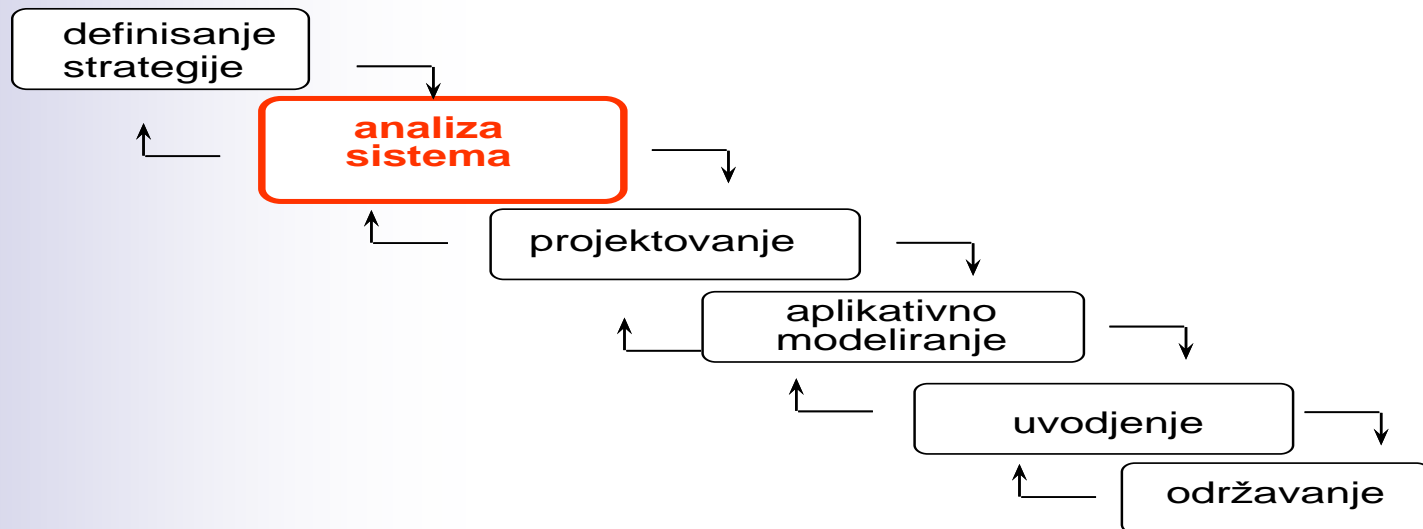


# Analiza sistema



# Funkcionalni model sistema

- predstavlja sistem kao "crnu kutiju"
- prikazuje se funkcionalnost sistema na način kako je vide spoljni objekti
- predstavljaju se ulazi i izlazi iz sistema i funkcije koje transformišu ulaze (pobudu, stimulaciju) u izlaze
- predstavlja model **zahteva**, jer treba da pokaže potpuno, precizno i nedvosmisleno kako će objekti van sistema (korisnici, akteri) koristiti posmatrani sistem.


# Strukturalna sistem analiza (SSA)

- Potpuna konvencionalna metodologija za specifikaciju informacionog sistema
- Nastala je kao odgovor na problem neadekvatne specifikacije zahteva korisnika pomoću klasičnih sredstava funkcionalne analize
- Daje jasan i detaljan opis sistema, primenom metode apstrakcije, tako da se **sistem na višim nivoima apstrakcije opisuje opštije, a na nižim detaljno;**
- Predstavlja logičku, a ne fizičku specifikaciju procesa - specifikacija opisuje šta će budući sistem raditi i šta će pružati korisniku, a ne kako će biti implementiran;

# Strukturalna sistemska analiza (SSA)

## ■ Okvir SSA?

- Jasna grafička specifikacija, pogodna za komunikaciju sa korisnikom;
- SSA posmatra informacioni sistem kao funkciju (proces obrade) koja, na bazi ulaznih, generiše izlazne podatke.
- Ulazni podaci se dovode u proces obrade, a izlazni iz njega odvođe preko tokova podataka.
- Od interesa su samo sadržaj i struktura ulaznog toka, a ne i medijum - nosilac toka.



Primenom metode SSA se dobija model procesa posmatranog sistema, koji treba da da odgovor na sledeća pitanja:

- Koje funkcije poseduje realni sistem, odnosno koje funkcije mora imati budući IS?
- Kakve veze postoje između funkcija?
- Kakve transformacije treba sistem da izvrši?
- Koje ulaze (podatke) procesom obrade transformisati u koje izlaze?
- Koje zadatke treba da izvrši sistem?
- Odakle sistem povlači informacije potrebne za izvršavanje zadataka?
- Kuda odlaze rezultati obrade?

# Osnovni koncepti SSA

- Proces (process)
- Tok podataka (data flow)
- Skladište podataka (data store)
- Spoljni objekat (interface - external agent)
- Spojevi (junctions)

# Proces (process)

obrada  
naloga

- Proces obrade podataka je aktivna komponenta sistema, koja vrši transformaciju strukture i sadržaja ulaznog toka u izlazni tok.
- Svaki proces ima svoju referencu i svoj naziv.
- Reference služe za praćenje dekompozicije procesa, a ne za definisanje redosleda izvršavanja.
- Naziv procesa treba precizno da označava funkciju koju on obavlja. Najčešće je u obliku glagolske fraze.

*(ako analitičar ne zna da dodeli ime procesu, znači da ne razume funkciju koju proces obavlja).*

# Tok podataka (data flow)

- Tok podataka je vod kroz koji protiču grupe podataka poznate strukture.
- Tok podataka ostvaruje vezu između ostalih komponenti sistema i na dijagramu toka podataka predstavlja se imenovanom, orijentisanom linijom.
- Svaki tok podataka mora imati jedinstveno ime koje odražava značenje podataka koje on nosi. Izuzetak su tokovi koji idu prema, odnosno od skladišta podataka, a koriste celu strukturu tog skladišta. Oni tada ne moraju biti imenovani.
- Interni i eksterni tokovi podataka.



potpisan ugovor



# Skladište podataka (data store)

- Skladište podataka (kartoteka, fascikla, datoteka) služi za čuvanje podataka.
- Definiše se kao tok podataka u mirovanju.
- Predstavljaju interna stanja posmatranog sistema
- Veza sa primitivnim funkcijama isključivo preko tokova podataka.
- **Skladište podataka treba da omogući:**
  - tok podataka KA skladištu označava operaciju održavanja, tj. ubacivanja, izbacivanja i promene sadržaja;
  - tok podataka OD skladišta označava korišćenje podataka skladišta u smislu čitanja podataka i za izveštavanje.

# Spoljni objekat - interface

- Spoljni objekat (interfejs) je objekat koji se nalazi izvan posmatranog sistema a sa kojim informacioni sistem komunicira.



# Spojevi (junctions)

- Spojevi pomažu u organizovanju i pojašnjavanju dijagrama toka podataka.
- Postoje **divergentni** spojevi koji dele tok podataka na dva ili više izlaza, kao i **konvergentni** spojevi, koji spajaju dva ili više toka podataka u jedan.
- Spojevi sa jednim ulazom i jednim izlazom zovu se linearni spojevi.



# Specifikacija SSA

1. Hijerarhijski organizovan skup dijagrama toka podataka;
2. Rečnik podataka koji opisuje sadržaj i strukturu tokova i skladišta podataka.
3. Specifikacija logike primitivnih procesa

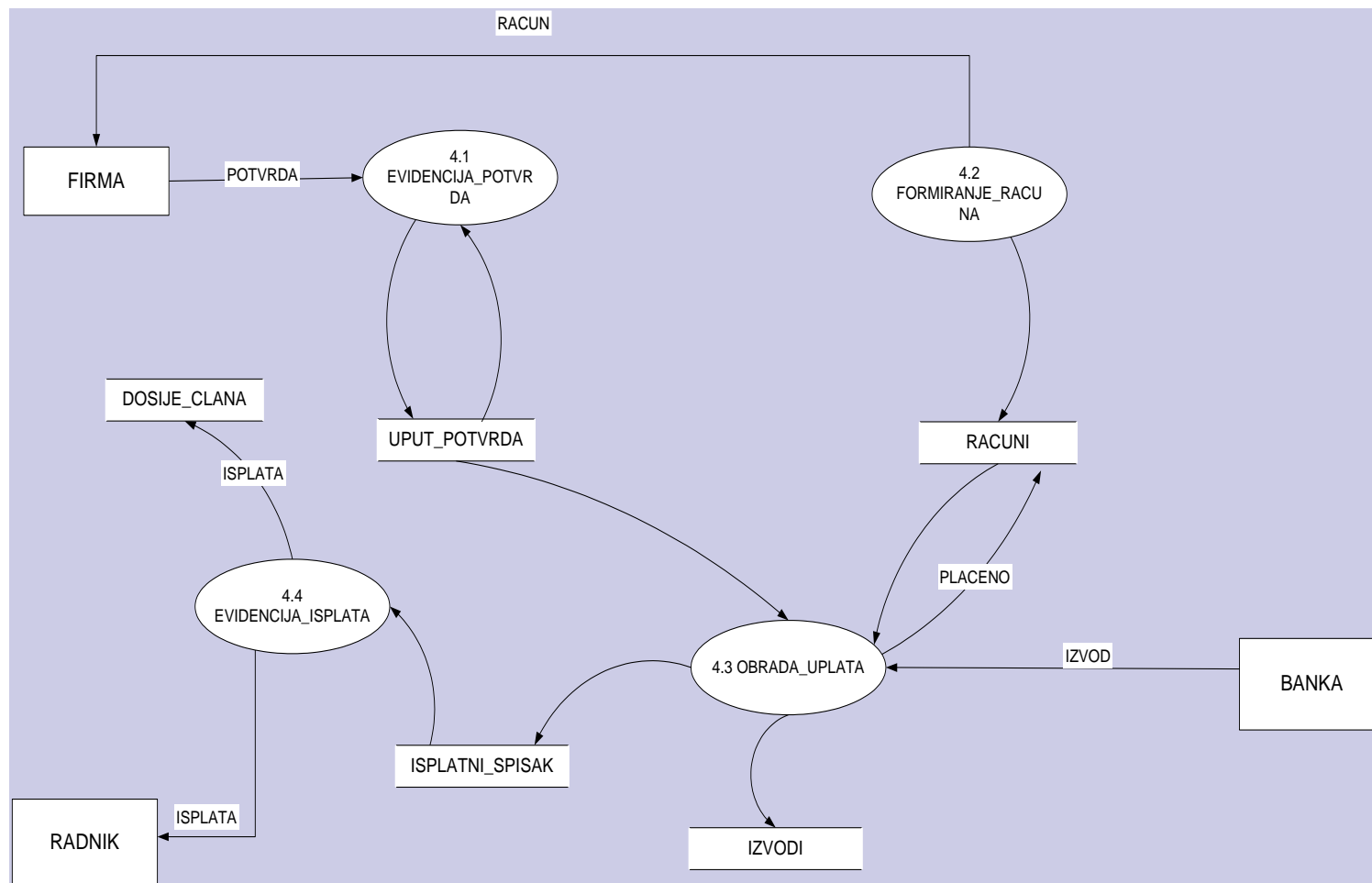
# SSA – odnos koncepata

- Međusobni odnos osnovnih koncepata se prikazuje preko

## **dijagrama toka podataka (DTP)**

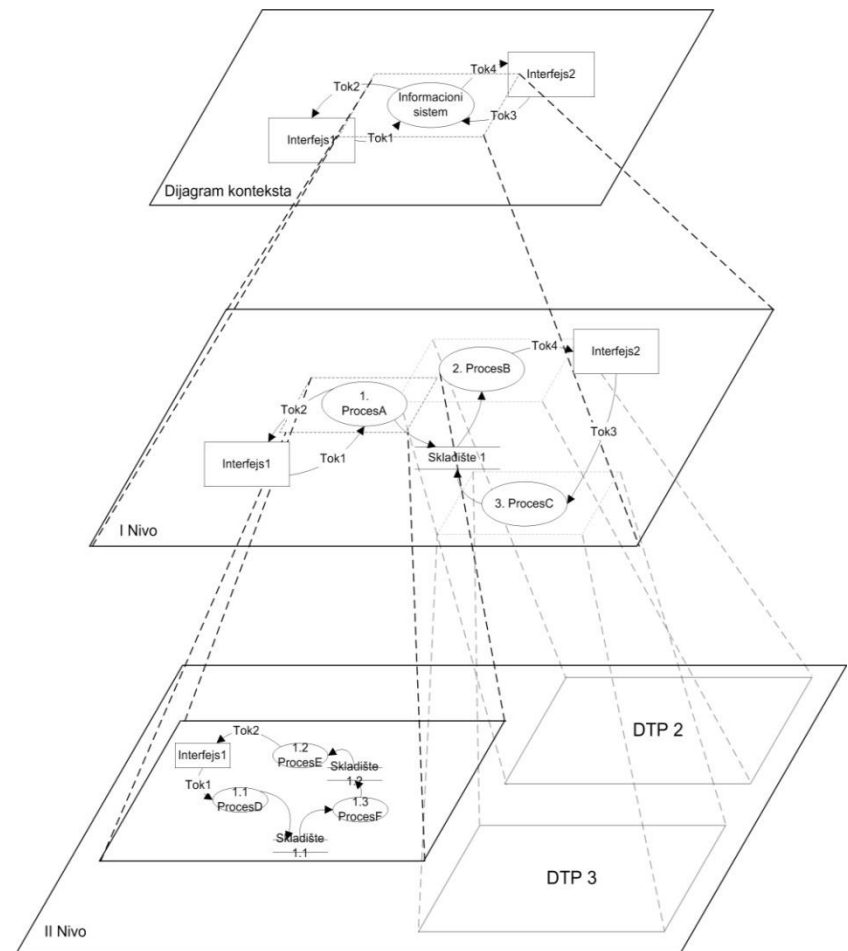
koji prikazuju vezu interfejsa ili skladišta kao izvora/ponora podataka, sa odgovarajućim procesima, kao i međusobnu vezu procesa.

# Dijagram toka podataka - DTP (data flow diagram - DFD)



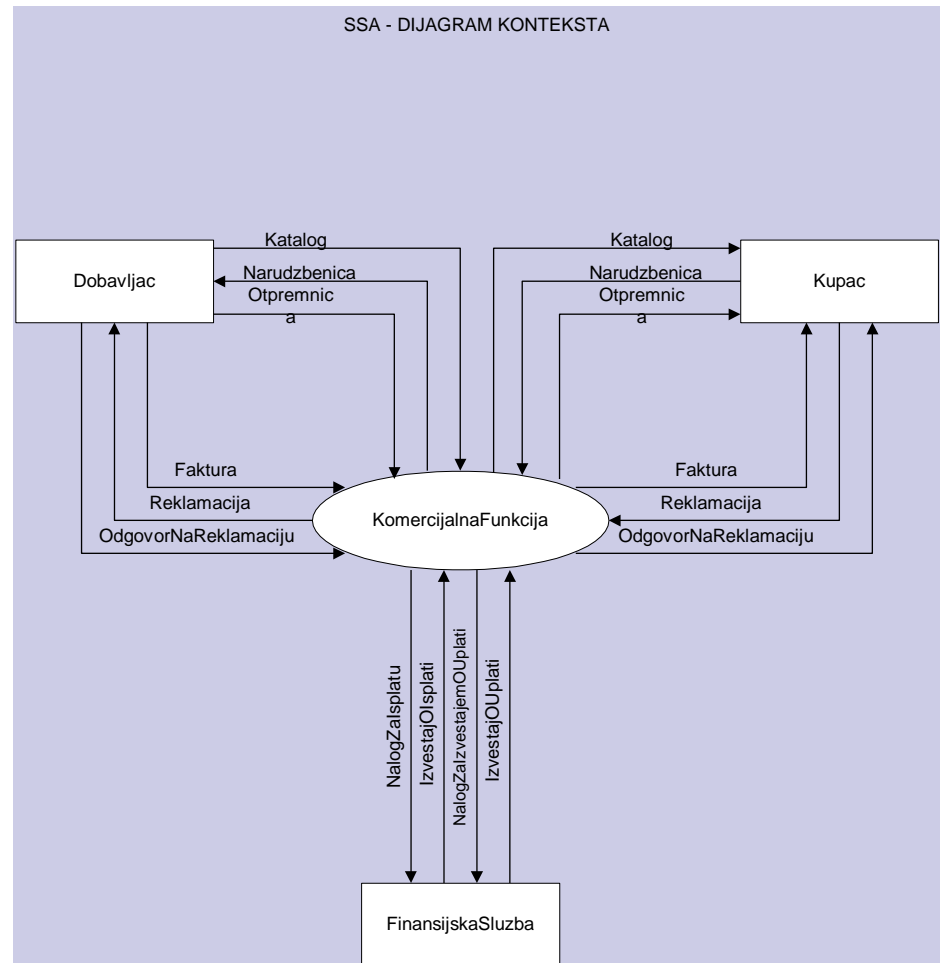
# Tehnika dekompozicije DTP-a

- IS može biti veoma složen i sadržati veliki broj procesa, tokova podataka, skladišta podataka i spoljnih objekata.
- Jasna i detaljna specifikacija sistema zahteva da se pri predstavljanju sistema pomoću dijagrama toka podataka primeni metoda dekompozicije.
- Dekompozicija se realizuje kroz primenu koncepta apstrakcije.



# Dijagram konteksta

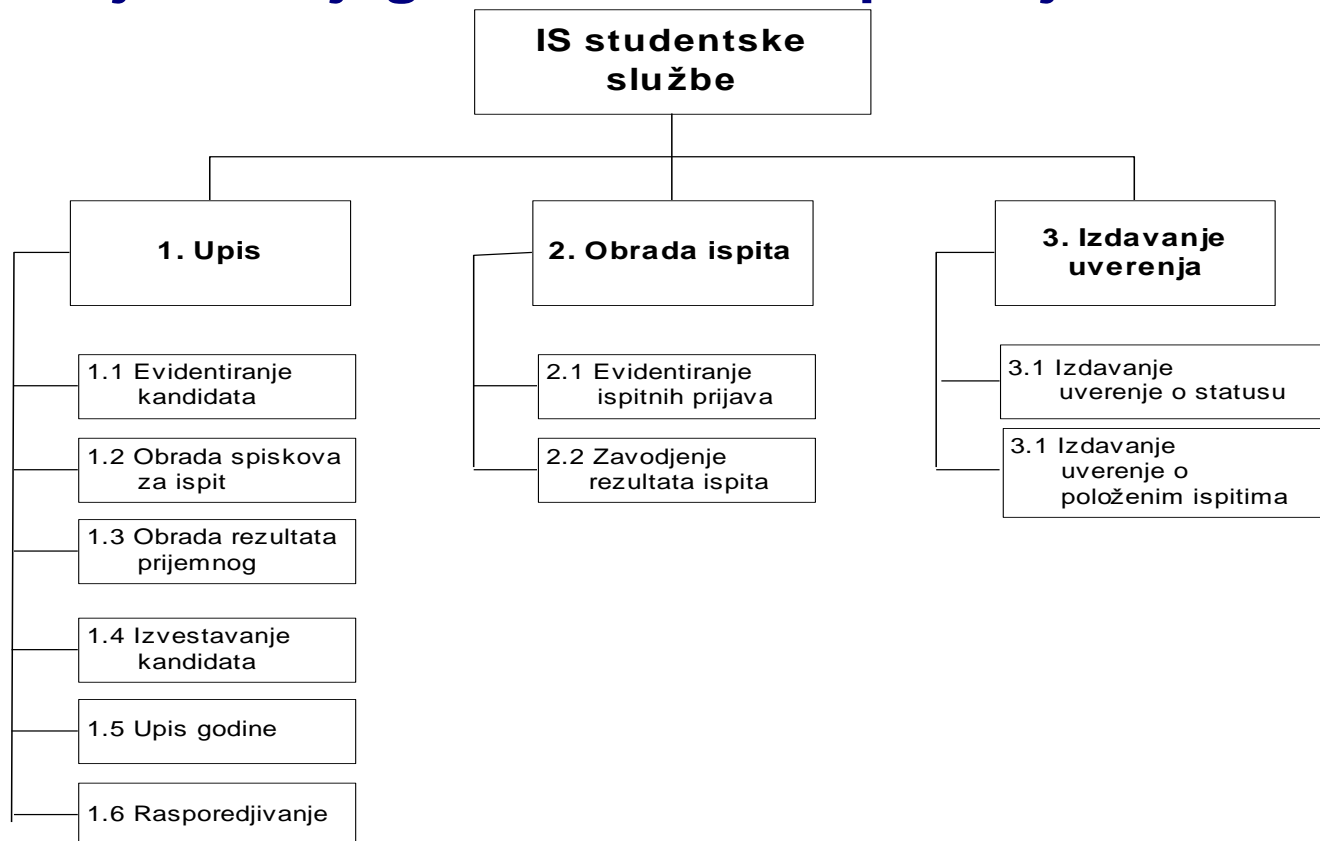
- Dijagram na najvišem nivou hijerarhije
- Sadrži proces visokog apstraktnog nivoa i njegovu komunikaciju (ulazne i izlazne tokove) sa okruženjem,
- Predstavlja granice sistema koji se analizira.





# Dijagram dekompozicije

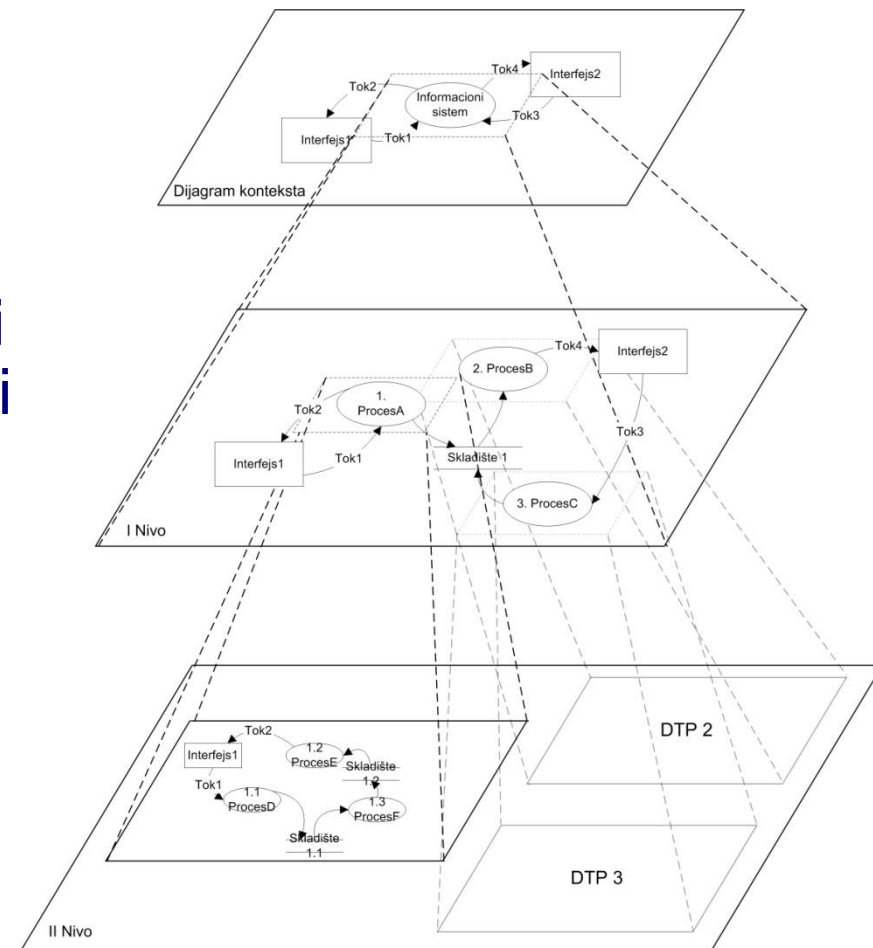
- Celokupan skup hijerarhijski dekomponovanih dijagrama predstavlja se dijagramom dekompozicije.



# Dekompozicija dijagrama konteksta

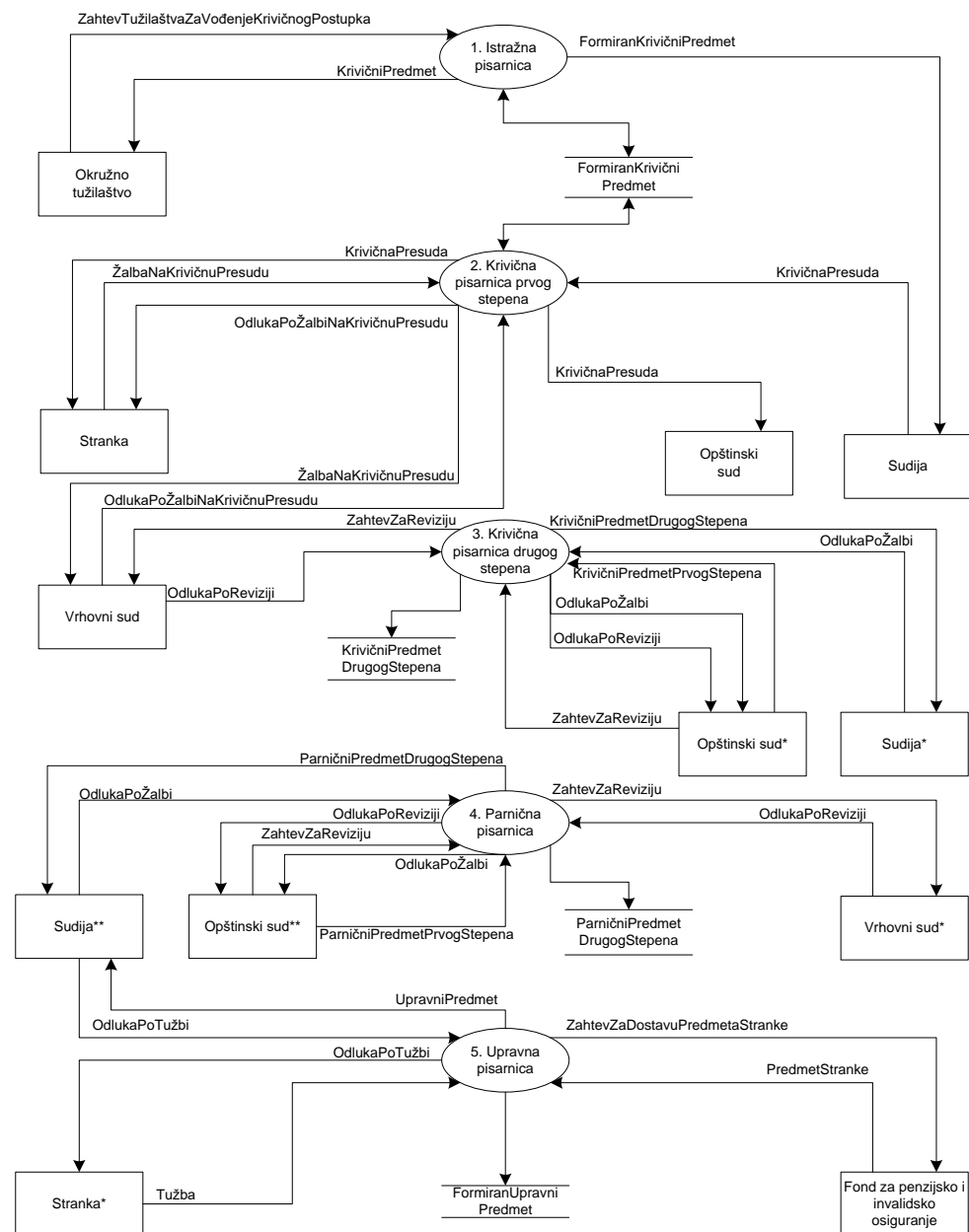
- Dijagram konteksta se dekomponuje na dijagrame u više nivoa. Broj takvih nivoa zavisi od složenosti sistema koji se analizira kao i od postupnosti u procesu dekompozicije.
- Preporučuje se da takvi nivoi sadrže najviše 2-7 procesa.

*(Ako procesa ima više, znači da je preskočen jedan apstakcioni nivo.)*



# Dijagram prvog nivoa dekompozicije

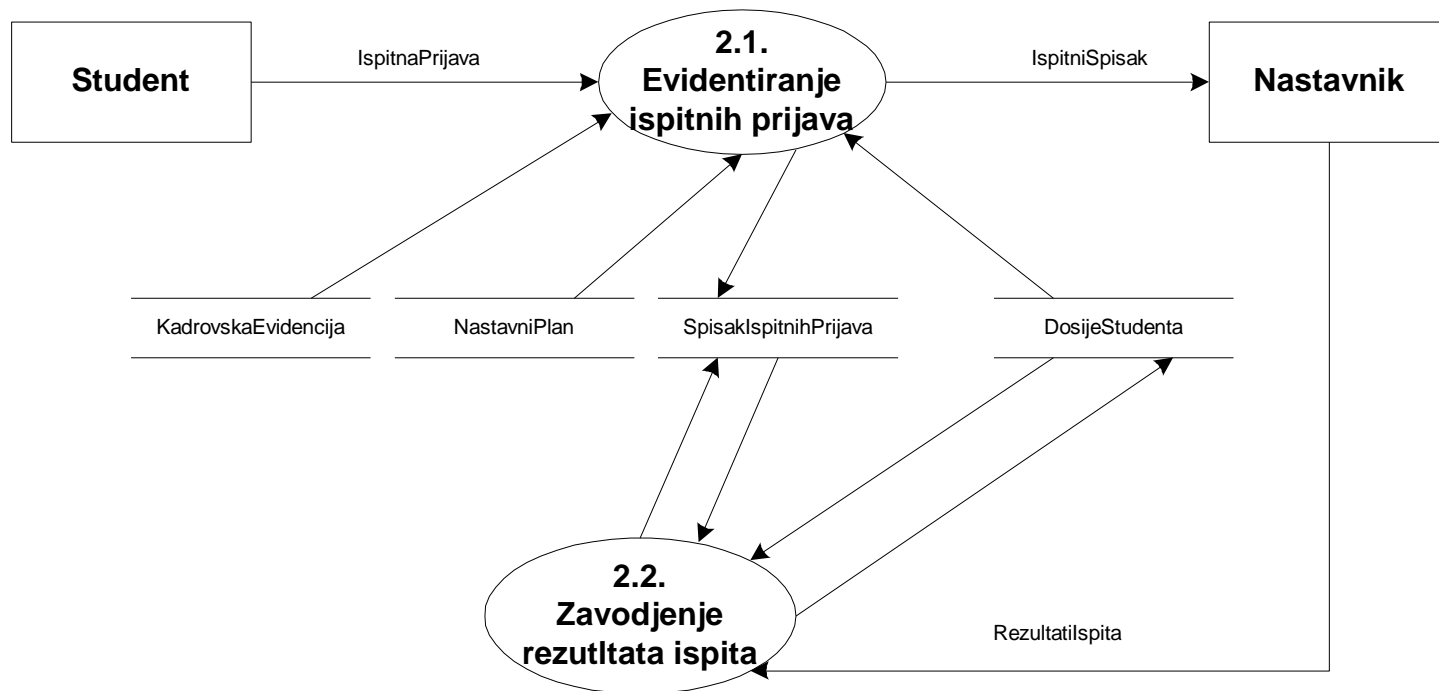
- Dijagram prvog nivoa predstavlja dekompoziciju dijagrama konteksta.
- Procesi na njemu označavaju se brojevima 1,2,3,...
- Dijagrami nižih nivoa su označeni oznakom procesa čije detalje predstavljaju, a procesi na njima povlače sa sobom brojnu oznaku nadređenog procesa, odnosno posmatranog dijagrama.



# Primitivni procesi

- Primitivni proces (funkcija) je onaj koji se dalje ne dekomponuje.
- Svaki primitivni proces definisan je:
  - ulaznim tokom podataka,
  - logikom obrade podataka iz ulaznog toka ili nekog skladišta podataka
  - izlaznim tokom podataka i/ili ažuriranim podacima u skladištu podataka.

# DTP – primitivni proces





# SSA - sintaksna i metodološka pravila

# Pravila i preporuke za kreiranje Dijagrama tokova podataka (1)

- Svaki *proces* mora da ima bar jedan ulazni i bar jedan izlazni tok podataka. Proces bez ulaznog toka generisao bi izlaz ni iz čega, a proces bez izlaznog toka je nesvrshodan.
  - Proces može da generiše izlaz samo iz svih za obradu potrebnih ulaznih tokova podataka, i obratno:
  - Proces može da generiše izlaz samo na osnovu za obradu raspoloživih tokova podataka.
- Jedan *proces* povezuje se sa drugim procesom samo posredno, preko skladišta podataka. Takvo pravilo se prvenstveno odnosi na niže nivoe dekompozicije. Nepostojanjem skladišta podataka između dva procesa sistem se lišava mogućnosti da zapamti sopstveno međustanje (nakon završetka jedne i početka druge obrade), a to je jedna od osnovnih osobina skladišta podataka.

Proces koji ima više ulaznih nego izlaznih tokova je **sintetički**, onaj koji ima više izlaznih nego ulaznih – **analitički**;

# Pravila i preporuke za kreiranje Dijagrama tokova podataka (2)

- Samo *tokovi podataka* koji idu ka, odnosno od skladišta podataka ne moraju biti imenovani.
  - Ako tok između procesa i skladišta nije imenovan, podrazumeva se da tok nosi celokupan sadržaj i strukturu podataka tog skladišta. Ukoliko to nije slučaj, tj. ako tok sadrži samo deo strukture podataka, treba ga adekvatno imenovati.
- *Tokovi podataka* koji poniru u jedno skladište ili iz njega izvire, mogu da prenose samo one pakete podataka koji se u skladištu mogu čuvati.

**PRAVILO BALANSA TOKOVA:** Ulazni i izlazni tokovi procesa P na nekom nivou dekompozicije, moraju se javiti i na njegovom dekomponovanom DTP-u.



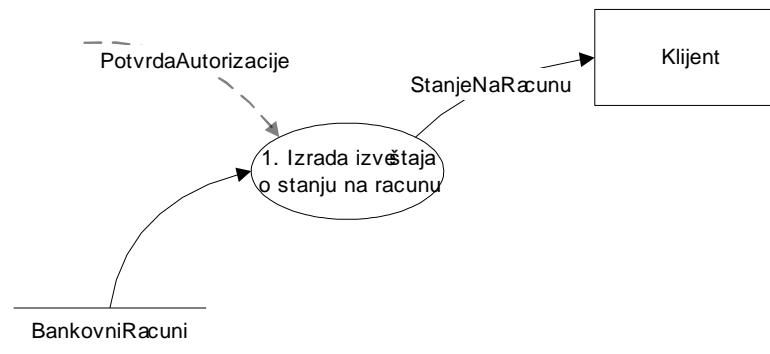
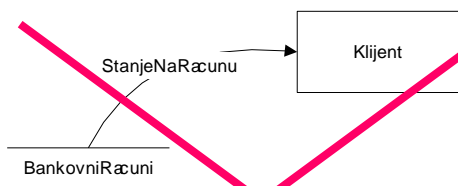
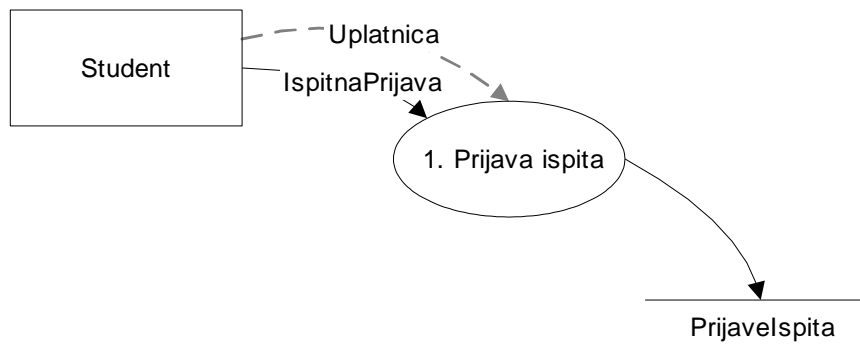
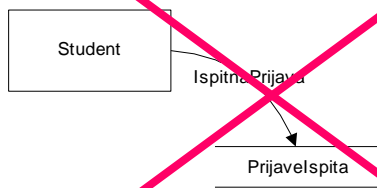
# Pravila i preporuke za kreiranje Dijagrama tokova podataka (3)

- *Tok podataka* mora da ima izvor i ponor. Iz ovog pravila sledi da dva interfejsa, dva skladišta, ili interfejs i skladište ne mogu direktno biti povezani tokom podataka.
  - *Skladišta* ne mogu međusobno direktno biti povezana tokom podataka, jer skladišta, kao pasivne komponente, ne vrše nikakvu obradu.
  - *Interfejsi* ne mogu međusobno direktno biti povezani tokom podataka, jer bi se na taj način opisivala komunikacija dva objekta van sistema koja, iako je moguća, nije od interesa za sistem koji posmatramo.
  - *Interfejs i skladište* ne mogu direktno biti povezani tokom podataka, jer bi to značilo da spoljni objekat direktno, bez kontrole internog procesa, poseduje pristup skladištu podataka. Ovakve situacije se razrešavaju uvođenjem odgovarajućeg procesa između interfejsa i skladišta.

# Pravila i preporuke za kreiranje Dijagrama tokova podataka (4)

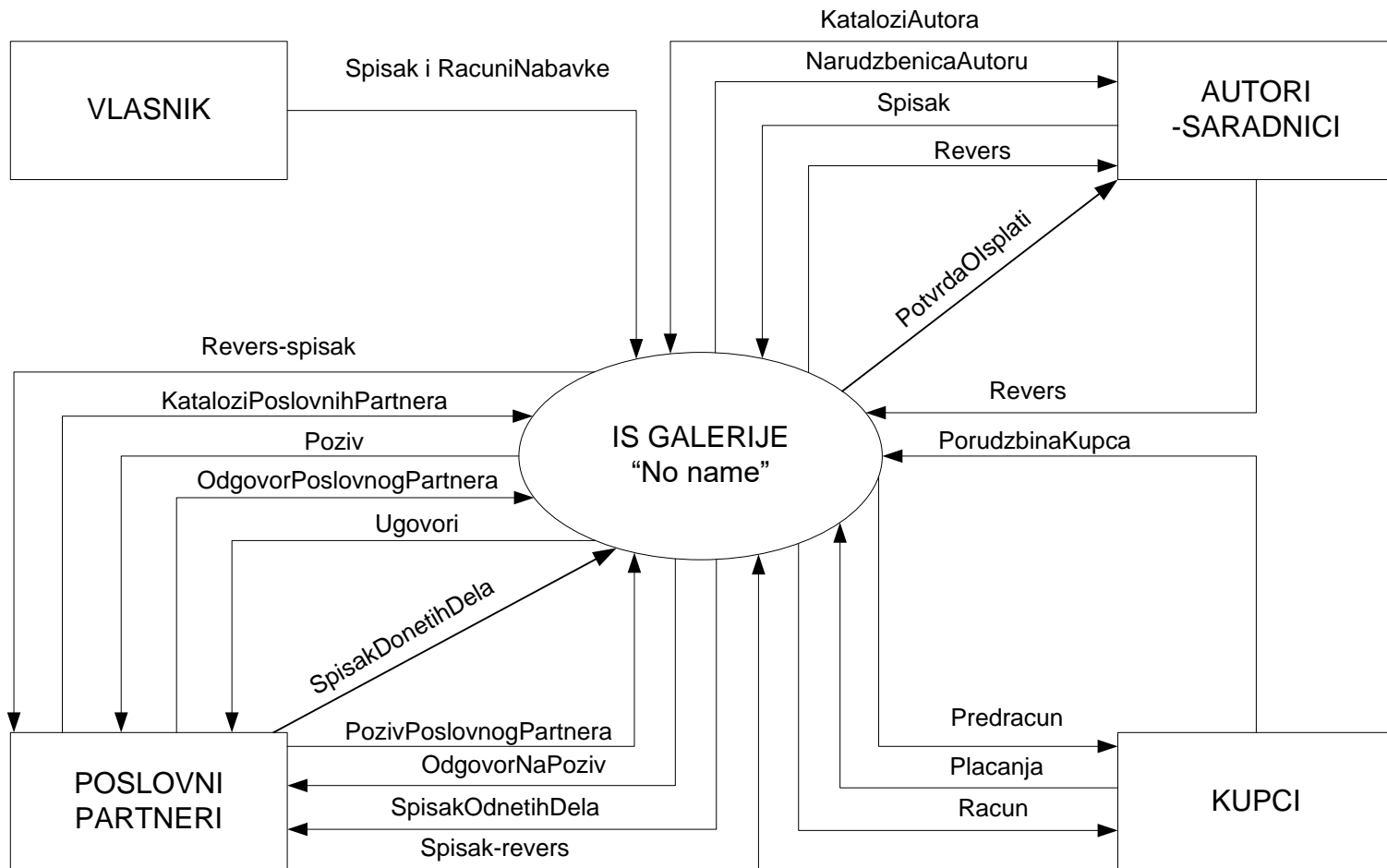
- Svako *skladište* mora da ima bar jedan ulazni ili bar jedan izlazni tok podataka. Često se za skladište najpre ustanovi samo jedan od tokova podataka, ali se kasnijom analizom drugog dela sistema utvrdi i ostatak tokova.
- *Interfejsi* moraju biti povezani sa sistemom, odnosno procesima sistema bar sa jednim ulaznim ili izlaznim tokom podataka.
- Preporuka vezana za preglednost dijagrama: u cilju izbegavanja nepotrebnog presecanja linija, *bilo skladište bilo interfejs na jednoj slici može se višestruko ponoviti*. U tom slučaju potrebno je samo pored imena koncepta dodati znak „\*“.
- **CRUD matrica** definiše način komunikacije sa skladištem.

# Primeri



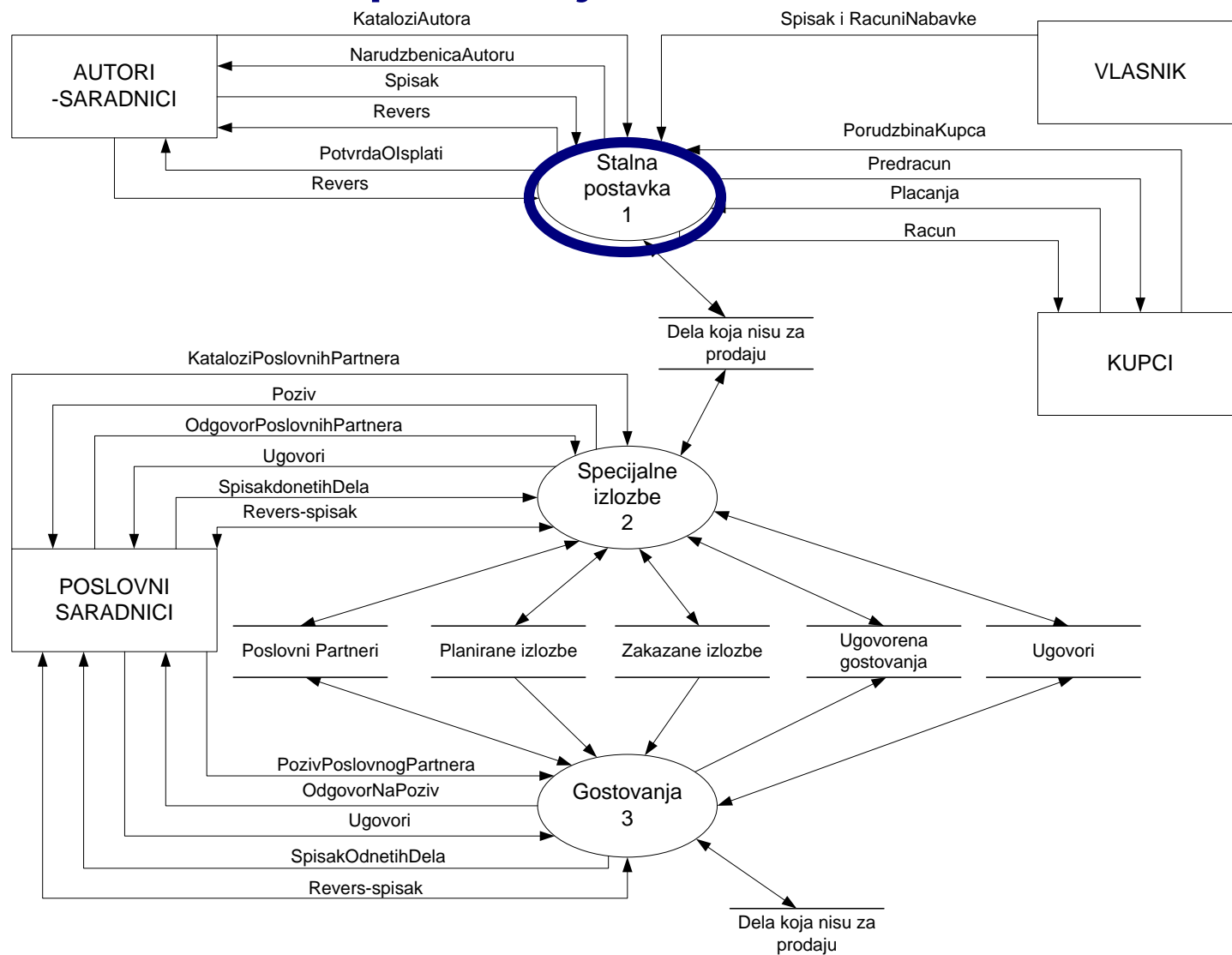
# Primer – SSA galerije

## DIJAGRAM KONTEKSTA



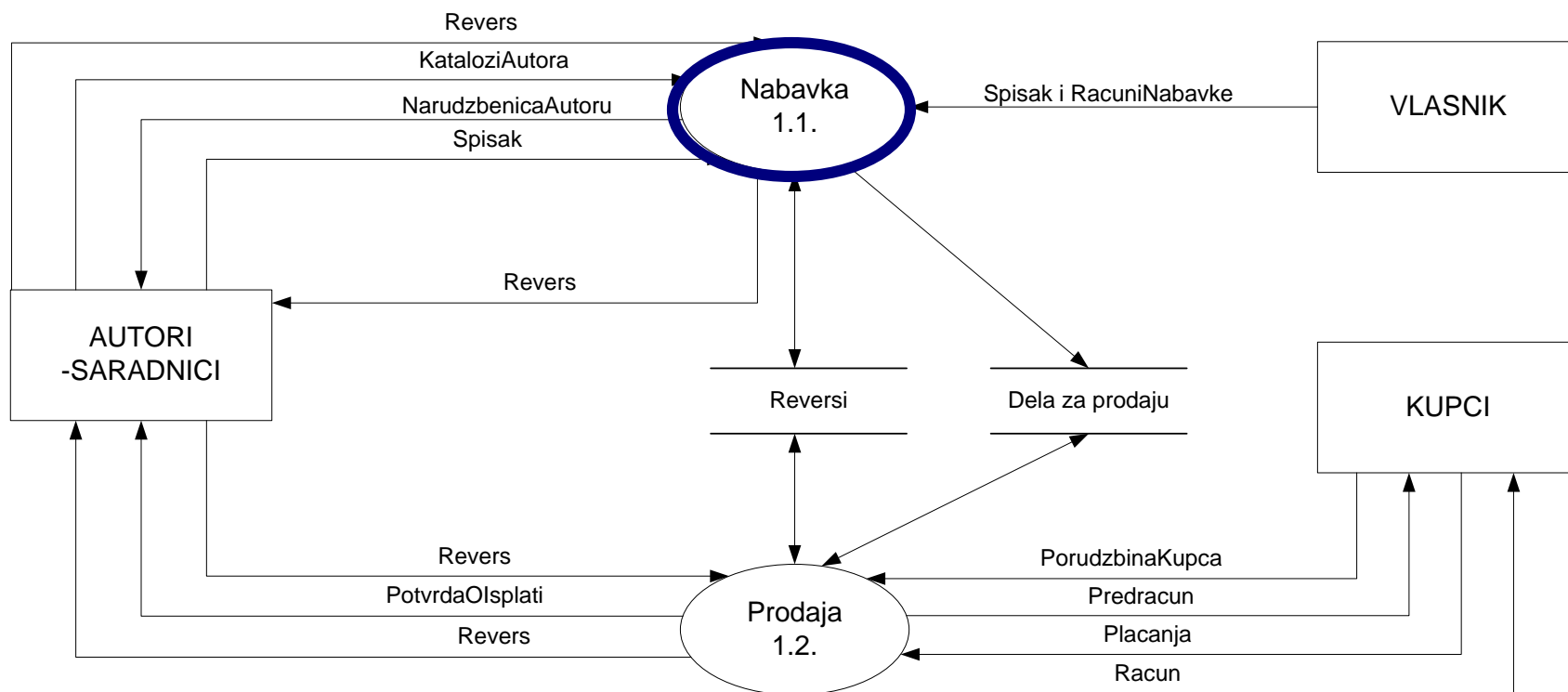
# Primer – SSA galerije

## I nivo dekompozicije



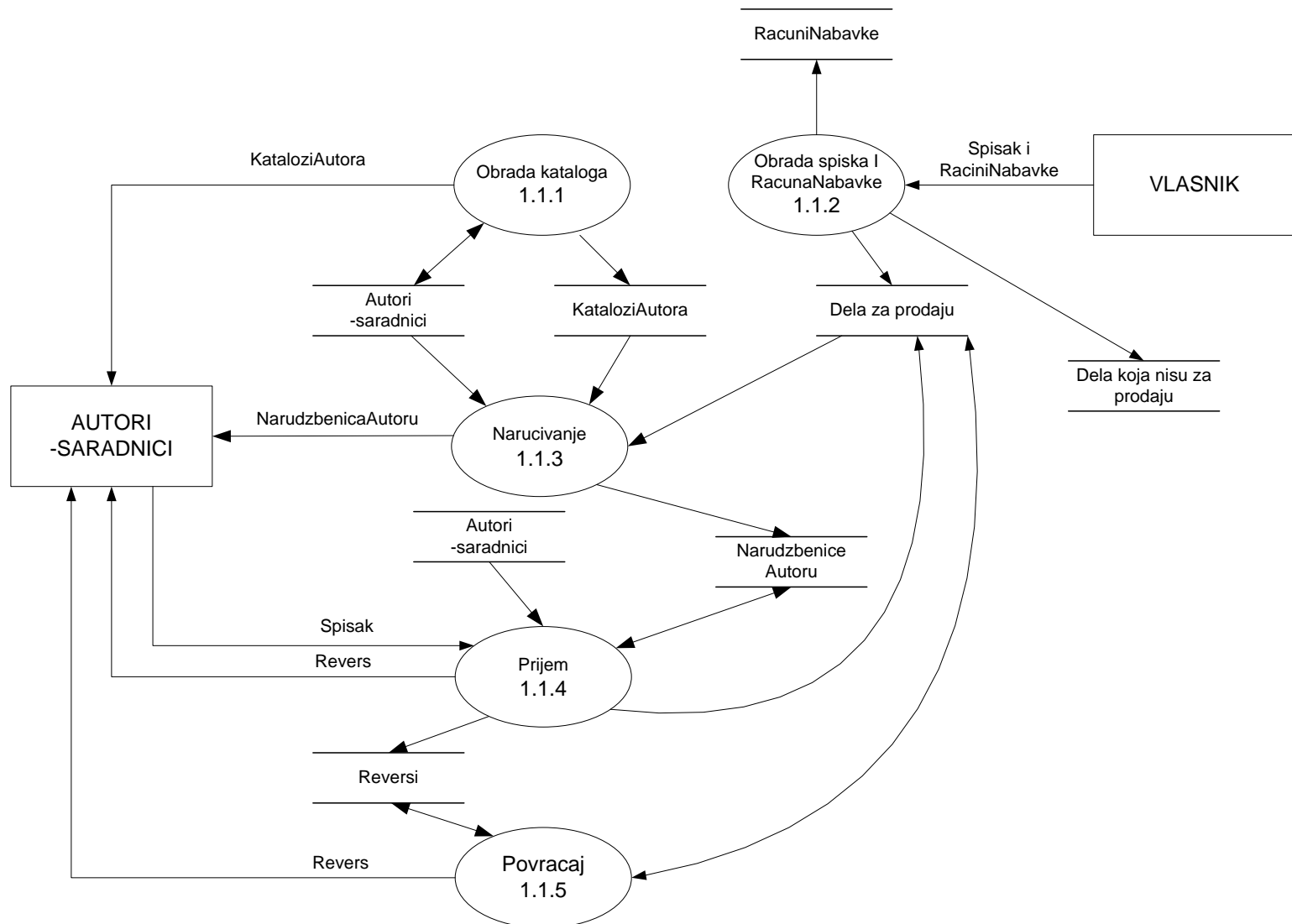
# Primer – SSA galerije

## 1. Stalna postavka - II nivo dekompozicije



# Primer – SSA galerije

## 1.1 Nabavka - III nivo dekompozicije, primitivna f-ja





Sledeća tema:  
Model podataka