

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Roman Thaler

**Načrtovanje upravljanja procesov
v sistemu za obdelavo dokumentov**

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: doc. dr. Mojca Ciglarič

Ljubljana, 2011



Št. naloge: 01732/2011

Datum: 15.03.2011

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

Kandidat: **ROMAN THALER**

Naslov: **NAČRTOVANJE UPRAVLJANJA PROCESOV V SISTEMU ZA
OBDELAVO DOKUMENTOV**
**PROCESS MANAGEMENT DESIGN IN DOCUMENT MANAGEMENT
SYSTEM**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Opišite obstoječe stanje tehnologije na področju obdelave dokumentov ter upravljanja s procesi v podjetju IUS Software. Pojasnite razloge za potrebo po prenovi. Pripravite načrt prenovljenih procesov in sistema za njihovo upravljanje. Sistem naj ima visoko stopnjo interoperabilnosti in arhitekturo SOA ter naj bo zgrajen na ogrodju .NET. Opišite izvedbo in delovanje prototipa, možnosti za razširitve in morebitno nadaljnje delo. Svoje delo kritično ovrednotite.

Mentor:

M. Ciglarič

doc. dr. Mojca Ciglarič



Dekan:

N. Zimic

prof. dr. Nikolaj Zimic

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

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Namesto te strani **vstavite** original izdane teme diplomskega dela s podpisom mentorja in dekana ter žigom fakultete, ki ga diplomant dvigne v študentskem referatu, preden odda izdelek v vezavo!

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani **Roman Thaler**,

z vpisno številko **63040165**,

sem avtor diplomskega dela z naslovom:

Načrtovanje upravljanja procesov v sistemu za obdelavo dokumentov

S svojim podpisom zagotavljam, da:

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

V Ljubljani, dne 06.04.2011

Podpis avtorja:

Zahvala

Zahvaljujem se mentorici, doc. dr. Mojci Ciglarič, ki mi je pomagala in svetovala pri izdelavi diplome. Zahvaljujem se tudi sodelavcem v podjetju IUS SOFTWARE, še posebej vodji razvoja, Romanu Zvonarju, ki mi je s svojim poznavanjem sistemov v podjetju pomagal pri analizi procesov in načrtovanju sistema.

Hvala staršem, ki so me ves čas študija bodrili in nesebično podpirali.

Posebej se zahvaljujem Andreji, ki me je skozi celoten študij spodbujala.

Staršem

Kazalo

Povzetek	1
Abstract	2
1 Uvod	4
1.1 Sistem IUS CONTENT	6
1.1.1 IUS TIME 3.0	6
1.1.2 IUS TIME 4.0	8
1.2 Opis obstoječih procesov za obdelavo dokumentov sodne prakse	9
2 Popis, prenova in modeliranje procesov	12
2.1 Popis obstoječih procesov in zajem funkcionalnih zahtev	14
2.1.1 Proces obdelave dokumentov Ustavnega sodišča	15
2.1.2 Proces obdelav dokumentov Vrhovnega sodišča	18
2.1.3 Proces obdelave dokumentov Višjih sodišč, Upravnega sodišča in povzetkov negmotnih škod	21
2.1.4 Proces obdelave dokumentov Računskega sodišča	25
2.1.5 Tok podatkov v obstoječih procesih	25
2.2 Prenova procesov	27
2.2.1 Zajem dokumentov in tehnična pretvorba	28
2.2.2 Priprava dokumentov na uvoz v IUS DOCUMENT	31
2.2.3 Urejanje in dopolnitev vsebine dokumentov	33
2.2.4 Prenos dokumentov v podatkovna skladišča	35
3 Sistem za upravljanje procesov	37
3.1 Opis sistema	37
3.2 Arhitektura sistema in načrtovanje	38
3.2.1 Aplikacijska plast	42

3.2.2	Storitvena plast	42
3.2.3	Podatkovna plast	44
3.3	Implementacija prototipa sistema	48
3.3.1	Tehnologije	48
3.3.2	Implementacija spletne storitve za okrestracijo procesa	52
3.3.3	Implementacija podpornih spletnih storitev	61
3.3.4	Implementacija spletnega vmesnika	61
3.4	Razširitve sistema in alternativne rešitve	63
3.4.1	Windows Workflow Foundation	63
3.4.2	Orodja za izvajanje BPEL (BPEL engine)	65
4	Sklepne ugotovitve	69
4.1	Prednosti	71
4.2	Slabosti	72
4.3	Priložnosti	72
4.4	Nevarnosti	73
	Dodatki	74
	A Diagram prenovljenega procesa za obdelavo dokumentov sodne prakse	74
	Seznam slik	78
	Literatura	79

Seznam uporabljenih kratic in simbolov

- BPMN** notacija za modeliranje poslovnih procesov (ang. Business Process Management Notation)
- BPEL** jezik za izvajanje poslovnih procesov (ang. Business Process Execution Language)
- ETL** zajemi, pretvori, naloži (ang. Extract-Transform-Load)
- IIS** kratica Microsoftovega spletnega strežnika (ang. Internet Information Services)
- SOAP** protokol za izmenjavo strukturiranih sporočil med spletnimi storitvami (ang. Simple Object Access Protocol)
- SOA** storitveno usmerjena arhitektura (ang. Service-oriented Architecture)
- UML** poenoteni jezik za modeliranje (ang. Unified modeling language)
- WAS** kratica strežnika za aktivacijo procesov (ang. Windows Process Activation Service)
- WCF** ogrodje za komunikacijo v .NET (ang. Windows Communication Foundation)

Povzetek

Dobro zasnovani in primerno upravljani poslovni procesi so pomemben del uspešnega poslovanja podjetja.

Diplomska naloga opisuje postopek popisa in prenove poslovnih procesov za obdelavo dokumentov v podjetju IUS SOFTWARE ® ter načrtovanje in razvoj sistema za upravljanje teh procesov.

Projekt popisa in prenove procesov v podjetju je bil vzpostavljen zaradi zastarelih, slabo dokumentiranih procesov in potrebe po njihovi optimizaciji in poenostavitvi. Prenovljeni proces, ki sem ga izdelal na podlagi popisa in modeliranja obstoječih procesov, je predstavljene v prvem delu, skupaj z omenjenimi procesi.

Sistem za upravljanje procesov, opisan v drugem delu, sem zasnoval zaradi potrebe po učinkovitem vodenju in nadziranju procesov obdelave dokumentov v podjetju. Temelji na prenovljenemu procesu. Arhitektura načrtovanega sistema sledi smernicam storitveno-usmerjene arhitekture (SOA) in načelom interoperabilnosti. Za praktičen prikaz delovanja sistema sem izdelal prototip. Ta temelji na Microsoftovemu ogrodju .NET in pripadajočih tehnologijah. Za implementacijo prototipa v tem ogrodju sem se odločil zaradi večje kompatibilnosti z obstoječo infrastrukturo v podjetju.

Poleg omenjenega sistema sem ocenil tudi obstoječe alternativne rešitve komercialnih ponudnikov. Te rešitve niso primerne zaradi specifičnosti procesov in zaradi njihove kompleksnosti, ki jih podjetje ne potrebuje.

Izdelani prototip potrjuje, da je sistem zasnovan pravilno, in da omogoča učinkovito upravljanje s procesi. Ustreza vsem potrebnim kriterijem in je primerna rešitev za upravljanje procesov v podjetju.

Ključne besede:

prenova procesov, upravljanje procesov, BPMN, SOA, spletne storitve

Abstract

Well designed and properly managed business processes are a vital part of successful businesses.

Subject of this thesis is documentation and redesign of business processes in the company IUS SOFTWARE. The thesis also describes the process of design and planning of a Business Process Management System. The project for business process redesign was started because of the need to optimize and update currently running processes in the company. I developed a new process (redesigned process), based on documentation and modeling of existing processes. The redesign and the process itself are described in the first part of the thesis.

Business Process Management System is described in the second part of the thesis. The need for this system comes from the demand for efficient management and monitoring of processes, and their execution. System architecture follows the guidelines of the Service-oriented Architecture and interoperability.

To demonstrate the system, I designed a prototype. It is implemented in Microsoft .NET Framework and associated technologies. I chose this technology, because of the compatibility with the current with current company infrastructure. In addition to the fore mentioned system, I evaluated other possible commercial alternatives. As I concluded, these are not appropriate for use in the company, due to their complexity and the abundance of features, which the company actually does not require.

Developed prototype confirms, that the design of the system is appropriate and that it allows efficient management of business processes. It conforms with all requirements and is a suitable solution for the company.

Key words:

process redesign, process management, BPMN, SOA, web services

Poglavje 1

Uvod

Učinkoviti poslovni procesi so pomemben del uspešnega delovanja združb. Določajo zaporedje dela, hkrati pa so neposredno povezani z učinkovitostjo združbe. Dobri in optimizirani poslovni procesi pripomorejo h konkurenčnosti združbe na trgu in k zniževanju stroškov, zmanjšanju pritiska na zaposlene in boljšo izrabo različnih virov.

Poznavanje in razumevanje poslovnih procesov v podjetju je pomembno, ker nam omogoča, da zaznamo ozka grla in optimiziramo procese. Vsako podjetje mora poznati svoje poslovne procese: vedeti mora, kdo jih izvaja in koliko časa potrebuje za njihovo dokočanje (izvedbo).[4]

Veliko vlogo pri tem igrajo tudi sistemi za upravljanje procesov.

Diploma obravnava razvoj tovrstnega sistema za upravljanje procesov v podjetju IUS SOFTWARE. Nastala je v sklopu projekta prenove poslovnih procesov za obdelavo dokumentov. Cilj je izdelati sistem, ki bo omogočal učinkovito vodenje in upravljanje procesov, hkrati pa upošteval smernice storitveno-usmerjene arhitekture in bo interoperabilen. Osnova za izdelavo takega sistema so - enostavni in optimizirani - prenovljeni procesi.

Družba IUS SOFTWARE se ukvarja z načrtovanjem, razvojem, upravljanjem, vzdrževanjem in distribucijo podatkovnih baz pravne in poslovne narave ter razvojem informacijsko-tehnoloških in programskih rešitev za povezovalne sisteme. Ti omogočajo celovito upravljanje z vsebinami nestrukturiranih dokumentov (besedila, upravljanje z vsebinami, upravljanje z dokumenti).

Poglavitne dejavnosti družbe so spletni portali:

- Portal **IUS-INFO** [®], pravni in poslovni informacijski portal, je komercialna baza podatkov na spletu, ki vsebuje največji izbor pravnih informacij iz Slovenije in Evropske unije na enem mestu (<http://www.>

iusinfo.si),

- Portal **FinD-INFO**, finance in davki je najsodobnejši portal, ki temelji na tehnologiji IUS-INFO. Portal FinD-INFO ponuja celovit pregled finančnih, računovodskih, davčnih in pravnih informacij ter omogoča enostavno iskanje podlag za rešitev strokovnih vprašanj (<http://www.findinfo.si>),
- **Hrvaški portal IUS-INFO**, pravni in poslovni informacijski portal, je komercialna baza podatkov na spletu, ki vsebuje največji izbor hrvaških pravnih informacij in pravnih informacij Evropske unije na enem mestu (<http://www.iusinfo.hr>).

Celoten sistem spletnih portalov je zasnovan na enotni arhitekturi in deluje v večjezičnem okolju.

Procesi, obravnavani v tem diplomskem delu, so zelo pomembni za podjetje. Gre za procese, v sklopu katerih poteka uvoz različnih vrst dokumentov v glavni sistem za shranjevanje dokumentov v podjetju, sistem IUS CONTENT, in njihova obdelava. Sistem je podrobneje opisan v podpoglavju 1.1.

Projekt za prenovu procesov je bilo treba vzpostaviti zaradi potreb po izboljšavi obstoječih procesov, zmanjšanja časa obdelave dokumentov in izboljšanja kvalitete dokumentov, ki nastanejo med izvajanjem procesov. Cilj projekta je posodobiti (oziroma po potrebi prenoviti) procese in vzpostaviti sistem za upravljanje s procesi. Sistem mora omogočati sledljivost izvajanja procesov in sledenje dokumentom, pregled nad stanjem trenutne obdelave in avtomatizirano izvajanje aktivnosti procesov.

Zaradi velikega števila poslovnih procesov, ki potekajo v podjetju, se diplomsko delo osredotoča samo na procese obdelave dokumentov sodne prakse. To so odločbe sodišč, ki delujejo v Republiki Sloveniji. Omenjeni procesi so dovolj kompleksni in reprezentativni, da omogočajo razvoj in načrtovanje sistema za upravljanje procesov, ki bo podpiral upravljanje večine procesov. Podroben opis obstoječih procesov je v podpoglavju 1.2.

Diplomsko delo je razdeljeno na dva dela: v prvem delu je predstavljen popis in prenova obstoječih procesov. Ta del je osnova drugemu delu, v katerem je predstavljena zasnova sistema za upravljanje procesov, postopek načrtovanja in izdelava prototipa sistema.

1.1 Sistem IUS CONTENT

IUS CONTENT je sistem, razvit znotraj podjetja IUS SOFTWARE, namenjen hranjenju različnih vsebin, med drugimi tudi dokumentov in vseh podatkov v zvezi z njimi. Sestavljen je iz več sklopov podatkovnih baz, kamor sodi tudi IUS TIME. Trenutna verzija IUS TIME-a 3.0 ne ustreza več potrebam podjetja, zato je v teku projekt posodobitve sistema IUS CONTENT, kar obsega tudi spremembe v podatkovnih bazah IUS TIME, ki bo dobil tudi novo ime, in sicer IUS TIME 4.0. Vse te spremembe so v celoti upoštevane pri prenovi procesov.

1.1.1 IUS TIME 3.0

IUS TIME 3.0 sestavljata dve podatkovni bazi: IUS TIME **WORK** in IUS TIME **PUBLISH**.

V podatkovni bazi IUS TIME WORK so shranjeni dokumenti, nad katerimi poteka obdelava. V bazi IUS TIME PUBLISH pa so shranjeni dokumenti, ki so pripravljene za prikaz na portalih.

Dokument je v sistemu IUS CONTENT predstavljen s segmenti. Posamezen dokument ima enega ali več segmentov. Segment je tudi najmanjša enota obdelave v sistemu IUS CONTENT. Vsebina segmentov je različna za posamezne vrste dokumentov. Pri zakonih je segment posamezen člen zakona, včasih tudi odstavek v členu, medtem ko pri sodnih odločbah en segment vsebuje izrek, drugi segment obrazložitev, in tako dalje. Segmente se definira za vsak tip dokumentov posebej.

Struktura obeh podatkovnih baz je enaka in je s konceptualnim modelom prikazana na sliki 1.1. Glavni deli podatkovne baze so trije:

- del, kjer se shranjujejo dokumenti (rdeče),
- del, kjer se shranjujejo povezave (zeleno) in
- del, kjer se shranjujejo vrednosti šifrantov (temno zeleno).

Dokumenti so v podatkovni bazi shranjeni v obliki segmentov. Pri uvozu dokumente razčlenimo na vnaprej določene segmente. Vrednosti segmentov so shranjene v tabeli *DocSegment*, metapodatki o segmentih pa v tabelah *DocSegmentClass*, *DocSegmentDesc* in *DocSegmentType*.

Podatki o dokumentu, kot so enolični identifikator (SOPI), datum začetka veljavnosti, datum konca veljavnosti, in drugi, so shranjeni v tabeli *DocHeader*. Podatke o vseh vrstah dokumentov hranimo v tabeli *DocType*.

Poleg dokumentov so v podatkovni bazi IUS TIME shranjeni tudi podatki o povezavah med dokumenti. Povezave so zapisane v tabelah *DocLinkDoc*, *DocLinkSegments* in *DocLink*.

Dokumenti vsebujejo tudi vrednosti iz šifrantov. Šifranti se nahajajo v tabelah *DocCollection* in *DocCollectionElement*.

Vse ostale tabele, prikazane na sliki 1.1, so podporne tabele.

1.1.2 IUS TIME 4.0

Prenovljeni IUS TIME 4.0 je v zadnjih fazah razvoja in testiranja, zato (razen posameznih sklopov) še ni v produkciji. Sestavljen je iz več podatkovnih baz in skupin podatkovnih baz:

- podatkovne baze IUS STAGE,
- skupine podatkovnih baz IUS TIME WORK in
- podatkovnih skladišč IUS TIME PUBLISH.

Podatkovna baza IUS STAGE

Podatkovna baza IUS STAGE je novost v skupini podatkovnih baz IUS TIME. V njej se zbirajo dokumenti in metapodatki o dokumentih, zajetih iz zunanjih virov. Namen te baze je zbiranje podatkov in priprava na nadaljnjo obdelavo. Vsi prejeti dokumenti se poleg tega v bazi tudi arhivirajo. Nad podatki v tej bazi poteka osnovno preverjanje pravilnosti podatkov oziroma minimalna validacija.

V podatkovno bazo IUS STAGE se shranjujejo dokumenti v »surovi obliki«
tako po zajemu (prejem po e-pošti, zajem iz spleta, ročni vnos). Baza je osnova za nadaljnji operaciji obdelave dokumentov:

- **razčlenjevanje dokumentov**
(rezultate te operacije nato shranimo v IUS DOCUMENT)
- **uvoz vrednosti šifrantov**
(rezultate te operacijo pa shranimo v IUS COLLECTION).

Podatki se iz podatkovne baze IUS STAGE prenesejo v podatkovne baze v IUS TIME WORK.

Skupina podatkovnih baz IUS TIME WORK

Podatkovne baze v IUS TIME WORK so namenjene obdelavi dokumentov. Skupina teh podatkovnih baz znotraj novega IUS TIME 4.0 bo nadomestila podatkovno bazo IUS TIME WORK v starem IUS TIME-u. V IUS TIME WORK so naslednje podatkovne baze:

- IUS LINK,
- IUS DOCUMENT,
- IUS COLLECTION.

Podatkovna baza IUS LINK vsebuje podatke o povezavah med posameznimi dokumenti, ki se nahajajo v podatkovni bazi IUS DOCUMENT.

Podatkovna baza IUS DOCUMENT hrani podatke o dokumentih in vsebino dokumentov, razdeljeno na segmente.

Podatkovna baza IUS COLLECTION vsebuje podatke o šifrantih in klasifikacijah dokumentov.

Podatkovna skladišča IUS TIME PUBLISH

Podatkovna skladišča s skupnim imenom IUS TIME PUBLISH bodo nasledila staro podatkovno bazo IUS TIME PUBLISH. Podatkovna skladišča vsebujejo podatke, optimizirane za prikaz na spletnih portalih. Podatki se v podatkovna skladišča polnijo iz podatkovnih baz v IUS TIME WORK z ETL procesi.

Ker so podatkovne baze IUS TIME v zadnjih fazah razvoja in bodo kmalu prešle v produkcijo, je pomembno, da so prenovljeni procesi zasnovani na novem sistemu.

1.2 Opis obstoječih procesov za obdelavo dokumentov sodne prakse

Sodna praksa in judikati

Sodno prakso sestavljajo odločitve sodišč. To so odločbe oziroma judikati. Sodna praksa ne predstavlja formalnega vira prava, ampak služi kot posvetovalno orodje sodnikom. Sem sodijo judikati Vrhovnega sodišča RS (in primerljivih sodišč republik bivše Jugoslavije), odločbe višjih sodišč ter Upravnega sodišča RS. Poleg tega k sodni praksi sodijo tudi odločbe o odmeri pravične denarne odškodnine za nepremoženjsko škodo – povzetki negmotnih škod (<http:>

[//www.sodisce.si/sodni_postopki/objave/2009021320342893/](http://www.sodisce.si/sodni_postopki/objave/2009021320342893/)). V podjetju med dokumente sodne prakse uvrščamo še odločbe Ustavnega sodišča RS.

Družba od sodišč prejema judikate. Za obdelavo teh poteka pet procesov, in sicer za vsak tip sodnih odločb posebej. Odločbe prejemamo po elektronski pošti v formatu ODT (Open Document Text), razen odločb Ustavnega sodišča, ki so v formatu XML.

Trenutne procese je podjetje vzpostavilo pred osmimi leti. Med tem je prišlo do sprememb tako v formatu dokumentov, ki jih zajemamo, kot tudi v aktivnostih procesov. Poglavitni razlog za omenjene spremembe je spremenjen način prikaza dokumentov na portalu. Podjetje je v tem času namreč prešlo iz gradnje statičnih HTML dokumentov v dinamični prikaz dokumentov na spletnih straneh (tehnologija ASP.NET). Pri vpeljavi sprememb podjetje ni izvedlo ustreznega načrtovanja, zato se je kompleksnost procesov zelo povečala, razumljivost procesov pa se je temu primerno zmanjšala.

Tehnologija je napredovala in nekatere aplikacije so zastarele, delo z njimi pa težavno. Določene aplikacije so bile zgrajene še v 16 bitni tehnologiji in jih na sodobnih, 64 bitnih sistemih, ni več mogoče uporabljati. Uporabljati je potrebno virtualizacijo 32 bitnih sistemov, kar zahteva več časa, kompleksnost procesov pa se s tem še poveča. Poleg tega v aplikacijah poteka gradnja HTML datotek, kar zahteva star način priprave dokumentov za prikaz. Ta gradnja je sedaj odveč in po nepotrebnem podaljšuje čas obdelave dokumentov.

Podpora izvajanju aktivnosti procesov nudi 18 različnih aplikacij, razvitih znotraj podjetja, ki so napisane v različnih programskih jezikih. Večina od uporabnika zahteva samo pritisk na gumb, kar bi se dalo enostavno prenesti v popolnoma avtomatizirano aktivnost. Poleg teh se uporabljajo še splošni pisarniški programi, kot so Microsoft Word, Microsoft Outlook in UltraEdit.

Ena od težav, ki se pojavlja pri obstoječih procesih, je pomanjkanje sledljivosti dokumentov. Težko je namreč ugotoviti, iz katere datoteke izhajajo dokumenti. To je predvsem pomembno pri odpravljanju napak pri že prejetih in uvoženih dokumentih. Ko se proces zaključi, je potrebno viru poslati poročilo o uvozu dokumentov. Poročilo mora vsebovati podatke o številu uvoženih dokumentov, njihovih evidenčnih številkah in še nekatere druge podatke. Evidenco za pripravo poročila trenutno izvajalci procesov vodijo ročno, kar je zelo zamudno, hkrati pa dovzetno za napake.

Čeprav imajo obstoječe aplikacije vgrajene kompleksne in včasih tudi redundantne postopke, nimajo vgrajenih dovolj samodejnih kontrol za preverjanje pravilnosti in kakovosti vsebine dokumentov. Vsebina dokumentov zaradi tega ni najvišji možni ravni kakovosti.

Tudi nadzor nad izvajanjem procesov je pomanjkljiv. Trenutno je težko ugotoviti, v kateri fazi izvajanja je določen proces, kdo ga izvaja in kdo bi ga moral izvajati (pa ga ne). Poleg tega so slabo definirane vloge udeležencev v procesih. Velikokrat se zgodi, da proces izvajajo zaposleni v oddelku Informatike, ki naj bi nudili samo podporo izvajanju procesov, ne pa jih tudi izvajati, kar je delo oddelka za urejanje vsebin in oddelka za procese.

Zaradi vseh naštetih težav smo se odločili prenoviti procese za obdelavo dokumentov sodne prakse. Prenovljeni proces sem uporabil kot izhodišče pri načrtovanju sistema za upravljanje procesov.

Poglavje 2

Popis, prenova in modeliranje procesov

Poglavje podrobneje opisuje metodo dela, ki sem jo uporabil pri popisu obstoječih procesov in predlagano rešitev za prenovu procesov. Pojasnjuje, kako sem zastavil prenovu procesov, katero diagramsko tehniko sem uporabil, in katere programe. Vsak obstoječ proces je podrobneje opisan, skupaj s težavami, ki jih je pri prenovi procesa treba odpraviti. V drugem delu poglavja je predstavljena predlagana rešitev – prenovljen proces – podrobneje pa tudi njegove faze.

Najprej pa nekaj besed o diagramski tehniki BPMN, ki sem jo uporabil za modeliranje procesov.

Business process management notation (BPMN)

Diagramaska tehnika BPMN (www.bpmn.org) je namenjena modeliranju poslovnih procesov. Razvija jo skupina OMG (Object Management Group – www.omg.org). Tehnika je podobna diagramom aktivnosti v UML. Nudi standardizirano notacijo, ki je razumljiva tudi poslovnim uporabnikom. To je še posebej pomembno, saj sem med prenovu procesov sodeloval z uporabniki, ki niso vešči uporabe kompleksnih procesnih diagramov. Ker imajo pomembno vlogo pri izvajanju procesov, je bilo njihovo razumevanje diagramov nujno za ustrezno komunikacijo in rešitev problemov.

Za uporabo diagramske tehnike BPMN sem se odločil tudi zato, ker je v diagramih enostavno določiti aktivnosti, ki se izvajajo samodejno. Te aktivnosti so potencialni kandidati za spletne storitve, kar poenostavi načrtovanje

spletnih storitev. Pri modeliranju sem uporabljal specifikacijo BPMN 1.2 (<http://www.omg.org/spec/BPMN/1.2/>).

V času izdelave te diplomske naloge je specifikacija BPMN 2.0 še v razvoju.

Začetek prenove

Osnova za prenavo procesov so cilji, ki smo jih zastavili na več sestanki znotraj združbe in jih mora prenovljeni proces podpirati. Glavni nameni in cilji, ki jih mora prenovljeni proces podpirati, so:

- odprava ozkih grl v procesih,
- povečanje produktivnosti,
- izboljšanje kakovosti procesov,
- zmanjšanje tveganja za napake,
- izboljšanje informacijske podpore za vse udeležence:
 - vodstvo,
 - skrbnike procesov,
 - izvajalce procesov,
- izvedba v okviru predvidenega časa in stroškov,
- boljša sledljivost izvajanja procesov.

Prenovo procesov sem začel s popisom obstoječih procesov.

Najprej sem pregledal obstoječo dokumentacijo. Ker so se procesi v času, odkar so bili vzpostavljeni, spremenili, sem moral intervjuvati posamezne udeležence v procesih. Tako sem zajel dejansko stanje procesov. Ko sem zbral vse podatke v zvezi s posameznimi procesi, sem se lotil izdelave diagramov. Procese sem modeliral z orodjem BizAgi Process Modeler (<http://www.bizagi.com>). Pri izdelavi diagramov se nisem ukvarjal s preveč podrobnostmi, saj so te opisane v dokumentaciji. Izdelani diagrami so namenjeni celovitemu pogledu na procese. To mi je omogočilo, da sem iz njih razbral skupne aktivnosti, ki so podobne ali celo enake v vseh popisanih procesih.

Pri načrtovanju novega procesa sem poleg popisanih procesov upošteval tudi spremenjene okoliščine, novo tehnologijo in njene zmožnosti. Poleg intervjujev sem opravil tudi pogovore z udeleženci in zbral seznam funkcionalnih zahtev, ki jih mora novi proces podpirati.

2.1 Popis obstoječih procesov in zajem funkcionalnih zahtev

Glavni namen popisa procesov je zajem njihovega dejanskega stanja. S tem izdelamo tako imenovani as-is model. Za izdelavo obstoječega modela je potrebno pridobiti vse znanje o procesu. Popis obstoječih procesov sem začel s pregledom dokumentacije. Dokumentacija je bila izdelana skupaj s procesi, vendar ni bila dopolnjena, ko je prišlo do sprememb v procesih. Zato sem moral poleg pregleda dokumentacije tudi intervjuvati izvajalce in skrbnike procesov. Poleg tega, da sem ugotovil obseg sprememb, ki so se zgodile v času obstoja procesov, sem s podrobnimi intervjuji zajel skrito znanje zaposlencev, ki delajo na procesih. To znanje je pomembno, da zares dobimo vse informacije o procesih in njihovem izvajanju.[4]

Tako sem dobil dejansko stanje procesov, kot se trenutno izvajajo. Iz dokumentacije sem identificiral pet različnih procesov za obdelavo dokumentov sodne prakse. Ugotovil sem, da za vsak tip dokumentov poteka ločen proces za obdelavo. Posamezni procesi so podrobneje opisani v naslednjih podpoglavjih.

2.1.1 Proces obdelave dokumentov Ustavnega sodišča

Ustavno sodišče podjetju pošlje svoje odločbe po elektronski pošti. Odločbe so v datoteki XML. V posamezni datoteki je lahko ena ali več odločb. Priložena je tudi datoteka XLS, ki vsebuje metapodatke o dokumentih. Po prejemu datoteke izvajalec procesa pretvori določene značke v XML. Potem dopolni polje s povezavami. To mora storiti zaradi omejitev programa, ki kasneje obdeluje dokument, saj zna ta program prepoznati povezave le iz določenih polj.

Nato datoteko, ki vsebuje odločbe, naloži v program Evad. Program pretvori posebne znake in razvrsti posamezne odločbe v pripadajoča področja geslovnika. Geslovník je šifrant, njegove kategorije pa so v naprej določene s strani Evropske komisije za demokracijo skozi pravo/Beneška komisija (European Commission for Democracy through Law - Venice Commission, <http://www.venice.coe.int>). Izvajalec procesa ročno preveri pravilnost razvrstitve odločb v geslovník. V primeru nepravilnega razvrščanja, ki nastane zaradi slovničnih napak, popravi gesla in ponovi razvrščanje. Sledi uvoz dokumentov v podatkovno bazo IUS TIME. Po uspešnem uvozu se s posebnim programom LinksManager avtomatsko zgradijo povezave med odločbami in zakoni oziroma predpisi.

Zadnji korak tega procesa je prenos dokumentov v podatkovno skladišče IUS TIME PUBLISH.

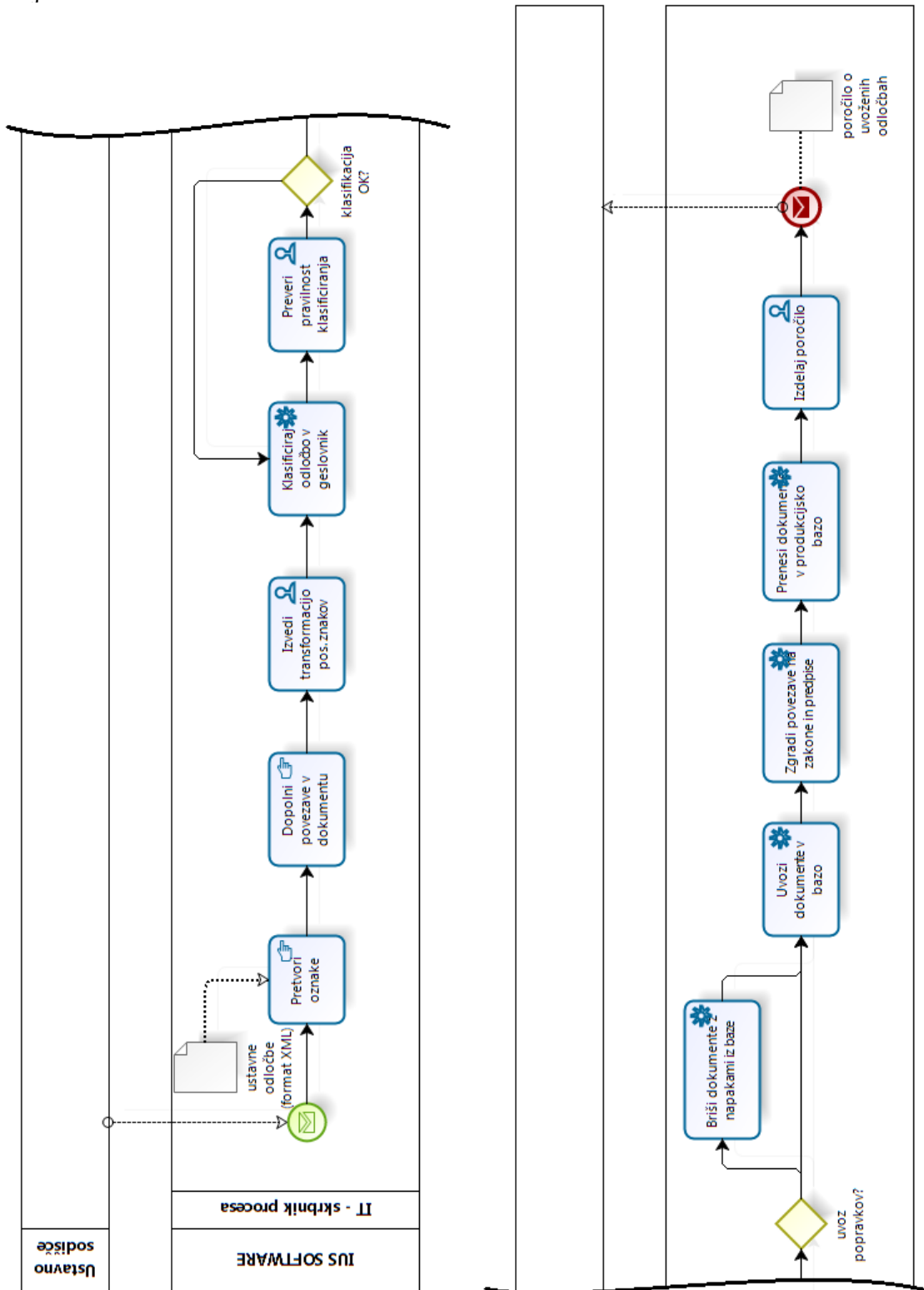
Diagram procesa obdelave dokumentov Ustavnega sodišča je prikazan na sliki 2.1.

Težave pri izvajanju procesa

Ker oddelek vsebin ne pregleda uvoženih dokumentov (odločb), je končna kakovost vsebine dokumenta manjša, kot bi lahko bila. Poleg tega veliko popravkov izvajalec procesa opravi ročno, čeprav so dovolj formalizirani, da bi se jih lahko izvajalo samodejno, z možnostjo dodatnega ročnega preverjanja.

2.1.

2.1 Popis obstoječih procesov
in zajem funkcionalnih zahtev



Slika 2.1: Diagram procesa obdelave dokumentov Ustavnega sodišča

Primer datoteke XML

```

<ZAPIS>
<DOKUMENT>US29270</DOKUMENT>
<ZAUPNOST></ZAUPNOST>
<REPUB>5</REPUB>
<VRSTAD>10</VRSTAD>
<DVLOG>20080604</DVLOG>
<VRSTAV>V25</VRSTAV>
<VRSTAA>A14</VRSTAA>
<DODL>20101014</DODL>
<VODL>1</VODL>
<VRES>R03</VRES>
<DVHOD>20101103</DVHOD>
<JEZIK>1</JEZIK>
<POD>PP05</POD>
<ZADEVA>Up-1678/08</ZADEVA>
<EVIDENCA>E-XI-71/10</EVIDENCA>
<IMEVLAG>F***** K*****, Ljubljana</IMEVLAG>
<AKT>Ustavni pritožbi zoper sodbo Vrhovnega sodišča št. I Ips 8/2007 z dne 17. 1. 2008 v
zvezi s sodbo Višjega sodišča v Ljubljani št. I Kp 1386/2005 z dne 12. 4. 2006 in s
sodbo Okrožnega sodišča v Ljubljani št. III K 156/2003 z dne 13. 4. 2005</AKT>
<PROBLEM>1.5.51.2.6 – Ustavno sodstvo – Odločbe – Vrste odločitev Ustavnega sodišča –
V postopku odločanja o ustavni pritožbi – Zavrnitev ustavne pritožbe.
5.3.13.52 – Temeljne pravice – Državlanske in politične pravice –
Procesna jamstva, pravica do obrambe in poštenega sojenja (19, 20, 21, 22, 23, 24, 25, 26,
27, 28, 29, 30, 31) – Pravica da se mu sodi v njegovi navzočnosti (29).
...
5.3.13.17 – Temeljne pravice – Državlanske in politične pravice –
Procesna jamstva, pravica do obrambe in poštenega sojenja (19, 20, 21, 22, 23, 24, 25, 26,
27, 28, 29, 30, 31) – Obrazložitev.</PROBLEM>
<PRAPODL>Člen 22, 23.1, 28.2, 29, Ustava [URS]
Člen 6, Konvencija o varstvu človekovih pravic in temeljnih svoboščin [EKČP]
</PRAPODL>
<PORPRED></PORPRED>
<SKLEPI></SKLEPI>
<IZREK>Ustavni pritožbi zoper sodbo Vrhovnega sodišča št. I Ips 8/2007 z dne 17. 1. 2008 v
zvezi s sodbo Višjega sodišča v Ljubljani št. I Kp 1386/2005 z dne 12. 4. 2006 in s
sodbo Okrožnega sodišča v Ljubljani št. III K 156/2003 z dne 13. 4. 2005 se zavrneta.
</IZREK>
<TEKST>Vzorec glasu, ki ne obsega inkriminirajočih izjav, ne posega v privilegij zoper
samooobtožbo, varovan s četrto alinejo 29. člena Ustave.
Stališče Vrhovnega sodišča, da bi moral pritožnik zahtevati izločitev razpravljajočega
sodnika do konca glavne obravnave, če je to možnost imel, ni v neskladju s pravico do
nepriistranskega sojenja iz prvega odstavka 23. člena Ustave.
...
</TEKST>
<AVTOR></AVTOR>
<OBJAVA>Uradni list RS, št. 88/2010</OBJAVA>
<POSILANO></POSILANO>
<OPOMBA>
K obravnavani zadevi Up-1678/08 je bila s sklepom Ustavnega sodišča z dne 10. 12. 2009
pridružena zadeva št. Up-1718/08 zaradi skupnega obravnavanja in odločanja.</OPOMBA>
<PTEKST>Up-1678/08-13
14. 10. 2010
ODLOČBA
Ustavno sodišče je v postopku odločanja o ustavnih pritožbah F. K., ki ga zastopa
Odvetniška družba Grobelnik in partnerji, o. p., d. n. o., Ljubljana, na seji 14.
oktobra 2010
odločilo:
...
1. Pritožnik je bil spoznan za krivega nadaljevanega kaznivega dejanja
...
17. Ustavno sodišče je sprejelo to odločbo na podlagi prvega odstavka 59. člena ZUstS v
sestavi: predsednik Jože Tratnik ter sodnice in sodniki mag. Marta Klampfer, dr.
Etelka Korpič – Horvat, mag. Miroslav Mozetič, dr. Ernest Petrič, Jasna Pogačar, mag.
Jadranka Sovdat in Jan Zobec. Sodnik dr. Mitja Deisinger je bil pri odločanju v
zadevi izločen. Odločbo je sprejelo soglasno.
Jože Tratnik
Predsednik
Opombe:
[1] Tako ESČP v sodbi v zadevi Saunders proti Združenemu kraljestvu,
na katero se sklicuje tudi Vrhovno sodišče, 68. točka; pred tem ESČP
uvrsti navedeni pravici v 6. člen EKČP že v sodbi v zadevi Funke proti
Franciji z dne 25. 2. 1993, 44. točka.
</PTEKST>
</ZAPIS>

```

2.1.2 Proces obdelav dokumentov Vrhovnega sodišča

Opis procesa

Odločbe Vrhovnega sodišča podjetje prejme po elektronski pošti v datotekah ODT. V eni datoteki je lahko ena odločba ali paket več odločb. Priložena je tudi datoteka XLS, ki vsebuje metapodatke o paketu dokumentov.

Izvajalec nad prejetimi datotekami požene makro v programu Microsoft Outlook. Makro pretvori datoteke v format DOC in jih skopira na lokacije za nadaljnjo obdelavo ter v arhiv. Iz teh lokacij programi v nadaljevanju procesa berejo datoteke.

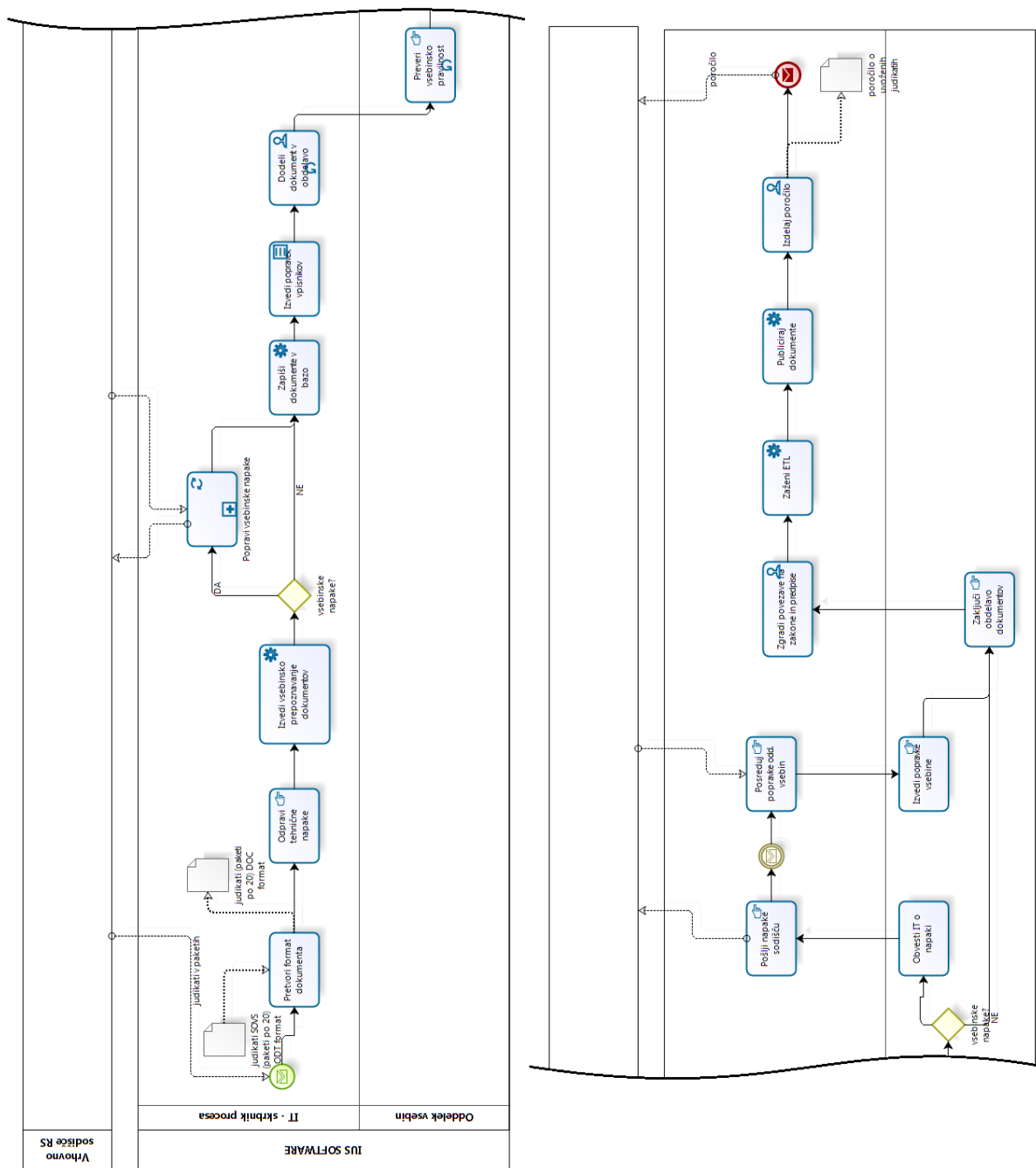
V naslednjem koraku procesa izvajalec s programom Evad odpre datoteke z dokumenti. Program izvede tehnično pretvorbo dokumentov in jih uvozi v podatkovno bazo IUS TIME WORK. Med uvozom lahko pride do napak. Če je le mogoče, jih izvajalec sproti odpravi, sicer jih pošlje v pregled viru. Ko sodišče pošlje popravke, dokumente ustrezno dopolni. Po končanem uvozu izvajalec procesa dodeli posamezne dokumente v pregled zaposlencem na oddelku vsebin. Oddelek vsebin izvede vsebinski pregled dokumentov in odpravi morebitne napake. Ko so napake odpravljene, se dokumentom določi status, da so primerni za objavo.

S programom LinksManager se avtomatsko zgradijo povezave na zakone oziroma predpise.

Za prikaz povezav tega tipa dokumentov se uporablja nova podatkovna baza IUS LINK, zato se po končani gradnji povezav zažene ETL proces, ki prenese podatke o povezavah v novo podatkovno bazo. Sledi prenos dokumentov v podatkovno skladišče.

Zadnji korak procesa je izdelava poročila za sodišče. Poročilo vsebuje podatke o dokumentih oziroma odločbah, ki so bile uvožene v sistem. Služi tako interni evidenci sodišča kot tudi evidenci v podjetju.

Slika 2.2 prikazuje BPMN diagram procesa obdelave dokumentov Vrhovnega sodišča.



Slika 2.2: Diagram procesa obdelave dokumentov Vrhovnega sodišča

Težave pri izvajanju procesa

Vsebina ne gre skozi vsebinski pregled, zato je kakovost vsebine manjša kot bi lahko bila. Prav tako se ne preverijo ustrezne povezave med dokumenti.

Primer datoteke

```
>dokument      : VS0013261
>oddelek       : 1
>datum         : 100930
>področje      : CIVILNO PROCESNO PRAVO – LASTNINJENJE
>institut      : načelo enakega varstva pravic – pravica do kontradiktornega
                postopka – pravica do izjave – udeležba drugih oseb v pravdi –
                intervencija – obvestitev drugega o pravdi – pravnomočnost –
                subjektivne meje pravnomočnosti – širjenje subjektivnih meja
                pravnomočnosti – predhodno vprašanje – načelo prirejenosti
                postopkov – upravni postopek – vezanost na upravni akt –
                lastninjenje – oškodovanje družbenega premoženja – postopek
                revizije – odločanje Agencije Republike Slovenije za plačilni
                promet, nadziranje in informiranje (APPNI) – odločba APPNI –
                ukrepi APPNI – trajne vloge delavcev – znižanje trajnih vlog –
                bistvena kršitev določb pravnega postopka
>jedro         :
                V položaju, ko pravnomočnost odločbe učinkuje tudi proti tretjim osebam (extra partes
                oziroma erga omnes), je nujno uveljaviti procesna sredstva, ki naj prizadetim
                zagotovijo možnost sodelovanja v postopku: zagotoviti jim je potrebno bodisi, da
                imajo možnost sodelovanja v prvem postopku, bodisi, da jih v njem izdana odločba ne
                bo zavezovala v drugih postopkih. Zgolj abstraktna možnost sodelovanja v predhodnem
                postopku ne zadošča; da jo prizadeta oseba lahko učinkovito izkoristi, mora biti o
                tem postopku obveščena.
>tekst         :
                Revizija se zavrne.

                Tožeča stranka mora v 15 dneh od vročitve te sodbe povrniti toženi stranki njene
                revizijske stroške v znesku 422,28 EUR.

                OBRAZLOŽITEV:

                1. Tožnica je od tožene stranke zahtevala plačilo valoriziranega zneska v višini 10.524,61
                EUR na podlagi pogodbe o trajni vlogi sredstev, ki jo je dne 28. 12. 1990 sklenila s
                pravno prednico tožene stranke.
                ...
                Citirano določilo se glasi : „Če pride med postopkom nekdo, ki dotlej ni bil stranka, in
                zahteva, da se pripusti kot stranka, preizkusi uradna oseba, ki vodi postopek, ali
                ima pravico biti stranka, in izda o tem sklep. Zoper sklep, s katerim se mu lastnost
                stranke ne prizna, je dovoljena posebna pritožba.“ Na sorodnost omenjenih institutov
                kaže še pojem stranke v upravnem postopku (tudi oseba, ki ima za varstvo svojih
                pravic ali pravnih koristi pravico udeleževati se postopka; 49. člen ZUP), ki je
                širši kot v pravdi in se približuje pojmu sosporniškega intervenienta (202. člen ZPP)
                .
                Op. št. (14):      Op. 2, str. 82–83.
                Op. št. (15):      Odločba Ustavnega sodišča Republike Slovenije Up–280/05 z dne 9. 2. 2006
                in sodba Vrhovnega sodišča Republike Slovenije VIII Ips 286/2008 z dne 10. 5. 2010.
>izvor         : Sodba II Ips 194/2010
>izvor2        : VSK Cp 481/2009
>zveza         : URS člen 22.
                ZPP člen 13, 202, 204, 319, 339, 339/2, 339/2–8.
                ZUP člen 139, 249, 249–1, 249–9, 250, 250/2.
                ZDSS–1 člen 51, 69.
                ZLPP člen 48a, 48b, 48b/2, 48b/5, 48c.
                ZZPLP člen 8.
                ZLPPOD člen 4, 4/1, 8, 8/1, 8/2, 8/3.
```

2.1.3 Proces obdelave dokumentov Višjih sodišč, Upravnega sodišča in povzetkov negmotnih škod

Opis procesa

Proces za obdelavo dokumentov Upravnega sodišča se od procesa za obdelavo dokumentov Višjih sodišč in povzetkov negmotnih škod razlikuje v tem, da se odločbe v datotekah obdeluje posamično, medtem ko se pri ostalih dveh procesih odločbe v datotekah obdelujejo v paketu. V vseh ostalih podrobnostih so procesi enaki. Diagram procesa v BPMN je prikazan na sliki 2.3.

Odločbe sodišč in povzetke negmotnih škod prejmemo po elektronski pošti v formatu ODT. Datoteka vsebuje eno ali več odločb, priložena pa je tudi preglednica XLS, v kateri so metapodatki o prejetih odločbah. Pri procesu obdelave dokumentov Upravnih sodišč se vsako odločbo posebej skopira v ločeno datoteko, pri tem pa se naredi še pregled pravilnosti polj. Vse napake, ki se odkrijejo v tej fazi, se sporočijo sodišču. Po prejemu popravkov gre odločba v nadaljnjo obdelavo.

V ostalih procesih se odločbe ne kopirajo v ločene datoteke, temveč se zgolj preveri pravilnost polj.

V naslednjem koraku se datoteke z makrojem v programu Microsoft Outlook skopirajo na posebej določene lokacije, odvisne od vrste dokumenta. Dokumente iz teh lokacij bere program Evad. Program izvede tehnično pretvorbo datotek in jih naloži v podatkovno bazo IUS TIME WORK.

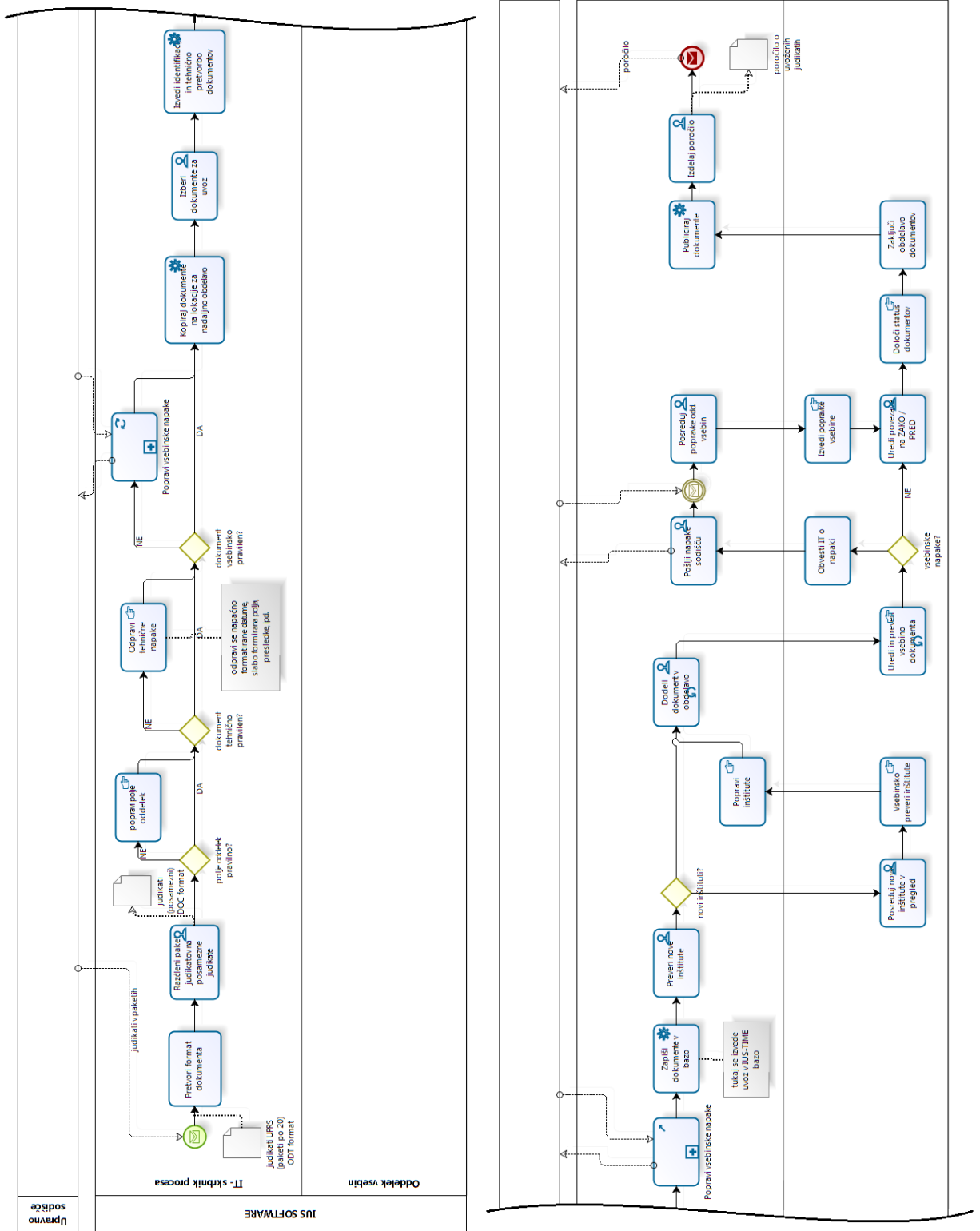
Zatem izvajalec ročno izvede poizvedbo SQL, s katero med novimi dokumenti poišče morebitne nove vrednosti polja institut. Seznam novih vrednosti izvajalec pošlje v pregled oddelku vsebin, ki preveri, če kakšni podobni instituti že obstajajo. Popravljen seznam vrne izvajalcu, ki nove vrednosti institutov vnese v šifrant.

Posamezne odločbe ali pakete izvajalec dodeli v obdelavo oddelku vsebin. Naloga zaposlencev tega oddelka je preveriti vsebinsko pravilnost dokumentov. Odločbo morajo tudi povezati na pripadajoče zakone oziroma predpise in šifrante.

Vse napake, ki jih odkrijejo v fazi vsebinskega preverjanja, zberejo in pošljejo viru (sodišču) po elektronski pošti. Po prejemu popravkov, te implementirajo in dokument vrnejo v nadaljnjo obdelavo. Ko je vsebinska obdelava dokumentov zaključena, se dokumente s programom *PublishingManager* prenese v podatkovno skladišče.

2.1 Popis obstoječih procesov in zajem funkcionalnih zahtev

21



Slika 2.3: Diagram procesa obdelave dokumentov Višjega sodišča, Upravnega sodišča, povzetkov negmotnih škod

Težave pri izvajanju procesa

Posamezne odločbe v prejetih datotekah izvajalec ročno kopira v ločene datoteke. To aktivnost bi se lahko avtomatiziralo, hkrati pa bi se lahko samodejno izvedlo tudi preverjanje pravilnosti polj in prisotnosti določenih obveznih polj. Veliko težav imajo tudi uredniki vsebine, saj je program CMS Manager zasnovan tako, da morajo vse, tudi pravilne ugotovitve avtomatike, potrditi, kar je zamudno.

Primer strukture prejete datoteke

```
>dokument      : UN0020295
>oddelek       : Upravni
>datum        : 100604
>področje      : ZUP
>institut      : popravni sklep - popravljanje pomot v odločbi - nepravilna uporaba
                 materialnega prava
>jedro        :
Pri izračunu obveznosti plačila obresti ni bilo upoštevano tudi povečanje za 2 odstotni
točki, kar je predpisano v 7. odstavku 62. g. člena ZZVZZ, to med strankama niti ni
sporno. Glede na navedeno sodišče ugotavlja, da tožena stranka pri izračunu
obveznosti očitno ni pravilno uporabila materialnega prava. Ker 223. člen ZUP popravno
takih napak ne omogoča, sodišče zaključuje, da tožena stranka za izdajo
izpodbijanega sklepa o popravni pomote ni imela pravne podlage. Za odpravo napak
storjenih zaradi očitnih kršitev materialnega prava v že izdani odločbi, je namreč v
ZUP predpisan poseben postopek, po katerem bi lahko tudi tožena stranka ugotovljeno
napako odpravila.
>tekst        :
Tožbi se ugoti, sklep Ministrstva za zdravje RS, Ljubljana številka 0070-38/2008-54 z dne
17.7.2009 se odpravi in se zadeva vrne toženi stranki v ponoven postopek.
OBRAZLOŽITEV :
Z izpodbijanim sklepom, navedenim v uvodu tega sklepa, je Ministrstvo za zdravje odločilo,
da se zneski določeni v drugi, tretji, četrti in peti točki izreka odločbe o
izravnavi št. 0070-38/2008-52 z dne 18.6.2009 nadomestijo z novimi višjimi zneski. V
obrazložitvi navaja, da je bila navedena odločba izdana na...
Tožeča stranka s tožbo izpodbija sklep o popravni napake v odločbi 0070-38/2008-52 zaradi
bistvenih kršitev določb Zakona o splošnem upravnem postopku (4. točka 1. odstavka
237. člena v zvezi s 1. odstavkom 259. člena ZUP). Navaja, da ministrstvo... Sodišču
zato predlaga, da izpodbijani sklep odpravi oz. podrejeno izreče za ničnega.
Tožena stranka, ki je sodišču predložila upravne spise, odgovora na tožbo ni podala.
Tožba je utemeljena.
Z izpodbijanim sklepom je bila tožeči stranki povečana obveznost določena v odločbi o
izravnavi št. 0070-38/2008-52 z dne 18.6.2009 iz zneska 1.017.851,52 EUR na...
učinkuje od dneva vročitve sklepa o popravku odločbe.
V obravnavani zadevi je sporno, ali je bil izpodbijani sklep o popravni pomote, utemeljeno
izdan na podlagi citirane določbe ZUP.
Dejstvo, da pri izračunu obveznosti plačila obresti, ki je bila za tožečo stranko določena
v odločbi z dne 18.6.2009, ni bilo upoštevano tudi povečanje za 2 odstotni točki, ki
je predpisano v 7. odstavku 62.f. člena ZZVZZ, med strankami ni sporno.
Na podlagi navedenega dejstva sodišče ugotavlja, da tožena stranka pri izračunu... lahko
tudi tožena stranka ugotovljeno napako odpravila.
Ker se tožena stranka v postopku izdaje izpodbijanega sklepa ni ravnala po pravilih
postopka, to pa je vplivalo na zakonitost in pravilnost odločitve, je sodišče tožbi
ugodilo in je izpodbijani sklep odpravilo v skladu s 3. točko prvega odstavka 64.
člena ZUS-1. Tožena stranka, ki ji je zadeva vrnjena v ponoven postopek, bo lahko v
skladu z določbo 4. odstavka istega člena izdala nov upravni akt.
>izvor        : sodba III U 430/2009 .
>izvor2       :
>zveza        : ZUP člen 223.
```

Struktura kontrolne datoteke

POROČILO O ODDAJI ODLOČB V SISTEM IUS-INFO

Datum oddaje:	Prejem 29. 10. 2010 od V***** R***** administratorica baze, prenos istega dne	Način: - e-pošta
Pošiljatelj: (ime, priimek, kontaktni naslov)	*****@sodisce.si	
Višje sodišče (CE, MB, LJ, KP, DSS, UprS)	Upravno sodišče RS	
Poslano prejemniku: (ime, priimek, naslov)	****@ius-software.si	
Baza: (IEPS, VDSS, UPRS)	UPRS	
Vsebina: 1. - datoteka – paket: - evidenčne številke odločb: - število odločb:	LMN.1 (UN0020294-UN0020313) UN0020294-UN0020313 20	
Opombe:		

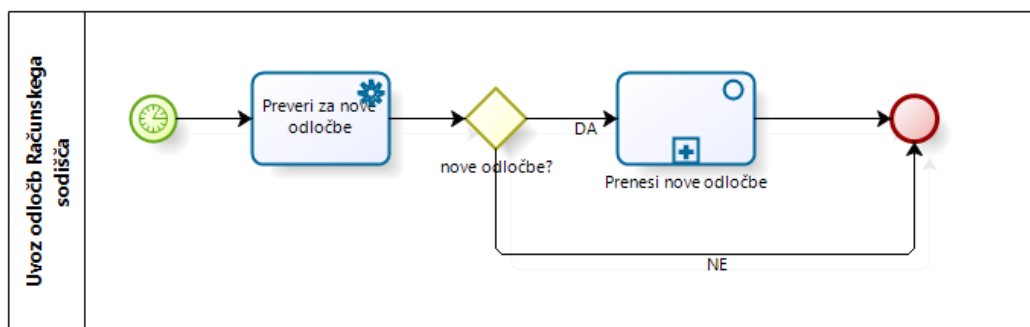
Slika 2.4: Struktua kontrolne datoteke

2.1.4 Proces obdelave dokumentov Računskega sodišča

Opis procesa

Proces obdelave dokumentov Računskega sodišča je od vseh opisanih najbolj enostaven. Dokumente podjetje prenaša iz spletne strani sodišča, jih obdela s programom ter shrani v podatkovno bazo IUS TIME WORK.

BPMN diagram procesa prikazuje slika 2.5.



Slika 2.5: Diagram procesa za obdelavo dokumentov Računskega sodišča

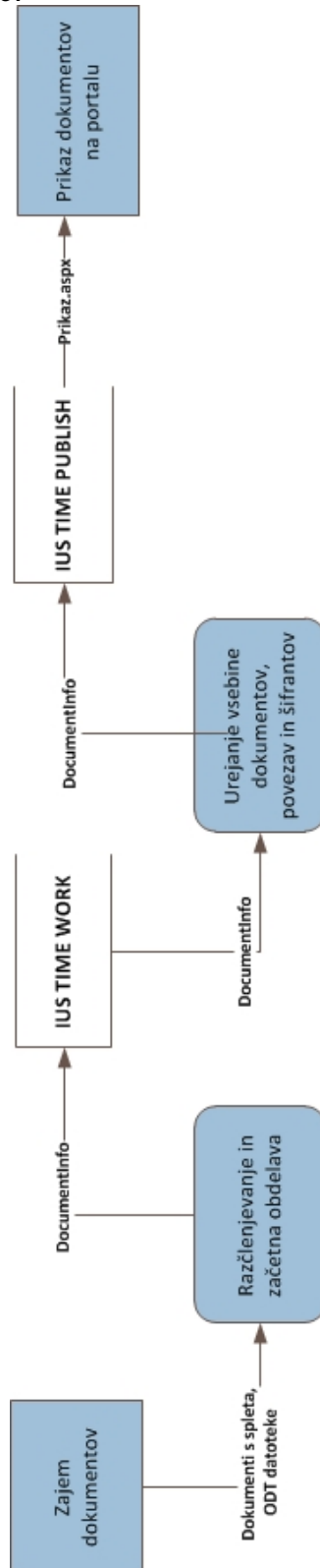
Težave pri izvajanju procesa

Dokumentov se vsebinsko ne pregleda, zaradi česar obstaja nevarnost, da je kvaliteta dokumentov v nekaterih primerih slabša. Prav tako se ne gradijo povezave na druge vrste dokumente, s čimer se izgublja potencialna dodana vrednost v podjetju.

2.1.5 Tok podatkov v obstoječih procesih

Po popisu vseh obstoječih procesov sem identificiral tudi tok podatkov med posameznimi podatkovnimi bazami in aktivnosti, ki se izvajajo nad temi podatki. Tok podatkov v teh procesih je prikazan na sliki 2.6.

2.1 Popis obstoječih procesov in zajem funkcionalnih zahtev



Slika 2.6: Tok podatkov v obstoječih procesih

2.2 Prenova procesov

Pri prenovi procesov sem upošteval cilje, ki smo si jih v podjetju zadali ob začetku projekta. Upošteval sem tudi funkcionalne zahteve, zajete v intervjujih z udeleženci v procesih.

Glavni cilj prenove procesov je prehod na novo tehnologijo in odprava težav, ki se pojavljajo pri obstoječih procesih. Pomemben poudarek sem namenil optimizaciji izvajanja procesov in čim večji avtomatizacija aktivnosti. Ročni poseg v proces se mora zgoditi le v primeru odkritih napak in pri preverjanju vsebinske pravilnosti. Vse ostale aktivnosti morajo biti čimbolj avtomatizirane.

Prenove procesov sem se lotil s pregledom obstoječih procesov. Najprej sem identificiral aktivnosti, ki so skupne vsem procesom. Ugotovil sem, da imajo procesi veliko enakih ali zelo podobnih aktivnosti. Na osnovi tega sem se odločil izdelati en sam proces, ki bo nadomestil obstoječih pet procesov. Zaradi tega bo izvajanje procesa bolj enostavno.

Opisan postopek prenove procesov sledi smernicam, kot jih opisuje M. Juric v knjigi *Business driven SOA using BPMN in BPEL*.

Zaradi zahteve po čim večji avtomatizaciji bo v novem procesu odveč veliko aplikacij, ki so prej skrbele za pretvorbo in razčlenjevanje. Namesto njih se ta logika prenese v aplikacijski strežnik, ki opravlja isto funkcijo samodejno. Edini program, ki ga bodo uredniki vsebin poleg spletnega vmesnika sistema za upravljanje procesov direktno uporabljali, je Stručko [6].

Upošteval sem tudi funkcionalne zahteve, ki sem jih zajel z intervjuji:

- povezovanje dokumentov se mora nujno izvesti pred ročni pregledom vsebine, tako da lahko uredniki vsebin preverijo pravilnost povezav.
- Nov proces in podporni sistem mora omogočati medsebojno povezovanje odločb. V primeru odločbe Vrhovnega sodišča mora obstajati odločba Višjega sodišča in nanjo je potrebno narediti povezavo. Enako velja za Ustavno sodišče, katerega odločbe se nanašajo na sodbe Vrhovnega sodišča. [5]
- Vzdrževati je treba evidenco dokumentov (v katerem paketu se je nahajal dokument), in sicer zaradi kasnejše revizije in vnašanja popravkov, pa tudi zaradi izdelave poročil.
- Pred prenosom dokumentov v podatkovno skladišče je treba obvezno pregledati vsebino. To morajo narediti uredniki, ki preverijo vsebino dokumenta, povezave in razvrstitev dokumenta v šifrantne ter kazala.

Nov proces je prilagojen arhitekturi prenovljenih podatkovnih baz IUS TIME 4.0. Zaradi tega so v procesu nove faze, ki jih v starih procesih ni bilo. Novi proces obdelave dokumentov sodne prakse je razdeljen v pet faz in tri prehodne (slika 2.7).

V nadaljevanju je opisanih pet glavnih faz procesa:

- zajem dokumentov in tehnična pretvorba,
- priprava dokumentov na uvoz v pod. bazo IUS DOCUMENT,
- urejanje in dopolnitev vsebine dokumentov,
- gradnja in preverjanje povezav in
- prenos dokumentov v podatkovna skladišča.

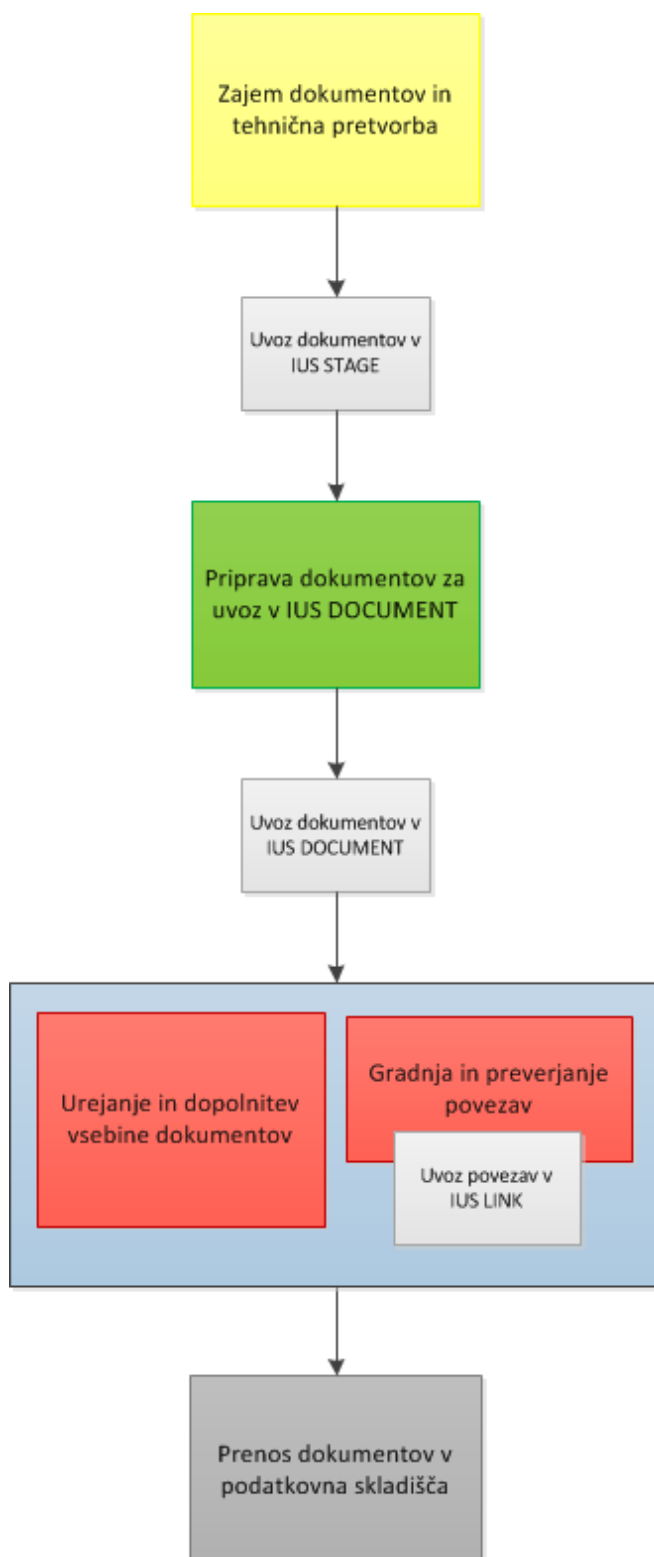
Opis prehodnih faz ni potreben, ker so preproste in samoumevne. Diagram celotnega novega procesa v BPMN je razviden v dodatku A.

2.2.1 Zajem dokumentov in tehnična pretvorba

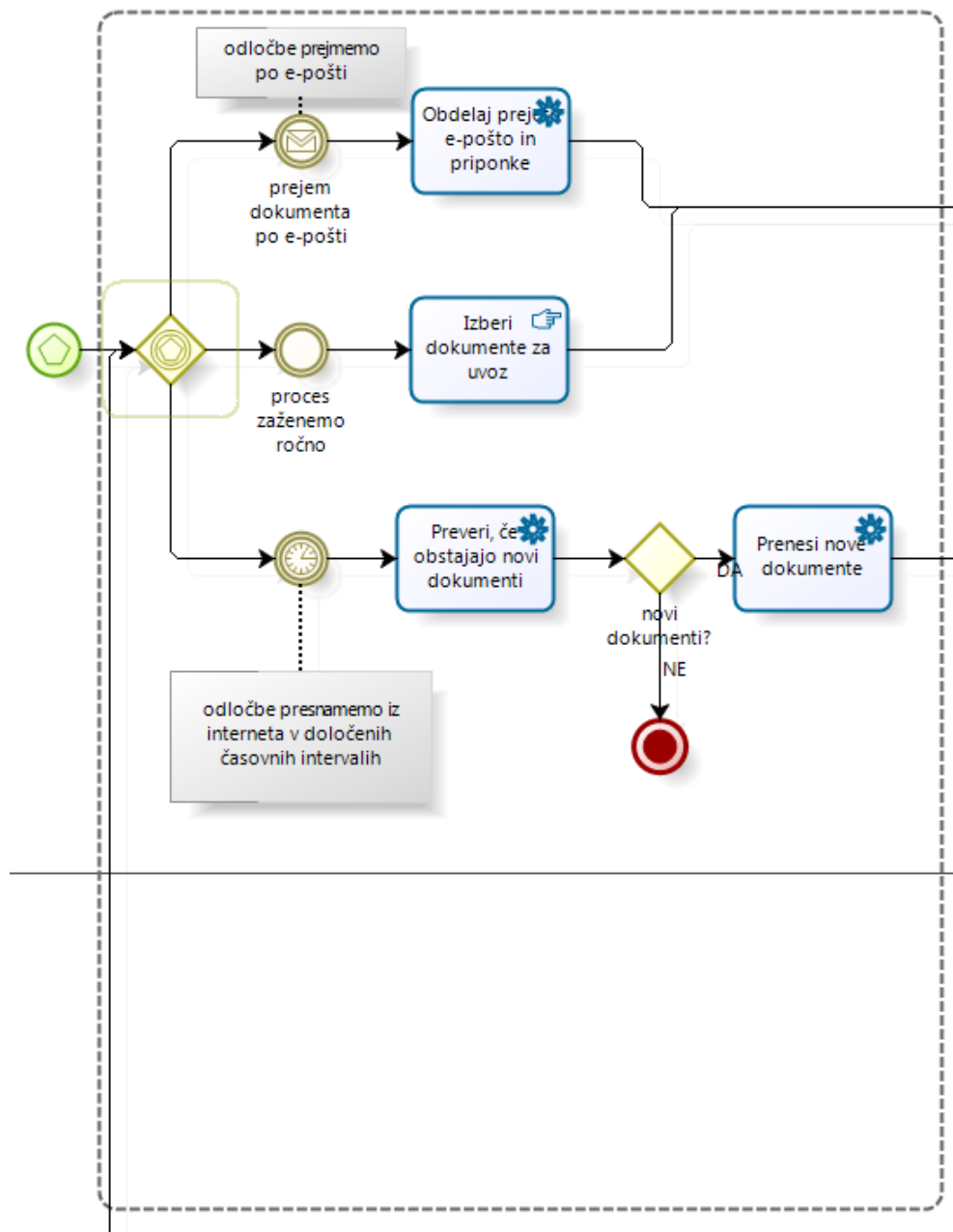
Zajem dokumentov je prvi korak novega procesa za obdelavo dokumentov sodne prakse. Dokumente lahko prejmemo iz različnih virov. Način zajema iz dveh virov je avtomatiziran. Gre za primer, ko prejmemo odločbe po elektronski pošti, ali ko jih prenesemo s spleta. Ta dva vira sta najbolj pogosto uporabljena in tudi dovolj formalizirana, da jih je mogoče avtomatizirati.

Vsi ostali dokumenti, ki jih zajamemo preko drugih virov, se vnesejo ročno, zatem pa se tudi proces zažene ročno.

Pri samodejnem zajemu dokumentov operacija zajema samodejno začne proces. Kot je razvidno s slike 2.8, se v tej fazi izvede samo zajem dokumentov (branje e-pošte, pretakanje dokumentov s spleta, ročni vnos). Zatem se dokumenti v surovi obliki (as is) shranijo v podatkovno bazo IUS STAGE, kjer so tudi arhivirani.



Slika 2.7: Konceptualni pogled na proces obdelave dokumentov sodne prakse



Slika 2.8: Proces obdelave dokumentov sodne prakse, faza 'Zajem dokumentov in tehnična pretvorba'

2.2.2 Priprava dokumentov na uvoz v IUS DOCUMENT

Druga faza procesa za obdelavo dokumentov sodne prakse je priprava dokumentov za uvoz v podatkovno bazo IUS DOCUMENT. Izsek procesa, ki opisuje fazo, prikazuje slika 2.9.

Aktivnosti procesa se razlikujejo glede na način zajema dokumenta:

Priprava dokumentov, prejetih po e-pošti ali ročno vnešenih

Ko so prejeti dokumenti shranjeni v podatkovni bazi IUS STAGE, se preveri pravilnost strukture dokumentov (ali ima vsa obvezna polja).

V primeru pomanjkljive strukture, se dokumente zbere in pošlje v pregled sodišču, proces pa se zaključi (terminira). To popravljene dokumente ponovno pošlje po e-pošti in proces se začne znova.

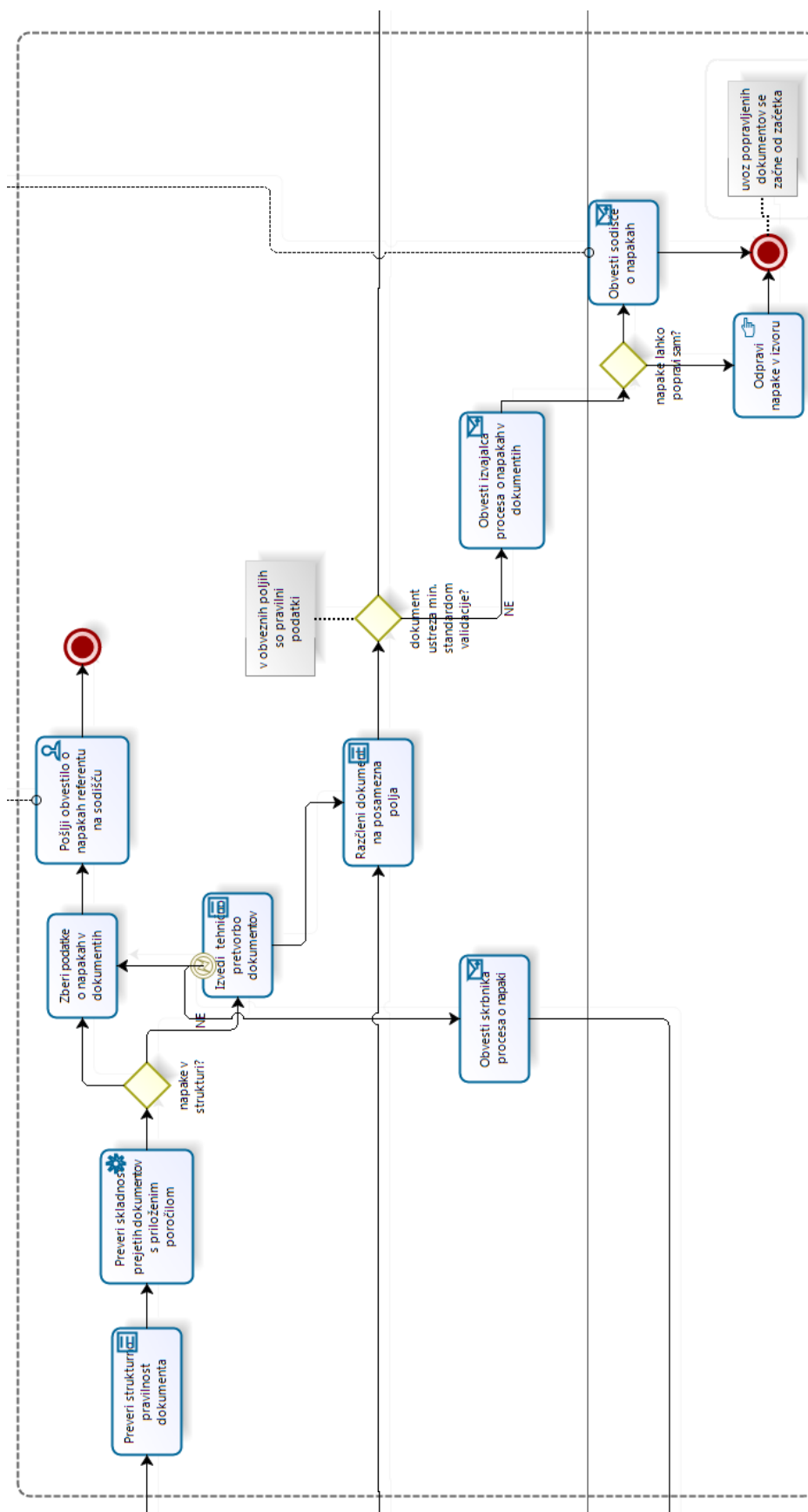
Temu sledi tehnična pretvorba dokumentov. Pri tem se pretvori posebne znake v format, ki je primerem za prenos dokumentov v podatkovno bazo IUS DOCUMENT. Izvede se tudi preimenovanje polj v dokumentu, spreminjanje formata dokumenta (UTF-8) in (po potrebi) združevanje polj.

Priprava dokumentov, prenešenih s spleta

Dokumentov, ki jih prenesemo iz spleta, ni treba posebej obdelati, saj smo že med prenosom poskrbeli za pravilnost polj in formatiranje dokumenta.

Po končani tehnični pretvorbi se zaženejo operacije za razčlenjevanje dokumenta. Pri tem se samodejno preveri, ali vsebuje napake. Samodejno preverjanje deluje na osnovi definicije minimalnega standarda, ki ga mora dokument izpolnjevati, da se ga lahko prenese v podatkovno bazo IUS DOCUMENT.

Ko pride do sintaktičnih napak, se jih odpravi, v kolikor pa gre za vsebinske napake, je treba o tem obvestiti sodišče. Dokument se v tem primeru zavrže. Sodišče ponovno pošlje popravljen dokument in proces se začne znova.



Slika 2.9: Faza priprave dokumentov na uvoz v PB IUS DOCUMENT in IUS LINK

2.2.3 Urejanje in dopolnitev vsebine dokumentov

Urejanje in dopolnitev vsebine dokumentov je najpomembnejša faza v procesu obdelave dokumentov sodne prakse. V tej fazi se preveri vsebino dokumentov in povezave ter uvrstitev vrednosti polj v šifrante. Ta faza je prva, v kateri se aktivnosti razlikujejo glede na vrsto dokumenta, ki ga uvažamo. Pri vseh ostalih fazah so aktivnosti enake za vse vrste dokumentov.

Takoj po uvozu dokumentov v podatkovno bazo IUS DOCUMENT se izvede še nadaljnje samodejno razčlenjevanje dokumentov. Po končani aktivnosti sistem izdela poročilo s seznamom dokumentov, ki so bili uspešno razčlenjeni in čakajo na ročni pregled vsebine. Poročilo vsebuje tudi seznam dokumentov, pri katerih samodejno razčlenjevanje ni uspelo. Te dokumente lahko uredniki ročno popravijo in spet predajo avtomatiki, da ponovno poskusi razčleniti dokument.

Po končanem razčlenjevanju, vzporedno z izdelavo poročila, sistem začne izvajati aktivnosti v fazi '**Gradnja in preverjanje povezav**'.

Pri gradnji povezav se povezave s posebnimi algoritmi (njihove vsebine ne morem razkriti, saj so avtorska last podjetja IUS SOFTWARE) izdelajo, podatki o njih pa se pripravijo za uvoz v podatkovno bazo IUS LINK. Ko so podatki o povezavah dovolj zreli za uvoz, sej jih prenese. Izdela se poročilo o uvozu, ki se ga pošlje izvajalcu procesa. Povezave, ki niso zrele za uvoz, pregleda urednik, jih dopolni in jih šele nato prenese v podatkovno bazo IUS LINK.

Med pregledom vsebine dokumentov lahko uredniki naletijo na vsebinske napake. Zberejo jih in jih pošljejo sodišču v pregled. Obdelava dokumentov se ustavi, dokler sodišče ne pošlje popravkov. Dokument se vrne v obdelavo in napake se popravijo.

V vsebinski pregled dokumentov sodi tudi pregled samodejno zgrajenih povezav na druge dokumente. Če pri nekaterih povezavah sistem ne uspe zgraditi povezave - na primer, da je na voljo več možnosti (zakoni, ki imajo več verzij), na to opozori urednika vsebine. Hkrati mu sistem ponudi tudi seznam povezav, za katere je ugotovil, da so možni kandidati.

Ob koncu obdelave vsebine uredniki določijo zrelost dokumenta za objavo. Včasih se lahko zgodi, da je potrebno počakati nekatere druge dokumente, ali pa preveriti podatke, preden bo dokument primeren za objavo. Vsi dokumenti, ki so označeni kot primerni za objavo, se v naslednji fazi prenesejo v podatkovno skladišče.

Aktivnosti procesa v tej fazi so prikazane na sliki 2.10.

2.2.4 Prenos dokumentov v podatkovna skladišča

V zadnji fazi procesa se dokumenti, ki so bili označeni kot primerni za objavo z ETL procesi, prenesejo v podatkovno skladišče. ETL procesi zberejo podatke iz vseh podatkovnih baz v sklopu IUS TIME.

Po končanem prenosu podatkov v podatkovno skladišče, sistem zgradi poročilo o uvozu dokumentov in o dogajanju v času izvajanja procesa. Poročilo vsebuje:

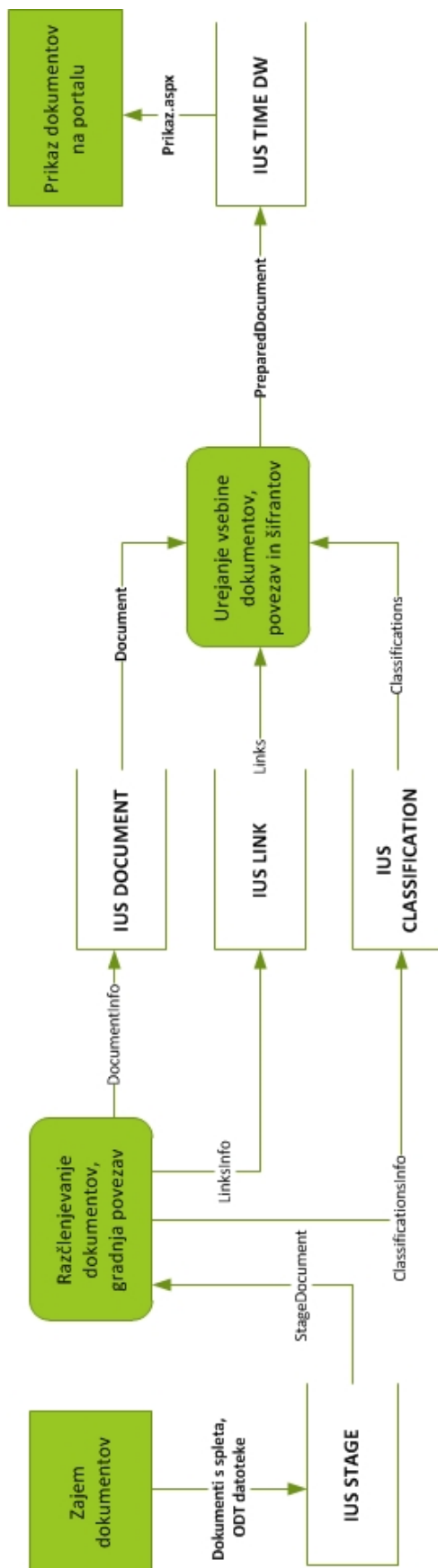
- število dokumentov, zajetih na viru, in koliko jih je bilo prenešenih v podatkovno skladišče,
- kateri dokumenti so bili prenešeni,
- kateri dokumenti niso bili prenešeni ter vzroke za to,
- podrobni podatki o posameznih dokumentih.

Poročilo se pošlje izvajalcu procesa in skrbniku procesa.

Če se med izvajanjem procesa pojavijo napake, je o tem obveščen oddelk IT, ki skrbi za podporo procesov. Skrbnik procesa v oddelku IT poskusi odpraviti napako, v primeru, ko je napaka hujša, pa o tem obvesti oddelek razvoja.

Tok Podatkov

Končni tok podatkov v novem procesu z upoštevanjem nove zgradbe sistema IUS CONTENT je prikazan na sliki 2.11.



Slika 2.11: Prikaz toka podatkov v novem procesu in sistemu IUS CONTENT

Poglavje 3

Sistem za upravljanje procesov

Sistem za upravljanje procesov nudi podporo izvajanju in nadziranju izvajanja procesov v podjetju IUS SOFTWARE. Zasnovan je na osnovi procesa, ki je opisan v poglavju 2.2. Predstavlja dovolj dobro osnovo za nadaljnje razširitve na ostale procese.

V poglavju 3.1 je kratek opis sistema, funkcionalnosti, ki jih mora podpirati, in seznam funkcionalnih zahtev, ki sem jih moral upoštevati pri načrtovanju sistema. Postopek načrtovanja sistema je opisan v poglavju 3.2.

Za prikaz delovanja sistema sem izdelal prototip sistema. Uporabljena tehnologija, koncepti in postopek izdelave so opisani v poglavju 3.3.

V poglavju 3.4 so predstavljene možne razširitve sistema in potencial trenutno razpoložljivih komercialnih rešitev, ki bi jih podjetje lahko uporabilo namesto načrtovanega sistema. Njihova primernost je kritično ocenjena.

3.1 Opis sistema

Sistem za upravljanje procesov je zasnovan na podlagi prenovljenega procesa za obdelavo dokumentov sodne prakse, ki je opisan v poglavju 2.2. Razvit je popolnoma na novo in ne nadomešča nobenega sistema. To je tudi osrednji razlog za njegov razvoj, saj podjetje trenutno nima sistema, ki bi omogočal nadzorovanje procesov, natančen vpogled v njihovo izvajanje in primeren nadzor nad kvaliteto vsebine.

Funkcionalne zahteve

Med opravljanjem intervjujev z udeleženci v procesih sem poleg zahtev za preno-ovo procesov zajel tudi funkcionalne zahteve, ki jih mora novi sistem podpirati.

Vse funkcionalne zahteve izražajo potrebe, ki so se pokazale v času izvajanja obstoječih procesov in do zdaj niso bile implementirane.

Nekaj glavnih funkcionalnosti v novem sistemu:

- avtomatizacija izvajanja določenih aktivnosti,
- upravljanje procesa (začetek izvajanja, začasna zaustavitev, prekinitev izvajanja, dodelitev aktivnosti, ...) preko spletne storitve,
- nadzor nad potekom procesa (Business Activity Monitoring),
- dodeljevanje aktivnosti posameznim udeležencem v procesu,
- sledljivost dokumentov (iz katerih datotek v katerem paketu prihaja dokument),
- pregledovanje zgodovine izvajanja procesov,
- samodejno generiranje poročil o prenosu dokumentov in o poteku procesa.

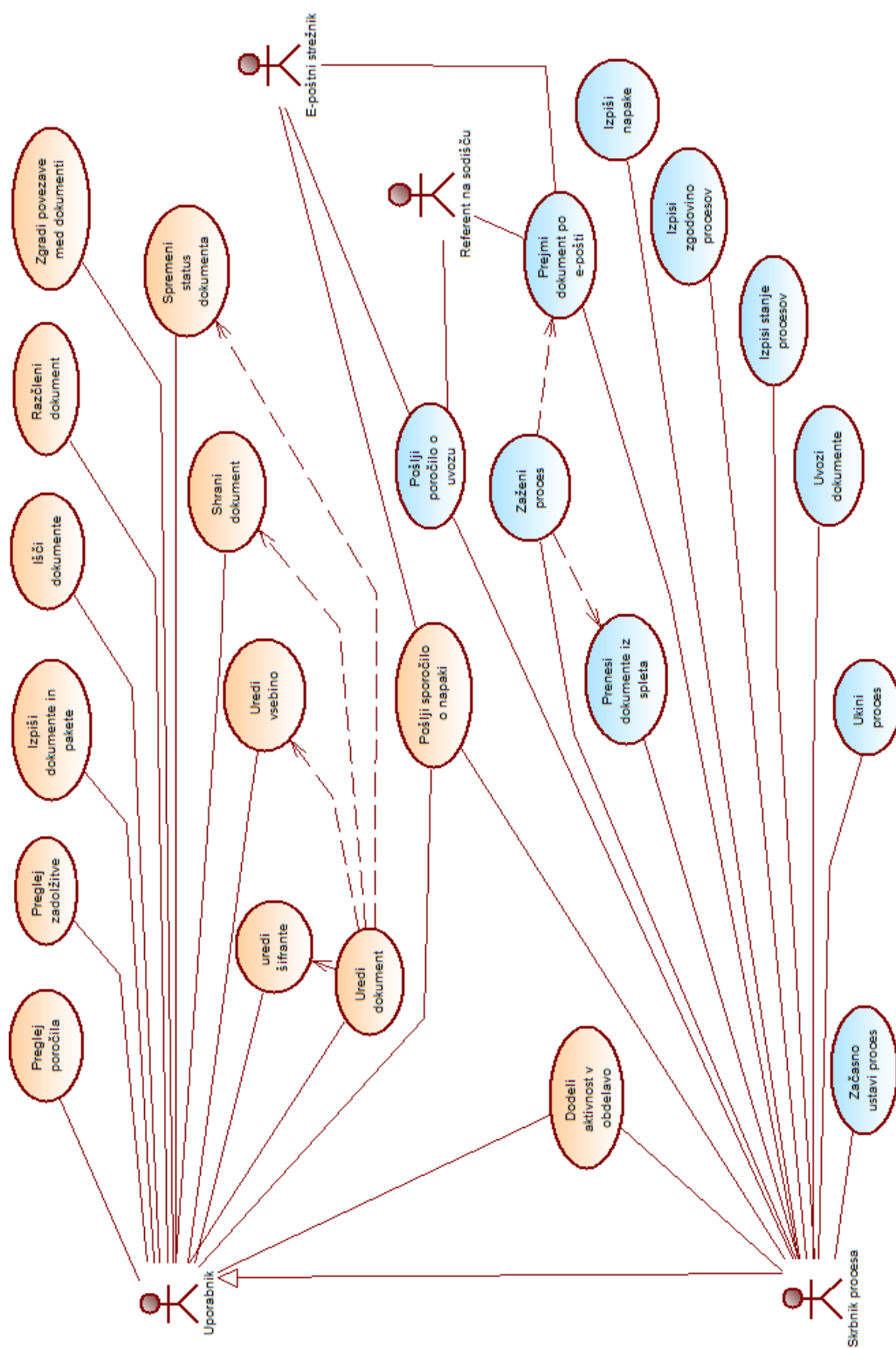
Možni primeri uporabe sistema za upravljanje procesov so prikazani v diagramu na sliki 3.1. Sistem uporabnikom omogoča spremljanje izvajanja procesa v spletnem vmesniku, kjer so jim na voljo tudi vse funkcije za njegovo upravljanje.

V spletnem vmesniku so na voljo tudi različni pregledi podatkov o delovanju procesov.

3.2 Arhitektura sistema in načrtovanje

Arhitektura sistema je podobna storitveno usmerjeni arhitekturi. Sledi namreč nekaterim smernicam storitveno usmerjene arhitekture [1, 2], ki so:

- šibka sklopljenost,
- interoperabilnost,
- ponovna uporabljivost,
- možnost odkritja in
- upravljanje.



Slika 3.1: Diagram primerov uporabe sistema za upravljanje procesov

Arhitektura sistema za upravljanje procesov upošteva prve tri smernice. Šibka sklopljenost je vidna pri aplikacijah, ki podatke o procesu dobivajo preko klicev spletne storitve za orkestracijo procesov. Ta storitev ima vnaprej definiran vmesnik. Vse aplikacije, ki delajo s sistemom, so popolnoma neodvisne od sprememb na nižjih plasteh.

Spletna storitev za orkestracijo procesa je interoperabilna, saj za komunikacijo uporablja SOAP, ki je standardiziran in neodvisen od platforme.

Tako spletna storitev za orkestracijo procesa kot podporne spletne storitve, so ponovno uporabljive, saj jih lahko uporabimo pri implementaciji logike več različnih procesov.

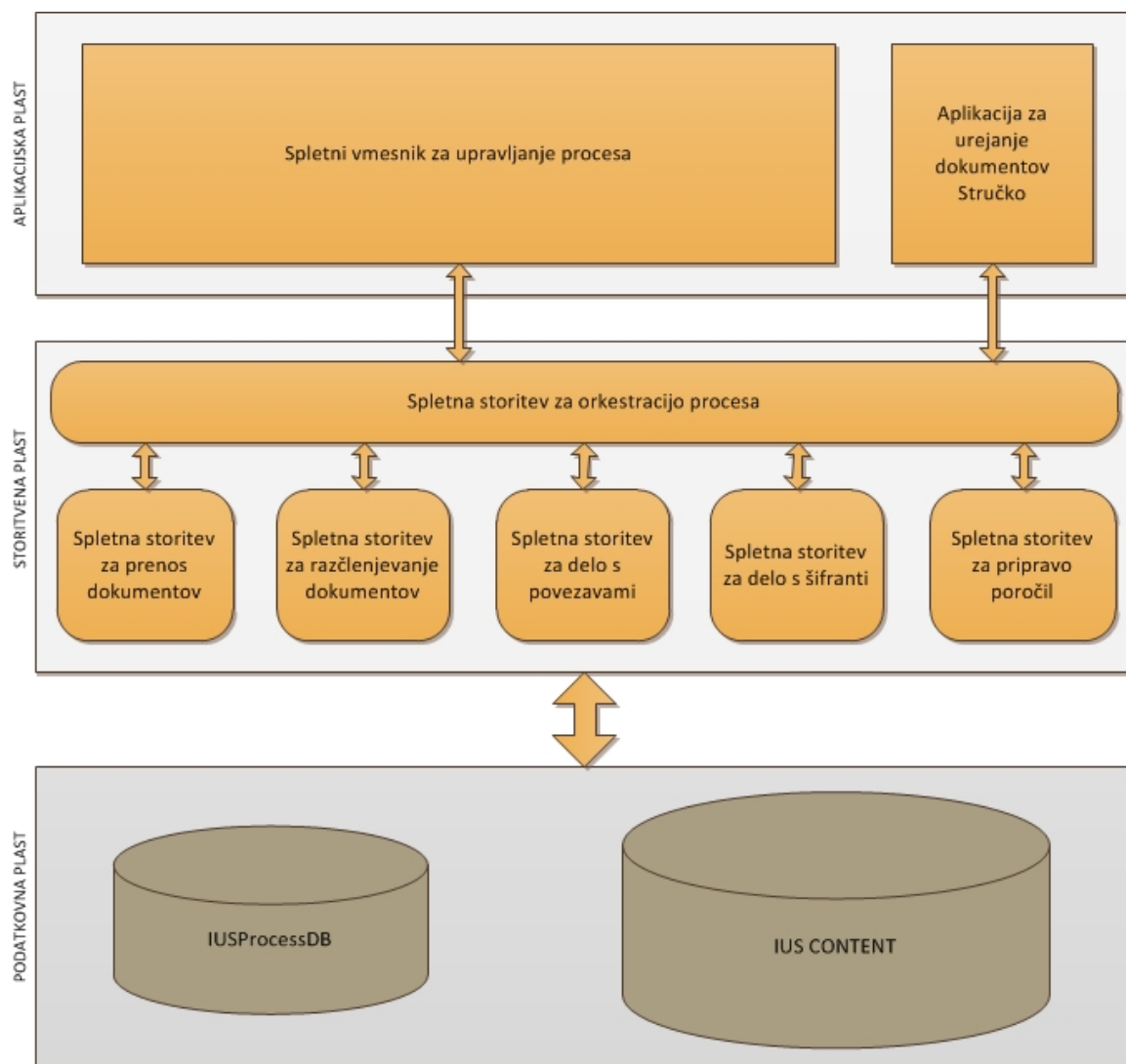
Trenutno je stanje pred vzpostavitvijo novega sistema tako, da logika procesa teče v posameznih aplikacijah na računalnikih uporabnikov. V primerjavi s prejšnjim stanjem se zdaj večina aktivnosti procesa izvaja samodejno na strežniku. S takim pristopom sem odpravil potrebo po namestitvi programov na vse računalnike.

Vmesnik za upravljanje procesa je sedaj dostopen preko brskalnika. To olajša upravljanje procesov, kot tudi postopek vzdrževanja vmesnika, saj ga je potrebno namestiti in kasneje posodabljati samo na enem mestu.

Arhitektura sistema je sestavljena iz treh plasti:

- **aplikacijska plast**, ki vsebuje aplikacije z uporabniškim vmesnikom, preko katerih je možno upravljati proces,
- **storitvena plast** s spletnimi storitvami za orkestracijo procesa in podpornimi storitvami, ki podpirajo izvajanje posameznih aktivnosti v procesu,
- **podatkovna plast**, v kateri se nahajajo podatkovna baza IUSProcesDB in vmesniki za dostop do sistema IUS CONTENT.

Arhitekturo sistema prikazuje slika 3.2. Sistem za upravljanje procesov sem zasnoval tako, da ga je možno razširiti na druge procese brez večjih sprememb v arhitekturi. Razširitev se lahko izvede tako, da se za posamezne procese doda nove spletne storitve za orkestracijo procesa. Tako vsak tip procesa uporablja svojo storitev.



Slika 3.2: Arhitektura sistema za upravljanje procesov

3.2.1 Aplikacijska plast

Aplikacijska plast vsebuje aplikacije, ki preko komuniciranja s spletno storitvijo za orkestracijo procesa upravljajo proces. To so lahko aplikacije z uporabniškim vmesnikom, ki nudijo boljšo uporabniško izkušnjo, ali pa so to skripte in druge storitve, ki kličejo omenjeno storitev za orkestracijo procesa (na primer storitev, ki prejema dokumente po elektronski pošti in jih shranjuje v podatkovno bazo, hkrati pa tudi sproži proces).

V to plast so vključene tudi aplikacije, ki niso del sistema, vendar s sistemom komunicirajo. Primer take aplikacije je urejevalnik dokumentov *Stručko*.

Aplikacije so lahko neodvisne od platforme, saj uporabljajo standardizirano spletno storitev. S tem se odpre možnost uporabe različnih operacijskih sistemov in tudi možnost uporabe spletnih tehnologij. Primer spletne aplikacije za upravljanje procesa je predstavljen v poglavju 3.3 pri implementaciji prototipa.

3.2.2 Storitvena plast

Storitvena plast vsebuje dva tipa storitev, glede na model storitve [2]:

- *spletna storitev za orkestracijo procesa* (poslovna storitev) in
- *podporna spletna storitev*.

Spletna storitev za orkestracijo procesa ima vnaprej določen vmesnik (interface). Vsebuje funkcije za izvajanje in upravljanje procesa. Za vsak proces, ki ga sistem podpira, lahko teče ločena spletna storitev za orkestracijo.

Pri orkestraciji centralno nameščena logika (v tem primeru spletna storitev za orkestracijo procesa) nadzira izvajanje in usmerja komunikacijo med več (interoperabilnimi) aplikacijami na različnih nivojih. [2] Orkestracija zmanjša kompleksnost sistema, saj je poslovna logika zbrana na enem mestu, ne pa zapisana v posameznih aplikacijah.

Poleg tega vsebuje tudi operacije, preko katerih lahko aplikacije dostopajo do podatkov v podatkovni bazi IUSProcesDB. Zaradi tega je uporabniški vmesnik lahko načrtovan in izdelan popolnoma neodvisno od storitve. Hkrati ni treba popravljati vseh aplikacij vsakokrat, ko se spremeni shema podatkovne baze, saj uporabljajo storitev, ki ima, kot rečeno, vnaprej določen vmesnik.

Spletna storitev za orkestracijo procesa nudi vse funkcije za upravljanje procesa. Te so:

- začetek nove instance procesa,
- začasna zaustavitev procesa,

- nadaljevanje procesa,
- ukinitvev procesa (terminate),
- dodeljevanje aktivnosti v obdelavo,
- prevzem aktivnosti,
- zaključek ročne aktivnosti.

Kot rečeno, storitev nudi tudi funkcije za izpis podatkov iz podatkovne baze IusProcesDB, na primer:

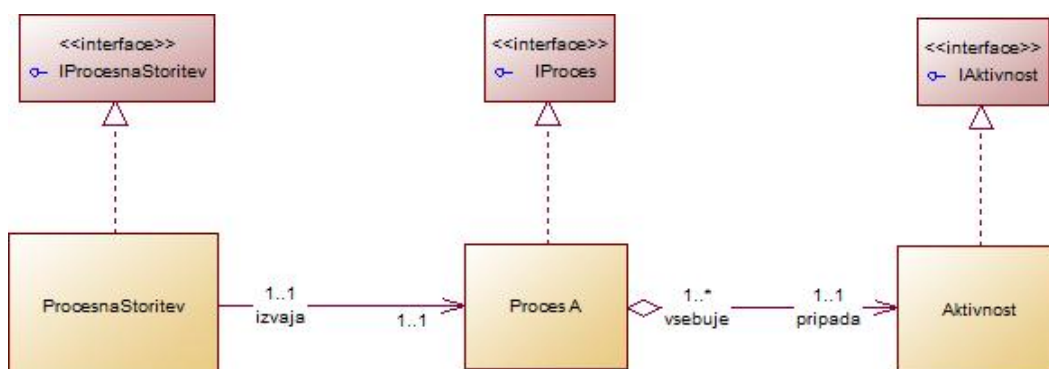
- seznam aktivnih procesov,
- seznam čakajočih aktivnosti,
- podrobnosti posameznega procesa, kar vključuje:
 - metapodatke o procesu,
 - končane aktivnosti,
 - izdelana poročila,
 - seznam obdelanih paketov in
 - seznam obdelanih dokumentov,
- seznam poročil,
- avtentikacijo uporabnikov
- in druge.

Storitev za orkestracijo procesa deluje tako, da upravlja z objektom, ki predstavlja proces. Ta objekt ima standardni vmesnik. To omogoča, da lahko vsako storitev priredimo za izvajanje različnih procesov.

Povezanost storitve in objekta procesa je prikazana na sliki 3.3. Tak način ločitve procesa od storitve, torej, da je logika procesa prenešana v objekt, omogoča razvoj in vzdrževanje procesov neodvisno od razvoja in vzdrževanja storitve.

Objekt vsebuje metode za izvajanje procesa in njegovih aktivnosti ter za beleženje izvajanja.

Ker je celotna logika in delno stanje procesa shranjeno v objektu, to omogoča, da celoten proces prekinemo in shranimo, kar je še posebej pomembno, če so v procesu ročne aktivnosti, ki se ne izvajajo v realnem času.



Slika 3.3: Prikaz povezanost storitve za orkestracijo procesa s procesni objektom in njegovimi aktivnostmi

Ko se na primer ročna aktivnost konča, procesni objekt ponovno naložimo in nadaljujemo z izvajanjem procesa.

Podporne spletne storitve nudijo spletni storitvi za orkestracijo procesa podporo pri izvajanju aktivnosti.

Vsaka od šestih podpornih spletnih storitev pokriva posamezno področje. Po potrebi se lahko dodajo še druge spletne storitve, tudi zunanje.

Podporne spletne storitve:

- *spletna storitev za prenos dokumentov,*
- *spletna storitev za razčlenjevanje dokumentov,*
- *spletna storitev za delo s povezavami,*
- *spletna storitev za delo s šifranti,*
- *spletna storitev za pripravo poročil.*

Posamezno podporno storitev lahko uporabimo pri več različnih procesih. Ker posamezni procesi komunicirajo s temi storitvami preko vnaprej določenih vmesnikov, je vzdrževanje in njihovo nadaljnje razvijanje enostavno, saj ob spremembah ni potrebno popravljati vseh procesov, ki te storitve uporabljajo.

3.2.3 Podatkovna plast

Podatkovno plast sestavljajo podatkovna baza IUSProcesDB in vmesniki za dostop do sistema IUS CONTENT. Največkrat se do tega sistema oziroma sklopa podatkovnih baz IUS TIME dostopa kar preko aplikacijskega strežnika.

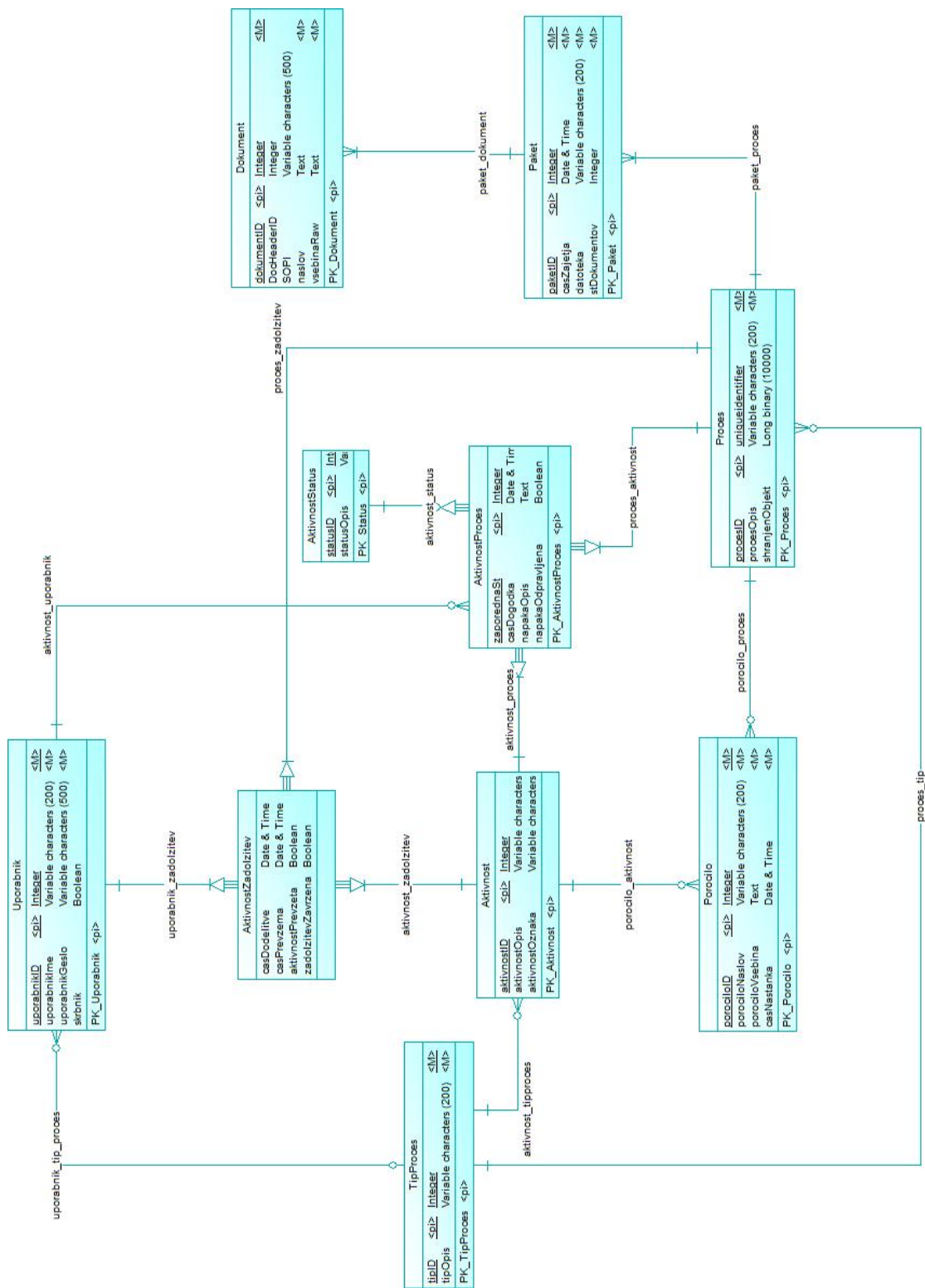
Podatkovna baza IUSProcesDB

Podatkovna baza IUSProcesDB podpira delovanje sistema za upravljanje procesov. V njej se shranjujejo podatki o prejetih dokumentih in paketih dokumentov, zagnanih procesih in njihovih aktivnostih ter o stanju teh procesov.

Glavni del podatkovne baze je relacija *AktivnostProces*. Ta relacija služi kot dnevnik, saj se vanjo zapisujejo podatki o začetku in koncu posameznih aktivnosti ter o morebitnih napakah, ki so nastale pri izvajanju aktivnosti, in drugih stanjih.

Do podatkovne baze dostopajo samo spletne storitve za orkestracijo procesa. Vse aplikacije do podatkov v podatkovni bazi dostopajo s pomočjo teh spletnih storitev in se nikoli neposredno ne povezujejo s podatkovno bazo. Zaradi tega je lažje vpeljati morebitne spremembe v podatkovni bazi, saj je treba popraviti ali dopolniti samo spletno storitev, ne pa tudi vseh aplikacij.

Strukturo podatkovne baze IUSProcesDB prikazuje konceptualni model na sliki 3.4.



Slika 3.4: Konceptualni model relacijske podatkovne baze IUSProcesDB

Relacijska shema

TipProces (tipID, tipOpis)

Hrani podatke o tipih procesov, ki jih sistem podpira.

Proces (procesID, procesOpis, shranjenObjekt, #tipID)

Hrani seznam vseh procesov, ki so se kadarkoli izvajali ali pa se še izvajajo v sistemu. V polje shranjenObjekt se shranjuje objekt, ki vsebuje proces, ob vsaki začasni zaustavitvi procesa in ob izvajanju ročnih aktivnosti (Human tasks).

Paket (paketID, #procesID, casZajetja, datoteka, stDokumentov)

Vsebuje podatke o paketih dokumentov, ki so prispeli v sistem. To omogoča, da skrbnik procesa lahko določi, iz katerega paketa izvira določen dokument, kdaj smo ga prejeli, in podobno.

Dokument (dokumentID, DocHeaderID, SOPI, naslov, vsebinaRaw, #paketID)

Hrani podatke o dokumentih, ki so bili naloženi v sistem s procesi za obdelavo dokumentov in s katerimi paketi so prišli. To je zelo pomembna tabela za sledljivost.

Aktivnost (aktivnostID, aktivnostOpis, aktivnostOznaka, #tipID)

Vsebuje seznam aktivnosti za vsak tip procesa. Seznam aktivnosti je treba redno posodabljati, da ustreza dejanskemu stanju procesov.

AktivnostStatus (statusID, statusOpis)

Šifrant, ki vsebuje vsa možna stanja, v katerih lahko aktivnosti v procesih nastopajo.

Privzeto so definirana naslednja stanja:

- v izvajanju,
- uspešno končano,
- končano z napako,
- čaka na dodelitev,
- čaka na izvedbo,
- začasno prekinjen in
- ukinjen.

AktivnostProces (zaporednaSt, #procesID, #aktivnostID, #statusID, casDogodka, napakaOpis, napakaOdpravljena, #uporabnikID)

Glavna relacija v podatkovni bazi IUSProcesDB. Vsebuje dnevnik vseh dogodkov, ki so se zgodili med izvajanjem procesov. Hrani podatke o prehodih med aktivnostmi, času prehodov in morebitnih napakah, ki so se zgodile med njihovim izvajanjem.

Beleži tudi napake in njihovo stanje – če so že odpravljene, ali ne.

AktivnostZadolžitev (#procesID,#aktivnostID,#uporabnikID, casDodelitve, casPrezema, aktivnostPrezeta, zadolžitevZavrzena)

V to relacijo se zapisujejo podatki o dodeljenih aktivnostih posameznim uporabnikom – zadolžitvah za izvajanje aktivnosti. Beleži se tudi, ali je bila zadolžitev prevzeta v izvajanje, in kdaj se je to zgodilo.

Uporabnik (uporabnikID, uporabnikIme, uporabnikGeslo, skrbnik, #tipID)

Hrani podatke o uporabnikih, ki uporabljajo sistem. Podatki so pomembni tako pri dodeljevanju posameznih aktivnosti udeležencem v procesih, kot tudi za beleženje, kdo je izvajal določeno aktivnost, in kdaj.

Poročilo (porociloID, porociloNaslov, porociloVsebina, casNastanka,#procesID, #aktivnostID)

Relacija vsebuje podatke o poročilih, ki so bila zgrajena med izvajanjem procesa v posameznih aktivnostih, in njihovo vsebino.

3.3 Implementacija prototipa sistema

Za prikaz delovanja sistema sem izdelal enostaven prototip. Proces in njegove aktivnosti niso realizirane v celoti, ampak zgolj toliko, da prikažejo delovanje.

Izdelal sem spletno storitev za orkestracijo procesa, proces in njegove aktivnosti in spletni vmesnik za upravljanje sistema.

3.3.1 Tehnologije

Prototip v celoti temelji na Microsoftovih tehnologijah. Uporabil sem programski jezik C#, skupaj z ogrodjem Microsoft .NET 4. Spletna storitev uporablja

Windows Communication Foundation, ki je del .NET ogrodja in nudi zelo dobro podporo razvoju in uporabi spletnih storitev vseh vrst.

Spletne storitve gostujejo na strežniku Microsoft IIS 7.5, uporabljajo pa tudi dodatek za IIS, AppFabric, ki omogoča lažje upravljanje in gostovanje storitev. Več o AppFabric v nadaljevanju.

Za izdelavo spletnega vmesnika sem uporabil ASP.NET C#, spletni vmesnik pa prav tako gostuje na strežniku Microsoft IIS 7.5.

Tudi sistem za upravljanje podatkovne baze je Microsoftov, in sicer gre za SQL Server 2008.

Za uporabo Microsoftovih produktov sem se odločil zato, ker razvoj v podjetju poteka na tej platformi in je zelo verjetno, da bo na njej izdelan tudi končni sistem.

Uporaba enake tehnologije na vseh plasteh prinaša veliko prednosti. Tehnologije na isti platformi so optimizirane za povezovanje med seboj, hkrati pa je možna uporaba kompleksnih podatkovnih tipov (v spletnih storitvah). To v primeru delovanja na različnih platformah ne bi bilo tako enostavno, ali pa bi bilo celo nemogoče, s čimer bi se povečala kompleksnost same implementacije.

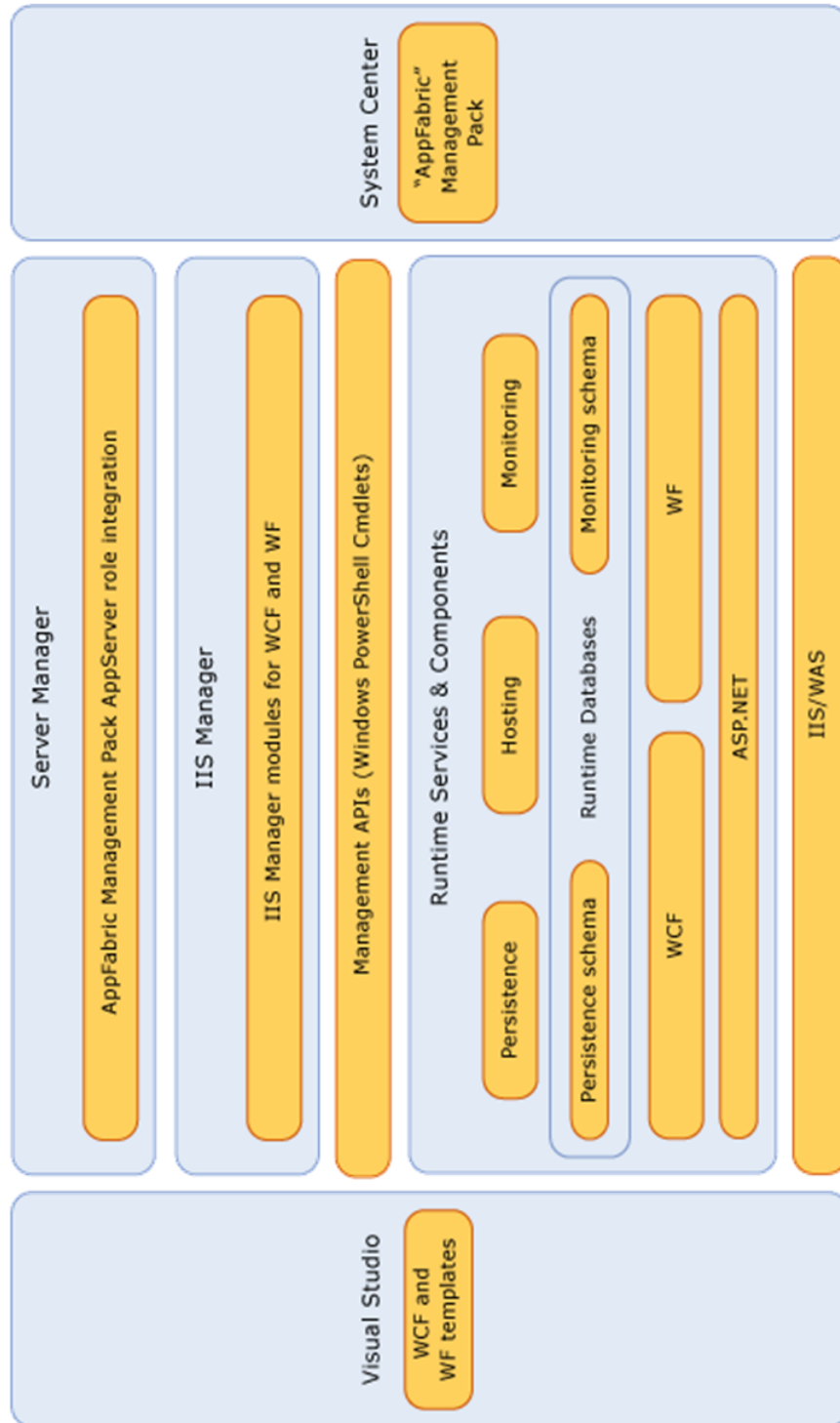
Windows Server AppFabric

AppFabric je skupek dodatkov za Windows Server, ki razvijalcem omogoča, da lažje upravljajo spletne storitve, hkrati pa ponuja dodatne možnosti kot so predpomnenje, avtomatski zagon storitev in druge [8]. Razdeljen je na dva dela:

- *AppFabric Caching Services* – storitve za predpomnenje
- *AppFabric Hosting Services* – storitve za upravljanje in gostovanje spletnih storitev, izdelanih z WCF.

Opisal bom samo drugi del, ki sem ga tudi uporabil pri implementaciji prototipa.

AppFabric Hosting Services dopolnjuje obstoječo infrastrukturo za gostovanje, ki jo nudita IIS in Windows Process Activation Service (WAS).

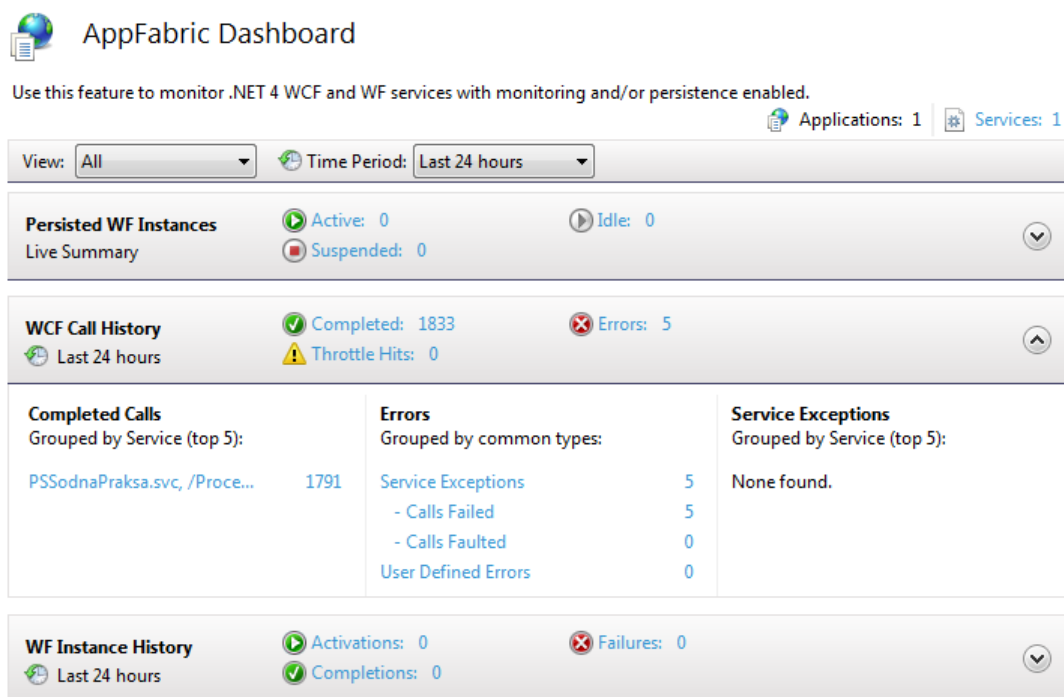


Slika 3.5: Prikaz dopolnitve IIS in WAS z AppFabric

Kot je prikazano na sliki 3.5, WCF spletne storitve tečejo v IIS in ne v AppFabric-u. Ta obstoječi infrastrukturi IIS in WAS dodaja funkcionalnosti, kot je na primer samodejni zagon spletne storitve takoj, ko je naložena na strežnik, s čimer se izognemo nadležnemu čakanju ob prvem klicu storitve. AppFabric storitev, ki dlje časa ni uporabljena, odstrani iz pomnilnika in shrani v podatkovno bazo, s tem pa daje infrastrukturi večjo skalabilnost. Ko je storitev spet klicana, jo naloži nazaj v pomnilnik in se odzove na klic.

Glavna prednost AppFabrica pa je nadzorna plošča (slika 3.6), ki je na voljo administratorju. Na nadzorni plošči so zbrani podrobni podatki o vseh klicih posameznih storitev in morebitnih napakah.

AppFabric omogoča tudi podrobno opazovanje delovanja storitev, tako imenovani monitoring. To je še posebej uporabno med razvojem storitev, pa tudi med samim delovanjem za identifikacijo in odpravo napak.



Slika 3.6: Nadzorna plošča AppFabric

3.3.2 Implementacija spletne storitve za orkestracijo procesa

Spletna storitev za orkestracijo procesa je realizirana z WCF. Razred, ki realizira spletno storitev, implementira vmesnik *IProcessService*:

```
[ServiceContract]
public interface IProcessService

    [OperationContract]
    Guid ZazeniProces(string uporabniskoIme, string oznakaProces, int datotekaID);

    [OperationContract]
    string VrniStatusProcesa(string uporabniskoIme);

    [OperationContract]
    string UkiniProces(Guid procesID, string uporabniskoIme);

    [OperationContract]
    string NadaljujProces(Guid procesID, string uporabniskoIme);

    [OperationContract]
    string ZacasnoUstaviProces(Guid procesID, string uporabniskoIme);

    [OperationContract]
    string OdpraviNapako(string uporabnik, string aktivnostiOznaka, Guid procesID);

    [OperationContract]
    ProcesZgodovina[] VrniSeznamKoncanihProcesov();

    [OperationContract]
    Proces VrniPodatkeProces(Guid procesID);

    [OperationContract]
    ProcesAktivni[] VrniSeznamAktivnihProcesov();

    [OperationContract]
    Aktivnost[] VrniSeznamAktivnosti();

    [OperationContract]
    Zadolzitev[] VrniSeznamZadolzitevUporabnik(string uporabnikIme);

    [OperationContract]
    Paket[] VrniSeznamPaketov();

    [OperationContract]
    Dokument[] VrniDokumentePaket(int paketID);

    [OperationContract]
    Dokument[] IsciDokumenti(string iskalniNiz);

    [OperationContract]
    string DodeliAktivnostUporabniku(string adminUporabnik, string uporabnikIme,
    string aktivnostOznanka, Guid procesID);

    [OperationContract]
    string RocnaAktivnostDokoncana(string uporabniskoIme, string aktivnostOznanka,
    Guid procesID);

    [OperationContract]
    string PrevzemiAktivnost(string uporabniskoIme, string aktivnostOznanka, Guid procesID);

    [OperationContract]
    Aktivnost[] VrniAktivneAktivnostiUporabnika(string uporabnik);

    [OperationContract]
    Aktivnost[] VrniZgodovinoUporabnika(string uporabnik);

    [OperationContract]
```

```

Aktivnost[] VrniAktivnostiNapake(string uporabnik);

[OperationContract]
Uporabnik[] VrniSeznamUporabnikov();

[OperationContract]
Zadolzitev[] VrniCakajoceAktivnosti();

[OperationContract]
Porocilo[] VrniSeznamPorocil(int porociloID);

[OperationContract]
ProcesAktivni[] VrniSeznamZacasnoUstavljenihProcesov();

[OperationContract]
ProcesAktivni[] VrniSeznamUkinjenihProcesov();

[OperationContract]
bool[] AvtenticirajUporabnika(string uporabniskoIme, string geslo);

```

Spletna storitev je konfigurirana kot *Singleton* objekt. To pomeni, da obstaja samo ena instanca storitve, ki jo kličejo vsi odjemalci, in se ne kreira instanca za vsak klic storitve. Spletna storitev se uniči le ob zaustavitvi strežnika.

Konfiguracija storitve:

```

[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single,
ConcurrencyMode = ConcurrencyMode.Multiple)]

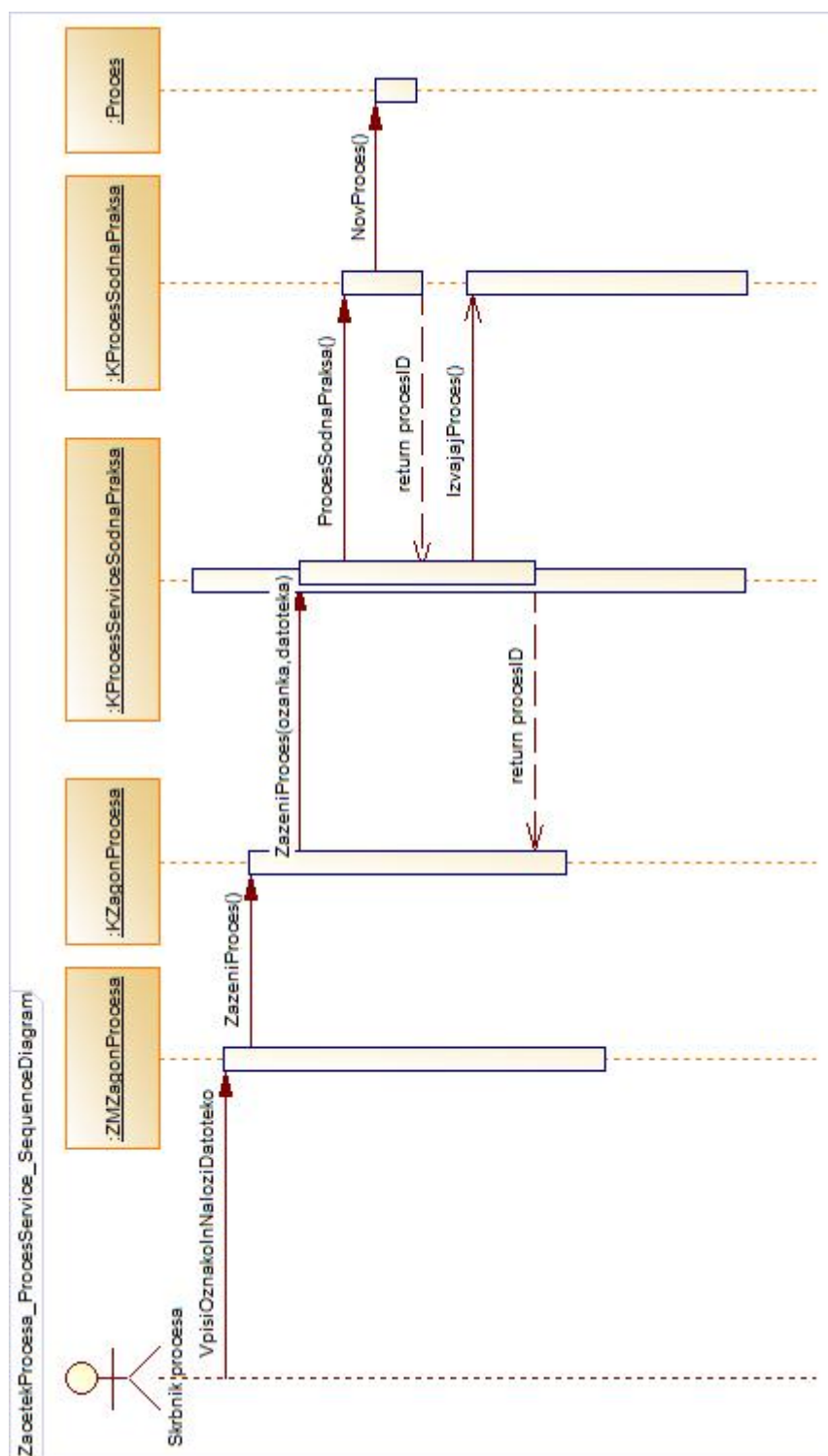
```

Kot kaže konfiguracija, je omogočen hkraten dostop do storitve, kar pomeni, da lahko do objekta dostopa več kot ena nit naenkrat. Zato sem moral poskrbeti za sinhronizacijo objekta.

Proces je v spletni storitvi predstavljen z razredom tipa *ProcesSodnePrakse*.

Ko kličemo metodo *ZazeniProces*, se kreira nova instanca razreda. Po začetnih nastavitvah se asinhrono kliče metoda *IzvajajProces* v razredu *ProcesSodnePrakse*, ki začne izvajati proces. Klic metode se konča s sporočilom uporabniku, proces pa se izvaja naprej v ločeni niti.

Diagram zaporedja, ki prikazuje zagon procesa, je prikazan na sliki 3.7.



Slika 3.7: Diagram zaporedja, ki prikazuje začetek nove instance procesa

V primeru, da želimo proces prekiniti zaradi različnih razlogov, se izvajanje procesa prekine, sam objekt, ki vsebuje proces, pa se serializira in shrani v podatkovno bazo.

Ko želimo proces nadaljevati, storitev serializiran objekt prebere iz podatkovne baze, ga de-serializira in naloži v pomnilnik, nato pa spet asinhrono zažene nadaljevanje izvajanja procesa, klic storitve pa se zaključi s sporočilom klicatelju. Programska koda metode *NadaljuyProces* je vidna na sliki 3.8.

```

/// <summary>
/// Nadaljuje proces, ki je bil začasno ustavljen
/// </summary>
/// <param name="uporabnikskoIme">uporabniško ime uporabnika, ki kliče storitev</param>
/// <returns>string sporočilo o uspehu operacije</returns>
public string NadaljuyProces(Guid procesID, string uporabnikskoIme)
{
    lock (_procesObjLock)
    {
        if (procesObj == null && !ProcesAktiviran)
        {
            ProcesEntity procesObstojecci = (from p in context.Proces where p.procesID == procesID select p).First();
            string shranjenObjekt = procesObstojecci.shranjenObjekt;
            if (shranjenObjekt != null)
            {
                BinaryFormatter bformatter = new BinaryFormatter();
                procesObj = (ProcesSodnaPraksa)bformatter.Deserialize(new MemoryStream(System.Convert.FromBase64String(shranjenObjekt)));

                if (procesObj != null)
                {
                    var query = from u in context.Uporabnik where u.uporabnikIme == uporabnikskoIme select u.uporabnikID;
                    int iTrenutniUporabnik = -1;

                    if (query.Any())
                        iTrenutniUporabnik = query.FirstOrDefault();
                    else
                        throw new UnauthorizedAccessException();

                    procesObj.Prekinjen += new ProcesSodnaPraksa.ProcesEventHandler(procesObj_Prekinjen);
                    procesObj.Napaka += new ProcesSodnaPraksa.ProcesEventHandler(procesObj_Napaka);

                    ipd = new IzvajajProcesDelegate(procesObj.IzvajajProces);

                    ipd.BeginInvoke(iTrenutniUporabnik, new AsyncCallback(CallbackMethod), null);
                    ProcesAktiviran = true;
                    return "proces se nadaljuje!";
                }
                else
                {
                    throw new Exception("Proces se ni pravilno naložil!");
                }
            }
            else
            {
                throw new Exception("Procesa ni možno nadaljevati: ni podatkov o stanju procesa. Lahko je bil ukinjen ali pa je prišlo do napake!");
            }
        }
        else
        {
            throw new Exception("Trenutno teče proces z ID: " + procesID + " in ni mogoče nadaljevati obstoječega procesa!");
        }
    }
}

```

Slika 3.8: Prikaz metode *NadaljuyProces*

Procesni objekt

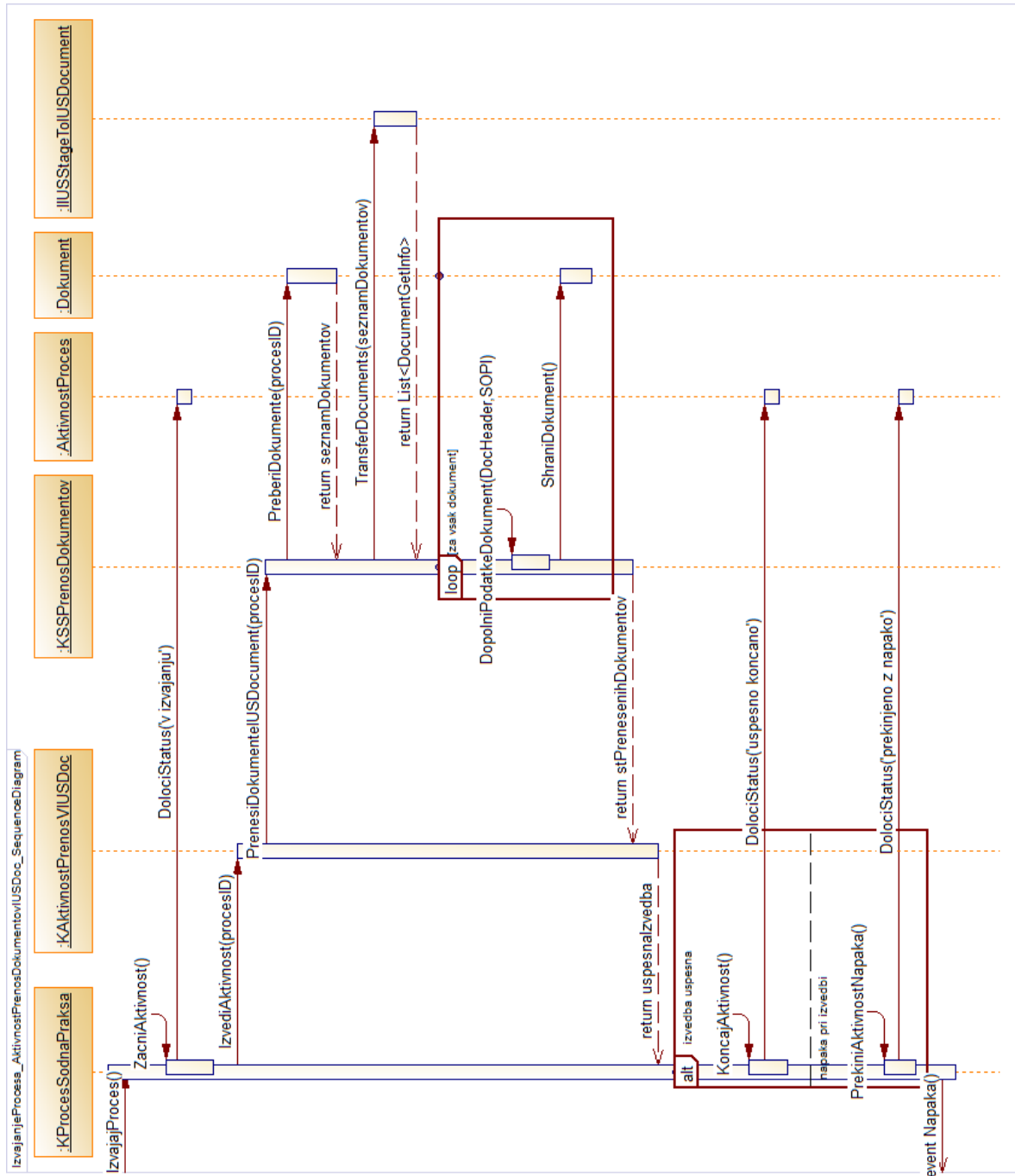
Proces je v sistemu realiziran z razredom `ProcesSodnaPraksa`. Razred implementira vmesnik `IProces`:

```
interface IProces
{
    void Dispose();
    void IzvajajProces(int iUporabnikID);
    void KoncajAktivnost(IAktivnostObject aktivnost, int iUporabnikID);
    event ProcesEventHandler Napaka;
    string Oznaka { get; set; }
    void PrekiniIzvajanje(int iUporabnikID);
    event ProcesEventHandler Prekinjen;
    Guid procesID { get; set; }
    IAKtivnostObject TrenutnaAktivnost { get; }
    void Ukini(int iUporabnikID);
    void UkiniAktivnost(IAktivnostObject aktivnost, int iUporabnikID);
    void ZacasnoPrekiniAktivnost(IAktivnostObject aktivnost, int iUporabnikID);
    void ZacniAktivnost(IAktivnostObject aktivnost, int iUporabnikID);
}
```

Najpomembnejša je metoda `IzvajajProces`. Metoda iterira čez seznam aktivnosti `List<IAktivnostObject> seznamAktivnosti` in pri vsaki aktivnosti kliče njeno metodo `Izvedi()`. Ta metoda vsebuje logiko same aktivnosti, ki je lahko direktno implementirana, ali pa gre za klic podporne spletne storitve oziroma aplikacijskega strežnika.

Razred vsebuje tudi metode, ki skrbijo, da se vsi dogodki beležijo v podatkovno bazo `IUSProcesDB`.

Delovanje metode `IzvajajProces` je prikazano na primeru aktivnosti, ki za izvedbo svoje logike kliče spletno storitev, ta pa za izvedbo uporablja aplikacijski strežnik. Delovanje opisuje diagram zaporedja na sliki 3.9.



Slika 3.9: Diagram zaporedja prikazuje izvajanje procesa

Aktivnost

Vsaka aktivnost je realizirana v svojem razredu, vsak razred pa razširja vmesnik *IAktivnostObject*:

```
public interface IAktivnostObject
{
    /// <summary>
    /// Opis aktivnosti
    /// </summary>
    string Opis { get; set; }

    /// <summary>
    /// ID aktivnosti v podatkovni bazi
    /// </summary>
    int ID { get; set; }

    /// <summary>
    /// Enolična oznaka aktivnosti v procesu
    /// </summary>
    string Oznaka { get; }

    /// <summary>
    /// Uspešna ali neuspešna izvedba aktivnosti
    /// </summary>
    bool UspesnoIzvedena { get; }

    /// <summary>
    /// Aktivnost se izvaja ročno (Human Task)
    /// </summary>
    bool RocnaIzvedba { get; }

    /// <summary>
    /// Napaka, ki je nastala pri izvajanju aktivnosti
    /// </summary>
    Exception Napaka { get; }

    /// <summary>
    /// Izvede aktivnosti
    /// </summary>
    /// <returns>uspešnost izvedbe</returns>
    bool Izvedi(Guid procesID);
}
```

To omogoča, da je vsaka aktivnost neodvisna enota, ki se jo da ločeno razvijati in vzdrževati. V metodi *Izvedi()* vsebuje logiko, ki pripada tej aktivnosti.

Ročna aktivnost

Posebnost med aktivnostmi so ročne aktivnosti oziroma *Human Tasks*. To so aktivnosti, ki jih mora uporabnik izvesti ročno (lahko s pomočjo računalnika), ni pa jih možno avtomatizirati. Te aktivnosti se izvajajo dlje časa. Ker sta začetek izvajanja teh aktivnosti in njihovo trajanje neznana, je treba proces za ta čas suspendirati, njegovo stanje pa shraniti.

Sistem za upravljanje procesov deluje po enakem principu. Ko naleti na ročno aktivnosti (ta vsebuje oznako, da gre za ročno aktivnost), zabeleži v dnevnik izvajanja, da je prišel do ročne aktivnosti in da ta čaka na dodelitev uporabniku v izvajanje.

Zatem se proces ustavi, njegovo stanje (objekt) pa se serializira in shrani v podatkovno bazo.

Ko skrbnik procesa dodeli aktivnost uporabniku (preko uporabniškega vmesnika), storitev za orkestracijo procesa spremeni stanje aktivnosti iz *'čaka na dodelitev'* v *'čaka na izvedbo'*, kar pomeni, da aktivnost čaka prevzem.

Uporabnik, ki mu je bila dodeljena aktivnost, to prevzame v izvajanje oziroma samo potrdi prevzem preko uporabniškega vmesnika.

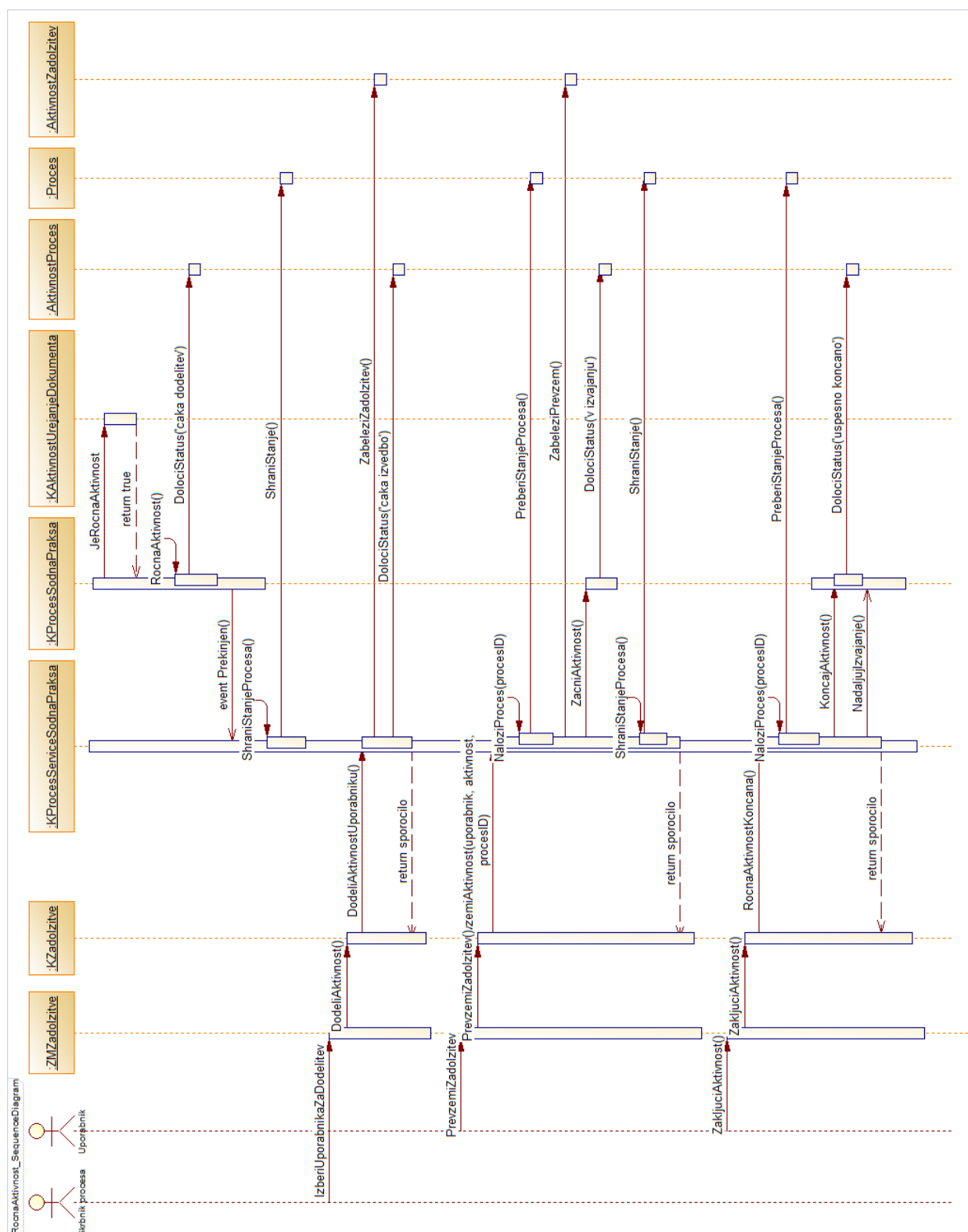
Na tem mestu naj omenim možnost, da se prevzem aktivnosti izvede samodejno, na primer v aplikaciji za urejanje dokumentov. Ko uporabnik odpre aplikacijo, lahko ta pridobi seznam aktivnosti in pripadajočih dokumentov, ki pripadajo uporabniku in čakajo na izvedbo. Uporabnik odpre dokument za urejanje, aplikacija pa sproži prevzem aktivnosti.

To omogoča zasnova sistema, katerega v celoti upravlja storitev za orkestracijo procesov in s katero aplikacija tudi komunicira. Aplikacije ni potrebno prirejati in ji omogočiti dostopa do podatkovne baze IUSProcesDB, dodati je treba le klic storitve.

Ko uporabnik na tak ali drugačen način prevzame aktivnost, storitev naloži stanje procesa iz podatkovne baze in zabeleži, da se je ročna aktivnost začela izvajati. Zatem zopet suspendira izvajanje procesa.

Po končanem izvajanju aktivnosti, uporabnik potrdi zaključek aktivnosti. Storitev za orkestracijo procesa, naloži stanje procesa, označi aktivnost kot končano in asinhrono sproži nadaljevanje izvajanja procesa.

Celoten postopek izvedbe ročne aktivnosti je prikazan z diagramom zaporedja na sliki 3.10.



Slika 3.10: Izvajanje ročne aktivnosti, prikazano z diagramom zaporedja

3.3.3 Implementacija podpornih spletnih storitev

Podporne spletne storitve so implementirane s šestimi razredi:

- *WSPorocanje*
- *WSPovezave*
- *WSPrenosDokumentov*
- *WSRazclenjevanje*
- *WSSifranti*

Imena razredov so dovolj samoumevna, zato posamezni razredi ne potrebujejo posebne razlage. Primer klica spletne storitve iz aktivnosti je prikazan na diagramu zaporedja v sliki 3.9.

3.3.4 Implementacija spletnega vmesnika

Spletni vmesnik je izdelan v tehnologiji ASP.NET.

Pred uporabo spletne aplikacije se je potrebno prijaviti, za kar poskrbi sistem sam. Med prijavo se zabeleži, ali je uporabnik tudi skrbnik procesa. Glede na vlogo uporabnika, ki je bila ugotovljena pri prijavi, se mu prikaže primerna vsebina.

Glavni del spletne aplikacije za upravljanje procesa je nadzorna plošča, kjer so uporabniku na enem mestu na voljo povzetki podatkov o izvajanju in stanju procesov. Prikaz vsebine nadzorne plošče se razlikuje glede na tip uporabnika. Posnetek nadzorne plošče skrbniku je prikazan na sliki 3.11.

Nadzorna plošča vsebuje podatke o:

- aktivnih procesih,
- končanih procesih,
- napakah pri procesih,
- izdelanih poročilih in
- dodeljenih aktivnostih uporabnikom.

Poleg nadzorne plošče spletna aplikacija za upravljanje vsebuje še zaslonske maske za:

- začetek novega procesa,
- pregled vseh procesov (slika 3.12),
- pregled napak procesov,
- pregled vseh dodeljenih in čakajočih zadolžitev,
- pregled končanih aktivnosti,
- pregled obdelanih paketov in dokumentov ter iskanje po dokumentih,
- pregled izdelanih poročil.

Podatke spletna aplikacija dobi s klicem ustreznih metod spletne storitve za orkestracijo procesa in nikoli sama ne dostopa do podatkovne baze IUSProcessDB.

3.4 Razširitve sistema in alternativne rešitve

Razširitev sistema za upravljanje več procesov je možna z dodajanjem dodatnih spletnih storitev za orkestracijo procesa in po potrebi tudi z dodajanjem podpornih spletnih storitev, ki podpirajo izvajanje aktivnosti v procesih.


3.4.1 Windows Workflow Foundation

V primeru, da so spremembe v procesih pogoste, bi bilo smiselno vpeljati boljši način realizacije logike procesov. Primer izboljšave, ki pa ni platformno neodvisna, ampak ja vezana na Microsoftovo ogrodje .NET, je uporaba Windows Workflow Foundation (WF).


WF nudi razvijalcu možnost, da proces implementira z uporabo grafičnega urejevalnika v razvojnem orodju Microsoft Visual Studio. Za vsak proces lahko zgradi poljubne aktivnosti, ki jih potem prenaša v urejevalnik in jih med seboj povezuje s standardnimi programskimi konstrukti, kot so na primer *if* stavki, *for* in *while* zanke, in tudi diagrami poteka.

To daje razvijalcu možnost, da namesto v kodi proces definira z uporabo gradnikov. S tem je spreminjanje procesa lažje.

Kljub temu pa mora spremembe v procesih vpeljati razvijalec, saj je za izdelavo Workflow diagramov potrebno programersko znanje in razumevanje programskih konstruktov. Analitiki procesov teh znanj večinoma nimajo.



IUS PROCES - UPRAVLJANJE PROCESOV



roman [odjava]

Nadzorna plošča

Začni proces

Procesi

Napake

Zadolžitve





Aktivnosti

Paket in dokumenti

Poročila

PREGLED PROCESOV

AKTIVNI PROCESI
Procesi, ki se trenutno izvajajo:

proces ID	trenutna aktivnost	čas začetka aktivnosti	status aktivnosti
a9e3891a-504c-4ef9-aac3-351ebb8f6629	Urejanje dokumenta	17.03.2011 20:31:53	čaka na dodelitev  
b22c9833-9c1f-4cdc-8d31-5fc0b53b1973	Urejanje dokumenta	17.03.2011 20:30:25	čaka na izvedbo  

ZAČASNO PREKINJENI PROCESI
noben proces ni začasno ustavljen

UKINJENI PROCESI
Procesi, ki so bili ukinjeni:

proces ID	zadnja izvedena aktivnost	čas začetka	čas ukinitve
cdd23425-9f47-4552-93ed-754002d7e714	Preverjanje skladnosti poročila	17.03.2011 10:35:18	17.03.2011 10:35:24
fab417d9-e40c-4c8c-9e7f-5bc450a4768e	Prenos v IUS DOCUMENT	17.03.2011 09:39:10	17.03.2011 09:39:55

USPEŠNO KONČANI PROCESI
Procesi, ki so bili uspešno končani:

proces ID	čas začetka	čas zaključka
777a5548-ff18-4c37-b007-4fc1f3911587	17.03.2011 09:32:00	17.03.2011 09:33:32
30e4f55e-9ec0-4ad9-b4e3-1c7f1503ac14	17.03.2011 09:36:17	17.03.2011 09:38:35
920a82b6-070b-46b9-ba3c-ff9d0030f6ce	17.03.2011 10:37:27	17.03.2011 10:47:11

© 2011 - Roman Thaler

Slika 3.12: Spletni vmesnik za prikaz vseh procesov

3.4.2 Orodja za izvajanje BPEL (BPEL engine)

Orodja za izvajanje BPEL omogočajo, da spremembe v procesih implementirajo izvajalci procesov ali analitiki sami. To storijo tako, da popravijo model procesa v BPMN. Ta se potem prevede v BPEL (Business Process Execution Language), ki se izvaja na procesnem strežniku. Spremembe so v proces vpeljane takoj.

V obstoječem sistemu bi orodje za izvajanje BPEL nadomestilo storitev za orkestracijo procesa. Podporne storitve bi ostale enake, saj bi jih še vedno uporabljale aktivnosti. Spremenil bi se le način izvajanja procesa iz izvajanja kode v izvajanje BPEL. Umestitev orodje za izvajanje BPEL prikazuje slika 3.13.

Na trgu obstaja veliko različic teh orodji, in sicer za različne platforme. Od kompleksnejših, kot so BizTalk Server in IBM Websphere, do enostavnejših.

Glede na to, da večina procesov v podjetju temelji na Microsoftovi platformi .NET, sem primarno iskal alternativne možnosti za vpeljavo takega orodja, ki bi deloval na tej platformi. Edina možna alternativna rešitev na platformi .NET je BizTalk Server oziroma BizTalk podjetja Microsoft.

Microsoft BizTalk Server

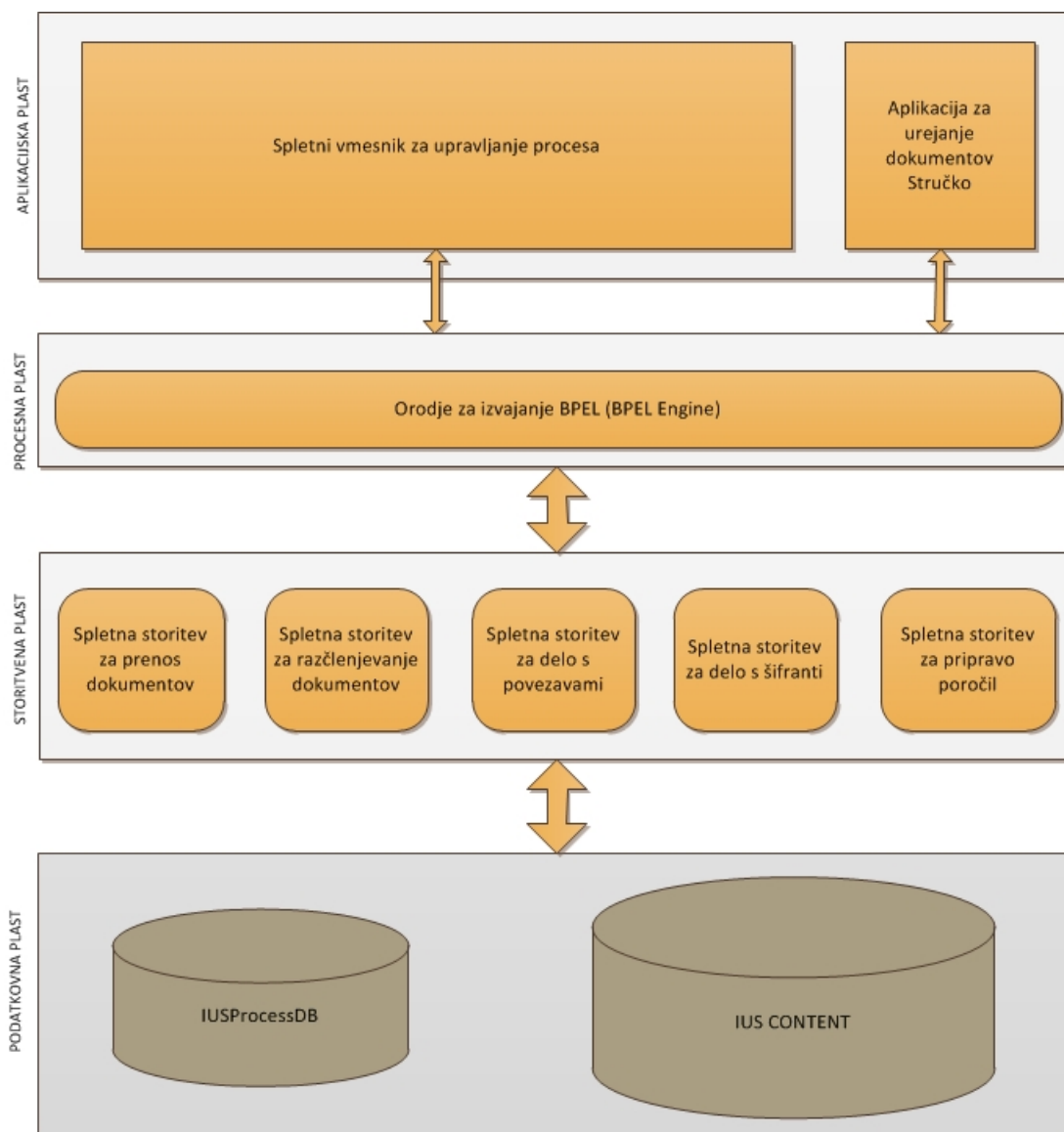
BizTalk pravzaprav nudi celoten spekter informacijskih storitev, ki jih potrebuje veliko podjetje pri svojem delovanju. Orodje za izvajanje procesov je le ena izmed njih.

Orkestracija procesov v BizTalk-u ni implementirana z BPEL jezikom samim, ampak s kompatibilno različico jezika xLANG. Urejanje procesov poteka v urejevalniku Orchestration Designer. Pripravljeni procesi se v obliki datotek shranijo na strežnik, kjer jih BizTalk Orchestration Engine izvede, ko je to potrebno. Pri gradnji procesov se lahko uporabijo vsi gradniki iz ogrodja .NET.

BizTalk Orchestration Engine izvaja tudi naslednje operacije:

- kreira in izvaja instance procesov,
- vzdržuje stanje aktivnih procesov,
- izvaja optimizacijo procesov, ki se trenutno izvajajo.

Poleg orodja za izvajanje procesov BizTalk nudi tudi storitve za nadziranje procesov (BAM).



Slika 3.13: Primer arhitekture sistema z orodjem za izvajanje BPEL

Kljub temu, da BizTalk nudi vse storitve, ki jih podjetje potrebuje za obvladovanje procesov, pa je rešitev preveč kompleksna za uporabo na enostavnejših procesih. Čas, potreben za vpeljavo in prehod na to rešitev, je prevelik. Poleg tega procesi niso tako kritični, da bi morale biti spremembe uvedene čim hitreje ali celo v realnem času.

Druga orodja za izvajanje BPEL

Glavni alternativni Microsoftovemu produktu sta IBM WebSphere in webMethods podjetja Software AG. Obe rešitvi sta enako ali pa celo še bolj kompleksni kot Microsoft BizTalk, poleg tega pa delujeta na drugi platformi. To sicer ni ovira, vendar je izkoristek tehnologije veliko boljši, če uporabljamo orodja iste platforme, saj lahko tako maksimalno izkoristimo njene zmožnosti.

IBM Websphere

IBM Websphere je obsežen paket storitev in strežnikov. Eden od teh strežnikov je tudi Websphere Process Server, ki omogoča izvajanje procesov. Temelji na strežniku Websphere Application Server. Za načrtovanje procesov se uporablja razvojno orodje Websphere Integration Developer, ki temelji na Eclipse-u.

Software AG webMethods

webMethods v svojem spektru produktov nudi tudi BPMS (sistem za upravljanje procesov), ki vsebuje orodje za modeliranje procesov (temelji na orodju Eclipse), orodja za simulacijo izvajanja procesov in tudi orodja za upravljanje procesov za velika podjetja. Omogoča izdelavo delovnih tokov (workflow) in njihovo avtomatizacijo.

Druge rešitve:

ActiveVOS

Produkt ActiveVOS je skupek orodij namenjen sistemskih načrtovalcem, razvijalcem in analitikom za modeliranje in izvajanje poslovnih procesov. Vsebuje ActiveVOS Designer, ki omogoča načrtovanje procesov v BPMN in ActiveVOS Server, na katerem se procesi izvajajo. Temelji na Javi.

bonitasoft BPM Suite

Bonita Open Solution združuje tri rešitve:

- Studio za modeliranje procesov,

- BPM & Workflow engine za izvajanje procesov in
- Bonita User Experience portal, ki omogoča upravljanje in nadzorovanje procesov

Prednost te rešitve je odprtost, saj je izdana pod licenco GPL.

Apache ODE

Apache ODE (Orchestration Director Engine) izvaja poslovne procese, ki so napisani po standardu WS-BPEL. Omogoča komunikacijo z drugimi spletnimi storitvami, prejemanje in pošiljanje sporočil, delo s podatki in ponovno vzpostavljanje stanja ob napakah. Podpira tudi tako imenovane long-running procese oziroma procese, ki se ne izvajajo v realnem času.

jBoss jBPMN

jBPMN je paket BPM. Vsebuje orodje za izvajanje procesov, napisanih v BPMN, hkrati pa nudi dodatek za razvojno okolje Eclipse, ki omogoča načrtovanje procesov.

Poglavje 4

Sklepne ugotovitve

Tako prenovljeni proces, kot tudi zasnovani sistem, bosta nedvomno izboljšala učinkovitost izvajanja procesov za obdelavo dokumentov.

Proces je izboljššan in optimiziran, zmanjšano je tudi število korakov. Največji doprinos k učinkovitosti je gotovo manjše število procesov, saj namesto petih procesov izvajamo le enega.

Odločitev za načrtovanje novega sistema, namesto uporabe in prilagoditve obstoječe rešitve komercialnih ponudnikov, se lahko zaradi časa, potrebnega za njegovo načrtovanje in izdelavo, sprva ne zdi popolnoma optimalna. Vendarle pa je treba upoštevati, da je opisani sistem v celoti izdelan po meri obstoječih procesov v podjetju. Tako raven optimizacije je s prilagajanjem generičnih rešitev težko doseči. Dodaten argument v prid izdelavi lastnega sistema je tudi kompleksnost komercialnih oziroma generičnih rešitev. Te so namenjene uporabi v različnih poslovnih okoljih, za katere sicer nudijo veliko storitev. Večine teh storitev podjetje ne potrebuje in samo večajo kompleksnost rešitve.

Čeprav obravnavani sistem prenove procesov stremi k odličnosti, sem ga kritično ocenil. V ta namen sem za pregled prednosti in nekaterih potencialnih slabosti sistema izdelal SWOT oziroma PSPN analizo, kjer so predstavljeni notranji dejavniki, ki vplivajo na sistem: prednosti in slabosti sistema, izpostavil pa sem tudi zunanje dejavnike: priložnosti in nevarnosti.

Vsi dejavniki so zbrani v tabeli 4.1, podrobno pa so opisani v nadaljevanju.

<p style="text-align: center;">PREDNOSTI</p> <ul style="list-style-type: none"> • uporaba spletnih storitev • razširljivost na druge procese • sledenje izvoru dokumenta • spletni vmesnik • prilagodljivost • skalabilnost 	<p style="text-align: center;">SLABOSTI</p> <ul style="list-style-type: none"> • logika procesov shranjena v kodi • v izjemnih primerih izguba stanja procesov ob izpadih sistema • zapletena implementacija v interoperabilnih okoljih
<p style="text-align: center;">PRILOŽNOSTI</p> <ul style="list-style-type: none"> • sistem postane produkt, ki ga podjetje trži kot samostojno rešitev • sistem se začne uporabljati kot glavni sistem za upravljanje vseh procesov 	<p style="text-align: center;">NEVARNOSTI</p> <ul style="list-style-type: none"> • sistem v celoti nadomesti zunanja rešitev • podjetje spremeni svojo usmeritev in se ne ukvarja več z obdelavo dokumentov • slaba implementacija in prekinitve v delovanju onemogočijo redno ažuriranje dokumentov

Tabela 4.1: SWOT analiza sistema za upravljanje procesov

4.1 Prednosti

Uporaba spletnih storitev

Glavna prednost sistema je njegova prilagodljivost, saj ni vezan na uporabo določenega vmesnika, ampak ga lahko uporablja več različnih aplikacij. To omogoča storitvena plast in spletna storitev za orkestracijo procesa, ki preko standardiziranega vmesnika nudi dostop do podatkov o procesih.

Prav tako so spletne storitve trenutno zelo aktualne in se bodo še naprej razvijale in izboljševale.

Razširljivost na druge procese

Sistem je zasnovan tako, da se ga lahko razširi za uporabo pri drugih procesih, tudi tistih, ki se ne izvajajo za obdelavo dokumentov.

Sledenje izvoru dokumenta

Ker sistem beleži datoteke in pakete, v katerih so dokumenti prišli v sistem, nudi uporabnikom možnost, da v primeru napake ali nepravilnosti v dokumentu le tega enostavno izsledijo do izvora, ga tam popravijo in ponovijo proces uvoza.

Spletni vmesnik

Sistem sicer ni vezan samo na uporabo spletnega vmesnika, vendar pa spletni vmesnik v celoti odstrani potrebo po namestitvi aplikacije na vse računalnike. Enostavnejše je tudi vzdrževanje saj je spletno aplikacijo treba posodobiti le na enem mestu.

Prilagodljivost

Posamezne plasti, še posebej pa aplikacijska in storitvena plast, so šibko sklopljene, saj komunicirajo preko vnaprej določenih standardiziranih vmesnikov. Tako v primeru popravkov na podatkovni bazi ni treba popravljati še vseh aplikacij, ki komunicirajo s sistemom.

Skalabilnost

Vsaka spletna storitev se teoretično lahko izvaja na svojem strežniku. Bolj smotrna rešitev je izvajanje spletnih storitev za orkestracijo posameznih pro-

cesov na ločenih strežnikih, še posebej, če gre za zahtevnejše procese.

4.2 Slabosti

Logika procesov je shranjena v kodi

V trenutni zasnovi sistema je predvideno, da je logika sistema izdelana v programski kodi in shranjena v razredih. Za spreminjanje te logike potrebujemo programerja, saj od analitikov ne pričakujemo poznavanje programskih jezikov. Zaradi tega se čas vpeljave sprememb lahko podaljša.

Ta problem sem opisal tudi v poglavju 3.4. Rešiti se ga da z razširitvijo sistema z uporabo orodja za izvajanje BPEL, ali pa z Windows Workflow Foundation. Tako bi se procesi lahko spreminjali ad-hoc, kar bi to slabost lahko spremenilo v prednost.

Izguba stanja procesov ob skrajnih izpadih sistema

Zaradi narave izvajanja procesov in shranjevanja njihovega stanja, obstaja možnost, da se lahko ob izpadu strežnikov del informacije o izvajanju procesa izgubi. V tem primeru je potrebno proces povrniti na zadnje uspešno izvedeno stanje in od tam naprej nadaljevati izvajanje. Pri tem bi bili podatki o aktivnosti, ki se je izvajala ob izpadu sistema, izgubljeni.

Zapletena implementacija v interoperabilnih okoljih

V primeru, da se implementacija posameznih plasti sistema izvaja na različnih platformah, lahko pride do težav. Pri uporabi enotne platforme (kot je prikazano v implementaciji prototipa) ni težav s prenašanjem kompleksnih podatkovnih tipov, saj sta izvor in ponor na isti platformi. V primeru različnih platform pa je treba kompleksnejše tipe poenostaviti, kar lahko privede do večjega števila klicev spletnih storitev in njihove kompleksnejše logike.

4.3 Priložnosti

Sistem postane produkt, ki ga podjetje trži kot samostojno rešitev

Sistem lahko postane priznana rešitev, ki jo podjetje vključi v svoj prodajni program in jo ponuja na trgu.

Sistem se začne uporabljati kot glavni sistem za upravljanje vseh procesov

Če se sistem skozi uporabo zelo dobro izkaže, ga podjetje zaradi poenostavitve svoje infrastrukture lahko uporabi tudi za izvajanje drugih poslovnih procesov. V končni fazi lahko ta sistem izvaja vse poslovne procese v podjetju. V tem primeru bi bilo treba izdelati komponente za povezavo do CRM in ERP sistemov v podjetju.

4.4 Nevarnosti

Sistem v celoti nadomesti zunanja rešitev

Vodstvo podjetja se lahko odloči, da ne bo implementiralo sistema ali pa ga po vpeljavi preneha uporabljati in se odloči za že izdelano rešitev zunanjega ponudnika.

Podjetje spremeni svojo usmeritev in se ne ukvarja več z uvozom in obdelavo dokumentov

Zaradi prilagajanja razmeram v gospodarstvu podjetje lahko spremeni svojo poslovno usmeritev in se usmeri v drugo panogo, kjer tak sistem ni več potreben.

Slaba implementacija in prekinitve v delovanju onemogočijo redno ažuriranje dokumentov

Slaba implementacija sistema, ki povzroči (daljše) prekinitve v izvajanju procesov, lahko pripelje do situacije, v kateri podjetje ne bo zmožno redno osveževati vsebine dokumentov ali dodajati novih, pri čemer lahko podjetje utрпи tudi poslovno škodo.

Dodatek A

Diagram prenovljenega procesa za obdelavo dokumentov sodne prakse

Zaradi večje preglednosti je diagram razdeljen na dva dela.

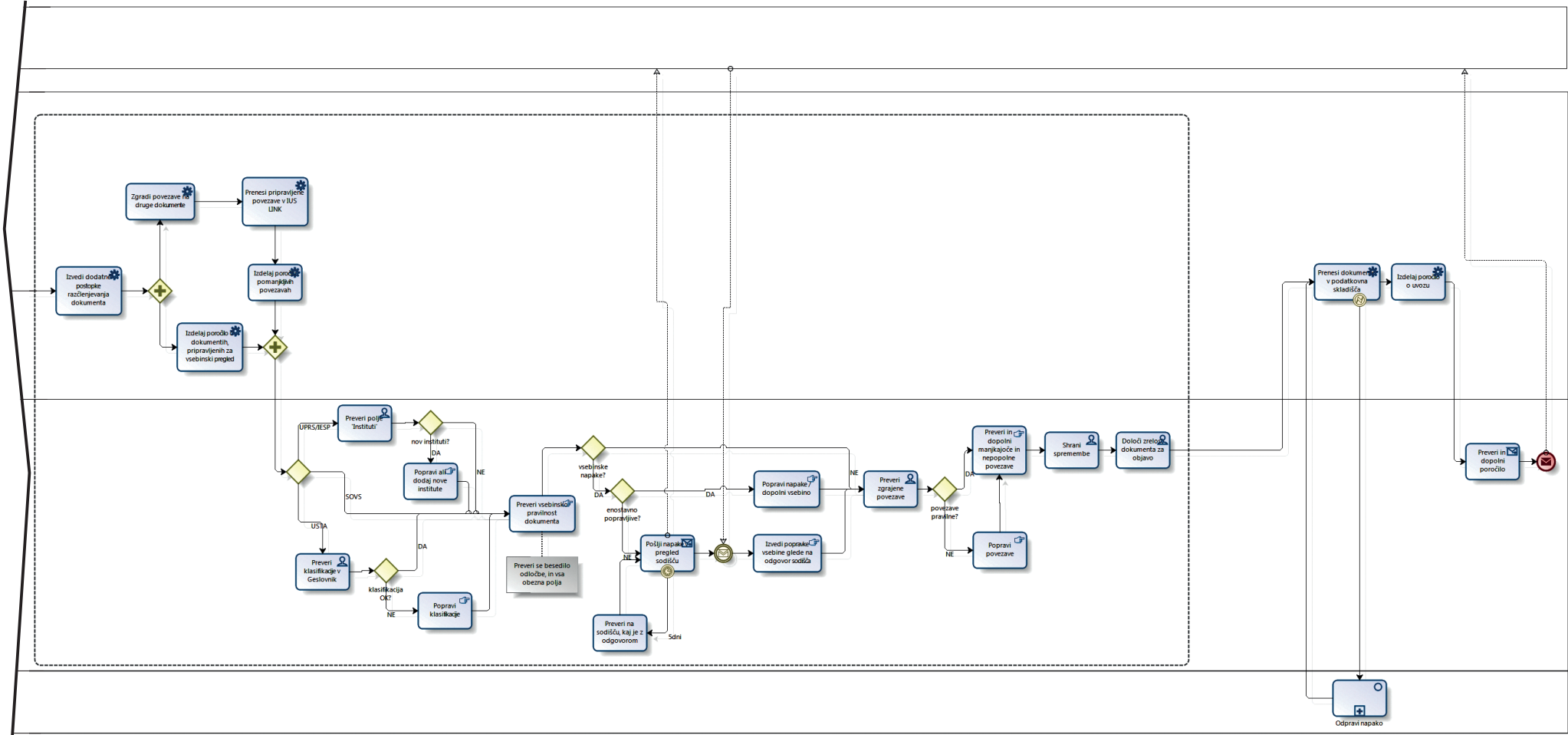


Diagram prenovljenega procesa za obdelavo dokumentov sodne prakse II. del

Slike

1.1	Konceptualni model podatkovne baze IUS TIME 3.0	7
2.1	Diagram procesa obdelave dokumentov Ustavnega sodišča	16
2.2	Diagram procesa obdelave dokumentov Vrhovnega sodišča	19
2.3	Diagram procesa obdelave dokumentov Višjega sodišča, Up- ravnega sodišča, povzetkov negmotnih škod	22
2.4	Struktua kontrolne datoteke	24
2.5	Diagram procesa za obdelavo dokumentov Računskega sodišča .	25
2.6	Tok podatkov v obstoječih procesih	26
2.7	Konceptualni pogled na proces obdelave dokumentov sodne prakse	29
2.8	Proces obdelave dokumentov sodne prakse, faza 'Zajem doku- mentov in tehnična pretvorba'	30
2.9	Faza priprave dokumentov na uvoz v PB IUS DOCUMENT in IUS LINK	32
2.10	Prikaz faze 'urejanje in dopolnitev vsebine dokumentov'	34
2.11	Prikaz toka podatkov v novem procesu in sistemu IUS CONTENT	36
3.1	Diagram primerov uporabe sistema za upravljanje procesov	39
3.2	Arhitektura sistema za upravljanje procesov	41
3.3	Prikaz povezanost storitve za orkestracijo procesa s procesni ob- jektovm in jegovimi aktivnostmi	44
3.4	Konceptualni model relacijske podatkovne baze IUSProcesDB .	46
3.5	Prikaz dopolnitve IIS in WAS z AppFabric	50
3.6	Nadzorna plošča AppFabric	51
3.7	Diagram zaporedja, ki prikazuje začetek nove instance procesa .	54
3.8	Prikaz metode NadaljujProces	55
3.9	Diagram zaporedja prikazuje izvajanje procesa	57
3.10	Izvajanje ročne aktivnosti, prikazano z diagramom zaporedja . .	60
3.11	Nadzorna plošča spletnega vmesnika	62
3.12	Spletni vmesnik za prikaz vseh procesov	64
3.13	Primer arhitekture sistema z orodjem za izvajanje BPEL	66

Literatura

- [1] J. Bean, SOA and Web services interface design : principles, techniques, and standards, Amsterdam: Morgan Kaufmann: Elsevier, 2010, str. 25-41
- [2] T.Erl, Service-Oriented architecture : concepts, technology, and design, Upper Saddle River (New Jersey) : Prentice Hall Professional Technical Reference, 2005, str. 126-130, 200-208 in 280-310
- [3] S. Jablonski, C. Bussler, Workflow management : modeling concepts, architecture and implementation, London: International Thomson Computer Press, 1996, str. 217
- [4] M. B. Juric, K. Pant, *Business process driven SOA using BPMN and BPEL : from business process modeling to orchestration and service oriented architecture*, Birmingham ; Mumbai: Packt Publishing, 2008, str. 7, 8, 69-71
- [5] I. Kaučič, F. Grad, Ustavna ureditev Slovenije - 4., spremenjena in dopolnjena izd., Ljubljana : GV založba, 2007, str. 294-298
- [6] K.Kočila, Urejevalnik dokumentov in povezav IUS-TIME, diplomsko delo: Univerza v Ljubljani, 2010
- [7] David Chappel, Introducing Windows Server AppFabric, članek, San Francisco: Chappell & Associates, 2010. Dostopno na: http://download.microsoft.com/download/7/F/8/7F8BD8A0-EB05-4DB5-A5A4-DD1D3C909A0A/Introducing_Windows_Server_AppFabric.pdf
- [8] E. Velinov, M. McKeown, Windows Server AppFabric Architecture Guide (2010). Dostopno na: <http://download.microsoft.com/download/A/5/B/A5B0ED08-844F-4E37-9F0F-BEE9F3043907/AppFabricArchGuide.docx>