

Fakultet tehničkih nauka, Novi Sad

Inženjering informacionih sistema

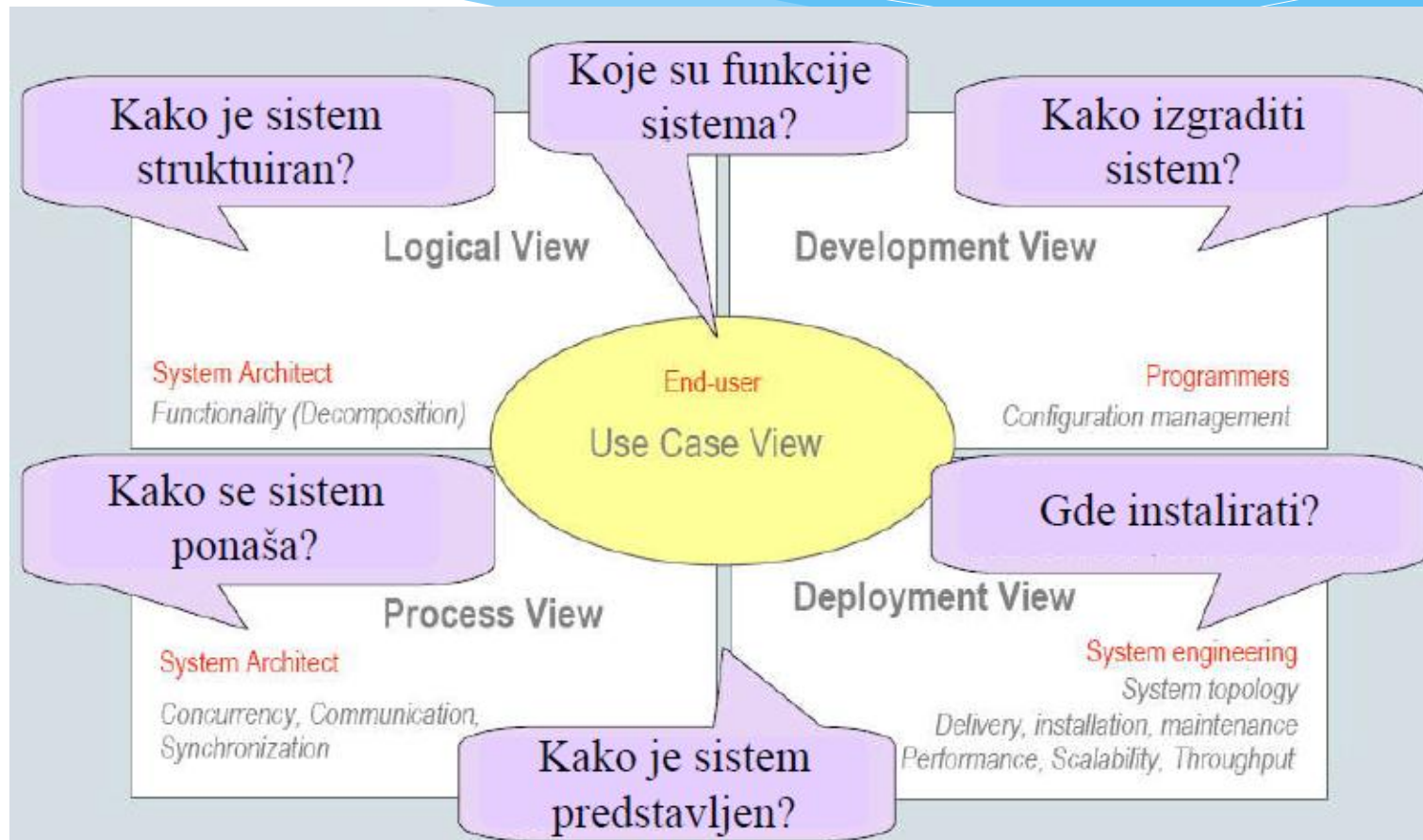
Dr Ivan Luković
Dr Slavica Kordić
Nikola Obrenović
Milanka Bjelica

Dr Jelena Borocki
Dr Milan Delić

UML

- * **UML (Unified Modeling Language)** – objedinjeni vizuelni jezik za poslovno i softversko modelovanje u svim fazama razvoja i za sve tipove sistema, kao i za generalno modelovanje kojim se definišu statičke strukture i dinamičko ponašanje.
- * Standardni jezik za:
 - * vizuelizaciju,
 - * specifikaciju,
 - * konstruisanje i
 - * dokumentovanje softverskih sistema.

Arhitektura softverskih sistema



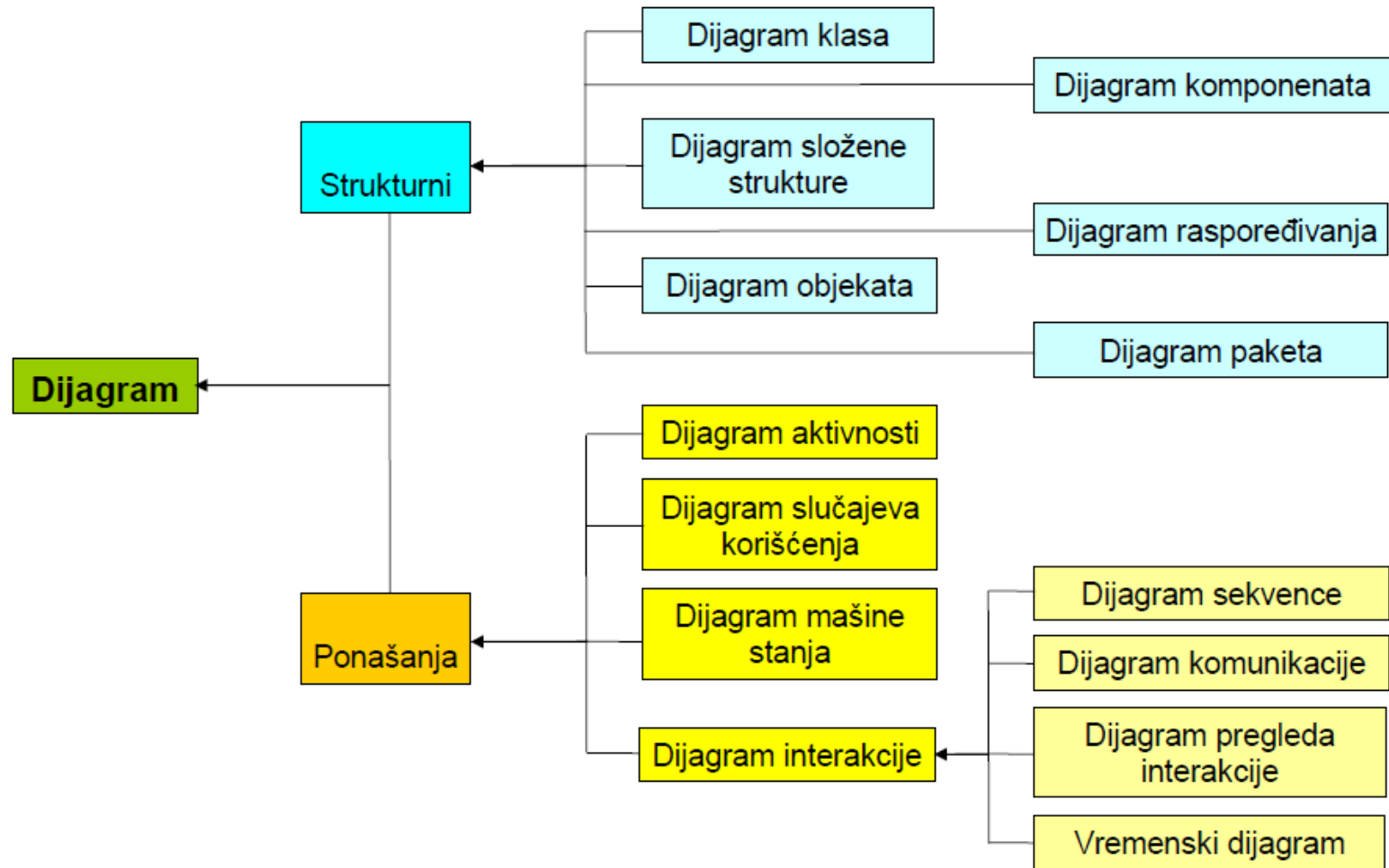
Kategorije korisnika

- * UML koriste sledeće kategorije korisnika:
 - * **Sistem analitičari i krajnji korisnici** – specifikacija zahtevane strukture i ponašanje sistema,
 - * **Arhitekta sistema** – projektanti sistema koji će zadovoljiti zahteve,
 - * **Razvojni inženjeri (developers)** – transformišu arhitekturu u izvršni kod,
 - * **Kontrolori kvaliteta** – provera strukture i ponašanja sistema,
 - * **Rukovodioci projekta (managers)** - vode i usmeravaju kadrove i resurse.

UML dijagrami

- * Dijagrami u UML-u – grafička predstava skupa elemenata – iscrtan kao graf čvorova (stvari) i lukova (relacija).
- * Dijagrami UML-a prikazuju sistem iz više uglova.
- * **Strukturni dijagrami** ističu šta treba modelovati u sistemu.
- * **Dijagrami ponašanja** ističu šta se dešava u sistemu koji se modeluje.
- * **Dijagrami interakcija**, kao podskup dijagrama ponašanja, prikazuju tok kontrola i podataka u sistemu.

UML dijagrami

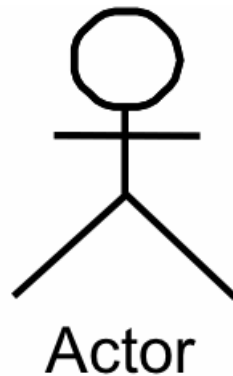


Dijagrami slučajeve upotrebe (Use-Case Diagram)

- * Spadaju u dijagrame ponašanja.
- * Pogled korisnika na funkcionisanje sistema (šta sistem radi, a ne kako sistem funkcioniše).
- * Razvoj dijagrama slučajeve upotrebe definiše se sledećim aktivnostima:
 - * Definisanjem učesnika
 - * Definisanjem slučajeve upotrebe
 - * Definisanjem tipova veza između učesnika i slučajeve upotrebe
 - * Izradom dijagrama slučajeve upotrebe.

Dijagrami slučajeve upotrebe – definisiranje učesnika

- * Korisnik je čovek koji koristi sistem, dok je učesnik specifična uloga koju korisnik ima u komunikaciji sa sistemom.
- * Učesnik je osoba ili veštački entitet (softver ili sistem) koji učestvuje u slučaju upotrebe.



Dijagrami slučajeva upotrebe – definisanje slučajeva upotrebe

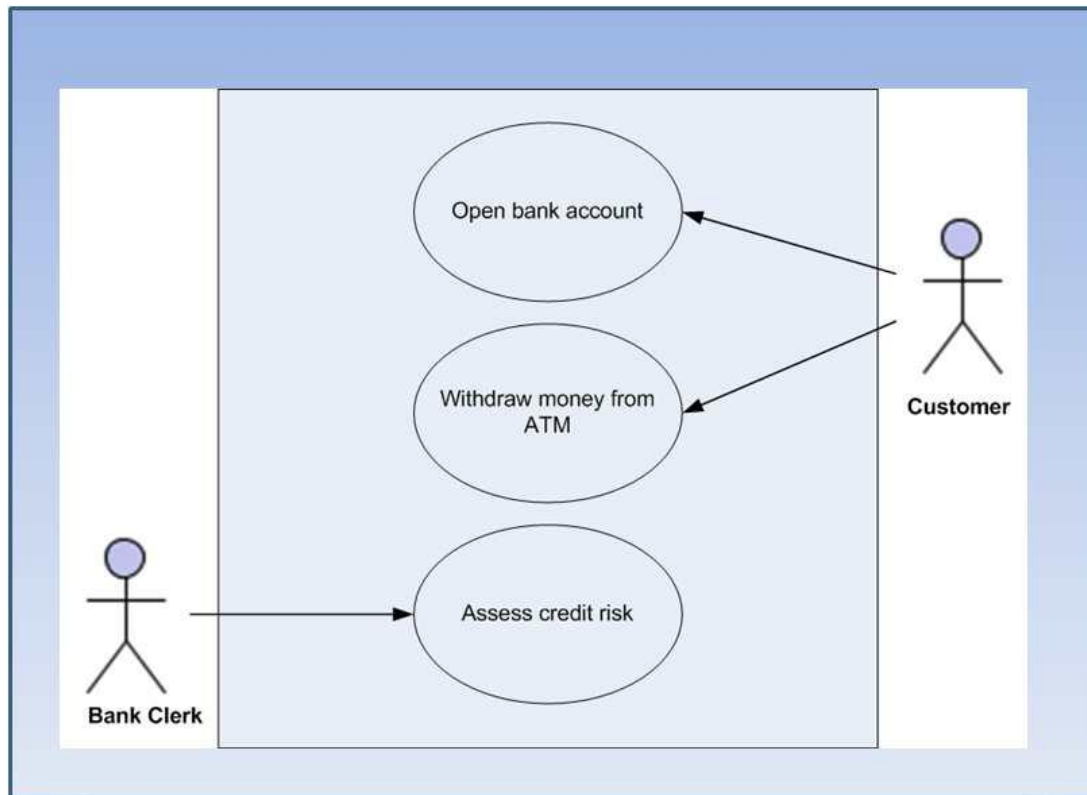
- * **Slučaj upotrebe – definiše funkcionalnost sistema sa stanovišta učesnika – šablon ponašanja delova sistema.**
- * **Za svaki slučaj upotrebe potrebno je opisati sledeće:**
 - * **Osnovni (uspešni) tok događaja – kada se započinje izvršavanje i ko inicira izvršavanje**
 - * **Alternativni tok događaja, ukoliko ima proširenja**
 - * **Neuspešni tok događaja**
 - * **Navođenje potrebnih preduslova da bi se uopšte došlo do izvršavanja i**
 - * **Rezultati uspešnog i/ili neuspešnog izvršavanja.**

Dijagrami slučajeva upotrebe – definisanje veza između učesnika i slučajeva upotrebe

- * **Veze koje se uspostavljaju u dijagramu slučajeva upotrebe:**
 - * **Asocijacija**
 - * Asocijacija između slučajeva upotrebe tipa <<include>>
 - * Asocijacija između slučajeva upotrebe tipa <<extend>>
 - * **Generalizacija**

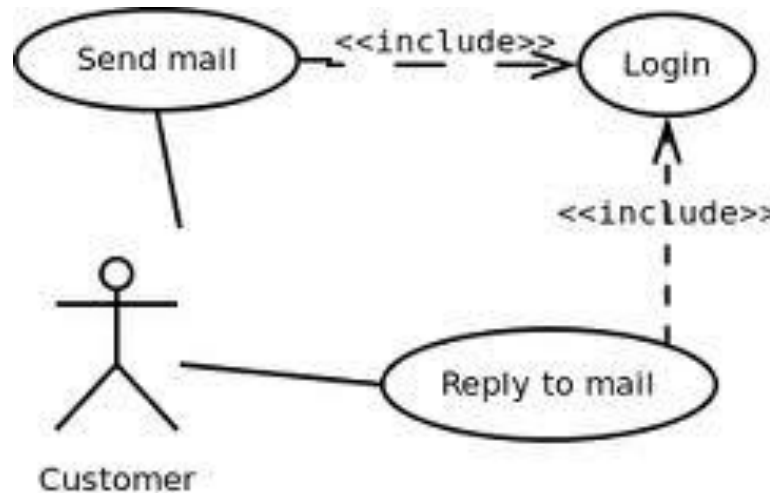
Dijagrami slučajeve upotrebe - asocijacija

- * Bidirekciona veza – linija koja spaja učesnike i slučajeve upotrebe.



Dijagrami slučajeja upotrebe – upotreba tipa <<include>>

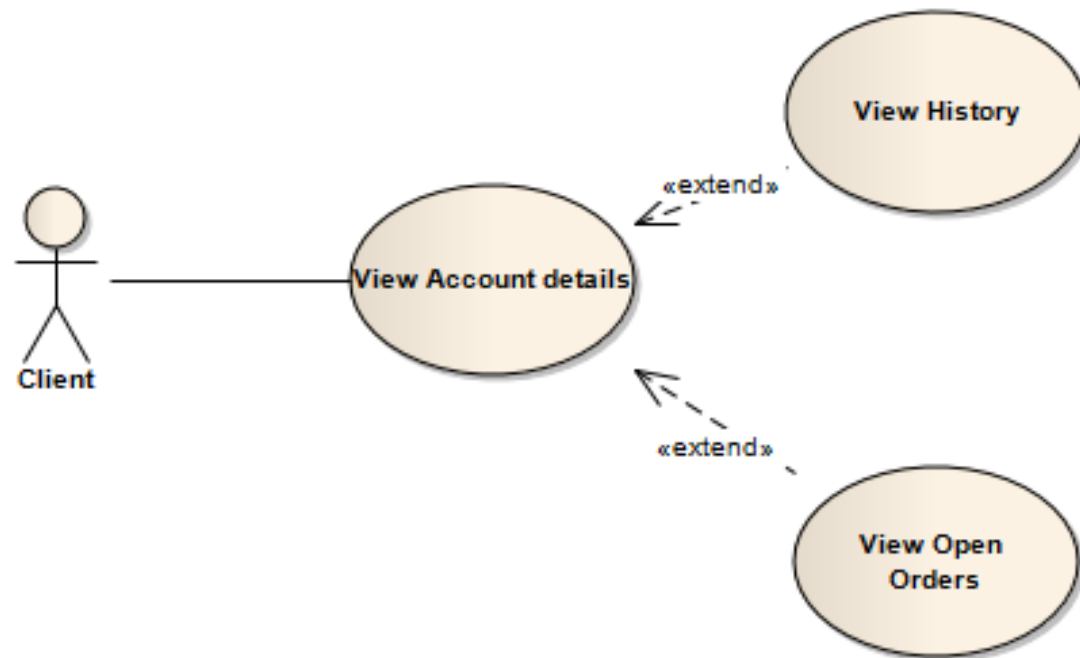
- * Veza <<include>> opisuje odnos između slučajeja upotrebe u kojem jedan slučaj upotrebe koristi usluge drugog.



Dijarami slučajeva upotrebe – upotreba tipa <<extend>>

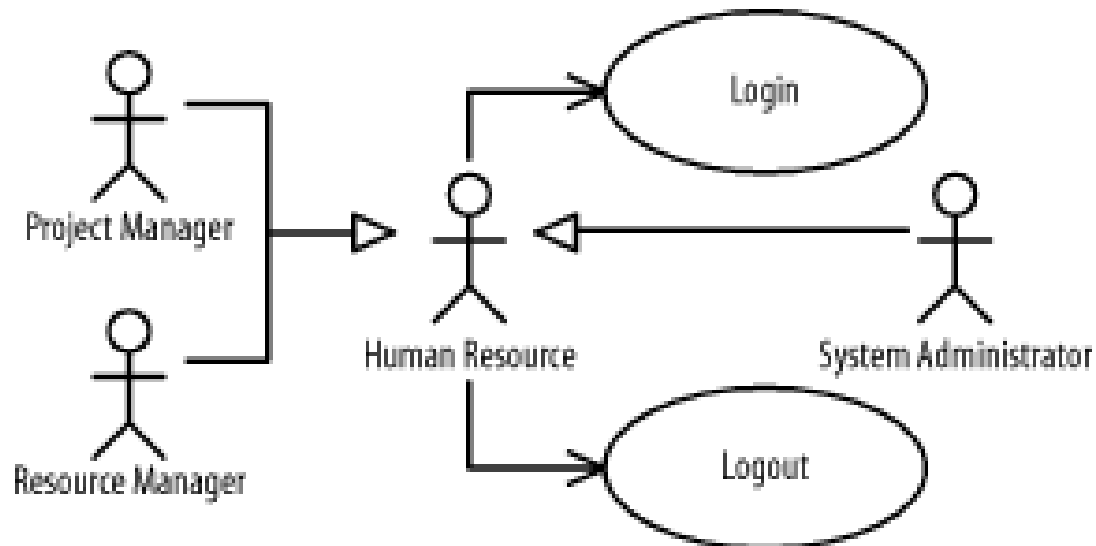
- * „Proširivanjem“ jednog slučaja upotrebe opisuje se neka složenija funkcija sistema.
- * Proširivanje se vrši sa jednim ili više drugih postojećih slučajeva upotrebe.
- * Ukoliko slučaj A proširuje slučaj B onda važi da:
 - * I slučaj A i slučaj B mogu da postoje sami
 - * Slučaj B može, a ne mora da bude proširen slučajem A.

Dijarami slučajeva upotrebe – upotreba tipa <<extend>>



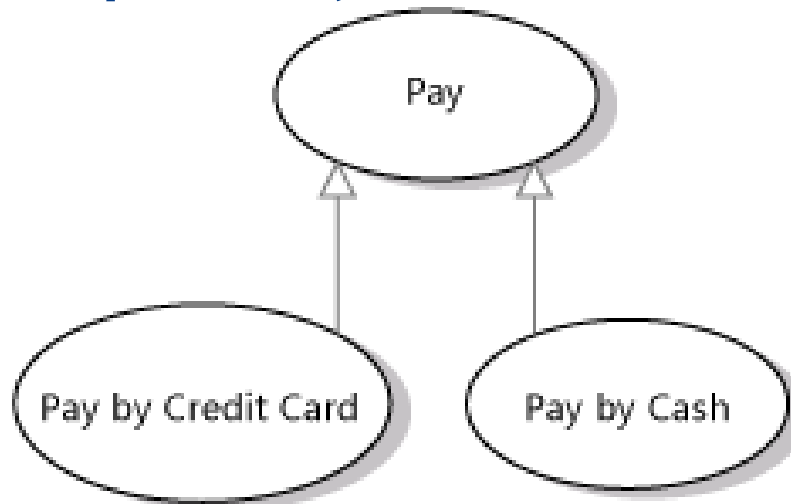
Dijagram slučajeja upotrebe - generalizacija

- * **Generalizacija – veza između roditelja i deteta – vezana za pojam nasleđivanja, dete nasleđuje osobine roditelja.**
- * **Generalizacija učesnika – izvedeni učesnik ima sve osobine i ponašanje osnovnog (apstraktnog) učesnika, ali može dodati osobine ili redefinisati ponašanje.**

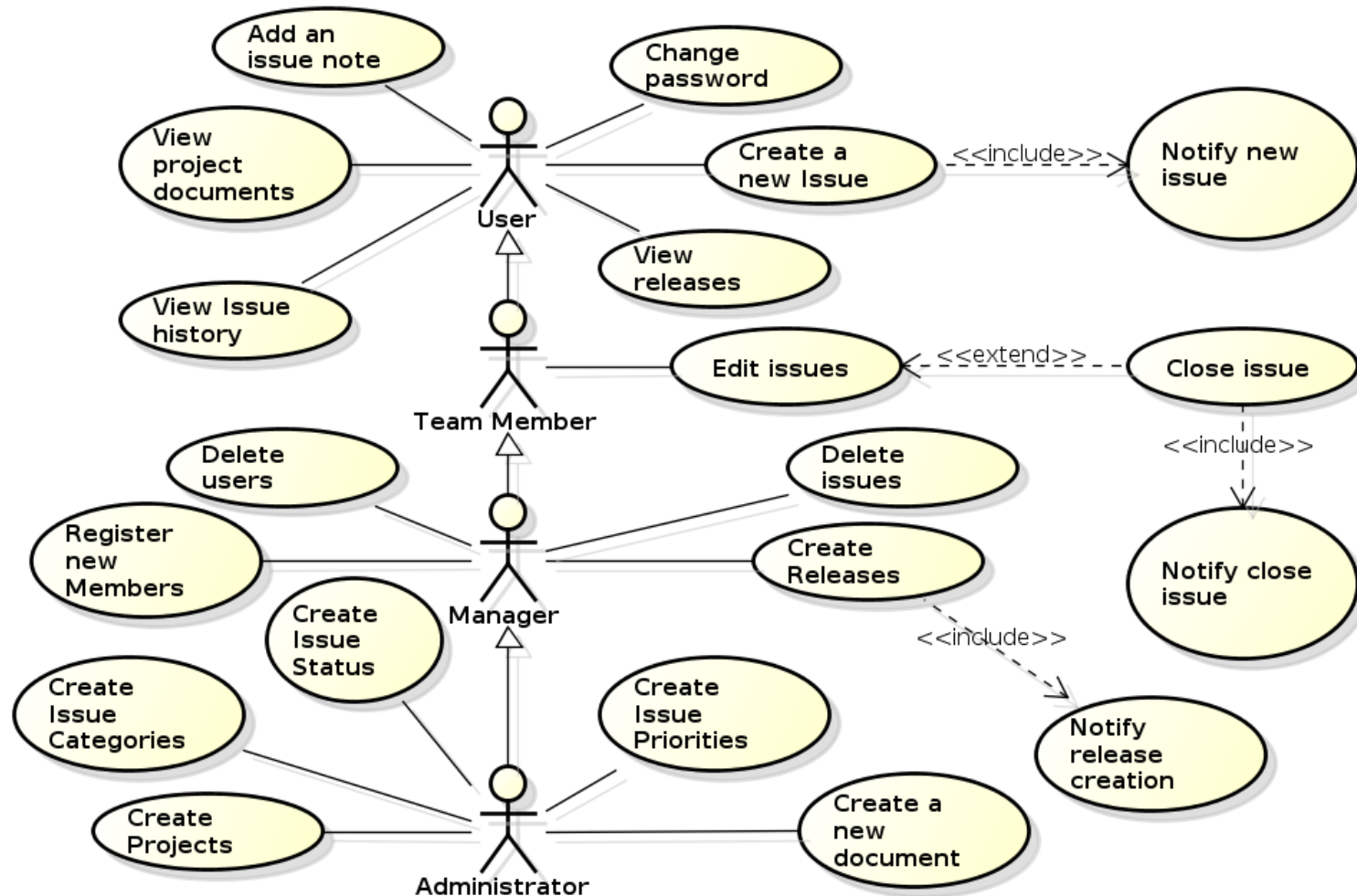


Dijagram slučajeva upotrebe - generalizacija

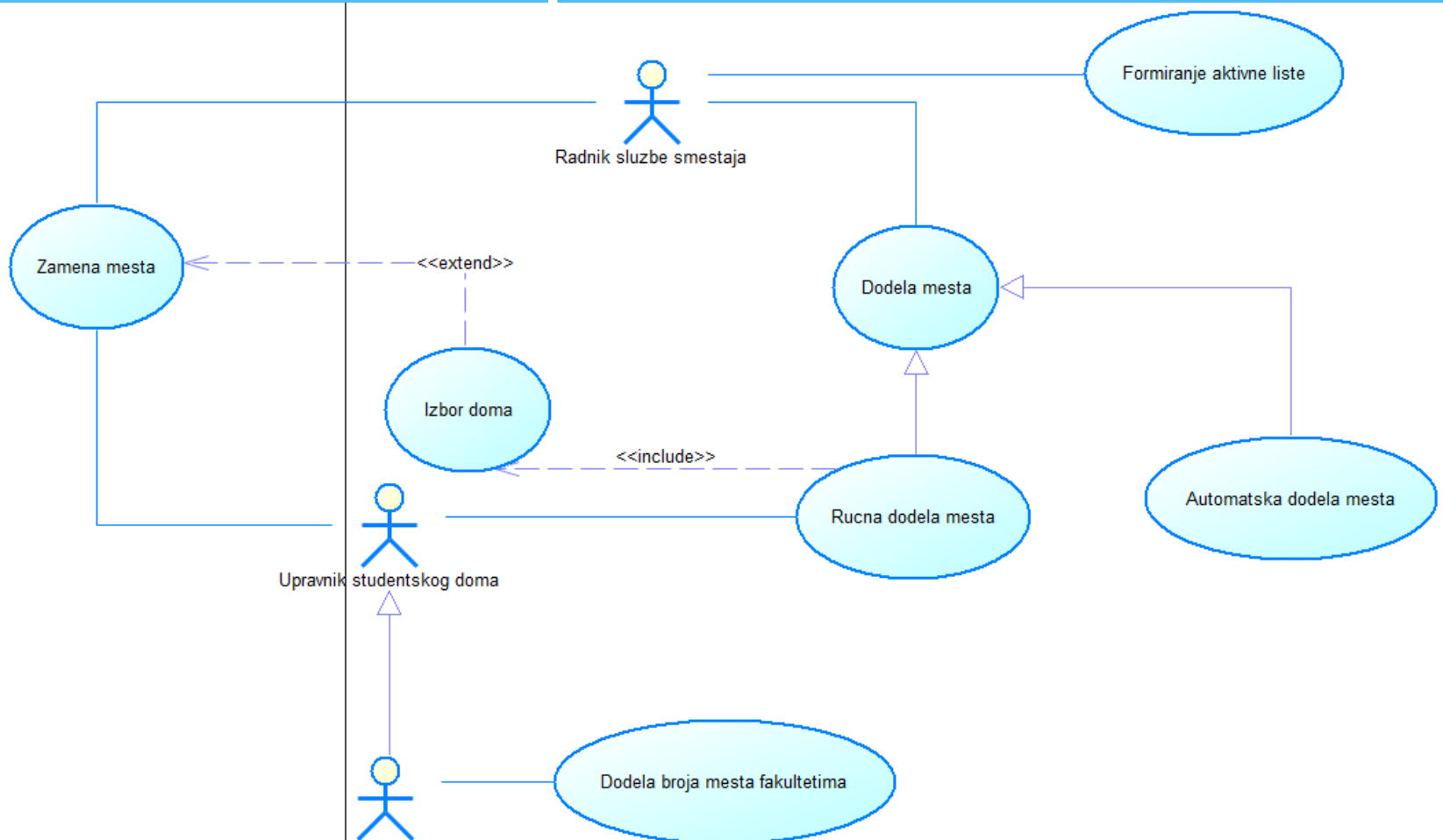
- * **Generalizacija slučajeva upotrebe – izvedeni slučaj upotrebe ima sve osobine i ponašanje apstraktnog slučaja upotrebe, ali može dodati osobine ili redefinisati ponašanje.**



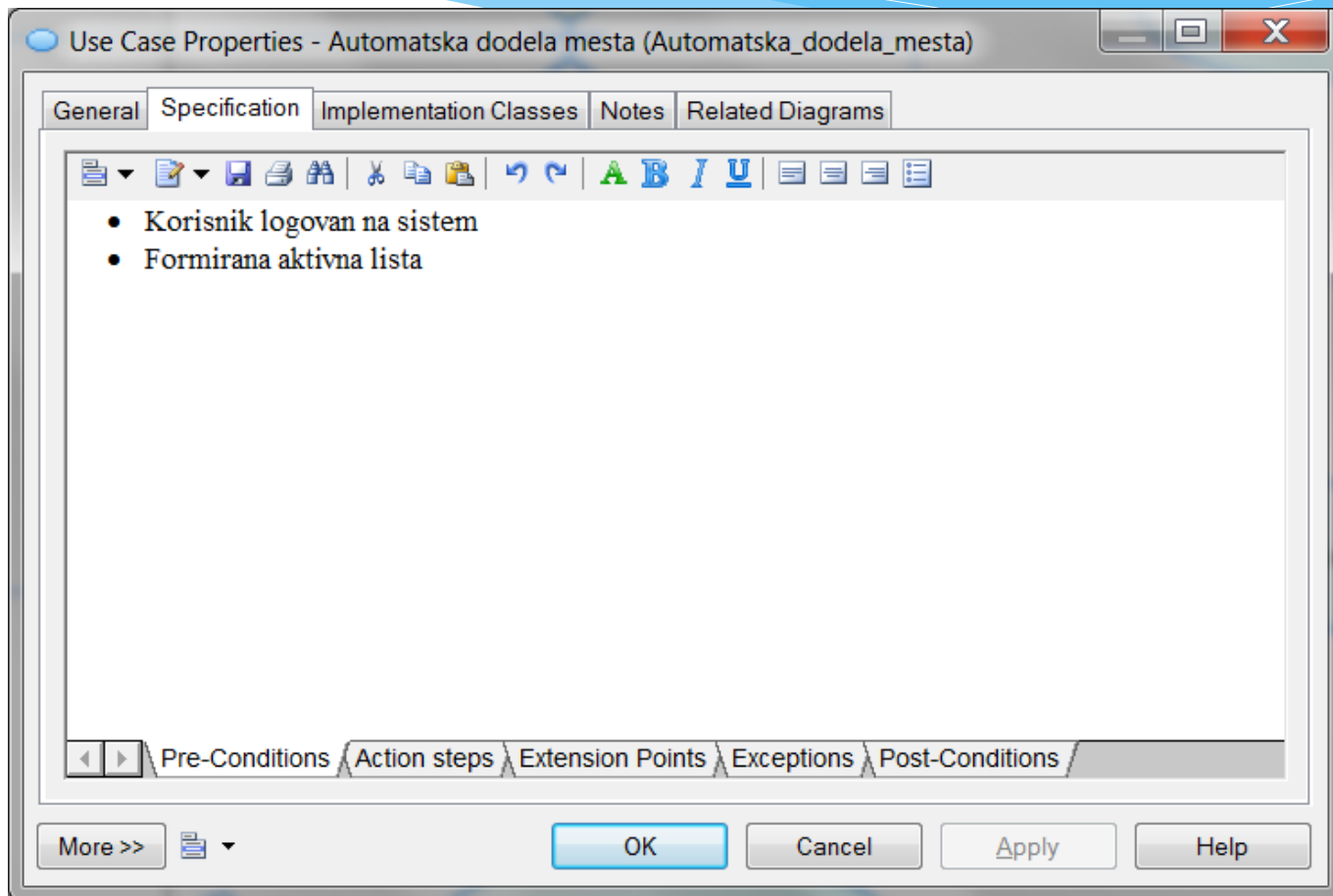
Dijagrami slučajeja upotrebe – primer 1



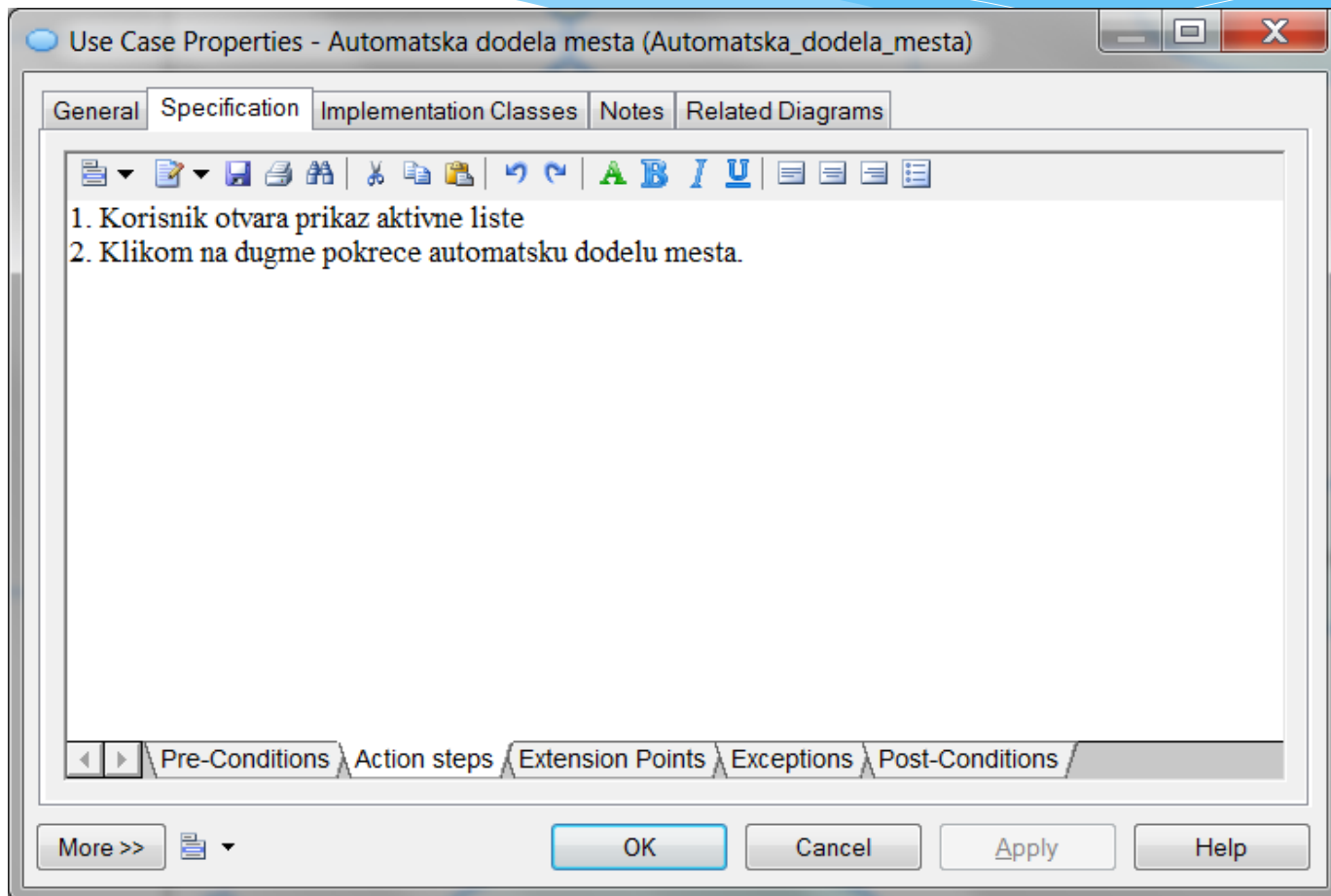
Dijagrami slučajeja upotrebe – primer 2



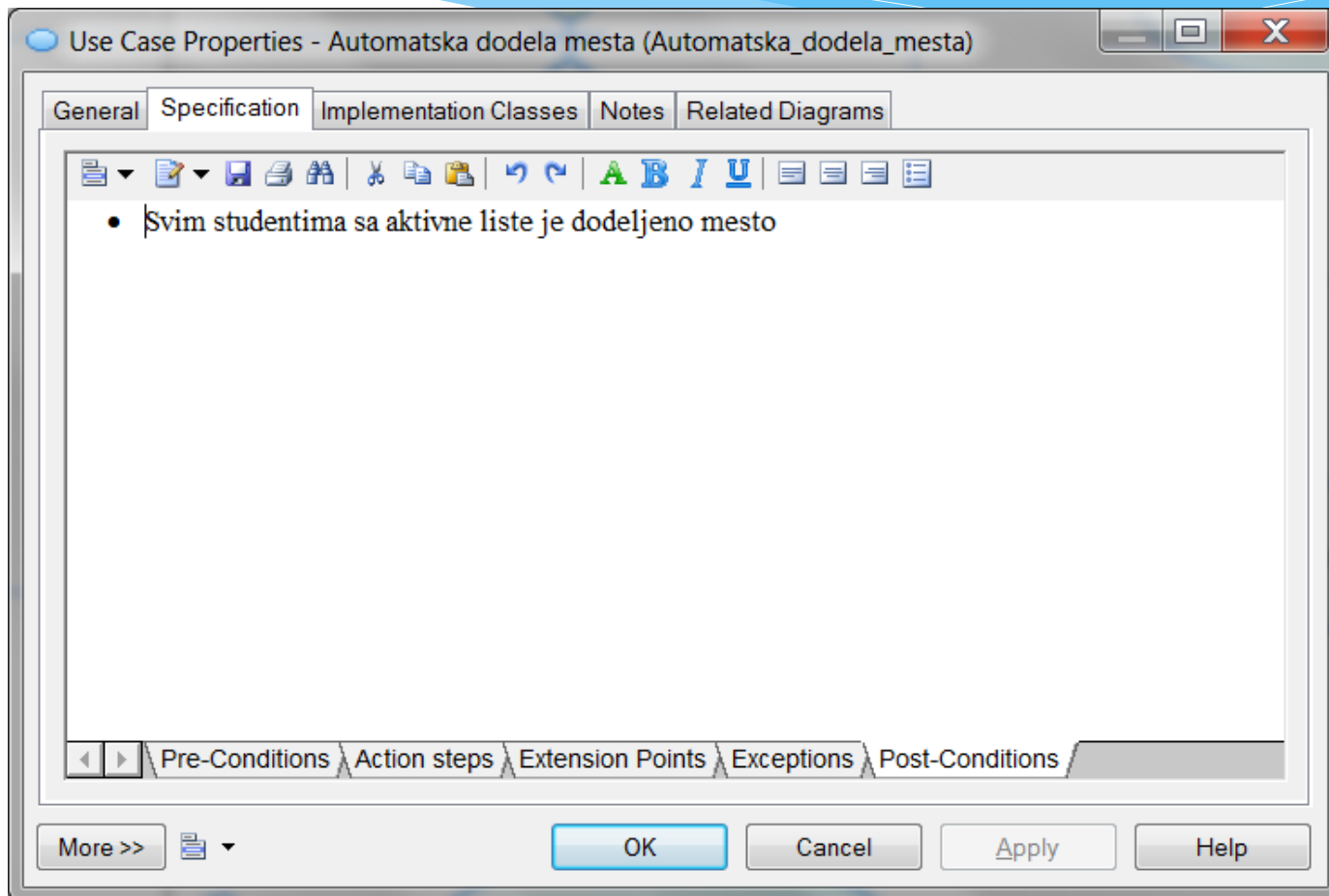
Dijagrami slučajeve upotrebe – primer 2



Dijagrami slučajeve upotrebe – primer 2



Dijagrami slučajeve upotrebe – primer 2

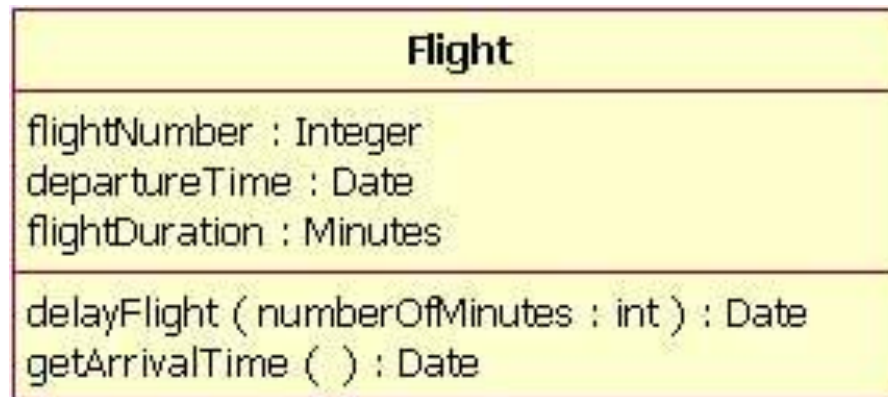


Dijagram klasa (Class Diagram)

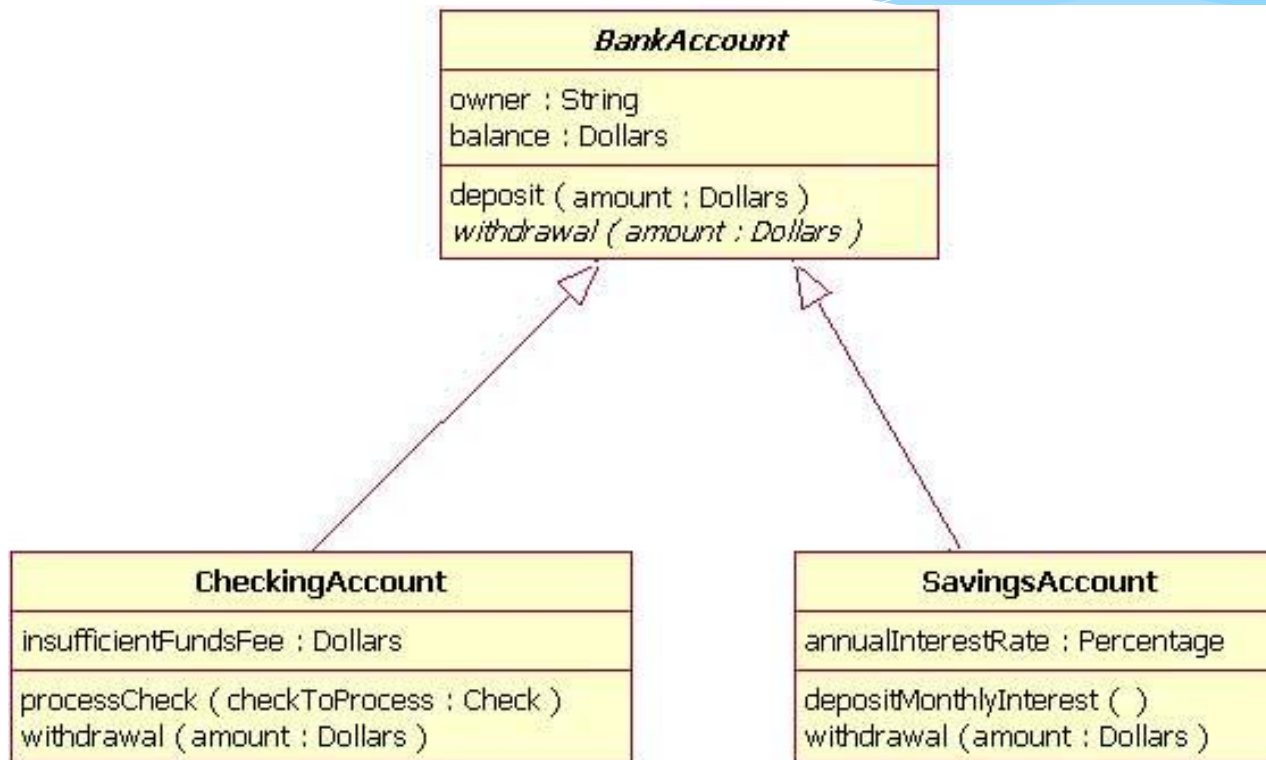
- * Spada u strukturne dijagrame.
- * Pokazuje statičku strukturu sistema i statičke odnose među njima.
- * Može se koristiti da pokaže implementacione klase, koje su stvari kojima se obično programeri bave.
- * Prikazuju klase sistema, njihove međusobne relacije, operacije i attribute klasa.

Dijagram klasa – prikaz

- * UML reprezentacija klase u okviru dijagrama klasa je pravougaonik u okviru koga su razlikuju tri dela.
- * Pregrada na vrhu ukazuje na naziv klase, srednji deo navodi attribute, a deo na dnu prikazuje metode klase.

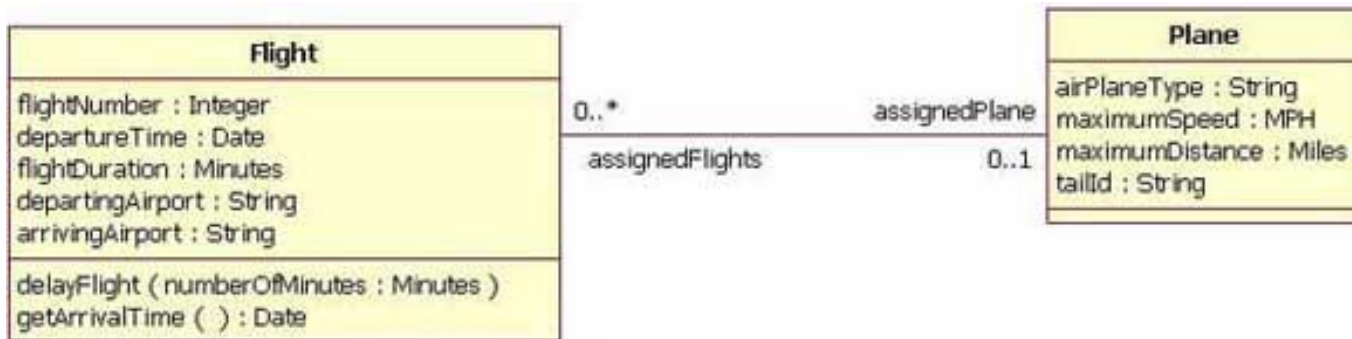


Dijagram klasa - nasleđivanje



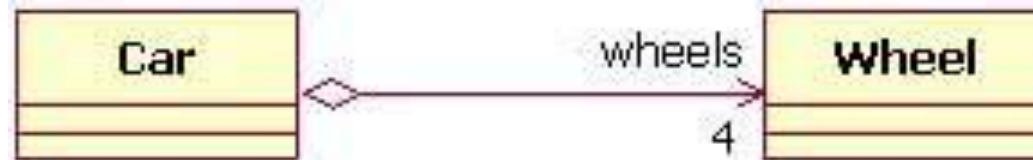
Dijagram klasa - asocijacija

- * Asocijacija predstavlja vezu između dve klase.
- * Na svakom kraju linije se postavi ime uloge i brojnu vrednost koja ukazuje na koji način su ove dve klase povezane.



Dijagram klasa - agregacija

- * Agregacija je specijalna vrsta asocijacije koja modeluje odnos *celina-deo*, pri čemu su delovi samostalne klase čije instance mogu da žive i nezavisno od klase koja ih agregira.

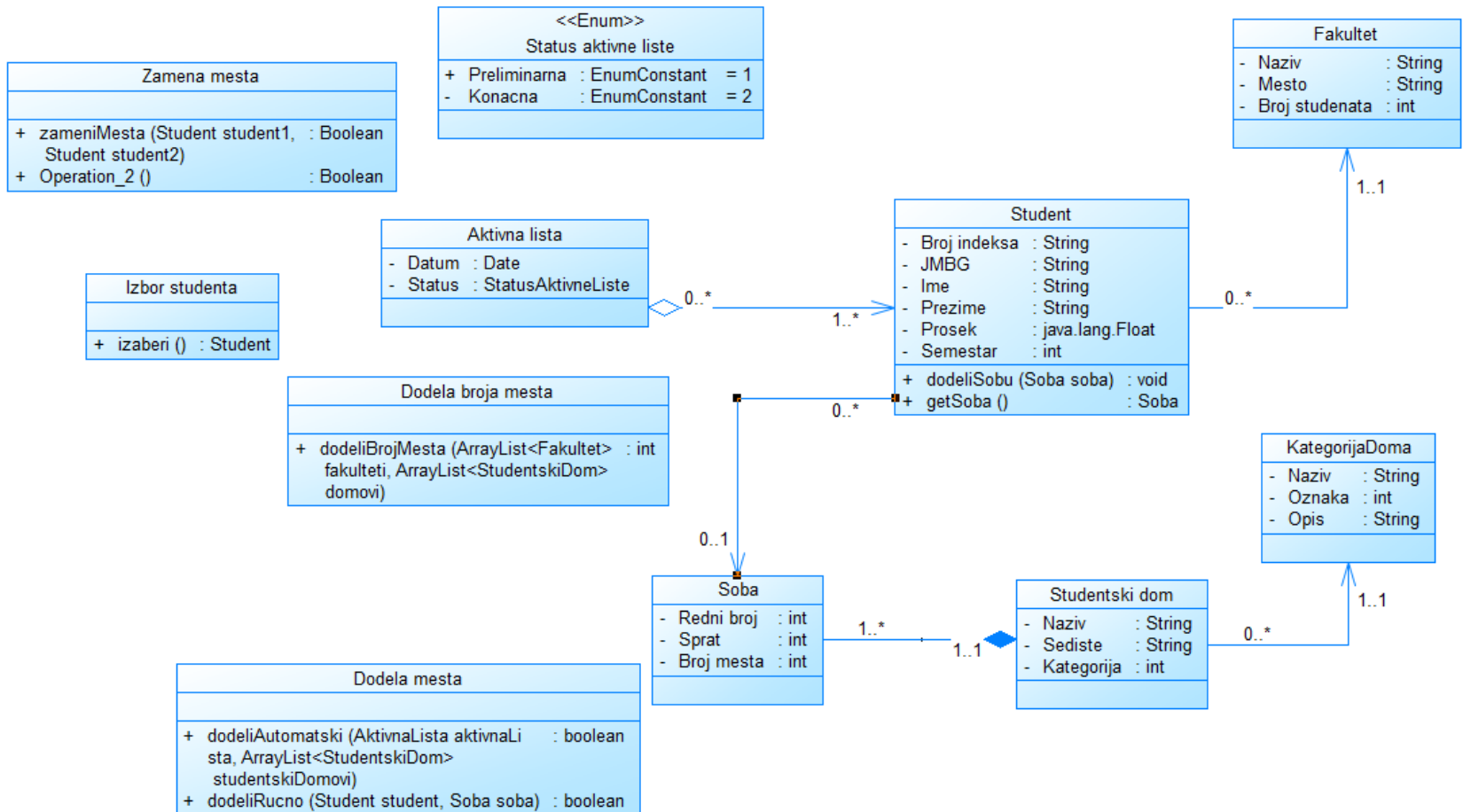


Dijagram klasa - kompozicija

- * Kompozicija je specijalna vrsta agregacije kod koje su delovi slabi objekti koji ne mogu samostalno da postoje bez osnovne klase.






Dijagram klasa – primer



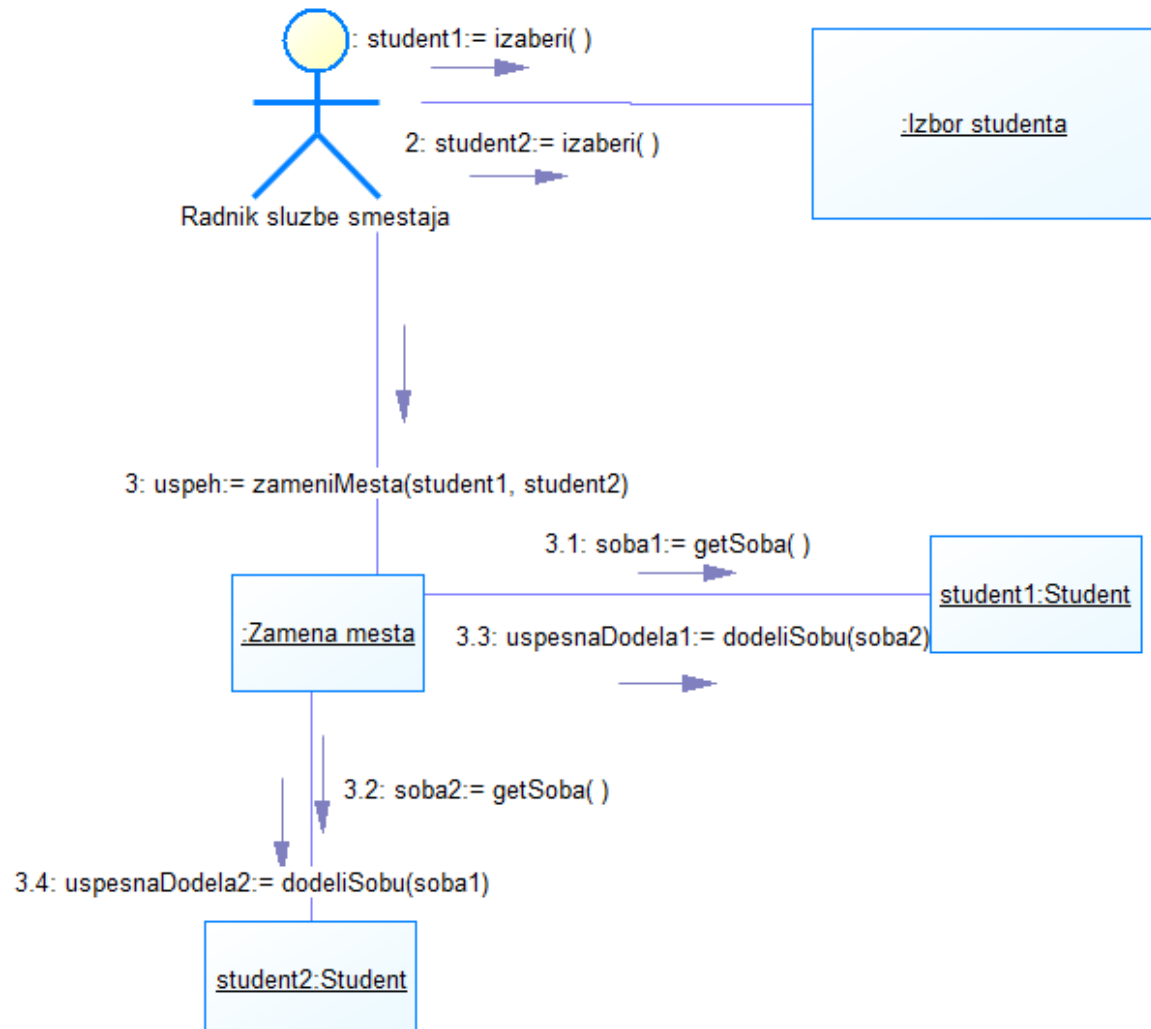
Dijagram saradnje

- * Spada u dijagrame interakcija
- * Prikazuje veze između objekata i sekvence poruka koje se razmenjuju između njih
- * Dodeljuje funkcionalnosti objektima
- * Slobodan raspored objekata i poruka na dijagramu
- * Objekti su povezani asocijacijama, a poruke se prenose preko veza
- * Redosled poruka je određen rednim brojem koji je sastavni deo naziva poruke

Dijagram saradnje - elementi

Element	Simbol
<p>Objekat: Učesnik interakcije u sistemu. Pun naziv objekta u formi <nazivObj>:<nazivKlase> gde je <nazivObj> opcioni deo.</p>	
<p>Asocijacija: Veza između objekata. Kardinalitet se može ali ne mora navesti.</p>	
<p>Poruka: Model interakcije između objekata, npr. poziv metode u OO jeziku. Broj u nazivu označava redosled slanja poruke.</p>	

Dijagram saradnje – primer

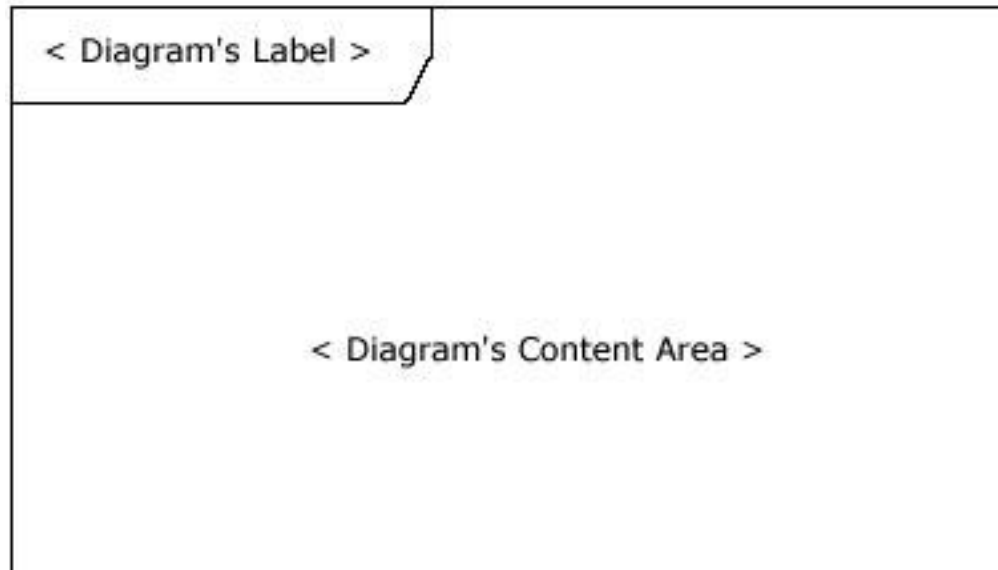


Dijagram sekvenci

- * Spada u dijagrame interakcija – modeluje dinamički aspekt sistema
- * Implementacija slučaja korišćenja
- * Prikazuje interakcije između objekata u redosledu u kom se dešavaju
- * Interakcija – razmena poruka između objekata
- * Fokus je na redosledu i tipu poruka, a ne na detaljima implementacije poruka
- * Dve dimenzije:
 - * Vremenska (vertikalna) - prikazuje životni vek objekata
 - * Kolekcija objekata i aktera (horizontalna)

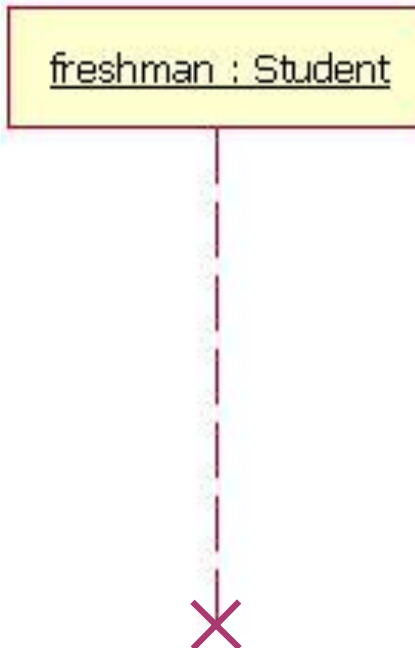
Dijagram sekvenci - okvir

- * Okvir - opcioni element
- * Vizuelna granica dijagrama
- * Može da odvoji deo dijagrama od celog dijagrama
- * Naziv dijagrama počinje sa 'sd' – UML 2



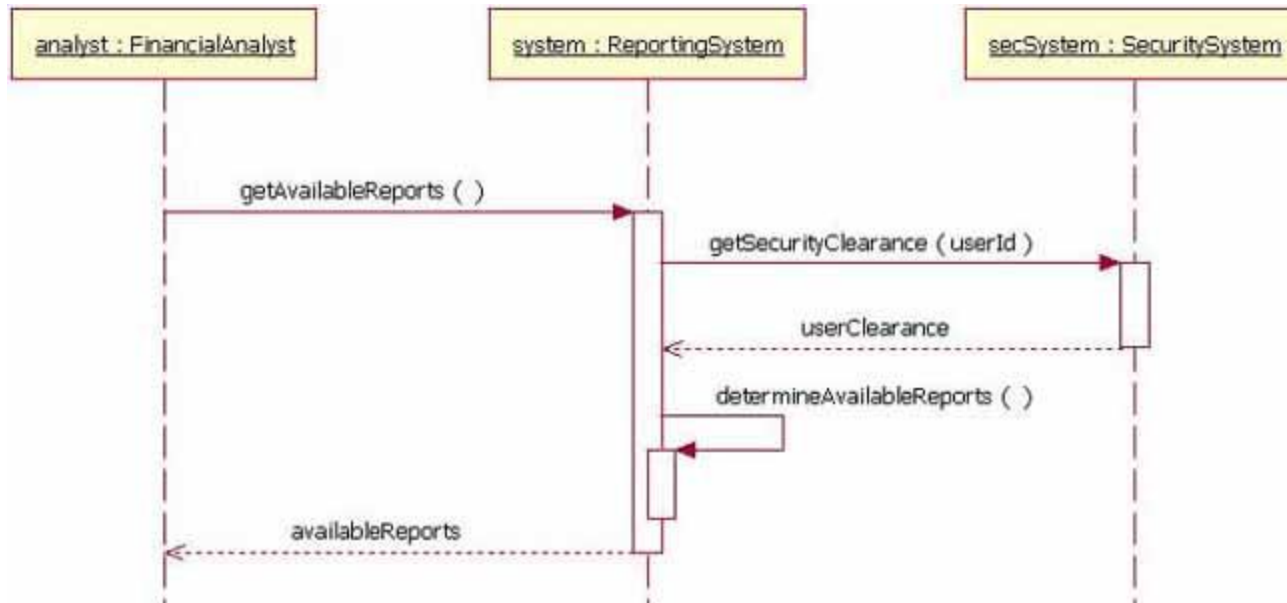
Dijagram sekvenci - objekat

- * Objekat – osnovni element dijagrama
- * Životni vek je definisan isprekidanom linijom
- * Naziv objekta je opcion






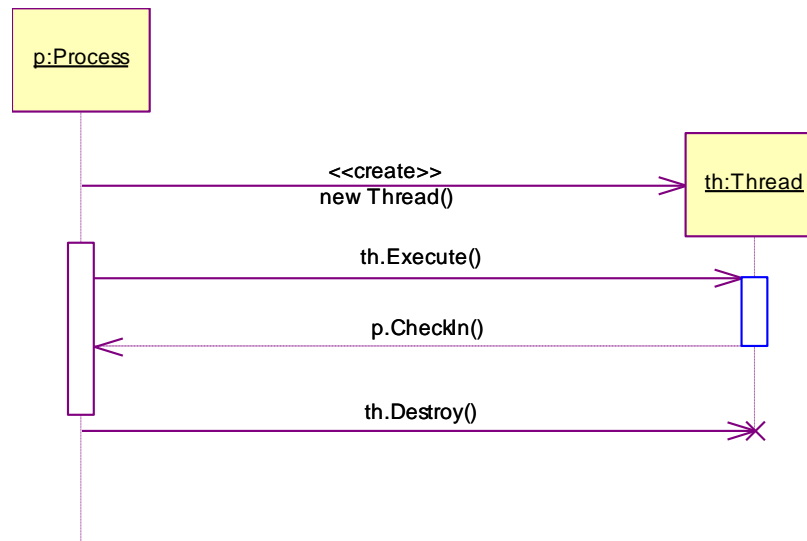
Dijagram sekvenci - poruka

- * Model interakcije između objekata
- * Prijem poruke izaziva izvršenje operacije
- * Poruka je definisana nazivom i parametrima



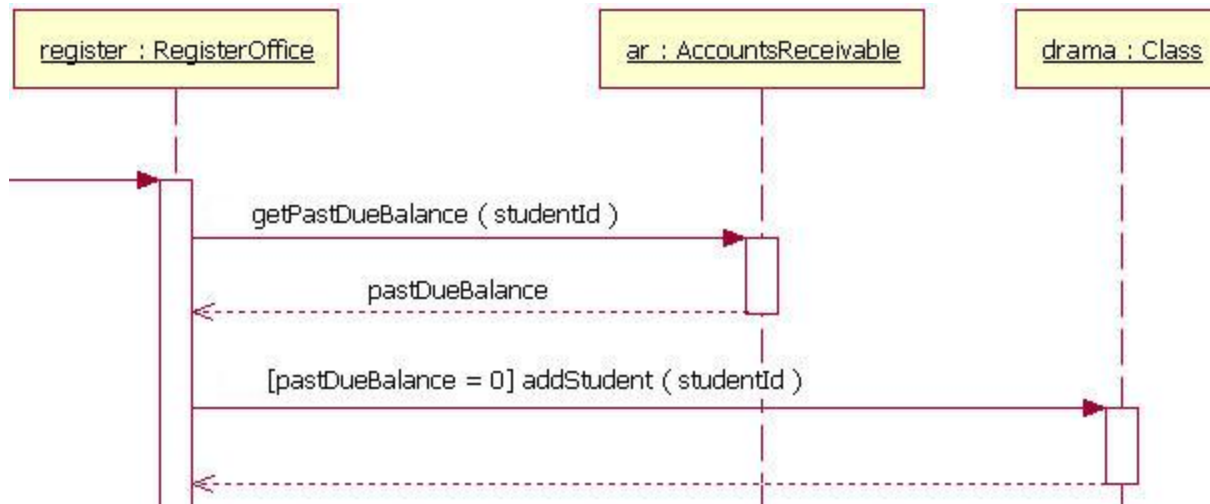
Dijagram sekvenci – tipovi poruka

- * Poziv – sinhrona poruka 
- * Odgovor 
- * Asinhrona poruka 
- * Rekurzivna poruka



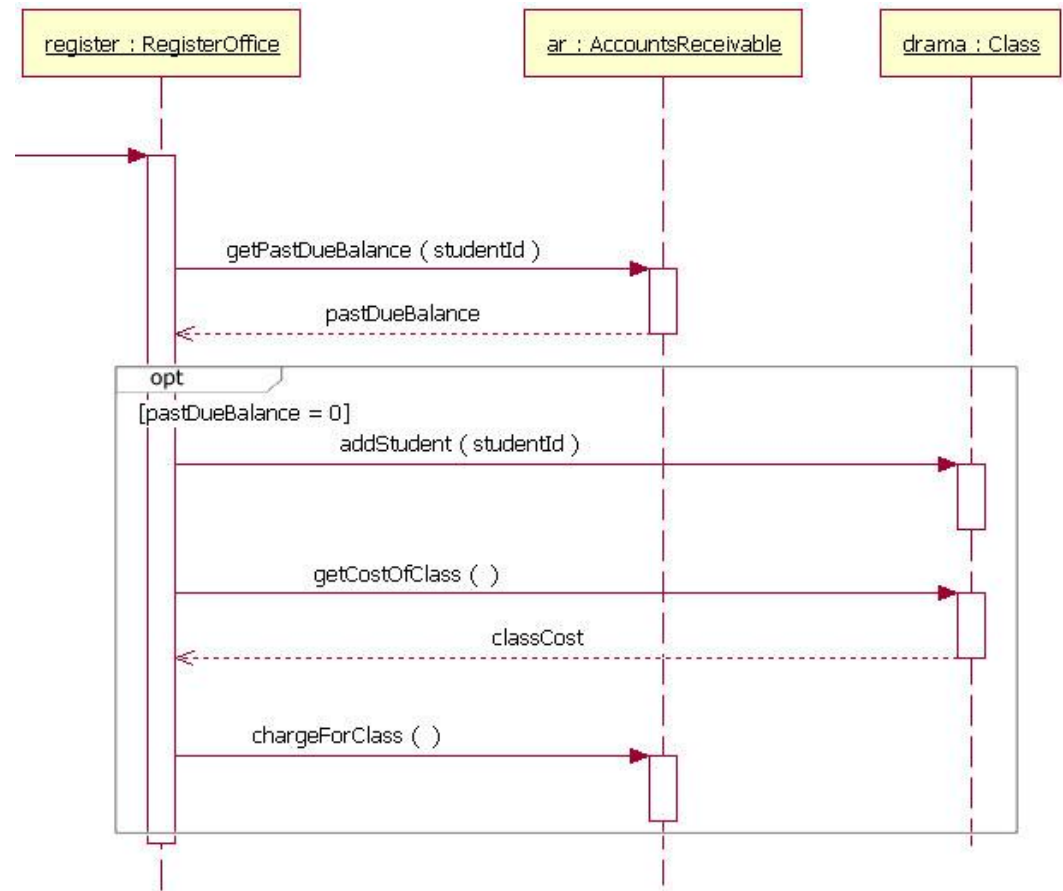
Dijagram sekvenci – uslovne poruke

- * Uslov mora biti ispunjen da bi poruka bila poslata



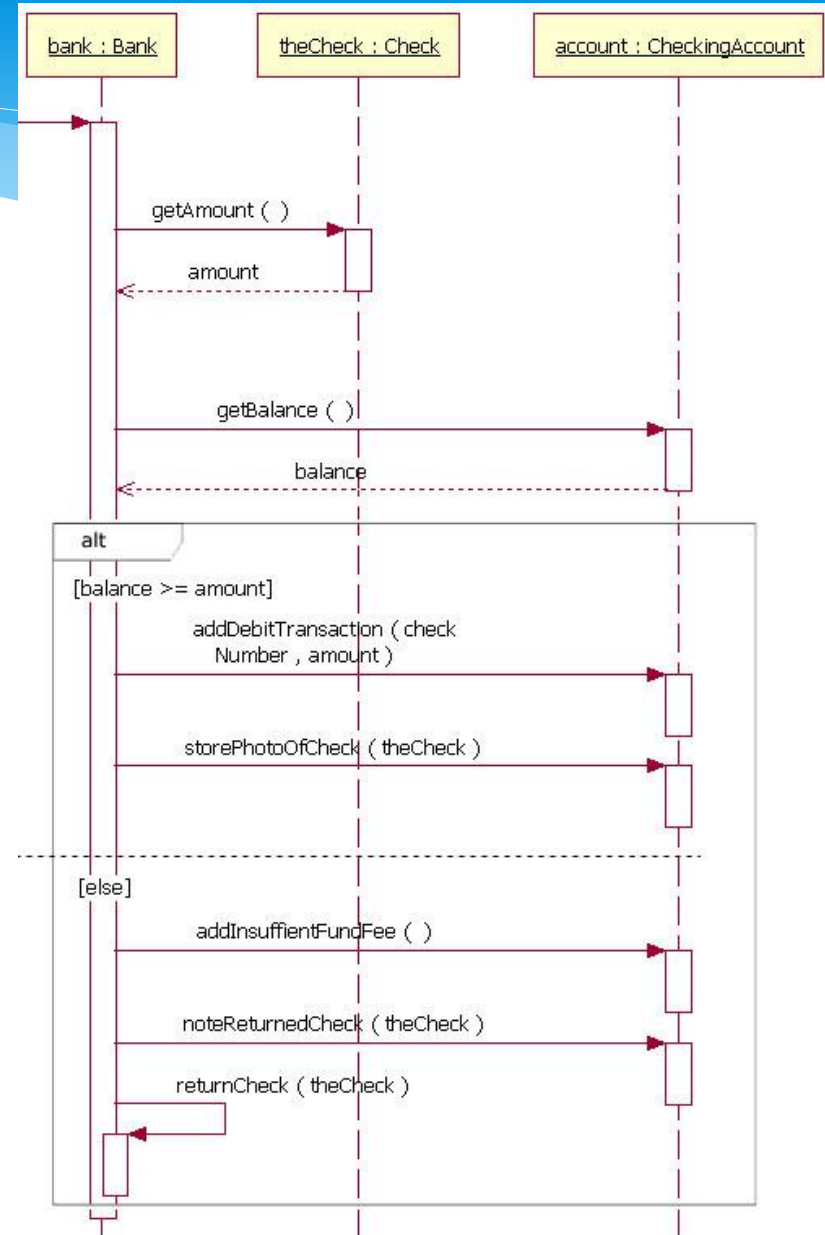
Dijagram sekvenci – uslovne sekvence

- * Sekvenca se izvršava ako je uslov ispunjen
- * Model „if then” naredbe

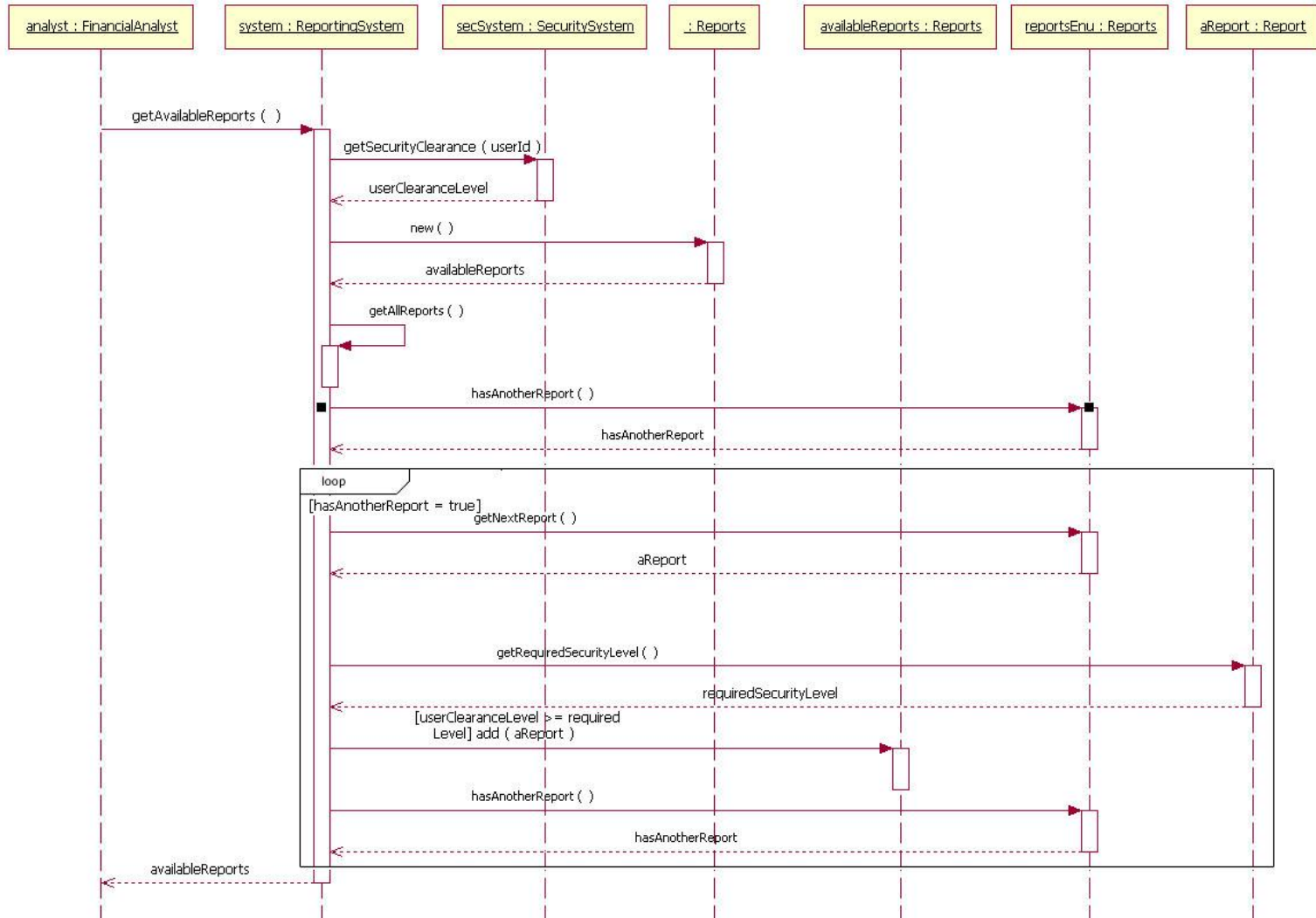


Dijagram sekvenci – alternativne

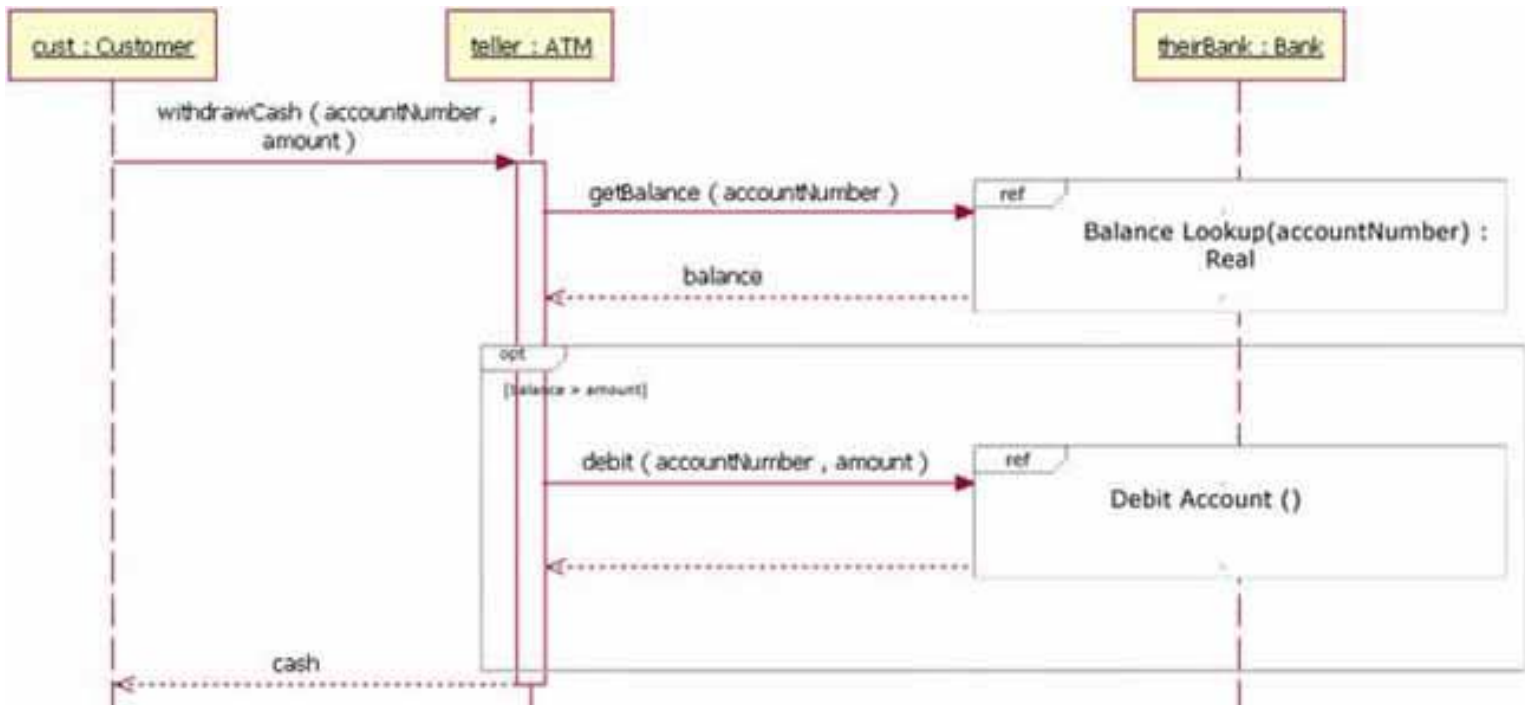
- * Predstavljanje alternativnih sekvenci razmena poruka
- * Model „if then ... else” naredbe



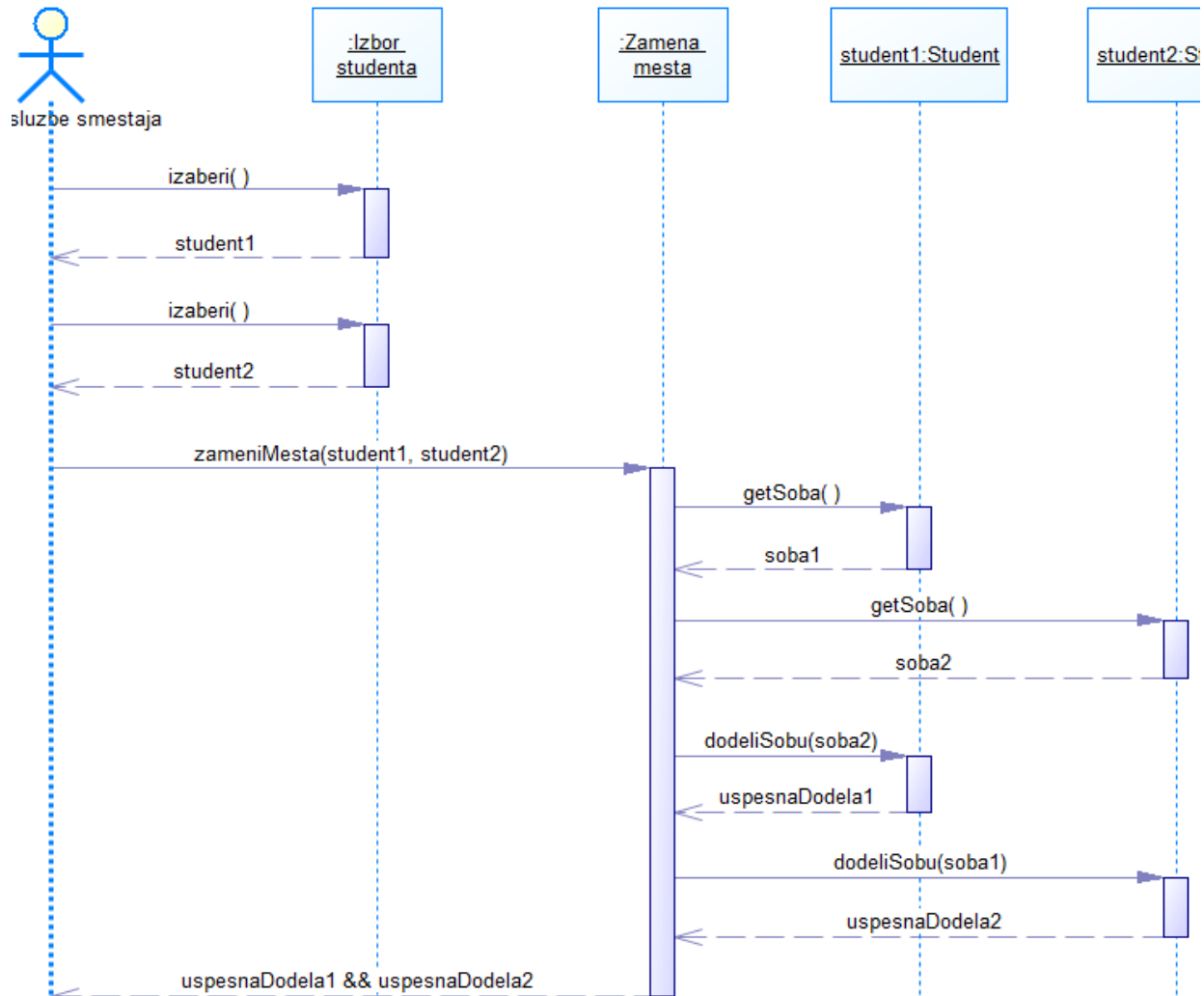
Dijagram sekvenci – petlja



Dijagram sekvenci – ugnježdeni dijagrami



Dijagram sekvenci – primer



Fakultet tehničkih nauka, Novi Sad

Inženjering informacionih sistema

Dr Ivan Luković

Dr Slavica Horđić

Nikola Čučević

Vanja Bjelica

Dr Jelena Borocki

Dr Milan Delić

Kraj prezentacije