



Softversko inženjerstvo

Elektronski fakultet Niš

Metode inženjeringa zahteva



Elektronski fakultet u Nišu



Metode inženjeringa zahteva (RE)

- Procesi inženjeringa zahteva su obično “vođeni” odgovarajućom metodom.
- Metode inženjeringa zahteva definišu sistematske načine za dobijanje modela sistema.



Elektronski fakultet u Nišu



Elektronski fakultet u Nišu



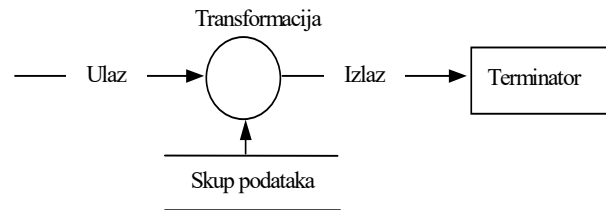
Osnovne karakteristike RE metode

- Pogodnost za usaglašavanje sa krajnjim korisnikom.
- Preciznost definicije i odgovarajuće notacije.
- Pomoć kod formulisanja zahteva.
- Fleksibilnost.
- Podržanost odgovarajućim SW alatima.

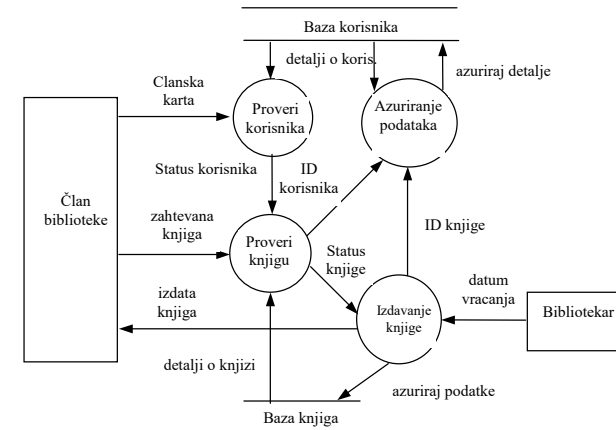
Najpoznatije RE metode

- DFD (Data-Flow-Diagram) dijagrami
- Metode koje koriste relacioni model podataka
- Objektno-orijentisane metode
- Formalne metode
- Metode zasnovane na ponašanju sistema
 - Use-case specifikacija
 - Viewpoint orijentisane metode

DFD dijagram

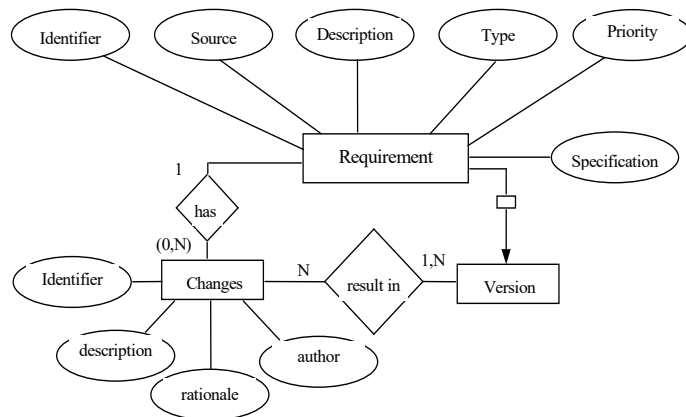


DFD dijagram – primer biblioteke



Metode koje koriste relacioni model

- Za opis zahteva se koristi prošireni ER model:



Formalne metode

- Polu-formalne i neformalne metode
 - Koriste prirodni jezik, dijagrame, tabele i prostu notaciju.
 - Uključuju strukturnu analizu i OO analizu.
- Formalne metode
 - Bazirane su na matematički formalnoj sintaksi i semantici.
 - Poznate metode: Z, B, VDM, LOTOS



Formalne metode

- Obezbeđuju načine za postizanje visoke sigurnosti da će sistem odgovarati specifikaciji zahteva.
- Ne garantuju apsolutnu korektnost sistema.



Komponente jezika formalne specifikacije

- *Sintaksa* - definiše specifičnu notaciju za zapis zahteva.
- *Semantika* - definiše objekte i njihovo značenje za opis sistema.
- *Relacije* - definišu pravila koja određuju koji objekat zadovoljava specifikaciju na pravi način.



Formalne metode - upotreba

- Nisu mnogo zaživele među ljudima koji razvijaju SW.
- Razlozi:
 - Teško razumevanje i učenje notacije.
 - Problemi formalizovanja pojedinih aspekata zahteva.
 - Učinak nije očigledan.



Primer Z metode za bibliotečki sistem

New
$\Delta \text{Library}$ book?: Book
$\text{stock}' = \text{stock} \cup \{\text{book?}\}$ $\text{onLoan}' = \text{onLoan}$

Return
$\Delta \text{Library}$ book?: Book
$\text{book?} \in \text{dom onLoan}$ $\text{dom onLoan}' = \text{dom onLoan} \oplus \text{book?}$ $\text{stock}' = \text{stock}$



Objektno-orijentisane metode

- Nasledile su ER model
- Bazirane su na sledećim metodama:
 - Shlaer and Mellor (1988)
 - Colbert (1989)
 - Coad and Yourdon (1989)
 - Wirf-Brock (1990)
 - Rumbaugh (1991)
 - Jacobson (1992)
 - Martin-Odell (1992)
- Notacije su različite za različite metode, ali je semantika slična.



Objekti i prikupljanje zahteva

- U procesu prikupljanja zahteva, osnovni elementi koji se posmatraju su **objekti**.
- Objekti se dobijaju analizom domena problema.
- Objekti uključuju:
 - Uređaje sa kojima sistem intereaguje
 - Sisteme koji sarađuju sa sistemom koji se razvija
 - Organizacione jedinice
 - Stvari o kojima se moraju čuvati podaci
 - Fizičke lokacije
 - Specifične uloge ljudi



Osnovni koncepti

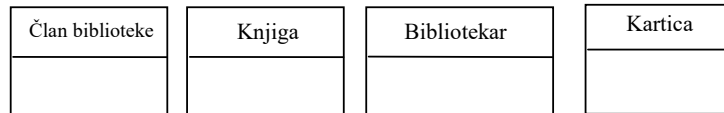
- Enkapsulacija
- Klase
- Nasleđivanje
- Atributi i servisi
- Poruke



Koraci kod OO metoda

- Većina OO metoda ima sledeće zajedničke korake u modeliranju sistema i prikupljanju zahteva:
 - Identifikacija osnovnih objekata (klasa)
 - Definisane objektna strukture i veza između klasa
 - Definisane atributa i servisa objekata
 - Definisane poruke koje objekti razmenjuju

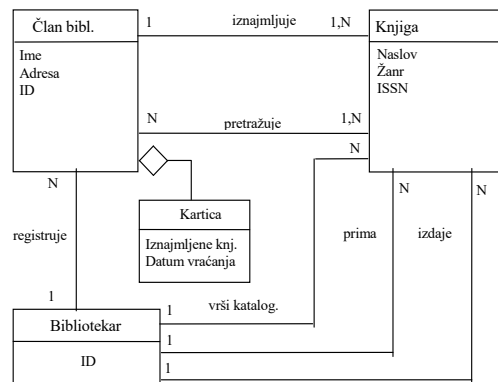
Korak 1 – Identifikacija osnovnih klasa



Korak 2 – Veze između klasa

- Mogu se identifikovati veze između sledećih zahteva:
 - (i) Član biblioteke **iznajmljuje** knjigu
 - (ii) Knjiga **je** publikovana od strane izdavača
 - (iii) Administrator sistema **registruje** člana
 - (iv) Članovi **su** studenti, osoblje i spoljni korisnici
 - (v) Administrator sistema **vrši katalogizaciju** knjiga
 - (vi) Bibliotekar **izdaje** knjige

Korak 2 – Klase i veze



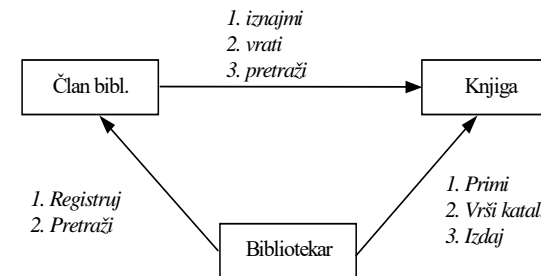
Korak 3 – Atributi objekata

- Atributi se dobijaju analizom zahteva
- Na primer, ako je zahtev da svi članovi moraju biti registrovani pre iznajmljivanja knjige:
 - to znači da se mora čuvati datum registracije člana
 - svaki član mora imati karticu kako bi se beležila sva izdavanja i vraćanja knjige
- Knjiga mora da ima attribute: Naslov, žanr i ISSN
- Član biblioteke mora da ima attribute: Ime i prezime, adresu i ID

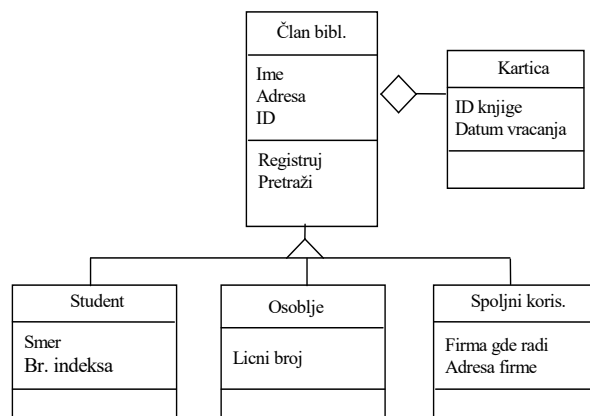
Korak 4 – Servisi objekata

- U ovom koraku se vrši identifikacija servisa
- Svakako mora da postoje servisi za pristup i izmenu atributa:
 - Get-Set servisi
- Jedan način identifikovanja servisa je praćenje poruka koje se razmenjuju između objekata.

Korak 4 – Razmena poruka



Korak 4 - Servisi



Metode zasnovane na ponašanju sistema

- Use-case specifikacija i scenariji ponašanja
- Viewpoint orijentisane metode



Use-case metoda

- Sistem se posmatra sa stanovišta korisnika sistema.
- Opisuju se slučajevi korišćenja sistema i scenariji ponašanja.
- Koristi se UML notacija.
- Koristi se kod RUP modela razvoja SW-a.



Use-case i scenariji događaja

- Servisi objekata mogu biti otkriveni i modeliranjem scenarija događaja za različite funkcije sistema.
- Događaji se prate do objekata koji reaguju na njih.
- Tipičan model scenarija događaja je interakcija između korisnika i sistema.



Scenarija

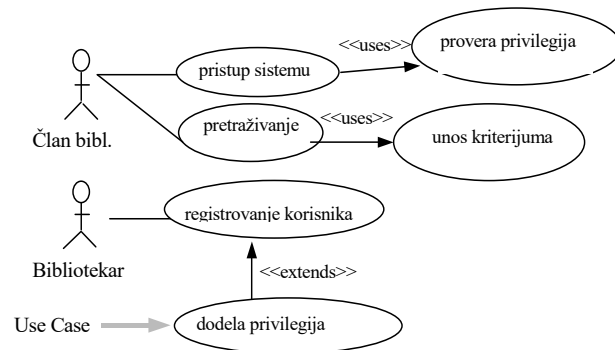
- Scenariji su primeri kako će sistem biti korišćen u realnom životu.
- Oni bi trebalo da uključe:
 - Opis početne situacije;
 - Opis normalnog toka događaja;
 - Opis izuzetaka (ako se izađe iz normalnog toka);
 - Informacije o konkurentnim aktivnostima;
 - Opis stanja gde se scenario završava.



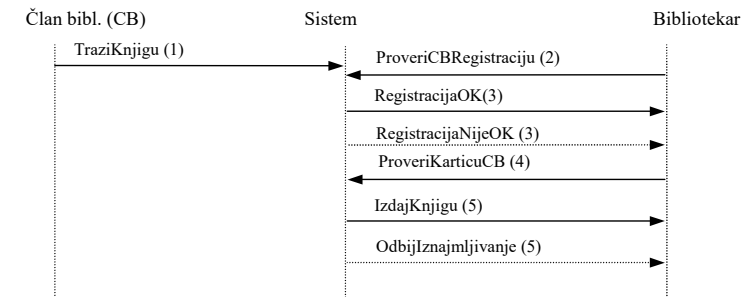
Slučajevi korišćenja (Use cases)

- Slučajevi korišćenja predstavljaju tehniku baziranu na scenarijima i zapisanu pomoću UML dijagrama koja identifikuje **aktere** u sistemu i **slučajeve korišćenja** sistema.
- Potpuni skup slučajeva korišćenja opisuje sve moguće interakcije korisnika sa sistemom.
- UML dijagrami sekvenci mogu biti korišćeni za detaljni opis slučajeva korišćenja. Oni daju sliku o obradi događaja u sistemu za izabrani slučaj korišćenja.

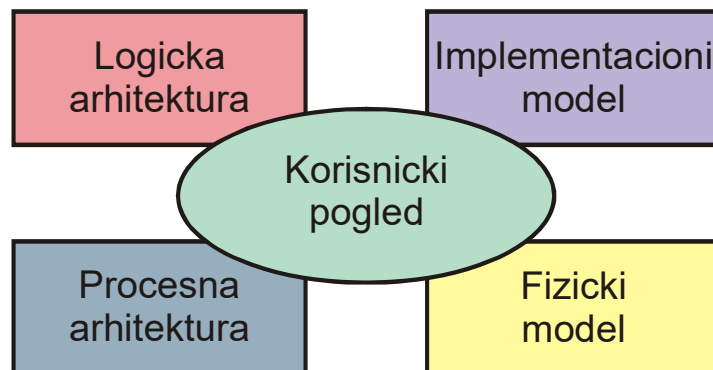
Tipičan use-case dijagram za primer bibliotečkog sistema



Scenario događaja za iznajmljivanje knjige



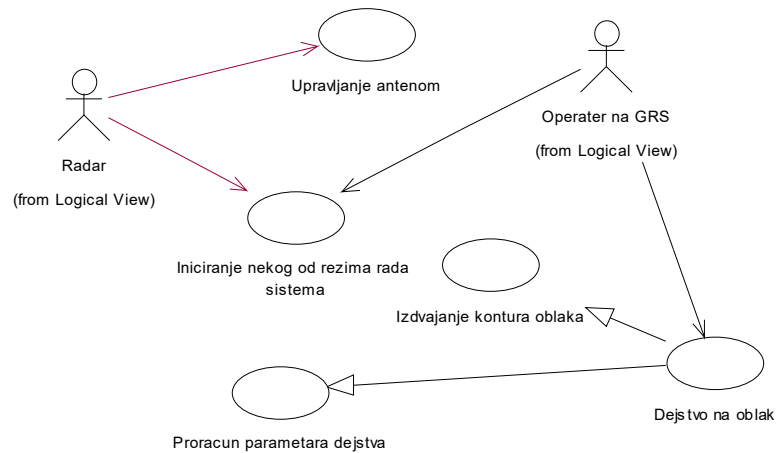
RUP - “4+1” Model sistema



RUP - Use-case specifikacija

- Use-case specifikacija je izlaz iz faze razrade (elaboracije) i sadrži:
 - Opis slučajeva korišćenja
 - Definisanje aktera u sistemu
 - Određivanje arhitekturno najznačajnijih slučajeva korišćenja

Slučajevi korišćenja – primer informacionog sistema odbrane od grada

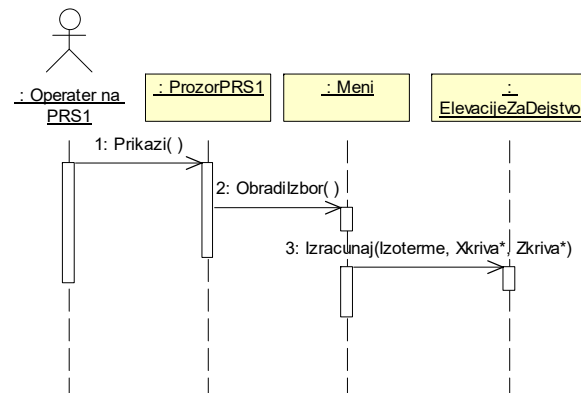


Opis slučaja korišćenja pomoću tabela

Naziv	Proračun elevacija za dejstvo	R. br. događaja	Akcija aktera	Reakcija sistema
Akteri	Operator PRS1	1.	Ovaj slučaj korišćenja inicira operator PRS1 tako što izda komandu za izračunavanje elevacija.	
Svrha algoritma	Proračun elevacija za lansiranje raketa	2.		Sistem daje informaciju o dužini trajanja proračuna a zatim vrši izračunavanje i na kraju izdaje poruku o završetku izračunavanja.
Opis	Nakon unosa novog sinoptičkog biltena u sistem potrebno je startovati izračunavanje novih elevacija za lansiranje raketa. Izračunavanje se vrši samo jednom nakon svake promene sinoptičkog biltena.			

RUP - Realizacija slučaja korišćenja

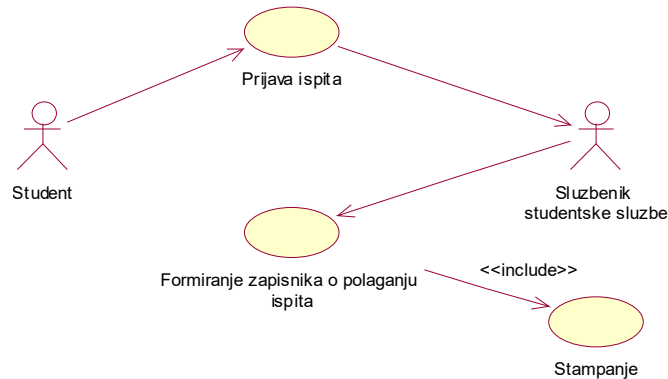
- Dijagrami sekvenci



Primer – IS fakulteta

- Na primer, treba napisati detaljnu use-case specifikaciju koja se odnosi na segment prijave ispita informacionog sistema fakulteta.

Primer – IS fakulteta



Primer – IS fakulteta

- Use-case: Prijava ispita.
- Kratak opis: Prijava ispita na osnovnim studijama.
- Akteri: Student, Službenik studentske službe.
- Preduslovi: Student je odslušao predavanja, odradio laboratorijske vežbe i predao projekat iz predmeta za koji se prijavljuje da polaže ispit.

Primer – IS fakulteta

- Opis:
 1. Student popunjava prijavu (ime, prezime, predmet, ...).
 2. Službenik prima prijavu [izuzetak: *pogrešni podaci u prijavi*].
 3. Službenik unosi podatke o prijavljenom ispitu u dosije studenta na računaru.
- Izuzeci:
 - [Pogrešni podaci u prijavi] Prijava se vraća studentu da unese tačne podatke.
- Posledice: Student je evidentiran za polaganje ispita iz odgovarajućeg predmeta.

Primer – IS fakulteta

- Use-case: Formiranje zapisnika o polaganju ispita.
- Kratak opis: Službenik prosleđuje zahtev za formiranje zapisnika o polaganju ispita na osnovu prijava studenata.
- Akteri: Službenik.
- Preduslovi: Završen rok za prijavu ispita.



Primer – IS fakulteta

- Opis:
 1. Službenik inicira izvršavanje funkcije formiranja zapisnika o polaganju ispita.
 2. Sistem prikazuje formu za unos predmeta za koji se formira zapisnik.
 3. Službenik unosi predmet.
 4. Sistem formira zapisnik i inicira štampanje.
 5. Izvršava se niz akcija definisanih use-case-om Štampanje.
- Izuzeci: -
- Posledice: Zapisnik je formiran pre datuma polaganja ispita.



Primer – IS fakulteta

- Use-case: Štampanje
- Kratak opis: Štampanje različitih dokumenata (uverjenja, zapisnika o polaganju ispita, ...)
- Akteri: -
- Preduslovi: Štampač je uključen i povezan sa računarom.



Primer – IS fakulteta

- Opis:
 1. Sistem prosleđuje zahtev za štampanje dokumenta.
 2. a) Ukoliko je štampač slobodan, zahtev se prosleđuje štampaču.
 3. b) Ukoliko nije, zahtev se stavlja na red čekanja, a kada dođe na red on se prosleđuje štampaču.
 4. Kada zahtev stigne do štampača, dokument se štampa.
[Izuzetak: Nema papira u štampaču]
[Izuzetak: Nema tonera]



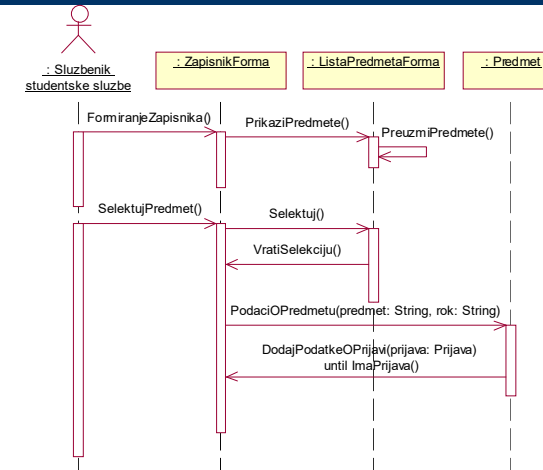
Primer – IS fakulteta.

- Izuzeci:
 - [Nema papira u štampaču] Neophodno je staviti papir.
 - [Nema tonera] Neophodno je isključiti štampač i promeniti toner, a zatim ponovo proslediti zahtev za štampanje dokumenta.
- Posledice: Kompletan dokument je odštampan.

Primer – IS fakulteta

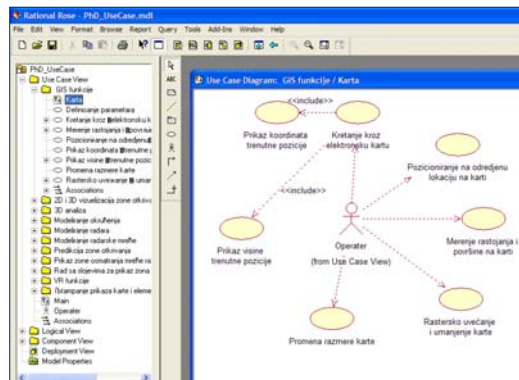
- Dijagramom sekvenci se vrši opis scenarija za formiranje zapisnika o polaganju ispita koji se formira na zahtev službenika administracije.

Primer – IS fakulteta



SW alati za Use-case metode

- IBM Rational Rose
- Microsoft Visio
- ...



Viewpoint orijentisane metode za inženjering zahteva

- Osnovna ideja: Sistem se posmatra sa različitih tačaka (aspekata) posmatranja (iz ugla različitih korisnika sistema) i na taj način se pokušava da se sistem sagleda sa svih strana i da se identifikuju svi validni zahtevi, kao i da se razreše kontradiktorni zahtevi.



Primer sistema za vozove

- Neka je u pitanju sistem koji se instalira u vozu i koji automatski zaustavlja voz ukoliko on pogrešno prođe na crveno svetlo.
- Neki primeri viewpoint-a za ovaj sistem i odgovarajućih zahteva bili bi:
 - *Vozač* – Zahtevi sa aspekta posmatranja vozača.
 - *Oprema na pruzi* – Zahtevi sa strane opreme na pruzi sa kojom novi sistem mora da sarađuje.
 - *Inženjer sigurnosti* – Zahtevi vezani za sigurnost sistema.
 - *Postojeći on-board sistem u vozu* – Zahtevi vezani za kompatibilnost sa ovim sistemom.
 - *Kočnice* - Zahtevi koji su vezani za karakteristike kočnica u vozu.



Prednosti viewpoint orijentisanih metoda

- Eksplicitno otkrivaju sve izvore zahteva za sistem.
- Obezbeđuju mehanizam organizovanja i struktuiranja svih zahteva.
- Pružaju osećaj potpunosti (kompletnosti).



Tipovi viewpoint-a

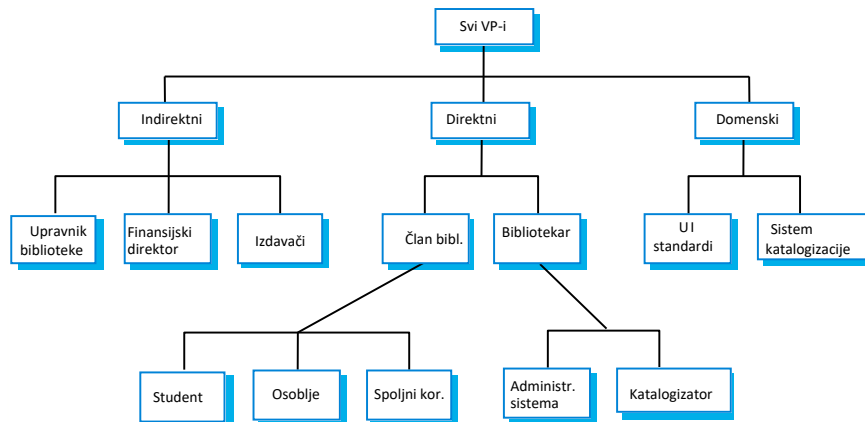
- Direktni aspekti posmatranja (interactor viewpoints)
 - Ljudi ili drugi sistemi koji imaju direktnu interakciju sa sistemom. Na primer, kod sistema ATM bankomata, korisnici bankomata i baza bankovnih računa korisnika su direktni VP-i.
- Indirektni aspekti posmatranja (Indirect viewpoints)
 - Korisnici koji ne koriste sistem direktno, ali imaju uticaja na zahteve sistema. Na primer, kod sistema ATM bankomata, menadžment i radnici obezbeđenja su indirektni VP-i.
- Domenski aspekti posmatranja (Domain viewpoints)
 - Ograničenja i karakteristike domena koje imaju uticaja na zahteve. Na primer, kod sistema ATM bankomata, standardi za komunikaciju između banaka bi bili domenski VP-i.



Identifikacija viewpointa

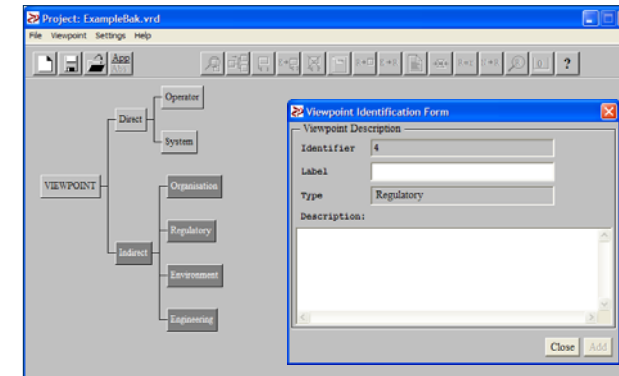
- Viewpointe treba tražiti među:
 - Pružaocima i korisnicima servisa sistema;
 - Sistemima sa kojima novi sistem treba da intereaguje;
 - Propisima i standardima;
 - Izvorima poslovnih i nefunkcionalnih zahteva.
 - Inženjerima koji će razviti i održavati sistem;
 - Marketinškim i ostalim poslovnim VP-ima.

Hijerarhija VP-a za bibliotečki sistem



SW alati za viewpoint metode

- VORD (I. Sommerwille)



SW alati za viewpoint metode

- VORD (I. Sommerwille)

