

Rukovalac dokumentima (RuDok)

Specifikacija projekta

Generacija: 2021/2022

SADRŽAJ

Predgovor	2
SOFTVERSKI PROCES:	2
SOFTVERSKI PROIZVOD:.....	3
1. Kako FORMALIZOVATI IDEJU? - Osnovni koncepti	4
2. DOMEN REŠENJA - APSTRAKCIJA (šta treba da, u opštem slučaju, istakne rešenje koje je predmet razvoja ?)	5
3. DOMEN REŠENJA - KONKRETIZACIJA	6
4. OPŠTE OSOBINE ALATA - GUI KOMPONENTA	7
5. Prototipski razvoj - Evolutivni prototip	8
6. Prilozi	9
6.1. Format modela zahteva	9
6.2. Primer modela zahteva	10
6.3. Modelovanje arhitekture programskog proizvoda na bazi orkestracije komponenti	11
6.3.1. Formulisanje organizacije artifakata aplikativnog radnog okvira	11
6.3.2. Primer - Objektni model Arhitekture Aplikacije (Aplikativni radni okvir)	12
6.3.3. Primer - Dijagram komponenti	13
6.4. Model statičke strukture komponenti - UML Dijagram KLASA	14
6.4.1. Primer nacрта modela DOKUMENTA.....	14
6.5. Elementi primopredaje	15
6.5.1. Power Designer projekat - Rukovalac dokumentima (Izvezen u HTML dokument)	15
6.5.2. Specifikacija zahteva - PowerDesigner DataArchitect PD16 - model zahteva prema opisanom standardu.....	15
6.5.3. Model Arhitekture - PowerDesigner - DataArchitect PD16.	15
6.5.4. Modeli Komponenti - PowerDesigner - DataArchitect PD15. - objektni (klasni) model	15
6.5.5. Izvršna verzija aplikacije - Rukovalac dokumentima	15
6.5.6. Dokumentovanje aplikacije - Rukovalac dokumentima	15

Predgovor

Predmet **Dizajn softvera** je nastavni predmet sa misijom podizanja nivoa veština i znanja potrebnih za izradu složenih softverskih proizvoda koje uključuju usvajanje metoda i tehnika:

1. specifikacije i modelovanja fleksibilnih softverskih proizvoda
2. oslonac na proverene koncepte komponentizacije i dizajna komponenti softverskih proizvoda
3. upotrebe šablona za refaktorisanje interne arhitekture komponenti.

SOFTVERSKI PROCES:

- **Model životnog ciklusa:**
 - **Agilni pristup razvoju softvera** (Agile Software Development)
 - **Prototipski razvoj** - evolutivni prototip
- **Metodologija - Modelom upravljani razvoj softvera:**
 - **Specifikacija i modelovanje zahteva** (uz oslonac na usvojeni alat za modelovanje);
 - **Specifikacija i modelovanje interakcija** (UML - USE CASE formalizam uz oslonac na usvojeni alat za modelovanje)
 - **Specifikacija i modelovanje arhitekture** - (paketiciranje i model komponenti)
 - **Specifikacija i modelovanje statičke strukture komponenti** - (UML - KLASNI MODELI uz oslonac na usvojeni alat za modelovanje)
 - **Specifikacija i modelovanje ponašanja** - (UML dinamički dijagrami odabrani iz skupine usvojenih formalizama za opis ponašanja softverskih sistema - dijagram stanja, dijagram aktivnosti, dijagram sekvence i dijagram saradnje)

SOFTVERSKI PROIZVOD:

- **Interaktivan, GUI orijentisan, Događajima upravljani programski Alat**
- **Prototip** - *Evolutivni prototip uz oslonac na Objektno Orijentisanu Paradigmu (JAVA programski jezik, SWING GUI biblioteka i razvojno okruženje Eclipse)*
- **Standardizacija arhitekture:**
 - **Spoljašnja arhitektura** - dinamička kolekcija GUI komponenti
 - MVC (Model-View-Controller) - arhitektonski obrazac
 - Komponente koje manipulišu sa radnim okruženjem:
 - Komponente koje manipulišu sa **artifaktima softverskog proizvoda** (proizvodi koje softverski proizvod kreira i sa njima rukuje)
 - Uputstva i pomoć pri ovladavanju programskim proizvodom;
 - **Podešavanje programskog proizvoda** (Parametrizacija, Lokalizacija i personalizacija)
 - Logotip i o aplikaciji
 - **Unutrašnja arhitektura komponenti:**
 - MVC (Model-View-Controller) - arhitektonski obrazac
 - Refaktorizacija unutrašnje arhitekture - upotreba šablona (Observer, Singleton, State, Composite, Factory Method, Abstract Factory, Proxy, Bridge, Command)
- **Algoritmi, strukture podataka i organizacija podataka**
 - **Model dokumenta**
 - **Serijalizacija i deserijalizacija** - (čuvanje radnog prostora, kolekcija dokumenata, modela dokumenata i instanci modela dokumenata)

1. Kako FORMALIZOVATI IDEJU? - Osnovni koncepti

Rukovalac dokumentima: (RuDok)

- **Domen problema:** - Dizajniranje i operativna upotreba dokumenata (*Dokument - složeni proizvod sa potencijalno multimedijalnim sadržajem.*)
- **Zainteresovane strane:** - *Organizacioni sistemi i pojedinci koji u osnovi svoje delatnosti imaju rad sa dokumentima slobodne strukture i organizacije.*
 - **Koga kontaktirati u cilju smanjenja stepena nepoznavanja domena problema i domena rešenja?**
 - **Eksperte iz domena problema** - operativno se bave kreiranjem dokumenata (dizajneri dokumenata) i poseduju neophodna iskustva vezana za domen izvršavanja;
 - **Eksperte iz domena upotrebe** - operativni korisnici kreiranih dokumenata;
 - **Poslovni eksperti** - poseduju operativne pokazatelje (podatke/informacije/bazu znanja) vezane za poslovne aspekte softverskog proizvoda;
 - **Eksperte za razvoj softvera** - vladaju konceptima izgradnje fleksibilnih programskih proizvoda.
- **Vrsta proizvoda koji želimo razviti:** - **Alat** (proširivi radni okvir)
- **Način plasmana:** Slobodno tržište - ne projektuje se za poznatog kupca.

2. DOMEN REŠENJA - APSTRAKCIJA (šta treba da, u opštem slučaju, istakne rešenje koje je predmet razvoja ?)

- **Rukovanje Parametrima Alata**
- **Rukovanje Radnim prostorima alata** - Slobodno formiranje radnih prostora koji sadrže kolekcije dokumenata;
- **Rukovanje Kolekcijama dokumenata** - Slobodno formiranje kolekcija arhitekture dokumenata u skladu sa modelom dokumenta (Prilog 6.4.1.);
- **Rukovanje Dokumentima** - Opis, definisanje, operativna upotreba i trajno čuvanje dokumenta proizvoljne strukture;
- **Delenje celih dokumenata** - Dokumente je moguće deliti između kolekcija dokumenata;
- **Organizacija i strukturiranje dokumenta** - podela unutrašnje strukture dokumenta na stranice i kolekcije stranica. Stranicu definisati kao apstraktni element unutrašnje organizacije dokumenta. *Kao konkretizaciju apstraktne stranice uvesti - PREZENTACIONI SLAJD.*
- **Rukovanje elementima unutrašnje strukture dokumenta** -
- **Organizacija i struktura stranice dokumenta** - Podela stranice na Slotove koji su nosioci tipiziranog sadržaja.
- **Rukovanje elementima unutrašnje strukture stranice**
- **Delenje delova dokumenata** - Stranice je moguće deliti između različitih dokumenata;
- **Delenje delova stranica** - Delove stranica je moguće deliti između različitih Stranica ili Dokumenata;
- **Rukovanje tipiziranim sadržajima slotova** - rukovaoci monotipovima.

3. DOMEN REŠENJA - KONKRETIZACIJA

Projektovati editor prezentacija slajdova. Prezentacija se sastoji od slajdova, a svaki slajd može da sadrži proizvoljnu kombinaciju tekstualnih elemenata (*plain* tekst) i slika, raspoređenih na proizvoljnim pozicijama na slajdu.

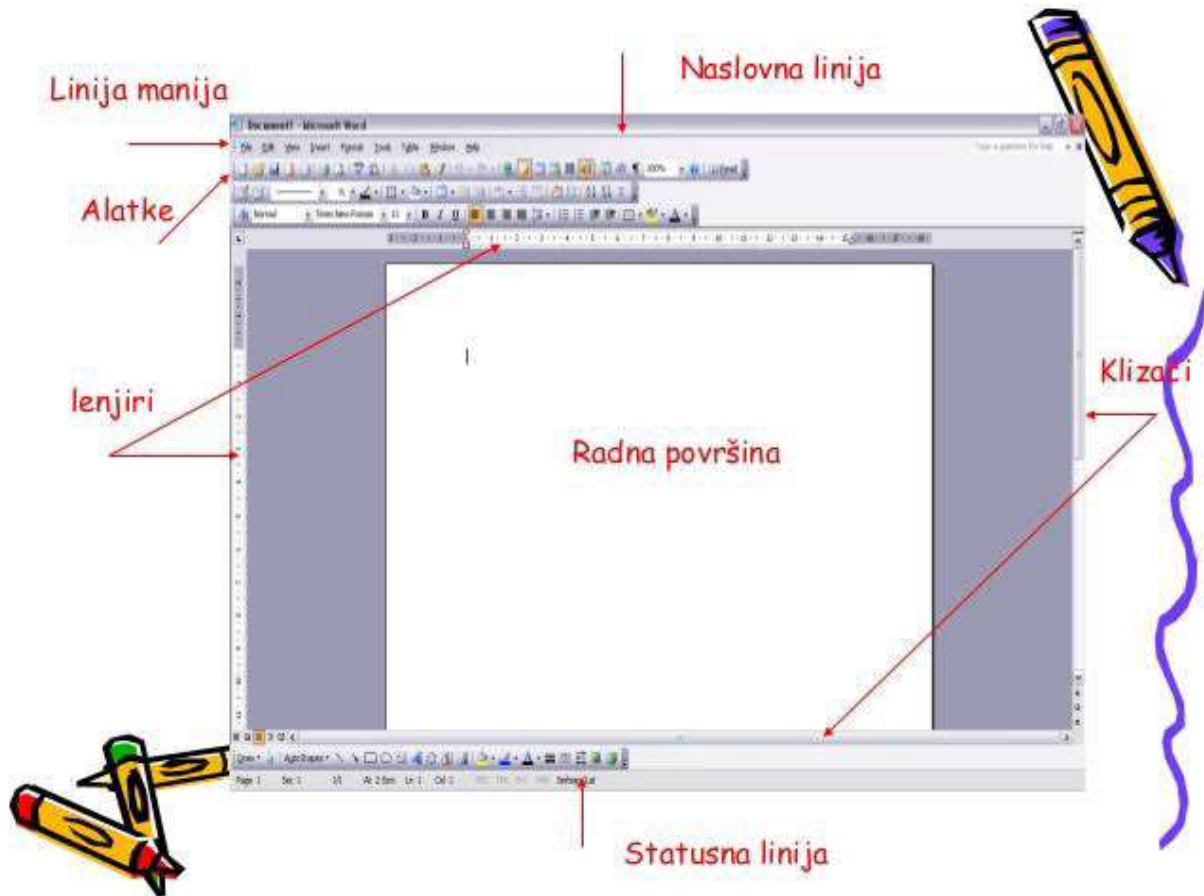
Svi slajdovi imaju istu sliku za svoju pozadinu, isto zaglavlje (*header*) i podnožje (*footer*). Zaglavlje i podnožje mogu da sadrže: naziv prezentacije, ime i prezime autora, datum kreiranja, broj tekućeg slajda, ukupan broj slajdova.

Editor treba da podržava osnovne funkcije za manipulaciju sadržajem prezentacije: dodavanje, brisanje slajdova i njihovih elemenata, pomeranje i izmenu veličine elemenata i da radi u dva gruba režima rada: **dizajniranje prezentacija** i **vizualizacija prezentacija**. Rešenje projektovati tako da omogući naknadno dodavanje novih tipova elemenata bez izmene postojećih klasa.

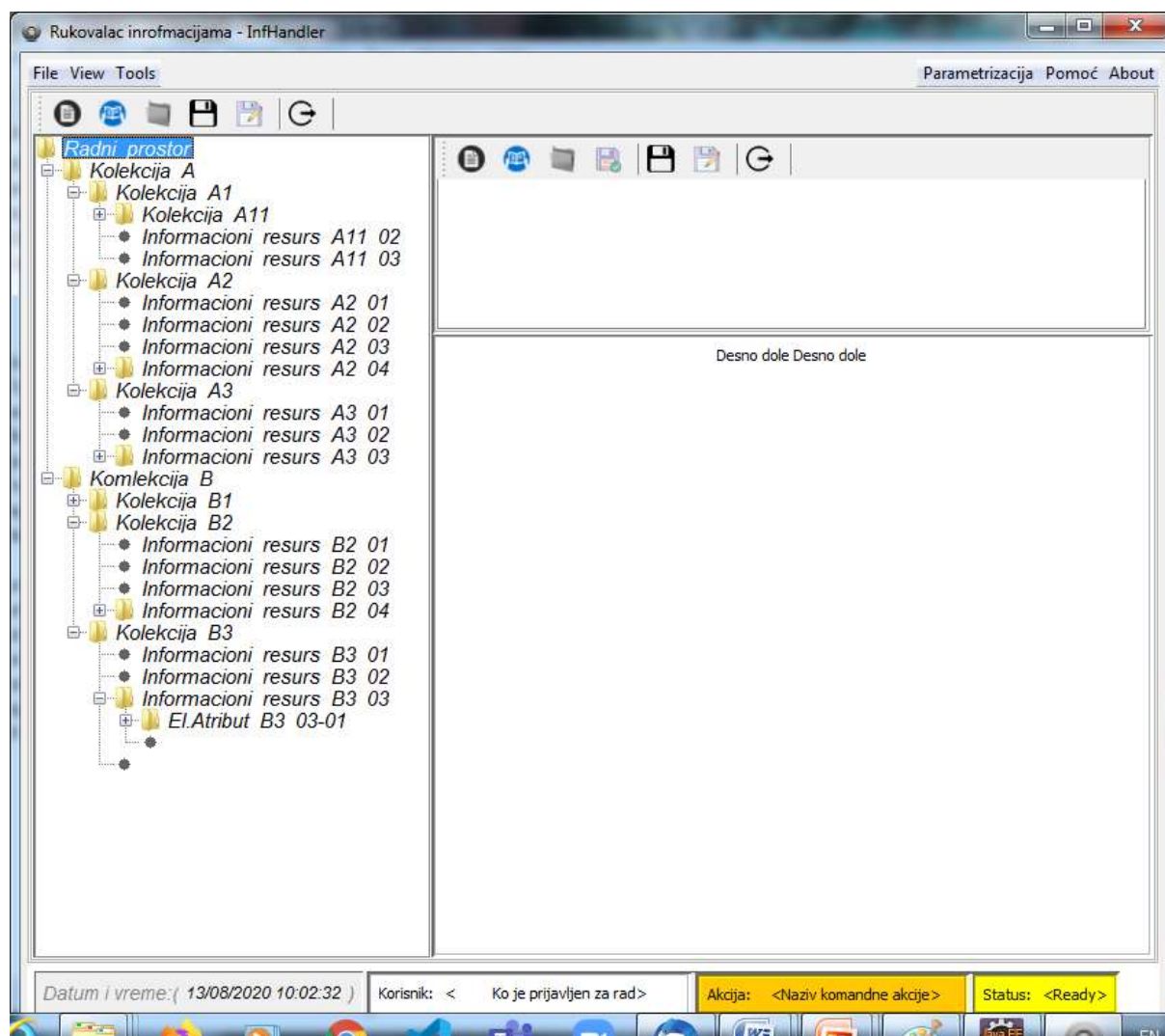
4. OPŠTE OSOBINE ALATA - GUI KOMPONENTA

Opšta Arhitektura GUI komponente softverskog proizvoda:

1. Naslovna linija (*titlebar*)
2. Glavna linija menija (*menubar*)
3. Paleta alatki (*toolbar*)
4. Radna površina (*workspace*)
5. Statusna linija (*statusbar*)



5. Prototipski razvoj - Evolutivni prototip



6. Prilozi

6.1. Format modela zahteva

Model zahteva je neophodno predstaviti u donjoj tabelarnoj formi:

Tabela 5.1.1. Format modela zahteva

R.br.	Opis	Tip	Prioritet	Zainteresovane strane	Rizik	Status	Procena napora
1.	<i>Tekstualna formulacija zahteva</i>						<i>Efektivni broj dana</i>
2.							
n-							

Status	Značenje
Dr	Draft
De	Definisan
Po	Potvrđen

Rizik	Značenje
Vi	Visok
Sr	Srednji
Ni	Nizak

Oznaka	Naziv
1	
2	
3	
k	

Prioritet	Opis tipa
ES	Esencijalni
VP	Vrlo poželjan
PO	Poželjan
OP	Opcion
NP	Nepoželjan

Tip	Opis tipa
F	Funkcionalni
N	Nefunkcionalni

Model zahteva treba da ima potpun obuhvat (sve grupe zahteva - Domen problema, Domen rešenje, Domen izvršavanja) uz specificiranje detalja u skladu sa rezultatima pojedinačnih etapa u procesu prikupljanja i analize zahteva.

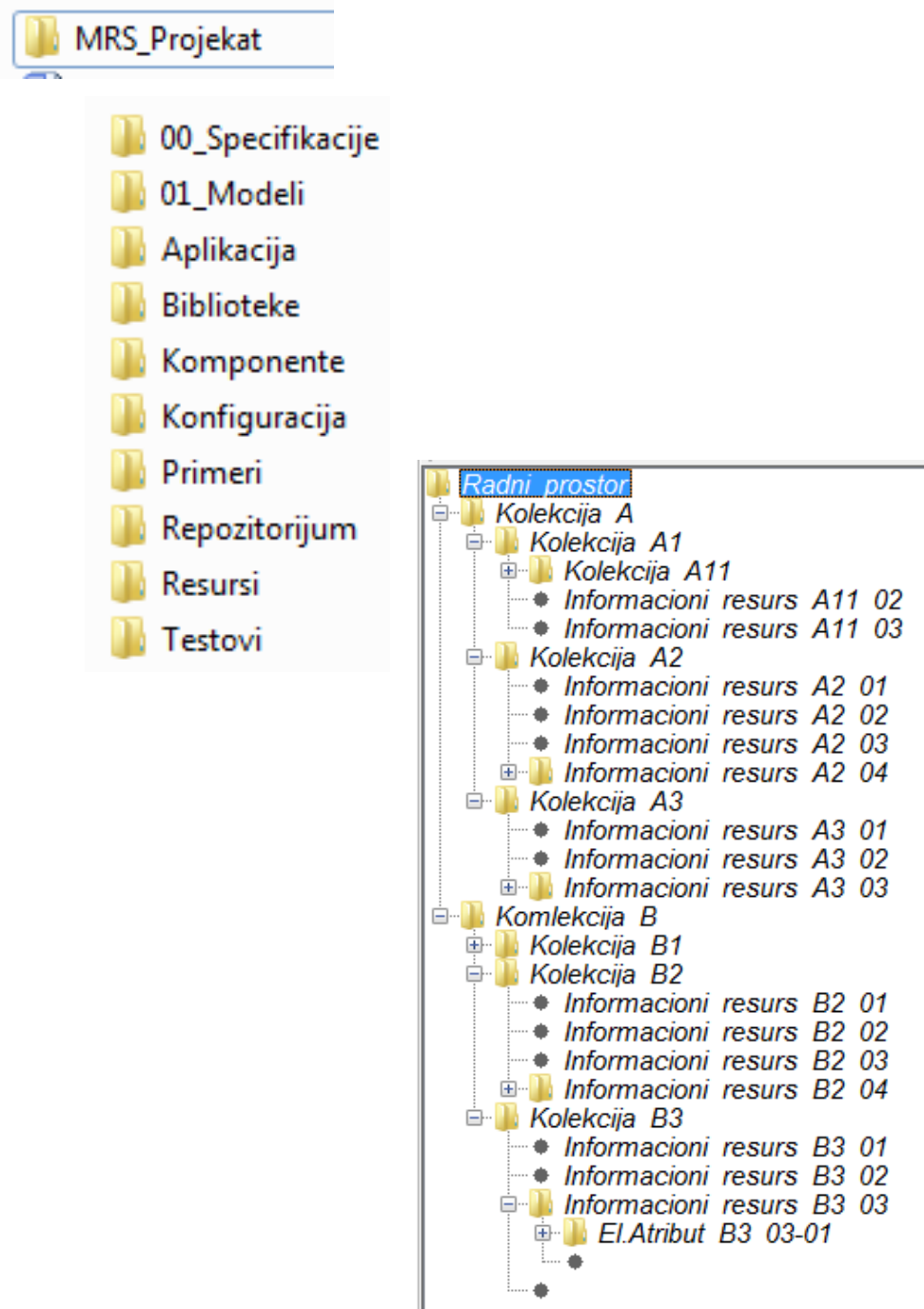
6.2. Primer modela zahteva

R.br.	Opis	Tip	Prioritet	Zainteresovane strane	Rizik	Status	Procena napora
1.	Instalacija Neophodno je obezbediti podršku instalaciji PROGRAMSKOG PROIZVODA.	N	Op	Administrator alata	Vi	De	35
2.	Pokretanje programskog proizvoda Neophodno je obezbediti pokretanje programskog proizvoda na način kako je to definisano prilikom njegove instalacije. Prilikom pokretanja neophodno je dovesti programski proizvod u kontekst u kom je ostavljen kod poslednjeg prekida rada.	N	Es	Operativni korisnik	Vi	De	12
3.	Operativna upotreba programskog proizvoda	F	Es	Operativni korisnik	Vi	Dr	0
4.	Prekid rada programskog proizvoda Neophodno je, u bilo kom trenutku operativne upotrebe, obezbediti kontrolisani prekid rada programskog proizvoda. Prilikom prekida rada neophodno je očuvati konzistentnost svih artifakata sa kojima je manipulirano u toku rada. Potrebno je sačuvati kontekst upotrebe programskog alata u cilju njegove restauracije kod sledećeg pokretanja.	F	Es	Operativni korisnik	Vi	De	12
	.						
	.						
	.						
n-							

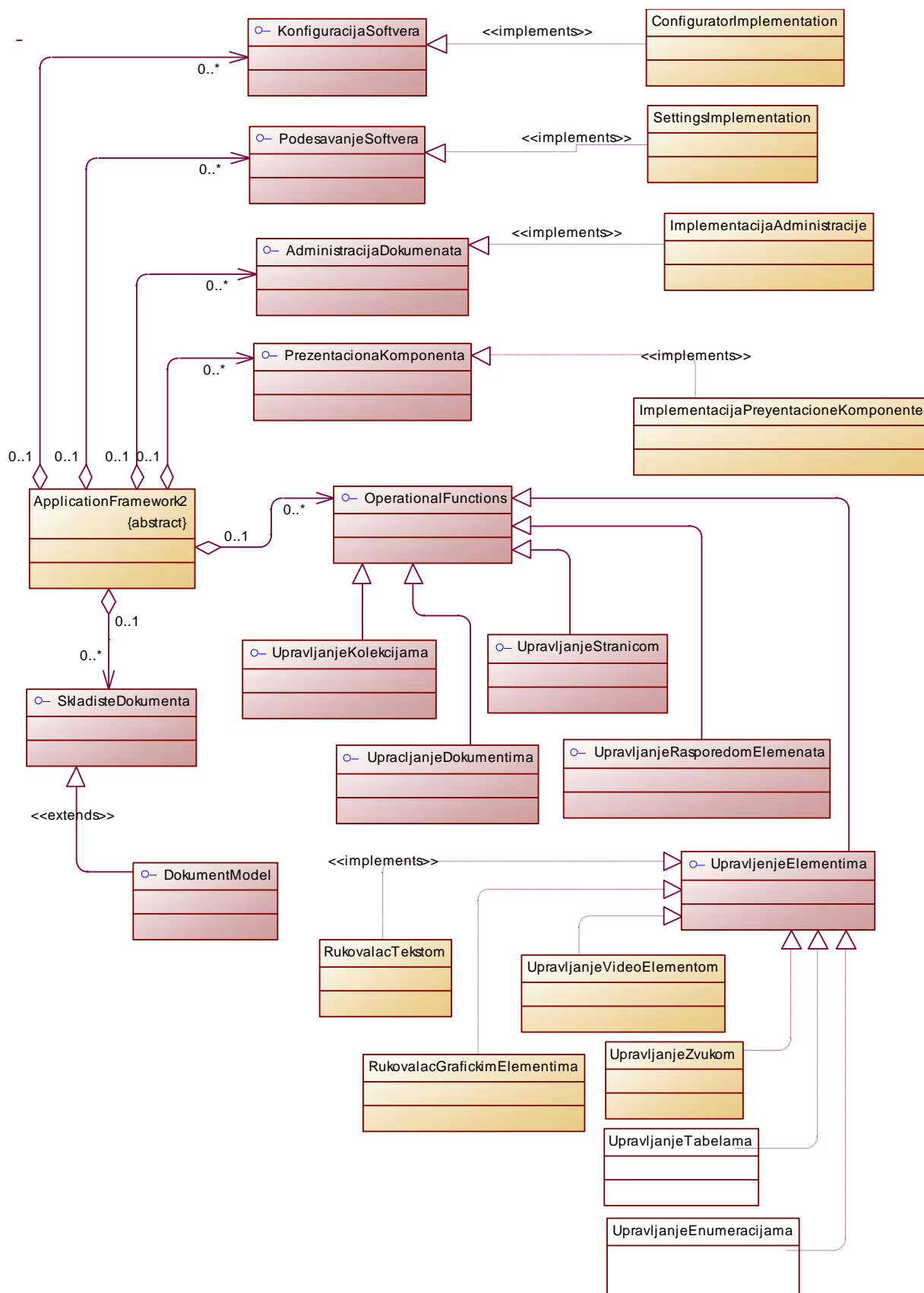
6.3. Modelovanje arhitekture programskog proizvoda na bazi orkestracije komponenti

6.3.1. Formulisanje organizacije artifakata aplikativnog radnog okvira

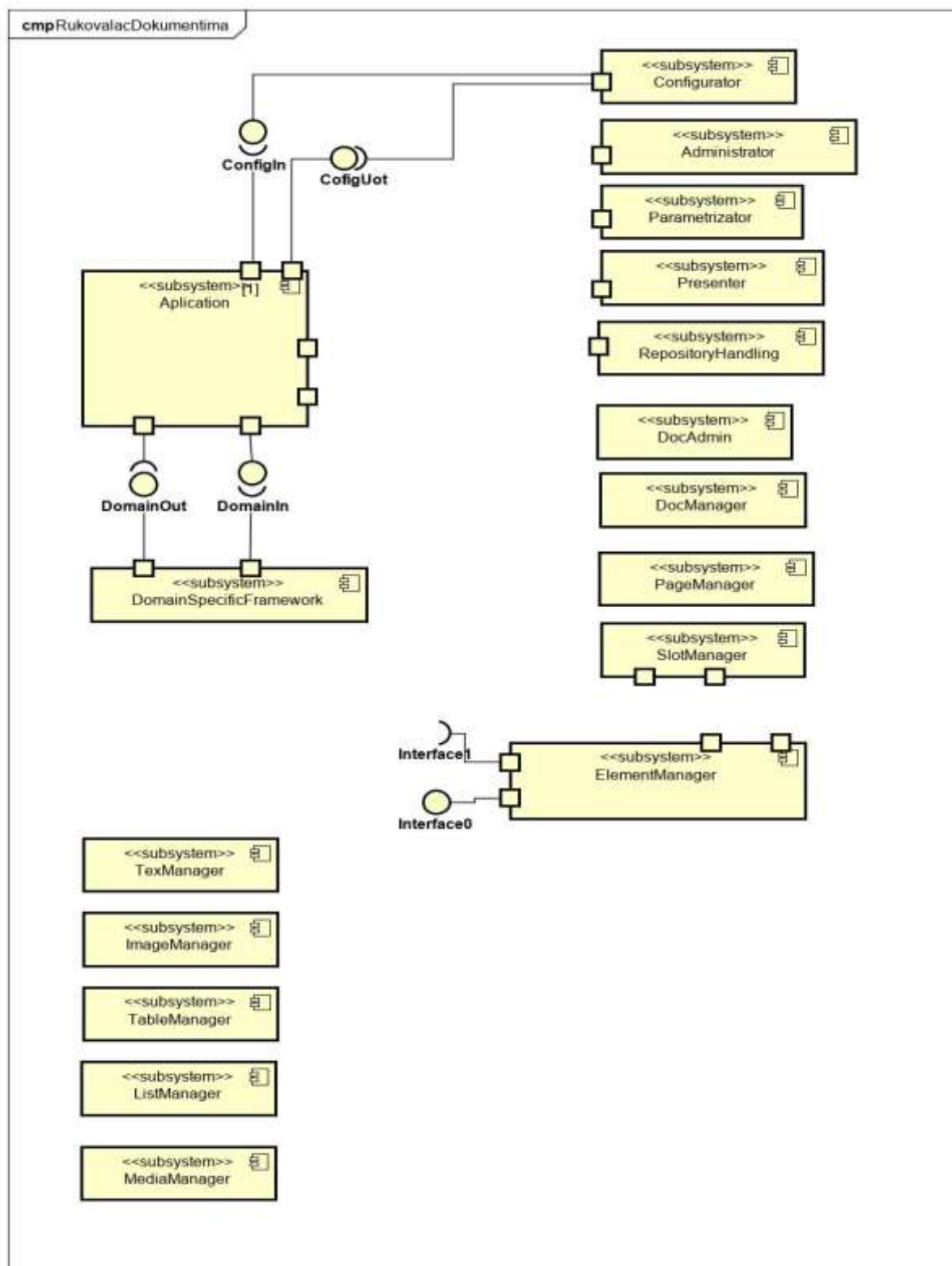
Grativne elemente aplikativnog radnog okvira je potrebno organizovati u sklopu strukture namenskih kataloga (mogući primer):



6.3.2. Primer - Objektni model Arhitekture Aplikacije (Aplikativni radni okvir)

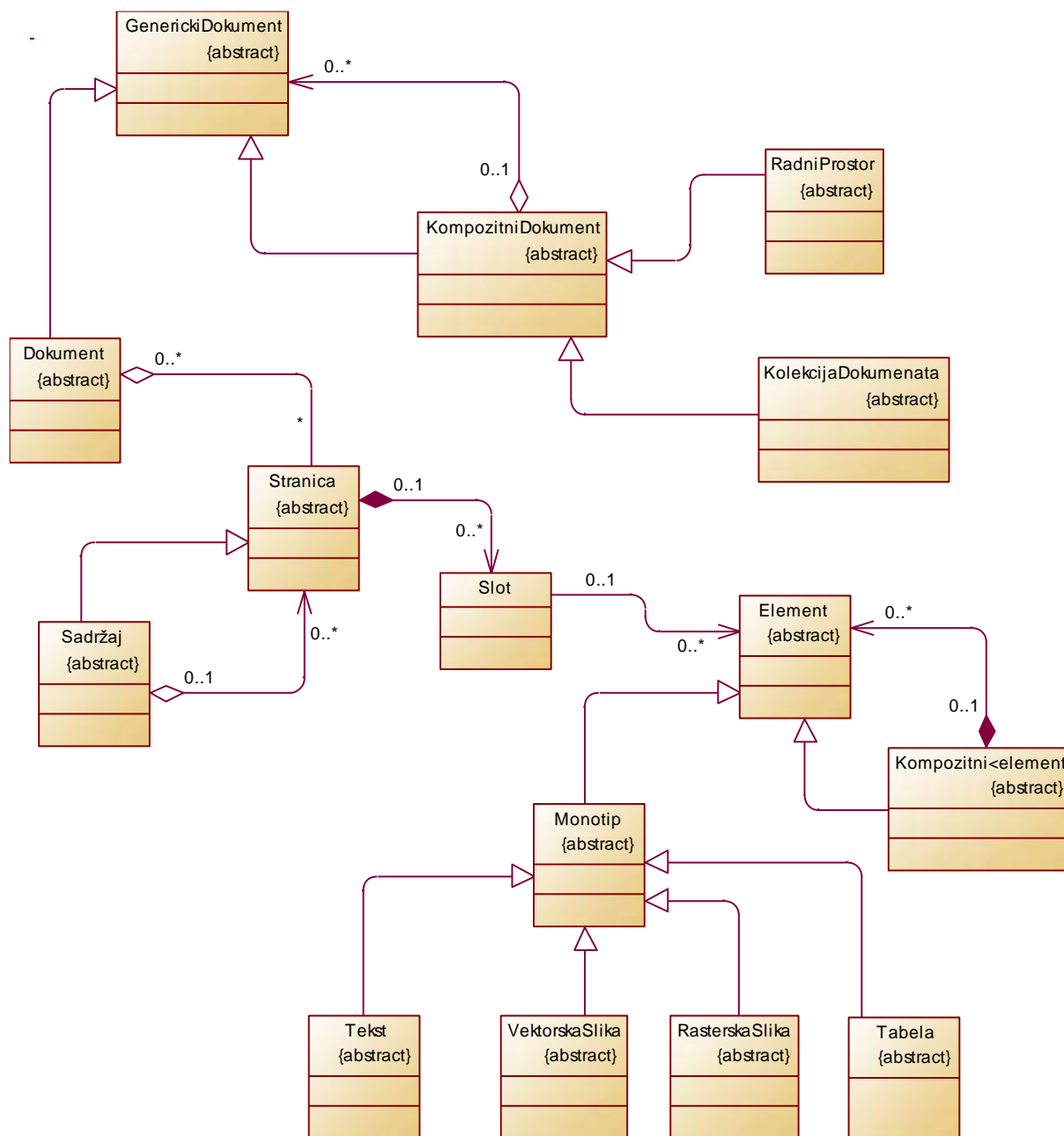


6.3.3. Primer - Dijagram komponenti



6.4. Model statičke strukture komponenti - UML Dijagram KLASA

6.4.1. Primer nacrtu modela DOKUMENTA



6.5. Elementi primopredaje

6.5.1. Power Designer projekat - Rukovalac dokumentima (Izvezen u HTML dokument)

6.5.2. Specifikacija zahteva - PowerDesigner DataArchitect PD16 - model zahteva prema opisanom standardu

Zahteve treba upotpuniti Poslovnim i Sistemskim slučajevima upotrebe (UML USE CASE dijagrami). Neophodno je jasno definisati veze između pojedinačnih zahteva i odgovarajućih UML USE_CASE dijagrama.

6.5.3. Model Arhitekture - PowerDesigner - DataArchitect PD16.

Oslonac na podsisteme, komponente , portove i interfejsse komponenti.
Formiranje Aplikativnog Radnog Okvira .

Modelovanje saradnje komponenti uz oslonac na specifikaciju interfejsa, modelovanje saradnje (definisanje protokola saradnje komponenti), definisanje razmene objekata između komponenti arhitekture (modelovanje razmene poruka - dijagrami sekvence)

6.5.4. Modeli Komponenti - PowerDesigner - DataArchitect PD15. - objektni (klasni) model

Oslonac na Modelovanje statičke strukture (klasni dijagrami) sa upotrebom MVC arhitektonskog šablona i odgovarajućih Dizajnerskih šablona. Specifikacija i modelovanje veza komponenti sa okruženjem (Interfejsi, Implementacija interfejsa).

Oslonac na modelovanje ponašanja. Dominantno unutrašnju dinamiku bazirati na upotrebi dijagrama stanja i šablona koji ga prate (State, Command, Structure, Factory, . . .). Pojedinačne komponente moraju slediti formu komponent radnog okvira. Unutrašnja arhitektura MVC sa upotrebom Observera. Unikat koristiti za enkapsulaciju deljenih elemenata unutar pojedinačnih komponenti.

Oslonac na dijagrame aktivnosti za specifikaciju saradnje objekata unutar komponenti.

6.5.5. Izvršna verzija aplikacije - Rukovalac dokumentima

U sklopu korišćenog sistema za upravljanje verzijama.

6.5.6. Dokumentovanje aplikacije - Rukovalac dokumentima

ON_LINE - Operativno uputstvo (Help) i About box.

OF_LINE - Elektronska forma operativne dokumentacije (izvoz u pdf)