začetek nadomeščanja 

Konec nadomeščanja 

Slika 32: Modul za nastavljanje splošnega nadomeščanja preko vseh sistemov v e-Križanki

6.5.2 Področno nadomeščanje

Nadomeščanje, opisano v prejšnjem poglavju, zadostuje v večini primerov. Gre za primere, ko neko osebo nadomešča na vseh področjih samo ena oseba. Problem splošnega nadomeščanja v praksi pa je, da za določena področja ena oseba, ki nadomešča, včasih ni dovolj. Oseba, ki je nadomeščana, je lahko npr. tudi član kakšne komisije oz. skupine. Znotraj komisij in skupin navadno veljajo posebna pravila, ki sicer ne veljajo na nivoju podjetja. Tudi o tem, kdo iz komisije oz. skupine nadomešča koga.

Prav zaradi zgoraj navedenih razlogov je bilo potrebno za potrebe potrjevanja dokumentov na posameznih področjih vzpostaviti tudi način, ki bi omogočal področno nadomeščanje. Prvi primer področnega nadomeščanja, na katerega smo naleteli v podjetju, je bilo področje osnovnih sredstev in drobnega inventarja. Na tem področju je v podjetju definirana komisija, ki je vključena tudi v verigo potrjevanja dokumentov, povezanih s tem področjem. Poslovni

proces za področje osnovnih sredstev in drobnega inventarja je sicer skoraj v celoti del sistema Dynamics NAV, edino večnivojsko potrjevanje ustvarjenih dokumentov se izvede v sistemu EBA DMS. Iz sistema Dynamics NAV so v sistem EBA DMS ustvarjeni naslednji tipi dokumentov:


- predlogi prenosov osnovnih sredstev,
- sklepi prenosov osnovnih sredstev,
- predlogi prenosov drobnega inventarja,
- sklepi prenosov drobnega inventarja,
- predlogi odpisov osnovnih sredstev,
- sklepi odpisov osnovnih sredstev,
- predlogi odpisov drobnega inventarja in
- sklepi odpisov drobnega inventarja.


Poleg samega dokumenta v obliki PDF sistem Dynamics NAV v sistem EBA DMS pošlje tudi nastavitveni niz za večnivojsko parafiranje, npr. 1=1354,2=1589,2=1486,3=1156. Pomen nastavitvenega niza je bil predstavljen v poglavju 4.2. Ker je sistem EBA DMS v celoti odgovoren za izvedbo večnivojskega parafiranja, je bilo za vpeljavo področnega nadomeščanja za potrebe področja osnovnih sredstev in drobnega inventarja potrebno samo v sistemu EBA DMS zagotoviti upoštevanje nastavitvev za področno nadomeščanje.


6.5.2.1 Vnos nastavitvev za področno nadomeščanje



Enako kot za splošno nadomeščanje je bil v spletni aplikaciji e-Križanka na strani Moje nastavitve poleg modula z splošno nadomeščanje dodan še en nov modul za nastavljanje področnega nadomeščanja. Uporabnik mora za uspešen vnos področnega nadomeščanja iz seznama zaposlenih izbrati namestnika in določiti obdobje v katerem ga bo namestnik nadomeščal. Vnos potrdi s klikom na potrditveni gumb. Za razliko od splošnega nadomeščanja se za področno nadomeščanje podatki shranijo le v podatkovni bazi spletne aplikacije e-Križanka. Shranjevanje v sistem EBA DMS in Dynamics NAV ni možno, ker sistema ne podpirata tovrstnih dodatnih nastavitvev.

Nastavljanje nadomeščanja za področje OS in DI.

Namestnik 

Začetek nadomeščanja 

Konec nadomeščanja 

Slika 33: Modul za nastavljanje področnega nadomeščanja preko vseh sistemov v e-Križanki

6.5.2.2 Modul NAV v sistemu EBA DMS

Modul NAV v sistemu EBA DMS, ki je bil sicer opisan v poglavju 4.2, je bilo potrebno za potrebe večnivojskega potrjevanja dokumentov s področja osnovnih sredstev in drobnega inventarja razširiti na način, da so se za dokumente s tega področja upoštevale še nastavitve področnega nadomeščanja, ki so bile sicer shranjene v spletni aplikaciji e-Križanka.

Splošno nadomeščanje v sistemu EBA DMS deluje na način, da osebi, ki nadomešča, sistem samo navidezno omogoči dostop do dokumentov, do katerih ima sicer dostop oseba, ki je nadomeščana. Ker je področno nadomeščanje neodvisno od splošnega nadomeščanja, moramo za omogočanje dostopa do dokumentov področja osebi, ki na nekem področju nadomešča, dodeliti dostop do dokumentov fizično in ne samo navidezno, kot se to zgodi v primeru splošnega nadomeščanja.

Kot je bilo navedeno v poglavju 4.2, modul NAV v sistemu EBA DMS v vsakem koraku oz. po vsaki akciji dokument pošlje v parafiranje vsem paraferjem, ki so na trenutnem nivoju in še niso parafirali dokumenta. Na tem delu se za dokumente s področja osnovnih sredstev in drobnega inventarja za vsakega paraferja, ki je na trenutnem nivoju parafiranja in še ni parafiral dokumenta, s poizvedbo na podatkovno bazo spletne aplikacije e-Križanka preveri, če ima parafer nastavljeno trenutno veljavno področno nadomeščanje. Če je področno nadomeščanje nastavljeno, se dokument ne pošlje v parafiranje osebi, ki je navedena v tabeli, ki določa potek večnivojskega parafiranja, ampak kar osebi, ki je nastavljena za nadomeščanje. V tabeli, ki določa potek večnivojskega parafiranja, se v vrstici osebe, ki ima nastavljeno nadomeščanje stolpcu Opomba nastavi vrednost Nastavljeno nadomeščanje.

Zgornja razširitev osnovnega načina delovanja modula NAV v sistemu EBA DMS zagotavlja samo, da se dokumenti po opravljenih akcijah (npr. po parafiranju enega izmed paraferjev) pošljejo osebam, za katere predvidevamo, da so prisotni v podjetju. Ampak še to samo v primerih, ko je področno nadomeščanje ustrezno nastavljeno.

Zgornja rešitev ima dve pomanjkljivosti. Prva je ta, da se pogosto zgodi, da je bila oseba v času parafiranja predhodnika še prisotna ali pa zgolj nadomeščanje še ni bilo nastavljeno. Področno nadomeščanje sicer lahko nastavimo naknadno, ampak dokument, ki ga mora oseba, ki je odstotna, parafirati, je že pri tej osebi v pisarni in čaka, da ga le-ta parafira. Druga pomanjkljivost pa je, da je bilo v času parafiranja predhodnika nastavljeno področno nadomeščanje za naslednjega paraferja in je bil dokument tudi poslan tej osebi v parafiranje. Ampak ta oseba zaradi kateregakoli razloga dokumenta ni parafirala. Ker so dokumenti pri področnem nadomeščanju fizično preposredovani med uporabniki, je po preteku ali preklicu nadomeščanja ta dokument vseeno obstal pri osebi, ki je nadomeščala, in čaka, da ga le-ta parafira. Ne glede na to, da dokument čaka, da ga ta oseba parafira, bi bila akcija parafiranja po preteku ali preklicu področnega nadomeščanja preklicana zaradi preverjanja, ali je oseba na vrsti za parafiranje ali ne.

Rešitev zgoraj opisanih pomanjkljivosti bo predstavljena v naslednjem poglavju.

6.5.2.3 Opravilo za uveljavitev sprememb glede na nastavitve področnega nadomeščanja

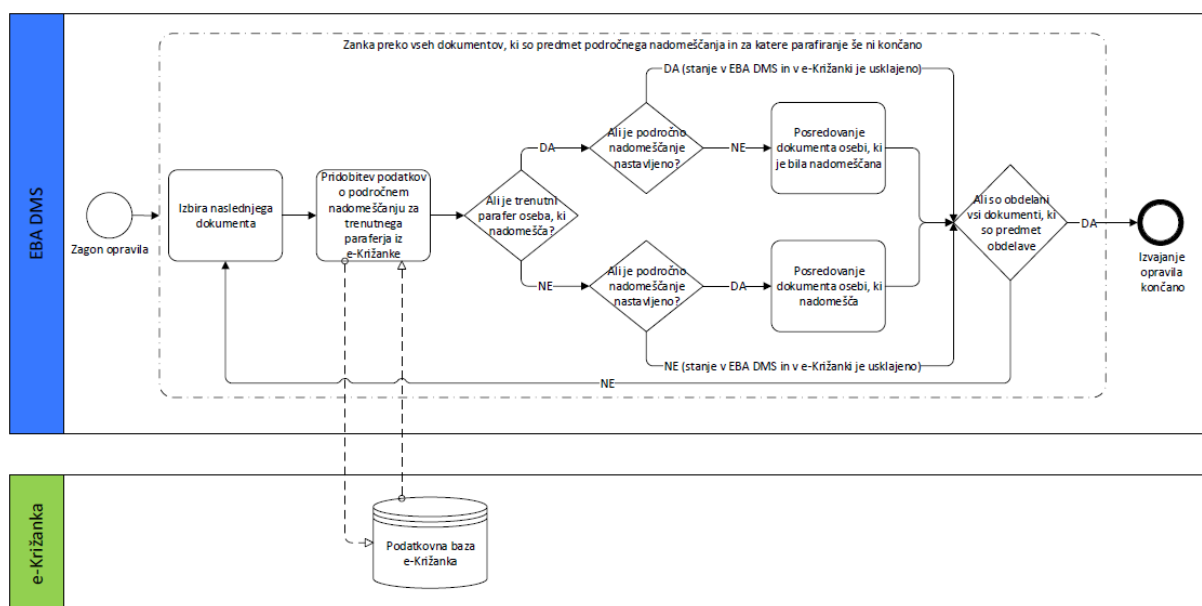
Za potrebe uveljavljanja sprememb, do katerih pride zaradi sprememb pri nastavitvah za področno nadomeščanje, je bilo treba pripraviti opravilo v sistemu EBA DMS, ki v določenih časovnih intervalih preposreduje dokumente, ki so predmet področnega nadomeščanja. Bodisi uveljavlja na novo nastavljena področna nadomeščanja bodisi uveljavlja na novo preklicana oz. pretečena področna nadomeščanja. V obeh primerih gre za posredovanje dokumentov. V prvem primeru od osebe, ki je nadomeščana, osebi, ki bo nadomeščala, v drugem primeru pa od osebe, ki je nadomeščala, osebi, ki je bila nadomeščana.

Opravilo za uveljavitev sprememb glede na nastavitve področnega nadomeščanja je eden izmed projektov, razvitih v razvojnem orodju EBA Developer. Projekt je kot vsak drug uvožen v sistem EBA DMS, za njegovo izvajanje pa skrbi strežniški del sistema EBA DMS, natančneje

storitev za izvajanje ponavljajočih se opravil (EBA Scheduler Service). Opravilo se izvaja enkrat na dan v jutranjem času.

Opravilo v zanki pregleda vse dokumente s področja osnovnih sredstev in drobnega inventarja, ki so v postopku potrjevanja. Za vsak dokument, ki je predmet obdelave, opravilo preveri, ali je za trenutnega paraferja v spletni aplikaciji e-Križanka nastavljeno področno nadomeščanje. Preverjanje se izvede s poizvedbo SQL nad podatkovno bazo spletne aplikacije e-Križanka. Pridobijo se podatki o tem, kdo nadomešča trenutnega paraferja in v katerem obdobju ga nadomešča. Glede na pridobljene podatke iz spletne aplikacije e-Križanka in glede na trenutno stanje dokumenta v sistemu EBA DMS ima opravilo tri možnosti:

- posredovanje dokumenta osebi, ki je nastavljena za nadomeščanje,
- posredovanje dokumenta nazaj osebi, ki je bila nadomeščana,
- obstoječe stanje je ustrezno.



Slika 34: Obdelava opravila za uveljavitev sprememb glede na nastavitve področnega nadomeščanja v spletni aplikaciji e-Križanka

6.6 Integracija e-Križanke in strežnika Microsoft Exchange

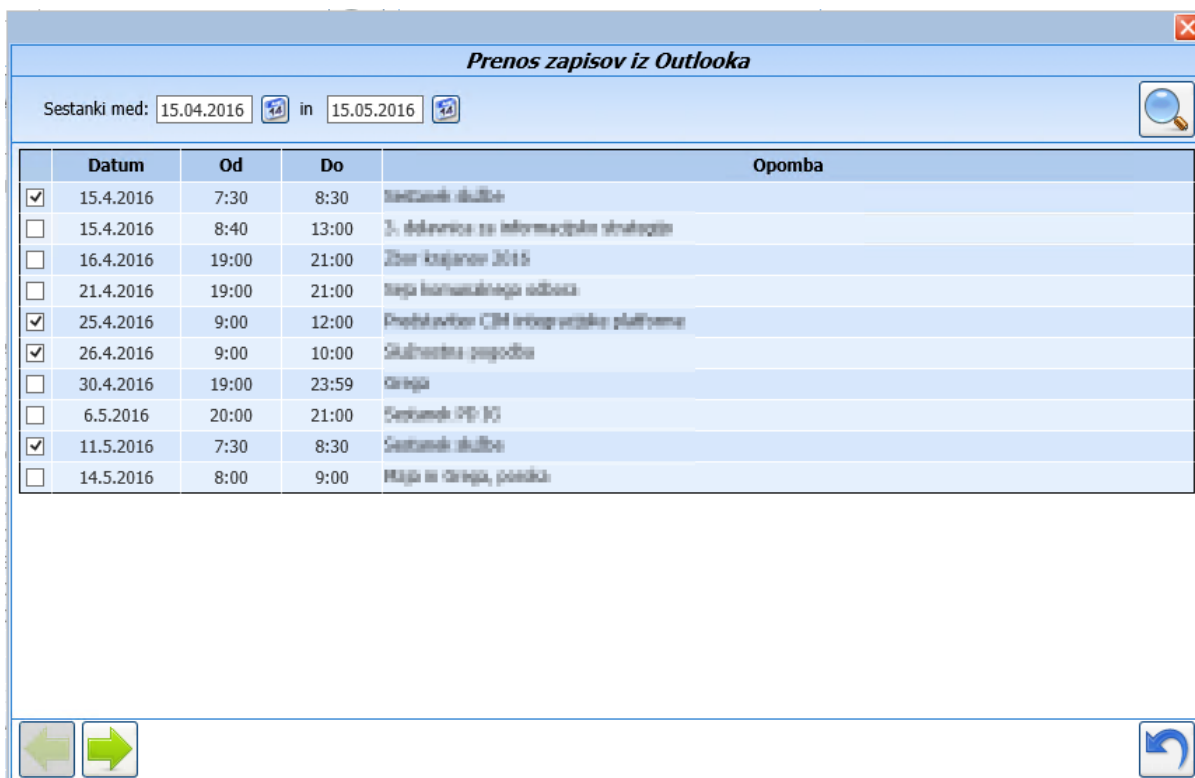
Na tem mestu lahko na kratko predstavimo še eno izmed integracij, ki neposredno sicer ni povezana z integracijo s sistemom EBA DMS, posredno pa.

Uporabniki morajo skozi mesec redno vnašati podatke o opravljenih delovnih urah na ustrezne delovne naloge v spletno aplikacijo e-Križanka. Ti vpisani podatki so lahko že takoj po vnosu predmet potrjevanja v sistemu EBA DMS (npr. dopusti, nadure ...). V nasprotnem primeru so predmet potrjevanja ob koncu meseca, ko se v sistemu EBA DMS potrjuje dokument Poročilo o opravljenem delu za zaposlenega. V končni fazi je vsak podatek, ki se vpiše v modul Vpis ur predmet potrjevanja v sistemu EBA DMS.

Vsi uporabniki e-Križanke imajo na voljo možnost, da uvozijo opravljene ure v modul Vpis ur v spletni aplikaciji e-Križanka, neposredno iz zapisov v koledarju strežnika Microsoft Exchange. Gre torej za podatke, ki so jih uporabniki sami vnesli v svoj koledar preko programa Microsoft Outlook ali pa so bili npr. avtomatsko generirani s sprejetjem povabila na sestanek.

Možnost je bila v prvi vrsti razvita za uporabnike, ki so na vodilnih položajih v podjetju. Prav pri njih se namreč dogaja, da imajo celoten dan zapolnjen s sestanki oz. vnaprej planiranimi dogodki. Le-ti so že v koledarju uporabnikov zelo podrobno definirani. Podatki, kot so datum in ura začetka ter konca dogodka in pa opis dogodka so pravzaprav vsi, ki jih potrebujemo za vpis podatkov v spletno aplikacijo e-Križanka. Potrebno je le še določiti delovni nalog, na katerega se bodo ure zabeležile in je postopek zaključen. Privzeto se za vsak dogodek izbere režijski delovni nalog uporabnika. Vedno pa lahko uporabnik spremeni vse podatke.

Uporabniški vmesnik za uvoz podatkov iz koledarja v spletno aplikacijo e-Križanka je sestavljen iz dveh oken. Na prvem oknu uporabnik določi obdobje, za katerega naj se mu prikažejo dogodki iz koledarja. Privzeto je izbrano obdobje trenutnega tedna. V seznamu prikazanih dogodkov ima nato uporabnik možnost izbire dogodkov, za katere želi, da se podatki uvozijo v e-Križanko.



Slika 35: Uporabniški vmesnik za izbiro dogodkov iz koledarja za uvoz v spletno aplikacijo e-Križanka

Ko uporabnik izbere dogodke za uvoz, se s klikom na gumb premakne na drugo okno uporabniškega vmesnika za uvoz podatkov iz koledarja. V drugem oknu se v zgornjem delu okna izpišejo dogodki, ki so bili s strani uporabnika izbrani na prvem oknu. V spodnjem delu okna pa se ob izbiri dogodka iz seznama v zgornjem delu okna prikaže uporabniški vmesnik za urejanje podrobnosti o izbranem dogodku. Uporabnik lahko spremeni privzeto nastavljen oz. izbran režijski delovni nalog, časovne podatke o dogodku, vrsto plačila in tudi opis dogodka. S potrditvijo sprememb se ustrezno posodobi seznam dogodkov v zgornjem delu okna.

Dogodki v seznamu v zgornjem delu okna so lahko izpisani s črno ali pa z rdečo barvo. Črno obarvani so tisti, ki so lahko brez posredovanja uporabnika že uspešno uvoženi v spletno aplikacijo e-Križanka. Rdeče obarvane pa mora uporabnik pred uspešnim uvozom najprej ustrezno popraviti. Vzrok za rdeče obarvane dogodke je navadno časovno prekrivanje podatkov v seznamu dogodkov s podatki, ki so že vpisani v spletno aplikacijo e-Križanka. S klikom na potrditveni gumb uporabnik uvozi črno obarvane dogodke v spletno aplikacijo e-Križanka in s tem tudi zaključi postopek uvoza podatkov iz strežnika Microsoft Exchange.

Vpis podatkov v ure

Datum	Od	Do	Vpis od	Vpis do	Naloga	Opomba
15.4.2016	7:30	8:30	15:00	16:00	REŽIJA	Seznanek službe
25.4.2016	9:00	12:00	15:00	16:00	REŽIJA	Preoblikovanje CIM integracijske platforme
26.4.2016	9:00	10:00	15:00	16:00	REŽIJA	Službena pogodba
11.5.2016	7:30	8:30	7:00	9:00	REŽIJA	Seznanek službe

Naloga:

Začetek dela: : Konec dela: :

Vsebina dela:

Vrsta plačila:

Delovni nalog:

Slika 36: Uporabniški vmesnik za urejanje podrobnosti o posameznih dogodkih iz koledarja za uvoz v spletno aplikacijo e-Križanka

6.7 Povzetek

V tem poglavju smo se seznanili s spletno aplikacijo e-Križanka, ki je poleg elektronskega dokumentnega sistema EBA DMS edini izmed produktov, predstavljenih v tem magistrskem delu, ki ga uporabljajo prav vsi zaposleni v podjetju.

Predstavljene so bile štiri komponente, preko katerih je izvedena integracija z elektronskim dokumentnim sistemom EBA DMS. Kot smo spoznali, poteka komunikacija med temi komponentami izključno preko spletnih storitev. Edino za potrebe nastavljanja nadomeščanja v sistemu EBA DMS je integracija zagotovljena na podatkovnem nivoju.

E-Križanka pa ni tesno integrirana le s sistemom EBA DMS. Kar v nekaterih modulih spletne aplikacije je namreč izvedena tudi tesna integracija s sistemom Dynamics NAV, kar pa že presega predmet te magistrske naloge.

7 Razvoj upravljanja z elektronskimi dokumenti v prihodnje

Razvoj na področju upravljanja z elektronskimi dokumenti v podjetju pa s tem še zdaleč ni zaključen. Postavljenih imamo več ciljev, ki še čakajo na realizacijo.

Prvi cilj je enotno izdajno mesto za vse izhodne dokumente v podjetju. V podjetju je v uporabi še kar nekaj aplikacij, iz katerih se dokumenti tiskajo na papir neposredno iz aplikacije same, arhivirajo pa se šele kasneje bodisi s skeniranjem lastnoročno podpisanega prej natisnjene dokumenta bodisi z uvozom datoteke PDF, ki je bila pripravljena za arhiviranje. Na ta način nikoli ne moremo biti popolnoma prepričani, da je dokument, ki je bil poslan stranki, enak arhiviranemu. Stanje, ki ga želimo doseči je, da vsaka aplikacija izhodne dokumente vstavlja neposredno v sistem EBA DMS, od koder se potem tudi tiskajo in kasneje oddajo na pošto. Na ta način bi pridobili poenoten dnevnik vseh izhodnih dokumentov. Trenutno je ta vzpostavljen samo za elektronsko poslano dokumente, tj. e-račune.

Drugi cilj v podjetju je vpeljava črtne kode na vseh izdanih dokumentih in avtomatsko verzioniranje dokumentov v sistemu EBA DMS v primeru, da se neka koda ob uvozu v sistem EBA DMS pojavi ponovno.

Tretji cilj v podjetju je zagotovitev popolnega elektronskega poslovanja za zunanje uporabnike. Cilj je, da se zunanjim uporabnikom zagotovi informacijska podpora za oddajanje vlog in drugih dokumentov v elektronski obliki, avtomatsko shranjevanje le-teh v elektronskem dokumentnem sistemu EBA DMS in avtomatsko pošiljanje le-teh v delovne tokove poslovnih procesov, v katerem nastopajo.

Četrty in tudi najobsežnejši cilj pa bo prehod na prihajajočo spletno verzijo sistema EBA DMS. Postopek prehoda bo lahko postopen, saj bodo v vmesnem obdobju zagotovljeni vmesniki, ki bodo zagotavljali delovanje vseh obstoječih rešitev. Prehod na spletno verzijo bo lepa priložnost za izboljšanje integraciji vseh sistemov, povezanih s sistemom EBA DMS. Možnosti za izboljšave so številčne, predvsem pri poenotenju načinov integracij med različnimi sistemi, ki so integrirani z elektronskim dokumentnim sistemom EBA DMS. Iz uporabniškega vidika bo uvedba spletne verzije EBA DMS največji korak naprej prav zaradi možnosti dostopanja do elektronskega dokumentnega sistema EBA DMS iz vseh naprav kadarkoli in kjerkoli.

8 Zaključek

V magistrskem delu so bili prikazani načini, s katerimi smo elektronski dokumentni sistem EBA DMS integrirali z ostalimi aplikacijami in informacijskimi sistemi v podjetju.

Iz dela je razvidno, da so bile vse integracije izvedene po principu od točke do točke, ki je najnižji nivo integracijske arhitekture. Glavni razlog za izbiro te integracijske arhitekture je predvsem v dejstvu, da so se integracije posameznih aplikacij in informacijskih sistemov z elektronskim dokumentnim sistemom EBA DMS izvajale postopoma, tj. vzporedno s postopno celovito prenovo informacijskega sistema, ki je v podjetju trajala več let. Drugi razlog pa je dejstvo, da v integraciji še vedno ne nastopa toliko sistemov, da bi bilo število integracijskih povezav neobvladljivo.

Prav tako je iz dela razvidno, da je bila večina integracij izvedenih na aplikacijskem nivoju, ki je, kot rečeno v poglavju 2.2.2.2, najboljši način integriranja aplikacij. Na strani sistema EBA DMS nam programska knjižnica EBAXCOM in razvojno okolje EBA Developer namreč omogočata uporabo programskih vmesnikov sistema EBA DMS, ki nam v vsakem trenutku zagotavljajo ohranjanje celovitosti podatkov v elektronskem dokumentnem sistemu EBA DMS. Na nasprotni strani pa je celovitost podatkov v aplikacijah in informacijskih sistemih, ki so integrirani s sistemom EBA DMS, prav tako zagotovljena, in sicer na način, da sistem EBA DMS komunicira z njimi zgolj preko njihovih lastnih spletnih storitev, ki jih ponujajo.

V nekaterih primerih je bila predvsem zaradi hitre in enostavne rešitve uporabljena tudi integracija na podatkovnem nivoju, npr. branje neke vrednosti neposredno iz podatkovne baze integriranega sistema.

Kljub temu da je stanje integracijske arhitekture v podjetju na najnižji stopnji (integracija od točke do točke) pa tega ne moremo reči za nivo poslovno-informacijske arhitekture, saj lahko neimenovano podjetje umestimo na tretji nivo, tj. optimizacija jedra podjetja. Lahko trdimo, da je prav izbira integracije po principu od točke do točke omogočila, da so bile vse integracije sistema EBA DMS z ostalimi sistemi izvedene neodvisno ena od druge in še bolj tesno, kot bi bile sicer.

Integracijo med različnimi sistemi lahko označimo kot uspešno, ko so za končne uporabnike meje med temi sistemi zabrisane ali pa celo neopazne.

Predstavljeni model integracije med elektronskim dokumentnim sistemom EBA DMS in ostalimi sistemi v podjetju lahko na podlagi odzivov uporabnikov označimo kot zelo uspešen. Zavedati se moramo, da brez široke palete načinov za integracijo, ki so na voljo razvijalcu v okviru elektronskega dokumentnega sistema EBA DMS, taka stopnja integracije, kot je bila dosežena v tem primeru, sicer ne bi bila možna.

9 Priloge

9.1 WSDL spletnne storitve EBAINtegration v sistemu Dynamics NAV

```

<definitions targetNamespace="urn:microsoft-dynamics-schemas/codeunit/EBAINtegration"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:tns="urn:microsoft-dynamics-schemas/codeunit/EBAINtegration">
  <types>
    <schema elementFormDefault="qualified"
      targetNamespace="urn:microsoft-dynamics-schemas/codeunit/EBAINtegration"
      xmlns="http://www.w3.org/2001/XMLSchema">
      <element name="DMSChangeLocation">
        <complexType>
          <sequence>
            <element minOccurs="1" maxOccurs="1" name="documentNo" type="int"/>
            <element minOccurs="1" maxOccurs="1" name="locationNo" type="string"/>
          </sequence>
        </complexType>
      </element>
      <element name="DMSChangeLocation_Result">
        <complexType>
          <sequence>
            <element minOccurs="1" maxOccurs="1" name="return_value" type="boolean"/>
          </sequence>
        </complexType>
      </element>
      <element name="DMSChangeStatus">
        <complexType>
          <sequence>
            <element minOccurs="1" maxOccurs="1" name="documentNo" type="int"/>
            <element minOccurs="1" maxOccurs="1" name="status" type="int"/>
            <element minOccurs="1" maxOccurs="1" name="eBAAttachmentDocumentID" type="string"/>
          </sequence>
        </complexType>
      </element>
      <element name="DMSChangeStatus_Result">
        <complexType>
          <sequence>
            <element minOccurs="1" maxOccurs="1" name="return_value" type="boolean"/>
          </sequence>
        </complexType>
      </element>
      <element name="RegisterForEInvoice">
        <complexType>
          <sequence>
            <element minOccurs="1" maxOccurs="1" name="customerNo" type="string"/>
          </sequence>
        </complexType>
      </element>
      <element name="RegisterForEInvoice_Result">
        <complexType>
          <sequence>
            <element minOccurs="1" maxOccurs="1" name="return_value" type="boolean"/>
          </sequence>
        </complexType>
      </element>
      <element name="UnregisterForEInvoice">
        <complexType>
          <sequence>
            <element minOccurs="1" maxOccurs="1" name="customerNo" type="string"/>
          </sequence>
        </complexType>
      </element>
      <element name="UnregisterForEInvoice_Result">
        <complexType>
          <sequence>
            <element minOccurs="1" maxOccurs="1" name="return_value" type="boolean"/>
          </sequence>
        </complexType>
      </element>
      <element name="RecreateEInvoice">
        <complexType>
          <sequence>
            <element minOccurs="1" maxOccurs="1" name="entryNo" type="int"/>
          </sequence>
        </complexType>
      </element>
      <element name="RecreateEInvoice_Result">

```

```

    <complexType>
      <sequence/>
    </complexType>
  </element>
</schema>
</types>
<message name="DMSChangeLocation">
  <part name="parameters" element="tns:DMSChangeLocation"/>
</message>
<message name="DMSChangeLocation_Result">
  <part name="parameters" element="tns:DMSChangeLocation_Result"/>
</message>
<message name="DMSChangeStatus">
  <part name="parameters" element="tns:DMSChangeStatus"/>
</message>
<message name="DMSChangeStatus_Result">
  <part name="parameters" element="tns:DMSChangeStatus_Result"/>
</message>
<message name="RegisterForEInvoice">
  <part name="parameters" element="tns:RegisterForEInvoice"/>
</message>
<message name="RegisterForEInvoice_Result">
  <part name="parameters" element="tns:RegisterForEInvoice_Result"/>
</message>
<message name="UnregisterForEInvoice">
  <part name="parameters" element="tns:UnregisterForEInvoice"/>
</message>
<message name="UnregisterForEInvoice_Result">
  <part name="parameters" element="tns:UnregisterForEInvoice_Result"/>
</message>
<message name="RecreateEInvoice">
  <part name="parameters" element="tns:RecreateEInvoice"/>
</message>
<message name="RecreateEInvoice_Result">
  <part name="parameters" element="tns:RecreateEInvoice_Result"/>
</message>
<portType name="EBAIntegration_Port">
  <operation name="DMSChangeLocation">
    <input name="DMSChangeLocation" message="tns:DMSChangeLocation"/>
    <output name="DMSChangeLocation_Result" message="tns:DMSChangeLocation_Result"/>
  </operation>
  <operation name="DMSChangeStatus">
    <input name="DMSChangeStatus" message="tns:DMSChangeStatus"/>
    <output name="DMSChangeStatus_Result" message="tns:DMSChangeStatus_Result"/>
  </operation>
  <operation name="RegisterForEInvoice">
    <input name="RegisterForEInvoice" message="tns:RegisterForEInvoice"/>
    <output name="RegisterForEInvoice_Result" message="tns:RegisterForEInvoice_Result"/>
  </operation>
  <operation name="UnregisterForEInvoice">
    <input name="UnregisterForEInvoice" message="tns:UnregisterForEInvoice"/>
    <output name="UnregisterForEInvoice_Result" message="tns:UnregisterForEInvoice_Result"/>
  </operation>
  <operation name="RecreateEInvoice">
    <input name="RecreateEInvoice" message="tns:RecreateEInvoice"/>
    <output name="RecreateEInvoice_Result" message="tns:RecreateEInvoice_Result"/>
  </operation>
</portType>
<binding name="EBAIntegration_Binding" type="tns:EBAIntegration_Port">
  <binding transport="http://schemas.xmlsoap.org/soap/http" xmlns="http://schemas.xmlsoap.org/wsdl/soap/">
  <operation name="DMSChangeLocation">
    <operation soapAction="urn:microsoft-dynamics-schemas/codeunit/EBAIntegration:DMSChangeLocation"
      style="document" xmlns="http://schemas.xmlsoap.org/wsdl/soap/">
    <input name="DMSChangeLocation">
      <body use="literal" xmlns="http://schemas.xmlsoap.org/wsdl/soap/">
    </input>
    <output name="DMSChangeLocation_Result">
      <body use="literal" xmlns="http://schemas.xmlsoap.org/wsdl/soap/">
    </output>
  </operation>
  <operation name="DMSChangeStatus">
    <operation soapAction="urn:microsoft-dynamics-schemas/codeunit/EBAIntegration:DMSChangeStatus"
      style="document" xmlns="http://schemas.xmlsoap.org/wsdl/soap/">
    <input name="DMSChangeStatus">
      <body use="literal" xmlns="http://schemas.xmlsoap.org/wsdl/soap/">
    </input>
    <output name="DMSChangeStatus_Result">
      <body use="literal" xmlns="http://schemas.xmlsoap.org/wsdl/soap/">
    </output>
  </operation>
  <operation name="RegisterForEInvoice">
    <operation soapAction="urn:microsoft-dynamics-schemas/codeunit/EBAIntegration:RegisterForEInvoice"
      style="document" xmlns="http://schemas.xmlsoap.org/wsdl/soap/">
    <input name="RegisterForEInvoice">
      <body use="literal" xmlns="http://schemas.xmlsoap.org/wsdl/soap/">
    </input>
  </operation>

```

```

<output name="RegisterForEInvoice_Result">
  <body use="literal" xmlns="http://schemas.xmlsoap.org/wsdl/soap/" />
</output>
</operation>
<operation name="UnregisterForEInvoice">
  <operation soapAction="urn:microsoft-dynamics-schemas/codeunit/EBAIntegration:UnregisterForEInvoice"
    style="document" xmlns="http://schemas.xmlsoap.org/wsdl/soap/" />
  <input name="UnregisterForEInvoice">
    <body use="literal" xmlns="http://schemas.xmlsoap.org/wsdl/soap/" />
  </input>
  <output name="UnregisterForEInvoice_Result">
    <body use="literal" xmlns="http://schemas.xmlsoap.org/wsdl/soap/" />
  </output>
</operation>
<operation name="RecreateEInvoice">
  <operation soapAction="urn:microsoft-dynamics-schemas/codeunit/EBAIntegration:RecreateEInvoice"
    style="document" xmlns="http://schemas.xmlsoap.org/wsdl/soap/" />
  <input name="RecreateEInvoice">
    <body use="literal" xmlns="http://schemas.xmlsoap.org/wsdl/soap/" />
  </input>
  <output name="RecreateEInvoice_Result">
    <body use="literal" xmlns="http://schemas.xmlsoap.org/wsdl/soap/" />
  </output>
</operation>
</binding>
<service name="EBAIntegration">
  <port name="EBAIntegration_Port" binding="tns:EBAIntegration_Binding">
    <address location="http://srvbs02.ad.si:7047/DynamicsNAV71/WS/PROD/Codeunit/EBAIntegration?wsdl"
      xmlns="http://schemas.xmlsoap.org/wsdl/soap/" />
  </port>
</service>
</definitions>

```

9.2 WSDL spletna storitve GeneralDocuments v sistemu EBA DMS

```

<?xml version="1.0" encoding="UTF-8" ?>
<wsdl:definitions name="GeneralDocuments"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  targetNamespace="http://www.eba.si/wsdl/GeneralDocuments.wsdl"
  xmlns:tns="http://www.eba.si/wsdl/GeneralDocuments.wsdl"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <wsdl:types>
    <xsd:schema elementFormDefault="unqualified"
      attributeFormDefault="unqualified"
      targetNamespace="http://www.eba.si/wsdl/GeneralDocuments.wsdl">
      <xsd:complexType name="GetEbaDocIdFromEAD_Response">
        <xsd:sequence>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:int" name="status"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="ebadocid"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="error"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="ebadoctype"/>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="docdata">
        <xsd:sequence>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="userid"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="subject"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="doctype"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:int" name="direction"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:int" name="senderexternid"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:int" name="receiverexternid"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:int" name="classificationId"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="classificationNode"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:base64Binary" name="content"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="oldDocId"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="linkID"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="caseId"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="sendInternalTo"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="addAccessTo"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="barCode"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string"
            name="zco_ENTITY_ID_LIFECYCLE"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="zco_type"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:int" name="sendToArchive"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="backgroundName"/>
          <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="podpisniki"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:schema>
  </wsdl:types>

```

```

</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="InsertDocumentGeneralResponse">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="1" type="xsd:int" name="status"/>
    <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="ebadocid"/>
    <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="ead"/>
    <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="error"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>
</wsdl:types>
<wsdl:message name="getEbaDocIdFromEADRequest">
  <wsdl:part type="xsd:string" name="userid"/>
  <wsdl:part type="xsd:string" name="password"/>
  <wsdl:part type="xsd:string" name="ead"/>
</wsdl:message>
<wsdl:message name="getEbaDocIdFromEADResponse">
  <wsdl:part type="tns:GetEbaDocIdFromEAD_Response" name="response"/>
</wsdl:message>
<wsdl:message name="insertDocumentGeneralRequest">
  <wsdl:part type="xsd:string" name="userid"/>
  <wsdl:part type="xsd:string" name="password"/>
  <wsdl:part type="tns:docdata" name="doc"/>
</wsdl:message>
<wsdl:message name="insertDocumentGeneralResponse">
  <wsdl:part type="tns:InsertDocumentGeneralResponse" name="response"/>
</wsdl:message>
<wsdl:portType name="GeneralDocumentsPortType">
  <wsdl:operation name="getEbaDocIdFromEAD">
    <wsdl:input message="tns:getEbaDocIdFromEADRequest"/>
    <wsdl:output message="tns:getEbaDocIdFromEADResponse"/>
  </wsdl:operation>
  <wsdl:operation name="insertDocumentGeneral">
    <wsdl:input message="tns:insertDocumentGeneralRequest"/>
    <wsdl:output message="tns:insertDocumentGeneralResponse"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding type="tns:GeneralDocumentsPortType" name="GeneralDocuments">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/>
  <wsdl:operation name="getEbaDocIdFromEAD">
    <soap:operation soapAction="getEbaDocIdFromEAD"/>
    <wsdl:input>
      <soap:body use="encoded" namespace="urn:eba:ws"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="encoded" namespace="urn:eba:ws"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="insertDocumentGeneral">
    <soap:operation soapAction="insertDocumentGeneral"/>
    <wsdl:input>
      <soap:body use="encoded" namespace="urn:eba:ws"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="encoded" namespace="urn:eba:ws"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="GeneralDocuments">
  <wsdl:port binding="tns:GeneralDocuments" name="GeneralDocuments">
    <soap:address location="https://eba01:9878/general"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

9.3 WSDL spletne storitve za masovno pripravo dokumentov

```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions name="WSObdelavaDokumentov" targetNamespace="http://tempuri.org/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
  xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
  xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"
  xmlns:microsoft="http://schemas.microsoft.com/ws/2005/12/wsdl/contract"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"

```

```

xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="http://tempuri.org/"
xmlns:wsa10="http://www.w3.org/2005/08/addressing"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
<wsdl:types>
  <xs:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:import namespace="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
    <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
    <xs:element name="DodajInObdelajPaket">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="Glava" nillable="true" type="q1:dcGlava"
            xmlns:q1="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
          <xs:element minOccurs="0" name="Postavke" nillable="true"
            type="q2:ArrayOfdcPostavka"
            xmlns:q2="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="DodajInObdelajPaketResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="DodajInObdelajPaketResult" nillable="true"
            type="q3:dcResponse"
            xmlns:q3="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="DodajPaket">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="Glava" nillable="true" type="q4:dcGlava"
            xmlns:q4="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
          <xs:element minOccurs="0" name="Postavke" nillable="true"
            type="q5:ArrayOfdcPostavka"
            xmlns:q5="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="DodajPaketResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="DodajPaketResult" nillable="true"
            type="q6:dcResponse"
            xmlns:q6="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="DodajVPaket">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="PaketTip" type="xs:int"/>
          <xs:element minOccurs="0" name="PaketID" type="xs:int"/>
          <xs:element minOccurs="0" name="Postavke" nillable="true"
            type="q7:ArrayOfdcPostavka"
            xmlns:q7="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="DodajVPaketResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="DodajVPaketResult" nillable="true"
            type="q8:dcResponse"
            xmlns:q8="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="ObdelajPaket">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="PaketTip" type="xs:int"/>
          <xs:element minOccurs="0" name="PaketID" type="xs:int"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="ObdelajPaketResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="ObdelajPaketResult" nillable="true"
            type="q9:dcResponse"
            xmlns:q9="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</wsdl:types>

```

```

        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="BrisiPaket">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="PaketTip" type="xs:int"/>
          <xs:element minOccurs="0" name="PaketID" type="xs:int"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="BrisiPaketResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="BrisiPaketResult" nillable="true"
            type="q10:dcResponse"
            xmlns:q10="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="OdpriPaket">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="PaketTip" type="xs:int"/>
          <xs:element minOccurs="0" name="PaketID" type="xs:int"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="OdpriPaketResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="OdpriPaketResult" nillable="true"
            type="q11:dcResponse"
            xmlns:q11="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="ZapriPaket">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="PaketTip" type="xs:int"/>
          <xs:element minOccurs="0" name="PaketID" type="xs:int"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="ZapriPaketResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="ZapriPaketResult" nillable="true"
            type="q12:dcResponse"
            xmlns:q12="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="PreberiPaket">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="PaketTip" type="xs:int"/>
          <xs:element minOccurs="0" name="PaketID" type="xs:int"/>
          <xs:element minOccurs="0" name="VrniPodrobnostiPostavk" type="xs:boolean"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="PreberiPaketResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="PreberiPaketResult" nillable="true"
            type="q13:dcResponse"
            xmlns:q13="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
          <xs:element minOccurs="0" name="StatusPaketa" nillable="true"
            type="q14:dcStatusPaketa"
            xmlns:q14="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="PreberiPostavko">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="PaketTip" type="xs:int"/>
          <xs:element minOccurs="0" name="PaketID" type="xs:int"/>
          <xs:element minOccurs="0" name="PostavkaID" type="xs:int"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="PreberiPostavkoResponse">
      <xs:complexType>

```



```

    <xs:sequence>
      <xs:element minOccurs="0" name="PreberiPostavkoResult" nillable="true"
        type="q15:dcResponse"
        xmlns:q15="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
      <xs:element minOccurs="0" name="Postavka" type="q16:sPostavka"
        xmlns:q16="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="IzdelajPdfZaPostavko">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="PostavkaUID" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="IzdelajPdfZaPostavkoResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="IzdelajPdfZaPostavkoResult" nillable="true"
        type="q17:dcResponse"
        xmlns:q17="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
      <xs:element minOccurs="0" name="Postavka" type="q18:sPostavka"
        xmlns:q18="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="PripraviPaketZaTiskPoMeri">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="SeznamPostavkaUID" nillable="true"
        type="q19:ArrayOfint"
        xmlns:q19="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
      <xs:element minOccurs="0" name="SteviloDokumentovNaPaket" type="xs:int"/>
      <xs:element minOccurs="0" name="PotDoDatotekeZaTisk" nillable="true"
        type="xs:string"/>
      <xs:element minOccurs="0" name="PotDoDatotekeZaTiskSistem" nillable="true"
        type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="PripraviPaketZaTiskPoMeriResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="PripraviPaketZaTiskPoMeriResult" nillable="true"
        type="q20:dcResponse"
        xmlns:q20="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
      <xs:element minOccurs="0" name="PotDoDatotekeZaTisk" nillable="true"
        type="xs:string"/>
      <xs:element minOccurs="0" name="PotDoDatotekeZaTiskSistem" nillable="true"
        type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="TestSendMail">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="PaketTip" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="TestSendMailResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="TestSendMailResult" nillable="true"
        type="q21:dcResponse"
        xmlns:q21="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
<xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
  targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/">
  <xs:element name="anyType" nillable="true" type="xs:anyType"/>
  <xs:element name="anyURI" nillable="true" type="xs:anyURI"/>
  <xs:element name="base64Binary" nillable="true" type="xs:base64Binary"/>
  <xs:element name="boolean" nillable="true" type="xs:boolean"/>
  <xs:element name="byte" nillable="true" type="xs:byte"/>
  <xs:element name="dateTime" nillable="true" type="xs:dateTime"/>
  <xs:element name="decimal" nillable="true" type="xs:decimal"/>
  <xs:element name="double" nillable="true" type="xs:double"/>
  <xs:element name="float" nillable="true" type="xs:float"/>
  <xs:element name="int" nillable="true" type="xs:int"/>
  <xs:element name="long" nillable="true" type="xs:long"/>

```

```

<xs:element name="QName" nillable="true" type="xs:QName"/>
<xs:element name="short" nillable="true" type="xs:short"/>
<xs:element name="string" nillable="true" type="xs:string"/>
<xs:element name="unsignedByte" nillable="true" type="xs:unsignedByte"/>
<xs:element name="unsignedInt" nillable="true" type="xs:unsignedInt"/>
<xs:element name="unsignedLong" nillable="true" type="xs:unsignedLong"/>
<xs:element name="unsignedShort" nillable="true" type="xs:unsignedShort"/>
<xs:element name="char" nillable="true" type="tns:char"/>
<xs:simpleType name="char">
  <xs:restriction base="xs:int"/>
</xs:simpleType>
<xs:element name="duration" nillable="true" type="tns:duration"/>
<xs:simpleType name="duration">
  <xs:restriction base="xs:duration">
    <xs:pattern value="\-?P(\d*D)?(T(\d*H)?(\d*M)?(\d*(\.\d*)?S)?)?"/>
    <xs:minInclusive value="-P10675199DT2H48M5.4775808S"/>
    <xs:maxInclusive value="P10675199DT2H48M5.4775807S"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="guid" nillable="true" type="tns:guid"/>
<xs:simpleType name="guid">
  <xs:restriction base="xs:string">
    <xs:pattern value="[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}"/>
  </xs:restriction>
</xs:simpleType>
<xs:attribute name="FactoryType" type="xs:QName"/>
<xs:attribute name="Id" type="xs:ID"/>
<xs:attribute name="Ref" type="xs:IDREF"/>
</xs:schema>
<xs:schema elementFormDefault="qualified"
  targetNamespace="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://schemas.datacontract.org/2004/07/WSObdelavaDokumentov">
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization"/>
  <xs:complexType name="dcGlava">
    <xs:sequence>
      <xs:element minOccurs="0" name="IzdelajEBA" type="xs:int"/>
      <xs:element minOccurs="0" name="IzdelajPDF" type="xs:int"/>
      <xs:element minOccurs="0" name="IzdelajPaketeZaTisk" type="xs:int"/>
      <xs:element minOccurs="0" name="PaketTip" type="xs:int"/>
      <xs:element minOccurs="0" name="PaketZakljucen" type="xs:int"/>
      <xs:element minOccurs="0" name="PaketZunanjiID" type="xs:int"/>
      <xs:element minOccurs="0" name="Param1" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="Param2" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="Param3" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="Param4" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="PotDoPaketovZaTisk" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="dcGlava" nillable="true" type="tns:dcGlava"/>
  <xs:complexType name="ArrayOfdcPostavka">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="dcPostavka" nillable="true"
        type="tns:dcPostavka"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfdcPostavka" nillable="true" type="tns:ArrayOfdcPostavka"/>
  <xs:complexType name="dcPostavka">
    <xs:sequence>
      <xs:element minOccurs="0" name="PostavkaID" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="dcPostavka" nillable="true" type="tns:dcPostavka"/>
  <xs:complexType name="dcResponse">
    <xs:sequence>
      <xs:element minOccurs="0" name="MSG" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="RTC" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="dcResponse" nillable="true" type="tns:dcResponse"/>
  <xs:complexType name="dcStatusPaketa">
    <xs:sequence>
      <xs:element minOccurs="0" name="Paket_NapakaOpis" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="Paket_NapakaVObdelavi" type="xs:boolean"/>
      <xs:element minOccurs="0" name="Paket_Obdelan" type="xs:boolean"/>
      <xs:element minOccurs="0" name="Postavke_StNapak" type="xs:int"/>
      <xs:element minOccurs="0" name="Postavke_StObdelanihDOCX" type="xs:int"/>
      <xs:element minOccurs="0" name="Postavke_StObdelanihPDF" type="xs:int"/>
      <xs:element minOccurs="0" name="Postavke_StObdelanihVCeloti" type="xs:int"/>
      <xs:element minOccurs="0" name="Postavke_StVseh" type="xs:int"/>
      <xs:element minOccurs="0" name="postavke" nillable="true" type="tns:ArrayOfsPostavka"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="dcStatusPaketa" nillable="true" type="tns:dcStatusPaketa"/>
  <xs:complexType name="ArrayOfsPostavka">

```

```

<xs:sequence>
  <xs:element minOccurs="0" maxOccurs="unbounded" name="sPostavka" type="tns:sPostavka"/>
</xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfPostavka" nillable="true" type="tns:ArrayOfPostavka"/>
<xs:complexType name="sPostavka">
  <xs:annotation>
    <xs:appinfo>
      <IsValueType
        xmlns="http://schemas.microsoft.com/2003/10/Serialization/">true</IsValueType>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element minOccurs="0" name="ODP_DATUMOBDELAVE" type="xs:dateTime"/>
    <xs:element minOccurs="0" name="ODP_DATUMVPISA" type="xs:dateTime"/>
    <xs:element minOccurs="0" name="ODP_NAPAKA" type="xs:boolean"/>
    <xs:element minOccurs="0" name="ODP_NAPAKA_OPIS" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="ODP_OBDELANO" type="xs:boolean"/>
    <xs:element minOccurs="0" name="ODP_ODG_UID" type="xs:int"/>
    <xs:element minOccurs="0" name="ODP_POSTAVKA_ID" type="xs:int"/>
    <xs:element minOccurs="0" name="ODP_REF_DOCX" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="ODP_REF_EBA" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="ODP_REF_PDF" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="ODP_SYSDATE" type="xs:dateTime"/>
    <xs:element minOccurs="0" name="ODP_SYSUSER" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="ODP_UID" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="sPostavka" nillable="true" type="tns:sPostavka"/>
</xs:schema>
<xs:schema elementFormDefault="qualified"
  targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
  <xs:complexType name="ArrayOfint">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="int" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfint" nillable="true" type="tns:ArrayOfint"/>
</xs:schema>
</wsdl:types>
<wsdl:message name="IWSObdelavaDokumentov_DodajInObdelajPaket_InputMessage">
  <wsdl:part name="parameters" element="tns:DodajInObdelajPaket"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_DodajInObdelajPaket_OutputMessage">
  <wsdl:part name="parameters" element="tns:DodajInObdelajPaketResponse"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_DodajPaket_InputMessage">
  <wsdl:part name="parameters" element="tns:DodajPaket"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_DodajPaket_OutputMessage">
  <wsdl:part name="parameters" element="tns:DodajPaketResponse"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_DodajVPaket_InputMessage">
  <wsdl:part name="parameters" element="tns:DodajVPaket"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_DodajVPaket_OutputMessage">
  <wsdl:part name="parameters" element="tns:DodajVPaketResponse"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_ObdelajPaket_InputMessage">
  <wsdl:part name="parameters" element="tns:ObdelajPaket"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_ObdelajPaket_OutputMessage">
  <wsdl:part name="parameters" element="tns:ObdelajPaketResponse"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_BrisiPaket_InputMessage">
  <wsdl:part name="parameters" element="tns:BrisiPaket"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_BrisiPaket_OutputMessage">
  <wsdl:part name="parameters" element="tns:BrisiPaketResponse"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_OdpriPaket_InputMessage">
  <wsdl:part name="parameters" element="tns:OdpriPaket"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_OdpriPaket_OutputMessage">
  <wsdl:part name="parameters" element="tns:OdpriPaketResponse"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_ZapriPaket_InputMessage">
  <wsdl:part name="parameters" element="tns:ZapriPaket"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_ZapriPaket_OutputMessage">
  <wsdl:part name="parameters" element="tns:ZapriPaketResponse"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_PreberiPaket_InputMessage">
  <wsdl:part name="parameters" element="tns:PreberiPaket"/>

```

```

</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_PreberiPaket_OutputMessage">
  <wsdl:part name="parameters" element="tns:PreberiPaketResponse"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_PreberiPostavko_InputMessage">
  <wsdl:part name="parameters" element="tns:PreberiPostavko"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_PreberiPostavko_OutputMessage">
  <wsdl:part name="parameters" element="tns:PreberiPostavkoResponse"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_IzdelajPdfZaPostavko_InputMessage">
  <wsdl:part name="parameters" element="tns:IzdelajPdfZaPostavko"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_IzdelajPdfZaPostavko_OutputMessage">
  <wsdl:part name="parameters" element="tns:IzdelajPdfZaPostavkoResponse"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_PripraviPaketZaTiskPoMeri_InputMessage">
  <wsdl:part name="parameters" element="tns:PripraviPaketZaTiskPoMeri"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_PripraviPaketZaTiskPoMeri_OutputMessage">
  <wsdl:part name="parameters" element="tns:PripraviPaketZaTiskPoMeriResponse"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_TestSendMail_InputMessage">
  <wsdl:part name="parameters" element="tns:TestSendMail"/>
</wsdl:message>
<wsdl:message name="IWSObdelavaDokumentov_TestSendMail_OutputMessage">
  <wsdl:part name="parameters" element="tns:TestSendMailResponse"/>
</wsdl:message>
<wsdl:portType name="IWSObdelavaDokumentov">
  <wsdl:operation name="DodajInObdelajPaket">
    <wsdl:input wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/DodajInObdelajPaket"
      message="tns:IWSObdelavaDokumentov_DodajInObdelajPaket_InputMessage"/>
    <wsdl:output wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/DodajInObdelajPaketResponse"
      message="tns:IWSObdelavaDokumentov_DodajInObdelajPaket_OutputMessage"/>
  </wsdl:operation>
  <wsdl:operation name="DodajPaket">
    <wsdl:input wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/DodajPaket"
      message="tns:IWSObdelavaDokumentov_DodajPaket_InputMessage"/>
    <wsdl:output wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/DodajPaketResponse"
      message="tns:IWSObdelavaDokumentov_DodajPaket_OutputMessage"/>
  </wsdl:operation>
  <wsdl:operation name="DodajVPaket">
    <wsdl:input wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/DodajVPaket"
      message="tns:IWSObdelavaDokumentov_DodajVPaket_InputMessage"/>
    <wsdl:output wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/DodajVPaketResponse"
      message="tns:IWSObdelavaDokumentov_DodajVPaket_OutputMessage"/>
  </wsdl:operation>
  <wsdl:operation name="ObdelajPaket">
    <wsdl:input wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/ObdelajPaket"
      message="tns:IWSObdelavaDokumentov_ObdelajPaket_InputMessage"/>
    <wsdl:output wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/ObdelajPaketResponse"
      message="tns:IWSObdelavaDokumentov_ObdelajPaket_OutputMessage"/>
  </wsdl:operation>
  <wsdl:operation name="BrisiPaket">
    <wsdl:input wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/BrisiPaket"
      message="tns:IWSObdelavaDokumentov_BrisiPaket_InputMessage"/>
    <wsdl:output wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/BrisiPaketResponse"
      message="tns:IWSObdelavaDokumentov_BrisiPaket_OutputMessage"/>
  </wsdl:operation>
  <wsdl:operation name="OdpriPaket">
    <wsdl:input wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/OdpriPaket"
      message="tns:IWSObdelavaDokumentov_OdpriPaket_InputMessage"/>
    <wsdl:output wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/OdpriPaketResponse"
      message="tns:IWSObdelavaDokumentov_OdpriPaket_OutputMessage"/>
  </wsdl:operation>
  <wsdl:operation name="ZapriPaket">
    <wsdl:input wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/ZapriPaket"
      message="tns:IWSObdelavaDokumentov_ZapriPaket_InputMessage"/>
    <wsdl:output wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/ZapriPaketResponse"
      message="tns:IWSObdelavaDokumentov_ZapriPaket_OutputMessage"/>
  </wsdl:operation>
  <wsdl:operation name="PreberiPaket">
    <wsdl:input wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/PreberiPaket"
      message="tns:IWSObdelavaDokumentov_PreberiPaket_InputMessage"/>
    <wsdl:output wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/PreberiPaketResponse"
      message="tns:IWSObdelavaDokumentov_PreberiPaket_OutputMessage"/>
  </wsdl:operation>
  <wsdl:operation name="PreberiPostavko">
    <wsdl:input wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/PreberiPostavko"
      message="tns:IWSObdelavaDokumentov_PreberiPostavko_InputMessage"/>
    <wsdl:output wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/PreberiPostavkoResponse"
      message="tns:IWSObdelavaDokumentov_PreberiPostavko_OutputMessage"/>
  </wsdl:operation>
  <wsdl:operation name="IzdelajPdfZaPostavko">
    <wsdl:input wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/IzdelajPdfZaPostavko"
      message="tns:IWSObdelavaDokumentov_IzdelajPdfZaPostavko_InputMessage"/>
  </wsdl:operation>

```

```

    <wsdl:output
      wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/IzdelajPdfZaPostavkoResponse"
      message="tns:IWSObdelavaDokumentov_IzdelajPdfZaPostavko_OutputMessage"/>
  </wsdl:operation>
  <wsdl:operation name="PripraviPaketZaTiskPoMeri">
    <wsdl:input wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/PripraviPaketZaTiskPoMeri"
      message="tns:IWSObdelavaDokumentov_PripraviPaketZaTiskPoMeri_InputMessage"/>
    <wsdl:output
      wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/PripraviPaketZaTiskPoMeriResponse"
      message="tns:IWSObdelavaDokumentov_PripraviPaketZaTiskPoMeri_OutputMessage"/>
  </wsdl:operation>
  <wsdl:operation name="TestSendMail">
    <wsdl:input wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/TestSendMail"
      message="tns:IWSObdelavaDokumentov_TestSendMail_InputMessage"/>
    <wsdl:output wsaw:Action="http://tempuri.org/IWSObdelavaDokumentov/TestSendMailResponse"
      message="tns:IWSObdelavaDokumentov_TestSendMail_OutputMessage"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="BasicHttpBinding_IWSObdelavaDokumentov" type="tns:IWSObdelavaDokumentov">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="DodajInObdelajPaket">
    <soap:operation soapAction="http://tempuri.org/IWSObdelavaDokumentov/DodajInObdelajPaket"
      style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="DodajPaket">
    <soap:operation soapAction="http://tempuri.org/IWSObdelavaDokumentov/DodajPaket"
      style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="DodajVPaket">
    <soap:operation soapAction="http://tempuri.org/IWSObdelavaDokumentov/DodajVPaket"
      style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ObdelajPaket">
    <soap:operation soapAction="http://tempuri.org/IWSObdelavaDokumentov/ObdelajPaket"
      style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="BrisiPaket">
    <soap:operation soapAction="http://tempuri.org/IWSObdelavaDokumentov/BrisiPaket"
      style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="OdpriPaket">
    <soap:operation soapAction="http://tempuri.org/IWSObdelavaDokumentov/OdpriPaket"
      style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ZapriPaket">
    <soap:operation soapAction="http://tempuri.org/IWSObdelavaDokumentov/ZapriPaket"
      style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
  </wsdl:operation>

```

```

        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="PreberiPaket">
        <soap:operation soapAction="http://tempuri.org/IWSObdelavaDokumentov/PreberiPaket"
            style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="PreberiPostavko">
        <soap:operation soapAction="http://tempuri.org/IWSObdelavaDokumentov/PreberiPostavko"
            style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="IzdelajPdfZaPostavko">
        <soap:operation soapAction="http://tempuri.org/IWSObdelavaDokumentov/IzdelajPdfZaPostavko"
            style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="PripraviPaketZaTiskPoMeri">
        <soap:operation soapAction="http://tempuri.org/IWSObdelavaDokumentov/PripraviPaketZaTiskPoMeri"
            style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="TestSendMail">
        <soap:operation soapAction="http://tempuri.org/IWSObdelavaDokumentov/TestSendMail"
            style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="WSObdelavaDokumentov">
    <wsdl:port name="BasicHttpBinding_IWSObdelavaDokumentov"
        binding="tns:BasicHttpBinding_IWSObdelavaDokumentov">
        <soap:address location="http://srvsuzenj:8080/WSObdelavaDokumentov.svc"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

9.4 WSDL spletne storitve Križanka v sistemu EBA DMS

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="Krizanka"
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    targetNamespace="http://www.eba.si/wsdl/Krizanka.wsdl"
    xmlns:tns="http://www.eba.si/wsdl/Krizanka.wsdl"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
    <wsdl:types>
        <xsd:schema
            elementFormDefault="unqualified"
            attributeFormDefault="unqualified" targetNamespace="http://www.eba.si/wsdl/Krizanka.wsdl">
            <xsd:complexType name="document">
                <xsd:sequence>

```

```

<xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="externid"/>
<xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="docsubtype"/>
<xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="subject"/>
<xsd:element minOccurs="0" maxOccurs="1" type="xsd:int" name="status"/>
<xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="userid"/>
<xsd:element minOccurs="0" maxOccurs="1" type="xsd:base64Binary" name="content"/>
<xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="olddocid"/>
<xsd:element minOccurs="0" maxOccurs="1" type="tns:ArrayOfString" name="signers"/>
<xsd:element minOccurs="0" maxOccurs="1" type="xsd:int" name="actgyear"/>
<xsd:element minOccurs="0" maxOccurs="1" type="xsd:int" name="actgmonth"/>
<xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="docfor"/>
<xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="linkID"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfString">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute wsdl:arrayType="xsd:string[]" ref="SOAP-ENC:arrayType"/>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="InsertDocumentResponse">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="1" type="xsd:int" name="status"/>
    <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="ebadocid"/>
    <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="error"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="AmIHereResponse">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="1" type="xsd:int" name="status"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ShowRelayResponse">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="1" type="xsd:int" name="status"/>
    <xsd:element minOccurs="0" maxOccurs="1" type="xsd:string" name="error"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DestroyDocumentResponse">
  <xsd:sequence/>
</xsd:complexType>
</xsd:schema>
</wsdl:types>
<wsdl:message name="insertDocumentAsNewVersionOfRequest">
  <wsdl:part type="xsd:string" name="userid"/>
  <wsdl:part type="xsd:string" name="password"/>
  <wsdl:part type="tns:document" name="doc"/>
</wsdl:message>
<wsdl:message name="insertDocumentAsNewVersionOfResponse">
  <wsdl:part type="tns:InsertDocumentResponse" name="response"/>
</wsdl:message>
<wsdl:message name="insertDocumentRequest">
  <wsdl:part type="xsd:string" name="userid"/>
  <wsdl:part type="xsd:string" name="password"/>
  <wsdl:part type="tns:document" name="doc"/>
</wsdl:message>
<wsdl:message name="insertDocumentResponse">
  <wsdl:part type="tns:InsertDocumentResponse" name="response"/>
</wsdl:message>
<wsdl:message name="AmIHereRequest">
  <wsdl:part type="xsd:string" name="userid"/>
  <wsdl:part type="xsd:string" name="password"/>
</wsdl:message>
<wsdl:message name="AmIHereResponse">
  <wsdl:part type="tns:AmIHereResponse" name="response"/>
</wsdl:message>
<wsdl:message name="showRelayRequest">
  <wsdl:part type="xsd:string" name="userid"/>
  <wsdl:part type="xsd:string" name="password"/>
  <wsdl:part type="xsd:string" name="ebadocid"/>
  <wsdl:part type="xsd:string" name="requestip"/>
  <wsdl:part type="xsd:string" name="ebauserid"/>
  <wsdl:part type="xsd:string" name="ebauserpwd"/>
</wsdl:message>
<wsdl:message name="showRelayResponse">
  <wsdl:part type="tns:ShowRelayResponse" name="response"/>
</wsdl:message>
<wsdl:message name="destroyDocumentRequest">
  <wsdl:part type="xsd:string" name="userid"/>
  <wsdl:part type="xsd:string" name="password"/>
  <wsdl:part type="xsd:string" name="ebadocid"/>
</wsdl:message>
<wsdl:message name="destroyDocumentResponse">
  <wsdl:part type="tns:DestroyDocumentResponse" name="response"/>
</wsdl:message>

```

```

<wsdl:portType name="KrizankaPortType">
  <wsdl:operation name="insertDocumentAsNewVersionOf">
    <wsdl:input message="tns:insertDocumentAsNewVersionOfRequest"/>
    <wsdl:output message="tns:insertDocumentAsNewVersionOfResponse"/>
  </wsdl:operation>
  <wsdl:operation name="insertDocument">
    <wsdl:input message="tns:insertDocumentRequest"/>
    <wsdl:output message="tns:insertDocumentResponse"/>
  </wsdl:operation>
  <wsdl:operation name="AmIHere">
    <wsdl:input message="tns:AmIHereRequest"/>
    <wsdl:output message="tns:AmIHereResponse"/>
  </wsdl:operation>
  <wsdl:operation name="showRelay">
    <wsdl:input message="tns:showRelayRequest"/>
    <wsdl:output message="tns:showRelayResponse"/>
  </wsdl:operation>
  <wsdl:operation name="destroyDocument">
    <wsdl:input message="tns:destroyDocumentRequest"/>
    <wsdl:output message="tns:destroyDocumentResponse"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding type="tns:KrizankaPortType" name="Krizanka">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/>
  <wsdl:operation name="insertDocumentAsNewVersionOf">
    <soap:operation soapAction="insertDocumentAsNewVersionOf"/>
    <wsdl:input>
      <soap:body use="encoded" namespace="urn:eba:ws"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="encoded" namespace="urn:eba:ws"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="insertDocument">
    <soap:operation soapAction="insertDocument"/>
    <wsdl:input>
      <soap:body use="encoded" namespace="urn:eba:ws"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="encoded" namespace="urn:eba:ws"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="AmIHere">
    <soap:operation soapAction="AmIHere"/>
    <wsdl:input>
      <soap:body use="encoded" namespace="urn:eba:ws"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="encoded" namespace="urn:eba:ws"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="showRelay">
    <soap:operation soapAction="showRelay"/>
    <wsdl:input>
      <soap:body use="encoded" namespace="urn:eba:ws"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="encoded" namespace="urn:eba:ws"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="destroyDocument">
    <soap:operation soapAction="destroyDocument"/>
    <wsdl:input>
      <soap:body use="encoded" namespace="urn:eba:ws"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="encoded" namespace="urn:eba:ws"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="Krizanka">
  <wsdl:port binding="tns:Krizanka" name="Krizanka">
    <soap:address location="https://eba01:9878/krizanka"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```


9.5 WSDL spletne storitve e-Krizanka spletne aplikacije e-Križanka

```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns="http://www.elektro-gorenjska.si/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  targetNamespace="http://www.elektro-gorenjska.si/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://www.elektro-gorenjska.si/">
      <s:element name="ConfirmDocument">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="request" type="tns:cEBADocumentRequest" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="cEBADocumentRequest">
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="DocType" type="s:int" />
          <s:element minOccurs="1" maxOccurs="1" name="Action" type="s:int" />
          <s:element minOccurs="0" maxOccurs="1" name="EbaID" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="EbaUserID" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="Replacements" type="tns:ArrayOfString" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="ArrayOfString">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:element name="ConfirmDocumentResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="ConfirmDocumentResult"
              type="tns:cEBADocumentResponse" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="cEBADocumentResponse">
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="Status" type="s:int" />
          <s:element minOccurs="0" maxOccurs="1" name="Error" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="NextSigner" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="Archive" type="s:int" />
          <s:element minOccurs="0" maxOccurs="1" name="AddAccess" type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="ConfirmDocumentSoapIn">
    <wsdl:part name="parameters" element="tns:ConfirmDocument" />
  </wsdl:message>
  <wsdl:message name="ConfirmDocumentSoapOut">
    <wsdl:part name="parameters" element="tns:ConfirmDocumentResponse" />
  </wsdl:message>
  <wsdl:portType name="KrizankaEBAsSoap">
    <wsdl:operation name="ConfirmDocument">
      <wsdl:input message="tns:ConfirmDocumentSoapIn" />
      <wsdl:output message="tns:ConfirmDocumentSoapOut" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="KrizankaEBAsSoap" type="tns:KrizankaEBAsSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="ConfirmDocument">
      <soap:operation soapAction="http://www.elektro-gorenjska.si/ConfirmDocument" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="KrizankaEBAsSoap12" type="tns:KrizankaEBAsSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="ConfirmDocument">
      <soap12:operation soapAction="http://www.elektro-gorenjska.si/ConfirmDocument" style="document" />
      <wsdl:input>

```

```
        <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="KrizankaEBAws">
    <wsdl:port name="KrizankaEBAwsSoap" binding="tns:KrizankaEBAwsSoap">
        <soap:address location="http://eweb:90/KrizankaEBAws.asmx" />
    </wsdl:port>
    <wsdl:port name="KrizankaEBAwsSoap12" binding="tns:KrizankaEBAwsSoap12">
        <soap12:address location="http://eweb:90/KrizankaEBAws.asmx" />
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

Literatura in viri

Literatura:

- [1] M. Anderson, "Enterprise Integration in a Manufacturing Environment", Western Kentucky University, 2015.
- [2] M. Bijata, K. Piotrkowski, "Enterprise architecture as a tool to support the strategic management processes in an organization", *Hyperion International Journal of Econophysics & New Economy*, št. 1, zv. 7, str. 177–189, 2014.
- [3] C. A. Binildas, *Service Oriented Java Business Integration*, Birmingham, 2008.
- [4] M. Lankhorst, "Enterprise Architecture at Work, Modelling, Communication and Analysis, Second Edition", Verlag Berlin Heidelberg: Springer, 2009, pogl. 1.
- [5] G. Hohpe, B. Woolf, *Enterprise Integration Patterns - Designing, Building And Deploying Messaging Solutions*, Addison-Wesley Longman, Inc., 2004, pogl. 4.
- [6] D. Litan, M. Velicanu, L. Copcea (Teohari), M. Teohari, A. M. Mocanu (Virgolici), I. Surugiu, O. Raduta, "Business' New Requirement: Information Systems Integration – Methods and Technologies", *International Journal of Computers and Communications*, št. 3, zv. 5, str. 133–145, 2011.
- [7] H. Paulheim, F. Probst, "Application Integration on the User Interface Level: an Ontology-Based Approach", *Data and Knowledge Engineering*, št. 11, zv. 69, str. 1103–1116, 2010.
- [8] A. Rožanec, "The Importance of Enterprise Architecture for Effective Corporate Governance", *Journal of Economic and Business Sciences*, št. 2, zv. 1, 2014.
- [9] R. W. Schulte – The Enterprise Service Bus: Communication Backbone for SOA; <http://www.gartner.com>, 2007.
- [10] T. R. Soomro, A. H. Awan, "Challenges and Future of Enterprise Application Integration", *International Journal of Computer Applications*, št. 7, zv. 42, str. 42–45, 2012.
- [11] R. D. Zota, R. S. Moleavin, L. Ciovică, "Enterprise Service Bus - A Backbone for SOA", *Recent researches in business and economics*, Porto, Portugalska, 2012.

Ostali viri:

- [12] (2016) UJP. Kako je urejeno pošiljanje spremljajočih dokumentov (prilog) k eRačunom? Dostopno na: <https://www.ujp.gov.si/dokumenti/dokument.asp?id=372>
- [13] (2016) Kakšni tipi prilog e-računu so dovoljeni? Je število prilog omejeno? Je omejena velikosti? Dostopno na: <https://www.nlb.si/pogosta-vprasanja-faq-kategorija-podjetja?cat=106>
- [14] (2012) EBA DMS, EBA Product Whitepaper. Dostopno na: http://www.eba.si/dokumenti/EBA_DMS_2012.pdf
- [15] (2002) A. Kang, Enterprise application integration using J2EE. Dostopno na <http://www.javaworld.com/article/2074488/enterprise-java/enterprise-application-integration-using-j2ee.html>
- [16] (2008) Data Integration Architecture: What It Does, Where It's Going, and Why You Should Care. Dostopno na <https://tdwi.org/articles/2008/05/27/data-integration-architecture-what-it-does-where-its-going-and-why-you-should-care.aspx>
- [17] Understanding Enterprise Application Integration - The Benefits of ESB for EAI. Dostopno na <https://www.mulesoft.com/resources/esb/enterprise-application-integration-eai-and-esb>
- [18] (2006) G. Gruman, The Four Stages of Enterprise Architecture. Dostopno na <http://www.cio.com/article/2443291/service-oriented-architecture/the--four-stages-of--enterprise--architecture.html> (pridobljeno 26.5. 2016).
- [19] (2011) A. Harbot, The four stages of IT architecture maturity. Dostopno na <http://www.harbott.com/2011/12/15/the-four-stages-of-it-architecture-maturity/>