

## Projektovanje funkcionalnosti

Metodi funkcionalne dekompozicije:

- Funkcionalna dekompozicija I vrste (FD1)
- Dijagrami tokova podataka (DTP)
- Funkcionalna dekompozicija II vrste (FD2)

Suština metoda:

- Dekomponovanje funkcionalnosti na podfunkcije sa različitim nivoom detalja (svi metodi)
- Dekomponovanje podataka i tokova (DTP, FD2)
- Redosled, uslovljenost, ponavljanje (FD2)

## Projektovanje funkcionalnosti – FD1

Funkcionalna dekompozicija I vrste (FD1):

- Funkcionalnosti se dekomponuju na skup potencijalnih podfunkcija bez ikakvih dodatnih detalja.

Kod svih sistema u prvom koraku se dobija:

**KONKRETNI\_SISTEM**

OdrzavanjeMaticnihPodataka

ObradaPrometa

Izvestavanje

Dalja dekompozicija je specifična za svaki sistem.

# SI3IS1 2.1.3

C

## Projektovanje funkcionalnosti – FD1

### Primer FD1 - BIBLIOTEKA:

#### BIBLIOTEKA

OdrzavanjeMaticnihPodataka

ObradaPrometa

Izvestavanje

#### BIBLIOTEKA

OdrzavanjeMaticnihPodataka

ObradaPrometa

ObradaTraznjeNaslova

ObradaVracanjaKnjige

ObradaGubitkaKnjige

Izvestavanje

# SI3IS1 2.1.3

d

## Projektovanje funkcionalnosti – FD1

U daljoj dekompoziciji (samo po jednoj grani):

BIBLIOTEKA

...

ObradaPrometa

ObradaTraznjeNaslova

OcitavanjeNeophodnihPodataka

ProveraVecDrzi

ProveraImaSlobodno

EvidentiranjeIzdavanja

EvidentiranjeRezervacije

...

...

## Projektovanje funkcionalnosti – DTP

Metod dijagrama tokova podataka čine:

- Hijerarhijski skup dijagrama (stablo):
- Skup simbola
- Skup pravila formiranja
- Notacija za opis sadržaja

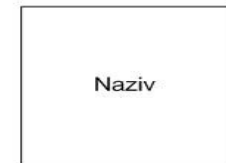
Hijerarhijski skup dijagrama (stablo) čine:

- \* dijagram konteksta (DC)
- \* dijagram 0 - nastaje dekompozicijom DC
- \* dijagrami procesa - nastaju dekompozicijom procesa

# Projektovanje funkcionalnosti – DTP

## Skup simbola čine:

**KONEKTOR:** Učesnik iz okruženja.  
Od i ka njemu idu tokovi podataka  
ka procesima.



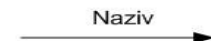
**PROCES:** Aktivna komponenta.  
Od i ka njemu idu tokovi podataka ka procesima,  
skladištima podataka I konektorima.  
Može sadržati procese i skladišta podataka.



**SKLADIŠTE:** Pasivna komponenta.  
Od i ka njemu idu tokovi podataka ka procesima.  
Mogu se dekomponovati.



**TOK:** Prenosnik podataka.  
Može se rastavljati I spajati..



## Projektovanje funkcionalnosti – DTP

Skup pravila formiranja čine:

- Pravilo očuvanja pri dekompoziciji procesa: Nastali dijagram procesa mora da sadrži sve nasleđene tokove podataka prvobitnog procesa.
- Pravilo očuvanja toka podataka. Tokovi nastali rastavljanjem toka moraju zbirno da imaju prvobitni sadržaj toka. Isto važi i za sastavljanje tokova.
- Pravilo očuvanja pri dekompoziciji skladišta podataka: Nastala skladišta podataka moraju zbirno da imaju prvobitni sadržaj.

## Projektovanje funkcionalnosti – DTP

Notacija za opis sadržaja tokova i skladišta je zasnovana na sintaksoj notaciji sa zagradama:

**Skup** ::= {Tip}

**Agregacija** ::=  $\leq$ Tip Tip, ..  $\geq$

**EkskluzivnaSpecijalizacija** ::= Tip[Tip], ..

**InkluzivnaSpecijalizacija** ::= Tip[Tip], ..

**Tip** ::= Element | Skup | Agregacija |

    EkskluzivnaSpecijalizacija

    InkluzivnaSpecijalizacija

Za **Element** se definiše kog je ogradenog tipa.



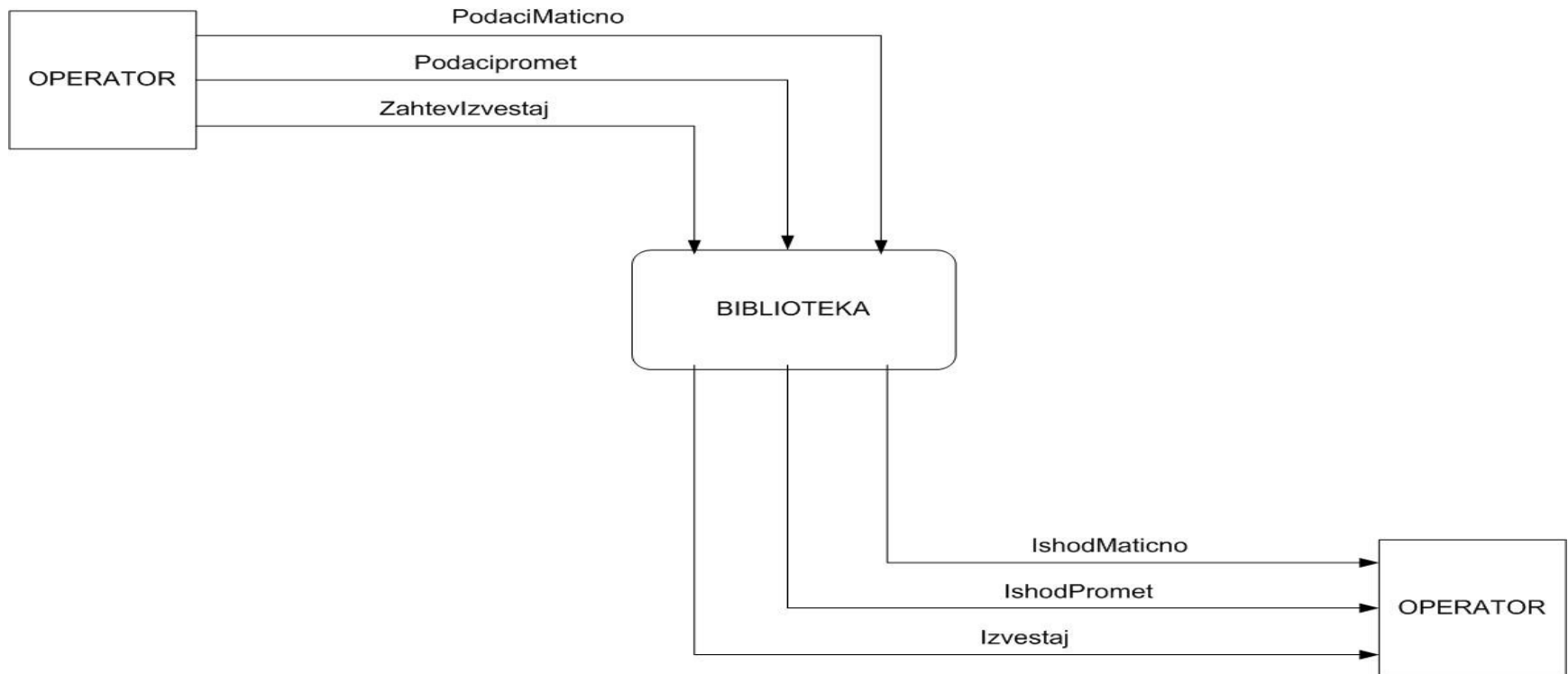
# SI3IS1 2.1.3

i

## Projektovanje funkcionalnosti – DTP

### DTP:

Dijagram konteksta – prikazuje odnos sistema i okruženja



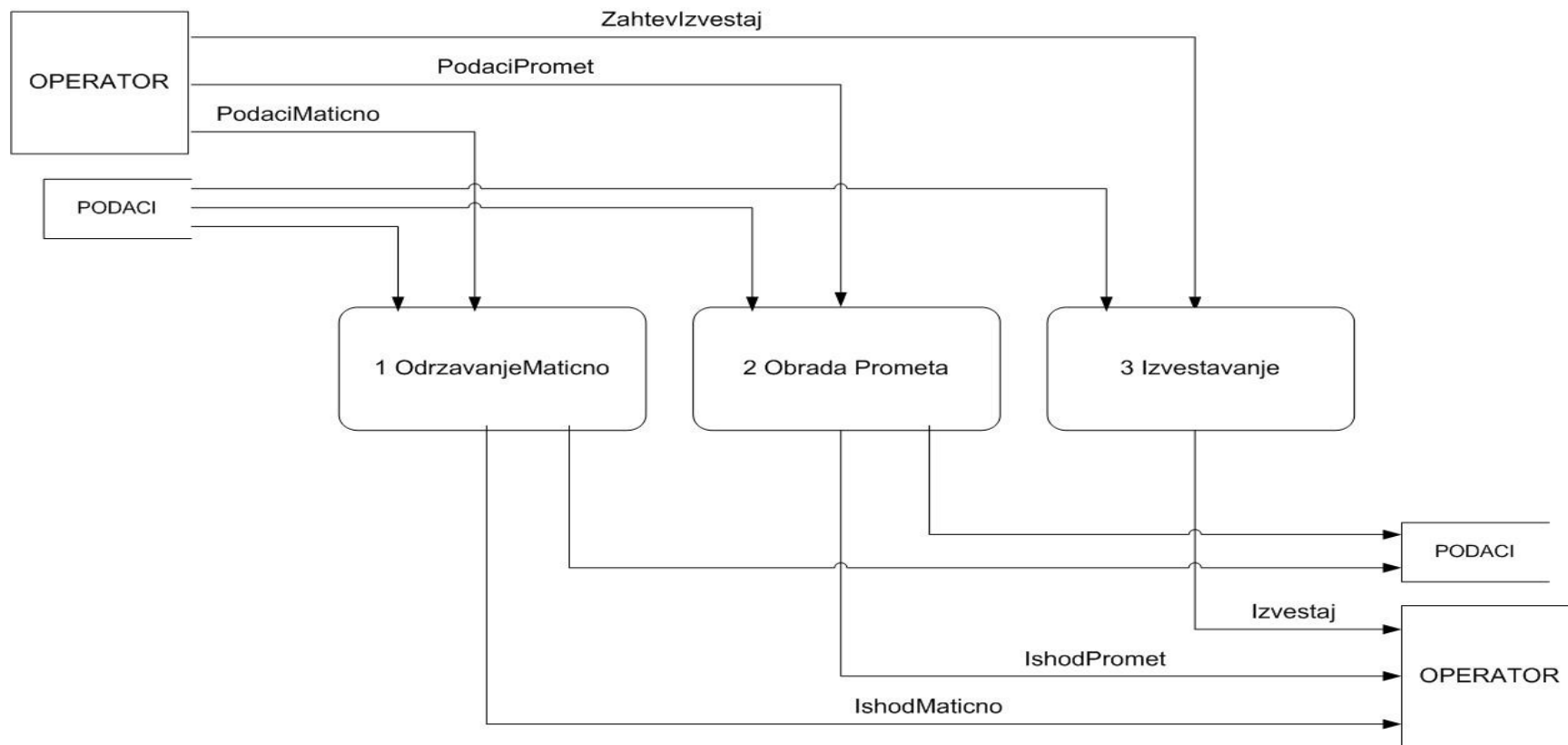
# SI3IS1 2.1.3

j

## Projektovanje funkcionalnosti – DTP

### DTP:

Dijagram 0 – nastaje dekompozicijom dijagrama konteksta



## Projektovanje funkcionalnosti – FD2

Funkcionalna dekompozicija II vrste (FD2):

- Funkcionalnosti se dekomponuju na skup potencijalnih podfunkcija sa dodatnim detaljima:
  - \* redosled                      **n** redni broj
  - \* uslovljenost                **?** ispred rednog broja i  
                                      **/Uslov** iza naziva podfunkcije
  - \* isključivost                **e** iza rednog broja
  - \* ponavljanje                **\*** ispred rednog broja i  
                                      **/Opis** iza naziva podfunkcije

Dve varijante FD2:

- Neanotirana FD2
- Anotirana FD2

# SI3IS1 2.1.3

|

## Projektovanje funkcionalnosti – FD2

### Primer neanotirane FD2 - BIBLIOTEKA:

#### BIBLIOTEKA

- 1e OdrzavanjeMaticnihPodataka

- 1e ObradaPrometa

  - 1e ObradaTraznjeNaslova

  - 1e ObradaVracanjaKnjige

  - 1e ObradaGubitkaKnjige

- 1e Izvestavanje

## Projektovanje funkcionalnosti – FD2

U daljoj dekompoziciji (samo po jednoj grani):

### BIBLIOTEKA

...

1e ObradaPrometa

1e ObradaTraznjeNaslova

1 OcitavanjeNeophodnihPodataka

2 ProveraVecDrzi

?3 ProveraImaSlobodno /NeDrzi

?4e EvidentiranjeIzdavanja /NeDrzi^Ima

?4e EvidentiranjeRezervacije /NeDrzi^Nema  
^ZeliRezervaciju

...

...

## Projektovanje funkcionalnosti – FD2

Anotirana FD2:

Uz funkcionalnosti se u zagradama naznačuju:

- imenovani i usmereni tokovi podataka:

=> I.Tok    ulazni tok od okruženja (na vrhu samo I)

<= O.Tok    izlazni tok ka okruženju (na vrhu samo I)

> Tok        ulazni tok ka funkcije od ostatka sistema

< Tok        izlazni tok od funkcije ka ostatku sistema

<> Tok        ulazno-izlazni tok funkcije ka ostatku sistema

## Projektovanje funkcionalnosti – FD1

Anotirana FD2:

- imenovani i usmereni pristupi podacima IS:
  - > D.Tok ulazni tok ka podacima (na vrhu samo D)
  - < D.Tok izlazni tok od podataka (na vrhu samo D)
  - <> D.Tok ulazno-izlazni tok (na vrhu samo D)

I kod tokova i kod podataka simbol . označava odnos sadržanosti – ono iza je na bilo koji način komponenta onog ispred.

Pri dekompoziciji mora biti ispoštovana usaglašenost.

## Projektovanje funkcionalnosti – FD2

### Anotirana FD2:

- Kada je razvijen i model podataka, notacija odnosa sa podacima se zamenjuje detaljnijom notacijom odnosa sa entitetima modela podataka, gde uz naziv entiteta ide:

C:Entitet kreiranje instance

R:Entitet uvid u instancu/klasu

U:Entitet izmena instance

D:Entitet uništavanje instance

- Moguće su i kombinacije više odnosa

RU:Entitet , CDU:Entitet ...



## Projektovanje funkcionalnosti – FD2

### Anotirana FD2:

- Kada je razvijen i relacioni model , notacija odnosa sa podacima se zamenjuje detaljnijom notacijom odnosa sa tabelama baze podataka, gde uz naziv relacije ide:

I:Tabela	ubacivanje reda	INSERT
S:Tabela	uvid u reda/tabelu	SELECT
U:Tabela	izmena reda	UPDATE
D:Tabela	brisanje reda	DELETE

- Moguće su i kombinacije više odnosa  
SU:Tabela , IDU:Tabela ...