Upravljanje informacijama i znanjem

Lekcija 3: Analiza IS - SSA

zima 2019/2020

Prof. dr Branimir M. Trenkić

Da ponovimo…

- ***Faze*** ***životnog cuklusa razvoja*** IS:
  1. Planiranje razvoja
  2. Analiza i specifikacija zahteva
  3. Projektovanje
  4. Implementacija (kodovanje, testiranje)
  5. Održavanje (vrednovanje i kontrola)
- Pojedinačnu fazu karakteriše:
  – Ciljevi (***rezultati***) te faze i
  – ***Metodologije*** koje možemo koristiti u toj fazi razvoja IS-a

Da ponovimo…

Faze su sledeće:

1. Planiranje razvoja
2. Analiza i specifikacija zahteva
3. Projektovanje
4. Implementacija
5. Održavanje

1. Planiranje

Rezultat je

- Opšta arhitektura IS
- Definisani su podsistemi
- Definisane su veze izmedju podsistema
- Definisani su prioriteti
- Dobijena je saglasnost za rad

- Metoda BSP (Business System Planning)

Strateško planiranje

- ***Prva faza*** u razvoju informacionog sistema je ***strateško planiranje***
- Sistem se posmatra ***sa najvišeg nivoa hijerarhije***
- ***Cilj (rezultati)*** **- Definišu se osnovni podsistemi**
  - Veze (interakcija) između podsistema,
  - **Prioriteti** prilikom projektovanja i uvođenja
- Nakon toga se pristupa projektovanju svakog od podsistema (***automatizacijom*** – ***CASE- alati***)

Strateško planiranje

- ***Tradicionalni pristup*** ***projektovanja*** i ***uvođenja*** sistema je "***od dna ka vrhu***"
- Odmah se pristupalo projektovanju i uvođenju podsistema ***bez prethodnog sagledavanja celine*** (***opšte arhitekture*** informacionog sistema)
- Vremenom je uočeno:
  - Automatizacija delova bez sagledavanja celine ima ***brojne nedostatke***
  - Nije bilo dobro sistematski odrađeno ***sagledavanje interakcije podsistema*** unutar sistema baze podataka

Strateško planiranje

- U cilju prevazilaženja ovih problema razvijeno je ***strateško planiranje***
- Da bi se izbegli ***brojni problemi*** koji se u ovakvim implementacijama javljaju, IBM je razvio ***metodu BSP*** kojom se
- ***Planiranje i analiza*** obavljaju ***od vrha ka dnu***
- ***Projektovanje i uvođenje*** obavljaju ***od dna ka vrhu***

BSP metoda

- ***Šta se postiže*** BSP metodologijom?
- Definiše se ***opšta arhitektura informacionog sistema na osnovu:***

1. ***Poslovnih procesa*** kao relativno ***najstabilnije komponente realnog sistema*** (u odnosu na organizacionu strukturu, način upravljanja i odlučivanja koji su često promenljivi)
2. ***Modela podataka***
   - kao ***osnova informacionog sistema***, koji tretira podatke kao posebne resurse u sistemu

BSP metoda

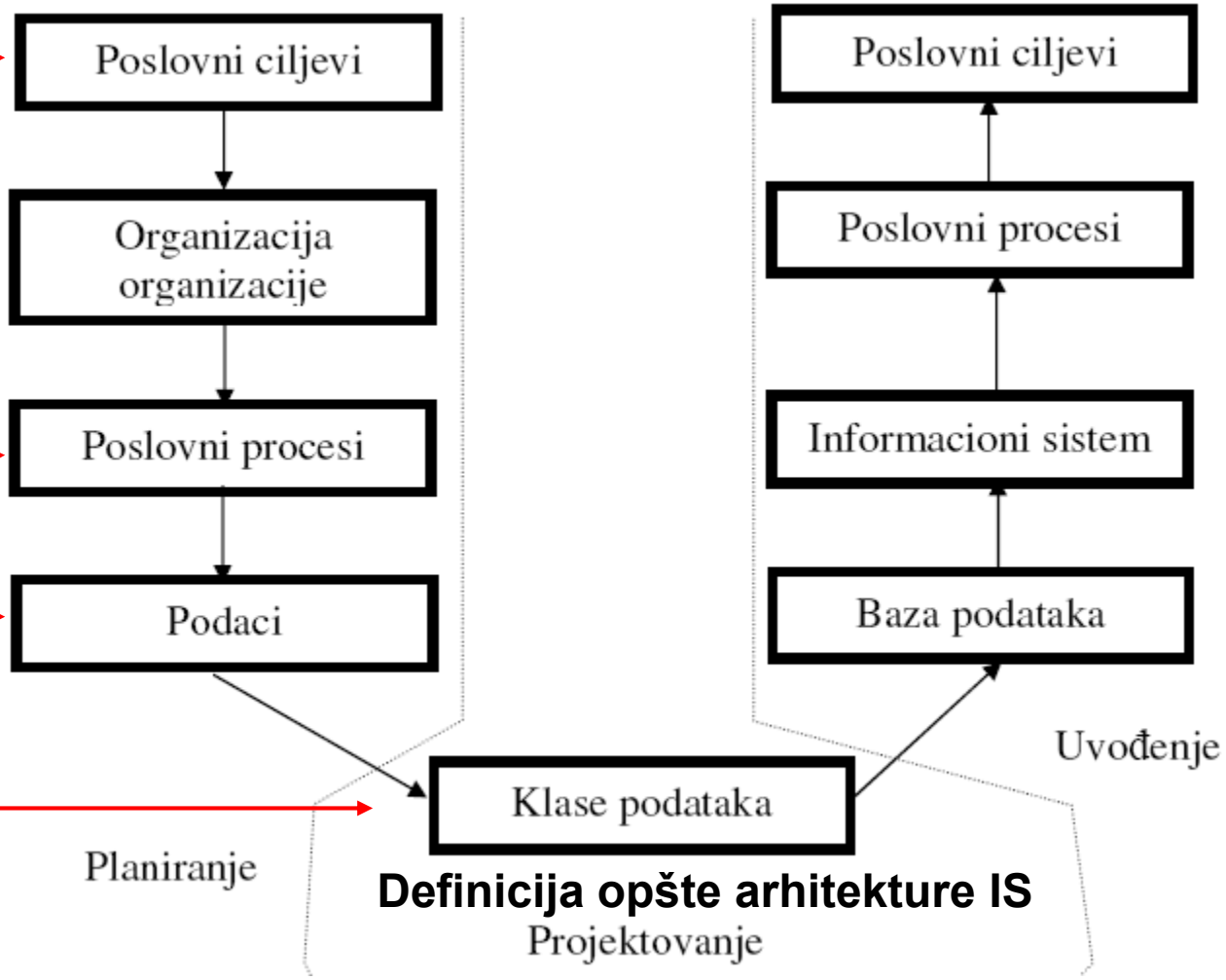
- Po ovoj metodoligiji ***IS se planira*** ***“od vrha ka dnu”***
  - osnovni podsistemi,
  - veze,
  - prioriteti,
  - projektovanje podsistema
- ***IS se*** zatim ***uvodi*** (implementira) ***od prostijih funkcija ka složenijim***
- ***Polazi se*** ***od poslovnih ciljeva organizacije*** koje treba da ***podrži*** IS ili ***unapredi***

BSP metoda

koje treba da podrži IS ili unapredi

izdvajanje poslovnih procesa
i
podataka koje ti procesi koriste

modelovanje podataka (ERwin)



Planiranje i uvođenje IS primenom BSP metode

Da ponovimo…

Faze su sledeće:

1. Planiranje razvoja
2. Analiza i specifikacija zahteva
3. Projektovanje
4. Implementacija
5. Održavanje

2. Analiza i specifikacija zahteva

- Specifikacija zahteva (šta korisnik hoće?)
- Model procesa budućeg stanja sistema
- Struktura podataka
- Logika procesa

- Metod: SSA (Strukturna Sistem Analiza)

SSA - Uvod

- ***Strukturna sistemska analiza*** (***SSA***) - ***potpuna metodologija*** za ***specifikaciju informacionog sistema***
- ***Ona se*** na različite načine ***može povezati sa metodama drugih faza*** u neku specifičnu ***metodologiju celokupnog razvoja IS***
- Metoda SSA se ovde tretira kao ***jedinstvena, samosvojna metoda***
- Kasnije se pokazuje kako se ona koristi u pojedinim koracima standardne metodologije razvoja informacionih sistema

SSA - Uvod

- Bitan ***preduslov za uspešno (I) projektovanje*** i ***(II) implementaciju*** sistema
- ***(I) potpuna***, ***(II) tačna***, ***(III) formalna*** i ***(IV) jasna specifikacija IS*** (specifikacija zahteva korisnika)
  - Zahteva koje budući sistem treba da zadovolji
- ***Očigledno*** je zbog čega specifikacija IS treba da bude ***potpuna i tačna***
- Da specifikacija bude ***formalna*** - formalna specifikacija ***osnov za "evolutivno" projektovanje*** i ***implementaciju***, odnosno za ***korišćenje CASE alata***

SSA - Uvod

- Da specifikacija bude ***jasna*** - ***jezik specifikacije***
- U specifikaciji IS u velikoj meri ***učestvuju*** korisnici sitema – ***neinformatičari***, zato
- ***Jezik specifikacije*** mora biti i ***njima prihvatljiv***
- ***Ideja SSA?***
- Da poseduje veoma jednostavne, ***grafičke***, pa samim tim i ***jasne koncepte***
- ***Strožija formalizacija*** samo ***za opis strukture (I)*** tokova i ***(II)*** skladišta podataka
  - Da bi se obezbedio specifičan evolutivni razvoj IS koji standardna metodologija zagovara

SSA - Uvod

- ***Specifikacija IS*** treba da prikaže (potpuno, tačno, formalno i jasno) ***šta*** budući informacioni sistem ***treba da radi***
- Veoma je bitno istaći da specifikacija IS prikazuje ***ŠTA IS treba da dâ***, a ***ne i KAKO*** to treba da ostvari
- ***Prerano definisanje "kako“*** (uvođenje nekih projektantskih rešenja u okviru specifikacije) - ***ograničava kasniji*** mogući ***izbor*** (optimizaciju) ***načina implementacije sistema***

SSA - Uvod

- Odgovor na pitanje "***kako***" daje se za
  - ***konkretno okruženje***,
  - ***organizaciju*** u kojoj se sistem implementira
  - definisanu ***tehnologiju*** i
- Specifikacija ***ne treba da sadrži*** tehnološki i organizaciono ograničena rešenja
  - Ona treba da opiše funkcionisanje IS u **"idealnoj tehnologiji“** - nikakva ograničenja ne postoje
- Ako je specifikacija ovako zadata - ***pre prelaska na*** dalje ***projektovanje*** - ***definišu se sva ograničenja*** koja nameće okolina u kojoj se sistem implementira

SSA - Uvod

- Zbog toga ***specifikacija IS*** treba da poseduje sledeća ***dva dela***:
  - ***funkcionalnu specifikaciju*** u kojoj se opisuje ***budući IS u "idealnoj tehnologiji"*** i
  - ***nefunkcionalnu specifikaciju*** koja ***definiše sva ograničenja*** implementacione okoline
- ***SSA*** u **potunosti** obuhvata ***samo funkcionalnu specifikaciju***
- Nefunkcionalne - ***samo delimično*** pokriva
  - prikazujući ***tokove podataka*** u novo-implementiranom sistemu

SSA - Uvod

- ***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 ***odvode*** ***preko tokova podataka***
- ***Tok podataka*** – kao ***vod*** ili kao pokretna traka ***kroz koji stalno teku podaci***
- Imajući u vidu zahtev da specifikacija treba da se oslobodi svih implementacionih detalja ***od interesa*** su ***samo sadržaj i struktura ulaznog toka***, a ne i medijum nosilac toka

SSA - Uvod

- ***Izvori ulaznih*** tokova podataka?
- ***Ponori izlaznih*** tokova podataka?

mogu biti

1. ***Objekti van IS*** koji sa IS komuniciraju
    - U SSA oni se nazivaju ***interfejsi***
2. ***Drugi procesi*** u sistemu ili
3. ***tzv***. ***Skladišta***
    - Skladišta podataka se posmatraju kao "tokovi u mirovanju", odnosno ***odloženi, akumulirani tokovi***,
    - Za skladišta kao i za tokove od interesa su isključivo njihov ***sadržaj i struktura***

SSA - Uvod

- ***Osnovni koncepti za specifikaciju*** IS u SSA su:

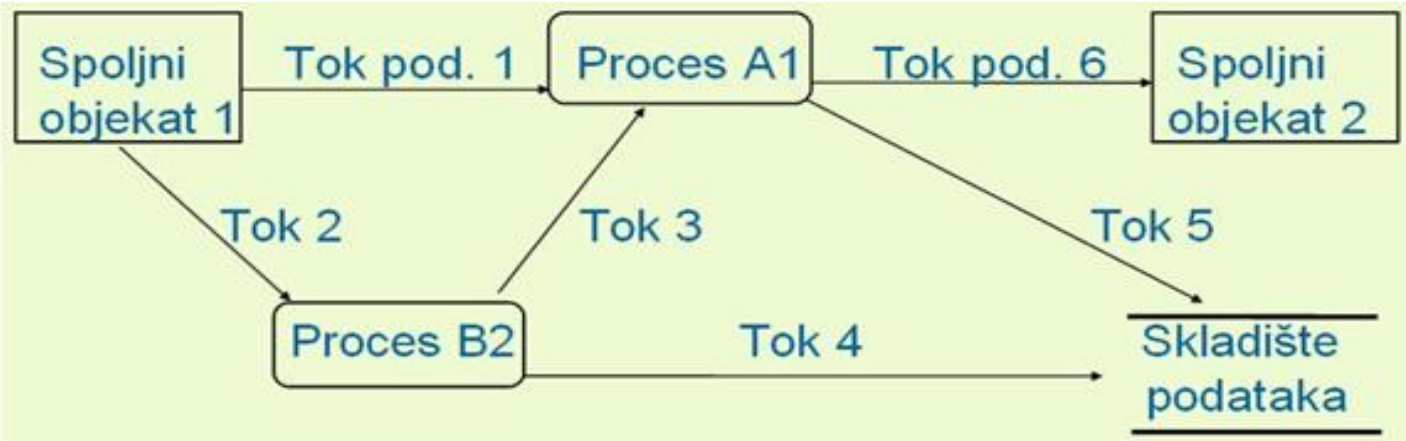
1. ***Funkcije***, odnosno ***procesi*** obrade podataka,
   - Koji na bazi ulaznih podataka generišu izlazne
2. ***Tokovi podataka***,
   - Preko njih se dovode ulazni podaci i odvode izlazni iz procesa
3. ***Skladišta podataka*** i
4. ***Interfejsi***

SSA - Uvod

- Njihov ***<u>međusobni odnos</u>*** se prikazuje preko ***<u>dijagrama toka podataka</u>*** (***<u>DTP</u>***) koji prikazuje:
  1. vezu interfejsa (skladišta) kao izvora (ponora) podataka sa odgovarajućim procesima
  2. međusobnu ***vezu procesa***

SSA - Uvod

- Slika prikazuje jedan opšti ***dijagram toka podataka*** (***DTP***) koji ima za cilj i da uvedemo sledeće grafičke simbole:



(I) krug ili elipsa pretstavlja funkciju ili proces obrade podataka,

(II) pravougaonik predstavlja interfejs,

(III) usmerena linija predstavlja tok podataka,

(IV) dve paralelne linije ("otvoreni" pravougaonik) predstavlja skladište podataka.

SSA - Uvod

- ***Jedan IS*** se sastoji iz ***mnoštva*** procesa, interfejsa, tokova i skladišta podataka
- ***Specifikacija IS*** treba da bude ***potpuna*** (detaljna) i ***jasna ?***
  a) Kada bi se ***jedan sistem*** detaljno opisao i prikazao ***jednim dijagramom toka podataka***, dobio bi se ***veoma nejasan opis*** sistema
  b) ***Paukova mreža*** procesa, tokova, skladišta i interfejsa

- ***Rešenje***: Detaljan i jasan opis sistema zahteva ***opis na "različitim nivoima apstrakcije"***

SSA - Uvod

- ***Hijerarhijski opis***:
- ***Na višim*** nivoima opis sistema je - ***opštiji***,
- ***Na nižim***, postepenim i organizovanim uvođenjem detalja, opis sistema je - potpuniji i ***detaljniji***
- ***Hijerarhijski opis*** sistema ***u tehnici dijagrama tokova podataka*** se svodi na to da se ***na višim nivoima definišu*** ***globalniji procesi***,
- Svaki ***globalni proces***, na sledećem nižem nivou- predstaviti ***novim dijagramom toka podataka***

SSA - Uvod

- ***Dijagram toka podataka na vrhu ovakve hijerarhije*** naziva se **dijagram konteksta**
- Procesi na ***najnižem nivou*** (procesi koji se dalje ne dekomponuju) nazivaju se **primitivni procesi**
- Imajući u vidu sve rečeno, jednu ***potpunu specifikaciju IS čine***:

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

Dijagrami toka podataka

- ***Dijagram toka podatka*** (***DTP***) predstavlja ***model sistema*** koji sadrži ***četiri osnovne komponente***:
  1. ***Procese obrade*** podataka (aktivne komponente sistema),
  2. ***Objekte okruženja*** (***interfejse***) sa kojima sistem komunicira,
  3. ***Skladišta podataka*** koje procesi koriste i/ili ažuriraju i
  4. ***Tokove podataka*** koji ***povezuju ostale komponente sistema*** u celinu

Dijagrami toka podataka

- ***Osnovne karakteristike*** DTP-a su:
  - ***jasna grafička specifikacija***, pogodna za komunikaciju sa korisnikom,
  - istovremeno ***jasan*** i ***detaljan opis sistema***, primenom ***metode apstrakcije***
  - sistem ***na višim nivoima*** apstrakcije se opisuje ***uopšteno***, a na ***nižim detaljno***

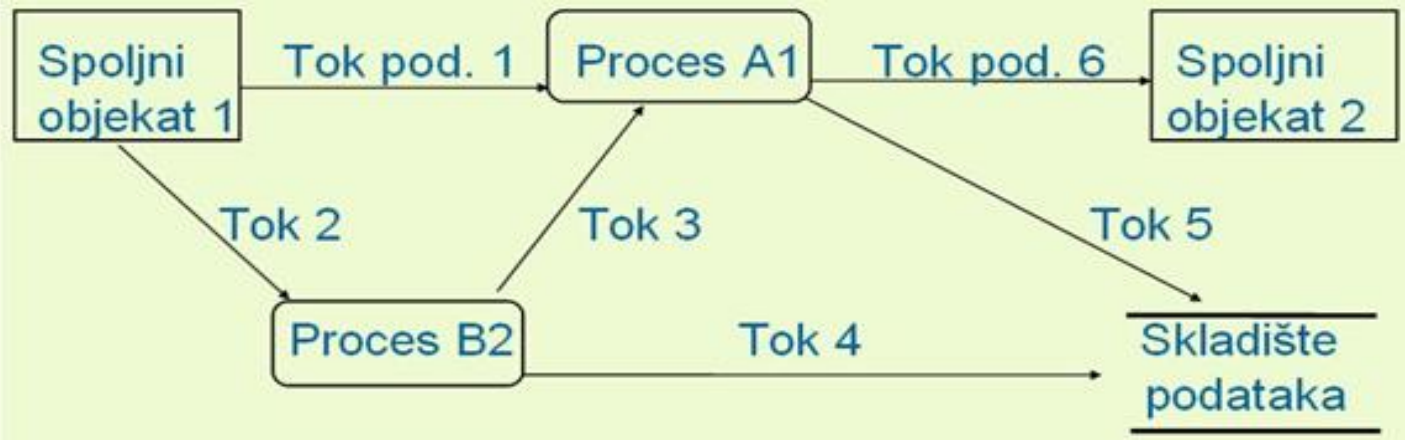
Dijagrami toka podataka

DTP - model sistema koji sadrži četiri osnovne komponente za prikaz IS:

- Funkcije - procesi obrade podataka
- Tokovi podataka
- Skladišta podataka
- Interfejsi – objekti okruzenja

Pravila kreiranja DTP-a

- ***(I)*** ***Tok podataka*** se može ***zamisliti*** kao ***pokretna traka*** u fabrici ili ***cev*** ***kojom*** od skladišta podataka (ili procesa ili...) ***teku struktuirani podaci*** (različita ***dokumenta***, ***formulari***, ***obrasci***,....)
- Ostvaruje ***vezu*** ***između*** ostalih ***komponenti*** sistema i na DTP-u se predstavlja ***(a)*** ***imenovanom*** i ***(b)*** ***orijentisanom*** ***linijom***

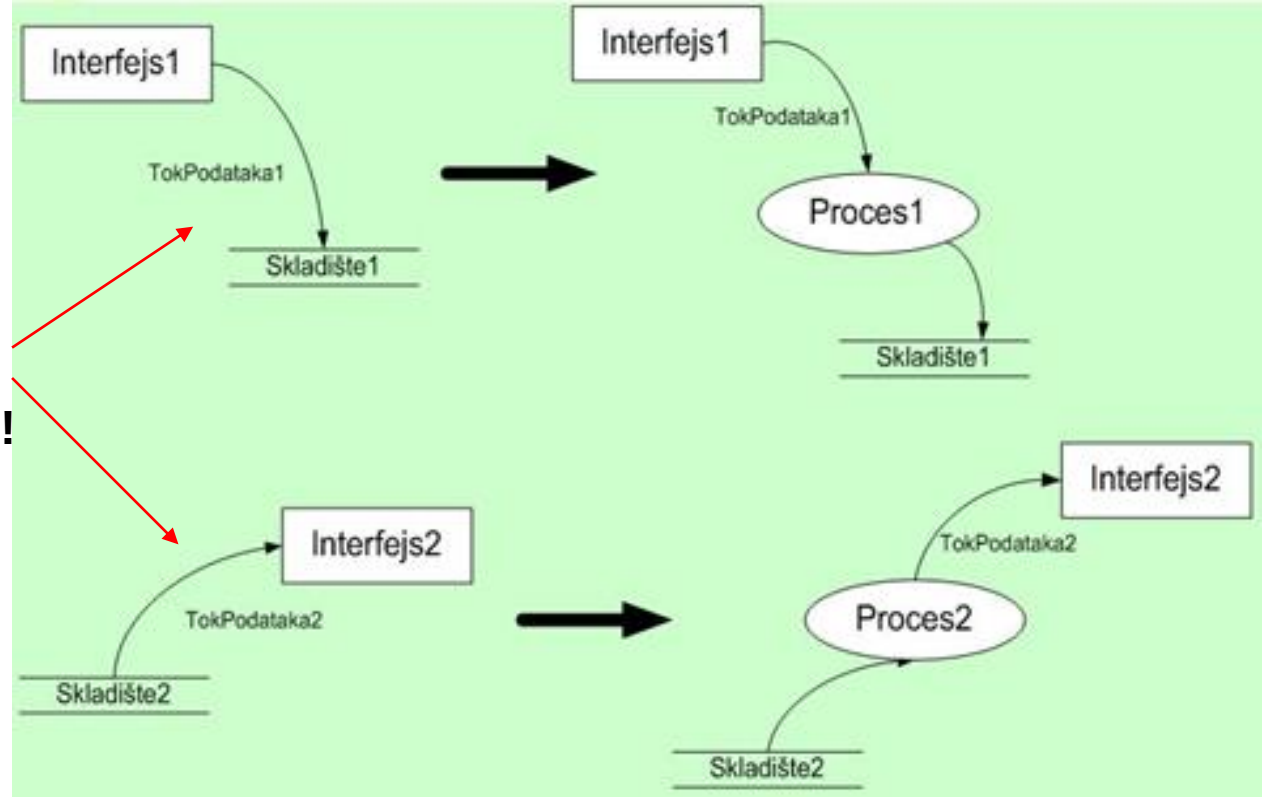


Pravila kreiranja DTP-a

- ***(II) Tok podataka*** mora da ***ima izvor i ponor***
  - ***Bilo koja*** druga ***komponeneta*** DTP ***može da bude izvor ili ponor***
  - ***Pravila*** izuzeća:
    - Za jedan tok, bilo izvor, bilo ponor (bilo oba) ***mora da bude proces***
    - Tokovima se **ne mogu** neposredno povezati ***dva skladišta*** (SS), ***dva interfejsa*** (II), ili ***skladište i interfejs*** (SI)

Pravila kreiranja DTP-a

Direktan pristup bazi podataka iz spoljnjeg sveta!!!



Primeri *nedozvoljenih situacija* kod DTP-a, II, SS, IS

Pravila kreiranja DTP-a

- ***(III)*** ***Svaki tok podatka*** na DTP-u ***mora imati ime***, koje treba da ***odražava*** ***značenje podataka*** koje on nosi
- Da bi se ostvarila čitljivost i razumljivost DTP-a - ***imena*** treba da budu ***prirodna***, a ne neka specifična, preterano skraćena imena
- ***Izuzetak*** su tokovi koji idu **ka (od)** ***skladišta podataka*** koji ***ne moraju biti imenovani***
  - Podrazumeva da tok nosi ***celokupan*** ***sadržaj*** i ***strukturu*** podataka tog skladišta
  - Ukoliko to nije slučaj, tj. ako tok sadrži ***samo deo strukture podataka***, treba ga imenovati

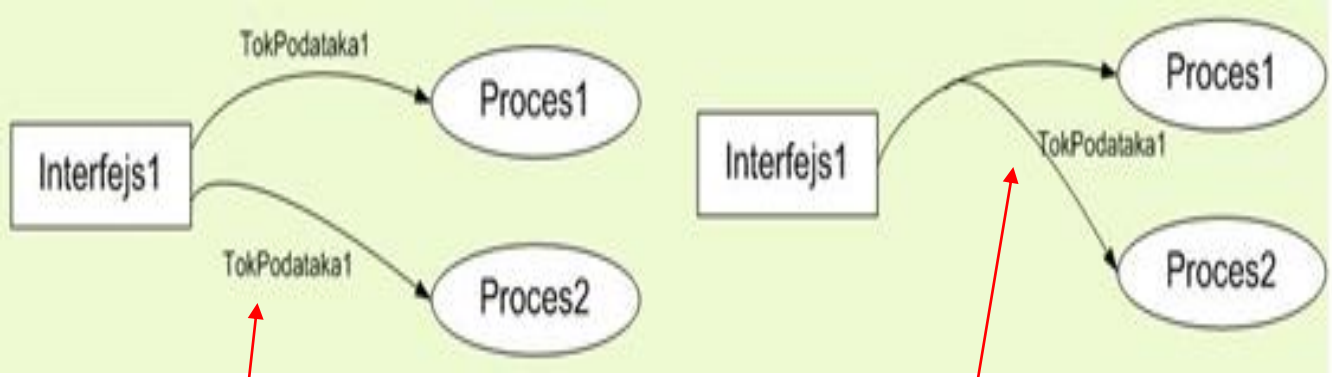
Pravila kreiranja DTP-a

- Pravila o imenovanju prikazane su na slici:

- ***Ako je tok*** između skladišta i procesa ***imenovan***, ***imenovana struktura*** mora biti ***deo strukture skladišta***, što treba da bude specificirano u ***Rečniku podataka***

Pravila kreiranja DTP-a

- ***(IV)*** ***<u>Tok podataka</u>*** se ***može granati***



eksplicitno prikazuje grananje jednoga toka

ista struktura, umesto preko grananja prikazana sa dva istoimena toka koja imaju isti izvor a različite ponore

- ***Istoimeni tokovi*** na DTP u suštini pretstavljaju ***grananje*** jednog toka, pa moraju imati ***<u>zajednički izvor</u>***, a mogu imati ***<u>različite ponore</u>***

Pravila kreiranja DTP-a

- ***(V)*** ***Proces*** obrade podataka je ***aktivna komponenta sistema*** koja vrši ***tranformaciju strukture*** i ***sadržaja*** ulaznog (ili ulaznih) tokova u izlazni (ili izlazne) tok podataka
- Svaki proces ima (***I***) ***naziv*** i (***II***) ***oznaku***
- ***Naziv procesa*** - precizno ***označava funkciju*** koju on obavlja
- ***Brojna oznaka*** - služi za ***referenciranje*** procesa
  - O načinu dodeljivanja brojne oznake govorićemo kasnije

Pravila kreiranja DTP-a

- ***(VI)*** ***Svaki proces*** mora da ***ima barem jedan ulazni i barem jedan izlazni tok podataka***

- ***(VII)*** ***Svako skladište***, takođe, treba da ***ima barem jedan ulazni i barem jedan izlazni tok***
  - Dozvoljava se da skladište ***nema ulazni tok***, podrazumevajući da se ***formira*** i ažurira ***u nekom drugom sistemu***

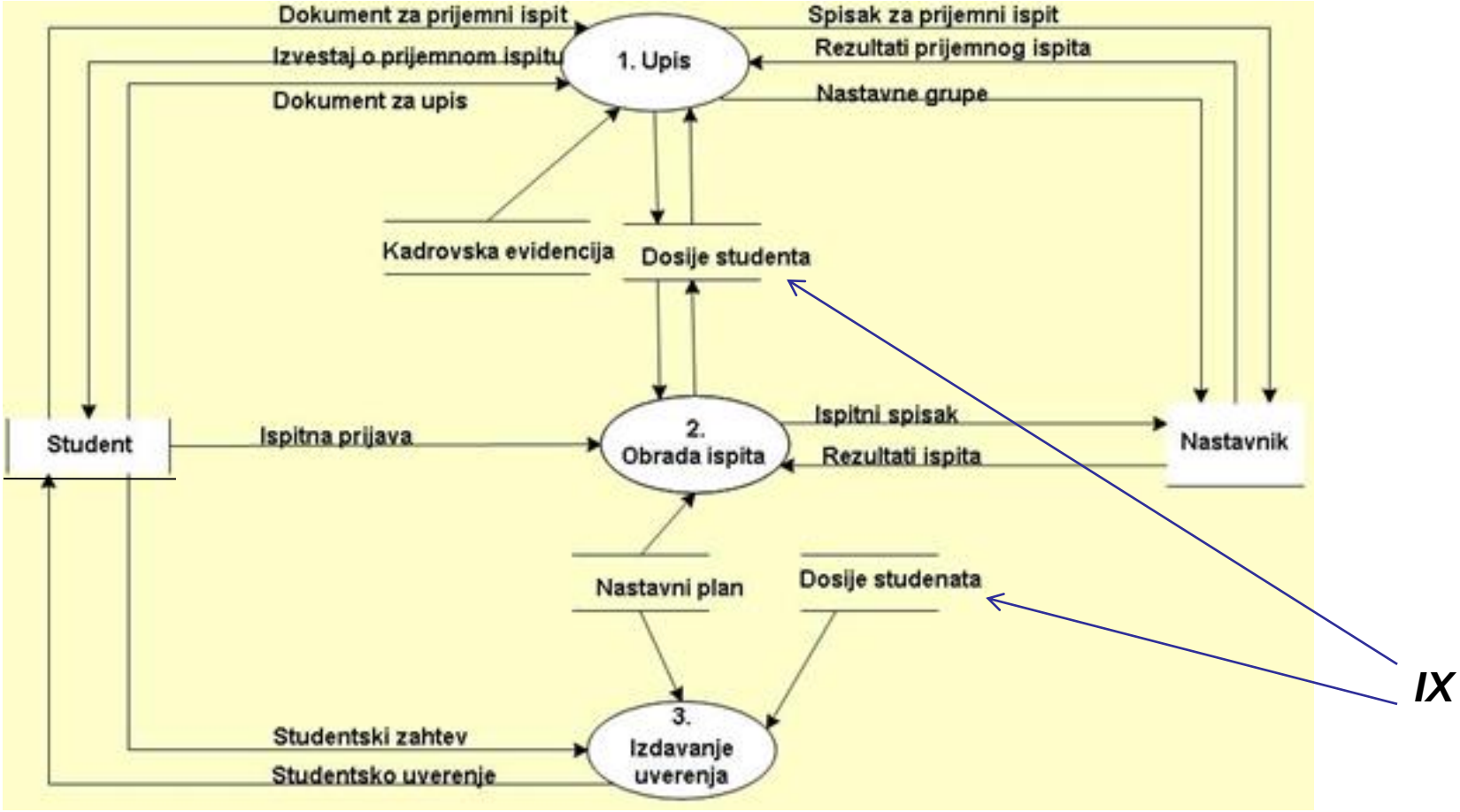
Pravila kreiranja DTP-a

- ***(VIII)*** ***Svaki interfejs*** mora da ima ***barem jedan, bilo ulazni, bilo izlazni tok podataka***, inače bi bio izolovan od ostalog dela sistema

- ***(IX)*** ***Dozvoljeno*** je višestruko ***ponavljanje skladišta i interfejsa***, da bi se ***minimiziralo presecanje tokova podataka***

Pravila kreiranja DTP-a

- Jedan konkretan DTP konstruisan saglasno navedenim pravilima ***I - IX***



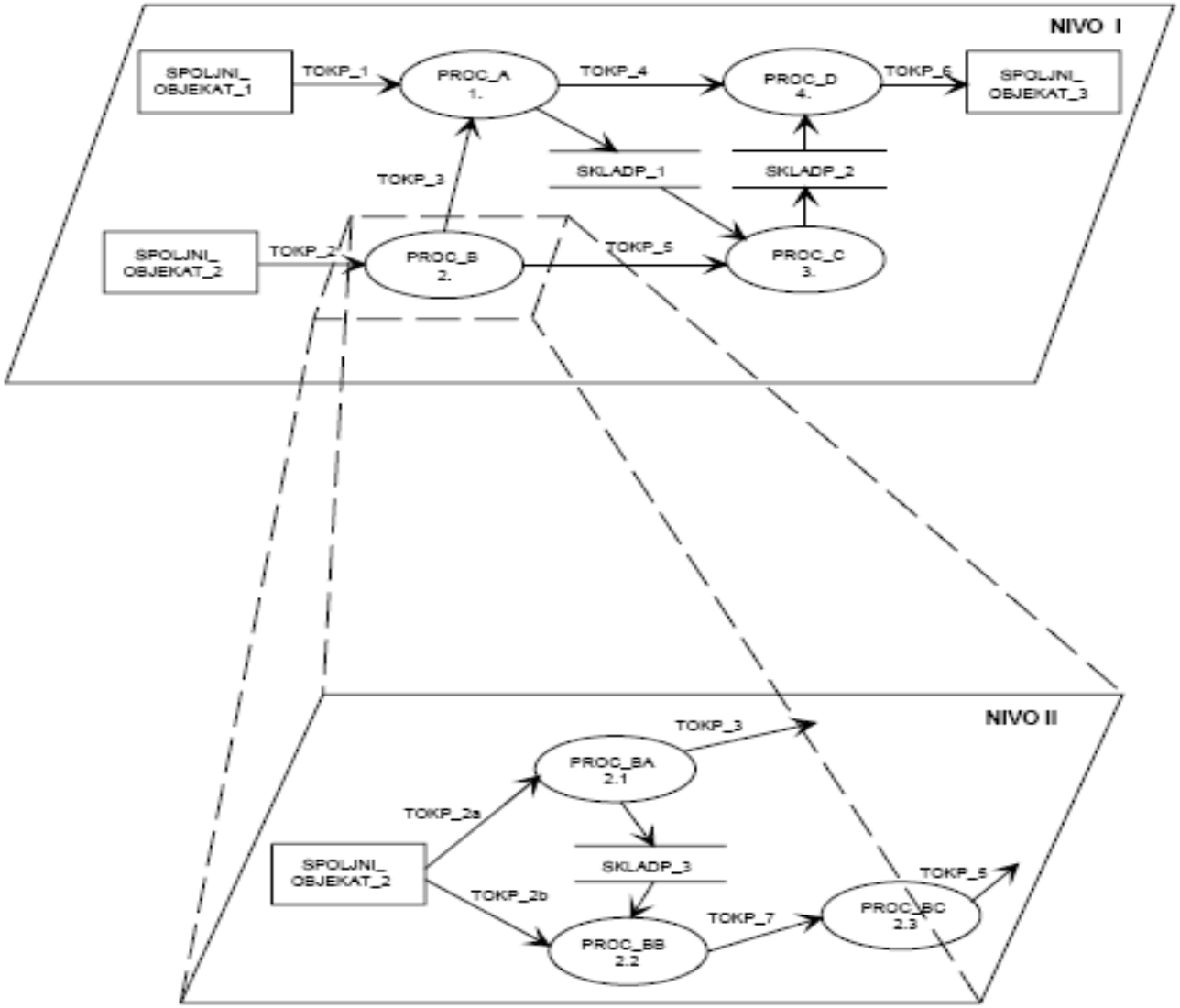
Hijerarhijska dekompozicija DTP-a

- Jedan ***informacioni sistem*** može biti ***veoma složen***
- Može sadržati ***veliki broj***
  - procesa,
  - tokova podataka,
  - skladišta podataka i
  - spoljnih objekata
- ***Jasna*** i ***detaljna*** ***specifikacija sistema*** zahteva da se i na predstavljanje sistema pomoću DTP-a ***primeni metoda apstrakcije***

Hijerarhijska dekompozicija DTP-a

- Primena ***metode apstrakcije*** na DTP
- Postiže se ***hijerarhijskom dekompozicijom DTP-a***
- Hijerarhijska dekompozicija DTP se izvodi na taj način što se ***jedan proces višeg nivoa apstrakcije*** dekomponuje i ***prikazuje pomoću novog celokupnog DTP-a*** na nižem nivou apstrakcije

Hijerarhijska dekompozicija DTP-a

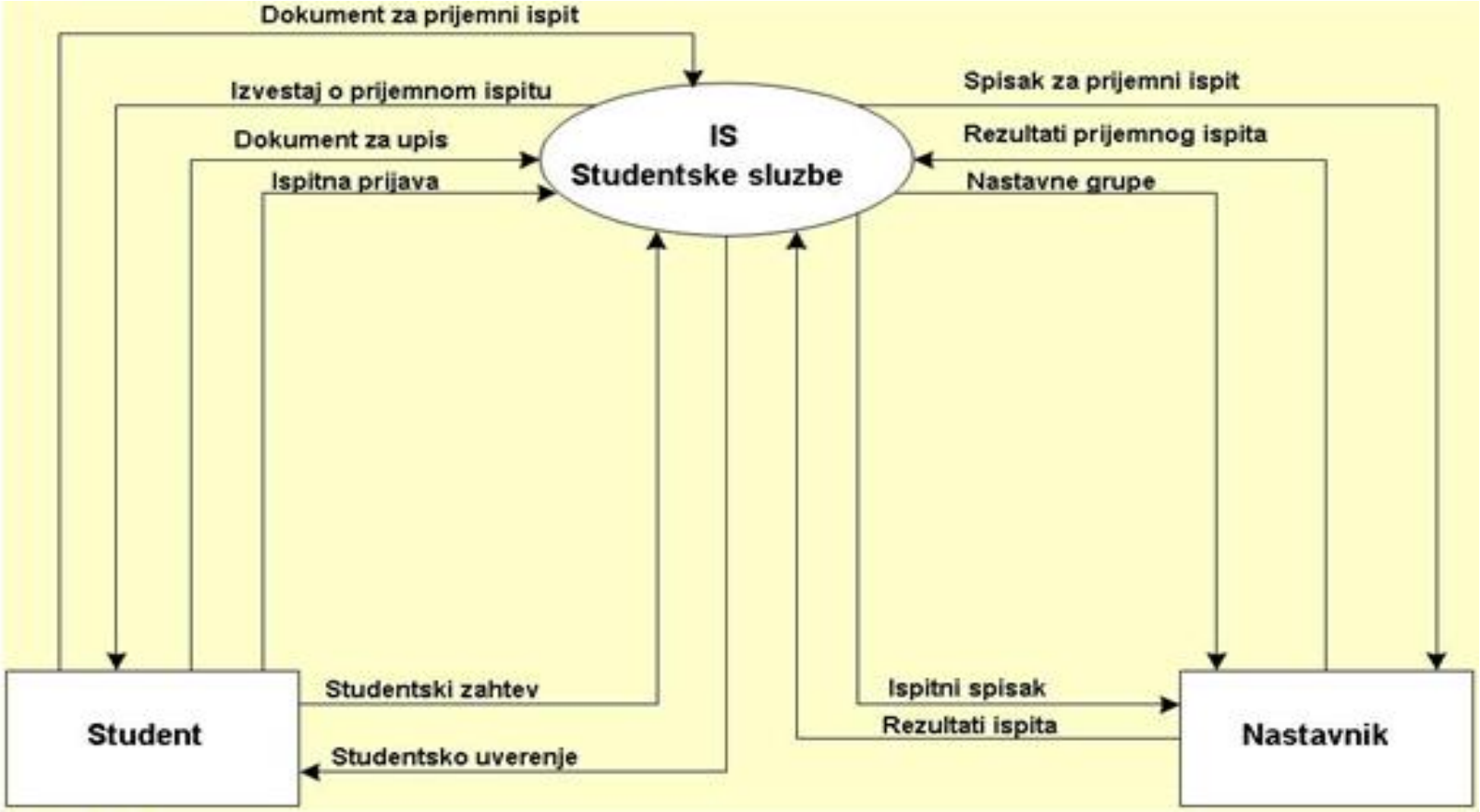


Hijerarhijska dekompozicija DTP-a - Pravila

- ***(I) Dijagram najvišeg nivoa***, koji po pravilu sadrži:
  - Samo ***jedan proces*** koji predstavlja ceo IS,
  - ***Interfejse*** sa kojima IS komunicira i
  - Odgovarajuće ***tokove podataka***
- Naziva se ***dijagram konteksta***

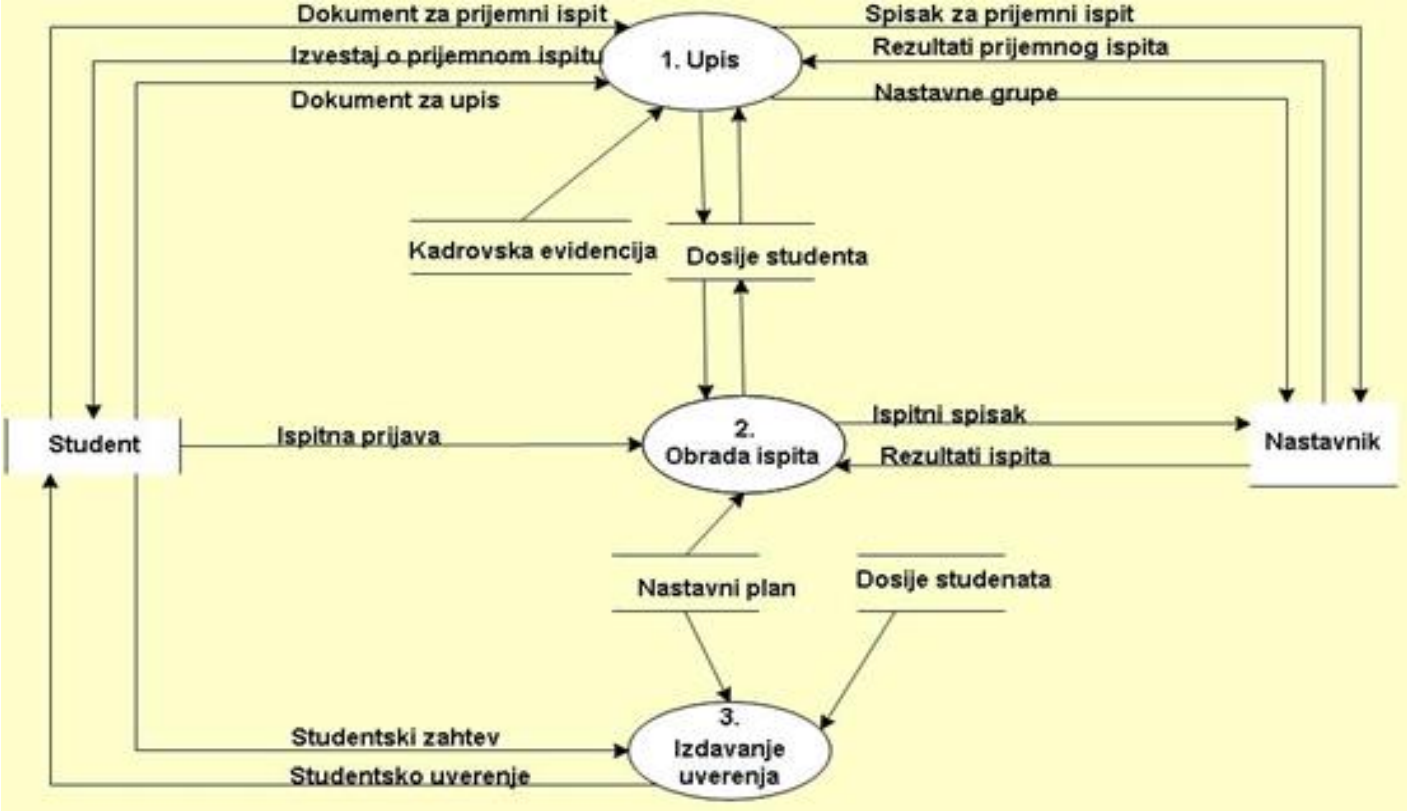
Hijerarhijska dekompozicija DTP-a - Pravila

- Primer ***dijagrama konteksta*** za ***Informacioni sistem studentske službe*** na jednom fakultetu



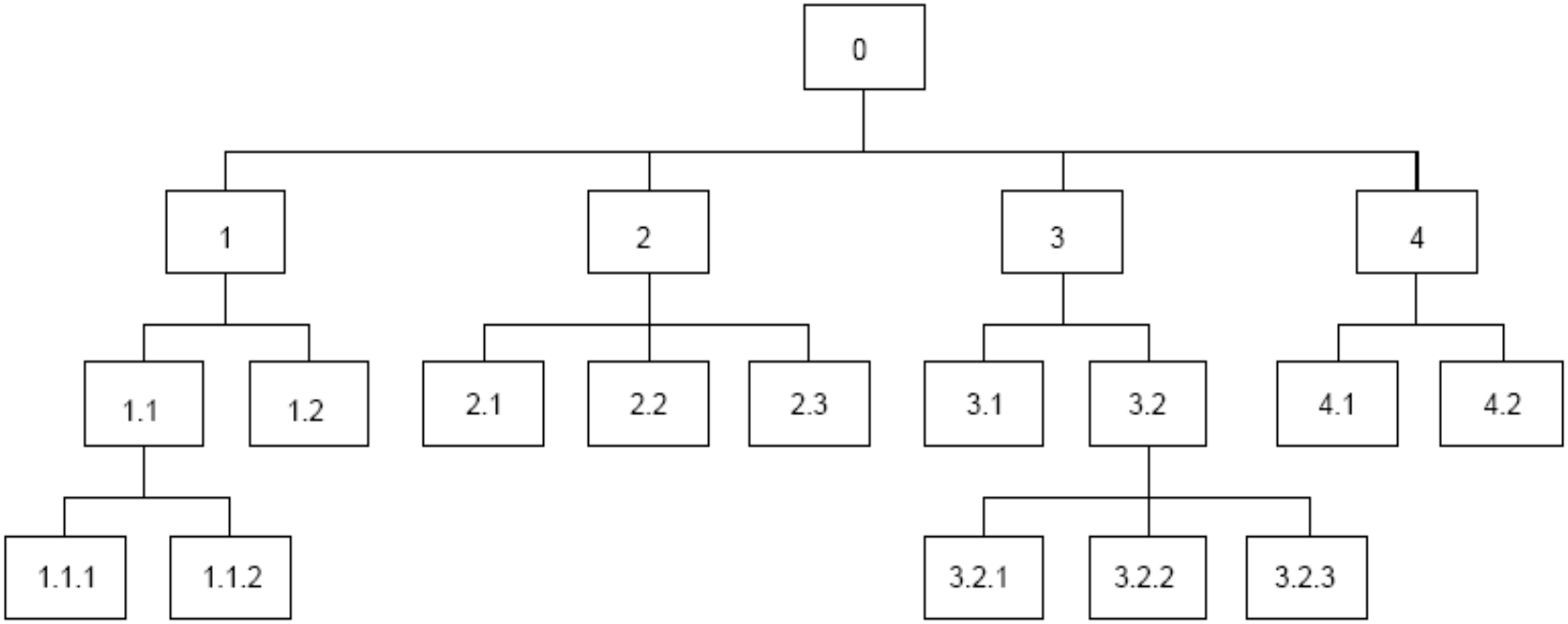
Hijerarhijska dekompozicija DTP-a - Pravila

- ***(II)*** Dijagram ***prvog nivoa*** (***root*** dijagram) pretstavlja dekompoziciju dijagrama konteksta
- ***Procesi*** na njemu označavaju se brojevima ***1, 2, 3, ...,***



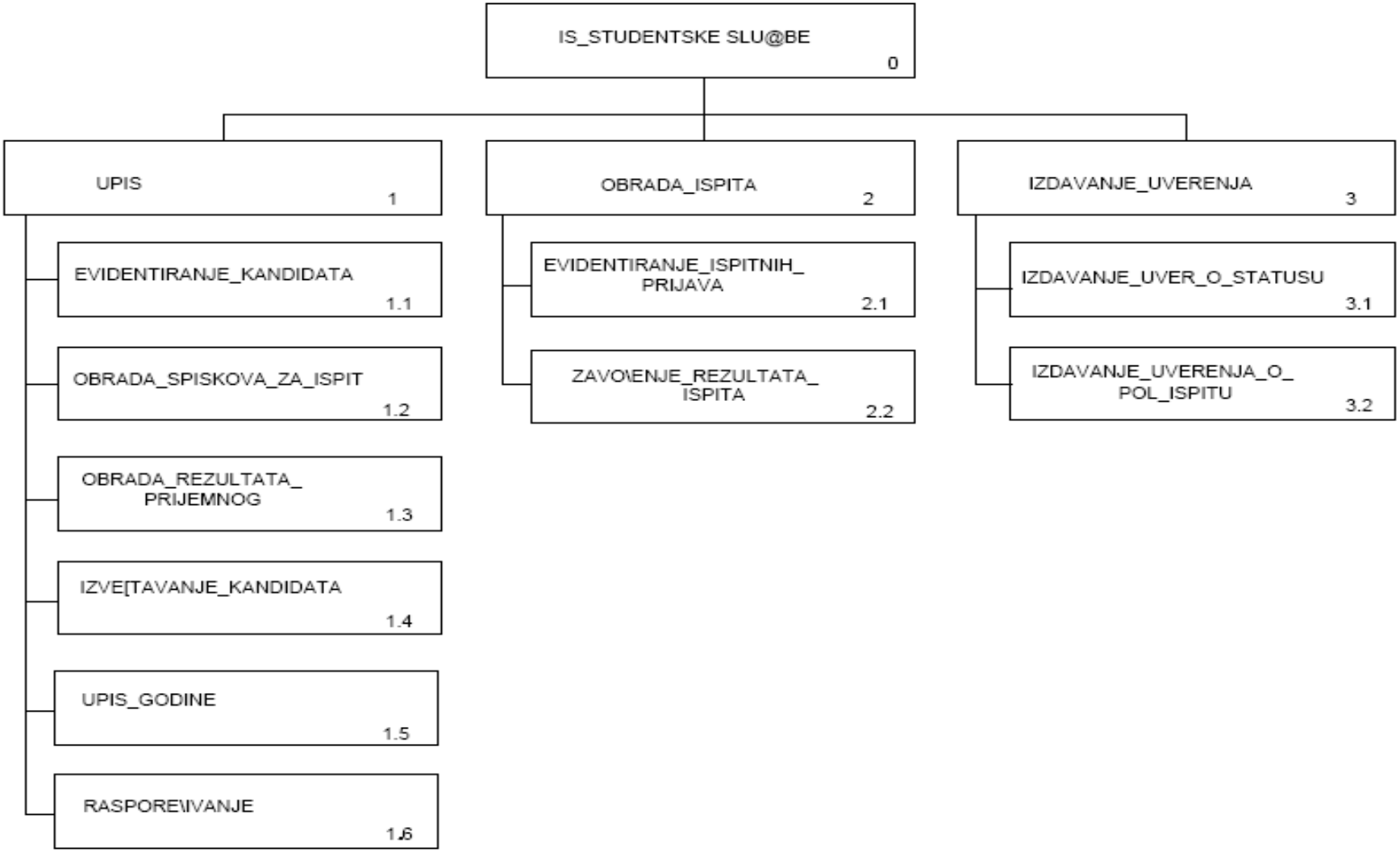
Hijerarhijska dekompozicija DTP-a - Pravila

- ***(III)*** ***Celokupan skup***, ili ***neki podskup hijerarhiski dekomponovanih dijagrama*** često se u dokumentaciji predstavlja ***dijagramom dekompozicije*** **Opšti *dijagram dekompozicije*:**



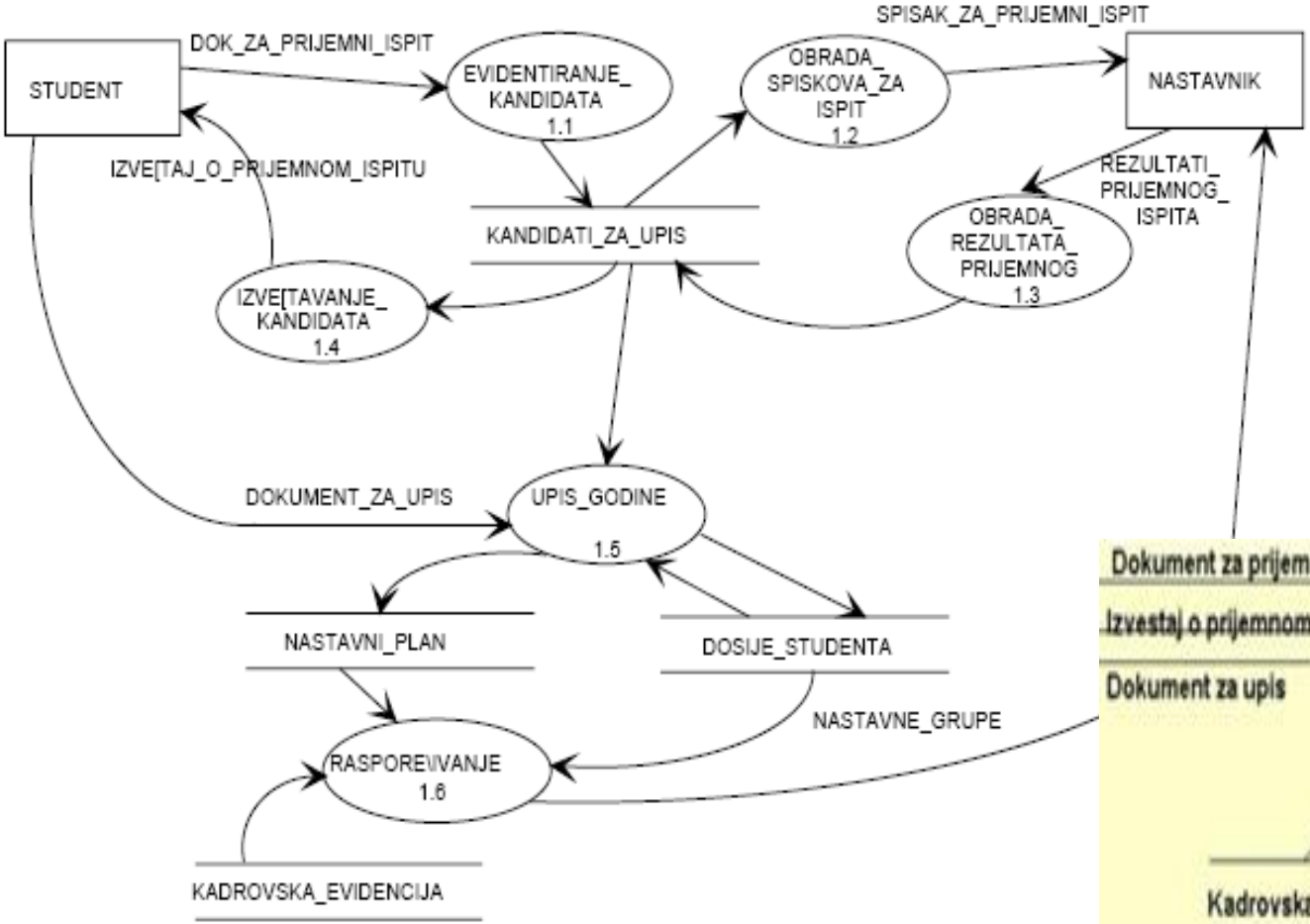
Hijerarhijska dekompozicija DTP-a - Pravila

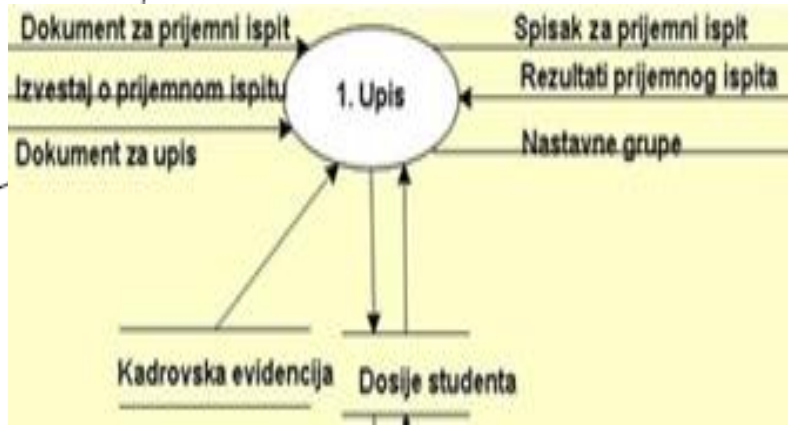
Dijagram dekompozicije za *IS studentske službe*:



Hijerarhijska dekompozicija DTP-a - Pravila

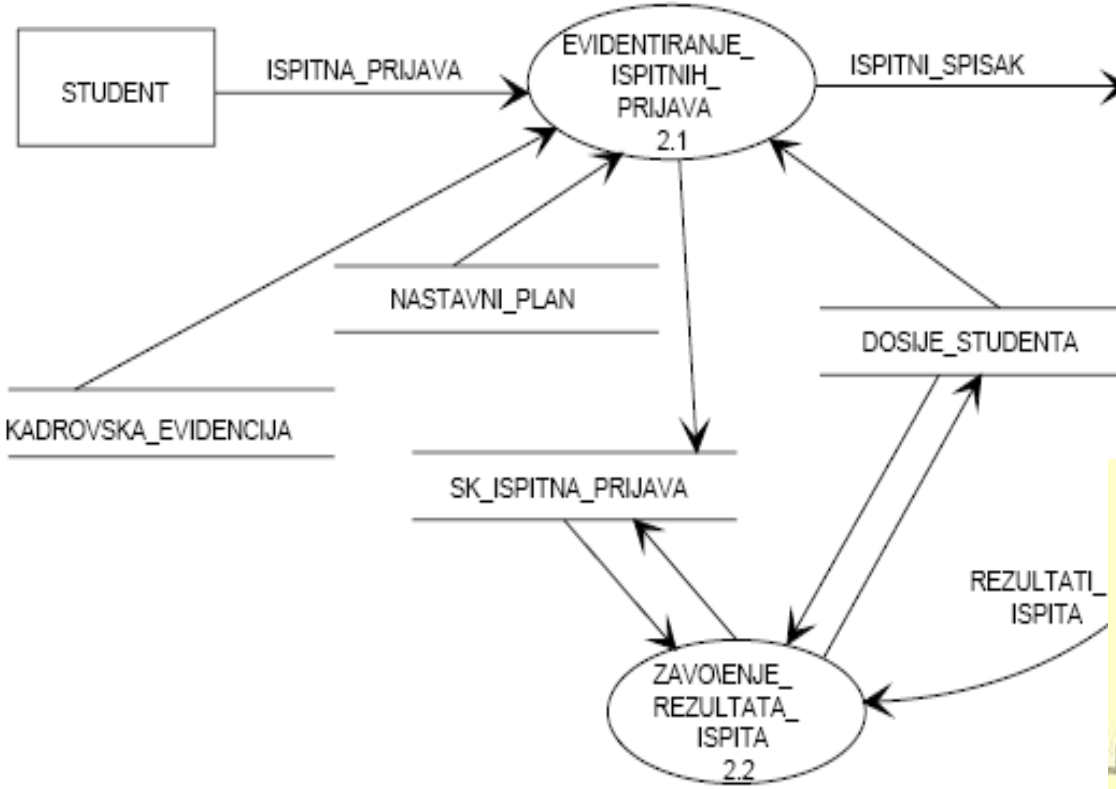
Dekompozicija procesa *Upis (1)*:

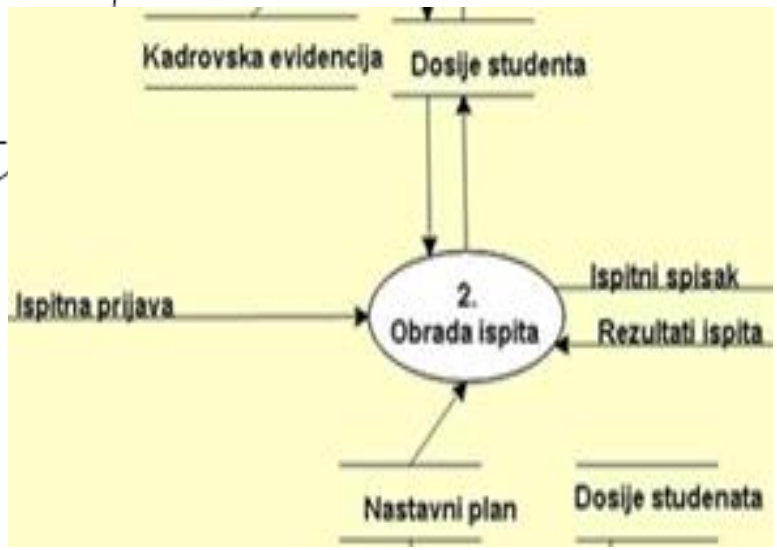




Hijerarhijska dekompozicija DTP-a - Pravila

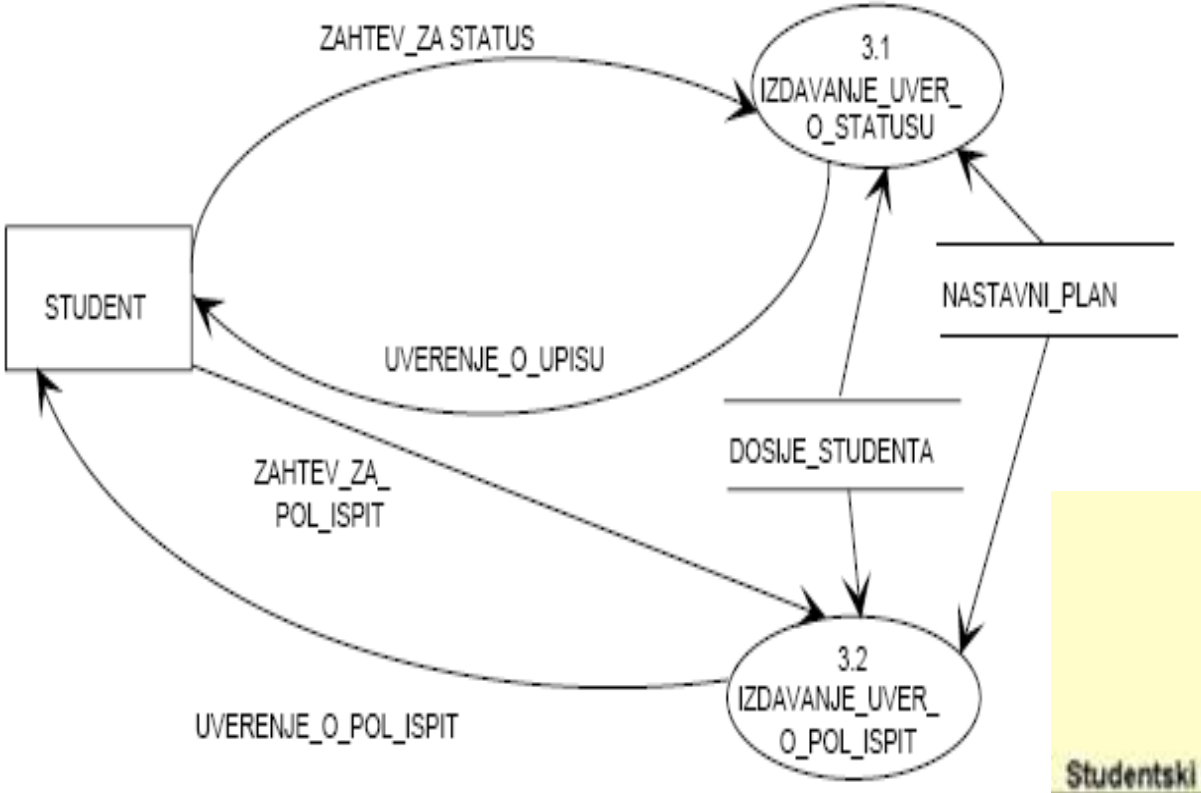
Dekompozicija procesa *Obrada ispita (2)*:

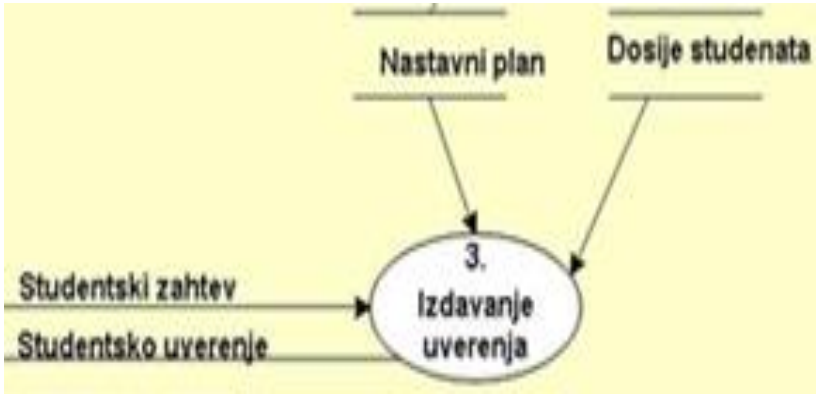




Hijerarhijska dekompozicija DTP-a - Pravila

Dekompozicija procesa *Izdavanje uverenja (3)*:





Hijerarhijska dekompozicija DTP-a - Pravila

- ***(IV)*** ***Procesi*** koji se dalje ***ne dekomponuju*** (procesi 1.1 do 1.5, 2.1 i 2.2) se nazivaju - ***primitivni procesi***
- Za njih se daje ***specifikacija logike*** njihove realizacije
- Opis logike primitivnih procesa naziva se ***mini-specifikacija sistema***

Hijerarhijska dekompozicija DTP-a - Pravila

- ***(V)*** Pored procesa, ***mogu se*** ***dekomponovati i tokovi i skladišta***
- Dekompozicija tokova i skladišta se ***ne prikazuje na DTP-u***, ***već u Rečniku podataka***, pomoću sintakse za opis strukture podatka, o čemu će kasnije biti više reči

Hijerarhijska dekompozicija DTP-a - Pravila

- ***Primer***: ***Zahtevi studenata*** i ***uverenja*** prikazana kao ***jedinstvenim tokovima*** podataka na ***dijagramu prvog nivoa*** – razlažu se na odgovarajuće komponente na dijagramu ***Izdavanje uverenja (3)***

STUD_ZAHTEV: [ZAHTEV_ZA_STATUS, ZAHTEV_ZA_POL_ISPITE]

STUD_UVERENJE: [UVERENJE_O_UPISU, UVERENJE_O_POL_ISPIT]

Ekskluzivna unija

Hijerarhijska dekompozicija DTP-a - Pravila

- ***(VI)*** ***Najznačajnije pravilo*** koje se mora poštovati pri dekompoziciji procesa
- ***Pravilo balansa tokova***:
- ***Ulazni i izlazni tokovi*** na celokupnom DTP-u koji je dobijen ***dekompozicijom nekog procesa P moraju biti ekvivalentni*** sa ulaznim i izlaznim tokovima toga procesa P na dijagramu višeg nivoa
- Pri tome se ***uzima u obzir*** ***dekompozicija tokova*** predstavljena u rečniku podataka

Hijerarhijska dekompozicija DTP-a - Pravila

- Pravilo balansa tokova na drugi način
- ***Svi tokovi*** koji ulaze, odnosno izlaze iz jednog procesa, ***moraju se pojaviti*** kao ulazni, odnosno izlazni tokovi ***na dijagramu*** u kojem je ***posmatrani proces dekomponovan***
- Na tom dijagramu ***ne može*** se ***pojaviti*** ***nijedan drugi*** ulazni i izlazni tok
- Na primer, svi tokovi podataka sa dijagrama konteksta, pojavljuju se kao ulazni i izlazni tokovi na dijagramu prvog nivoa dekompozicije

Hijerarhijska dekompozicija DTP-a - Pravila

- ***(VII)*** ***Skladišta podataka***, sa sistemske tačke gledišta, pretstavljaju ***stanja sistema***, odnosno fundamentalne, unutrašnje karakteristike kako celog IS, tako i svakog pojedinačnog procesa
- Zato, ***skladišta podataka od nivoa prvog pojavljivanja moraju se pojavljivati na svim nižim nivoima***

Hijerarhijska dekompozicija DTP-a - Pravila

- ***(VIII)*** O ***metodološkim aspektima dekompozicije***, o tome
  - ***kako*** se dekomponuje,
  - ***na koliko*** nivoa,
  - ***na kom nivou*** se prestaje sa dekompozicijom
- Za sada navodimo samo ***opštu preporuku***, da jedan DTP, po pravilu, treba da sadrži ***2 - 7 procesa***
- Veći broj procesa od ovoga značio bi da je "preskočen" jedan nivo apstrakcije

Rečnik podataka SSA

- ***Rečnik podataka*** daje ***opis*** ***(I)*** ***strukture*** i ***(II)*** ***sadržaja*** ***svih***
  - tokova i
  - skladišta podataka
- Sa stanovišta ***logičke strukture*** oni predstavljaju neku ***kompoziciju polja***
- Da bi ***precizno definisali*** ***logičku strukturu skladišta i tokova*** i definisali ***sintaksu rečnika*** neophodno je da ***uvedemo definicije*** svih ***koncepata rečnika***

Rečnik podataka SSA
Polja i domeni

- (**I**) ***Polje*** je ***elementarna (atomična) struktura*** koja se dalje ***ne dekomponuje*** i koja ***ima svoju vrednost***
- Na primer, u ***indeksu***, polja su

BROJ_INDEKSA, IME_I_PREZIME, OCENA, STATUS

i slično…..

Rečnik podataka SSA
Polja i domeni

- (**II**) Polja svoje *vrednosti* uzimaju iz ***skupova vrednosti*** koji se nazivaju ***domenima***
- Domeni mogu biti:

1. ***"predefinisani"***, odnosno standardni ***domeni definisani u većini programskih jezika***,
   - INTEGER, CHARACTER, REAL, LOGICAL i DATE
2. ***"semantički"***, kada se ***posebno definišu***, preko ***(I)*** svoga ***imena***, ***(II)*** predefinisanog ***domena*** i, eventualno, ***(III) ograničenja*** na mogući skup vrednosti predefinisanog domena

Rečnik podataka SSA
Polja i domeni

- Opšta ***sintaksa za iskazivanje ograničenja*** na predefinisani skup vrednosti za semantički domen je:

definicija_semantičkog__domena ::

naziv_domena 'DEFINED AS' predefinisani_domen

[ograničenje]

- Na primer, semantički domen SEMESTRI se definiše kao

SEMESTRI DEFINED_AS INTEGER (2) BETWEEN 1,10

Rečnik podataka SSA
Polja i domeni

- (**III**) Činjenica da ***polje*** uzima vrednost iz nekog domena ***označava se*** na sledeći način:

  naziv polja: domen [ograničenje]

- Na primer:
  - BI: CARACTER(7)
  - SEMESTAR: **SEMESTRI**
  - OCENA: INT(2) **IN (5,6,7,8,9,10)**
  - SEMESTVŠ: SEMESTRI **IN (1,2,3,4)** *Semestar više škole*

Rečnik podataka SSA
Polja i domeni

- (**IV**) ***Dva polja*** su ***semantički slična*** samo ako su definisana nad ***istim domenom***
- ***Semantički domeni*** ***uspostavljaju razliku*** između pojedinih istovrsnih predefinisanih domena koji nemaju semantičku sličnost
- Na primer, ako bi se i polje SEMESTAR definisalo direktno nad predefinisanim domenom INTEGER, kao

SEMESTAR: INTEGER (2) BETWEEN 1,10

tada bi polja **OCENA** i **SEMESTAR** bila ***semantički slična***, mogla bi se povezivati operatorima

Rečnik podataka SSA
Polja i domeni

- Na primer:

  IF SEMESTAR > OCENA THEN ....

  što očigledno ***nema smisla***. Prethodni izraz bi bio i ***sintaksno nekorektan***
- Različita polja se ***mogu povezati*** nekim operatorom samo ako su definisana ***nad istim domenom*** i ako je operator definisan u tom domenu
- Trebalo bi ***težiti*** što je moguće ***većem korišćenju semantičkih domena***
  - Retko koji računarski jezik podržava koncept semantičkog domena

Rečnik podataka SSA
Polja i domeni

- (**V**) ***Predefinisani domeni*** su standardni ***progrmsko-jezički domeni***, koji se ovde definišu na sledeći način:
  - INTEGER(***dužina***)
  - CHARACTER(dužina)
  - REAL(dužina celokupnog broja, dužina iza zareza)
  - LOGICAL
  - DATE

Rečnik podataka SSA

- (**VI**) Pored ***ograničenja na vrednosti polja***, definišu se i druga
- ***Ograničenja*** mogu biti ***(I)*** ***prosta*** i ***(II)*** ***složena***
- Dozvoljena ***prosta ograničenja*** su:
- ***(a)*** Θ ***konstanta***, gde je Θ (teta) bilo koji ***operator poređenja*** (na primer, <, >, =, <=, >= za brojne domene), a ***konstanta*** je neka ***definisana vrednost*** iz datog domena
- Na primer: **STAROST: INT(2) < 65**

Rečnik podataka SSA
Polja i domeni

- ***(b) BETWEEN konstanta1, konstanta2*** gde su konstante vrednosti iz datog domena
- Na primer: **SEMESTAR: INTEGER (2) BETWEEN 1,10**
- ***(c) IN (lista vrednosti)***, gde se lista formira od konstanti iz odgovarajućeg domena
- Na primer: **OCENA: INT(2) IN (5,6,7,8,9,10)**
- ***(d) NOT NULL***, kada dato polje ne može da dobije "null vrednost", odnosno mora uvek da ima vrednost
- Na primer:

  BROJ_INDEKSA: CHARACTER (7) NOT NULL

Rečnik podataka SSA
Polja i domeni

- ***Složena ograničenja*** se formiraju
- od prostih ili drugih složenih ograničenja
- vezujući ih **logičkim operatorima** ***AND***, ***OR*** i ***NOT***
- Na primer:

STAROST: INT(2) < 65 AND NOT NULL

Rečnik podataka SSA
Polja i domeni

- ***Polje*** ***je*** ***atomična***, nedeljiva ***komponenta***
- Ponekad je potrebno ***ući i u njegovu strukturu***
  - Npr. ***za potrebe definisanja ograničenja***
- Zbog toga je uvedena ***funkcija*** - ***SUBSTRING***
- Ima za cilj da ***izvuče deo polja*** i prikaže neke karakteristike toga dela
- Funkcija ***SUBSTRING*** se definiše kao

SUBSTRING (*položaj prvog člana*, *položaj poslednjeg člana*)

Rečnik podataka SSA
Polja i domeni

- Na primer:

JMBG CHARACTER(13) SUBSTRING(1,2) ≤ 31 AND SUBSTRING(3,4) ≤ 12 AND SUBSTRING(5,7) BETWEEN 000, 999 ....

- Prva dva karaktera predstavljaju datum u mesecu, druga dva mesec u godini itd.

Rečnik podataka SSA
Strukture

- ***Struktura*** tokova podataka i skladišta predstavlja neku ***kompoziciju polja*** - konstrukciju čije su komponente polja
- Kao ***komponenta*** jedne strukture može se, ***pored polja***, pojaviti i ***druga definisana struktura***
- ***Operacije*** (konstrukcije) kojim se ***od komponenata*** gradi ***struktura*** mogu biti:
  1. ***Agregacija*** komponenti,
  2. ***Eksluzivna specijalizacija*** (***unija***) komponenti,
  3. ***Neeksluzivna specijalizacija*** (***unija***) komponenti,
  4. ***Skup*** komponenti

Rečnik podataka SSA Strukture

- ***Agregacija komponenti***, pretstavlja se kao ***lista komponenti*** koje je čine u "špicastim" zagradama - **<a,b,c>**
- ***Vrednost agregacije*** je ***n-torka*** u kojoj svaki elemenat ima vrednost odgovarajuće komponente
- Na primer,

  ISPITNA_PRIJAVA: < BROJ_INDEKSA,
  IME_STUDENTA,
  NAZIV_PREDMETA,
  DATUM_POLAGANJA,
  OCENA,
  IME_NASTAVNIKA
  >

Rečnik podataka SSA Strukture

- ***Eksluzivna specijalizacija (unija) komponenti***, predstavlja se kao lista komponenti u ***uglastim zagradama*** –

[a,b,c]

- Označava da se u strukturi pojavljuje ***eksluzivno jedna od navedenih komponenti***, ***ili a ili b ili c***

- Ako se u uglastoj zagradi pojavi samo jedna komponenta, kao **[a]**, to znači da se u strukturi ***ova komponenta javlja ili ne javlja***

Rečnik podataka SSA Strukture

- ***Neeksluzivna specijalizacija (unija) komponenti***, pretstavlja se kao lista komponenti u ***kosim zagradama*** - **/a,b,c/**
- Označava da se u odgovarajućoj srtukturi pojavljuje bilo ***samo jedna*** komponenta, ***bilo dve***, ***bilo sve***
- ***Primer*** za neeksluzivnu specijalizaciju je:

  STUD_ZAHTEV: / ZAHTEV_ZA_UVER_STATUS, ZAHTEV_ZA_UVER_POL_ISPIT /

- Student može da podnese bilo zahtev za uvrenje o statusu (redovan ili vanredan) bilo zahteva za uverenje o položenim ispitima, bilo oba

Rečnik podataka SSA Strukture

- ***Skup komponenti***, koji se pretstavlja u vitičastim zagradama, na primer **{a}**, i koja kaže da se u odgovarajućoj strukturi ***komponenta može da pojavi više puta***
- ***Primer*** za skup komponenti je:

UVERENJE_O_POL_ISPIT: < BROJ_INDEKSA,

IME_STUDENTA,

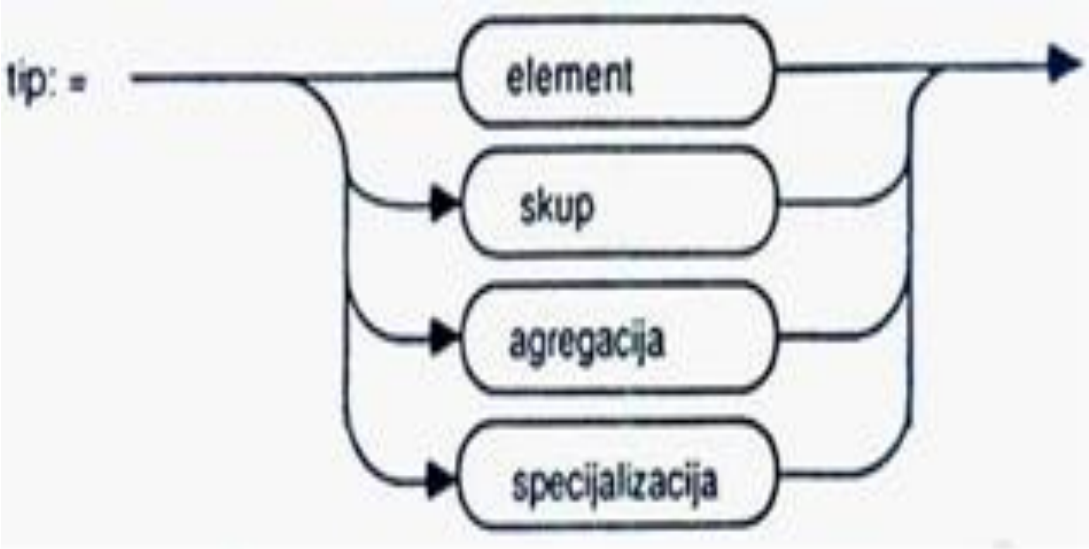
{< NAZIV_PREDMETA, OCENA>},

PROSEČNA_OCENA

>

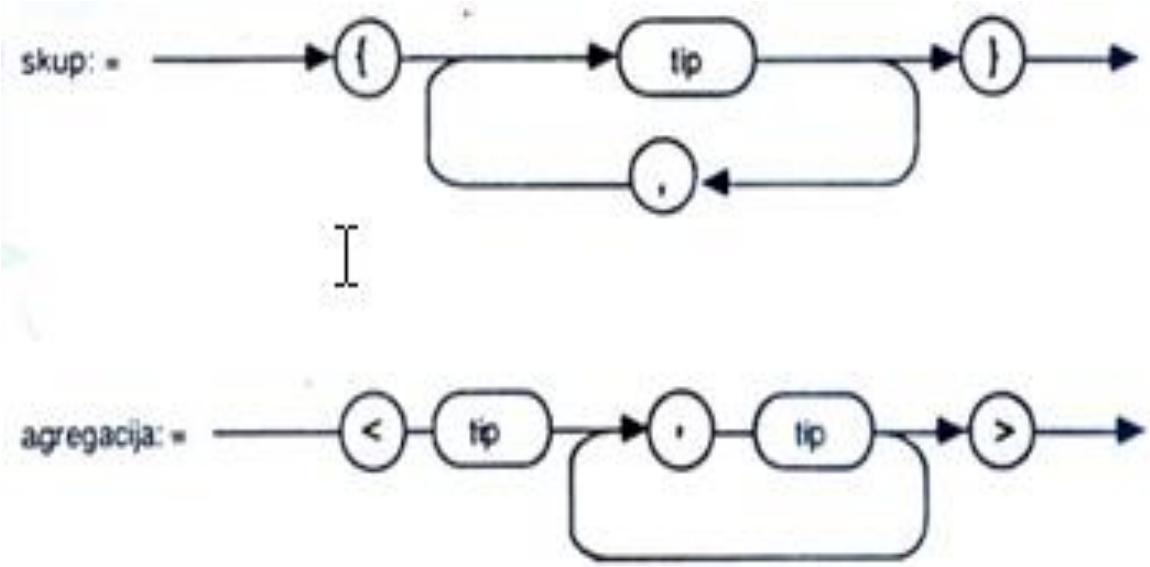
Rečnik podataka SSA Strukture

- ***Za opisivanje strukture*** skladišta ili tokova podataka mogu se koristiti i ***sintaksni dijagrami***



Rečnik podataka SSA Strukture

- ***Za opisivanje strukture*** skladišta ili tokova podataka mogu se koristiti i ***sintaksni dijagrami***



Specifikacija Rečnika podataka

- ***BNF notacija*** (***Backus-Naur Form***) daje ***potpunu sintaksu*** za specifikaciju Rečnika podataka
- Koristi se za opis sintakse programskih jezika
- ***Modifikovana BNF notacija***
- ***Ključne reči*** sintakse su date
  1) ***Velikim slovima***, ***ili***
  2) ***Između znakova navoda " "***
- ***** - Komentar – omeđen karakterom *
- **::** - Struktura sa leve strane se sastoji od delova sa desne (***"sastavljeno od"***)

Specifikacija Rečnika podataka

Rečnik podataka ::

STRUCTURES ***tačka-zarez-lista-struktura***

FIELDS ***tačka-zarez-lista-polja***

DOMAINS ***tačka-zarez-lista-domena***

CONSTRAINT_FUNCTIONS ***tačka-zarez-lista-logičkih-funkcija***

Specifikacija Rečnika podataka

- Ili,
- ***Rečnik podataka*** sardrži ***četiri*** osnovna ***segmenta***:
  1. za ***opis struktura*** skladišta i tokova
  2. za ***opis polja***,
  3. za ***opis semantičkih domena*** i
  4. za ***definiciju logičkih funkcija*** preko kojih se iskazuju ***složenija ograničenja***

Specifikacija Rečnika podataka

- **Segment za <u>opis struktura</u> (STRUCTURES)**

tačka-zarez-lista-struktura :: struktura | struktura ";" tačka-zarez-lista-struktura

[*tačka-zarez-lista-struktura** – opisuje ***listu struktura međusobno ***odvojenih sa ;]***

struktura :: naziv_strukture ":" opis_strukture

opis_strukture ::

"<" zarez-lista-komponenti ">“ * agregacija *

| "[" zarez-lista-komponenti "]” * eksl. specijal.*

| "/" zarez-lista-komponenti "/" * neeksl. specijal.*

| "{" zarez-lista-komponenti "}“ * skup *

komentar-objašnjenje

Specifikacija Rečnika podataka

zarez-lista- komponenti ::

komponenta | komponenta "," zarez-lista-komponenti

komponenta ::

naziv_polja * naziv polja koje je opisano u Rečniku *

| naziv_strukture * naziv strukt. koja je opisana u Rečn. *

| opis_strukture

| polje * potpuna definicija polja sa nazivom i domenom *

| struktura * potpuna definicija strukture sa nazivom i opisom *

mogu da posluže da se *u okviru definicije jedne strukture*, definiše i *druga struktura i* neko *polje* koji neće biti posebno definisani u Rečniku

Specifikacija Rečnika podataka

- **Segment za <u>opis polja</u> (FIELDS)**

tačka-zarez-lista-polja ::

polje | polje ";" tačka-zarez-lista-polja

polje ::

naziv_polja ":" opis_polja

| naziv_polja ":" naziv_domena

| naziv_polja ":" naziv_domena DEFINED_AS opis_polja

opis_polja ::

tip_polja

| tip_polja ograničenje

Specifikacija Rečnika podataka

tip_polja ::

INTEGER "(" dužina ")"

| CHARACTER "(" dužina ")"

| REAL "("ukupna_dužina "," dužina_posle_zapete ")"

| LOGICAL

| DATE

ograničenje :: prosto_ograničenje | složeno_ograničenje

prosto_ograničenje :: ⊖ **vrednost_iz_domena**

| BETWEEN vrednost_iz_domena "," vrednost_iz_domena

| IN "("skup_vrednosti")"

| NOT NULL

Specifikacija Rečnika podataka

⊖ :: > | < | =

složeno_ograničenje :: ograničenje AND ograničenje

| ograničenje OR ograničenje

| NOT ograničenje

| IF ograničenje THEN ograničenje

| naziv_logičke_funkcije

Specifikacija Rečnika podataka

- **Segment za opis domena (DOMAINS)**

tačka-zarez-lista-domena :: domen | domen ";"tačka-zarez-lista-domena

domen :: naziv_domena DEFINED_AS opis_polja

Specifikacija Rečnika podataka

- **Segment za <u>opis logičkih funkcija</u> (CONSTRAINT_FUNCTIONS)**

tačka-zarez-lista-logič_funkcija :: logička_funkcija | logička_funkcija ";"tačka-zarez-lista-logič_funkcija

logička_funkcija :: naziv_logičke_funkcije"=" logički_izraz

Primer opisa strukture u Rečniku podataka

```
STRUCTURES

DOK_ZA_PRIJEMNI_ISPIT:  <       DIPLOMA,
                                        {<SVEDOČANSTVO>},
                                        {<NAGRADA>}
                                >;


DIPLOMA:    <       NAZIV_ŠKOLE:CHAR (20),
                     VRSTA_ŠKOLE:VRSTE_ŠKOLA,
                    IME_KAND,
                    DATUM_DIPL:DATE
            >;
```

Primer opisa strukture u Rečniku podataka

```
SVEDOČANSTVO:   <       NAZIV_ŠKOLE,
                        VRSTA_ŠKOLE,
                        IME_KAND,
                        DATUM_SVED,
                        {<      NAZIV_ŠKOL_PRED,
                                OCENA_ŠKOL_PRED:INT(1) IN (1,2,3,4,5)
                        >},
                        PROSEK: REAL(1,2) < 5.00
                >;
```

Primer opisa strukture u Rečniku podataka

FIELDS

NAZIV_ŠKOLE: CHAR(20);
VRSTA_ŠKOLE: VRSTE_ŠKOLA;
IME_KAND: CHAR(25);
DATUM_DIPL: DATE;
DATUM_SVED: DATE;
NAZIV_ŠKOL-PRED: CHAR(15);
OCENA_ŠKOL_PRED: INT(1) IN (1,2,3,4,5);
PROSEK: REAL(1,2) < 5.00

DOMAINS

VRSTE_ŠKOLA: CHAR(20)
IN ('GIMNAZIJA', 'SREDNJETEHNIČKA', 'OSTALE')

Primer opisa strukture u Rečniku podataka

- Navedeni primer ***obiluje tzv. “mešovitim" zapisima*** u Rečniku (prilikom opisa strukture)
  - Daje se ***samo naziv polja***, ili
  - ***Uz naziv*** daje se ***i domen***, ili
  - Ponekad ***uz semantički domen*** i ***njegova definicija***
- Ovakav način - ***otežava čitanje*** i razumevanje Rečnika, ali
- ***Veoma je pogodan*** kada se Rečnik kreira pomoću ***CASE alata***

Primer opisa strukture u Rečniku podataka

- Mogućnost de se ***odmah uz naziv polja definiše i njegov domen i ograničenje***, a ***uz semantički domen*** odmah ***da i njegova definicija***, - ***veoma je pogodna*** kada se Rečnik kreira pomoću ***CASE alata***
- Tada se ***na istom ekranu***, pri definiciji polja daju i njegov domen, definicija domena i ograničenje - ***nije neophodno ići kroz poseban postupak***
- Međutim, ***pri ručnom formiranju*** i prikazivanju Rečnika treba izbegavati "mešovite" zapise

Primer opisa strukture u Rečniku podataka

- Treba ***usvojiti sledeću praksu***:
  - ***u opisu struktura*** navode se ***samo nazivi polja***, bez njihovih domena i bez definicije domena,
  - definiše se ***posebna tabela za opis polja sa kolonama*** **NAZIV POLJA**, **DOMEN** (predefinisani ili semantički) i **OGRANIČENJE**,
  - definiše se ***posebna tabela za opis semantičkih domena sa kolonama*** **NAZIV DOMENA PREDEFINISANI DOMEN** i **OGRANIČENJE**

Primer opisa strukture u Rečniku podataka

```
STRUCTURES

DOK_ZA_PRIJEMNI_ISPIT:  <       DIPLOMA,
                                        {<SVEDOČANSTVO>},
                                        {<NAGRADA>}
                                >;

DIPLOMA:    <       NAZIV_ŠKOLE,
                    VRSTA_ŠKOLE,
                    IME_KAND,
                    DATUM_DIPL
            >;


SVEDOČANSTVO:   <       NAZIV_ŠKOLE,
                                VRSTA_ŠKOLE,
                                IME_KAND,
                                DATUM_SVED,
                                {<      NAZIV_ŠKOL_PRED,
                                        OCENA_ŠKOL_PRED
                                >},
                                PROSEK
                        >;
```

Primer opisa strukture u Rečniku podataka

FIELDS

NAZIV POLJA	DOMEN	OGRANIČENJE
NAZIV_ŠKOLE	CHAR(20) -	
VRSTA_ŠKOLE	VRSTE_ŠKOLA-	
IME_KAND	CHAR(25) -	
DATUM_DIPL	DATE -	
DATUM_SVED	DATE -	
NAZIV_ŠKOL-PRED	CHAR(15);-	
OCENA_ŠKOL_PRED	INT(1)	IN (1,2,3,4,5)
PROSEK	REAL(1,2)	< 5.00

Primer opisa strukture u Rečniku podataka

DOMAINS

NAZIV DOMENA	PREDEFINISANI DOMEN	OGRANIČENJE
VRSTE_ŠKOLA	CHAR(20)	IN ('GIMNAZIJA', 'SREDNJE TEHNIČKA', 'OSTALE')

Specifikacija logike primitivnih procesa

- ***Primitivni procesi*** su procesi na ***najnižem nivou dekompozicije***
- Oni su ***po pravilu sekvencijalni*** - ***redosled*** aktivnosti u okviru njih je ***definisan***
- Za njihovu specifikaciju se moraju koristiti neki ***alati za specifikaciju sekvencijalnih procesa*** - obično se zovu ***alati za opis logike procesa***
- Postoji čitav ***niz ovakvih alata***

Specifikacija logike primitivnih procesa

- Postoji čitav ***niz ovakvih alata***
  - ***Dijagram toka programa*** ("Flowchart"),
  - ***Tabela odlučivanja***,
  - ***Stabla odlučivanja***,
  - Strukturni prirodni jezik
  - Pseudo jezici (***pseudokodovi***)

Specifikacija logike primitivnih procesa

- ***Tabela odlučivanja***
- ***Tabelarni prikaz kompleksne logike*** u kojoj ***redovi*** predstavljaju ***uslove i moguće akcije***, a ***kolone*** prikazuju ***koja kombinacija uslova rezultira u određenu akciju***

Tabela odlučivanja	Pravilo	Pravilo	Pravilo
[Uslov]	vrednost	vrednost	vrednost
[Uslov]	vrednost	vrednost	vrednost
[Uslov]	vrednost	vrednost	vrednost
[Niz akcija]	X		
[Niz akcija]		X	
[Niz akcija]			X

Specifikacija logike primitivnih procesa

- ***Pseudokod***
- ***Strukturni prirodni jezik***
  - Koristi ***rečnik*** nekog ***prirodnog jezika*** (npr. srpskog, engleskog,..)
  - Za konstrukciju rečenica i složenih sklopova koriste se ***osnovne strukture strukturnog programiranja***
- ***Pseudo jezici (pseudokodovi)***
  - Bliži su programskim jezicima
  - Koristi ***rečnik*** nekog ***programskog jezika***
  - Za konstrukciju rečenica i složenih sklopova koriste se osnovne strukture strukturnog programiranja

Specifikacija logike primitivnih procesa

- ***Pseudokod***
- Primer iskaza u prirodnom jeziku:

  "Svaki komitent banke koji ima na računu više od 10.000, čiji je srednji mesečni bilans veći od 5.000 ili koji poseduje račun više od pet godina ...."

  može se interpretirati ***barem na dva načina***:
  - "Svaki komitent **(**koji ima na računu više od 10.000 ***AND*** srednji mesečni bilans veći od 5.000 **)** ***OR*** poseduje račun više od pet godina.."
  - "Svaki komitent koji ima na računu više od 10.000 ***AND*** **(**srednji mesečni bilans veći od 5.000 ***OR*** poseduje račun više od pet godina..**)**"

Specifikacija logike primitivnih procesa

- ***Pseudokod -*** **Osnovne strukture** za struktuiranje prirodnog jezika
- (**I**) ***Sekvencija***
- ***Aktivnosti*** u sekvencijalnom bloku ***se odvijaju po redosledu navođenja***
- Ponekad je pogodno da se i ***sekvencijalni blok*** akcija ***ograniči*** sa ključnim rečima **BEGIN** i **END**

Specifikacija logike primitivnih procesa

- ***Pseudokod -*** **Osnovne strukture** za struktuiranje prirodnog jezika
- (**I**) ***Sekvencija***
- Na primer, ***opšti*** pseudokod za ***proces Evidentiranje kandidata*** bi mogao da bude

```
BEGIN
Unesi podatke o diplomi;
Unesi podatke sa svedočanstava;
Unesi podatke o nagradama;
Ažuriraj datoteku KANDIDATI_ZA_UPIS;
END;
```

Specifikacija logike primitivnih procesa

- ***Pseudokod -*** **Osnovne strukture** za struktuiranje prirodnog jezika
- (**II**) ***Selekcija***
- ***Redosled*** aktivnosti ***zavisi od nekog uslova***
  - Ako je uslov ispunjen obavlja se jedan, a ako nije drugi blok akcija
- Opšti iskaz selekcije je

  IF uslov THEN blok_akcija_1 ELSE blok_akcija_2;
- Na primer,

IF BROJ POENA > 85 Upiši kandidata ELSE Odbi kandidata;

Specifikacija logike primitivnih procesa

- ***Pseudokod -*** **Osnovne strukture** za struktuiranje prirodnog jezika
- (**III**) ***Case struktura***
- ***Specijalni slučaj selekcije*** kada se u zavisnosti ***od vrednosti*** jednog parametra može izvršavati ***više različitih blokova akcija***
- Opšti iskaz za ovu strukturu je:

CASE parametar OF

vrednost_parametra_1: bloka_akcija_1

vrednost_parametra_2: bloka_akcija_2

...

...

vrednost_parametra_n: bloka_akcija_n

Specifikacija logike primitivnih procesa

- ***Pseudokod*** **- Osnovne strukture** za struktuiranje prirodnog jezika
- (**III**) ***Case struktura***
- Na primer,

CASE ***SEMESTAR*** **OF**

1: Stavi_studenta u grupu za prvu godinu

3: Stavi_studenta u grupu za drugu godinu

5: Stavi_studenta u grupu za treću godinu

7: Stavi_studenta u grupu za četvrtu godinu;

Specifikacija logike primitivnih procesa

- ***Pseudokod -*** **Osnovne strukture** za struktuiranje prirodnog jezika
- (**IV**) ***Iteracija***
- ***Blok akcija*** se ***ponavlja*** dok se neki ***uslov*** ne ispuni ili dok se akcije ne obave za ***sve objekte nekog skupa***
- Osnovni oblici ove strukture su:

  DO WHILE uslov blok_akcija

  – uslov se ispituje **pre svakog izvršenja** bloka akcija, ***moguće*** je da se blok akcija ***ne izvrši nijednom***,

Specifikacija logike primitivnih procesa

- ***Pseudokod -*** Osnovne strukture za struktuiranje prirodnog jezika
- (**IV**) ***Iteracija***

  DO UNTIL uslov blok_ akcija

  – uslov se ispituje **na kraju svakog izvršenja** bloka akcija, blok akcija se izvršava ***barem jedanput***

FOR EACH naziv_ skupa_ objekata DO blok_akcija

Specifikacija logike primitivnih procesa

- ***Pseudokod***
- ***Primer*** - ***Sračunavanje prosečne ocene*** za sve studente:

```
FOR EACH STUDENT DO
Učitaj rekord studenta;
SUMA := 0.;
BROJPRED := 0;
DO WHILE Postoje položeni predmeti
SUMA := SUMA + OCENA;
BROJPRED := BROJPRED + 1;
END WHILE;
PROSOCENA := SUMA/BROJPRED;
Štampaj ime studenta i prosečnu ocenu;
END FOR;
```

Specifikacija logike primitivnih procesa

- ***Pseudokod***
- ***Primer*** – ***Provera raspoloživosti kopije filma*** u video klubu:

Opis:	Provera da li u video klubu postoji kopija filma koja se može iznajmiti
Ulazni tokovi:	Upit (Film), Rezervacije, Iznajmljivanje
Izlazni tokovi:	Film nije raspoloživ, Film je raspoloživ

Specifikacija logike primitivnih procesa

- ***Pseudokod***
- ***Primer*** – ***Provera raspoloživosti kopije filma*** u video klubu:

Logika procesa:	**Izračunaj** broj kopija traženog filma u video klubu
	Ako je broj kopija veći od nule **tada**
	  Ako je broj kopija veći od (broj Iznajmljivanja + broj rezervacija)
	  Rezultat = „Raspoloživ“
	  Else
	  Rezultat = „Nije raspoloživ“
	Inače
	  Rezultat = „Ne postoji“

Specifikacija logike primitivnih procesa

- ***Pseudokod***
- ***Primer*** - Opis logike primitivnog procesa za IS studentske službe:

```
EVIDENTIRANJE KANDIDATA

Prihvati DIPLOMU_SRED_ŠKOLE;
IF VRSTA_ŠKOLE ne odgovara THEN formiraj NEPRIHVATLJIV_DOKUMENTI;
FOR EACH SVEDOČANSTVO DO
Prihvati SVEDOČANSTVO
* Provera da li je PROSEK u svedočanstvu dobro sračunat*
SUMA := 0.; BROJPRED := 0;
DO WHILE Postoje predmeti u SVEDOČANSTVO
SUMA := SUMA + OCENA;
BROJPRED := BROJPRED + 1;
END WHILE;
PROSOCENA := SUMA/BROJPRED;
IF PROSEK NOT EQUAL PROSOCENA THEN
formiraj NEPRIHVATLJIV_DOKUMENTI;
Ubaci podatke sa SVEDOČANSTVO u KANDIDATI_ZA_UPIS;
END FOR;
FOR EACH NAGRADE DO
IF IME_NAGRADE odgovarajuće THEN Ubaci podatke sa NAGRADE u
KANDIDATI_ZA_UPIS;
END FOR;
```