

# Modelovanje poslovnih procesa

- ❏ Sistemska analiza
- ❏ Pojam poslovnih procesa
- ❏ Modelovanje poslovnih procesa
- ❏ Funkcionalno modelovanje
- ❏ Objektno-orijentisano modelovanje
- ❏ Servisno-orijentisano modelovanje



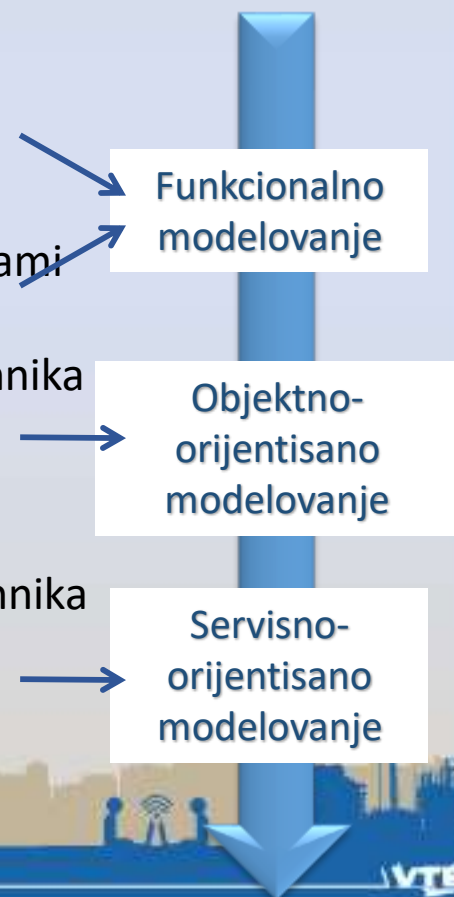
# Sistemska analiza

- Sistemska analiza je **najkritičnija faza jednog projekta**
- Tokom sistemske analize treba da se shvate:
  1. **problemi** konkretnog poslovnog sistema
  2. definišu **ciljevi** za njegovo poboljšanje i
  3. definišu detaljni poslovni **zahtevi** koje mora da ispuni bilo koje tehničko rešenje
- **Sistemska analiza** je metodološki **postupak dekompozicije** nekog sistema na podsisteme (komponente) sa ciljem da se prouči njihov međusobni uticaj i rad

# Strategije sistemske analize

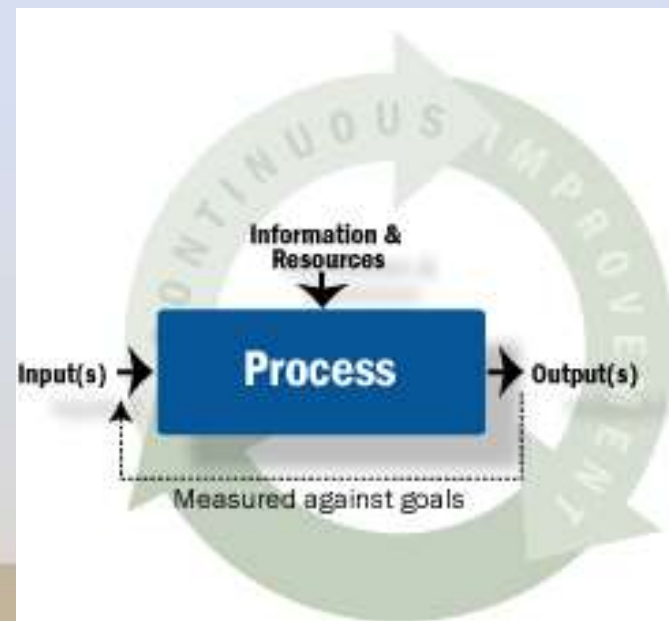
- S obzirom da postoji mnogo različitih pristupa za rešavanje problema, tako postoje i mnoge strategije i tehnike sistemske analize, neke od njih su:

- **Strukturalna analiza** (*Structured Analysis*) - najstarija tehnika orijentisana na procese
- **Informacioni inženjering i modeliranje podataka** - tehnika usredsređena na podatke. Modeli podataka se nazivaju dijagrami objekti-veze (*Entity Relationship Diagrams*)
- **Objektno-orijentisana analiza** (*Object-Oriented Analysis*) - tehnika koja eliminiše veštački razdvojene podatke od procesa. UML (*Unified Modeling Language*) obezbeđuje grafičku sintaksu za objektne modele
- **Servisno-orijentisana analiza** (*Service-Oriented Analysis*) – tehnika usredsređena na veb servise i rad u distribuiranom Internet okruženju (*Cloud Computing*)



# Pojam poslovnih procesa

- Celokupno poslovanje nekog preduzeća se odvija kroz poslovne procese
  - **Poslovni proces** čini kombinacija **poslovnih operativnih procedura**, poslovnih **pravila**, poslovnih **podataka** i **tehnologije** koja ih podržava
  - Mnogi poslovni procesi nisu dokumentovani, nisu optimizovani, ne efikasni su, skloni su greškama, nisu razumljivi ...
- 
- **Poslovni proces** je skup povezanih aktivnosti koje stvaraju neku vrednost preko transformacije ulaza u izlaze ispunjavajući pritom jedan poslovni cilj
  - Ulazi i izlazi mogu biti proizvodi i/ili informacije
  - Aktivnosti (tj. transformacije ulaza u izlaze) se obavljaju od strane ljudi i mašina



# Vrste poslovnih procesa

- Postoje različite vrste poslovnih procesa

- **Osnovna delatnost - operativni procesi** koji čine jezgro poslovanja

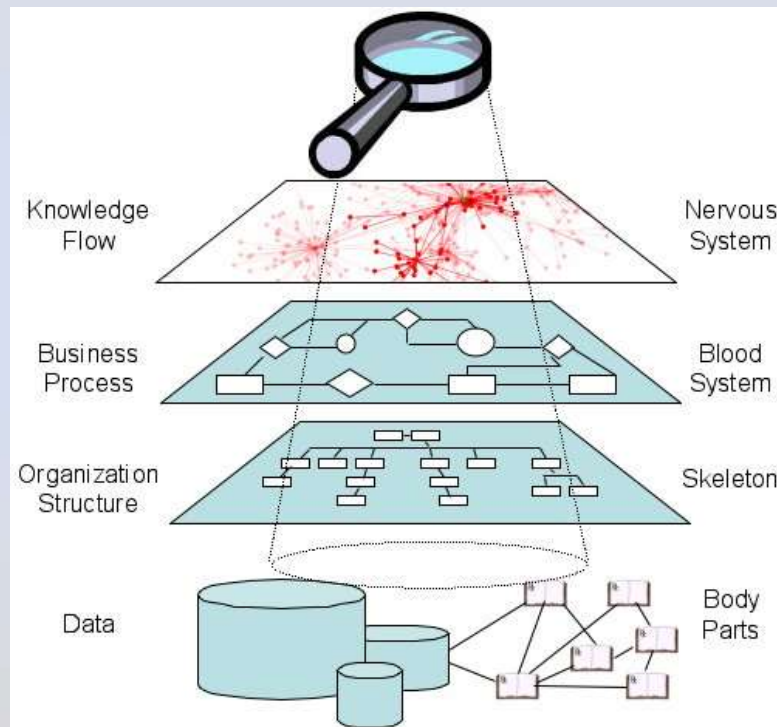
- proizvodnja, prodaja, nabavka, marketing ...

- **Procesi podrške**

- održavanje mašina, obuka radnika, tehnička podrška ...

- **Upravljački procesi**

- planiranje, odlučivanje, kontrola ...



# Kako nastaju poslovni procesi?



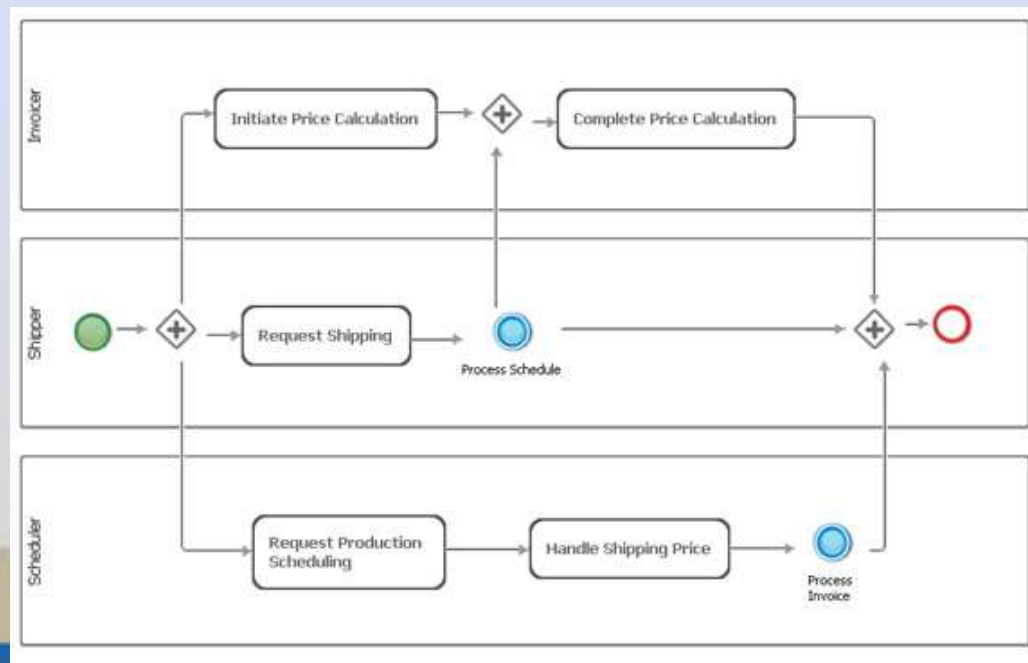
- Poslovni procesi su nastali kao **rezultat projektovanja organizacije**
  - Organizaciona struktura
  - Sistematizacija radnih mesta ...
- **Reinženjering** poslovnih procesa
  - Poboljšanje i pojednostavljenje postojećih poslovnih procesa kako bi poslovanje bilo efikasnije
- **Upravljanje promenama**
  - Uvođenje poboljšanih ili novih poslovnih procesa kako bi se izašlo u susret zahtevima korisnika i novonastalim promenama u okruženju
- **Upravljanje sistemom kvaliteta**
  - Standardi kvaliteta propisuju poslovne procese koji reflektuju najbolju praksu



# Modelovanje poslovnih procesa

## *Business Process Modeling, BPM*

- **Modelovanje poslovnih procesa** je aktivnost u kojoj se predstavljaju (specificiraju) poslovni procesi nekog preduzeća na način da se mogu analizirati, poboljšavati i automatizovati
  - Odnose se na postojeće ili buduće (poboljšane) poslovne procese
- BPM obavljaju **biznis analitičari** i **menadžeri** koji poboljšavaju efikasnost i kvalitet procesa
- Modeli tipično definišu:
  - Ko su **korisnici** (spoljni akteri)
  - Šta su **ulazi** i **izlazi**
  - Koje se **aktivnosti**
  - Način odvijanja poslova (*workflow* - tok izvršavanja)
  - Ko ih obavlja (unutrašnji akteri)
  - ...



# Značaj modelovanja poslovnih procesa

- Jasno, precizno i nedvosmisleno definisani poslovni procesi
- Jasno identifikovani poslovi preduzeća
- Formalno zapisano znanje o načinu obavljanja poslova u preduzeću
- Mogućnost transfera znanja (Obuka radnika za obavljanje poslova, transfer znanja između pojedinih organizacionih entiteta...)
- Standardizacija obavljanja poslova
- Mogućnost analize poslovnih procesa (Otkrivanje nekonzistentnosti i redundansi, poređenje različitih alternativa ...)
- Mogućnost poboljšanja poslovnih procesa (Povećavanje efikasnosti poslovanja, ušteda, povećanje kvaliteta proizvoda ili usluga ...)
- Mogućnost automatizacije (Formalni modeli se mogu prevesti u izvršne na računaru, prva faza razvoja informacionog sistema ...)





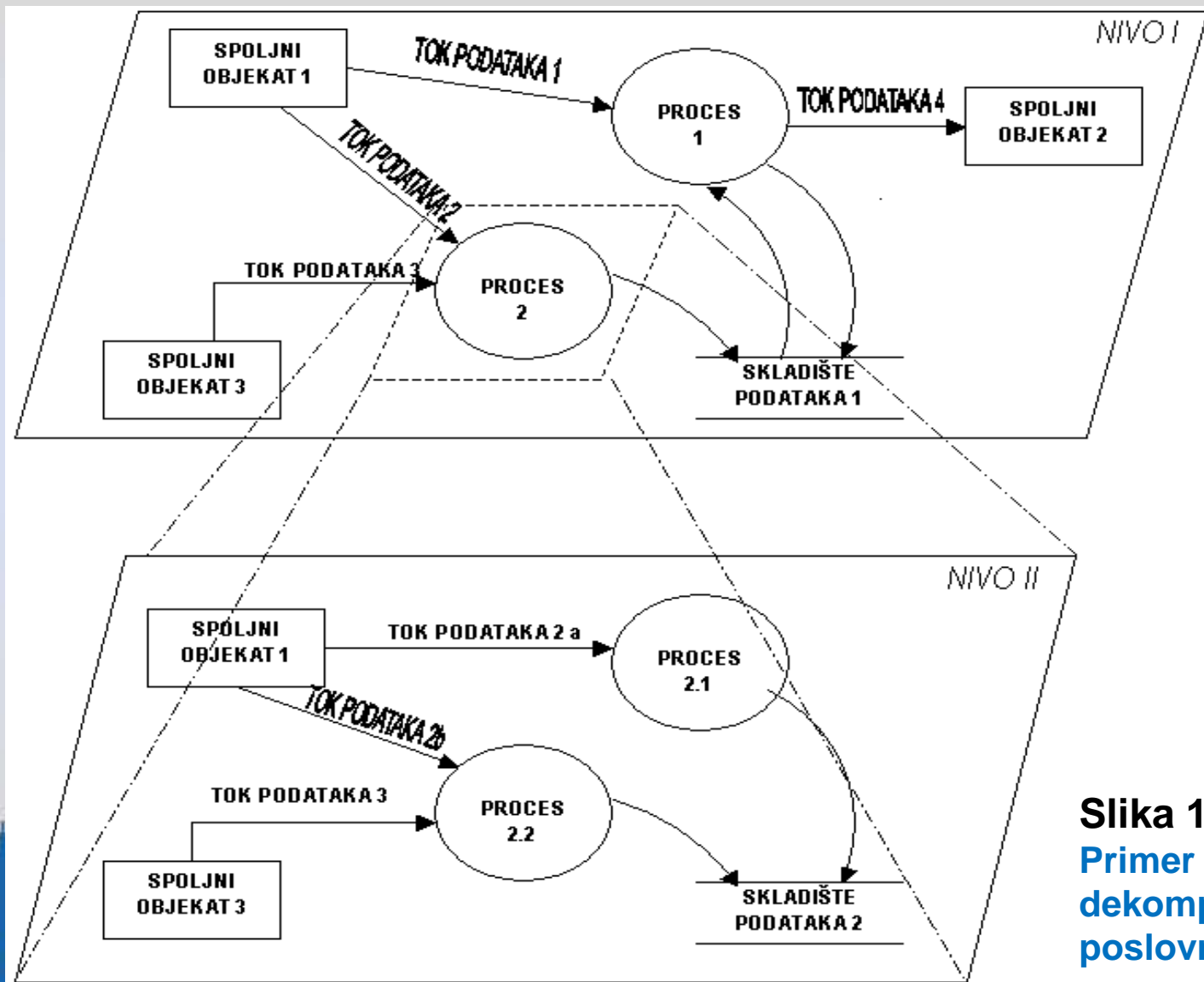
# Problem modelovanja poslovnih procesa

- **Kako** identifikovati i opisati procese
  - Kako nabrojati sve procese?
  - Kako utvrditi granice procesa?
  - Kako specificirati (opisati) karakteristike (statika) i način odvijanja (dinamika) procesa?
- Osnovni problem je **složenost**
  - Veliki broj procesa u organizaciji
- Metodologija
  - Postupak (proces) kako se identifikuju i opisuju procesi
  - Zasniva se na nekom od teorijskih pristupa i formalizama
  - Problem savladavanja složenosti

# Hijerarhijska dekompozicija

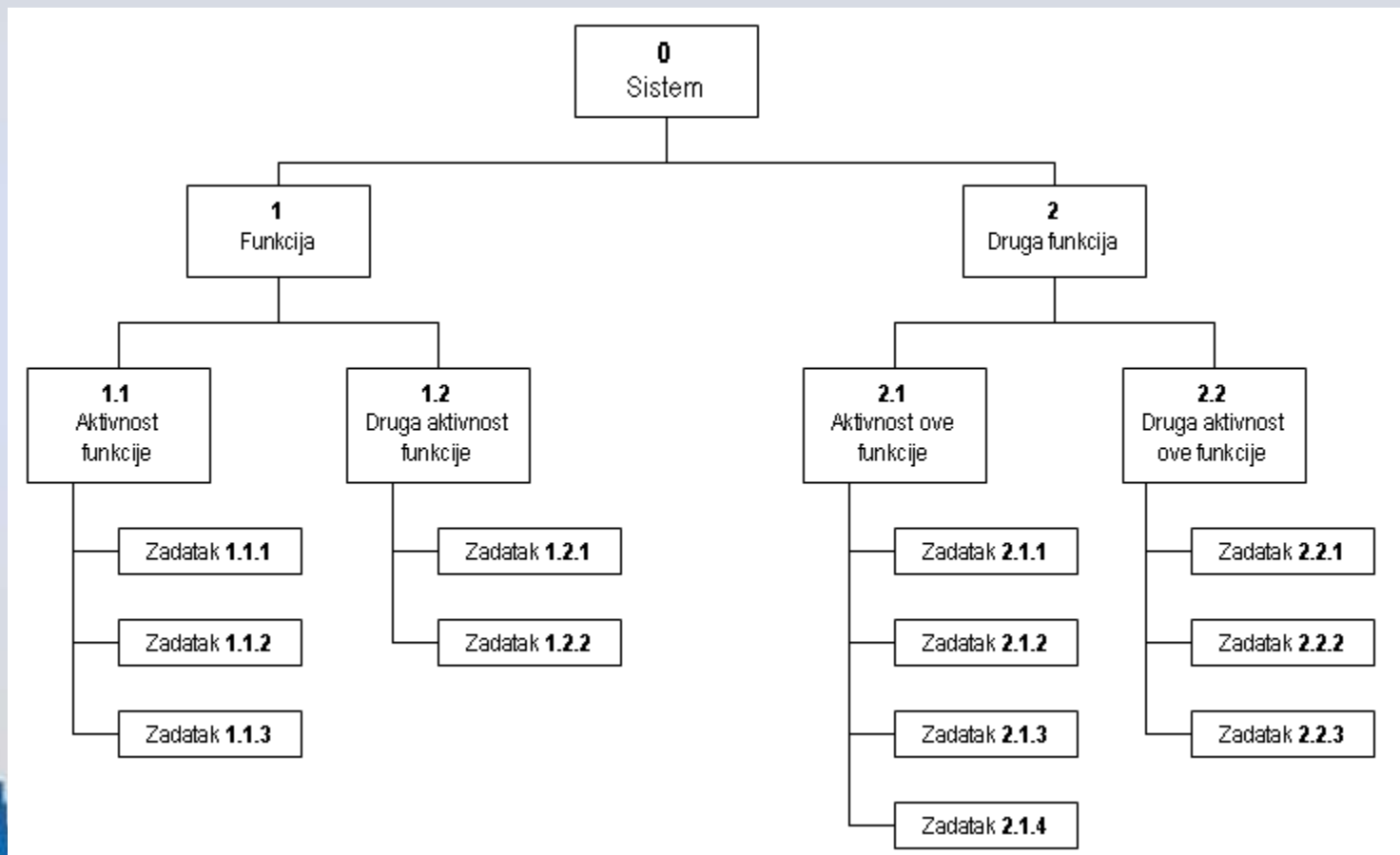
- **Dekompozicija** je način razlaganja sistema na njegove komponente: podsisteme, procese, podprocesse, aktivnosti i zadatke
- **Hijerarhijska dekompozicija** je postupak gde se jedan proces višeg nivoa apstrakcije dekomponuje na njegove podprocesse na nižem nivou apstrakcije
- Jedan od načina savladavanja složenosti poslovnih sistema je **funkcionalna dekompozicija**
  - **Strukturirana sistemska analiza** (*Structured System Analysis, SSA*) je jedna od metoda koja se zasniva na funkcionalnoj dekompoziciji





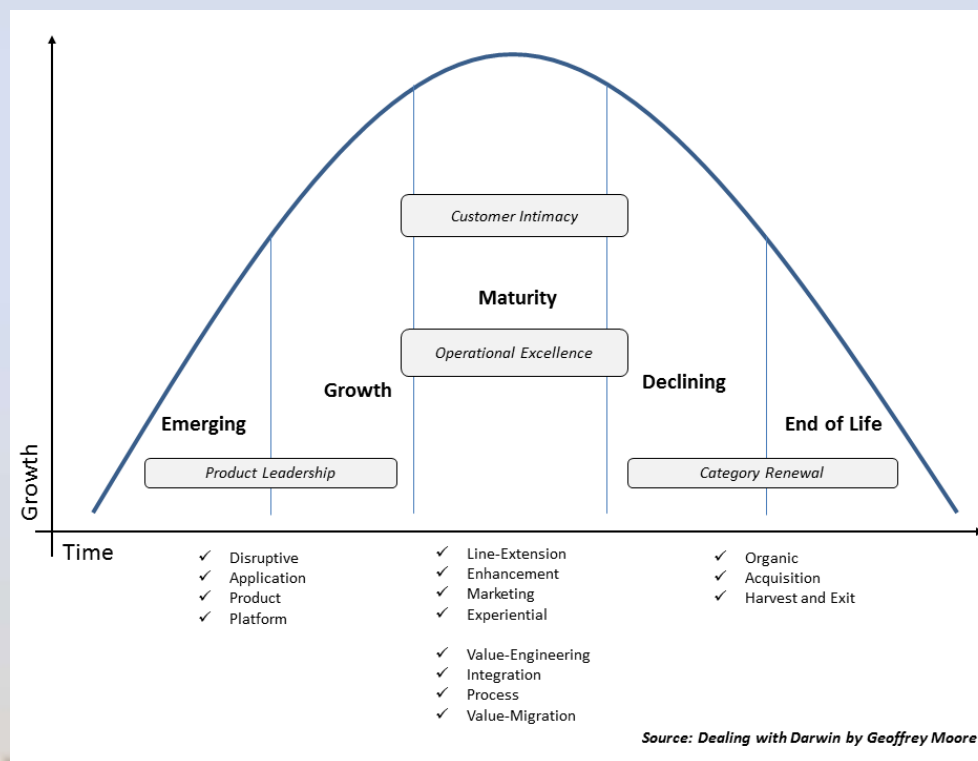
**Slika 1:**  
Primer hijerarhijske  
dekompozicija  
poslovnih funkcija

# Hijerarhijski dijagram strukture funkcija (Stablo aktivnosti sistema)



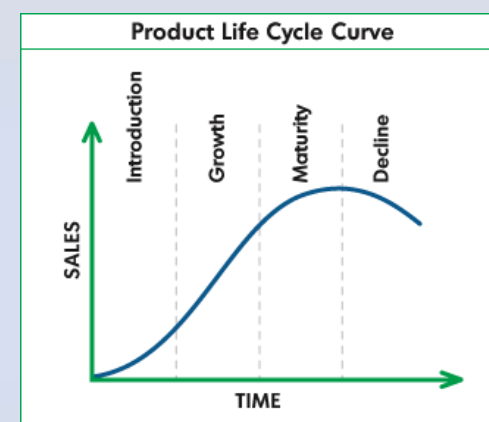
# Kako izvršiti dekompoziciju procesa?

- Potreban je metodološki postupak
  - Primer: životni ciklus entiteta
    - Entiteti mogu biti:
      - proizvodi ili usluge
      - resursi (oprema, kadrovi, novac ...)
      - dokumenta (odluke, planovi, operativna ...)
    - Entiteti prolaze kroz osnovni životni ciklus:
      - Rađanje
      - Razvoj
      - Opadanje
      - Nestanak



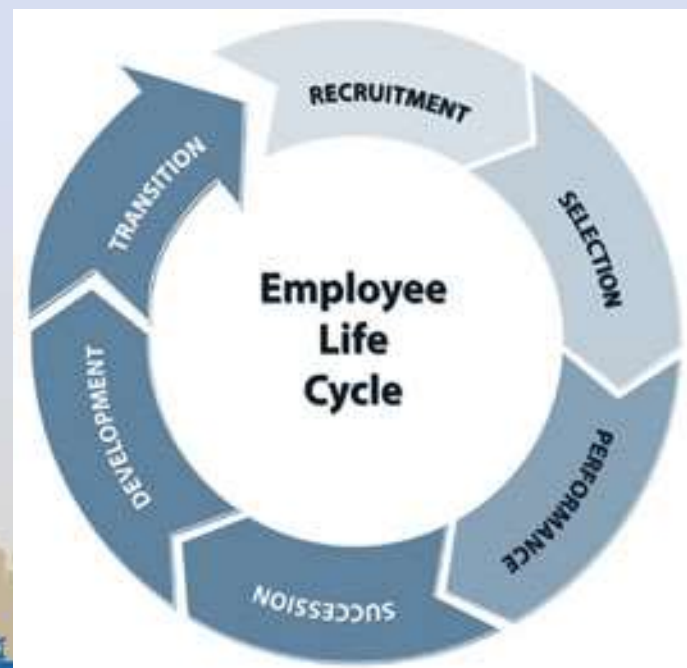
# Primer životnog ciklusa proizvoda

- Planiranje proizvodnje (Priprema rađanja)
- Nabavka (Priprema rađanja)
- Lansiranje proizvodnje (Rađanje)
- Proizvodnja (Razvoj)
- Prodaja (Nestanak)



# Primer životnog ciklusa zaposlenih

- Raspisivanje konkursa (Priprema rađanja)
- Prijem u radni odnos (Rađanje)
- Praćenje radnog odnosa (Razvoj)
- Prestanak radnog odnosa (Nestanak)



# Odakle početi?

- Izvori informacija za postupak dekomponovanja:
  - **Eksperti** u određenim domenima poslovanja
  - **Standardi** za pojedine oblasti poslovanja
  - Poslovni informacioni **sistemi** (ERP, CRM, SCM ...)
  - **Ostale tehnike**: posmatranje rada korisnika, intervjuisanje (slika: primer intervjua), fokus grupe, rad prema instrukcijama korisnika ...

|               |  |
|---------------|--|
| Intervjuisan: | Mirko Mirković, menadžer kreditnog odeljenja |
| Datum:        | 10. Mart 2006.                               |
| Vreme:        | 10.30.                                       |
| Mesto:        | Kancelarija 22, Zgrada administr.            |
| Tema:         | Politika provere kredita                     |

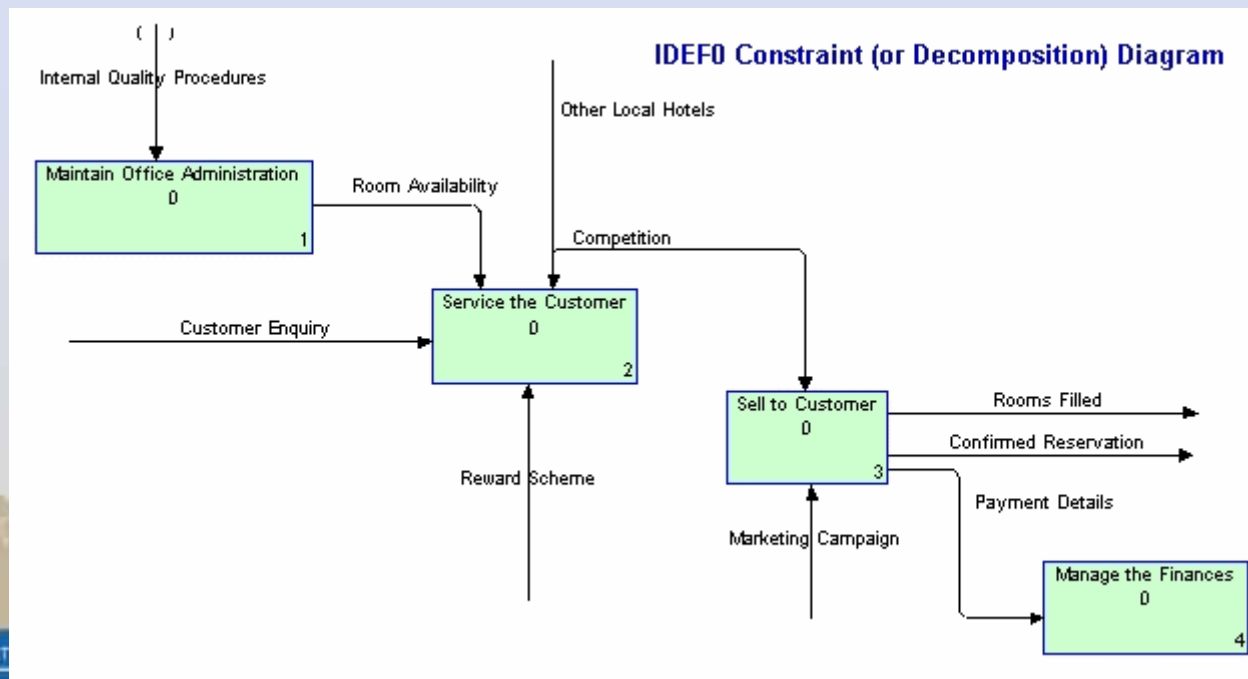
| Dodeljeno vreme | Intervjuista<br>Pitanja ili ciljevi   | Odgovori intervjuisanog |
|-----------------|---|-------------------------|
| 1-2 min.        | <b>Cilj</b><br><i>Otvaranje intervjua:</i> <ul style="list-style-type: none"> <li>▫ Predstaviti se</li> <li>▫ Zahvaliti se gospodinu Mirkoviću na dragocenom vremenu</li> <li>▫ Izložiti svrhu intervjua – razumevanje postojeće politike provere kredita.</li> </ul> |                         |
| 5 min.          | <b>Pitanje 1</b><br>Koje uslove klijent mora da ispuni da bi mu se odobrio kredit?  |                         |
| 3 min.          | <b>Pitanje 3</b><br>Kako se klijenti obaveštavaju da im kredit nije odobren?  |                         |
| 1 min.          | <b>Pitanje 4</b><br>Nakon što je odobren kredit i unešeni podaci u fajl, klijent može da zahteva neke izmene u podacima. Da li treba da se ponovi cela procedura odobravanja kredita iako je nova suma ista kao prethodna?  |                         |
| 1-3 min.        | <b>Pitanje 6</b><br>Sa vašim dopuštanjem, da li bih mogao da razgovaram sa tim pojedincima kako bih naučio proces provere kredita?<br>Ako da: Kada bi bilo odgovarajuće vreme da se sastanem sa svakim od njih?   |                         |
| 1 min.          | <b>Cilj</b><br><i>Zatvaranje intervjua:</i> <ul style="list-style-type: none"> <li>▫ Zahvaliti gospodinu Mirkoviću za saradnju i uveriti ga da će dobiti kopiju zapisnika sa intervjua.</li> </ul>  |                         |
| 21 min.         | Vreme za pitanja i ciljeve  |                         |
| 9 min.          | Vreme za podpitanja i preusmeravanja  |                         |
| 30 min.         | Vreme intervjua (10.30-11.00)   |                         |

**Komentari i beleške:**



# Funkcionalno modelovanje poslovnih procesa

- Dijagram toka podataka (*Data Flow Diagram, DFD*)
- IDEF0 standard za modelovanje poslovnih procesa

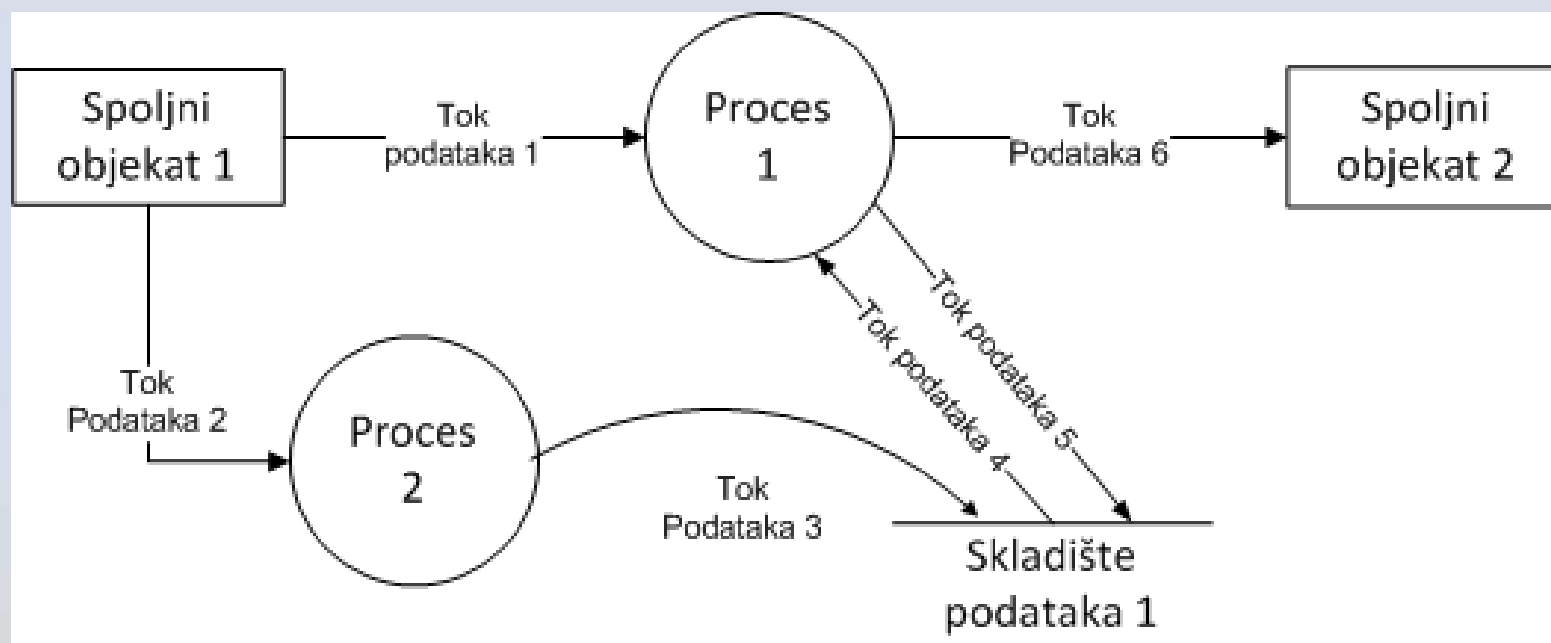


# Dijagram toka podataka

## *Data Flow Diagram (DFD)*

- **Modelovanje procesa** je tehnika koja organizuje i dokumentuje procese sistema i/ili implementira logiku, politike i procedure sistema
- **Dijagram toka podataka (DTP)** je alat koji opisuje tokove podataka kroz sistem i procese koji se izvršavaju u sistemu. Sadrži četiri osnovne komponente:
  - **proces** (*processes*) obrade podataka, koji predstavljaju aktivne komponente sistema (grafički simbol: krug)
  - **spoljne objekte ili spoljne agente** (*external agents*) sa kojima sistem komunicira (grafički simbol: pravougaonik)
  - **skladišta podataka** (*data stores*) koje procesi koriste i/ili ažuriraju (grafički simbol: dve paralelne linije)
  - **tokove podataka** (*data flows*) koji povezuju ostale komponente sistema u celinu (grafički simbol: usmerena linija)

# Osnovne komponente DTP-a

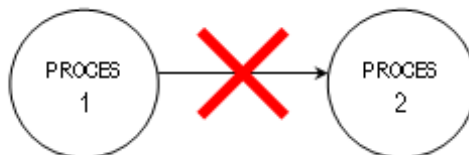
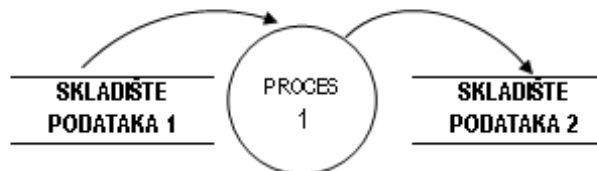
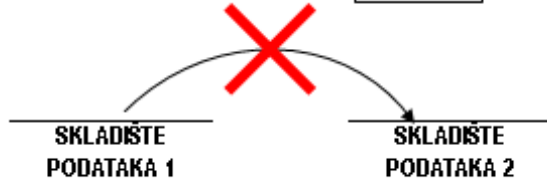
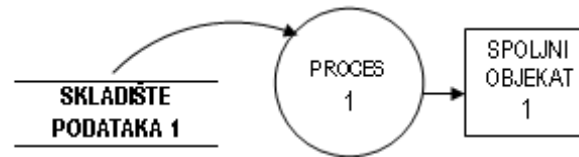
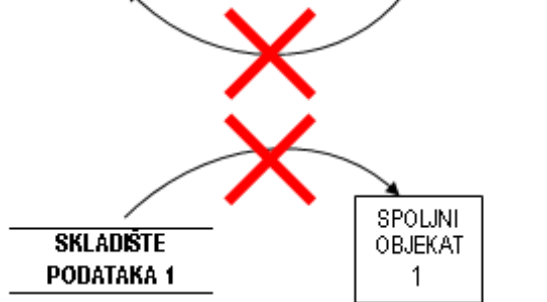
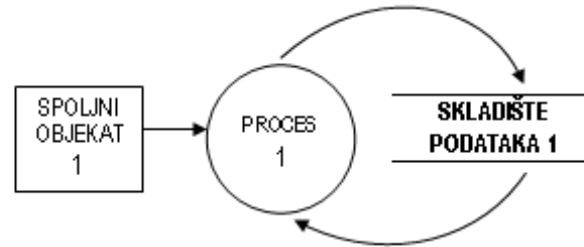
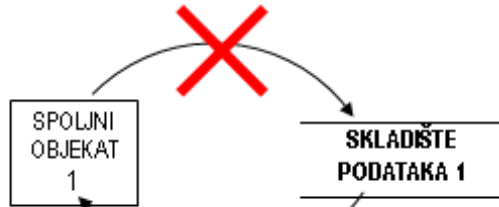




**Nekorektni tokovi podataka**



**Korektni tokovi podataka**

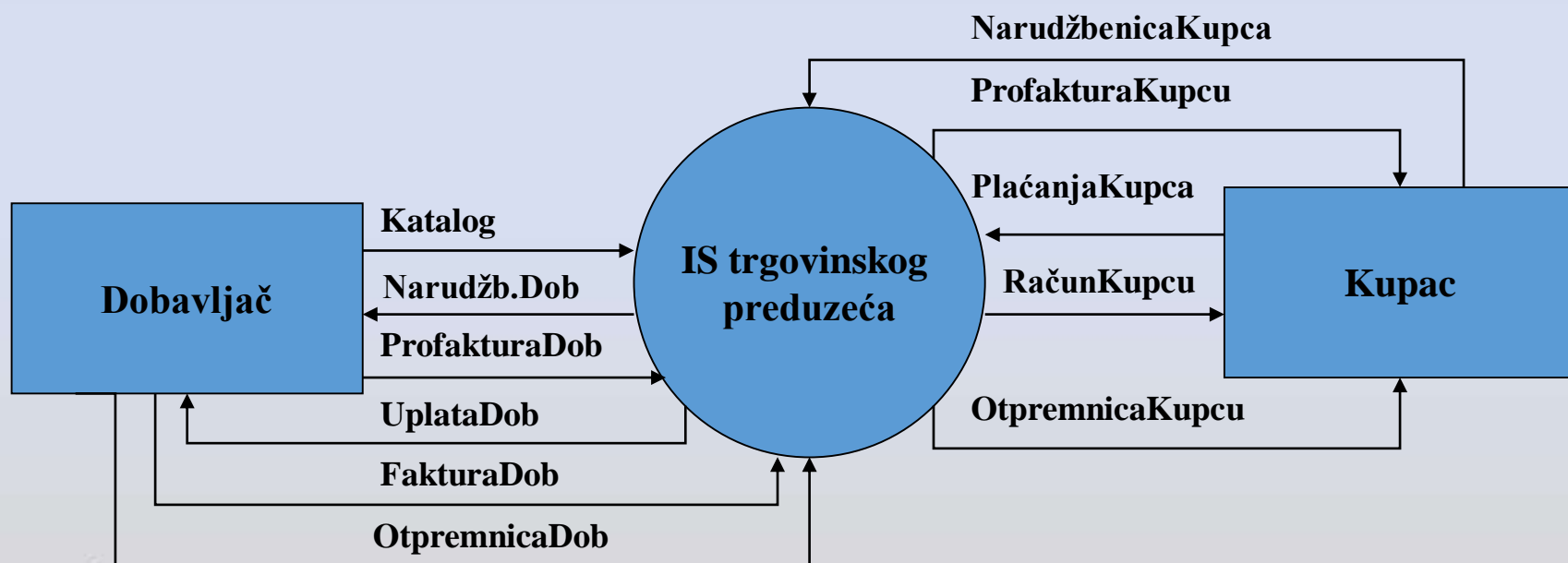


# Postupak hijerarhijske dekompozicije DTP-a

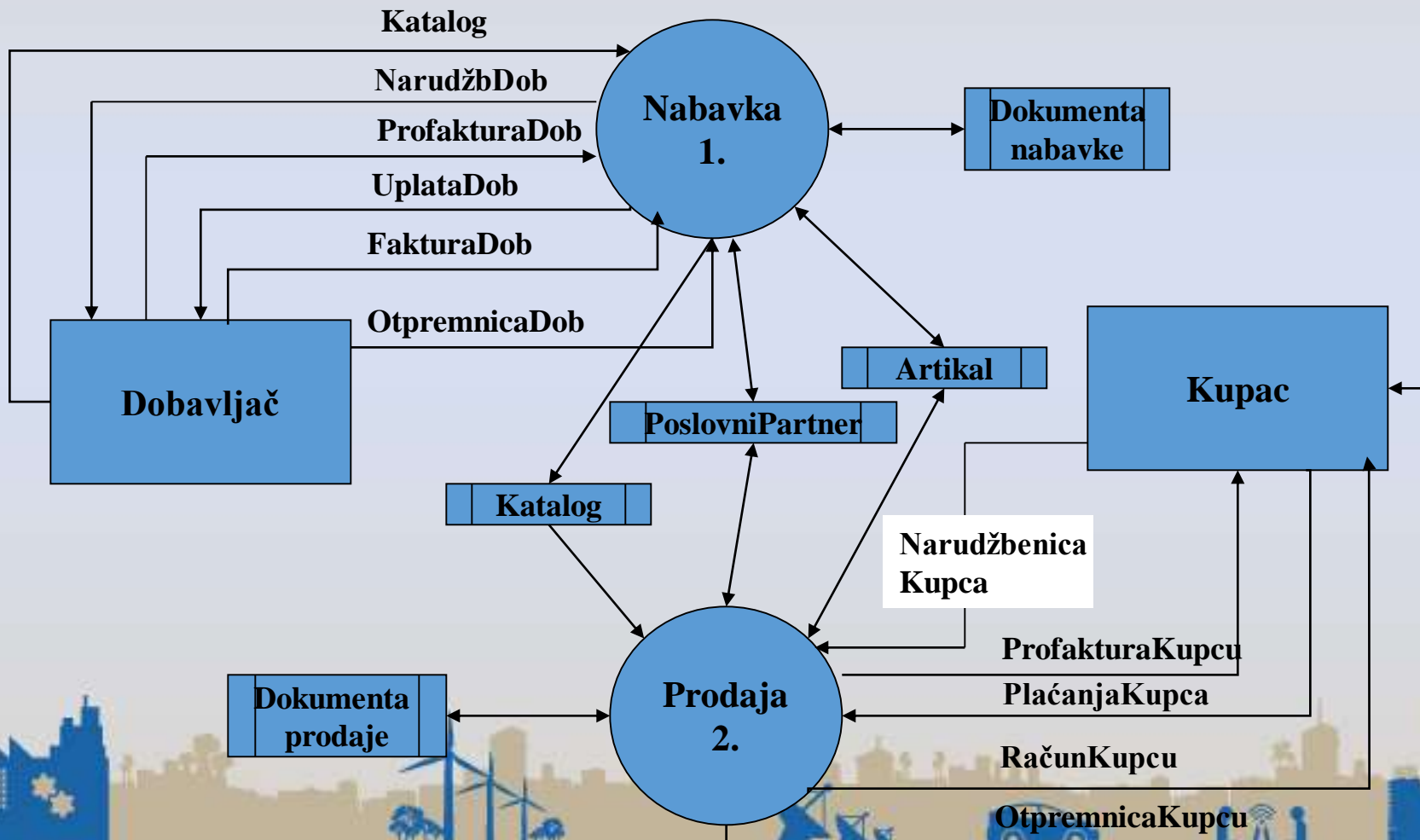
1. Dijagram najvišeg nivoa, koji po pravilu sadrži samo jedan proces koji predstavlja ceo IS, zatim spoljne objekte sa kojima IS komunicira i odgovarajuće tokove podataka - naziva se **dijagram konteksta**
2. Dijagram prvog nivoa predstavlja **dekompoziciju dijagrama konteksta** - Procesi se označavaju brojevima 1,2,3, ....
3. Svaki proces sa dijagrama prvog nivoa se dalje **dekomponuje do nivoa zadatka** (procesa koji se više ne dekomponuju)
4. **Dijagram dekompozicije** prikazuje *top-down* (sa vrha na dole) funkcionalnu dekompoziciju i strukturu sistema
5. Pored procesa, mogu se dekomponovati i tokovi i skladišta - njihov opis se detaljno daje u **rečniku podataka**
6. Najvažnije **pravilo** koje se mora poštovati pri dekompoziciji procesa je pravilo **balansa tokova!**



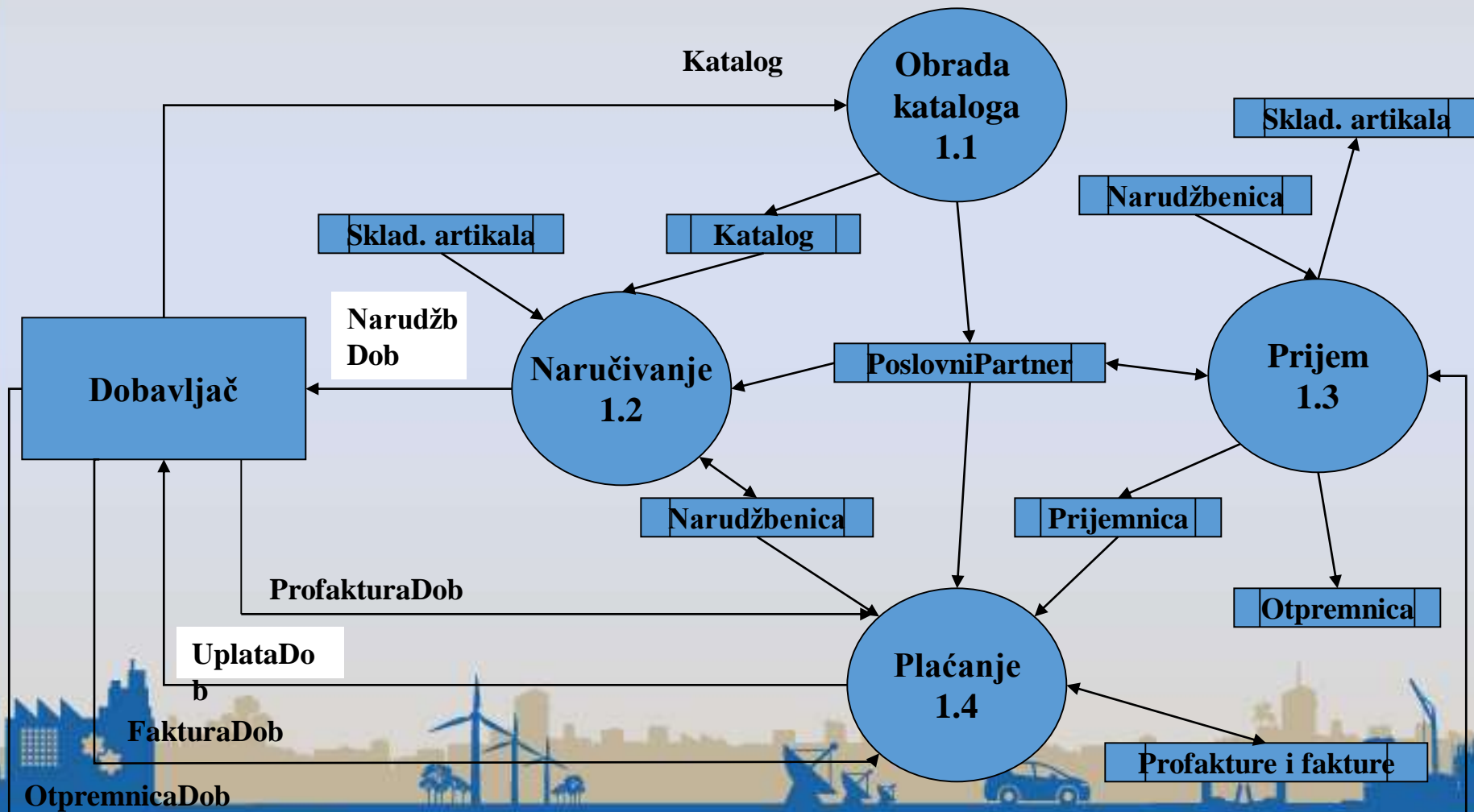
# Primer dijagram konteksta manjeg trgovinskog preduzeća



# Dijagram prvog nivoa

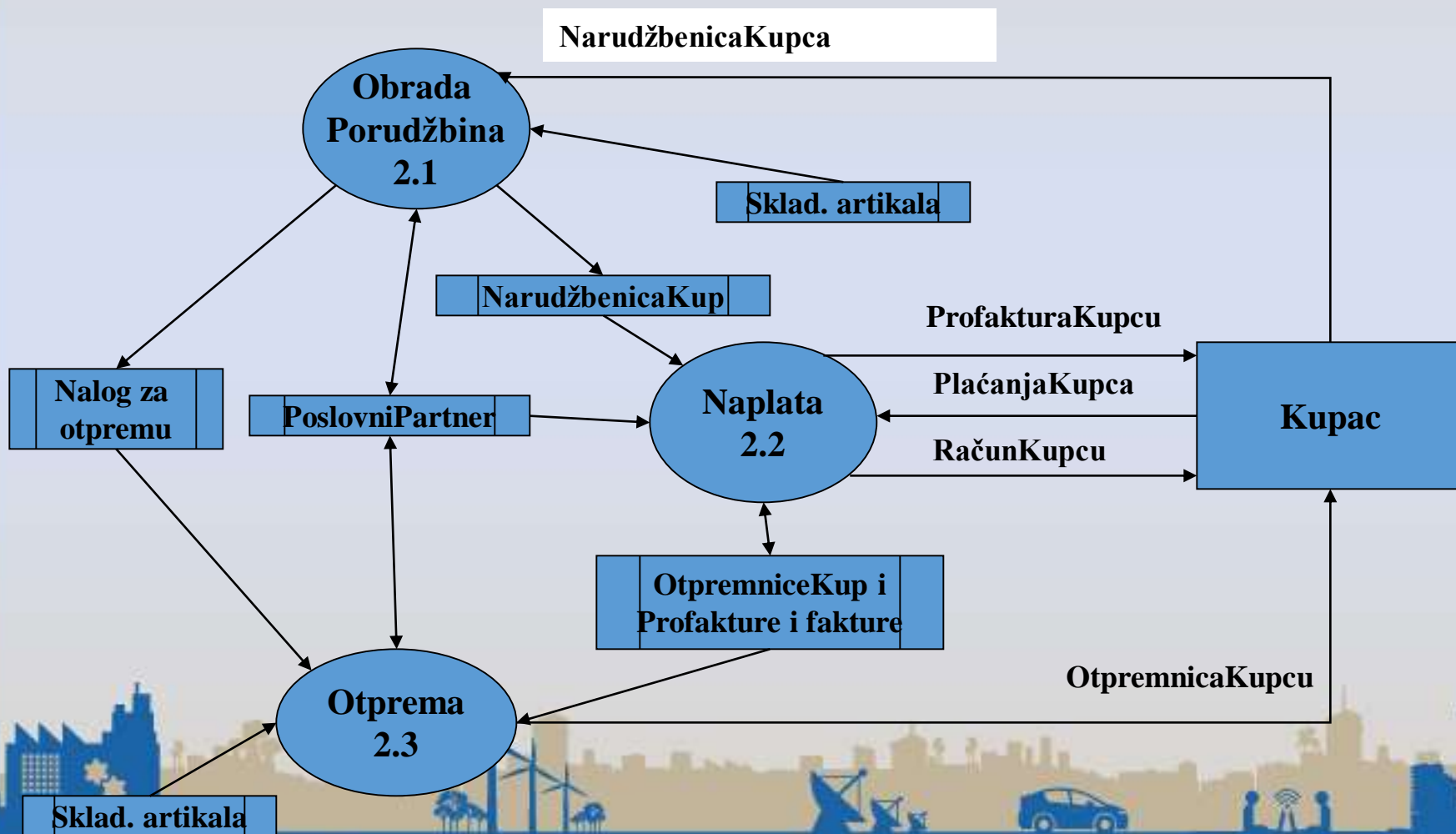


# Dekompozicioni dijagram procesa nabavke

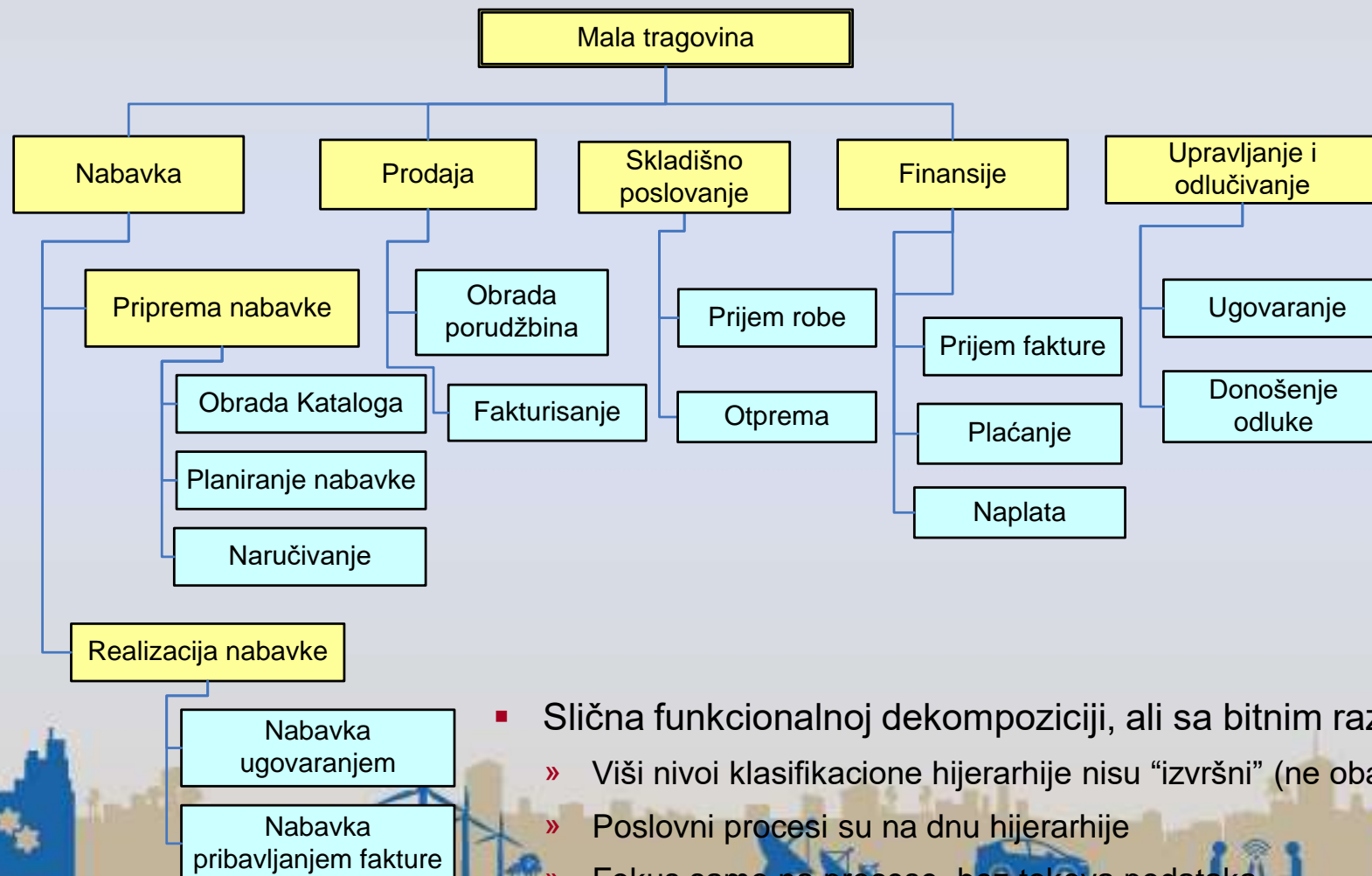




# Dekompozicioni dijagram procesa prodaje



# Klasifikaciona dekompozicija



- Slična funkcionalnoj dekompoziciji, ali sa bitnim razlikama
  - » Viši nivoi klasifikacione hijerarhije nisu “izvršni” (ne obavljaju se)
  - » Poslovni procesi su na dnu hijerarhije
  - » Fokus samo na procese, bez tokova podataka

# Rečnik podataka

- Rečnik podataka (*Data Dictionary*) predstavlja strukturirano skladište meta podataka, tj. podataka o podacima
  - Prvobitno se pojavio kao proširenje dijagrama toka podataka, za opisivanje sadržaja i strukture svih tokova i skladišta podataka
- Primer BNF notacije rečnika podataka (Backus-Naur Form):

| Znak | Opis  |
|------|---|
| =    | Struktura sa leve strane se sastoji od delova sa desne („sastavljeno od“) |
| +    | Agregacija elemenata  |
| ()   | Opcionalnost elemenata u zagradi (0 ili 1)                                |
| { }  | Ponavljanje (iteracija) elemenata u zagradi do konačnog broja puta        |
| [ ]  | Alternativa (selekcija) elemenata u zagradi                               |
|      | Odvajanje alternativnih elemenata u [ ] izrazu                            |
| -    | Početna i završna vrednost raspona definisanog [ ] izrazom                |
| **   | Komentar  |
| @    | Oznaka ključa   |

# Primer rečnika podataka

Račun i Stavke\_računa se notacijom BNF prikazuju:

$$\begin{aligned} \text{Racun} &= @\text{BrRac} + \text{DatRac} + \text{BrKupca} \\ &+ \{ \text{SifArt}, \text{NazArt}, \text{Cena}, \text{Kol}, \text{Vrednost} \} \\ &+ (\text{IznosRac}) \end{aligned}$$

Prethodni primer se može napisati i na sledeći način:

$$\begin{aligned} \text{Racun} &= @\text{BrRac} + \text{DatRac} + \text{BrKupca} \\ &+ \{ \text{StavkaRac} \} \\ &+ (\text{IznosRac}) \end{aligned}$$
$$\text{StavkeRacuna} = @\text{SifArt}, \text{NazArt}, \text{Cena}, \text{Kol}, \text{Vrednost}$$


# IDEF standardi za modelovanje sistema

- Ranih 90-tih IDEF (*Integration DEFINition*) Users Group usvojila je standarde:
  - za funkcionalno modelovanje: IDEF0
  - za informaciono modelovanje: IDEF1X
- Softverska realizacija IDEF standarda:
  - *BPwin (Business Process Windows)* firme LogicWorks za IDEF0 standard
  - *ERwin (Entity Relationships for windows)* alat za IDEF1X



# Semantika IDEFØ

- Ulaz

- » Materijal ili informacija koja se koristi ili transformiše radi definisanja izlaza

- Kontrola

- » Uslovi izvođenja aktivnosti: kako, kada i da li će se aktivnost izvesti

- » Pravila, politike, procedure ili standardi koje utiču na aktivnost

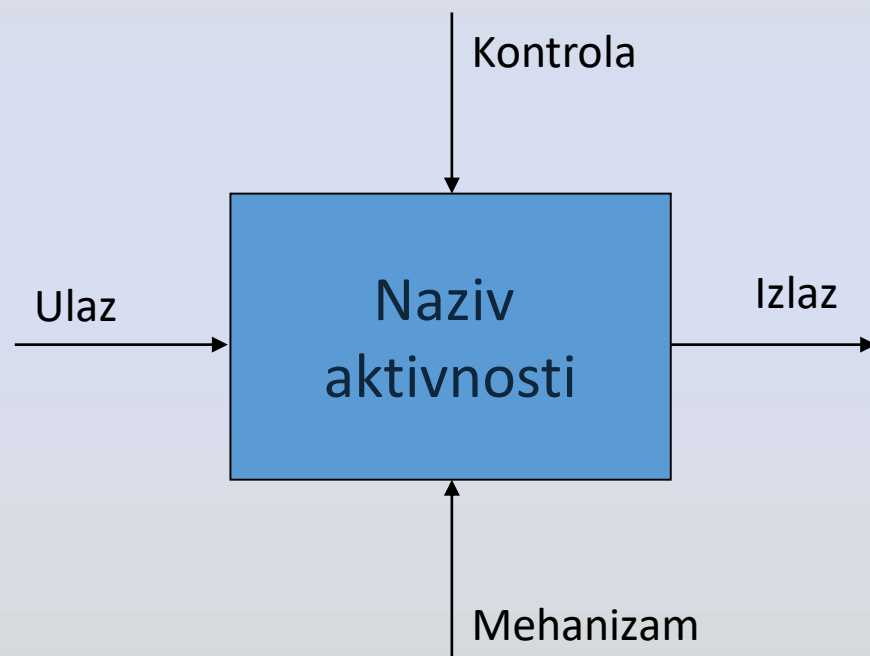
- Izlaz

- » Materijal ili informacije stvorene aktivnošću

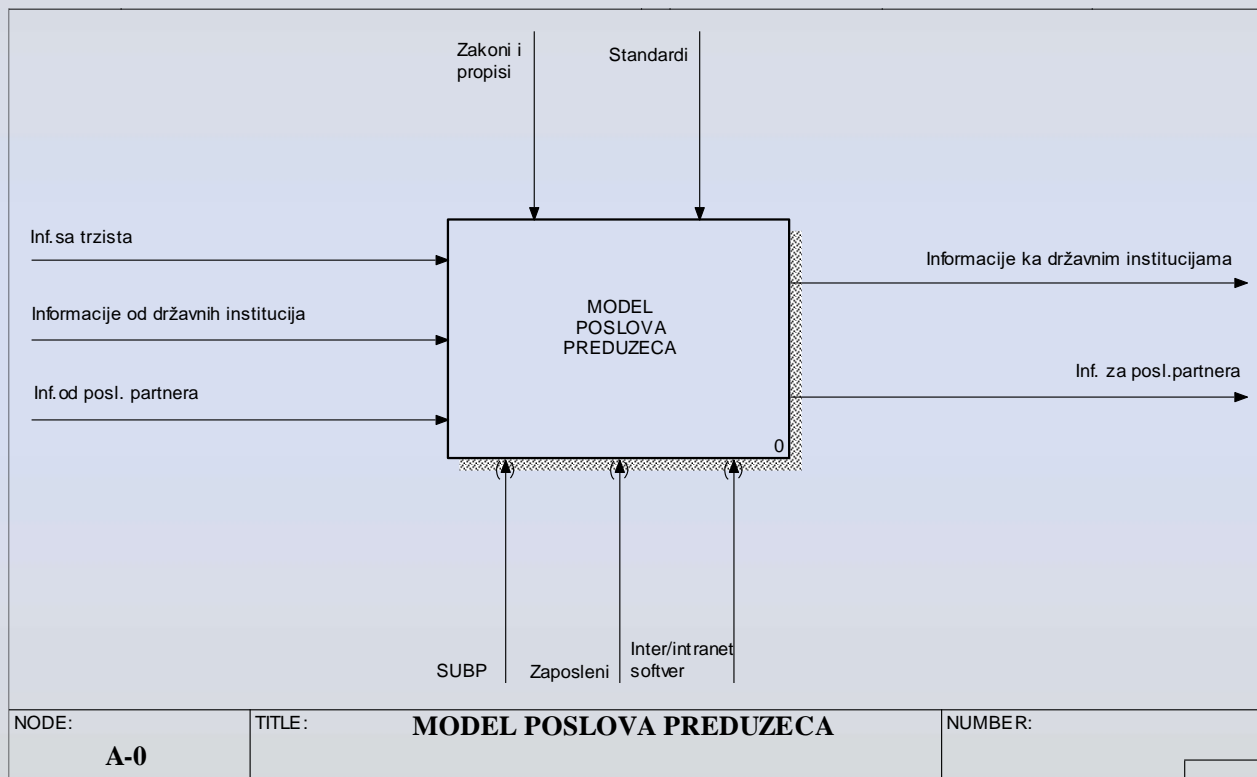
- Mehanizam

- » Ko izvodi ili gde se izvršava aktivnost

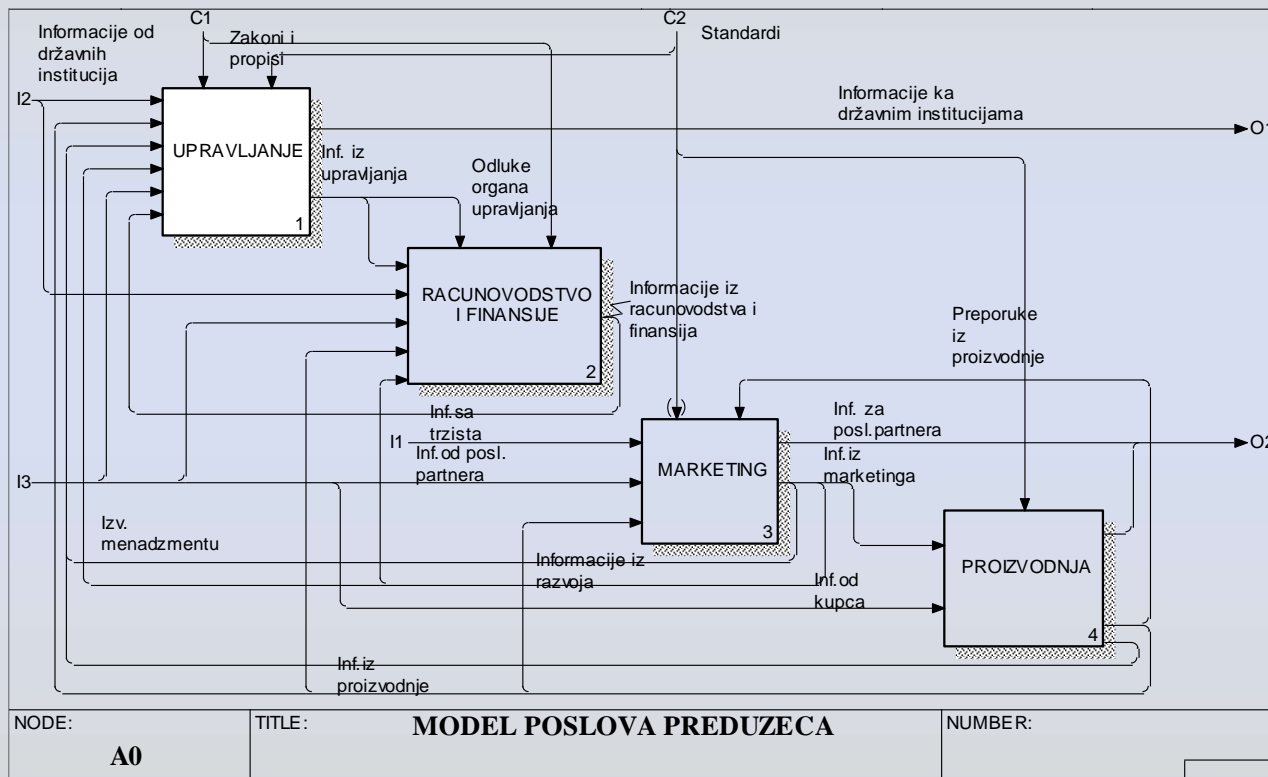
- » Ljudi, mašine, oprema, tehnologija i dr. sve ono što obezbeđuje energiju potrebnu za izvođenje aktivnosti



# Dijagram konteksta modela poslova preduzeća



# Dijagram nultog nivoa



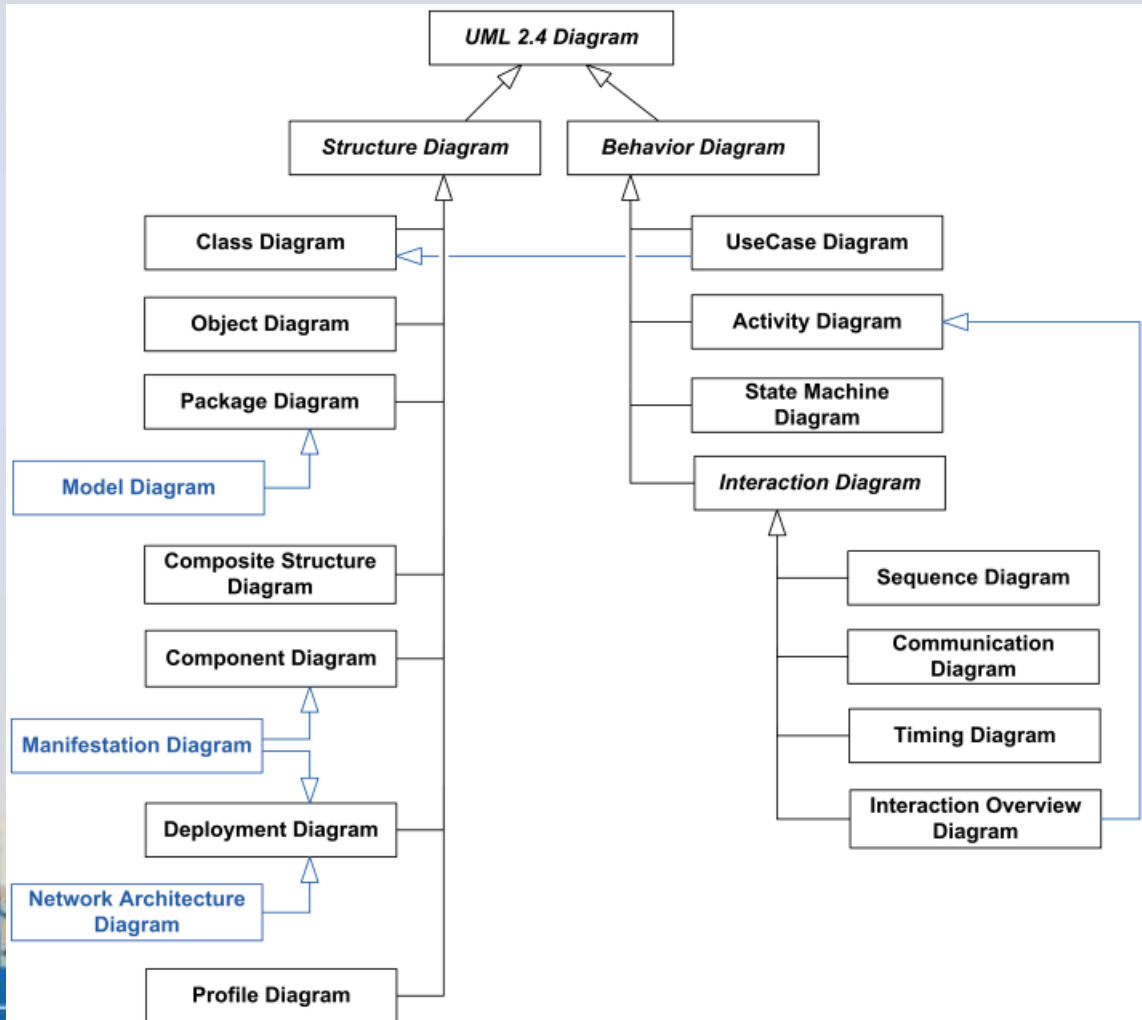


# Stablo aktivnosti procesa upravljanja kvalitetom



# Objektno-orijentisano modelovanje poslovnih procesa

- UML sadrži skup dijagrama za opis **strukture** i **dinamike** svih vrsta sistema



# UML

## *Unified Modeling Language*



- *OMG (Object Management Group)* – organizacija zadužena za brigu o standardizaciji UMLa
- UML je standardni vizuelni jezik za:
  - Modelovanje procesa
  - Analizu, projektovanje i implementaciju softverskih sistema
- Koriste ga **poslovni analitičari, softver arhitekta i developeri** za:
  - opisivanje, specifikaciju, dizajniranje i dokumentovanje postojećih ili novih poslovnih procesa, strukture i ponašanja softverskih sistema
- Najviše se koristi u **razvoju informacionih sistema**
- Trenutna verzija: **UML 2.4.1** (Avgust 2011)



# Nivoi specifikacije procesa

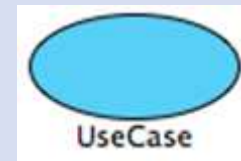
- Gruba specifikacija
  - Akteri
  - Složeni procesi
  - Mogu se predstaviti modelom slučajeva korišćenja
- Detaljna specifikacija
  - Dinamički aspekt – ponašanje sistema
    - Dijagram aktivnost, dijagram stanja, dijagrami interakcije (dijagram sekvenci, dijagram komunikacije, ...)
  - Statički aspekt – struktura sistema
    - Dijagram klasa, dijagram objekata, dijagram paketa, dijagram komponenti, dijagram uvođenja ...



# Model slučajeve korišćenja

## Osnovni koncepti modelovanja

- **Slučaj korišćenja** (*Use case*) predstavlja:
  - Funkcionalnost (grupa funkcionalnosti) sistema
  - Željeno ponašanje sistema koji se razvija
  - Sekvencu aktivnosti koje sistem izvršava, a koje su od posebne važnosti za aktera
  - Šta sistem radi, ali ne i kako radi, ne pokazuje kako će ponašanje (funkcionalnosti) sistema biti implementirano
  - Dogovor između naručioca sistema i razvojnog tima šta treba da se razvije
- **Akter** je:
  - Spoljni korisnik
  - interni korisnik koji ima određenu ulogu (rolu) u sistemu
  - informacioni sistem, aplikacija ...
  - hardverski uređaj
  - drugi sistemi koji su u interakciji sa posmatranim sistemom (slučajevima korišćenja sistema) koji se razvija



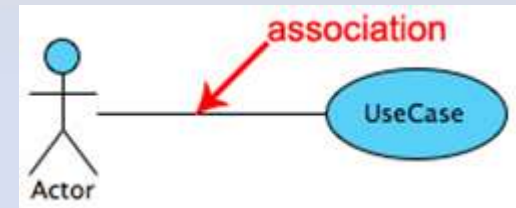
# Model slučajeva korišćenja

## Osnovni koncepti modelovanja

### • Relacije

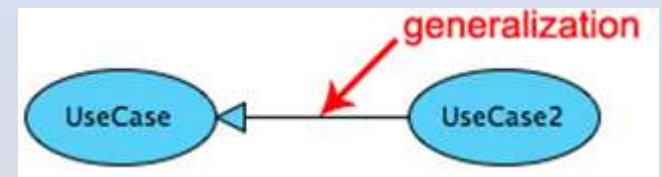
#### • Asocijacije

- komunikaciona relacija između aktera i use case-a
- Smer asocijacije ukazuje ko inicira komunikaciju
- Može biti jednosmerna ili dvosmerna



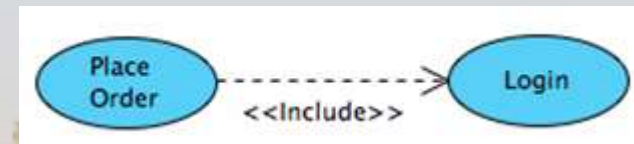
#### • Generalizacije

- Apstrakni prikaz aktera ili slučajeva korišćenja
- Ukazuje na opšti use case i njemu određenije / konkretnije use case-ove
- Veza opšteg i specifičnog use case-a koji nasleđuje opis opšteg use case-a



#### • Uključivanja (*Include*)

- Ukazuje koja joj funkcionalnost obavezno prethodi
- Ukoliko više use-case-ova koristi istu obaveznu funkcionalnost (npr. login), tada se ona može izdvojiti u poseban use case i *include* relacija je usmerena od osnovnog do izdvojenog use case-a

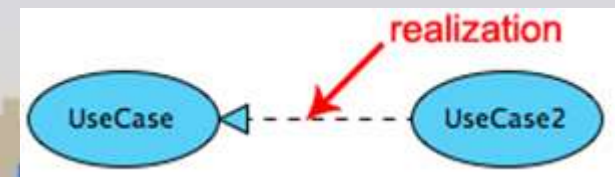
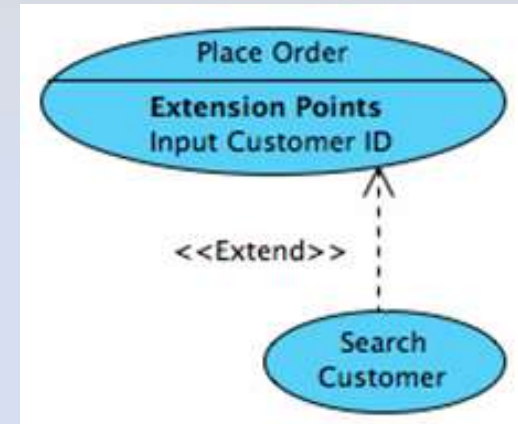


# Model slučajeva korišćenja

## Osnovni koncepti modelovanja

- **Relacije (nastavak):**

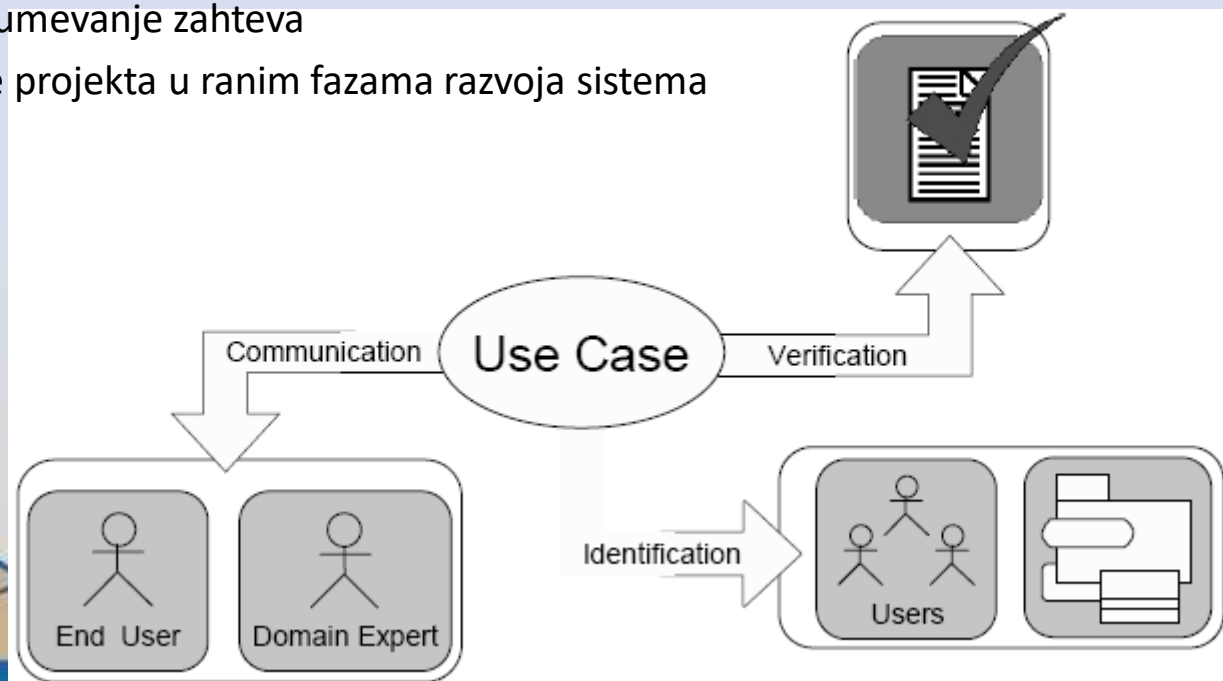
- Proširivanja (*Extend*)
  - Između use case-ova
  - Ukazuje se na alternativni, opcioni ili onaj use case koji se događa pod određenim uslovima
- Zavisnosti (*Dependency*)
  - Između use case-ova
  - Ukazuje na to da use case zahteva druge use case-ove za svoje određeniju specifikaciju ili implementaciju
- Realizacije (*Realization*)
  - Povezuje specifikaciju i implementaciju te specifikacije
  - Pokazuje detaljniju realizaciju ili implementaciju apstraktnog use case-a



# Model slučajeve korišćenja

## *Use-case model*

- **Identifikuje** korisnike sistema i ono šta sistem treba da radi
- **Verifikuje** sve prikupljene zahteve
  - Osigurava da razvojni tim razume zahteve korisnika
- Obezbeđuje **komunikaciju** sa krajnjim korisnicima i ekspertima
  - Osigurava obostrano razumevanje zahteva
  - Omogućava odobravanje projekta u ranim fazama razvoja sistema





# Model slučajeve korišćenja

- Pojednostavljeni grafički prikaz složenog sistema
- Glavni model slučajeve korišćenja (*main use case*):
  - Prikaz glavnih funkcionalnosti sistema
  - Glavnih aktera
  - Domen ili granice sistema
- Različite perspektive use case modela:
  - Model koji prikazuje sve use case-ove izabranog aktera
  - Model koji prikazuje sve use case-ove realizovane u jednoj iteraciji
  - Model koji prikazuje jedan use case i sve njene relacije



# Kreiranje modela slučajeva korišćenja

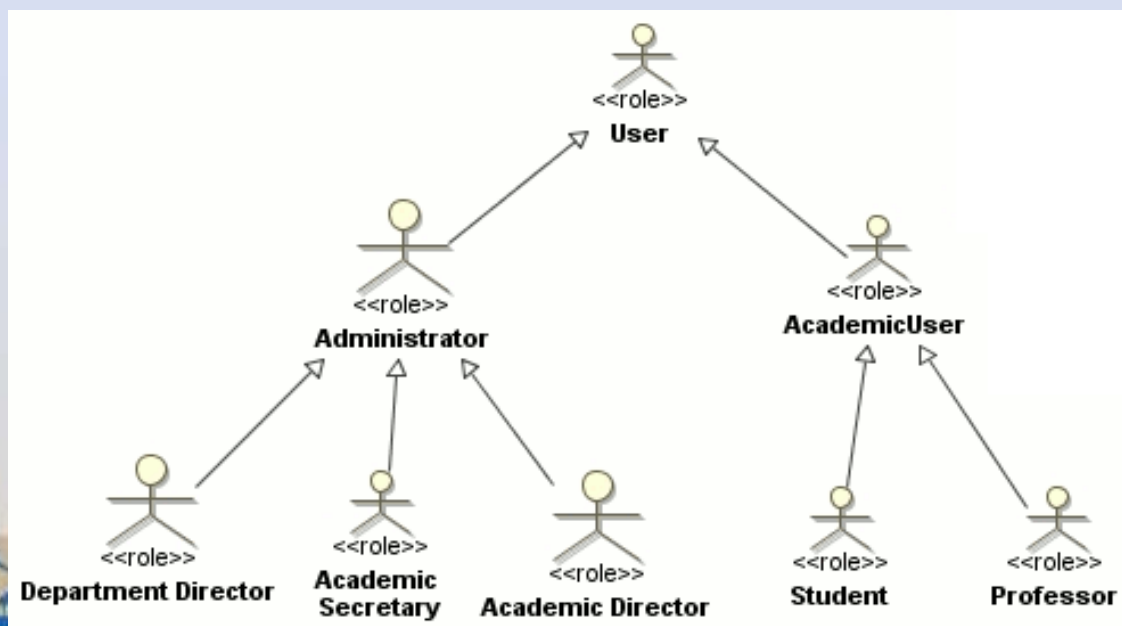
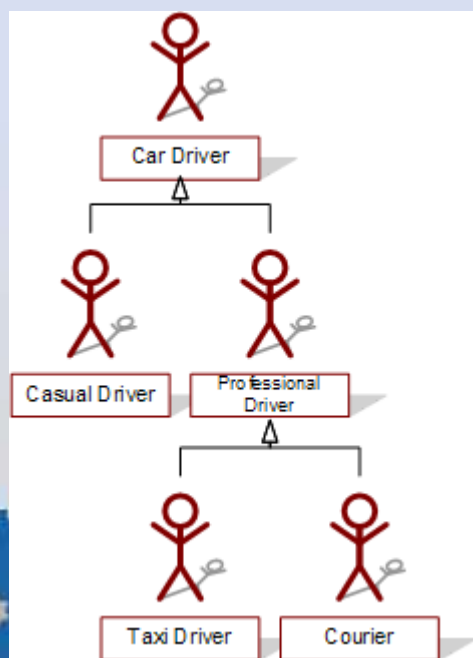
## Odrediti aktere

- Akteri (*Actors*)
  - nisu deo sistema već predstavljaju bilo koga ili bilo šta što mora da uspostavlja interakciju sa sistemom
  - Može samo da unosi i/ili prima informacije iz sistema
- Neka pitanja koja mogu da pomognu pri identifikovanju aktera:
  - Ko je zainteresovan za određeni zahtev?
  - Gde se u organizaciji koriste use case-ovi?
  - Ko će koristiti ili imati koristi od upotrebe aplikacije?
  - Ko će kreirati, ažurirati i brisati podatke?
  - Ko će pružati podršku i održavati sistem?
  - Da li ovaj sistem koristi spoljašnje resurse?
  - Da li jedna osoba ima nekoliko različitih uloga?
  - Da li nekoliko osoba ima istu ulogu/rolu?
  - Da li sistem uspostavlja interakciju sa nasleđenim sistemom?



# Apstraktni akter

- Ukoliko se isti slučaj korišćenja može povezati sa različitim akterima, pogodno je definisati apstraktnog aktera i opisati samo jedan slučaj korišćenja
- Kada dva aktera imaju slične uloge u odnosu na sistem oni mogu naslediti zajedničkog apstraktnog aktera



# Primer 1

## Odrediti aktere u sistemu za registraciju na kurseve

- Korisnički zahtevi :
  - Studenti žele da imaju mogućnost online registracije na željene kurseve
  - Profesori žele da imaju mogućnost online odabira kurseva koje mogu da podučavaju u narednom semestru
  - Referent studentske službe (*Registrar*) želi da mu se olakša kreiranje nastavnih planova i kataloga kurseva za semestar, kao i proces kreiranja i održavanja informacija o kursevima, profesorima i studentima
  - Sistem naplate (*Billing system*) treba da prihvata informacije koji studenti su registrovani na kurseve
- Na osnovu prethodnih zahteva identifikovani su sledeći akteri:
  - » Student
  - » Profesor
  - » Referent studentske službe
  - » Sistem naplate

# Kreiranje modela slučajeva korišćenja Odrediti slučajeve korišćenja

- Slučajevi korišćenja (*Use case*):
  - modeluju dijalog između aktera i sistema
  - predstavlja glavni deo neke kompletne funkcionalnosti od početka do kraja
  - niz aktivnosti koje izvodi sistem, a koji daje merljive rezultate od vrednosti za pojedinačnog aktera
- Pitanja koja mogu pomoći u identifikaciji slučajeva korišćenja sistema:
  - Koji su zadaci svakog aktera?
  - Da li će neki akter kreirati, čuvati, menjati, brisati ili čitati informacije u sistemu?
  - Koji će slučaj korišćenja kreirati, čuvati, menjati, brisati ili čitati ove informacije?
  - Da li će neki akter morati da informiše sistem o iznenadnim spoljašnjim promenama?
  - Da li postoji potreba da neki akter bude informisan o izvesnim događajima u sistemu?
  - Koji slučajevi korišćenja će podržavati i održavati sistem?
  - Da li svi funkcionalni zahtevi mogu biti izvršeni upotrebom slučajeva korišćenja?

# Primer 2

## Odrediti slučajeve korišćenja u sistemu za registraciju na kurseve

- Sistem mora da se bavi sledećim problemima:
  - Student koristi sistem da bi se registrovao na kurseve
  - Profesor koristi sistem da bi pregledao raspored kurseva kako bi selektovao kurseve koje može da podučava tokom semestra
  - Referent studentske službe je odgovoran za kreiranje kataloga kurseva u semestru i za održavanje svih informacija o kursevima, studentima i profesorima koje su potrebne sistemu
- Na osnovu zahteva mogu se identifikovati sledeći slučajevi korišćenja:
  - » Registrovanje na kurseve
  - » Pregled rasporeda kurseva
  - » Selektovanje kurseva koji će se predavati
  - » Kreiranje kataloga kurseva
  - » Održavanje informacija o kursevima
  - » Održavanje informacija o profesorima
  - » Održavanje informacija o studentima

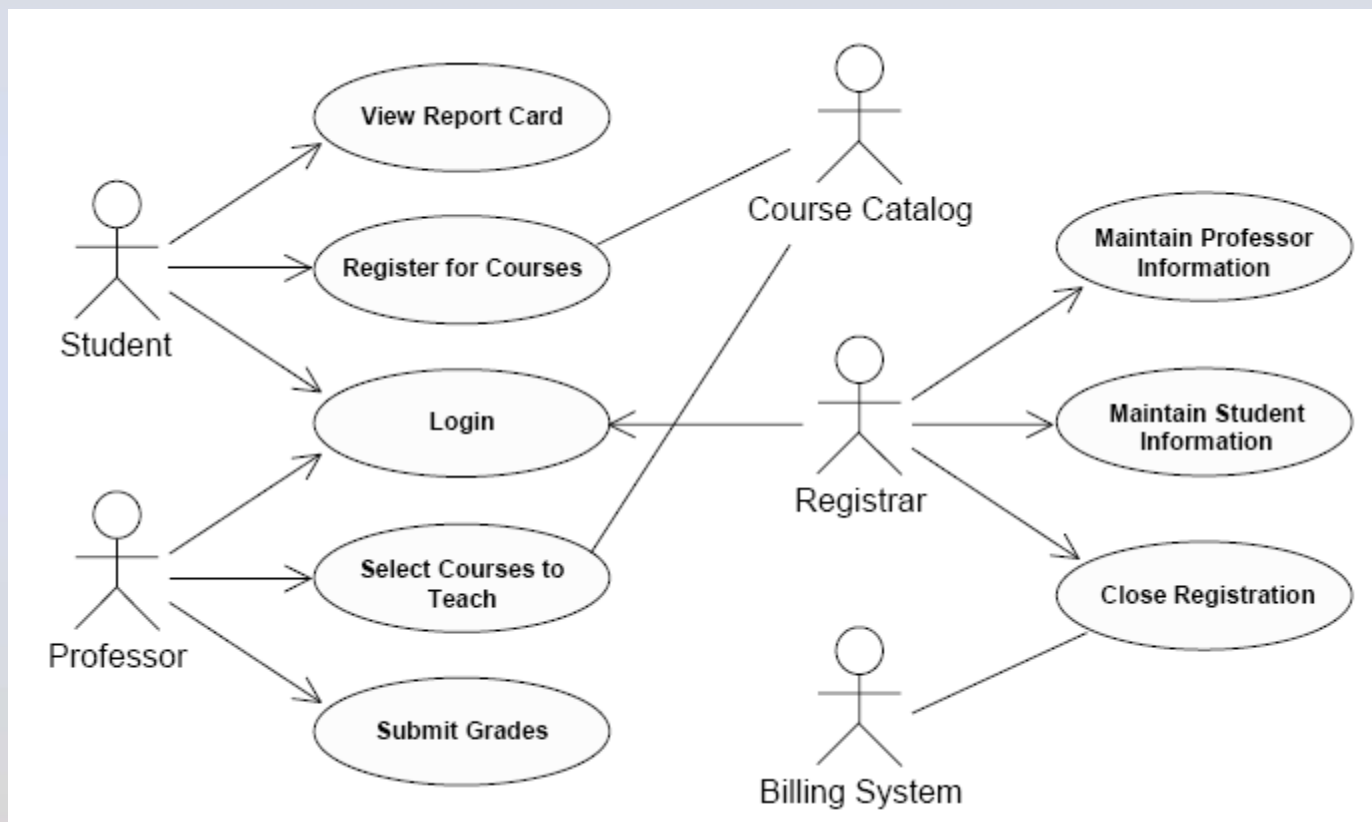
# Zadatak 1:

## Kreirati model slučajeve korišćenja

- Sistem treba da omogući:
  - Obavezno logovanje korisnika sistema
  - Registraciju studenata na željene kurseve
  - Profesorima selekciju kurseva i unos ocena
  - Referentu u studentskoj službi mogućnost održavanja dosijea studenata i profesora, kao i mogućnost dobijanja informacije o uplatama studenata od sistema za naplatu kako bi mogli da zatvore registraciju
  - Sistem kataloga kurseva treba da održava informacije o kursevima kako bi omogućio da studenti i profesori imaju ažurirane informacije



# Zadatak 1



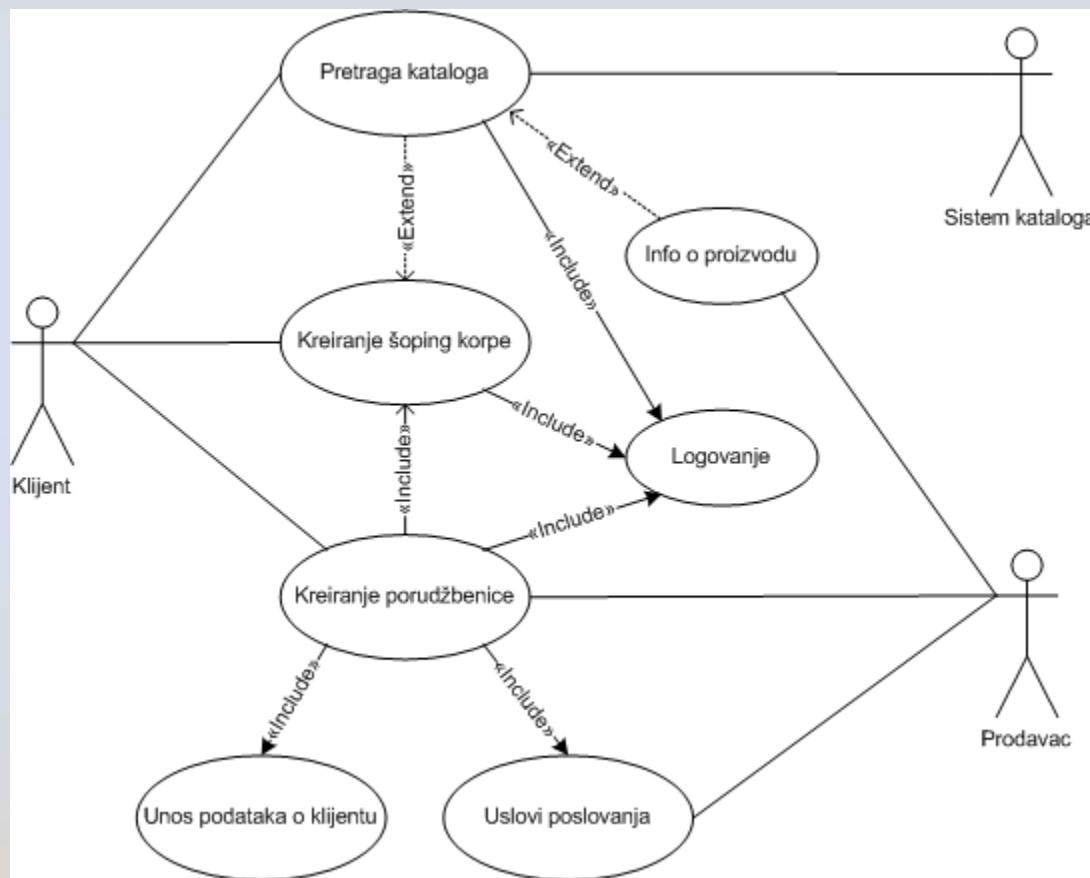


# Zadatak 2:

## Kreirati use case model za kreiranje online porudžbine

- Sistem treba da omogući:
  - Da klijent može da pretražuje kataloge proizvoda koje održava sistem kataloga proizvoda
  - Da klijent može da dobije dodatne informacije o proizvodima od prodavca
  - Da u katalogu može da čekira proizvode koje želi da doda u šoping korpu
  - Da klijent može da pregleda šoping korpu na osnovu koje se kreira dokument porudžbenice
  - Da može u svakom trenutku da pregleda katalog proizvoda kako bi eventualno dopunio šoping korpu
  - Da klijent popuni porudžbenicu sa svojim podacima
  - Da vidi uslove i ukupan iznos porudžbenice
  - Pre nego što pošalje porudžbenicu, mora da se složi sa uslovima plaćanja koje prodavac ažurira po potrebi

# Use case model: Kreiranje online porudžbine

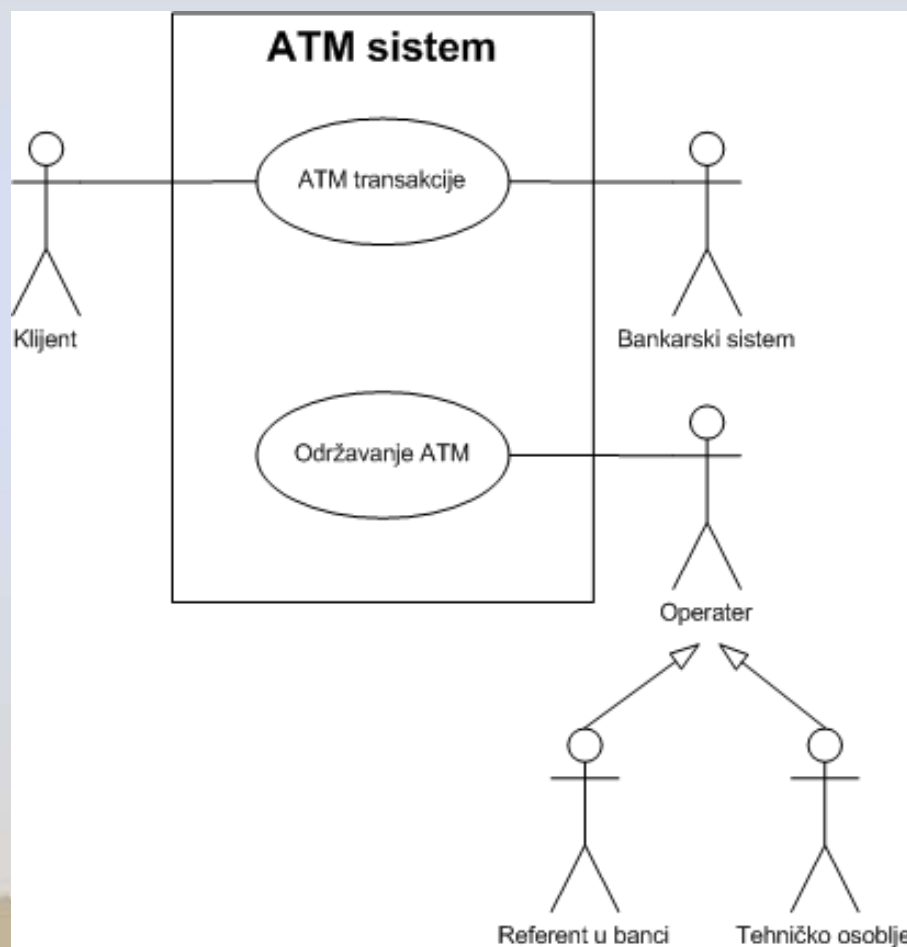


# Zadatak 3:

## Kreirati use case model za ATM sistem

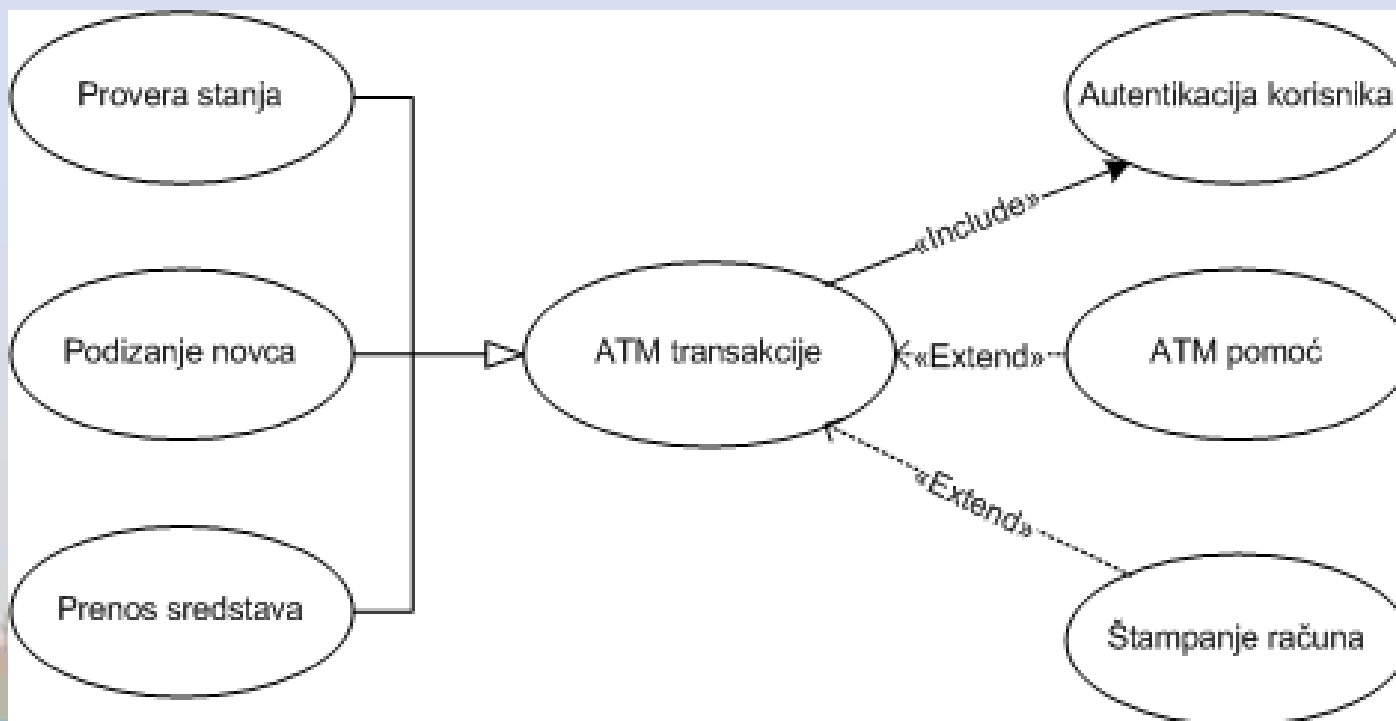
- ATM (*Automated Teller Machine*) sistem je podsistem bankarskog sistema koji omogućava klijentima banke obavljanje finansijskih transakcija na javnom mestu
- Kreirati **glavni (main use case)** i njemu odgovarajući **detaljniji use case model ATM sistema**
- ATM sistem omogućava:
  - Održavanje bankomata od strane operatera
  - Klijentima obavljanje finansijskih transakcija kojima upravlja bankarski sistem
  - Referent u banci održava bankomat, što uključuje punjenje, ažuriranje i dijagnosticiranje stanja
  - Inženjer na osnovu dijagnoze stanja vrši popravke ATM sistema
  - ATM transakcije koje su dostupne klijentima su: provera stanja, podizanje novca i prenos sredstava sa jednog na drugi račun
  - Dodatne opcije koje su na raspolaganju klijentima su ATM help i štampanje računa
  - ATM sistem zahteva obaveznu autentikaciju klijenata

# Glavni use case model ATM sistema



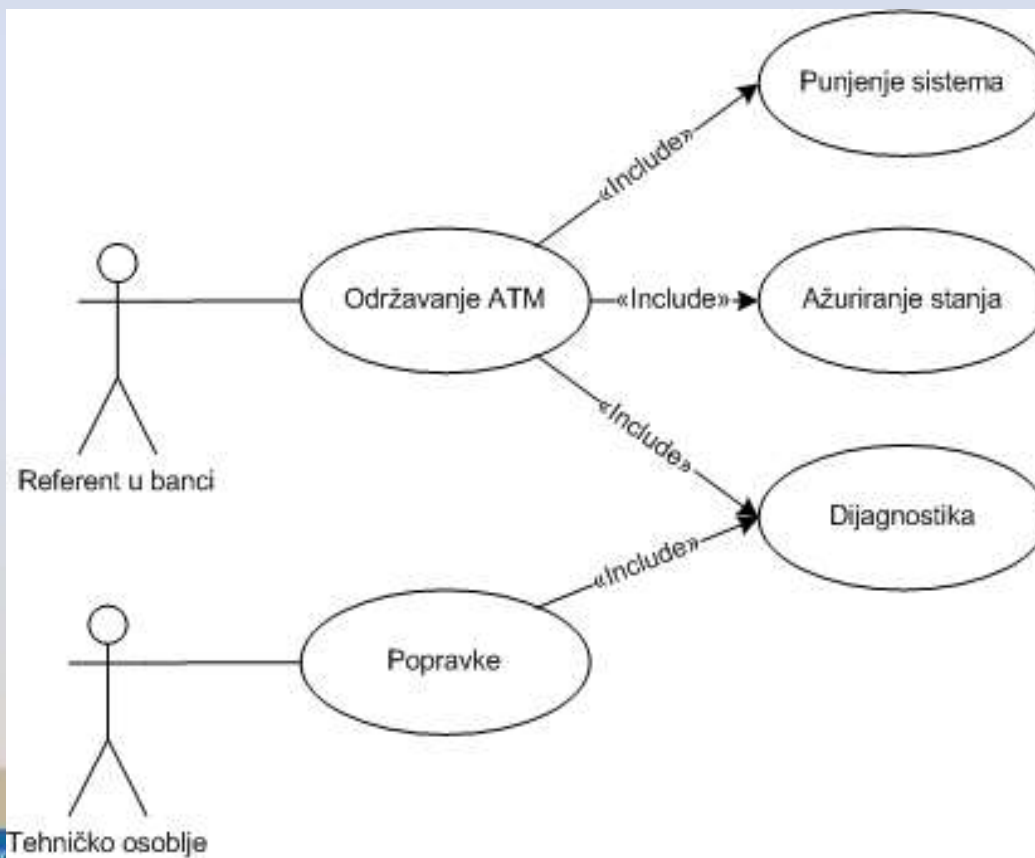
# Detaljniji use case model ATM sistema

- Use case model bankarskih ATM transakcija i autentikacije korisnika

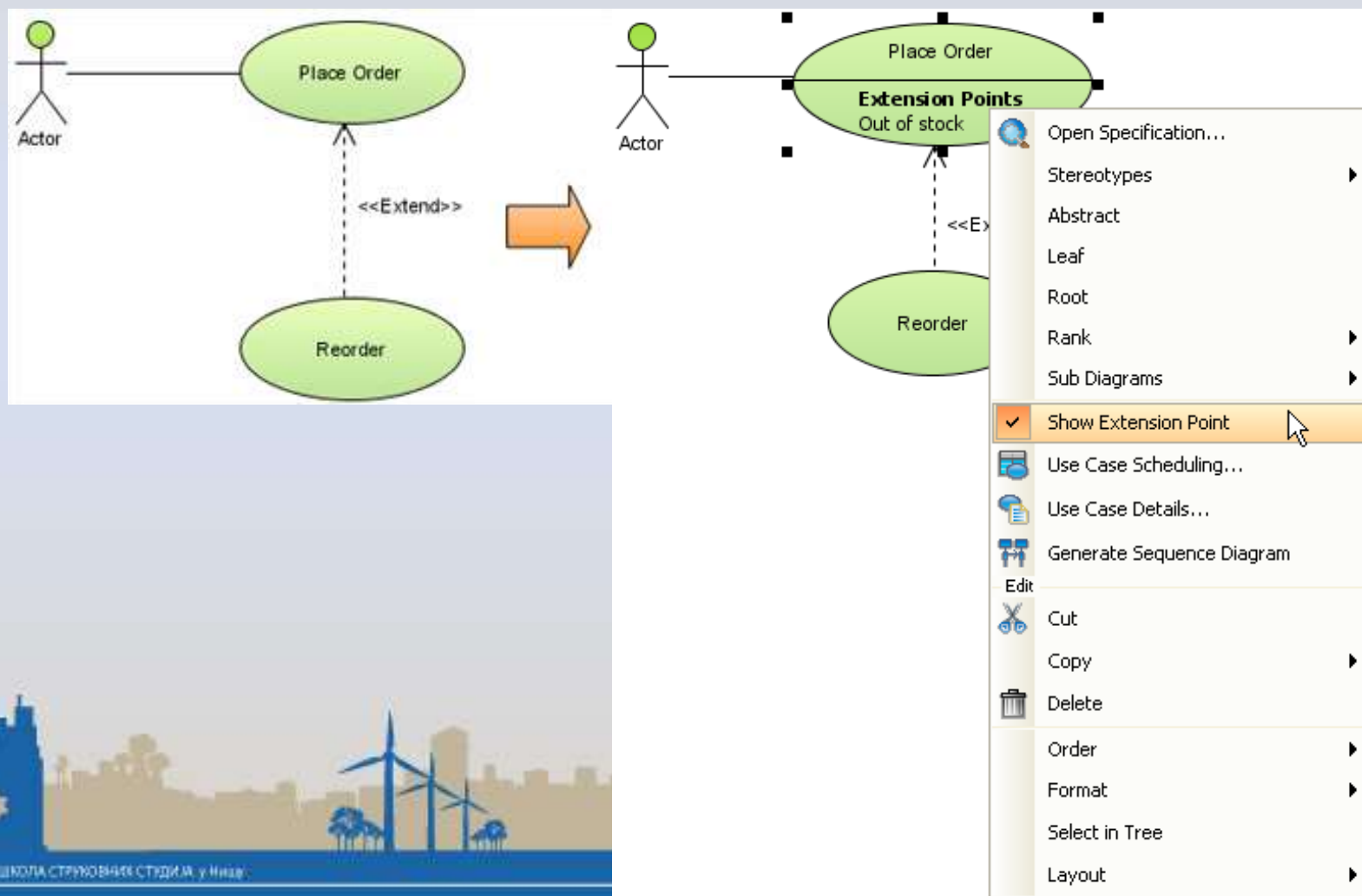


# Detaljniji use case model ATM sistema

- Use case model održavanja ATM



# Primer kod relacije Extend: *Extension Point*



## Sledeći korak u OOM: Detaljna specifikacija sistema

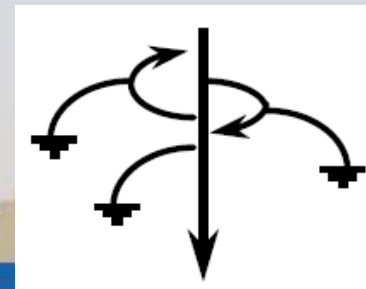
- Model slučajeve korišćenja predstavlja grubu specifikaciju sistema – opis sistema na višem nivou (sistemski pristup)
- Sledeći korak je dekompozicija sistema - u objektno-orijentisanom modelovanju (OOM) predstavlja se dijagramima koji detaljnije opisuju slučajeve korišćenja, njihove detaljne aktivnosti, scenarija, moguća ponašanja, interakcije itd.
- Ostali UML dijagrami prikazuju realizaciju konkretnih slučajeve korišćenja





# Opis slučaja korišćenja: SCENARIO

- Svaki slučaj korišćenja treba da bude detaljno opisan (realizacija use case-a)
- U početnoj fazi preporuka je da se koristi **struktuirani tekstualni opis** (Scrum okvir se bazira na korisničkim iskazima u pisanoj formi tzv. *user stories*)
- Obratiti pažnju da se prvo fokusira na **opis normalnog (osnovnog) toka događaja**, a potom na moguće **izuzetke ili alternativne tokove**
- Jedan slučaj korišćenja predstavlja **skup sekvenci događaja**
- Jedna sekvenca događaja se naziva **scenario**
- Postoji osnovni scenario i skup mogućih izuzetaka i alternativnih funkcionisanja



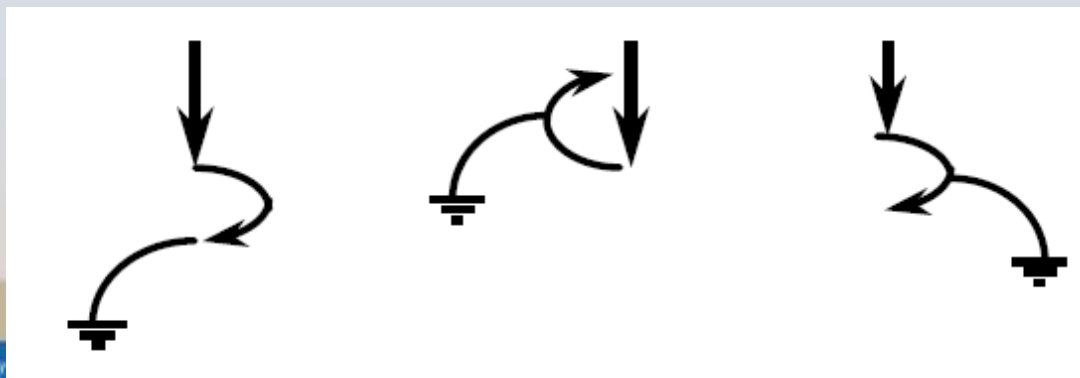
# Tok događaja nekog slučaja korišćenja

- Svaki use case se dokumentuje tokom događaja - sadržan u dokumentu koji se zove ***Use Case Specification***
- Tok događaja treba da ukaže na to **šta** sistem treba da uradi, a **ne kako** to sistem radi
  - Kada i kako use case počinje i završava se?
  - Kakvu interakciju ima use case sa akterima?
  - Koji podaci su potrebni use case-u?
  - Normalni niz događaja u samom slučaju korišćenja
  - Prikaz bilo kojih alternativnih ili izuzetnih tokova
- Dokumentacija toka događaja se pravi **iterativno**:
  - Prvo se napiše samo kratak opis potrebnih koraka u izvršavanju normalnog toka use case-a
  - Kako analiza napreduje, koraci se šire dodavanjem više detalja
  - Na kraju, dodaju se izuzetni tokovi



# Šta je scenario?

- Scenario je instanca use-case-a
  - To je jedan tok use case-a
  - Scenario može da uključi **osnovni** i **alternativne tokove**
- Koliko je potrebno scenarija?
  - Koliko god je potrebno da bi se bolje razumeo sistem koji se razvija
  - Neophodno je razraditi scenarija onih use-case-ova koji su **od velikog interesa ili visokog rizika**
  - Scenarija se mogu koristiti kako za razumevanje tako i za validaciju tokova događaja



# Osnovni scenario ATM sistema: Podizanje novca

- Osnovni scenario ATM sistema za podizanje novca uključuje sledeće funkcionalnosti:

## 1. Provera kartice

Klijent ubacuje karticu u automat. Automat čita karticu i proverava da li je prihvatljiva. Ako je prihvatljiva, zahteva se od klijenta da unese PIN.

## 2. Provera šifre

Klijent unosi PIN. ATM sistem šalje PIN na proveru bankarskom sistemu. Ako je šifra ispravna klijentu se prikazuje meni sa mogućim transakcijama.

## 3. Unos tipa transakcije

Klijent bira “podizanje novca” i automat prikazuje meni sa izborom mogućih novčanih iznosa ili opciju za unos željenog iznosa.

## 4. Podizanja novca

Klijent bira ili ukucava željeni novčani iznos. ATM sistem šalje bankarskom sistemu zahtev za podizanje datog iznosa sa računa. Bankarski sistem proverava da li iznos manji od postojećeg stanja na računu. Ako je u redu, pojavljuje se ekran koji pita klijenta da li želi da mu se odštampa račun o izvršenoj transakciji.

## 5. Kraj

Automat vraća karticu klijentu, izdaje zahtevani iznos i štampa račun

# Alternativna scenarija za podizanje novca

- **Kartica nije prihvatljiva**

Kartica se vraća klijentu sa zvučnim signalom

- **Pogrešan PIN kôd**

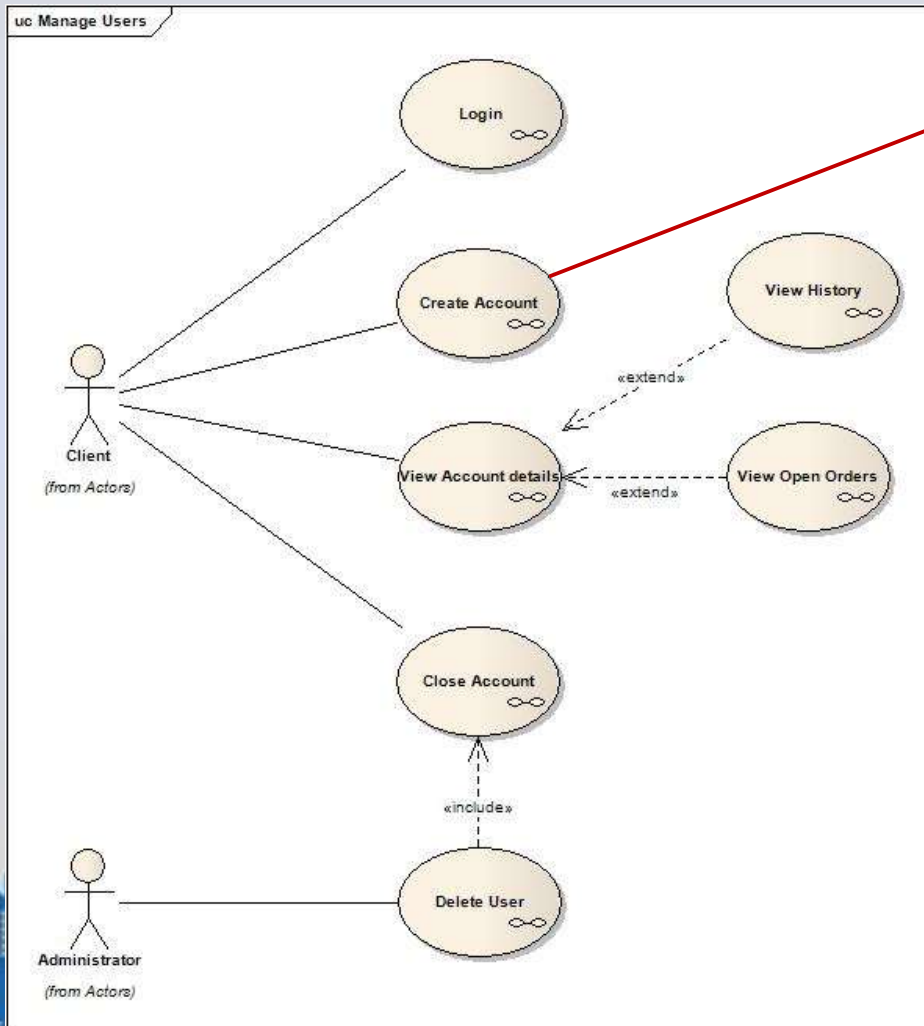
Prikazuje se poruka o grešci i daje se mogućnost ponovnog unosa. Klijent ima tri pokušaja, nakon čega mu se vraća kartica.

- **Prekid**

Klijent može u svakom trenutku da prekine transakciju. Poništiće se sve započete transakcije i vratiti kartica klijentu.



# Primer opisa scenarija za use case: Kreiranje računa (*Create Account*)



The screenshot shows the "UseCase: Create Account" dialog box. The "Scenario:" field is set to "Basic Path". The "Type:" dropdown is also set to "Basic Path". The scenario description is as follows:

- 1) Use case begins when the Client selects the "Create Account" command.
- 2) The Client enters name, address, e-mail address, password and password confirmation.
- 3) Client presses Submit button.
- 4) Submitted details are validated.
- 5) Processing message is displayed to the Client.
- 6) New account is created in the database.
- 7) Success message is displayed to Client.

Below the description is a table of scenarios:

| Name                      | Type       |
|---------------------------|------------|
| Basic Path                | Basic Path |
| No Submit                 | Alternate  |
| Validation fails          | Alternate  |
| Cannot Create New Account | Alternate  |

# Primer detaljne specifikacije use case-a “Odabir kurseva koji će predavati”

## 1.0. Naziv use case-a (*Use case name*)

Odabir kurseva koji će predavati

### 1.1. Kraći opis use case-a (*Brief Description*)

Ovaj use case pokreće Profesor. Omogućava da profesor izabere do četiri kursa koja će predavati tokom izabranog semestra.

## 2.0. Tok događaja (*Flow of Events*)

### 2.1. Osnovni tok (*basic flow*)

#### 2.1.1. Prijava na sistem

Ovaj use case počinje kada se profesor **prijavi na sistem za registraciju i unese svoju lozinku.**

Sistem verifikuje da je lozinka valjana (ukoliko lozinka nije valjana izvršava se alternativni tok 2.2.1)



# Detaljna specifikacija (nastavak)

## 2.1.2. Unos tipa transakcije: Izbor semestra

Sistem signalizira da profesor izabere tekući ili budući semestar (ukoliko se unese pogrešan semestar, izvršava se alternativni tok 2.2.2).

Profesor unosi željeni semestar. Sistem signalizira da profesor izabere željenu aktivnost: add, delete, review, print ili quit.

## 2.1.3. Izbor aktivnosti: Add

Ukoliko je izabrana aktivnost Add, sistem prikazuje ekran sa poljem za unos naziva i broja kursa.

Ukoliko profesor unese nevažeću kombinaciju ime/broj, izvršava se alternativni tok 2.2.3). Sistem prikazuje nuđene opcije za taj kurs (ako ime kursa ne može da se prikaže izvršava se alternativni tok 2.2.4).

Profesor bira neku od ponuda. Sistem povezuje tog profesora sa izabranom ponudom (ukoliko veza ne može da se napravi, izvršava se alternativni tok 2.2.5).





# Detaljna specifikacija (nastavak)

## 2.1.4. Izbor aktivnosti: Delete

Ukoliko je izabrana opcija Delete, sistem prikazuje ekran sa nuđenim opcijama za taj kurs sa poljem za ime opcije i za broj. Profesor unosi ime i broj opcije (ako se unese nevažeća kombinacija ime/broj, izvršava se alternativni tok 2.2.3). Sistem uklanja vezu sa tim profesorom (ako veza ne može da se ukloni, izvršava se alternativni tok 2.2.6).

## 2.1.5. Izbor aktivnosti: Review

Ukoliko je izabrana aktivnost Review, sistem poziva (ako informacija o kursu ne može biti pozvana, izvršava se alternativni tok 2.2.7) i prikazuje sledeće informacije o svim opcijama kurseva koje su njemu dodeljene: ime kursa, broj kursa, broj opcije kursa, dane u nedelji, vreme i lokaciju.

## 2.1.6. Izbor aktivnosti: Print

Ukoliko je izabrana aktivnost Print, sistem štampa profesorov raspored (ukoliko raspored ne može da bude odštampan, izvršava se alternativni tok 2.2.8).

## 2.1.7. Izbor aktivnosti: Quit

Ukoliko je izabrana aktivnost Quit, use case se završava



# Alternativni tokovi

## 2.2 . Alternativni tokovi

- 2.2.1 **Nevažeća lozinka** - Unešena je nevažeća lozinka. Korisnik može ponovo da je unese ili da okonča use case.
- 2.2.2 **Nevažeći semestar** - Sistem obaveštava korisnika da je semestar nevažeći. Korisnik može ili ponovo da unese semestar ili da okonča use case
- 2.2.3 **Nevažeće ime/broj kursa** - Sistem obaveštava korisnika da je nevažeće ime/broj kursa. Korisnik može ponovo da unese valjanu kombinaciju ili da okonča use case
- 2.2.4 **Opcija kursa ne može da se prikaže** - Korisnik biva informisan da ta opcija nije trenutno dostupna. Use case ponovo počinje.
- 2.2.5 **Ne može da se uspostavi veza između profesora i opcije kursa** - Ova informacija je zapamćena i sistem će kasnije uspostaviti tu vezu
- 2.2.6 **Veza između profesora i opcije kursa ne može da se ukloni** - Ova informacija ostaje zapamćena i sistem će kasnije ukloniti tu vezu
- 2.2.7 **Ne može da se dobije informacija o rasporedu** - Korisnik se obaveštava da ova opcija trenutno nije dostupna
- 2.2.8 **Raspored ne može da se odštampa** - Korisnik se obaveštava da ova opcija trenutno nije dostupna



# Detaljna specifikacija (nastavak)

## **3.0. Specijalni zahtevi** (*Special Requirements*)

Za ovaj use case ne postoje posebni zahtevi

## **4.0. Preduslovi** (*Preconditions*)

4.1. Pre nego što se pokrene ovaj use case mora biti završen podtok Kreiranje ponude kursa koji se nalazi u okviru use case-a Održavanje informacija o kursevima

## **5.0. Post stanja** (*Post Conditions*)

Ne postoje krajnji uslovi

## **6.0. Extension Points**

Ne postoje tačke proširivanja

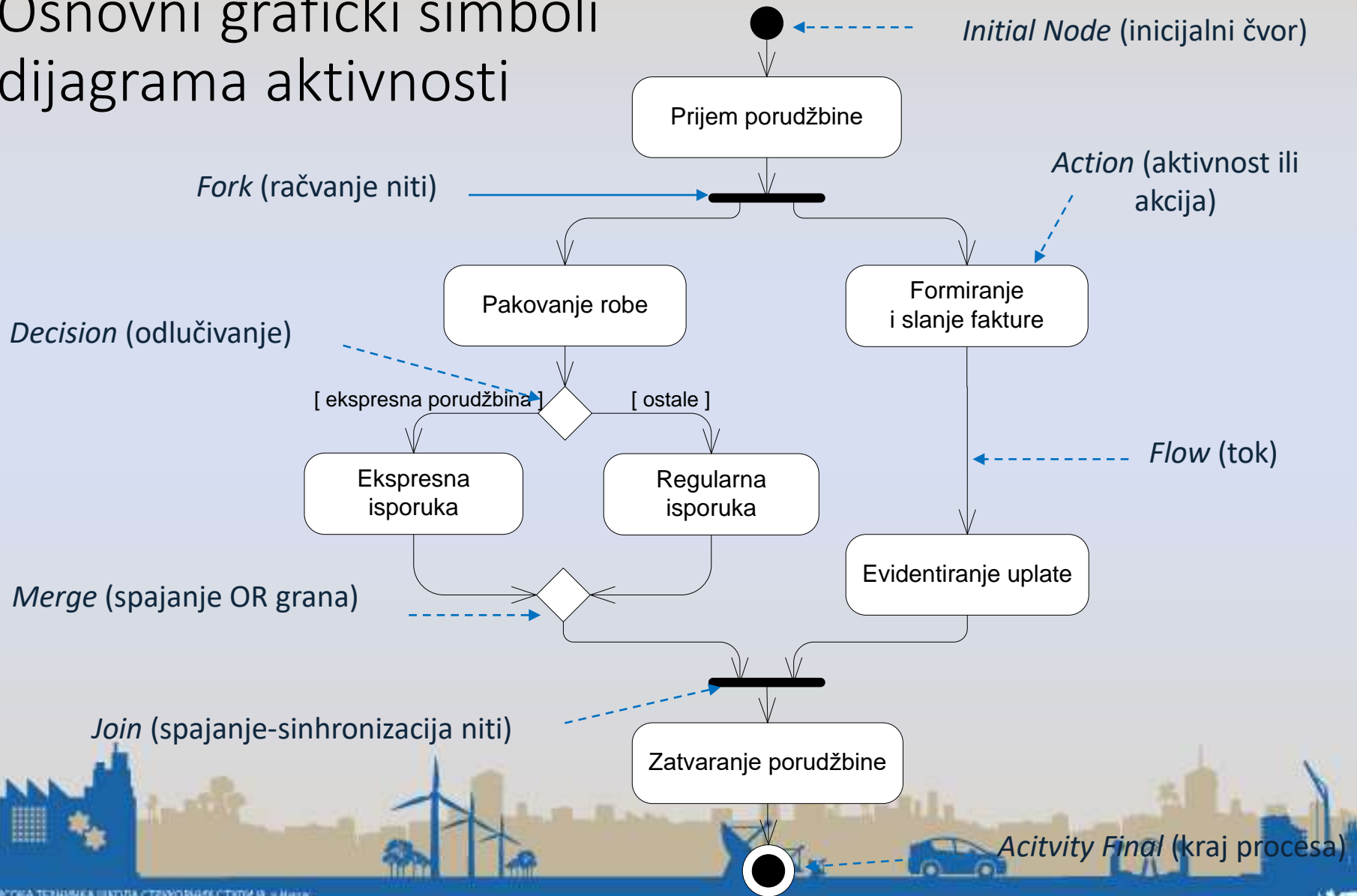


# Dijagrami aktivnosti

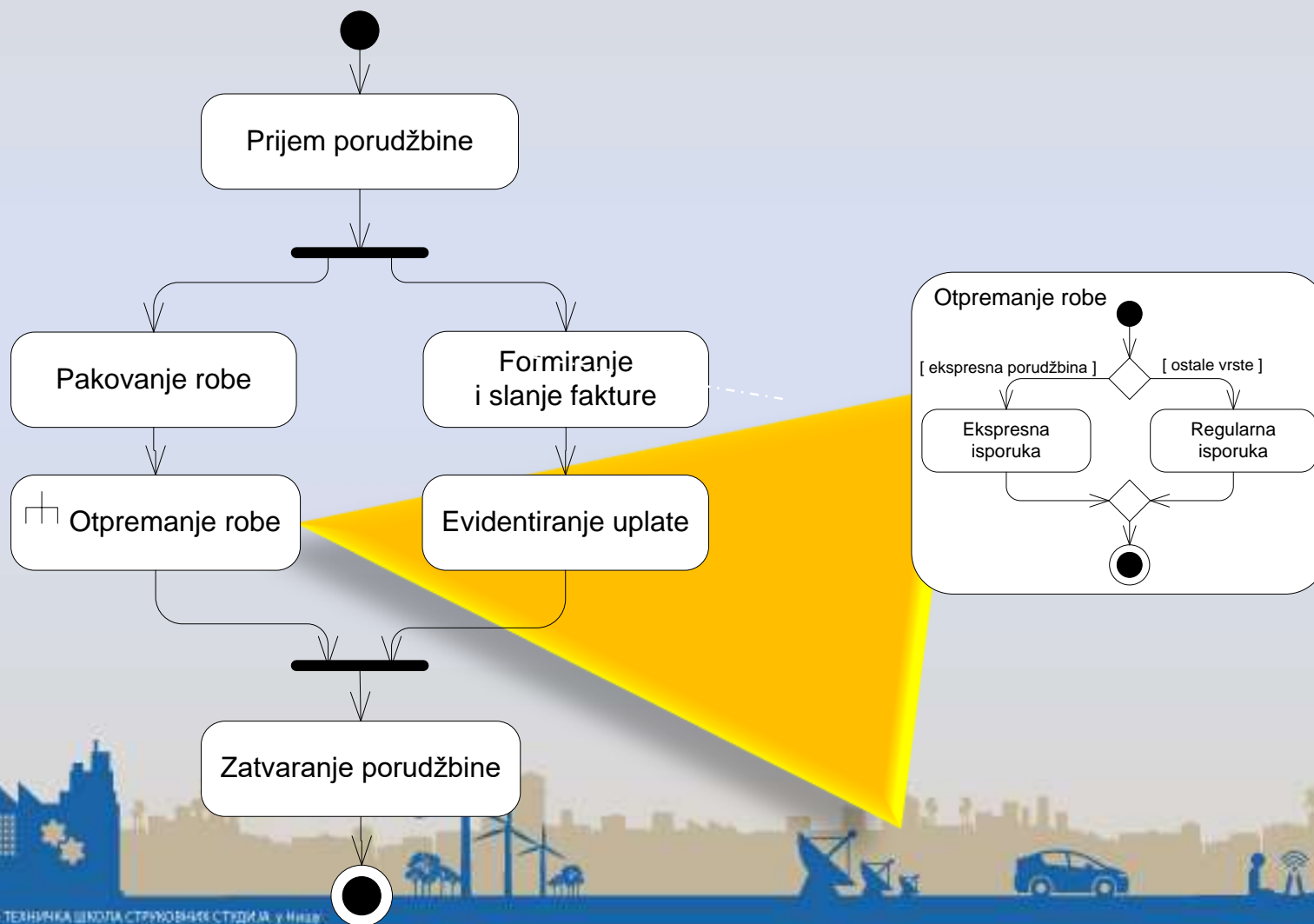
- Detaljna specifikacija use case modela se grafički predstavlja dijagramom aktivnosti
- Dijagram aktivnosti se koristi na najrazličitijim nivoima projektovanja softvera i to za opis:
  - Poslovnih procesa
  - *Workflow*-a
  - Složenih slučajeva korišćenja (*use-case*)
  - Proceduralne logike
  - Algoritama



# Osnovni grafički simboli dijagrama aktivnosti

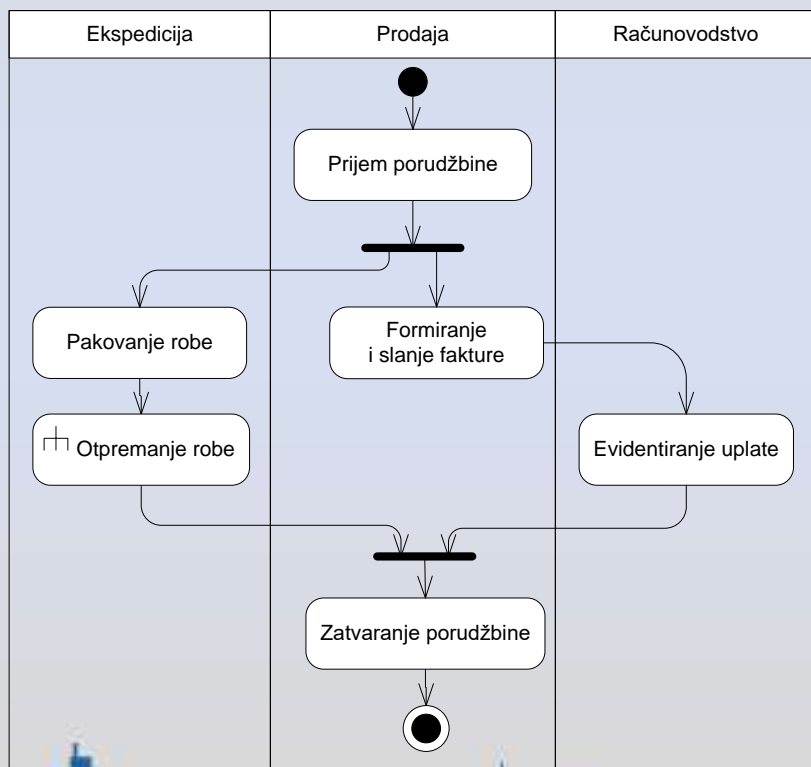


# Podprocesi (*Subactivities*)



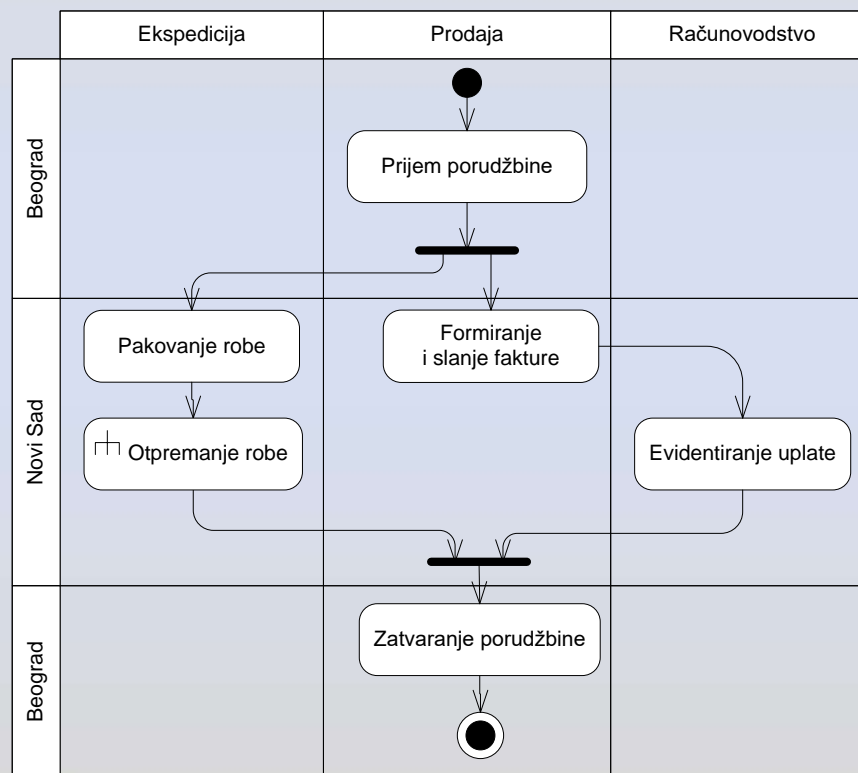
# Particije (*Partitions*)

- **Particije** predstavljaju aktere (učesnike) u modelu odvijanja poslovnog procesa



# Particije (*Partitions*)

- Dvodimenzionalni grid (particije po kolonama i redovima)
- Particije se mogu deliti na podparticije kako bi se prikazala hijerarhija particija





# Tokovi (*Flows*)

- Kontrolni tok (*Control flow*)

Prijem fakture

Plaćanje fakture

- Tok objekta (*Object flow*)

- Objekat predstavlja dokument koji se prosleđuje od jedne do druge aktivnosti

Prijem fakture

Nalog za plaćanje

Plaćanje fakture

- Skladište (*Datastore*) – skladište podataka (dokumenata)

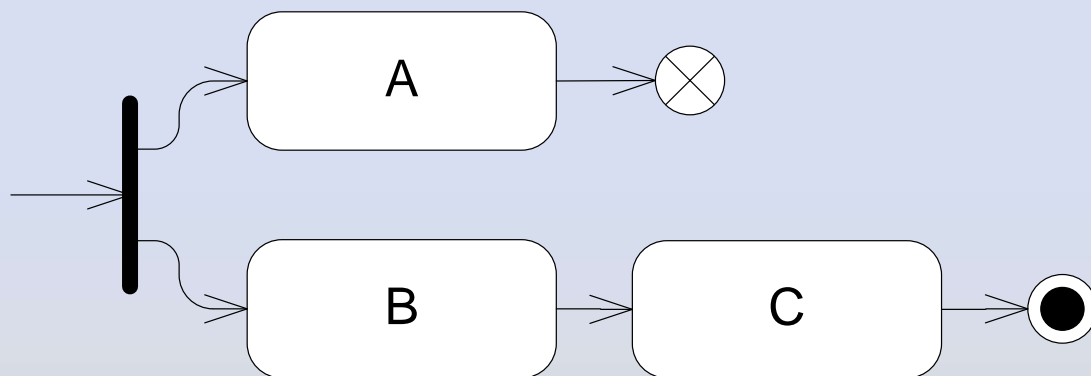
Prijem otpremnice

Formiranje prijemnice

<<datastore>>  
Otpremnice

## Kraj toka (*Flow final*)

- Označava kraj jednog toka/niti ili završetak celog procesa

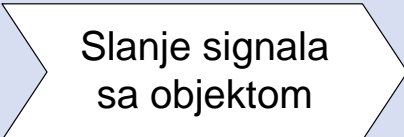


# Događaji - Signali (*Signals*)

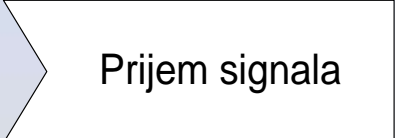
- Slanje signala (*send signal*)
- Slanje signala sa priključenim objektom (*send object*)
- Prijem signala (*receive signal*)
- Vremenski okidač (*time signal*)



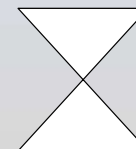
Slanje signala



Slanje signala  
sa objektom

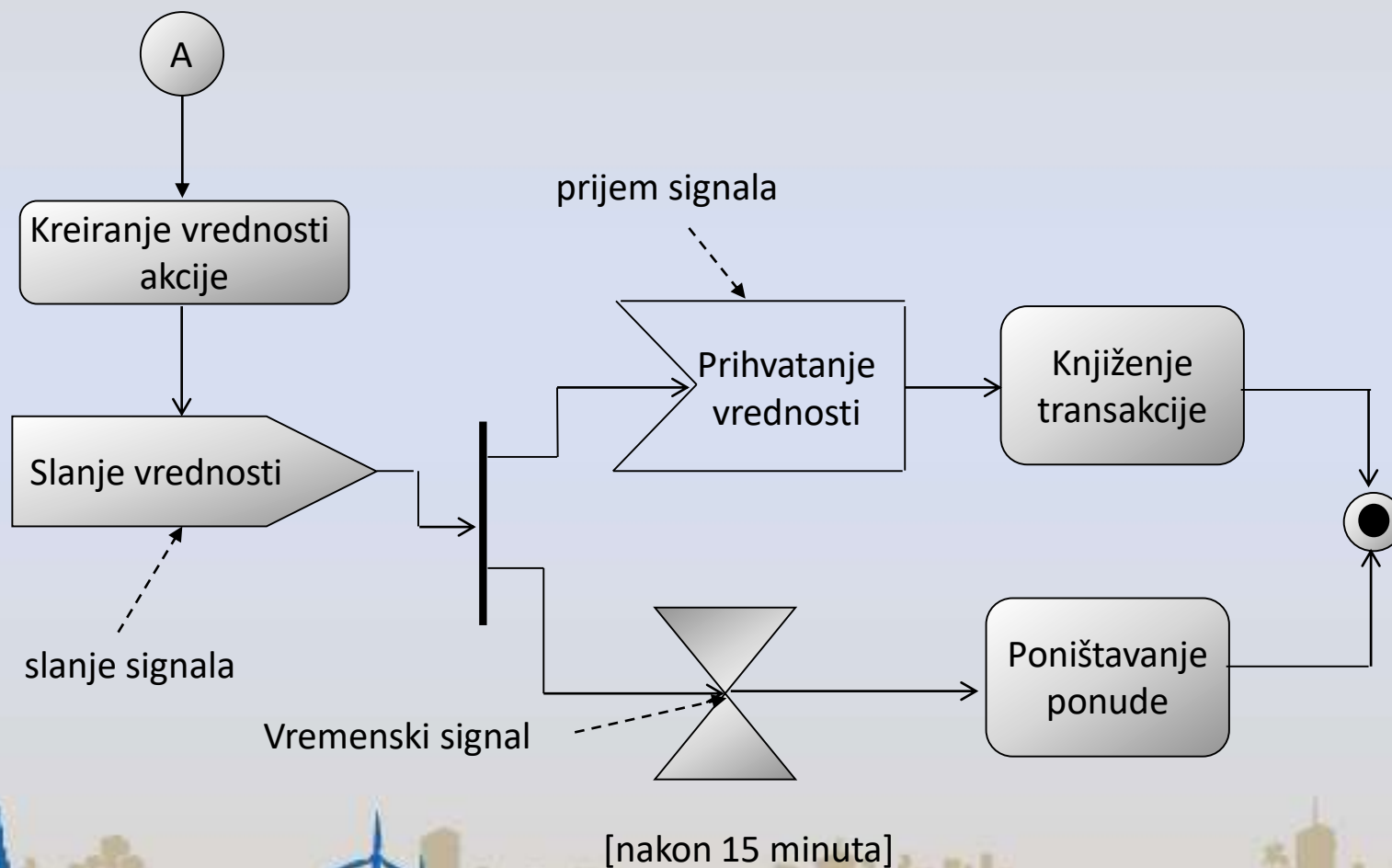


Prijem signala

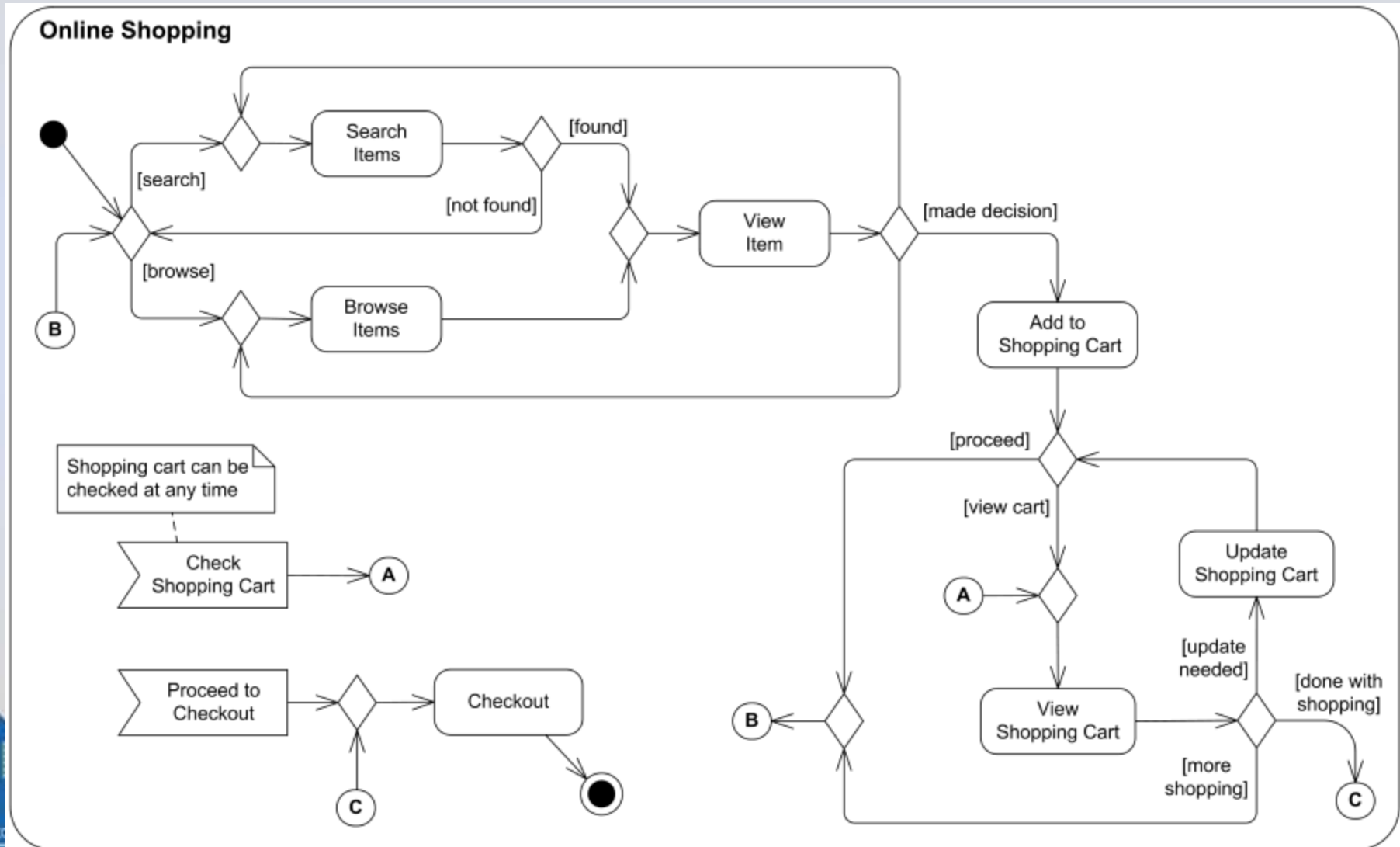


[uslov okidanja  
vremenskog događaja]

# Primer signala

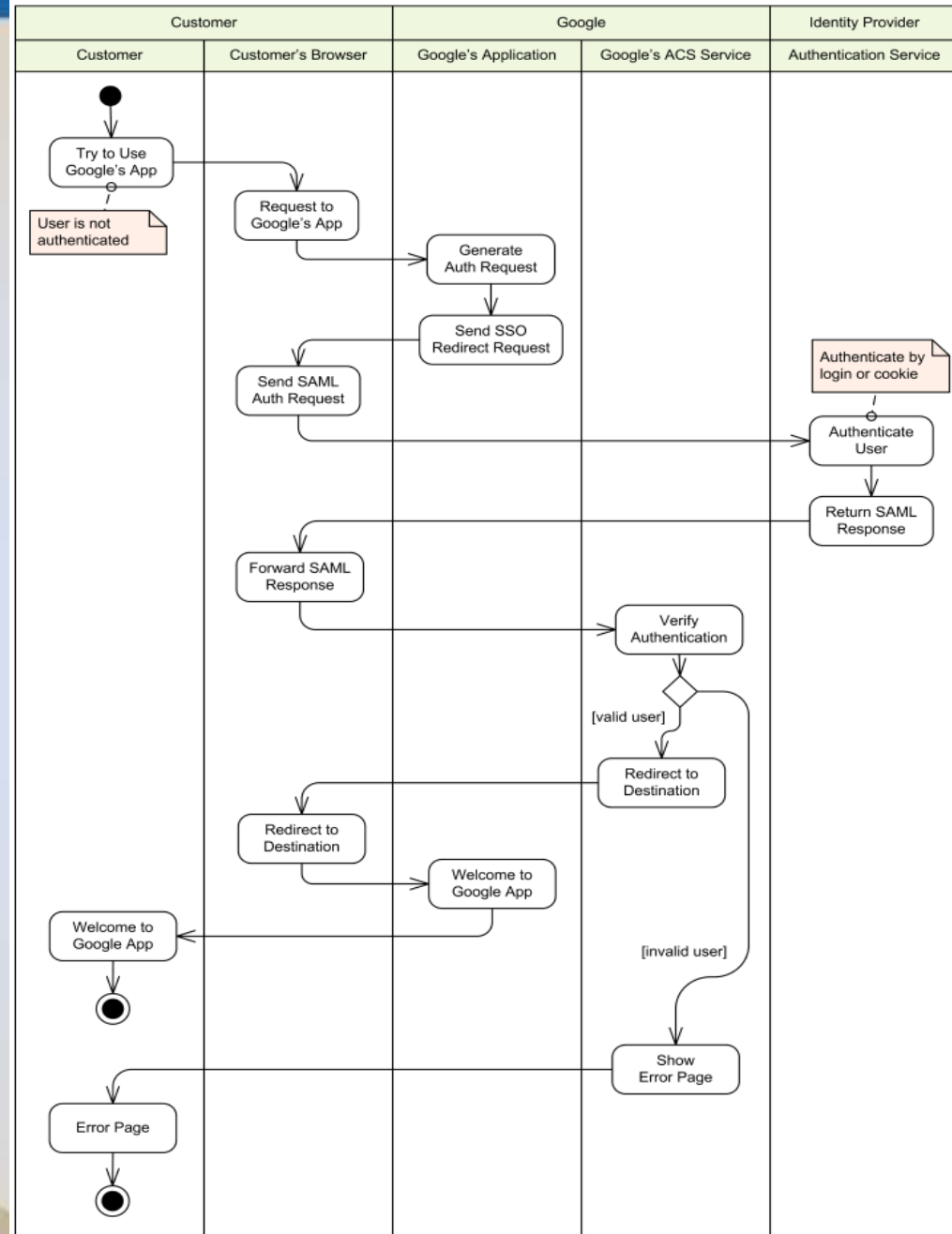


# Primer dijagrama aktivnosti: Online kupovina



# Primer *Single Sign-On* za *Google Apps*

- Da bi komunicirao sa svojim partnerima Google kompanija koristi Single Sign-On (SSO) na OASIS SAML 2.0 protokolu
- Partnerske kompanije (provajderi servisa identiteta) kontrolišu korisnička imena, šifre i druge informacije koje se koriste za identifikaciju, autentikaciju i autorizaciju korisnika za web aplikacije koje Google hostuje
- Svaki partner obezbeđuje Google sa URL-om njegovog SSO servisa kao i javne ključeve koje će Google koristiti za verifikaciju SAML odziva
- Odziv je prosleđen Google Assertion Consumer Service (ACS) koji verifikuje SAML odziv koristeći partnerov javni ključ. Ukoliko je odziv validan i identitet korisnika potvrđen od strane provajdera identiteta, ACS preusmerava korisnika na željeni URL, u suprotnom javlja grešku.



# Zadatak 1:

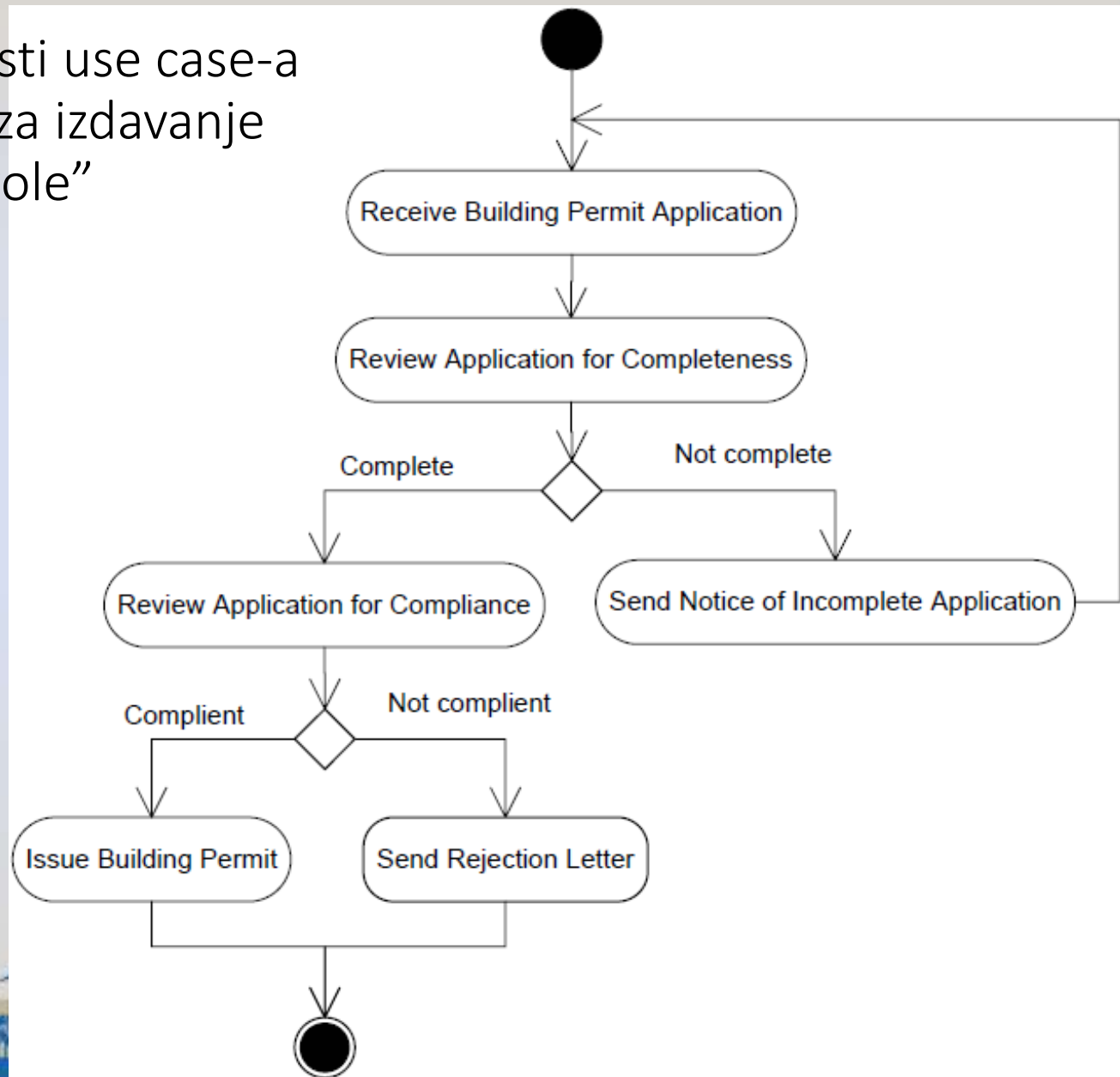
Kreirati dijagram aktivnosti za use case:

“Analiza zahteva za izdavanje građevinske dozvole”

- Proces analize zahteva za izdavanje građevinske dozvole prolazi kroz sledeće aktivnosti:
  - Prihvatanje pristiglih zahteva
  - Pregled zahteva radi utvrđivanja da li je dostavljena kompletna dokumentacija
  - Ukoliko nije, šalje se obaveštenje o nekompletnoj dokumentaciji
  - Ukoliko jeste, pregleda se sadržaj dokumentacije radi dobijanja saglasnosti
  - Ukoliko je dobijena saglasnost izdaje se građevinska dozvola
  - Ukoliko nije dobijena saglasnost šalje se obaveštenje o odbijanju zahteva



Dijagram aktivnosti use case-a  
“Analiza zahteva za izdavanje  
građevinske dozvole”





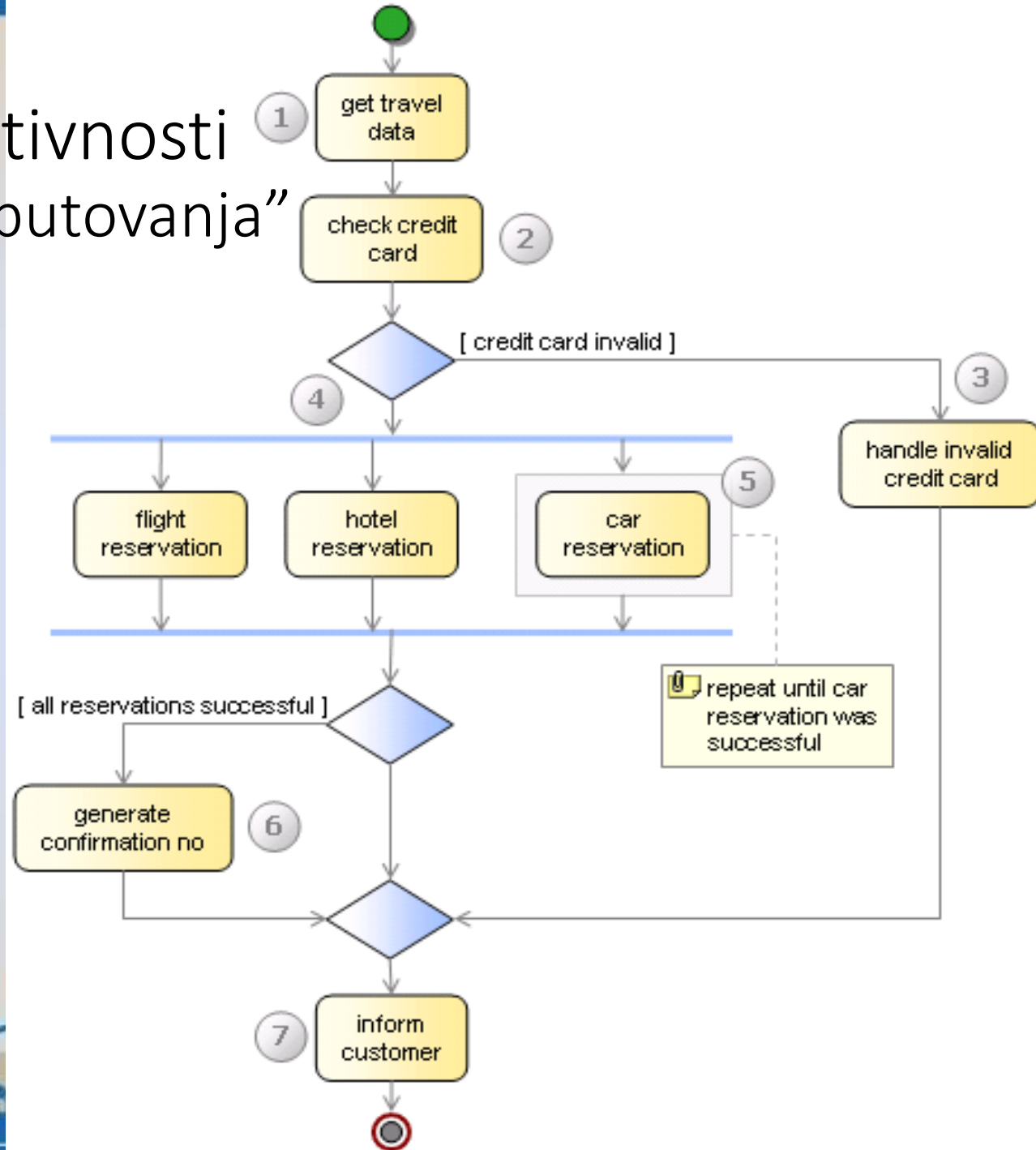
## Zadatak 2:

Kreirati dijagram aktivnosti za rezervaciju putovanja  
(*Travel booking*)

1. Proces rezervacije putovanja počinje kada Internet klijent unese podatke za konkretan aranžman
2. Proverava se ispravnost i validnost podataka kreditne kartice
3. Ukoliko su podaci na kreditnoj kartici nepotpuni ili ne važeći, proces se završava slanjem informacije o nevalidnosti kreditne kartice
4. Ukoliko su podaci o kreditnoj kartici ispravni, vrše se potrebne rezervacije letova, hotela i rentacar-a
5. Proces rezervacije se ponavlja sve dok rezervacije nisu uspešno završene
6. Ukoliko su sve rezervacije uspešno izvršene, generiše se broj potvrde
7. Poslednja aktivnost je Informisanje klijenta o rezultatima prethodnih aktivnosti



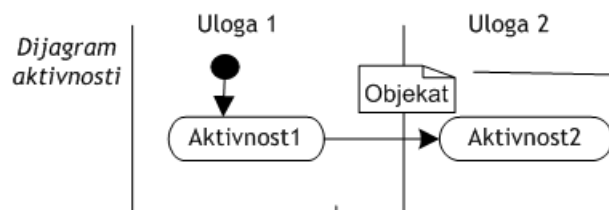
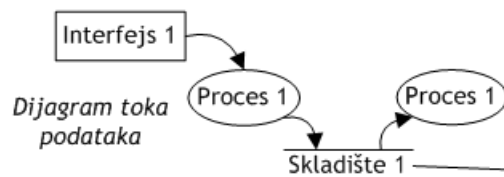
# Dijagram aktivnosti “Rezervacije putovanja”



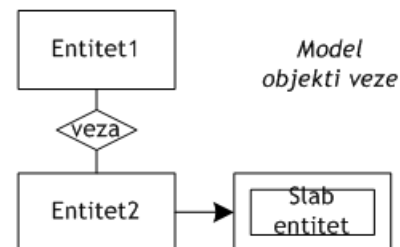
# Aspekti modelovanja

## Analiza poslovnog sistema

### Modelovanje poslovnih procesa

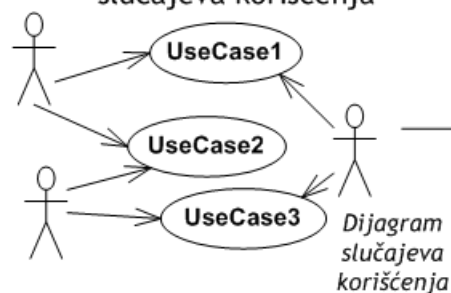


### Konceptualno modelovanje



## Specifikacija korisničkog zahteva

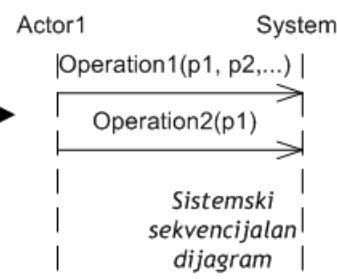
### Definisanje slučaja korišćenja



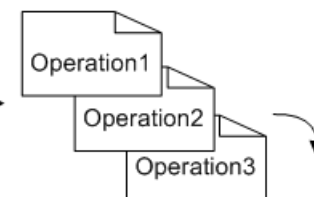
### Opis slučaja korišćenja



### Sistemske operacije



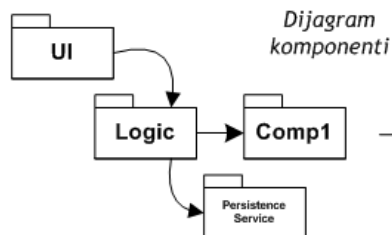
### Ugovori sistemskih operacija



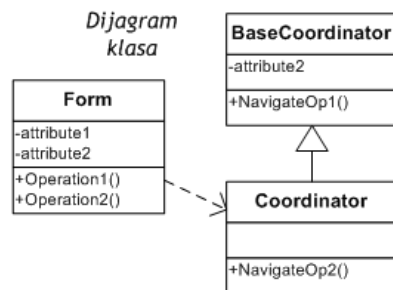
# Aspekti modelovanja (nastavak)

## Analiza i projektovanje

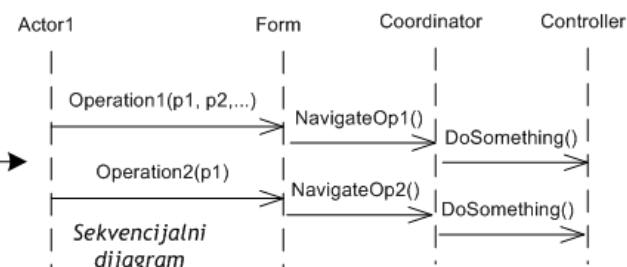
### Logička arhitektura



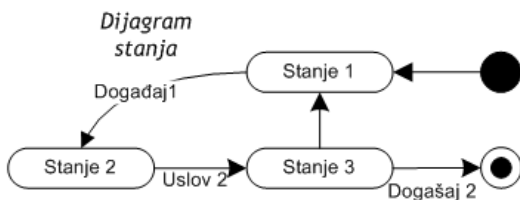
### Odnos klasa



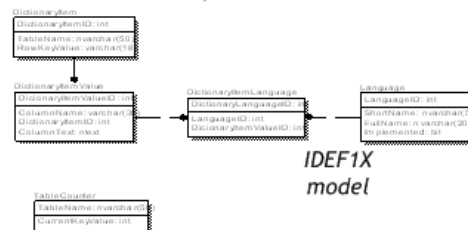
### Interakcija objekata



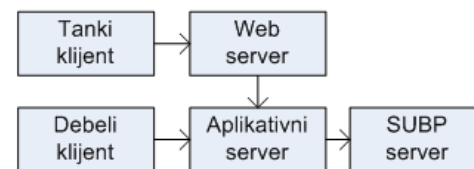
### Stanja objekata



### Model podataka



### Fizička arhitektura



## Implementacija

### Izvorni kod

```
public class Form
{
    public tip attribute1
    public tip attribute2

    public tip Operation1
    {
        ...
    }
}
```

```
public class Coordinator:
    BaseCoordinator
{
    public tip attribute2

    public tip NavigateOp1
    {
        ...
    }
}
```

### SQL kod

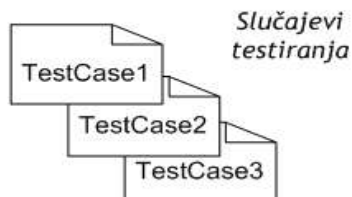
```
CREATE TABLE [dbo].[Country]
(
    [CountryID] [int] NOT NULL,
    [CountryName] [nvarchar](150) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL,
    [CountryShortName] [char](3) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL,
    CONSTRAINT [PK_Country] PRIMARY KEY CLUSTERED
(
    [CountryID] ASC
) WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
```

### Konfiguracija

```
<configSections>
  <section name="loggingConfiguration
PublicKeyToken=null" />
  <section name="dataConfiguration" />
  <sectionGroup name="applicationSettings">
    <section
name="DBTest.Properties.Settings">
  </sectionGroup>
  <section
name="securityCryptographyConfiguration">
</configSections>
```

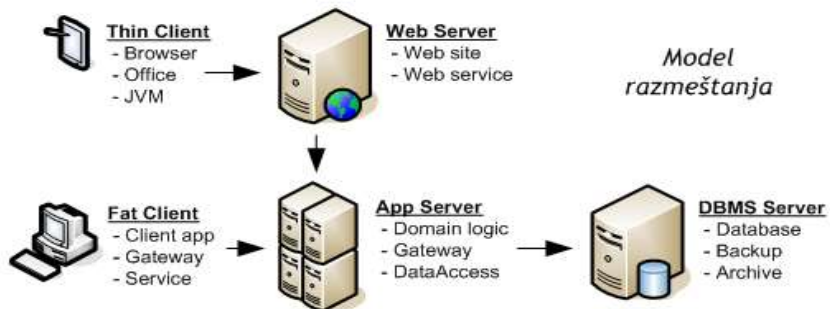
## Testiranje

### Planiranje testiranja



## Razmeštanje

### Planiranje instalacije



### Instalacija

#### Instalaciona procedura



#### Konfiguracioni skript

```
<configSections>
  <section name="loggingConfiguration
PublicKeyToken=null" />
  <section name="dataConfiguration" />
  <sectionGroup name="applicationSettings">
    <section
name="DBTest.Properties.Settings">
  </sectionGroup>
  <section
name="securityCryptographyConfiguratio
n"/>
```

#### Setup

