

# Projektiranje informacijskih sustava

SDLC faza analize - Modeliranje  
procesa (process modeling)

Ak. god. 2011/2012

# Model procesa

- Procesni model (*process model*) ili model procesa je formalni način predstavljanja kako poslovni sustav radi.
- Procesni model opisuje aktivnosti koje se izvode i kako podaci teku među njima.
- Proces se obavlja uvijek na jednak način tj. za određeni ulaz daje isti izlaz. Trajanje procesa je konačno i odredivo tj. poznat je početak procesa, završetak, i ponavljanje.

# Model procesa

- Uobičajena tehnika koja se koristi za stvaranje modela procesa je dijagram toka podataka (*Data Flow Diagrams*, DFD).
- Bez obzira što naziv dijagrama toka podataka može implicirati temeljenje na podacima u fokusu DFD tehnike su aktivnosti, procesi.
- Pored navedene metode koriste se i druge (npr. IDEF0 (*Integration Definition for Function Modeling*)).

# Model procesa

- U fazi analize se postavljaju logički modeli procesa koji opisuju procese bez da sugeriraju kako se oni realiziraju.
- U ovoj fazi je bitno da analitičar ispravno identificira procese bez opterećivanja implementacijskim detaljima.
- U fazi dizajna se logički modeli prebacuju u fizičke modele procesa koji pružaju informacije koje su neophodne da bi se izgradio konačni sustav.

# Model procesa

Logički model

- Kupac izabire proizvode koje će naručiti iz dućana.
- Kupac preko web sučelja izabire proizvode koje će naručiti preko automatski generirane narudžbe koja će se e-mailom poslati u dućan.

ILI

- Kupac preko tiskanog kataloga izabire proizvode koje će naručiti telefonom iz dućana.

ILI

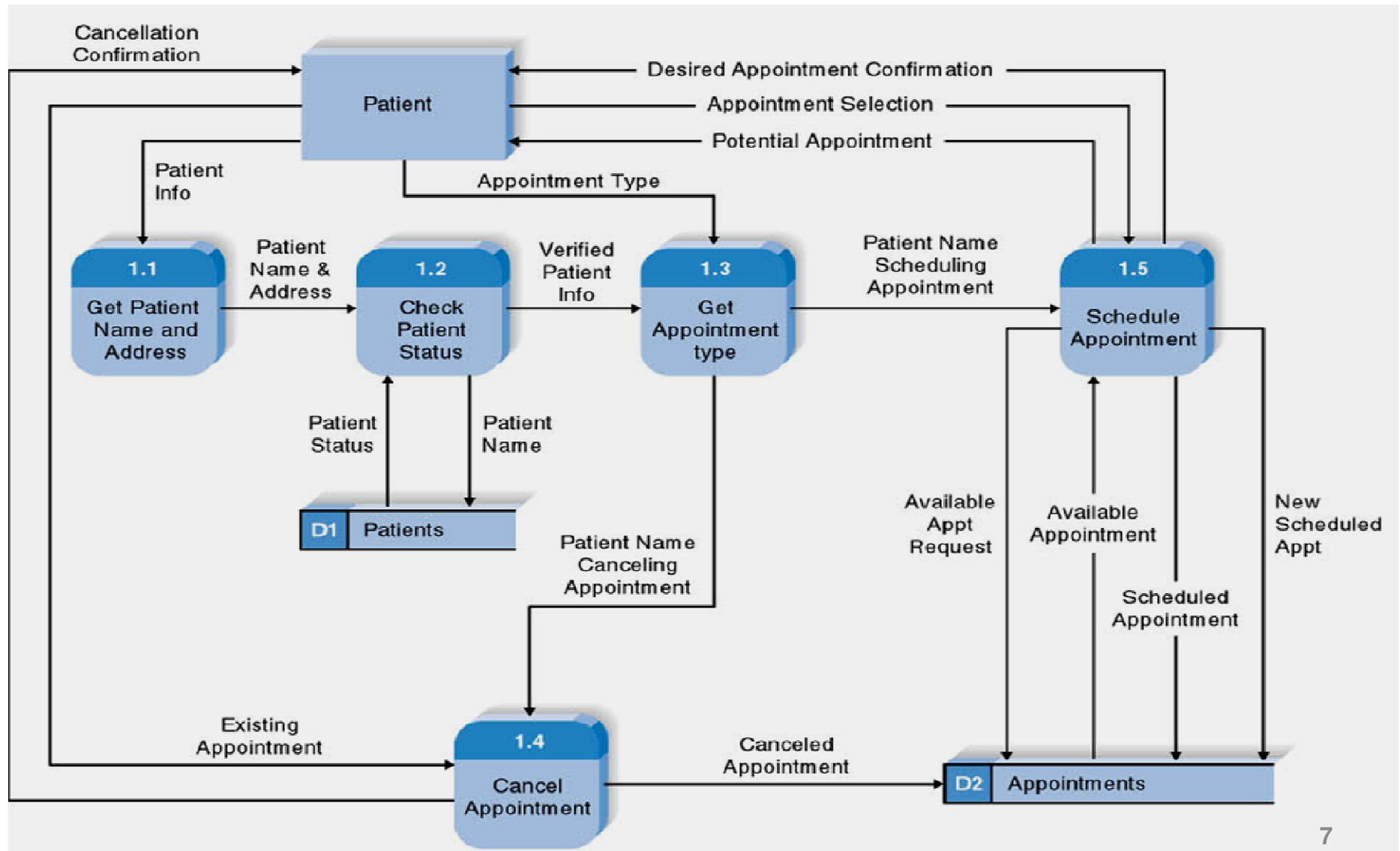
- Kupac dođe u dućan izabрати proizvode i usmeno kaže prodavatelju svoju narudžbu.

Fizički model

# Dijagram toka podataka (DFD)

- DFD omogućava grafički prikaz procesa u sustavu i tijeka podataka između procesa.
- DFD dijagram se oslanja na tzv. DFD jezik koji definira DFD simbole, sintaktička pravila, konvencije imenovanje.
- DFD ima četiri osnovna elementa koji su prikazani sa odgovarajućim grafičkim simbolom. Ti elementi su:
  1. proces
  2. tok podataka
  3. spremište podataka
  4. vanjski entitet

# Dijagram toka podataka (DFD)



# Dijagram toka podataka

- DFD je različit od dijagrama toka (*flowchart*) koji je namijenjen prikazivanju kontrole toka izvršavanja algoritma.
- DFD ne definira redoslijed izvršavanja procesa, niti da li se procesi izvršavaju paralelno ili sekvencijalno itd., dok dijagram toka ne definira ulazne i izlazne tokove podataka.
- Kod DFD-a se obično koriste dva stila simbola (Chris Gane i Trish Sarson, Tom DeMarco i Ed Yourdon), ali ne zajedno!






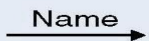




# Dijagram toka podataka

proces

tok  
podataka

spremište  
podataka

vanjski  
entitet

Data Flow Diagram Element	Typical Computer-Aided Software Engineering Fields	Gane and Sarson Symbol	DeMarco and Yourdan Symbol
<p>Every <i>process</i> has</p> <ul style="list-style-type: none"> <li>A number</li> <li>A name (verb phase)</li> <li>A description</li> <li>One or more output data flows</li> <li>Usually one or more input data flows</li> </ul>	<p>Label (name)</p> <p>Type (process)</p> <p>Description (what is it)</p> <p>Process number</p> <p>Process description (Structured English)</p> <p>Notes</p>		
<p>Every <i>data flow</i> has</p> <ul style="list-style-type: none"> <li>A name (a noun)</li> <li>A description</li> <li>One or more connections to a process</li> </ul>	<p>Label (name)</p> <p>Type (flow)</p> <p>Description</p> <p>Alias (another name)</p> <p>Composition (description of data elements)</p> <p>Notes</p>		
<p>Every <i>data store</i> has</p> <ul style="list-style-type: none"> <li>A number</li> <li>A name (a noun)</li> <li>A description</li> <li>One or more input data flows</li> <li>Usually one or more output data flows</li> </ul>	<p>Label (name)</p> <p>Type (store)</p> <p>Description</p> <p>Alias (another name)</p> <p>Composition (description of data elements)</p> <p>Notes</p>		
<p>Every <i>external entity</i> has</p> <ul style="list-style-type: none"> <li>A name (a noun)</li> <li>A description</li> </ul>	<p>Label (name)</p> <p>Type (entity)</p> <p>Description</p> <p>Alias (another name)</p> <p>Entity description</p> <p>Notes</p>		

# Dijagram toka podataka

- Proces
  - Aktivnost ili funkcija obavljena zbog specifičnog poslovnog razloga, bilo da je ručni ili kompjuteriziran.
  - Preporučeno imenovanje procesa je sa glagolom i imenicom (npr. “Naruči proizvod”, “Otvori kredit”, “Prijavi ispit”,...)
  - Svaki proces mora imati barem jedan ulazni tok i barem jedan izlazni tok podataka. Ako proces nešto radi generira nekakve izlazne podatke. Da bi proces nešto napravio treba imati nekakve ulazne podatke.

# Dijagram toka podataka

- Tok podataka
  - Jedan podatak (npr. ime proizvoda) ili logička kolekcija podataka (npr. podaci o proizvodu – šifra proizvoda, ime proizvoda, jedinična količina, ...).
  - Uvijek započinje ili završava unutar procesa (jedan kraj toka podataka mora biti povezan sa procesom), jer tokovi podataka služe za povezivanje procesa.
  - Tok podatka pokazuje koji ulazi idu u proces tj. koje izlaze proces kreira
  - Preporučeno imenovanje toka podatka je sa imenicom (npr. “Ime”, “Šifra”, “Proizvod”, “Ispit”,...)

# Dijagram toka podataka

- Spremište podataka
  - Skup podataka koji su na neki način pohranjeni.
  - Spremište podataka se označava imenicom, te mu se dodjeljuje identifikacijski broj i opis (npr. “B1 Proizvodi”,...).
  - Tokovi podataka koji izlaze iz spremišta pokazuju da je dohvaćen podatak iz spremišta.
  - Tokovi koji ulaze u spremište pokazuju da je dodan novi ili izmijenjen postojeći podatak.
  - Svako spremište podataka mora imati barem jedan ulazni tok i barem jedan izlazni tok podataka (ukoliko skladište ne održava ili koristi neki drugi informacijski sustav).
  - Ukoliko isti proces i dohvaća podatke iz skladišta i upisuje podatke u skladište tokovi podataka se trebaju prikazati odvojeno.

# Dijagram toka podataka

- Vanjski entitet
  - Osoba, organizacija ili sustav koji nije sastavni dio promatranog sustava, ali je u interakciji s njim.
  - Vanjski entiteti predstavljaju izvorište i/ili odredište podataka i predstavljaju granice sustava.
  - Svaki vanjski entitet treba imati oznaku i opis. Za označavanje vanjskog entiteta se koriste imenice (npr: “Student”, “Kupac”,...).
- Najvažnije je shvatiti da je vanjski entitet objekt iz “vanjskog” svijeta koji je povezan sa sustavom. Može i ne mora biti dio organizacije. Ljudi koji obavljaju proces NISU vanjski entiteti.

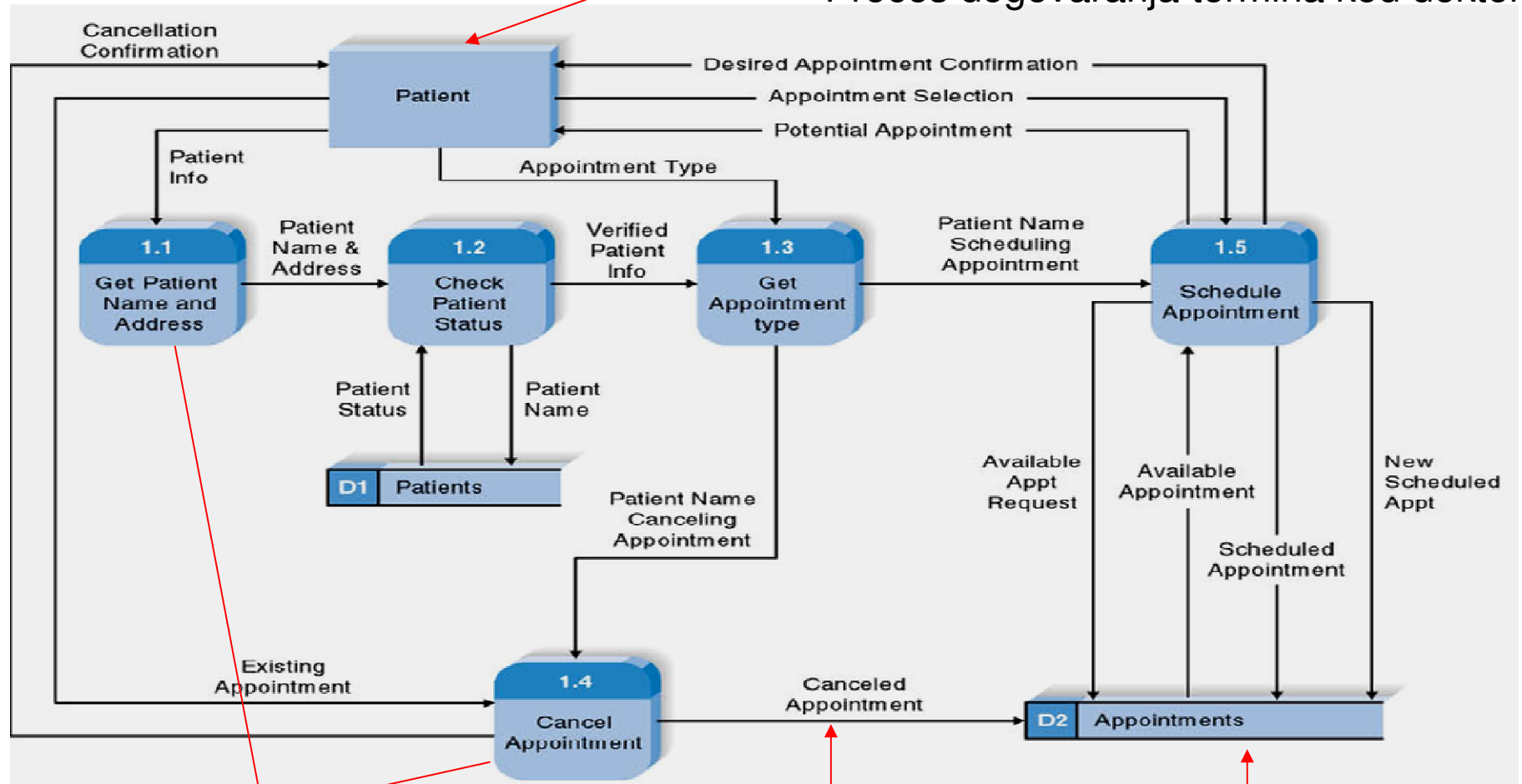
# CASE DFD

- Dataflow dijagrami se mogu razvijati korištenjem različitih CASE alata.
- Na slajdu 9 su pored simbola navedeni i podaci koje uobičajno CASE alati koriste prilikom kreiranja dijagrama.
- Alati mogu biti niske razine poput Visio alata koji omogućava samo prikaz dijagrama do alata više razine poput Visible Analyst Workbench alata koji omogućava i provjeru sintaktičke ispravnosti dijagrama ([http://www.visible.com/Solutions/edu\\_price.htm](http://www.visible.com/Solutions/edu_price.htm)).

# Dijagram toka podataka

vanjski entitet

Proces dogovaranja termina kod doktora



proces

tok podataka

spremište podataka

# Dijagram toka podataka

- Veliki poslovni procesi su presloženi da bi se prikazali jednim DFD-om, tako da je većina procesnih modela sastavljena od niza DFD-ova hijerarhijski organiziranih.
- Dijagram najviše razine bi trebao predstavljati kratak pregled cijelog sustava dok se na svakoj nižoj razini sustav prikazuje sve detaljnije.
- Pri hijarhijskom kreiranju DFD-a koriste se dva postupka:



# Dijagram toka podataka

1. Dekompozicija (*decomposition*) je postupak predstavljanja procesa kroz hijerarhiju DFD dijagrama pri čemu svaki *child* dijagram prikazuje dio dijagrama roditelja, ali sa više detalja.
2. Uravnoteženje ili balansiranje (*balancing*) osigurava da informacije predstavljene na jednoj razini hijerarhije DFD dijagrama odgovaraju informacijama predstavljenim na sljedećoj razini hijerarhije DFD dijagrama.

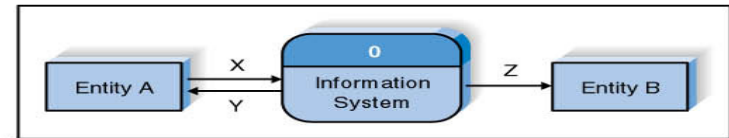
# Dijagram toka podataka

- Nije definirano na koji način se elementi DFD-a slažu.
- Uobičajeno je krenuti sa slaganjem procesa sa lijeva na desno i odozgo prema dole prema redosljedu kojim se procesi izvode, jer je to uobičajeni način pregledavanja dokumenta.
- Na dijagramu elemente treba organizirati tako da ima što manje križanja tokova podataka jer se tako povećava preglednost.
- Preporučeno je da dijagram ima između 3 i 7 (maksimalno) 9 procesa. Ako ima više od 9 procesa na toj razini predstavljanja sustava potrebno je neke procese grupirati zajedno, i prikazati ih na idućoj nižoj, detaljnijoj razini.

# Odnos između razina DFD-a

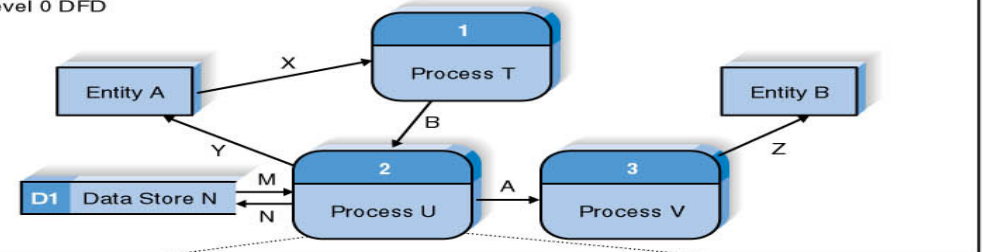
Kontekstni dijagram

Context Diagram



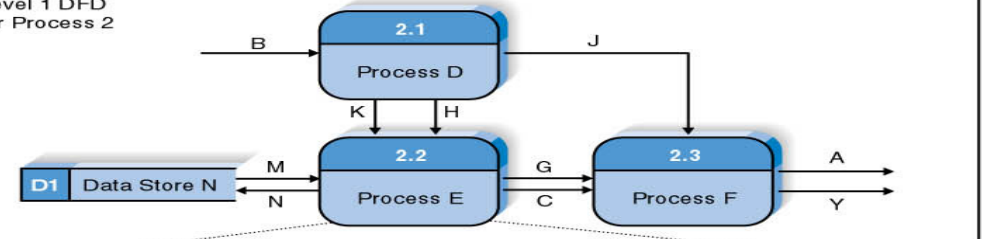
Dijagram 0 razine

Level 0 DFD



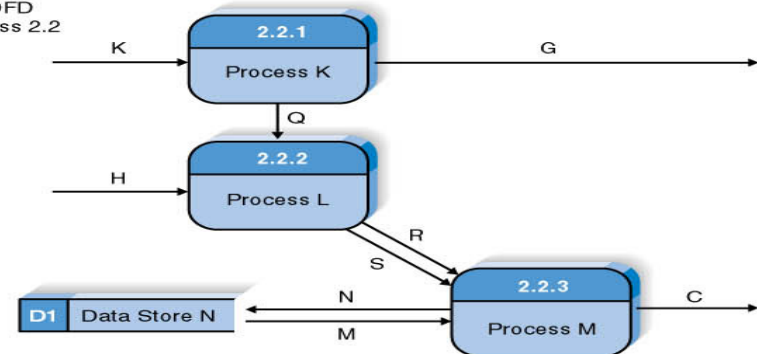
Dijagram 1 razine

Level 1 DFD for Process 2



Dijagram 2 razine

Level 2 DFD for Process 2.2



# Dijagram toka podataka

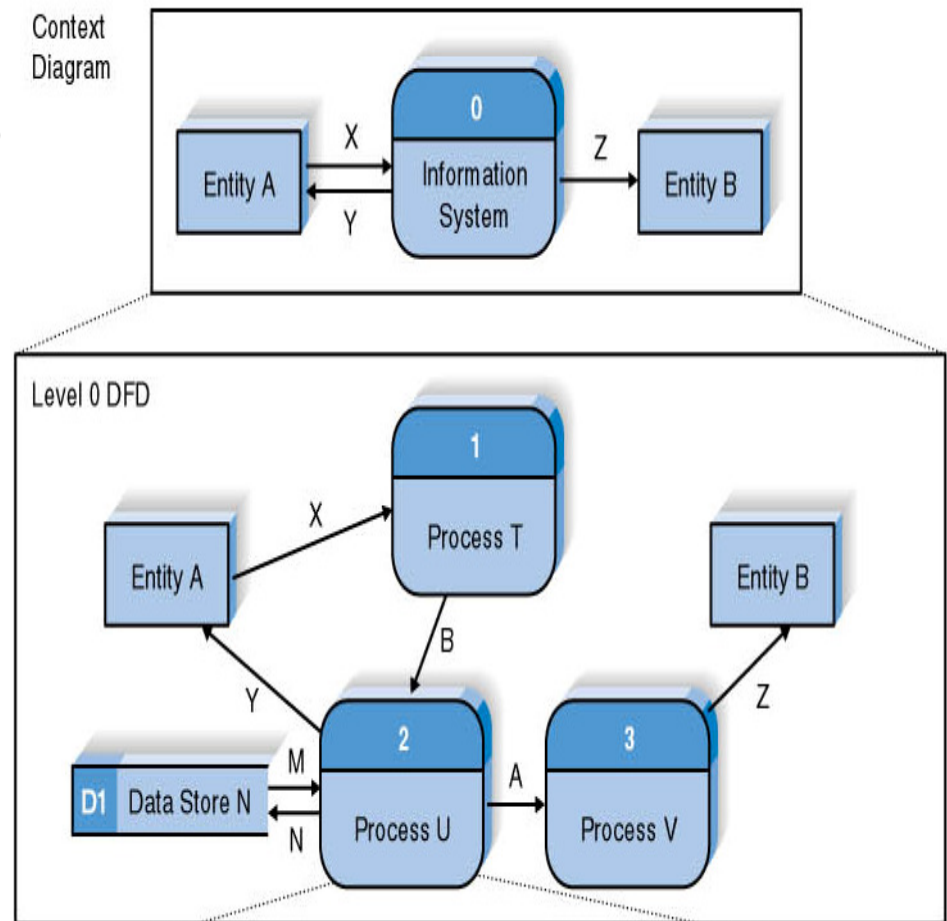
- Kontekсни dijagram (*context diagram*) je prvi DFD u svakom poslovnom procesu. Prikazuje kontekst u koji se poslovni proces uklapa.
- Sveukupni poslovni proces prikazuje kao samo jedan proces. Prikazuje sve vanjske entitete koji primaju informacije iz sustava ili donose informacije u sustav.
- Skladišta podataka se obično ne prikazuju u kontekstnom dijagramu (osim ako i sami nisu “vanjski” entiteti za promatrani sustav), a onda se i predstavljaju kao vanjski entiteti.

# Dijagram toka podataka

- Dijagram 0 razine (*level 0 diagram*) prikazuje sve glavne procese koje obuhvaća cjelokupni sustav.
- Prikazuje kako su glavni procesi povezani pomoću tokova podataka. Prikazuje vanjske entitete i glavne procese s kojima su oni u interakciji. Dodaje spremišta podataka.
- Svi procesni modeli imaju jedan i samo jedan dijagram 0 razine.

# Dijagram toka podataka

- Tokovi podataka (X,Y,Z) iz kontekstnog dijagram se također pojavljuju u dijagramu 0 razine.
- Dijagram 0 razine zamjenjuje jedini proces iz kontekstnog dijagrama (0) sa nova tri procesa (1,2,3). Dodaje se spremište podataka. Dodaju se dva nova toka podataka između procesa (A,B).
- Vanjski entiteti A i B su identični u kontekstnom dijagramu i u dijagramu 0 razine.



# Dijagram toka podataka

- Postupak dekompozicije se sastoji u razlaganju procesa sa promatrane razine hijerahije opisa sustava na manje sastavne dijelove. Ti sastavni dijelovi trebaju u potpunosti predstavljati proces više razine i uključivati sve funkcionalnosti razloženog procesa.
- Postupak balansiranja dijagrama osigurava da se sve informacije (tokovi podataka, skladišta) sa više razine hijerahije opisa sustava zadrže i na sljedećoj nižoj razini opisa sustava.

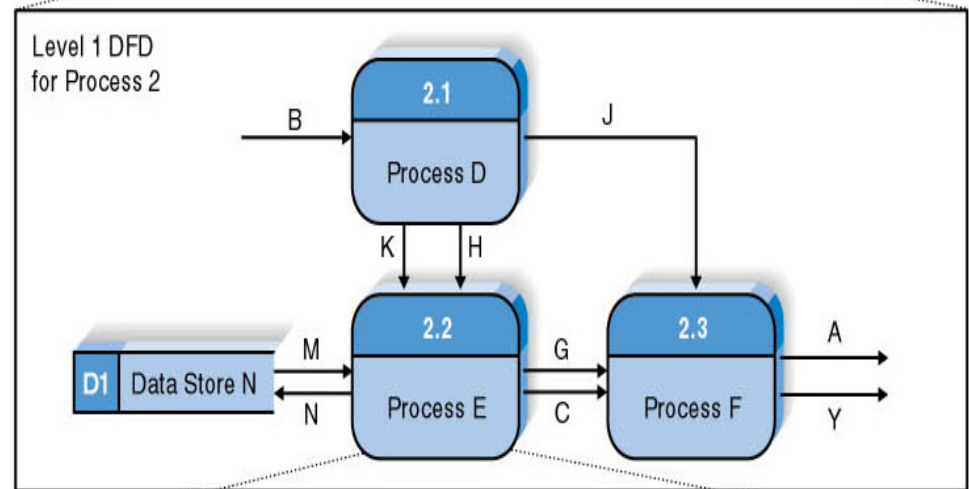
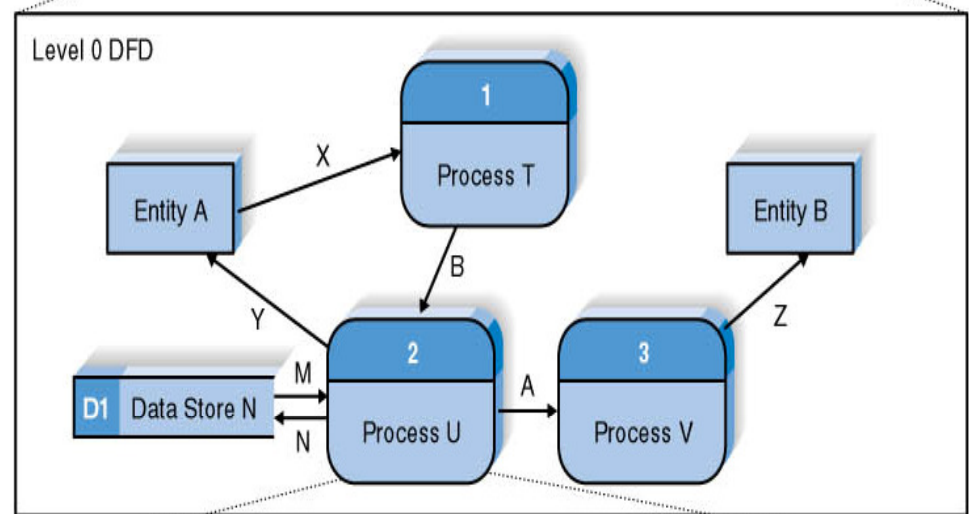
# Dijagram toka podataka

- Na sljedećoj razini se nalaze dijagrami 1 razine.
- Jedan dijagram 1 razine se radi za svaki glavni proces na dijagramu 0 razine. Prikazuje sve interne procese koje sačinjavaju jedan proces dijagrama 0 razine.
- Prikazuje kako se informacija kreće od i prema svakom od tih procesa.
- Ako se roditeljski (*parent*) proces rastavlja u tri procesa djecu (*child*), ova tri *child* procesa u potpunosti i kompletno izgrađuju *parent* proces.



# Dijagram toka podataka (DFD)

- Procesi 2.1 , 2.2 i 2.3 su “djeca” procesa 2 iz dijagrama 0 razine.
- Ti procesi u potpunosti izvršavaju sve funkcije procesa 2.
- Uobičajeno je prikazivati spremišta podataka na svim razinama DFD-a.

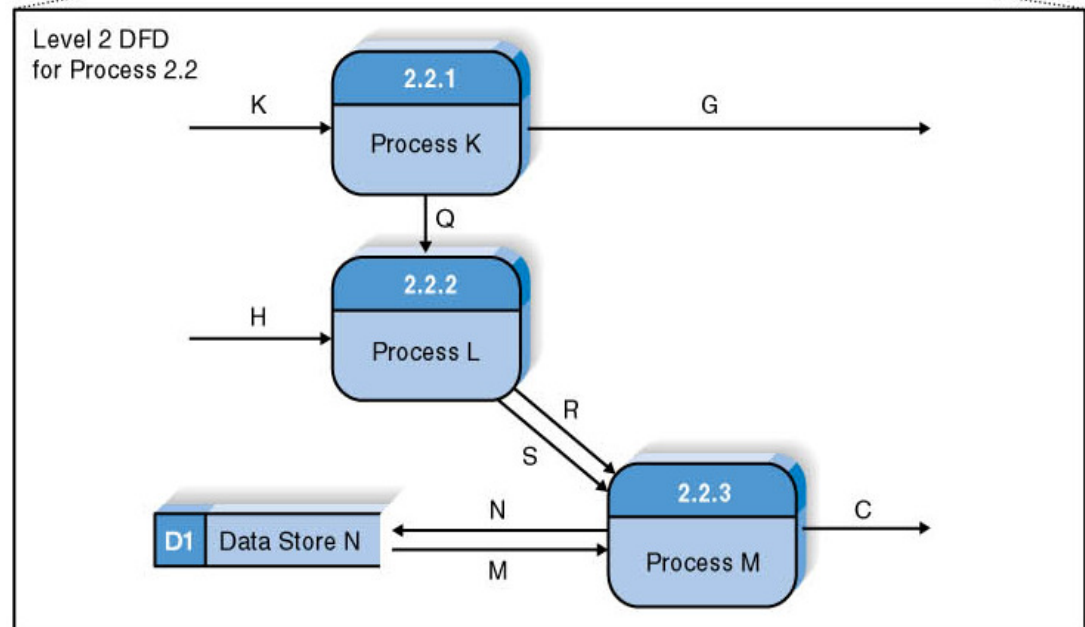
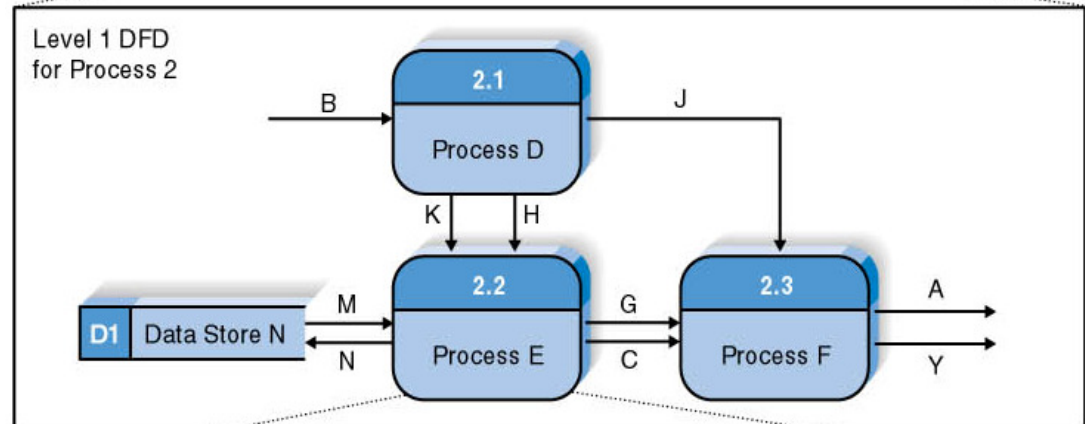


# Dijagram toka podataka

- Dijagram 2 razine prikazuje sve procese od kojih se sastoji jedan proces na dijagramu 1 razine.
- Prikazuje kako se informacija kreće od i prema svakom od ovih procesa. Dijagrami 2 razine nisu uvijek potrebni za sve procese 1 razine.
- Pravilno numeriranje (označavanje brojevima) svakog procesa pomaže korisniku razumjeti gdje se proces uklapa u cjelokupni sustav i na kojoj razini hijerarhijskog DFD-a se nalazi (2 proces iz dijagrama 0 razine, 2.1 proces iz dijagrama razine 1, 2.1.3 proces iz dijagrama razine 2,.....) .

# Dijagram toka podataka

- Proces 2.2 je rastavljen na procese 2.2.1, 2.2.2 i 2.2.3
- Konstantno treba voditi računa o uravnoteženju sa prethodnom razinom dijagrama



# Dijagram toka podataka (DFD)

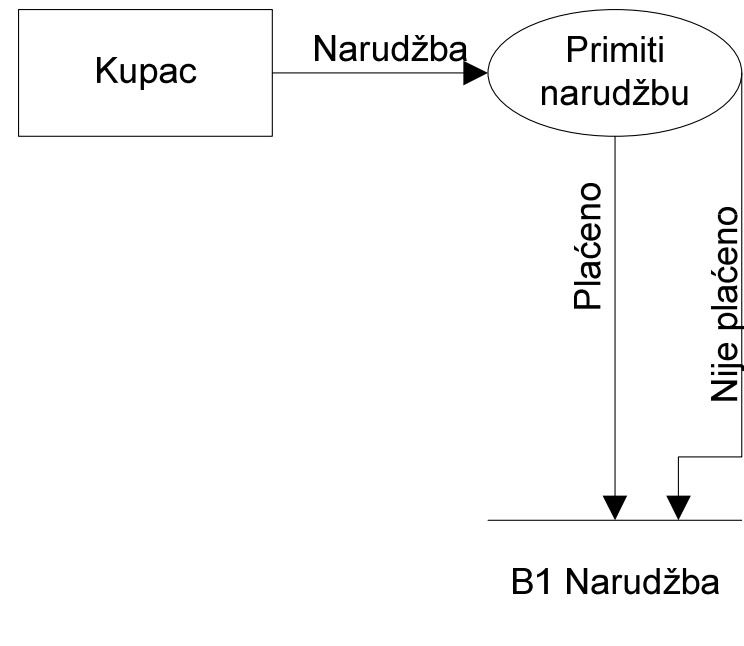
- Kad prestati rastavljati DFD-e?
- Nije potrebno sve procese dovesti do iste razine dekompozicije. Neki dijelovi sustava mogu biti složeniji od drugih pa zahtijevaju veću dekompoziciju.
- Jedno pravilo koje se može koristiti za dubinu dekompozicije je da se svaki proces može realizirati sa 25-50 linija koda. Ili da opis procesa ne bude veći od jedne stranice.

# Alternativni tokovi podataka

- Kada proces može proizvesti drugačije tokove podataka u drugačijim uvjetima kažemo da postoje alternativni tokovi podataka.
- Na dijagramu se uvijek zasebno prikazuju svi mogući tokovi podataka iz procesa i iz samog dijagrama nije vidljivo da li se tokovi podataka međusobno isključuju, da li se mogu dešavati istovremeno i sl.
- Naravno da je to bitno prilikom realizacije procesa pa je svaki proces iz DFD popraćen i tekstualnim opisom procesa (process description) koji detaljnije objašnjava proces pa i moguće višestruke tokove podataka.

# Alternativni tokovi podataka

- Npr. na ovom DFD dijagramu proces “Primiti narudžbu” ima dva izlazna toka “Plaćeno” i “Nije plaćeno”. Da li se ti tokovi mogu generirati paralelno, da li se međusobno isključuju,...?



# Opisi procesa

- Tekstualni opisi procesa pružaju više informacija o procesu.
- Npr. proces “Primi narudžbu” prima podatke od kupca od željenim proizvodima, te pohranjuje te podatke u bazu zajedno sa podatkom da li su proizvodi plaćeni ili nisu plaćeni.

# Opisi procesa

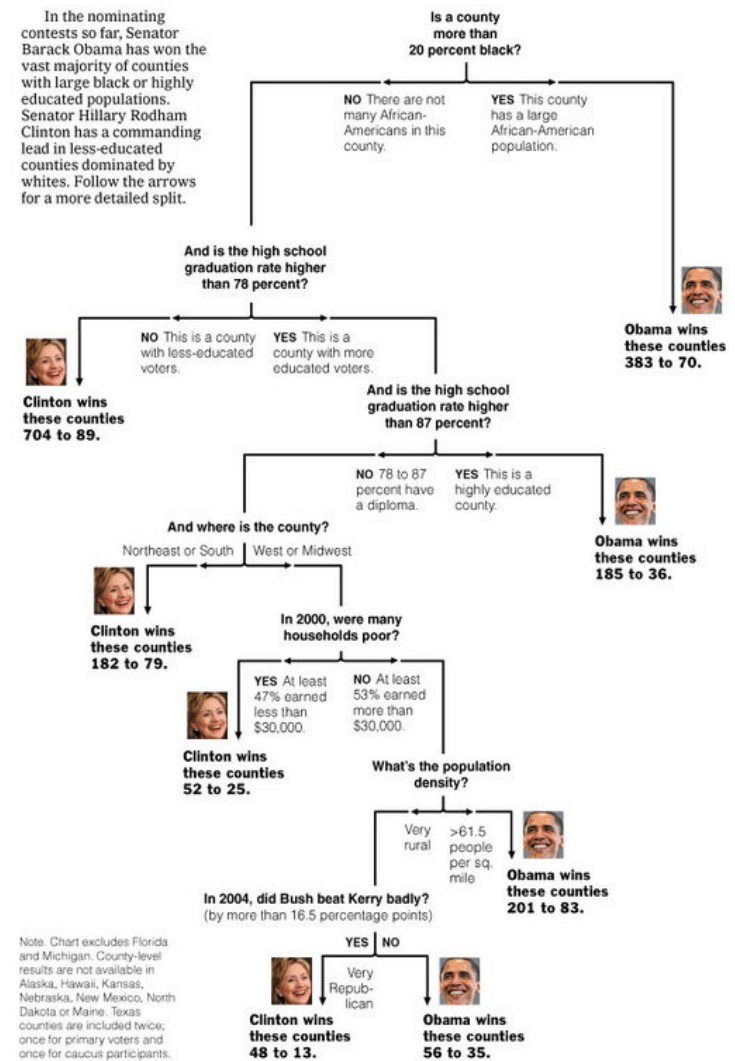
- Ako je logika na kojoj se temelji proces prilično kompleksna, može biti potrebno više detalja u obliku:
  1. Strukturiranog govornog jezika - kratke rečenice koje opisuju što proces radi.
  2. Stabla odlučivanja (*decision trees*) – prikazuju logiku odlučivanja u procesu kroz čvorove (pitanje) i grane (odgovore).
  3. Tablica odlučivanja (*decision tables*) – prikazuju logiku odlučivanja kroz pravila koja uvjete povezuju sa akcijom.



# Stabla odlučivanja

- Stabla odlučivanja su uobičajan način predstavljanja informacija (različitog tipa – numeričke, tekstualne,..) u obliku grafa.
- Podržavaju jednostavan mehanizam izvođenja zaključka.

Decision Tree: The Obama-Clinton Divide




Note. Chart excludes Florida and Michigan. County-level results are not available in Alaska, Hawaii, Kansas, Nebraska, New Mexico, North Dakota or Maine. Texas counties are included twice; once for primary voters and once for caucus participants.

Sources: Election results via The Associated Press; Census Bureau; Dave Leip's Atlas of U.S. Presidential Elections

AMANDA COX/  
THE NEW YORK TIMES

# Tablice odlučivanja

- Tablice odlučivanja slično kao i stabla odlučivanja omogućavaju modeliranje podataka u obliku if-then-else (ili switch-case) pravila.

Behaviour Status		Spends less than 100 dollars	Spends more than 100 dollars	Spends more than 150 dollars	Spends more than 200 dollars
1	 Customer with bronze status	Pays 5 dollars for item	Pays 4 dollars for item	Pays 3 dollars for item	Pays 2 dollars for item
2	 Customer with silver status	Pays 4 dollars for item	Pays 3 dollars for item	Pays 2 dollars for item	Pays 1 dollar for item
3	 Customer with gold status	Pays 3 dollars for item	Pays 2 dollars for item	Pays 1 dollar for item	Item is free

# Izrada DFD-a

- Izrada dijagrama toka podataka počinje sa slučajevima korištenja i definicijama zahtjeva.
- Identificirani slučajevi korištenja obično postaju procesi na 0 DFD razini, a imena slučajeva korištenja postaju imena procesa uz poštivanje navedenih pravila. Ulazi i izlazi slučaja korištenja postaju tokovi podataka s time da se obično se pojedinačni podaci spajaju u jedinstveni tok (ime, prezime + adresa = korisnički podaci).

# Izrada DFD-a

- Izrada DFD-a kreće sa kontekstnim dijagramom gdje se cijeli sustav prikazuje kao jedan proces.
- Kod kontekstnog dijagrama bitno je identificirati vanjske entitete.
- Sljedeći korak je izrada fragmenata DFD-a, po jednog za svaki slučaj korištenja.

# Izrada DFD-a

- DFD fragment je dio DFD-a koji će se kombinirati sa drugim fragmentima u jedinstveni dijagram pa na taj način iz slučajeva korištenja dobijemo DFD dijagram 0 razine.
- DFD-ovi spajaju definirane slučajeve korištenja u jedinstveni dijagram.

# Izrada DFD-a

- Nakon toga ide potrebna hijerarhijska dekompozicija do proizvoljne dubine. Potrebno je rastaviti procese 0 razine prema potrebi u dijagrame 1 razine, procese 1 razine u dijagrame 2 razine...
- Za ovo se koriste detalji slučajeva korištenja koji predstavljaju pojedinačne aktivnosti unutar slučaja korištenja.

# Izrada DFD-a

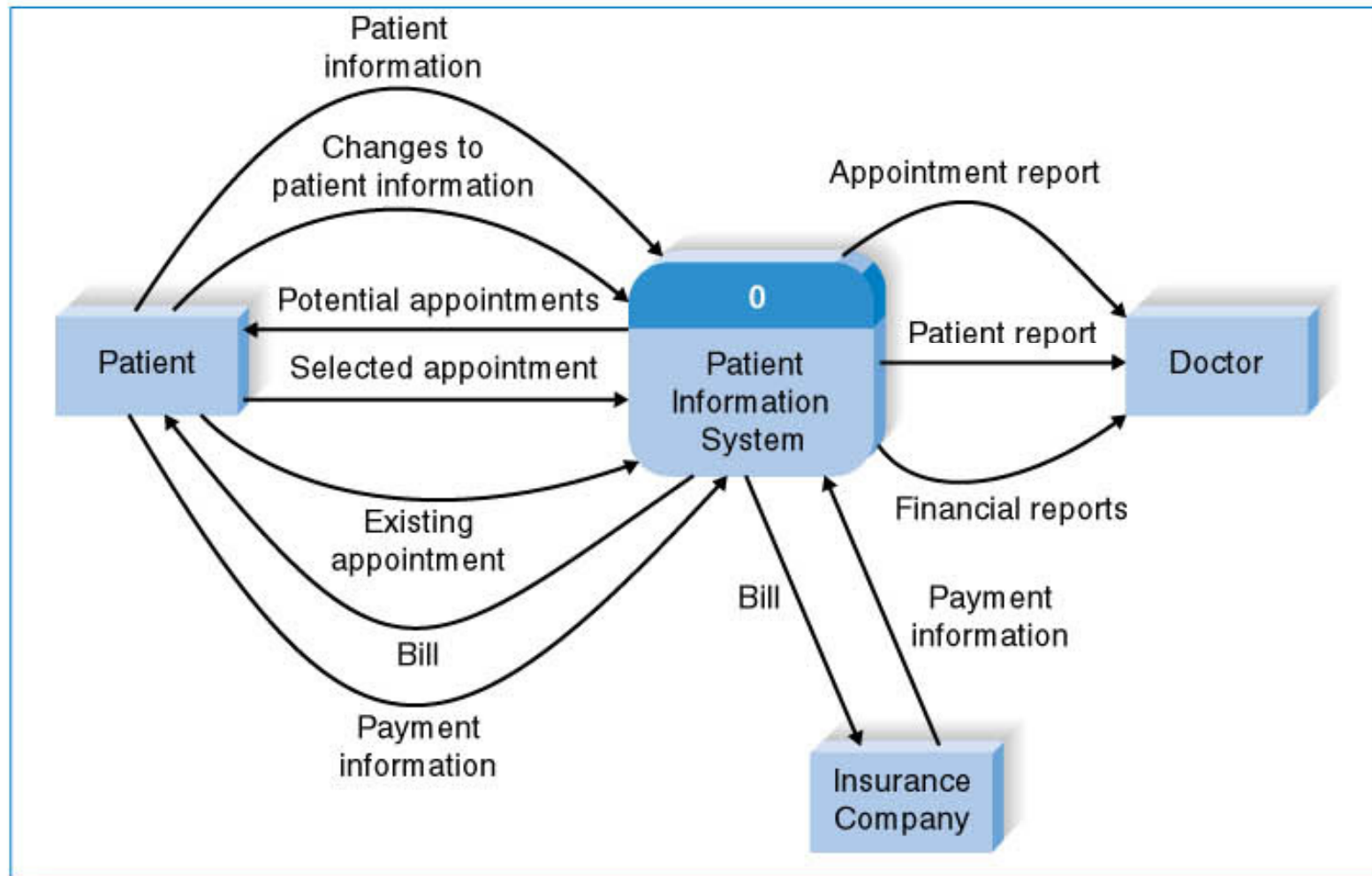
- Zadnji korak u izradi DFD-a je validacija dijagrama.
- Potrebno je napraviti provjeru sintaktičkih i semantičkih grešaka u dijagramima.
- Izrada DFD je iterativan proces, koji se često izvodi sa nekoliko ponavljanja navedenih koraka u izradi DFD-a.

# Izrada kontekstnog dijagrama

1. Nacrtati jedan proces koji predstavlja cijeli sustav (proces 0).
2. Pronaći sve ulaze i izlaze navedene na vrhu svih slučajeva korištenja koji dolaze iz ili idu prema vanjskim entitetima i nacrtati ih kao tok podataka u proces.
3. Nacrtati vanjske entitete kao izvor ili odredište toka podataka.



# Izrada kontekstnog dijagrama



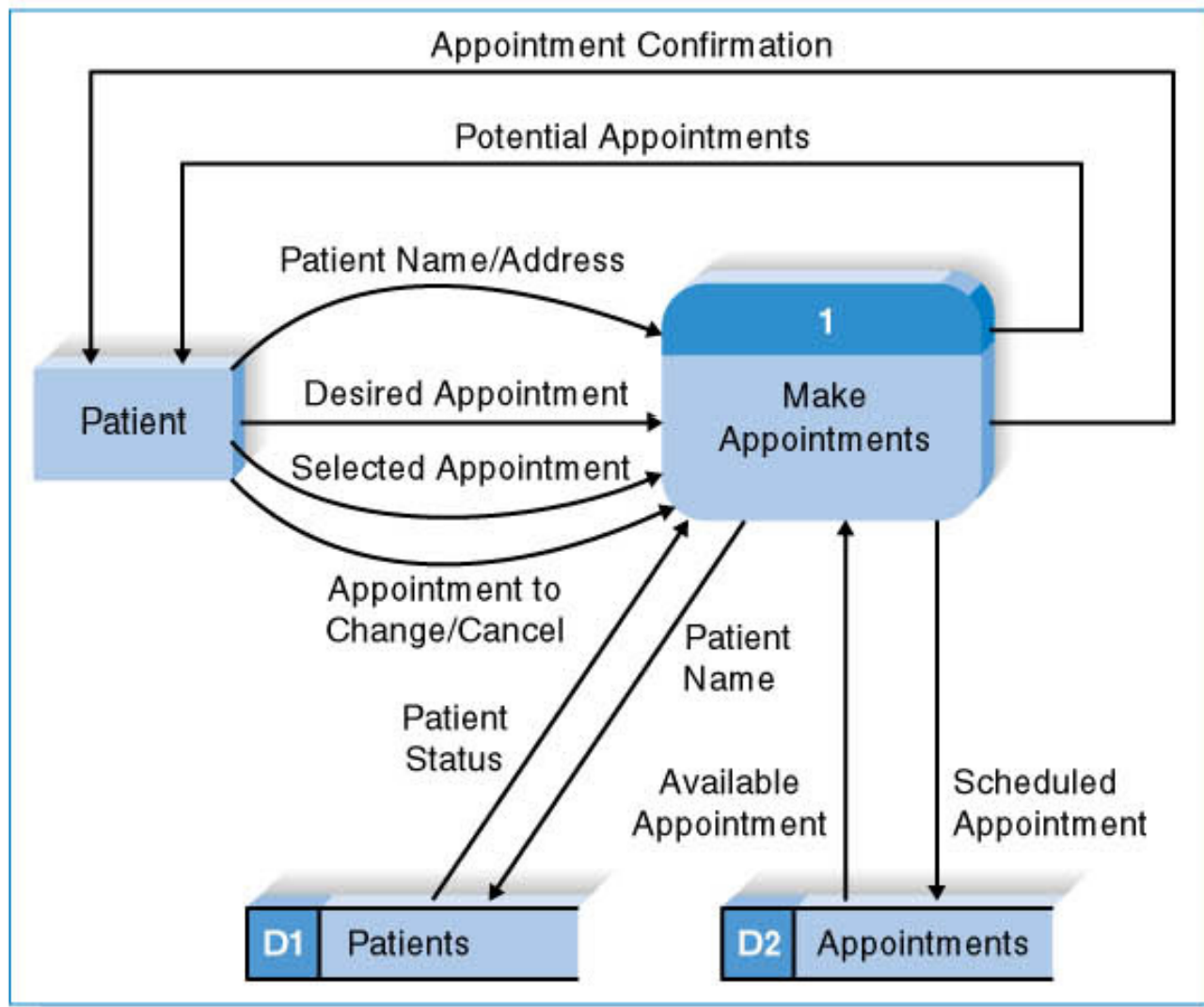
# Izrada dijagram 0 razine

1. Svaki slučaj korištenja pretvoriti u DFD fragment.
2. Numerirati proces istim brojem kojim je numeriran slučaj korištenja.
3. Imenovanje DFD procesa treba poštivati norme imenovanja pa se imena slučajeva korištenja trebaju prilagoditi.  
Promijeniti ime procesa u glagolski izraz.
4. Izrada DFD-a često uključuje i dodavanje dodatnih tokova podataka. Naime slučajevi korištenja se rade s obzirom na to kako sustav ostvaruje interakciju s korisnikom pa obično ne opisuju kako se dobavljaju podaci u sustavu.

# Izrada DFD-a

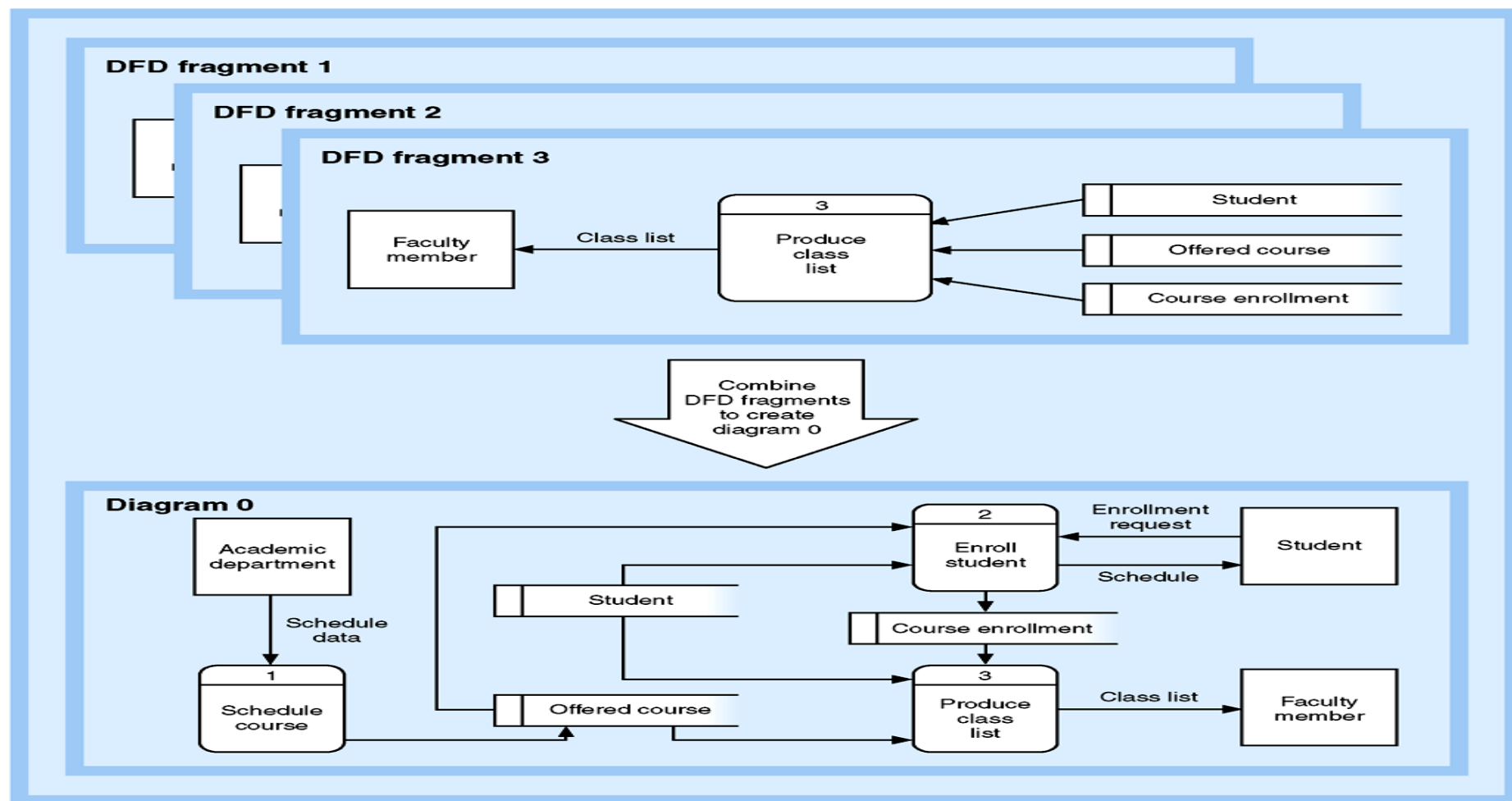
- Kako slučajevi korištenja ne opisuju kako se dobavljaju podaci u sustavu tako ne identificiraju spremišta podataka. Stoga je potrebno dodati tokove podataka da bi prikazali uporabu spremišta podataka kao izvorišta i odredište podataka.
- Ne postoje formalna pravila kako ćemo elemente DFD-a prikazati, ali je uobičajno da sheme (planovi) u pravilu stavljaju:
  - procese u središte
  - ulaze na lijevo
  - izlaze na desno
  - spremišta ispod procesa

# Dijagram toka podataka (DFD)



# Izrada dijagram 0 razine

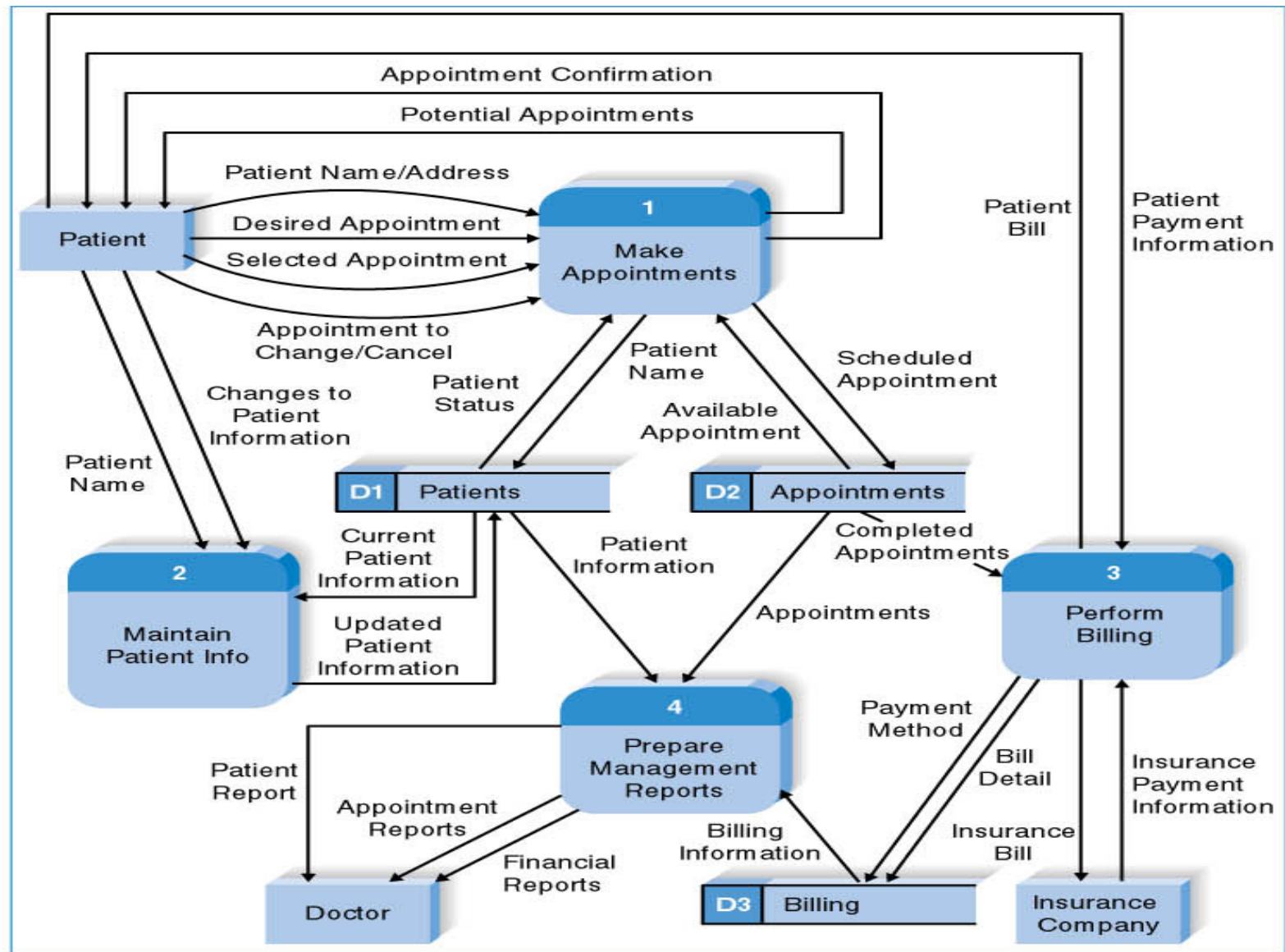
- Nakon izrade fragmenata skup DFD fragmenata spaja se u jedan dijagram.
- Fragmenti se obično spajaju tako da se kronološki poredani procesi smještaju od vrha prema dnu, s lijeva na desno (“prvi” proces ide skroz lijevo gore, ...).
- Poželjno je minimizirati križanje linija jer je dijagram tako pregledniji.
- Po potrebi postupak se iterativno ponavlja. Dijagrami toka podataka se često crtaju mnogo puta prije nego što se dovrše, čak iako ih crtaju vrlo iskusni analisti sustava.



**FIGURE 6-8**  
Combining DFD fragments to create the event-partitioned system model for the course registration system.

Kombiniranje fragmenata u jedinstveni DFD dijagram 0 razine

# Dijagram 0 razine



# Izrada dijagrama 1 razine

1. Svaki se slučaj korištenja pretvara u vlastiti DFD.
2. Uzeti korake popisane na slučajevima korištenja i prikazati svaki kao proces na DFD-u 1 razine.
3. Ulazi i izlazi navedeni na slučajevima korištenja postaju tokovi podataka na DFD-ima.
4. Uključiti izvore i odredišta tokova podataka prema procesima i spremištima unutar DFD-a.



# Validacija DFD-a

- Greške u DFD dijagramu mogu biti sintaktičke i semantičke.
- Sintaktičke greške se odnose na nepoštivanje sintaktičkih pravila DFD-a. Neki CASE alati mogu automatski provjeriti sintaktičke greške.
- Najčešće semantičke greške se odnose na narušavanje tzv. zakona o sačuvanju podataka (*law of conservation of data*):
  1. Data at rest stays at rest until moved by a process.
  2. Processes cannot consume or create data.

# Validacija DFD-a

- Za svaki dijagram provjeriti da li svaki proces ima:
  - Jedinstveno ime te broj i opis
  - Najmanje jedan ulazni tok podataka (procesi koji nemaju ulazne podatke nazivaju se *miracle error*)
  - Najmanje jedan izlazni tok podataka (procesi koji nemaju izlazne podatke nazivaju se *black hole error*)
  - Imena izlaznih tokova podataka drugačija od imena ulaznih tokova podataka.
  - Između 3 do 9 procesa u dijagramu.

# Validacija DFD-a

- Za svaki dijagram provjeriti da li svaki tok podataka ima:
  - Jedinstveno ime i opis
  - Spaja se na najmanje jedan proces
  - Pokazuje samo u jednom smjeru (ne dvosmjerne strelice)
  - Minimalan broj križanih linija

# Validacija DFD-a

- Za svaki dijagram provjeriti da li svaki spremište podataka ima:
  - Jedinstveno ime i opis
  - Najmanje jedan ulazni tok podataka
  - Najmanje jedan izlazni tok podataka
- Za svaki dijagram provjeriti da li svaki vanjski entitet ima:
  - Jedinstveno ime i opis
  - Najmanje jedan ulazni ili izlazni tok podataka

# Semantička validacija DFD-a

- Semantičke greške narušavaju točnost DFD-a u odnosu na stvarni poslovni proces. Obično su posljedica nerazumijevanja samog procesa od strane sistem analitičara.
- Jedan način semantička validacije se radi zajedno sa korisnikom kroz role-play proces slično kao i kod slučajeva korištenja.
- Korisnik prolazi kroz DFD i pokušava realizirati proces koristeći ulazne i izlazne tokove navedene u DFD-u.

# Semantička validacija DFD-a

- Najčešća semantička greška je upravo nedostatak ulaznih podataka u proces kako bi se kreirali izlazni podaci.
- Drugi način semantičke validacije je konzistentnosti DFD hijerarhijskih dijagrama kako bi osigurali dosljednu dekompoziciju.
- Također je potrebno imati konzistentnost i u terminologiji. Stoga je potrebno provjeriti pažljivo imena kako bi osigurali dosljednu uporabu izraza (npr. isti podaci se mogu na jednom DFD-u zvati “Podaci kupca”, a na drugom “Podaci klijenta”).

# Validacija DFD-a

- Za skupinu dijagrama provjeriti:
  - Kontekstni dijagram: Svaki grupa DFD-a mora imati samo jedan kontekstni dijagram.
  - Rastavljanje: Svaki proces je potpuno i u cijelosti opisan sa procesima na njegovim dijete DFD-ima
  - Uravnoteženost: Svaki tok podataka, spremište podataka i vanjski entitet na višim razinama DFD-a prikazan je na DFD-ima niže razine koji je sastavljaju. Ni jedan tok podataka i spremište se ne pojavljuju na DFD-ima niže razine ako se ne pojavljuju na njihovim roditeljskim DFD-ima.

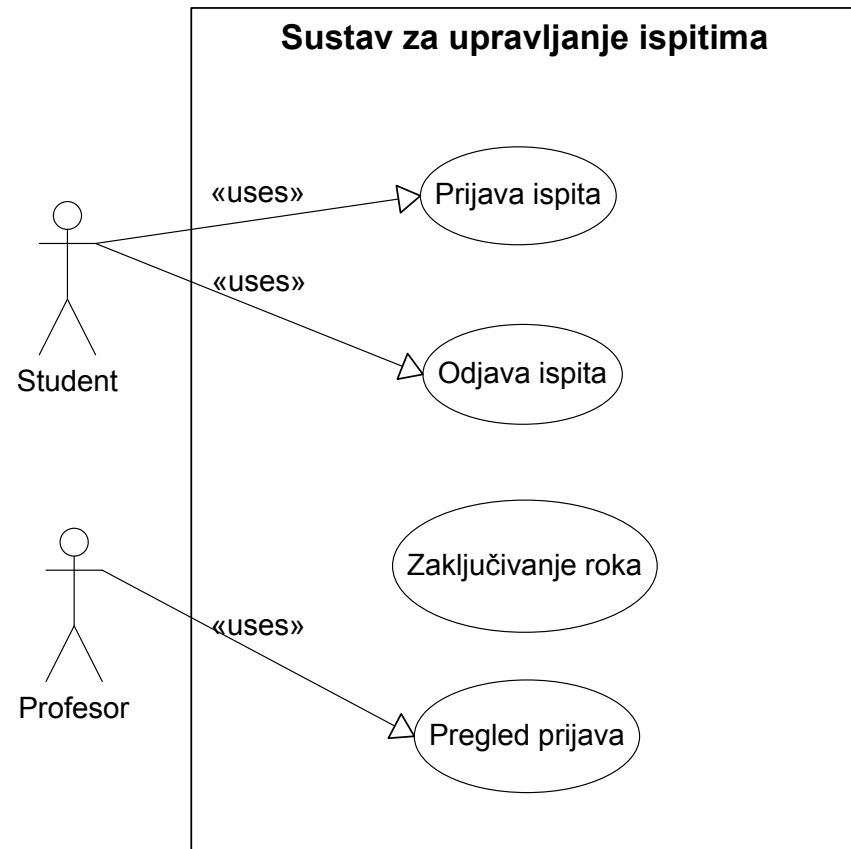
# PRIMJER – informacijski sustav za upravljanje ispitima

- Sustav za upravljanje ispitima treba omogućiti sve radnje vezane uz ispite, tj. omogućiti studentima prijavu i odjavu ispita, omogućiti profesorima pristup podacima o prijavljenim ispitima te omogućiti obradu podataka nakon ispita tj. zaključivanje ispitnog roka.



# PRIMJER – informacijski sustav za upravljanje ispitima

- Slučajevi korištenja prikazani u UML use-case dijagramu
- Identificirana su četiri slučaja korištenja – prijava ispita, odjava ispita, zaključivanje roka i pregled prijava



# PRIMJER – informacijski sustav za upravljanje ispitima

Slučaj korištenja:	Prijava ispita	ID:1	Važnost: Visoka
Primarni sudionik:	Student		
Kratki opis:	Ovaj slučaj korištenja opisuje kako student prijavljuje ispit		
Okidač:	Student prijavljuje ispit		
Tip okidača:	Vanjski		
Ulazi:	Izvorište:		
Podaci o studentu	Student		
Podaci o ispitu	Sustav		
Izlazi:	Odredište:		
Prijava	Sustav		

# PRIMJER – informacijski sustav za upravljanje ispitima

Slučaj korištenja:	Odjava ispita	ID:2	Važnost: Visoka
Primarni sudionik:	Student		
Kratki opis:	Ovaj slučaj korištenja opisuje kako student odjavljuje ispit		
Okidač:	Student odjavljuje ispit		
Tip okidača:	Vanjski		
Ulazi:	Izvorište:		
Podaci o studentu	Student		
Podaci o ispitu	Sustav		
Izlazi:	Odredište:		
Odjava	Sustav		

# PRIMJER – informacijski sustav za upravljanje ispitima

Slučaj korištenja: Zaključivanje ispita      ID:3      Važnost: Visoka  
Primarni sudionik: Sustav (tj. referada)  
Kratki opis:      Ovaj slučaj korištenja opisuje zaključivanje ispitnog roka  
Okidač:      Referada zaključuje ispit  
Tip okidača:      Vremenski  
Ulazi:      Izvorište:  
Podaci o studentima koji      Profesor  
su položili ispit  
Izlazi:      Odredište:  
Zaključena lista ispita      Sustav

# PRIMJER – informacijski sustav za upravljanje ispitima

Slučaj korištenja: Pregled prijava ID:4 Važnost: Visoka

Primarni sudionik: Profesor

Kratki opis: Ovaj slučaj korištenja opisuje kako se dobiva pregled prijavljenih studenata

Okidač: Profesor pregledava popis prijava za ispit

Tip okidača: Vanjski

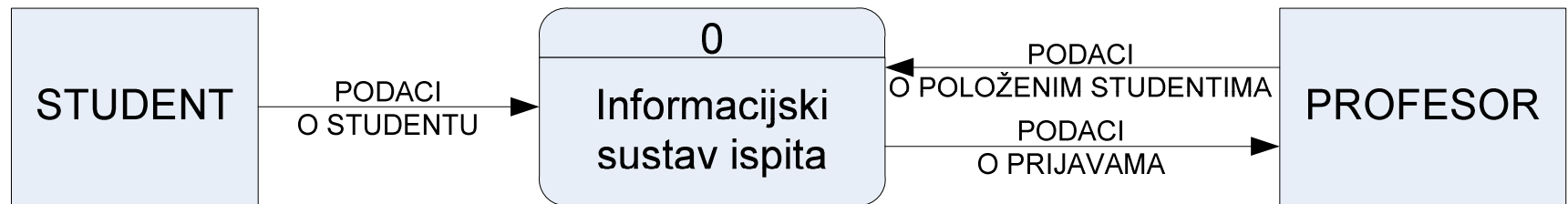
Ulazi: Izvorište:

Podaci o prijavljenim studentima Sustav

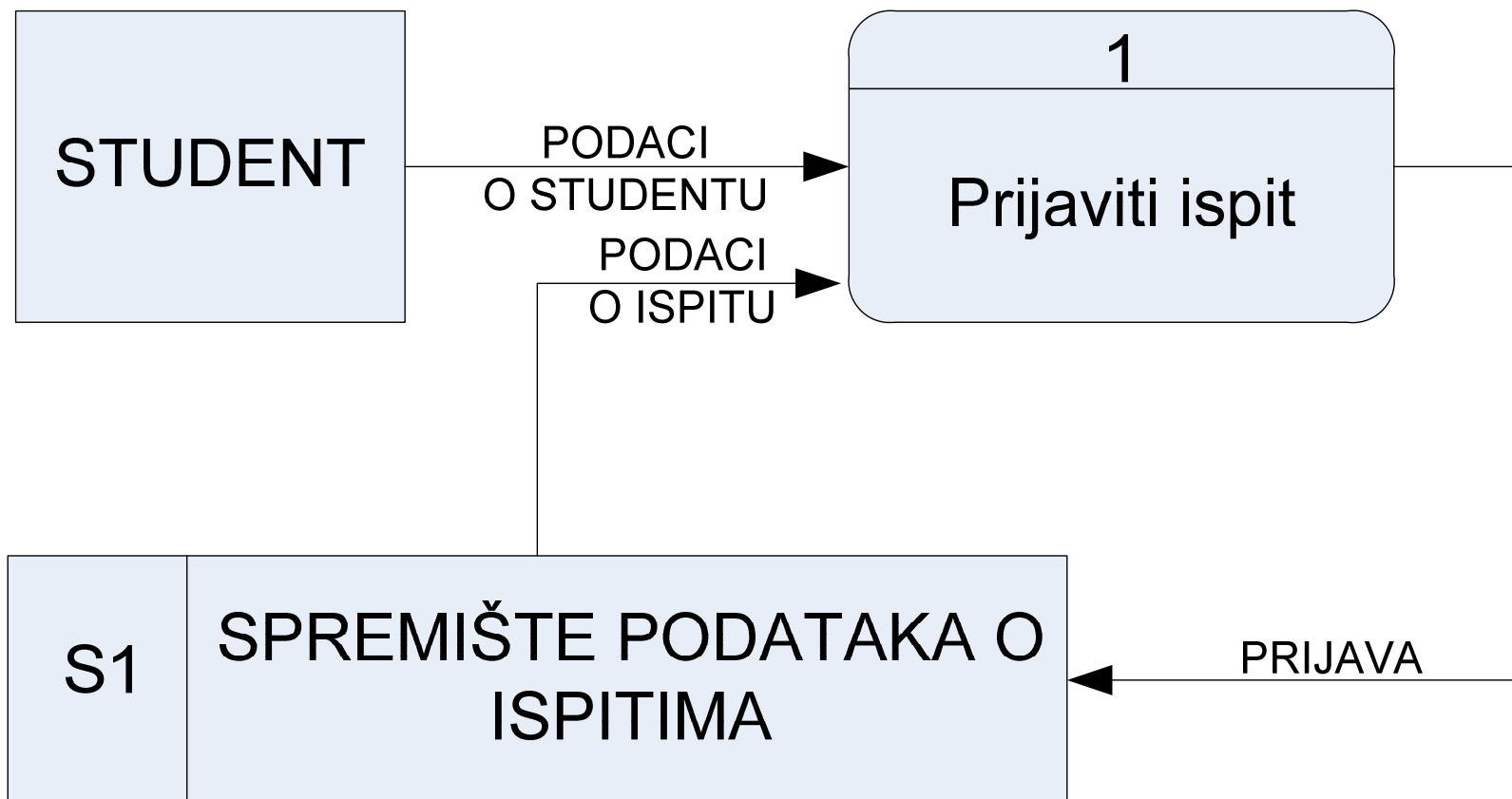
Izlazi: Odredište:

Podaci o studentima koji su prijavili ispit Profesor

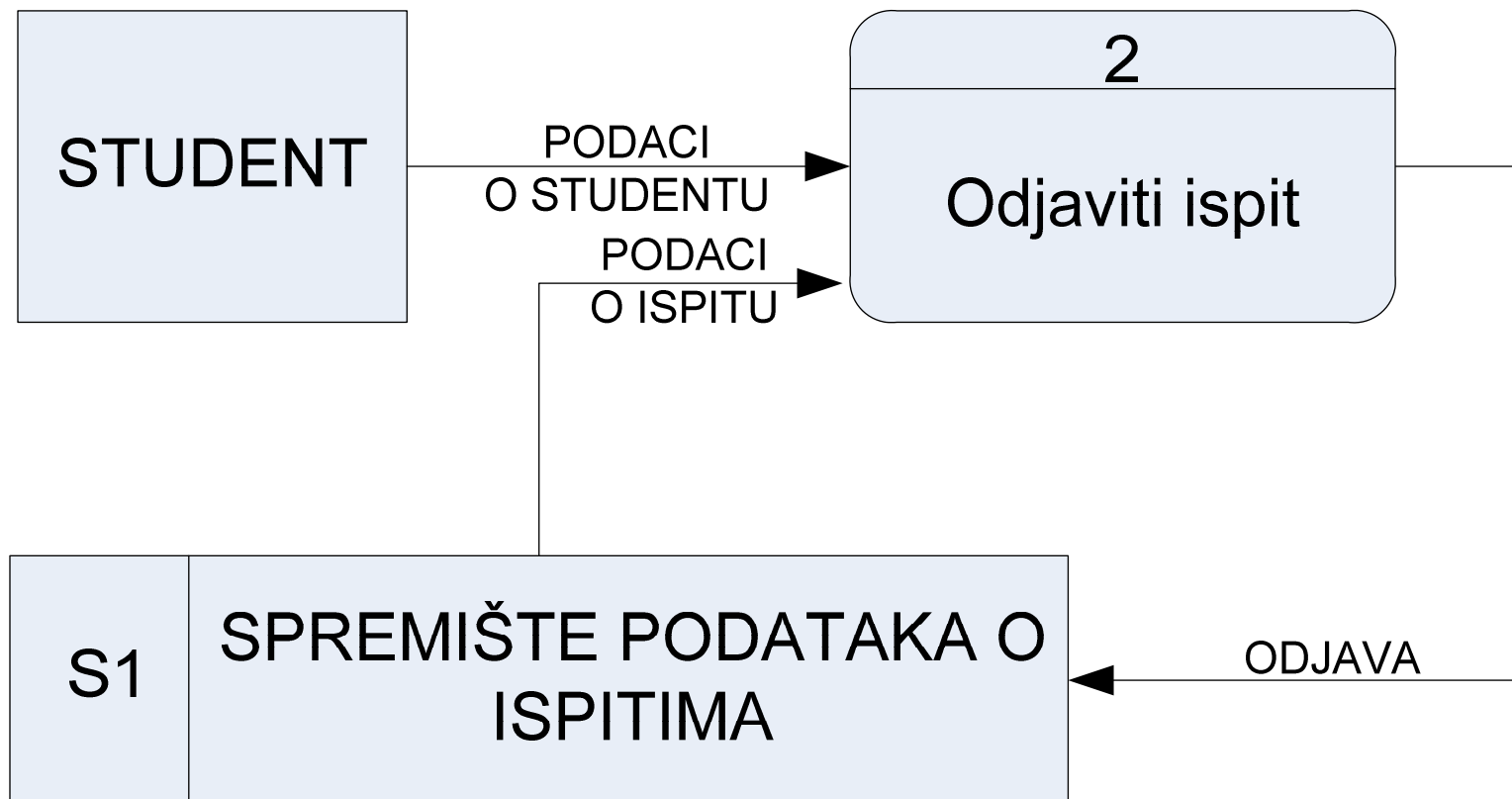
# PRIMJER – informacijski sustav za upravljanje ispitima



# PRIMJER – informacijski sustav za upravljanje ispitima

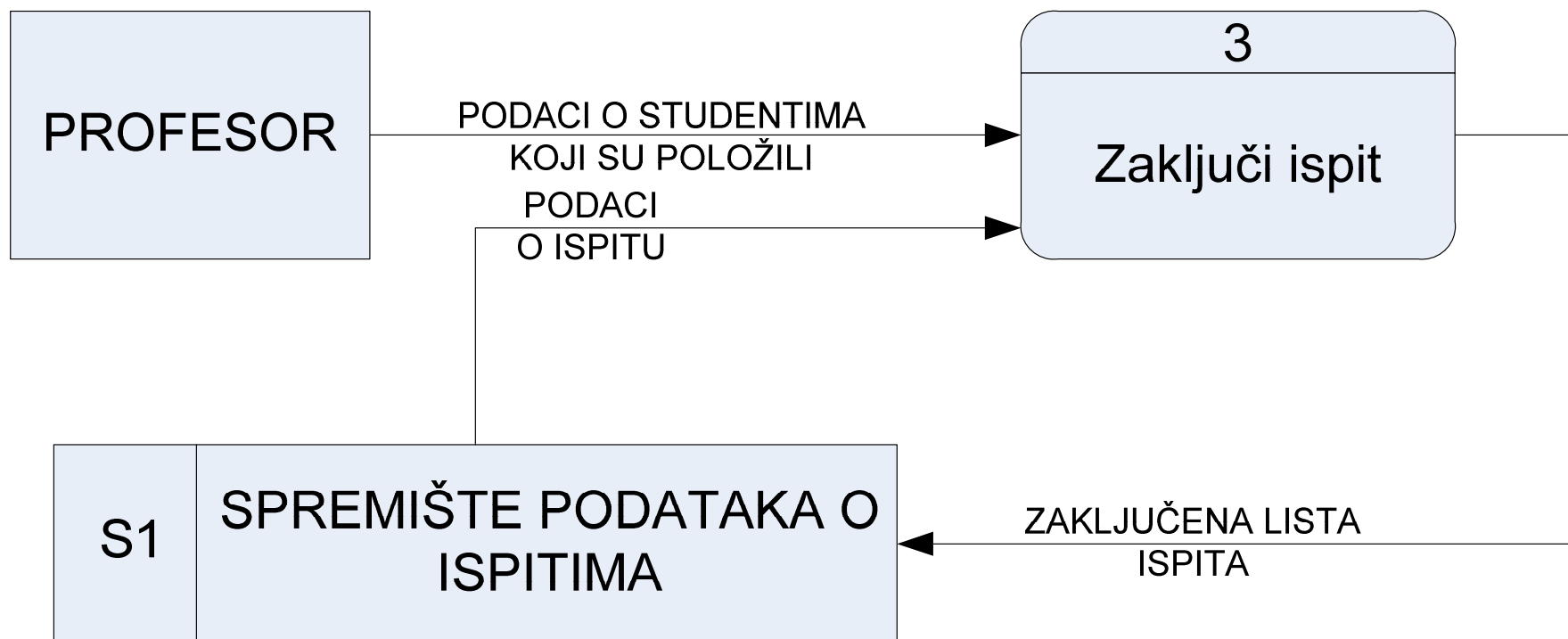


# PRIMJER – informacijski sustav za upravljanje ispitima

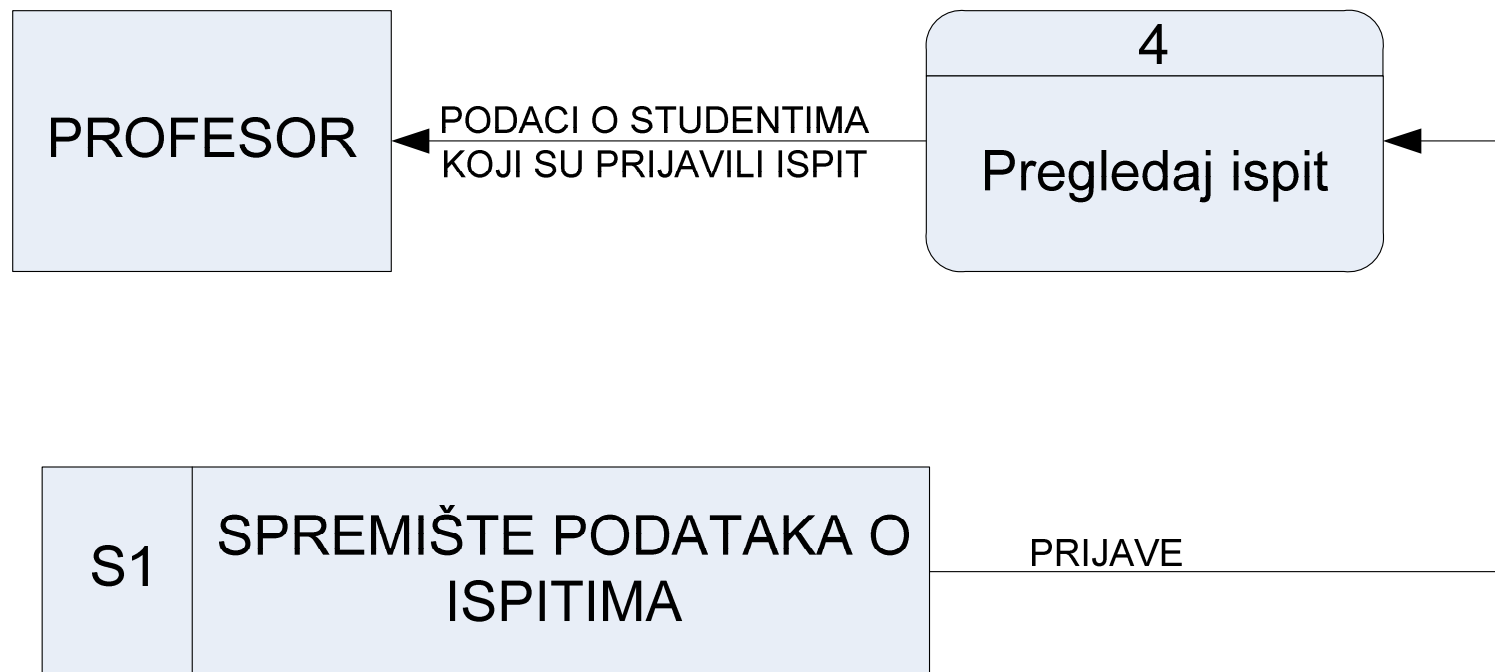




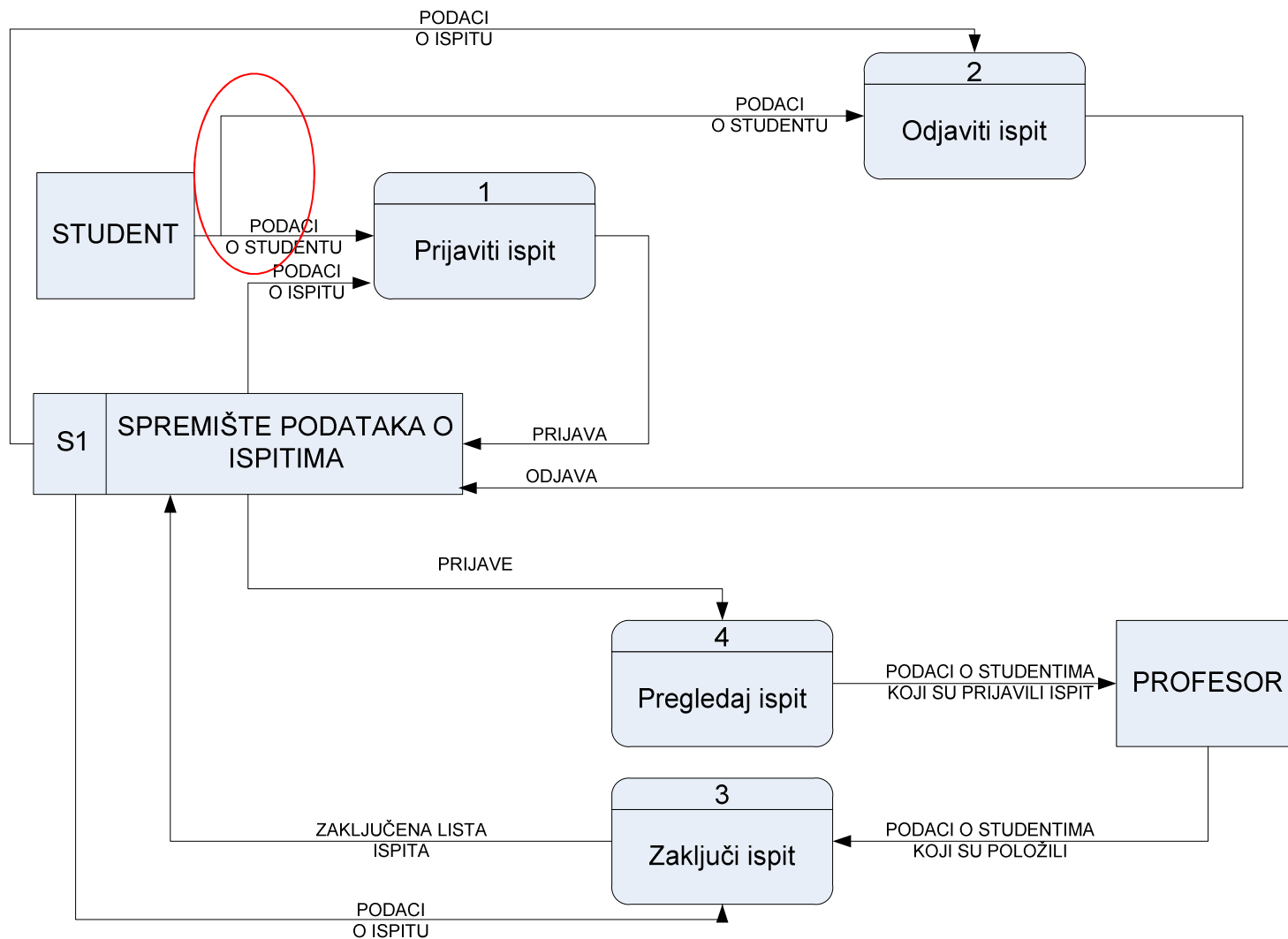
# PRIMJER – informacijski sustav za upravljanje ispitima



# PRIMJER – informacijski sustav za upravljanje ispitima



# PRIMJER – informacijski sustav za upravljanje ispitima



# PRIMJER – informacijski sustav za upravljanje ispitima

