



UML

(The Unified Modeling Language)

Objektno-orijentisano
modeliranje - PRIMERI

UML & OO (OBJEKTNO-ORIJENTISANO) MODELIRANJE

- Objektno orijentisano modeliranje softverskih sistema
- UML – pregled
- Proces razvoja softvera
- Projektovanje arhitekture
- Modeliranje sistema korišćenjem uml-a

OO MODELIRANJE SOFTVERSKIH SISTEMA

- Primarni proizvod: dobar softver
- Kompleksnost softvera
- Ciljevi modeliranja
- Principi modeliranja
- Objektno orijentisano (OO) modeliranje

OO MODELIRANJE: ciljevi

■ Model:

- Pojednostavljenje realnosti
- Bolje razumevanje sistema koji se razvija
- Vizuelizacija sistema
- Specifikacija strukture i ponašanja
- Vodič kroz konstrukciju sistema
- Dokumentovanje odluka
- Savlađivanje kompleksnosti

[OO MODELIRANJE: principi]

- Izbor modela utiče na način rešavanja problema
- Primeri:
 - kvantna fizika: interakcija fotona, Feynmann dijagrami
 - softver
 - perspektiva baza podataka – EER
 - algoritamski – SSA
 - OO – klase i obrasci interakcije
- Svaki model – razni nivoi preciznosti: primeri

OO MODELIRANJE: principi (nast.)

- Najbolji modeli su povezani sa realnošću
- Netrivijalni sistem zahteva mali skup skoro nezavisnih modela: primeri
 - model slučajeva korišćenja
 - logički model
 - model procesa – distribucija procesa i niti
 - model implementacije
 - model razmeštaja – sistemsko inženjerstvo
 - svaki sa strukturnim aspektima i aspektima ponašanja
 - različit značaj

Objektno-orijentisano modeliranje

- Tradicionalni pristup: algoritamski: gradivni blokovi softvera su procedure i funkcije - kontrola i idekompozicija – teško održavanje
- Savremen pristup: OO perspektiva: gradivni blokovi softverskih sistema su objekti i klase
- Objekat: stvar iz rečnika prostora problema ili rešenja; ima identitet, stanje i ponašanje
- Klasa – opis skupa sličnih objekata

Objektno-orijentisano modeliranje (nast.)

- Primer: troslojna arhitektura sistema naplate:
 - korisnički interfejs: objekti dugme, meni, dijalog boks
 - baza podataka – objekti tabele entiteta: kupci, proizvodi, računi
 - srednji sloj – transakcije i poslovna pravila i viši nivo pogleda na entitete
- OO – prilagođen sistemima svih stepena, veličina i kompleksnosti
- Savremeni jezici i operativni sistemi

UML – PREGLED

- UML: standardni jezik za vizuelizaciju, specifikaciju, konstrukciju i dokumentovanje OO sistema
- UML: razvoj
- UML: osnovni koncepti

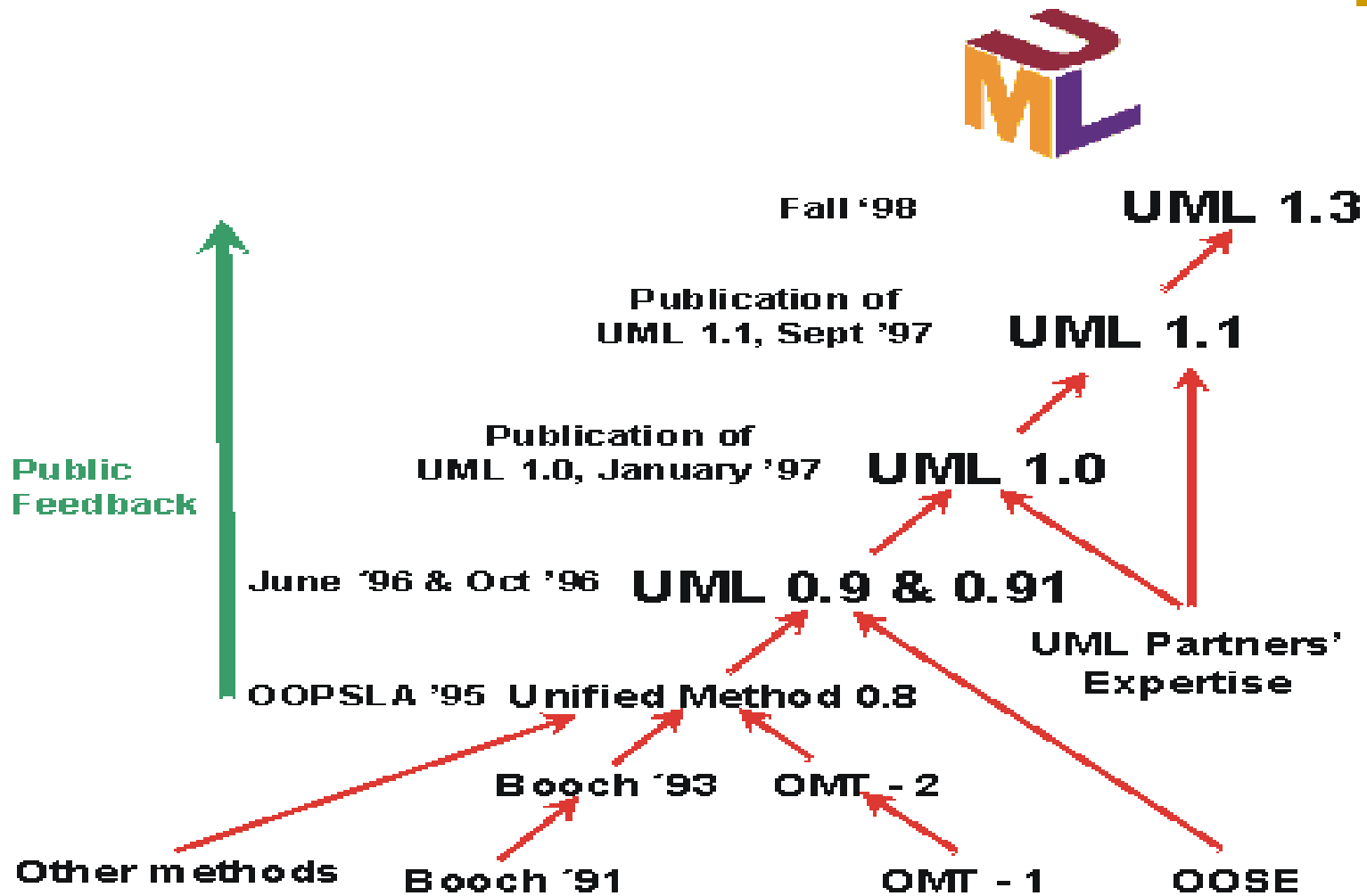
[UML – PREGLED: razvoj]

- OO jezici modeliranja (70.-80.)
- Grady Booch (OO method; projektovanje i konstrukcija; Rational Software Corporation)
- Ivar Jacobson (OOSE – Object-Oriented Software Engineering, specifikacija, slučajevi korišćenja; Objectory)
- James Rumbaugh (OMT, analiza i obrada podataka; General Electric)
- David Harel (dijagrami stanja; Booch, OMT, ponašanje)

UML – PREGLED: razvoj (nast.)

- Unifikacija - ciljevi:
 - Modeliranje softverskih sistema korišćenjem isključivo OO tehnika
 - Skaliranje kompleksnih sistema
 - Jezik modeliranja i za ljude i za mašine
- UML 1.1, OMG, 1997.
- Hewlett-Packard, I-Logix, IBM, ICON, Intellicorp, Oracle, Platinum Technology, Rational Software, TI, Unisys, etc.

UML – PREGLED: razvoj (nast.)



UML – PREGLED: razvoj (OMG) – UML 2

- UML, along with the Meta Object Facility (MOF™), also provides a key foundation for OMG's Model-Driven Architecture®, which unifies every step of development and integration from business modeling, through architectural and application modeling, to development, deployment, maintenance, and evolution.
- OMG is a not-for-profit computer industry specifications consortium; our members define and maintain the UML specification
- 2004: UML 2.0 (...2.x)
 - Superstructure specification
 - UML 2.0 Infrastructure
 - UML 2.0 Object Constraint Language (OCL)
 - UML 2.0 Diagram Interchange

UML – PREGLED: konceptualni model

- Osnovni gradivni blokovi
- Pravila kombinovanja gradivnih blokova
- Opšti mehanizmi koji se primenjuju kroz ceo UML

UML – PREGLED: gradivni blokovi

- Elementi (“stvari”)
- Odnosi – povezuju elemente
- Dijagram – grupišu kolekcije elemenata

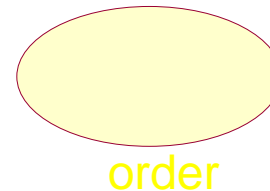
[UML – PREGLED: elementi]

- Strukturni elementi
- Elementi ponašanja
- Elementi grupisanja
- Elementi označavanja

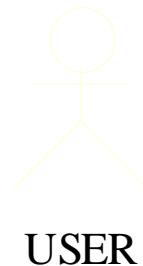
UML – PREGLED: strukturni elementi

- Statički deo modela: konceptualni ili fizički

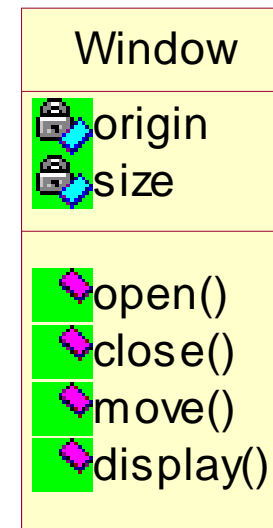
- Slučaj korišćenja



- Akter

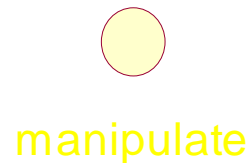


- Klasa

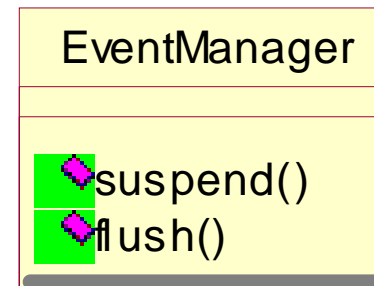


UML – PREGLED: strukturni elementi (nast.)

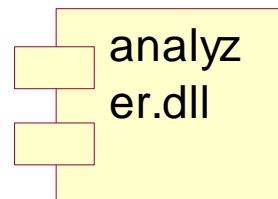
- Interfejs



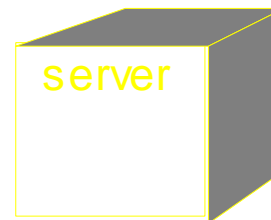
- Aktivna klasa



- Komponenta



- Čvor

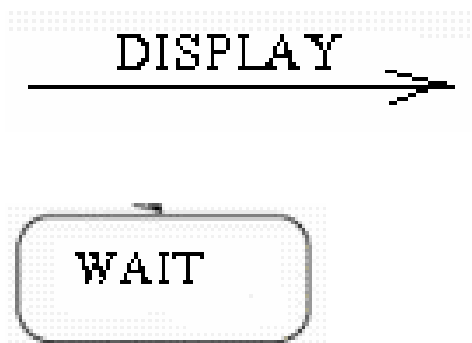


UML – PREGLED: elementi ponašanja

■ Dinamički deo modela

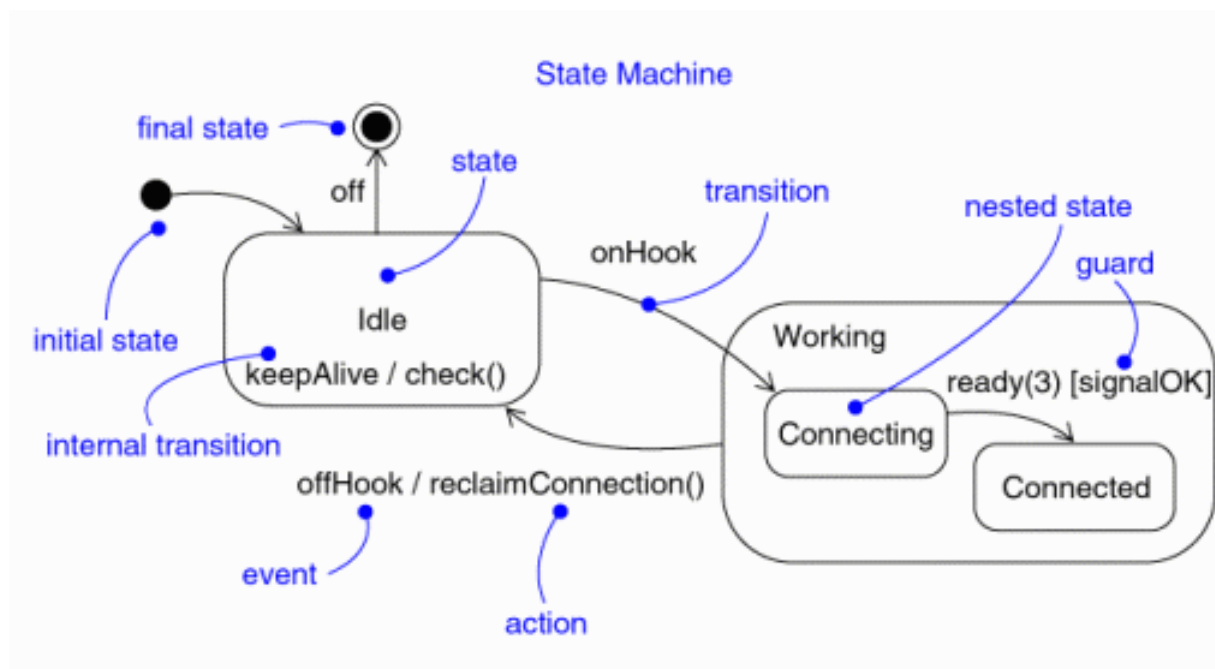
- Interakcija – poruke, akcije, veze

- Stanje:



UML – PREGLED: elementi ponašanja (nast.)

- Mašina stanja – niz stanja objekta: stanja, prelazi, događaji, aktivnosti



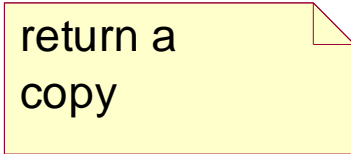
UML – PREGLED: elementi grupisanja

- Paket – grupiše strukturne, elemente ponašanja ili druge elemente
- Konceptualni element (nije fizički, kao komponenta);
- Varijante: okviri (frames), podsistemi

Business rules

UML – PREGLED: elementi označavanja

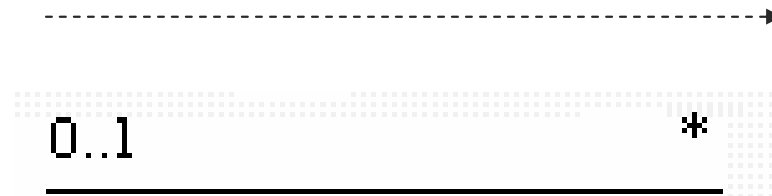
- Delovi modela koji objašnjavaju
- Napomena – komentar ili ograničenje
- Varijanta: zahtev (izvan modela)



return a
copy

[UML – PREGLED: odnosi]

- Zavisnost



- Asocijacija



- Generalizacija



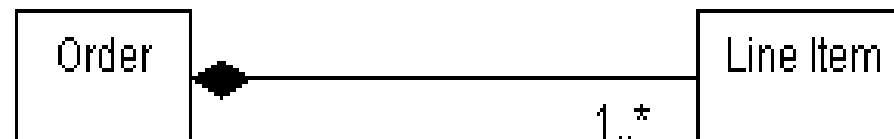
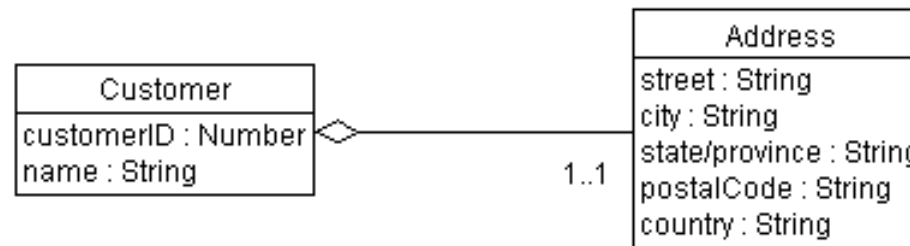
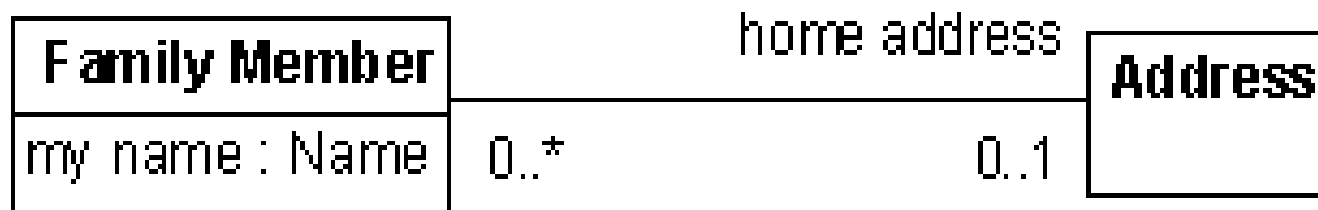
- Realizacija



- (između interfejsa i klase i /ili komponente)

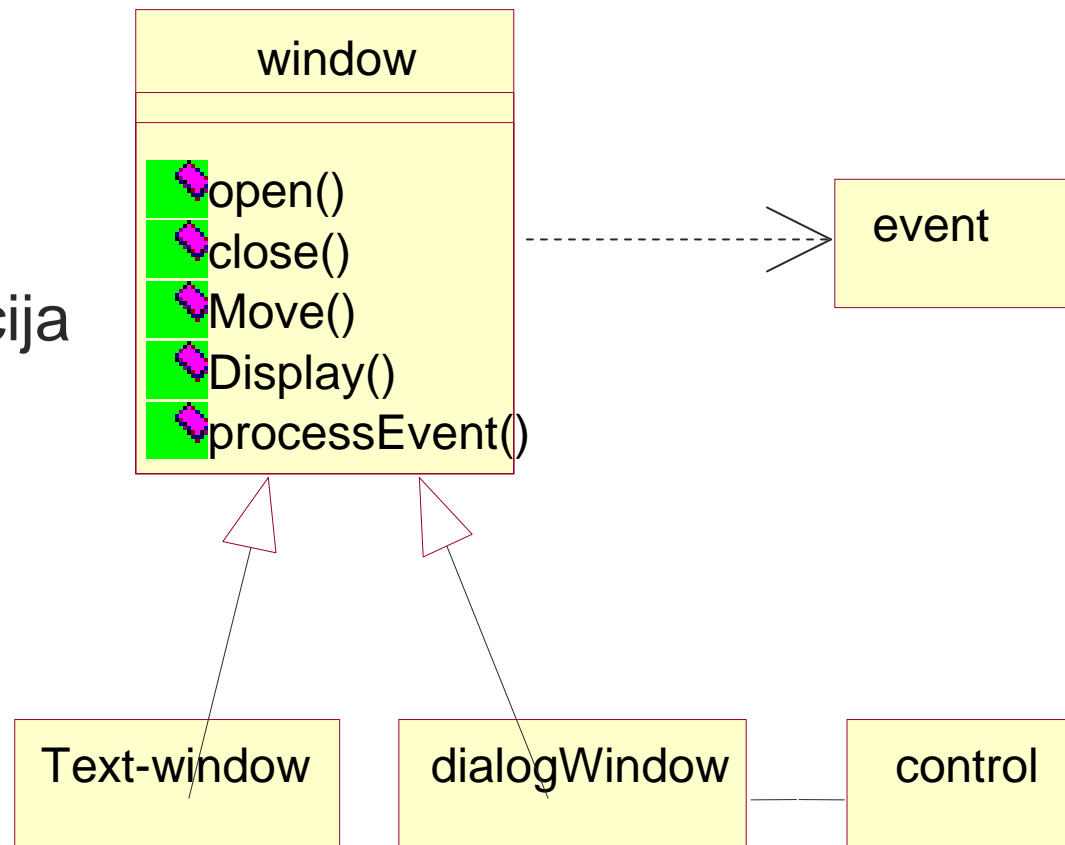
UML – PREGLED: odnosi (nast.)

- Primeri (asocijacija, agregacija, kompozicija):



[UML – PREGLED: odnosi (nast.)]

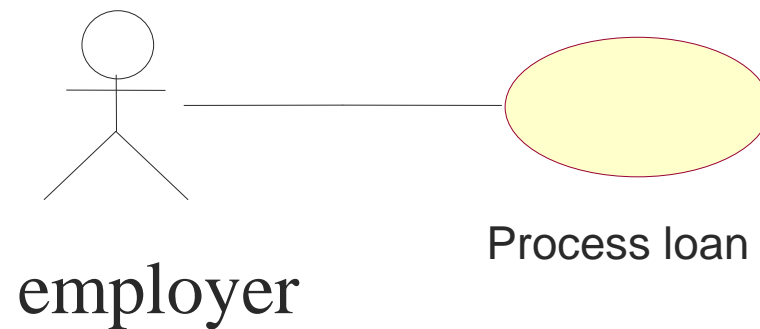
- Primeri:
- Zavisnost
- Generalizacija
- Asocijacija



Baza podataka i SUBP - sistemi za upravljenje bazama podataka

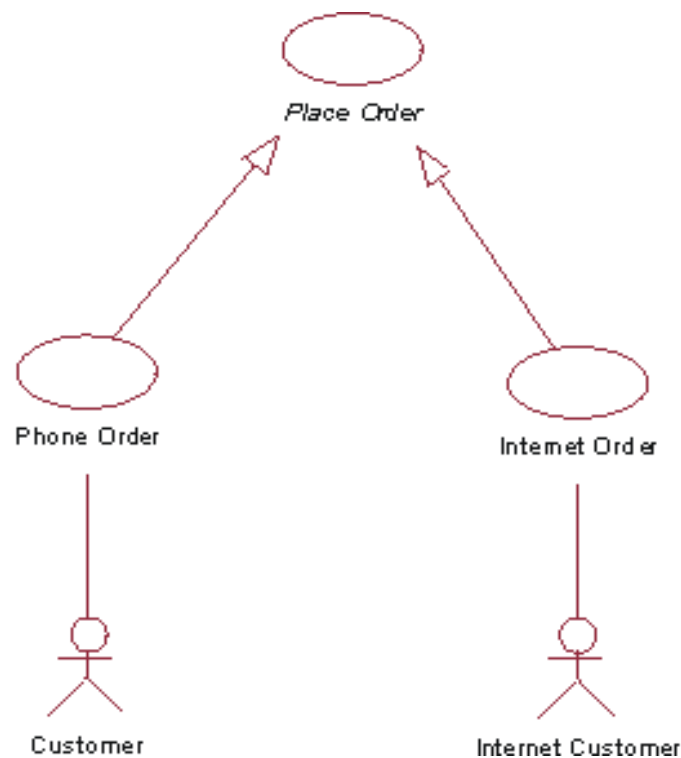
[UML – PREGLED: odnosi (nast.)]

- Primeri (asocijacija):



[UML – PREGLED: odnosi (nast.)]

- Primeri:
- Asocijacija
- Generalizacija



UML – PREGLED: dijagrami

- Grafička reprezentacija skupa elemenata
- Graf sa temenima (elementima) i granama (odnosima)
- Obično vezani za modele (poglede) (5)
- Devet vrsta dijagrama (najčešćih u praksi):

UML – PREGLED: dijagrami

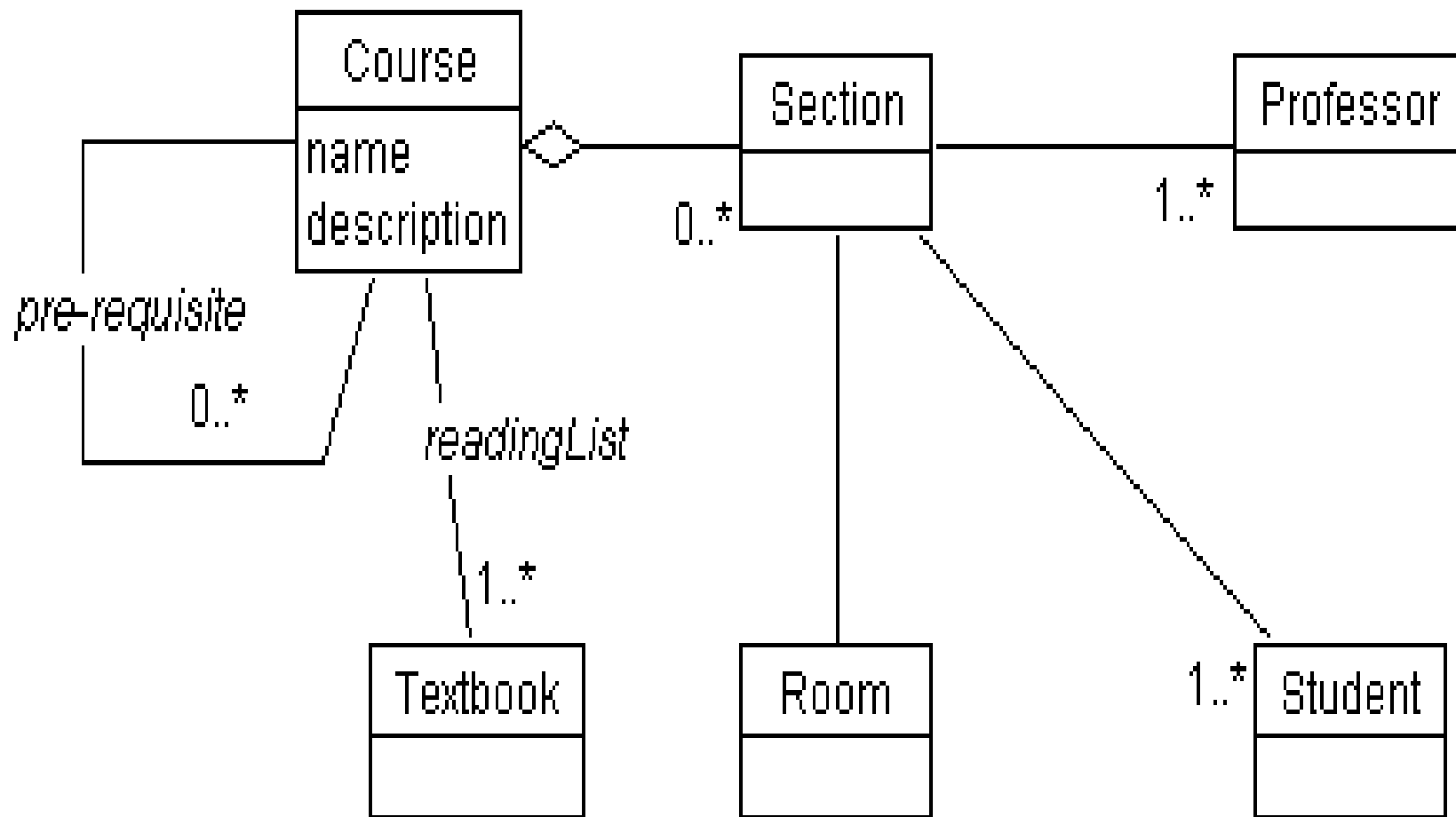
1. Dijagrami klasa

- Skup klasa, interfejsa, i njihovih odnosa
- Najčešći dijagrami u OO modeliranju
- Statički logički pogled

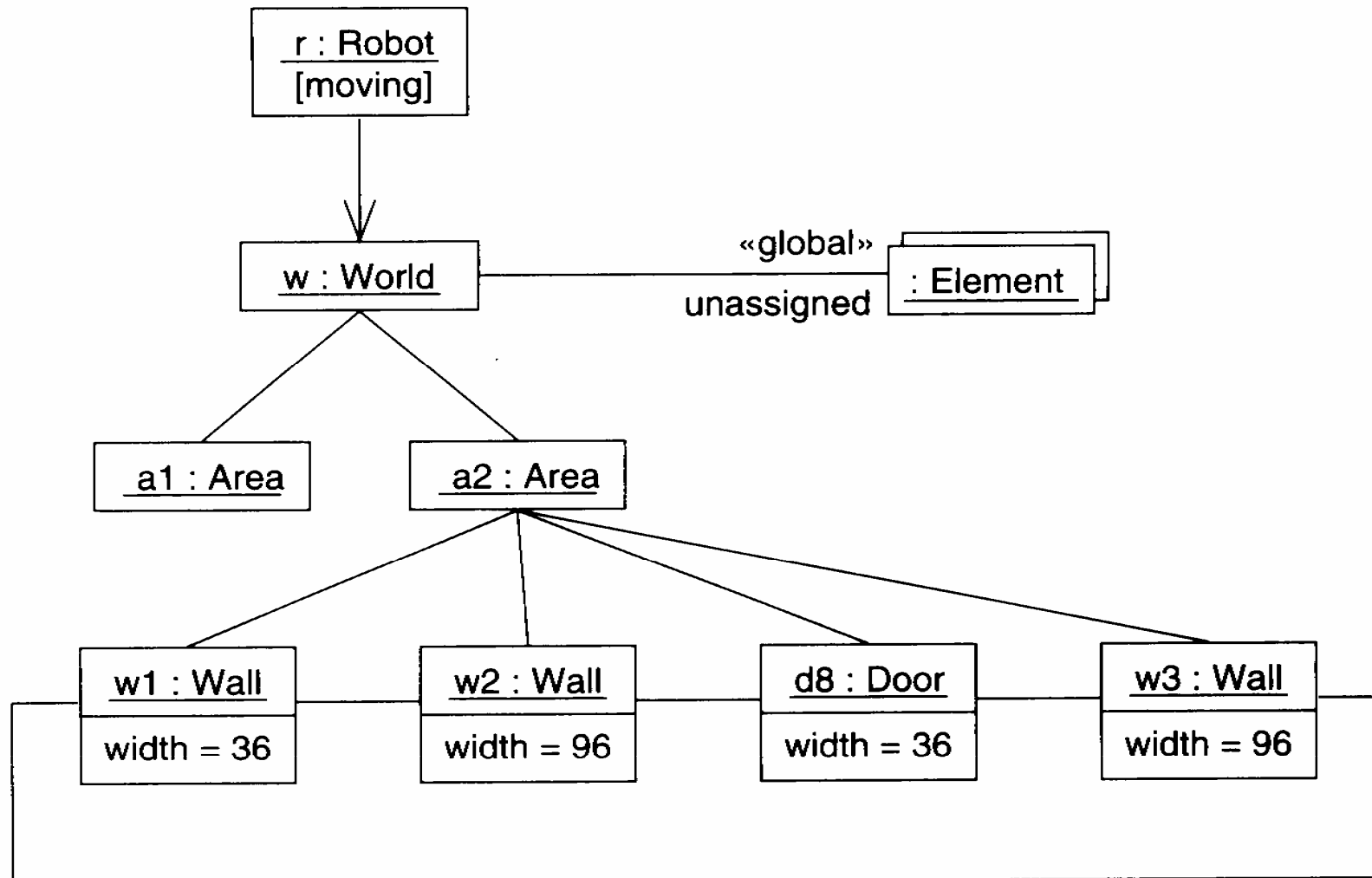
2. Dijagrami objekata

- Skup objekata i njihovih odnosa
- Prikaz instanci elemenata iz dijagrama klasa u jednom trenutku
- Statički logički pogled (realni ili prototipski slučaj)

Dijagram klasa: primer



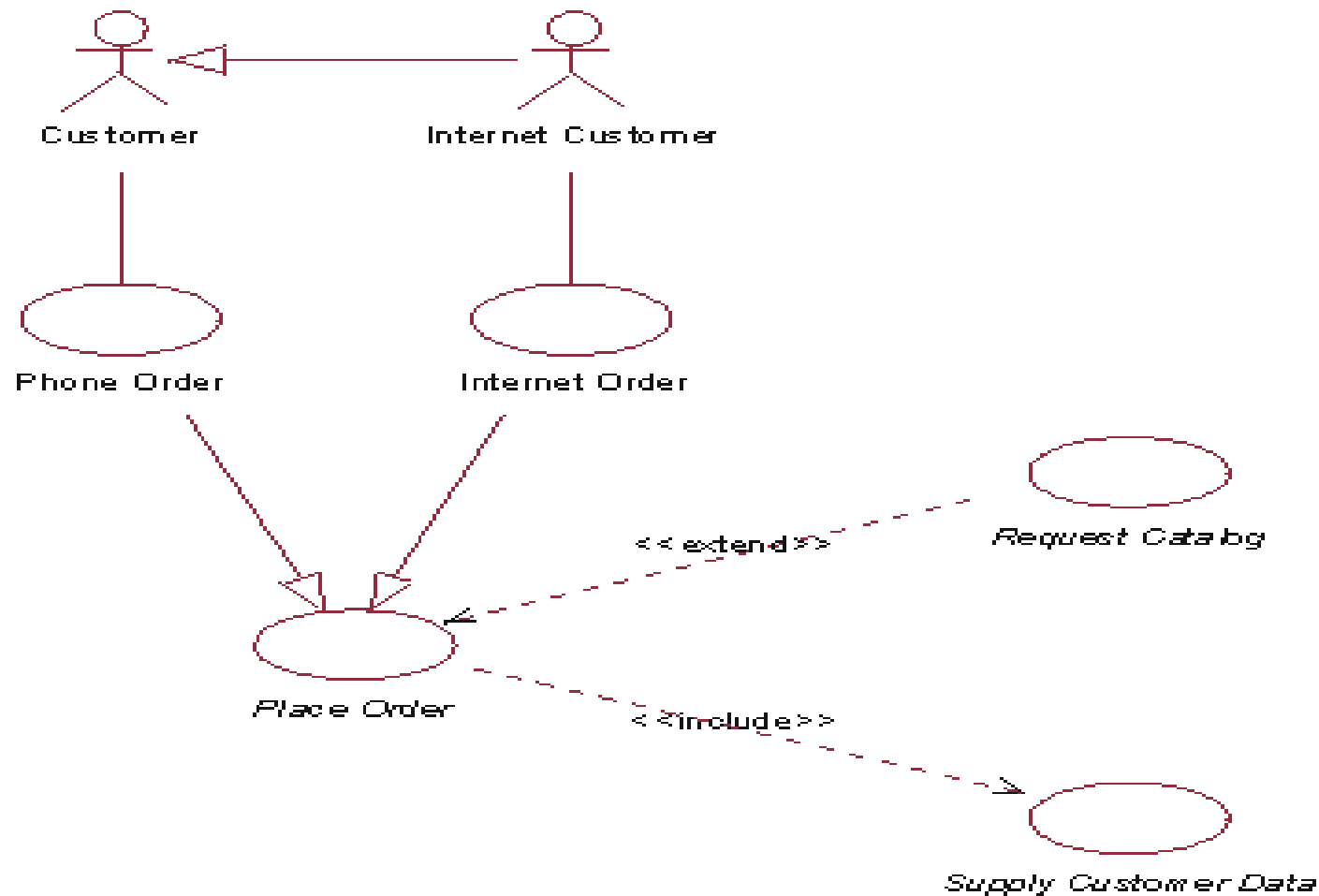
Dijagram objekata: primer



[Dijagrami (nast.)]

- ## 3. Dijagrami slučajeve korišćenja
- Skup slučajeve korišćenja i aktera, učesnika (specijalna vrsta klase) i njihovih odnosa
 - Statički pogled slučajeve korišćenja
 - Poseban značaj u organizovanju i modeliranju ponašanja sistema

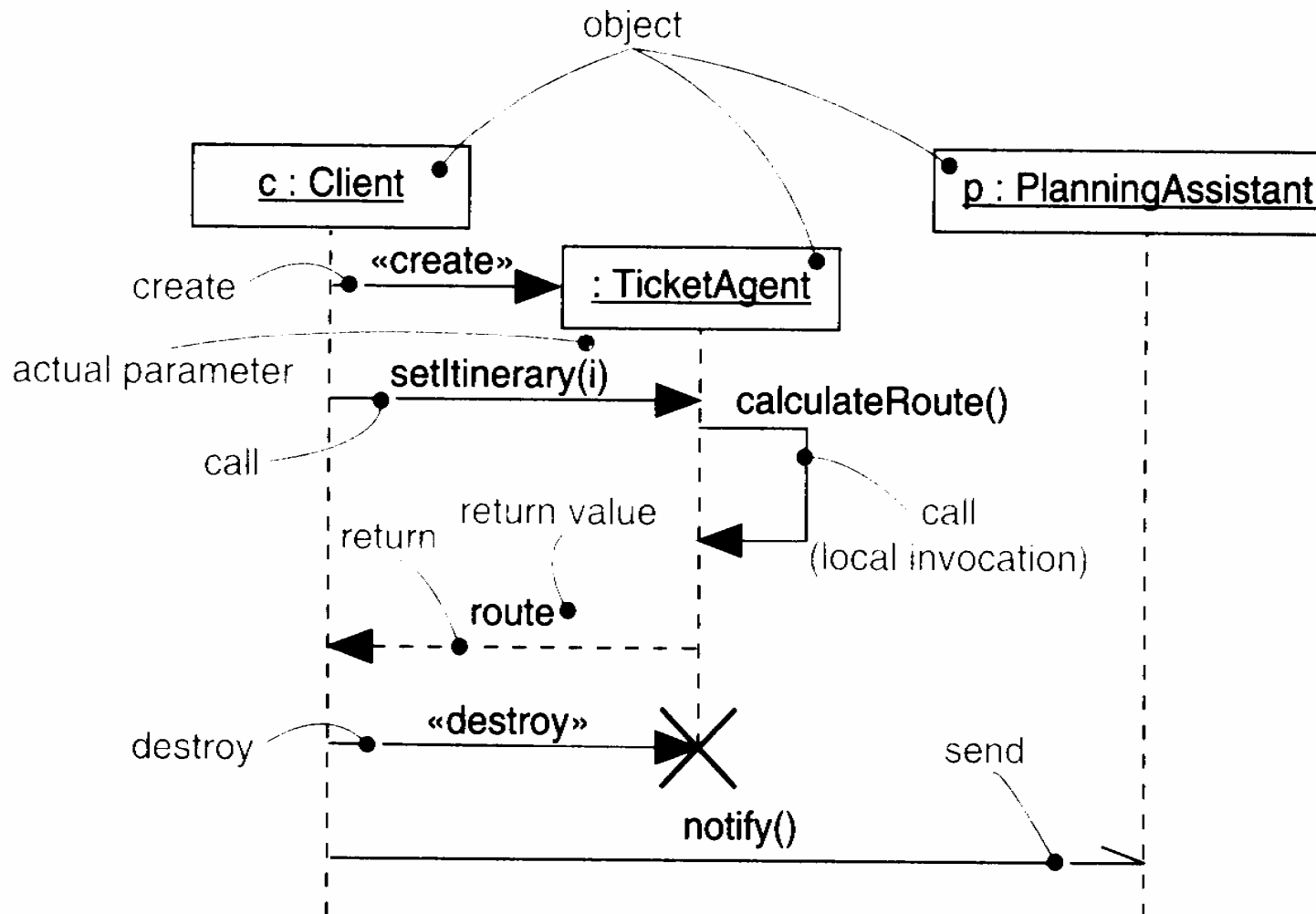
Dijagrami slučajeja korišćenja: primer



Dijagrami (nast.)

4. Dijagrami interakcije – redosleda i saradnje
- Dinamički pogled sistema
 - Objekti, odnosi, poruke
 - Dijagram redosleda
 - Naglašava vremenski redosled poruka
 - Dijagram saradnje
 - Naglašava strukturnu organizaciju objekata koji šalju i primaju poruke
 - Dijagrami redosleda i saradnje izomorfni

Dijagram redosleda: primer



[Dijagram redosleda: primer]

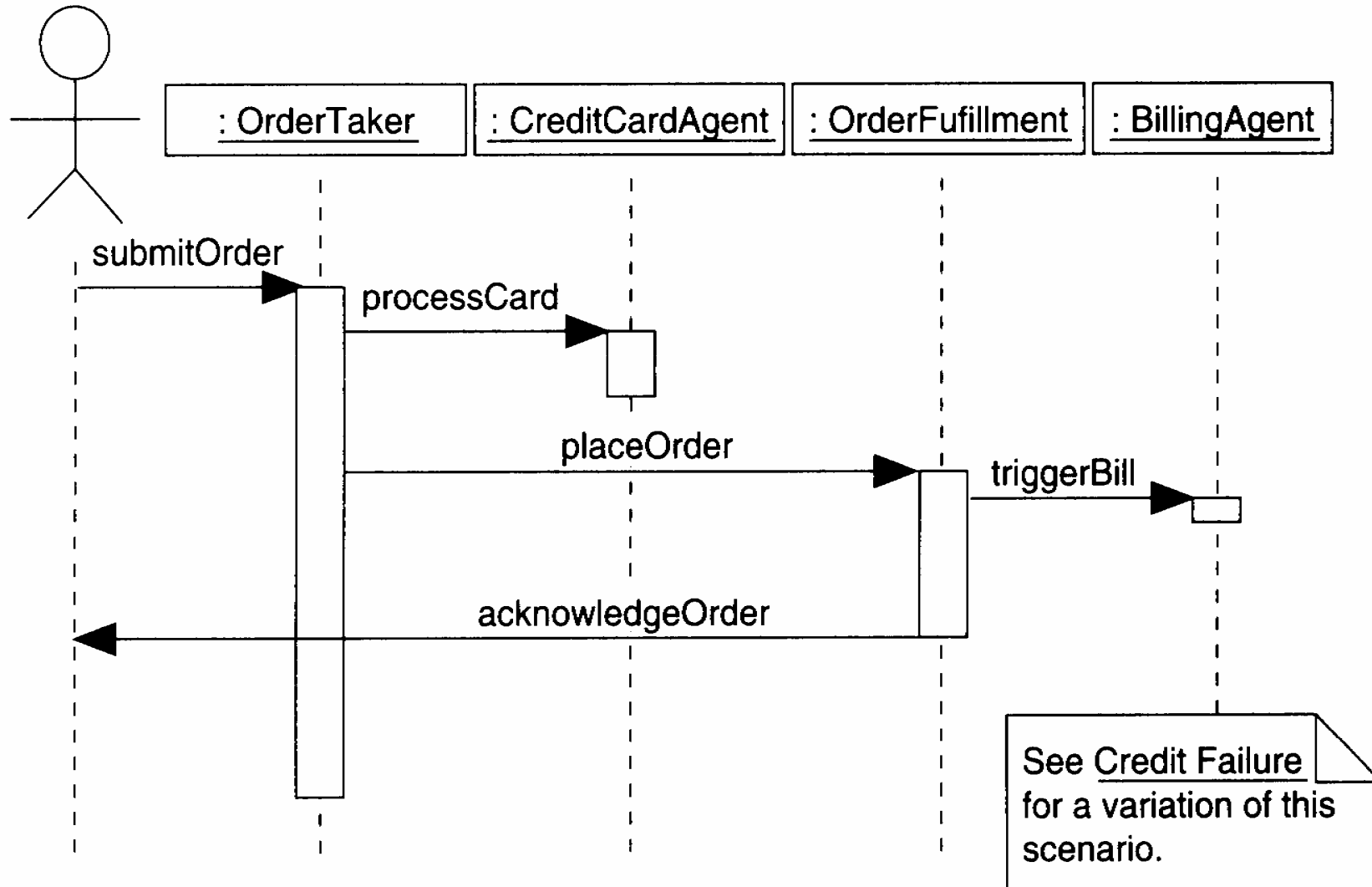
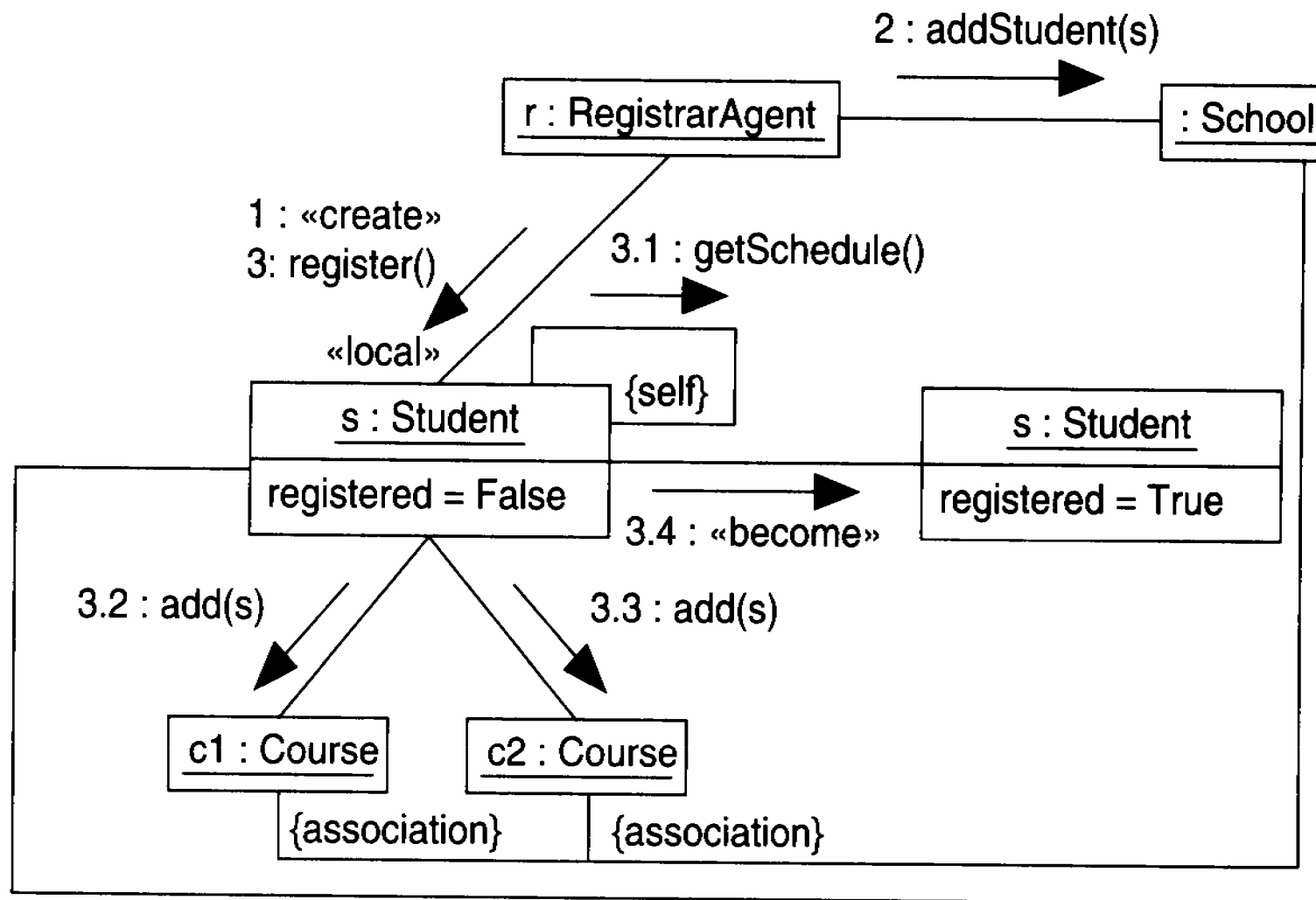


Diagrama saradnie: primer

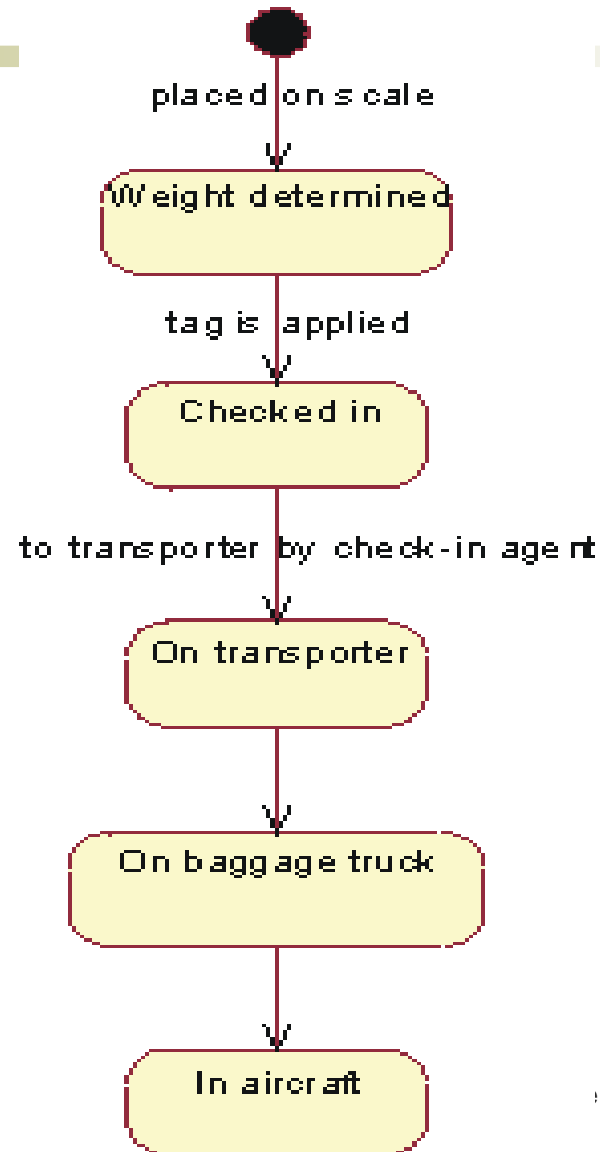


[Dijagrami (nast.)]

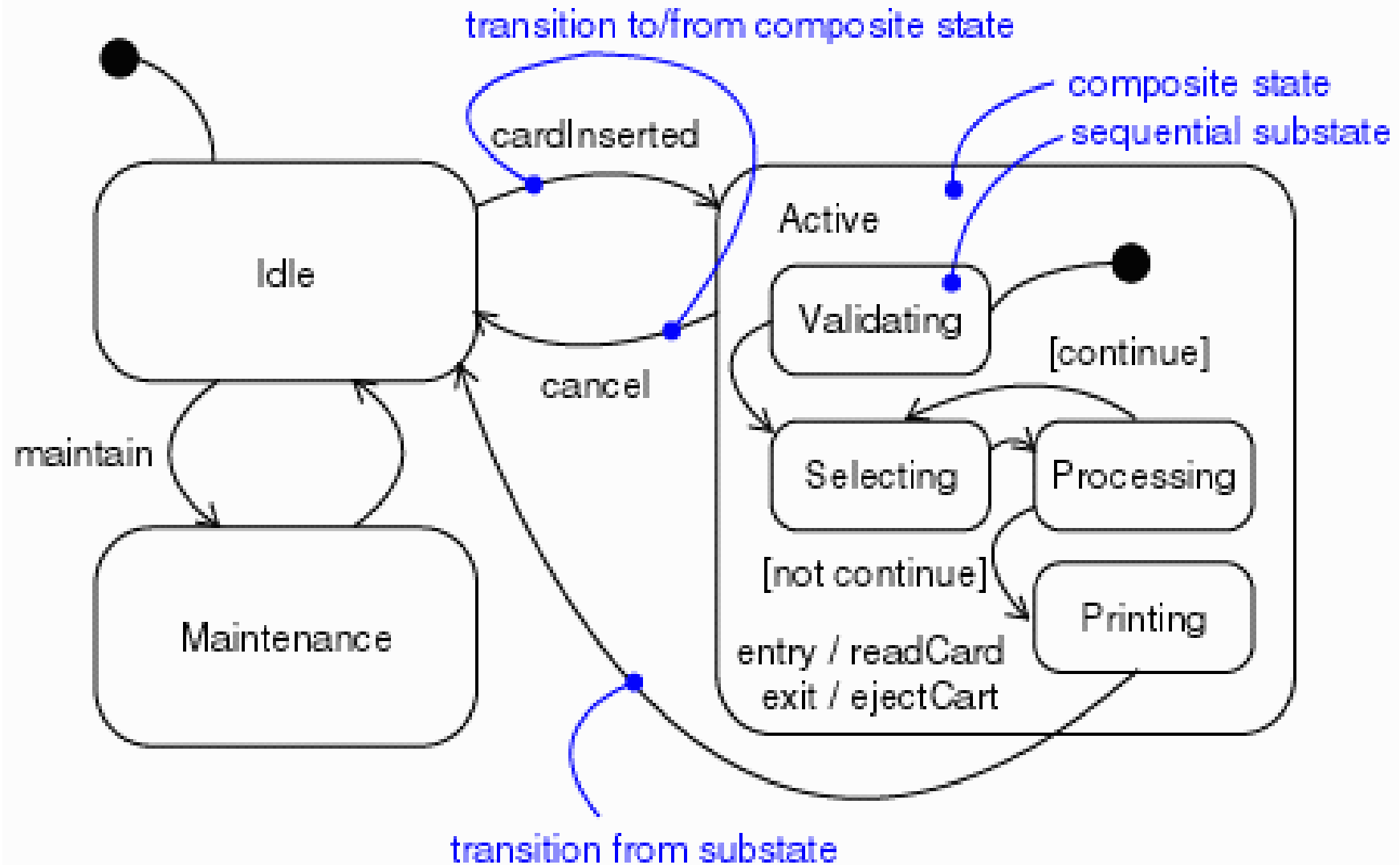
5. Dijagrami stanja

- Mašine stanja: stanja, prelazi, događaji, aktivnosti
- Dinamički pogled sistema
- Značajni za modeliranje ponašanja interfejsa, klase, saradnje
- Naglašavaju ponašanje objekta uređeno događajima

Dijagram stanja: primer



Dijagram stanja: primer



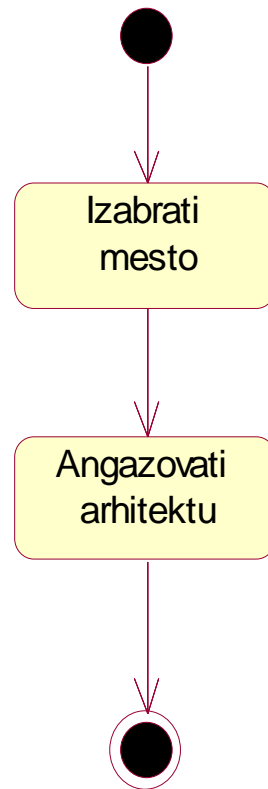
[Dijagrami (nast.)]

6. Dijagram aktivnosti

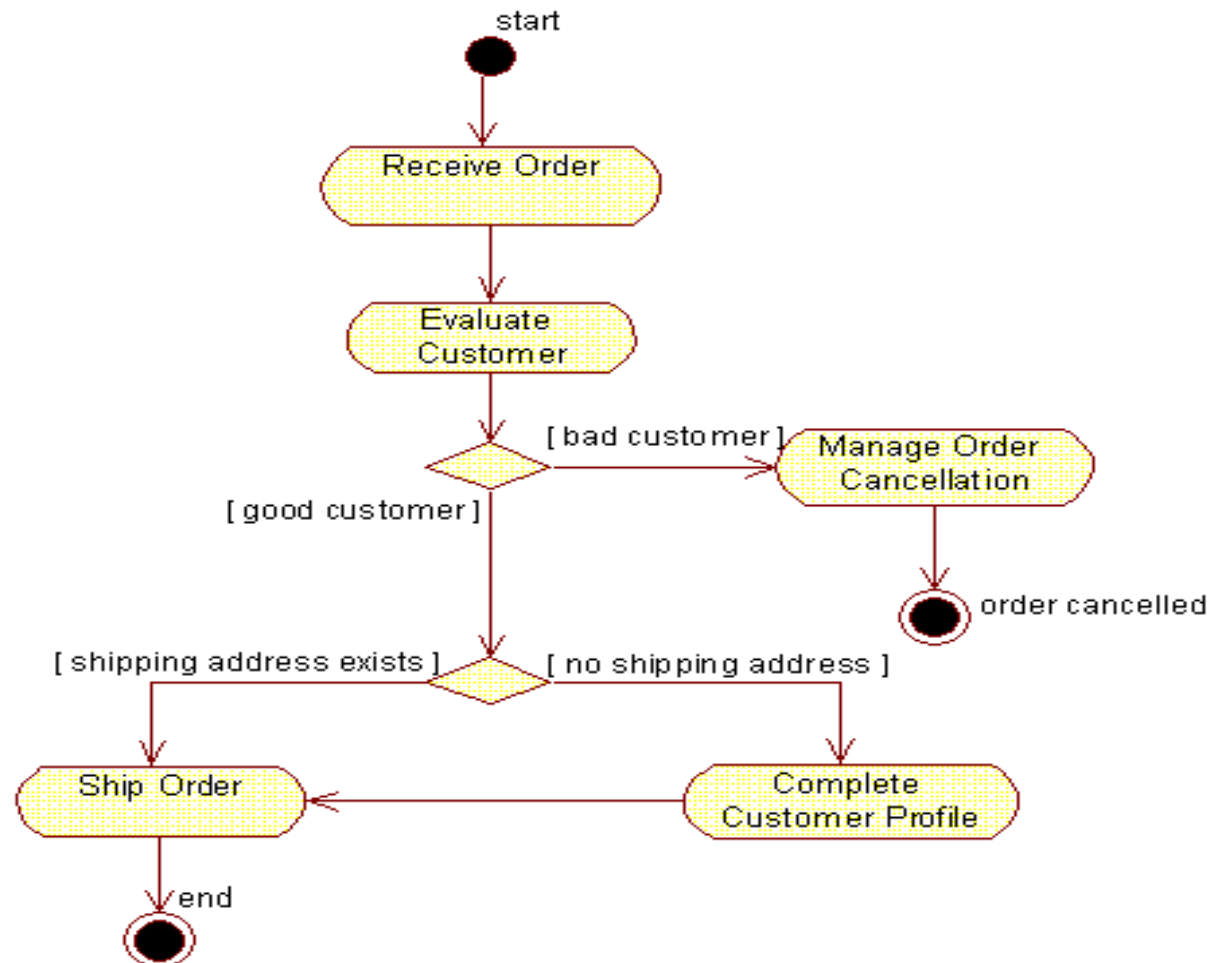
- Varijanta dijagrama stanja
- Tok od jedne do druge aktivnosti
- Značajni u modeliranju funkcije sistema
- Naglašava tok kontrole između objekata

[Dijagram aktivnosti: primer]

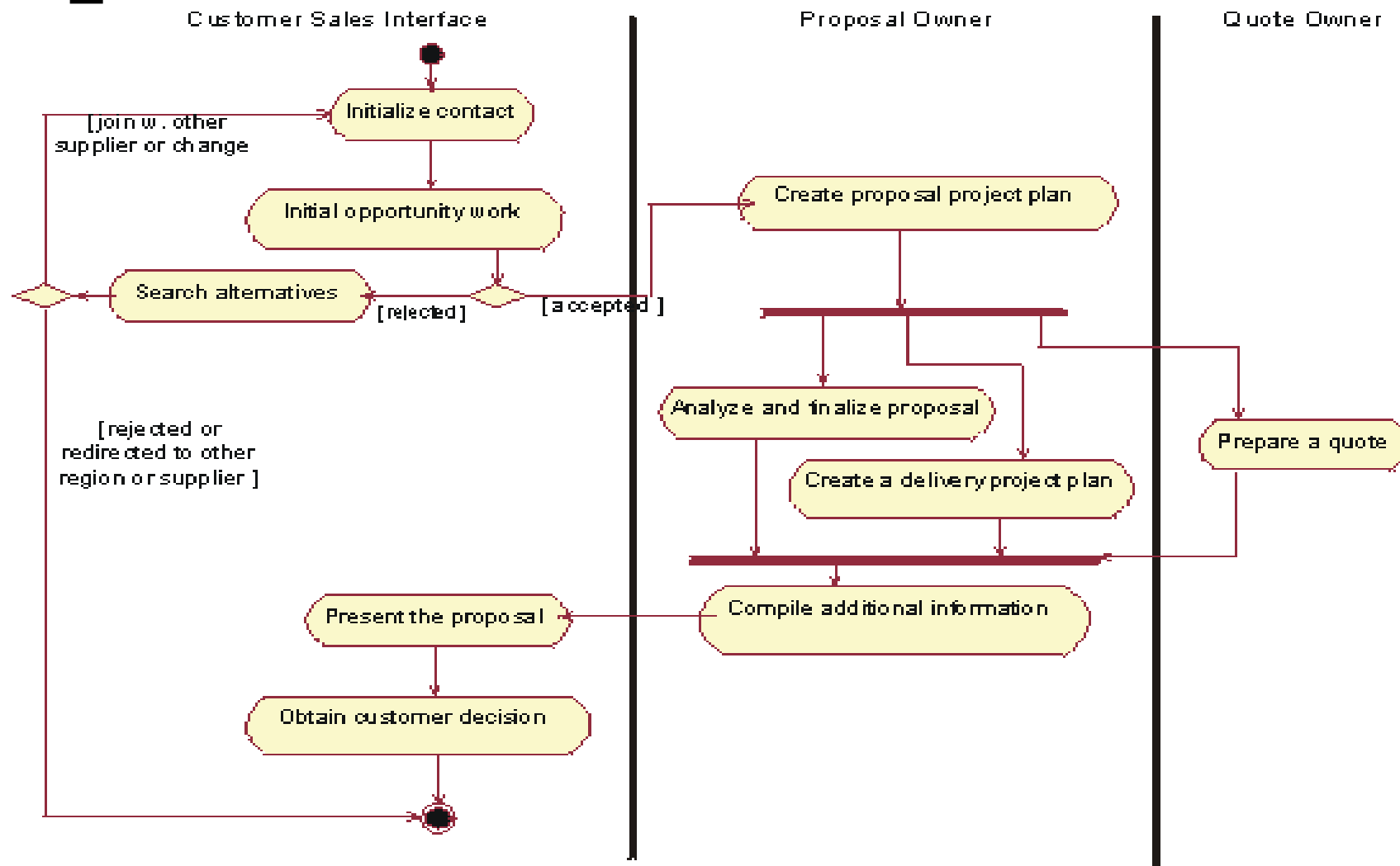
- Bezuslovni prelaz



Dijagram aktivnosti: grananje



Dijagram aktivnosti: razdvajanje i spajanje

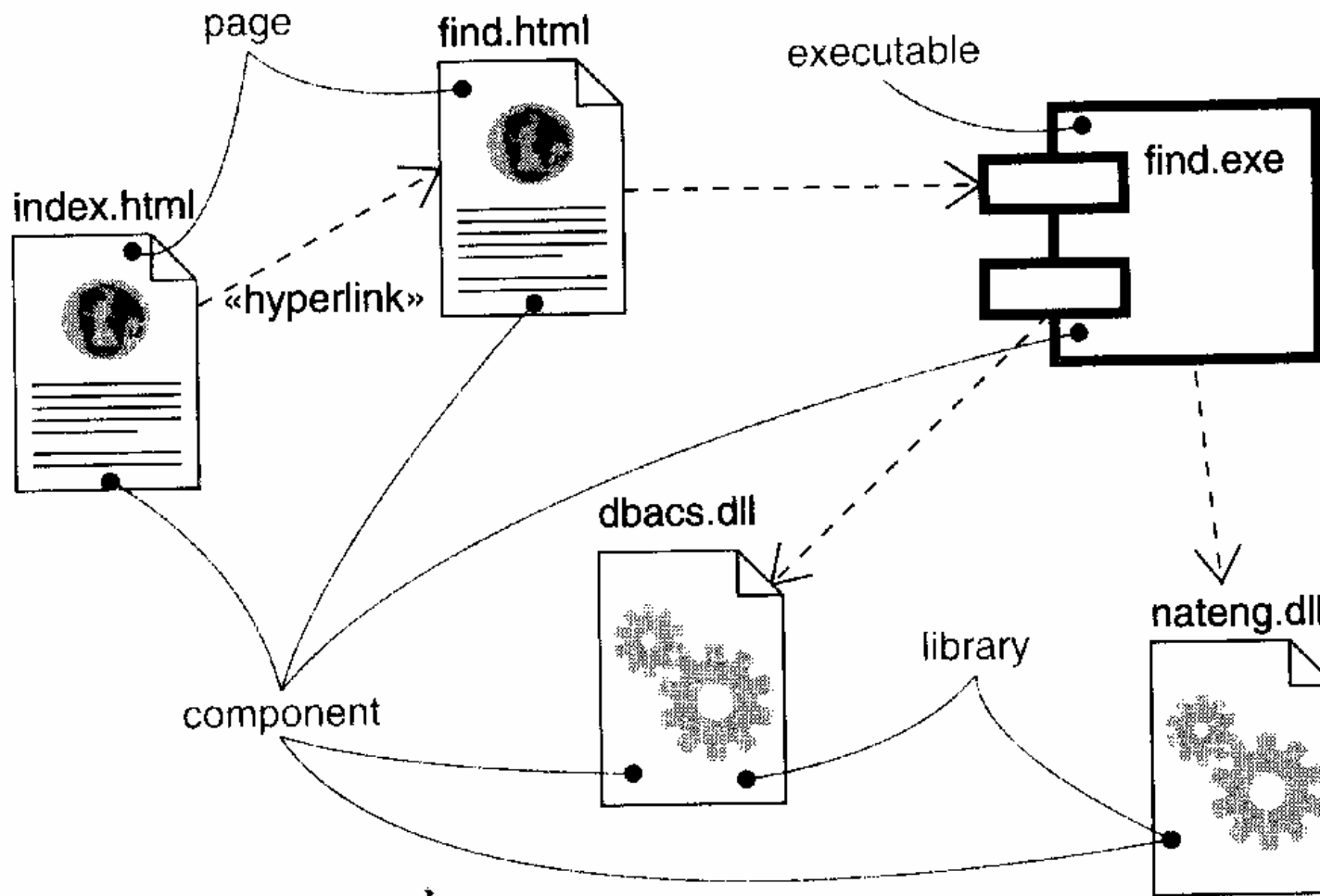


[Dijagrami (nast.)]

7. Dijagram komponenti

- Odnosi u skupu komponenti
- Statički pogled implementacije
- Komponenta obično implementira jednu ili više klasa ili interfejsa

Dijagram komponenti: primer

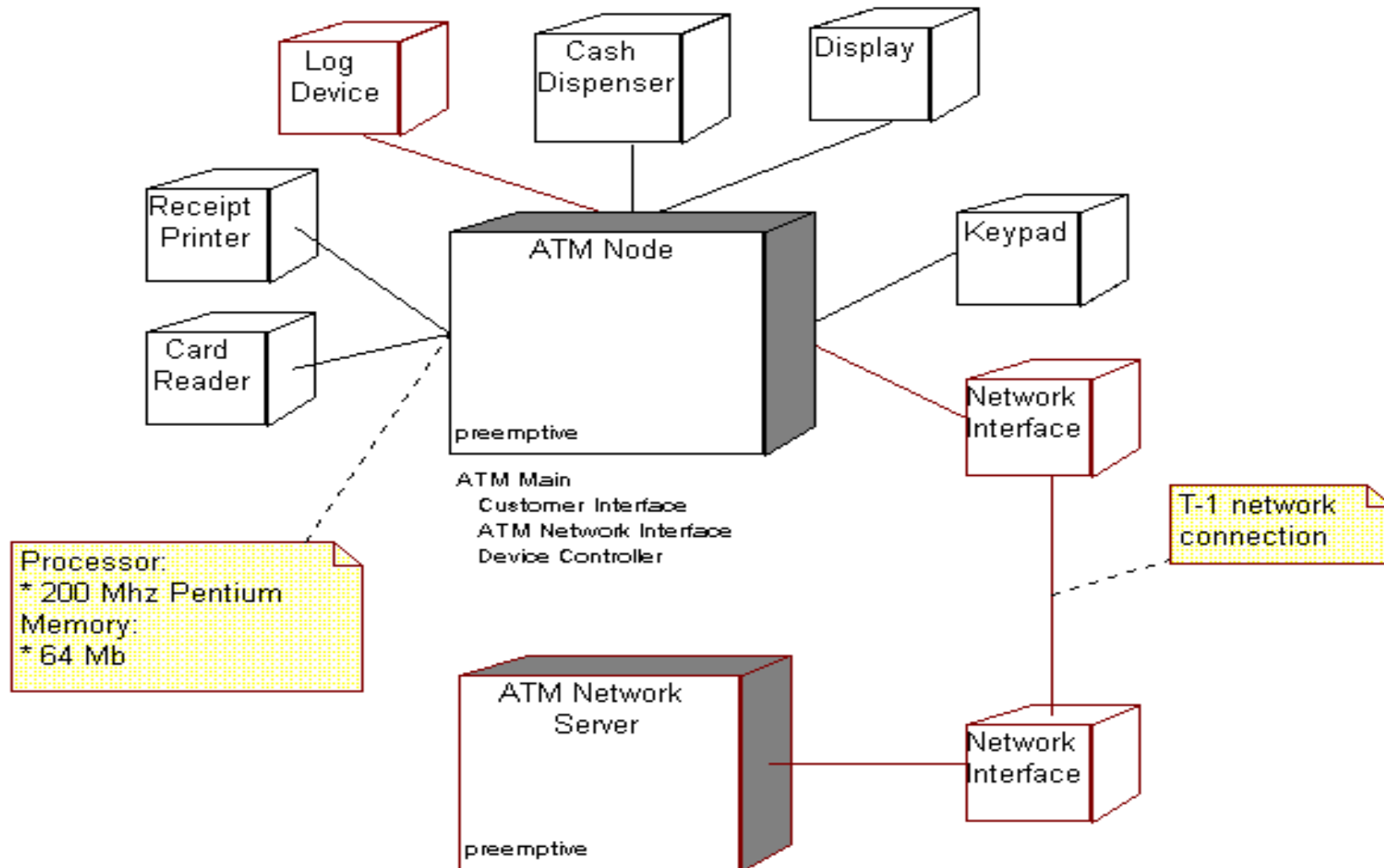


[Dijagrami (nast.)]

8. Dijagrami razmeštaja

- Skup čvorova i njihovih odnosa
- Konfiguracija čvorova obrade u vreme izvršavanja
- Statički pogled razmeštaja arhitekture
- U vezi sa dijagramima komponenti (čvor obično sadrži komponente)

Dijagrami razmeštaja: primer



[UML – opšti mehanizmi]

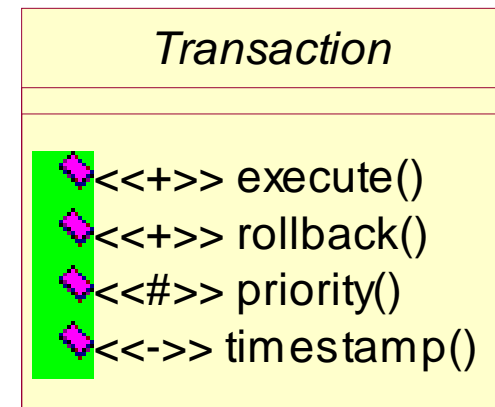
- Specifikacija
- Označavanje (ukrašavanje)
- Opšte podele
- Mehanizmi proširivosti

Opšti mehanizmi: specifikacija

- UML – više od grafičkog jezika
- Iza svakog grafičkog elementa – specifikacija: tekstualni zapis sintakse i semantike tog elementa
- Ikona za klasu: specifikacija sa punim skupom atributa, operacija (sa punom signaturom) i ponašanjem
- Ikona – samo deo specifikacije
- Drugi pogled iste klase – različit deo specifikacije
- Detalji sistema
- Generisanje koda i modela (direktno i inverzno)

[Opšti mehanizmi: ukrašavanje]

- Dodaci osnovnom simbolu
- Grafički i tekstualni
- Primer: klasa
 - apstraktna
 - vidljivost

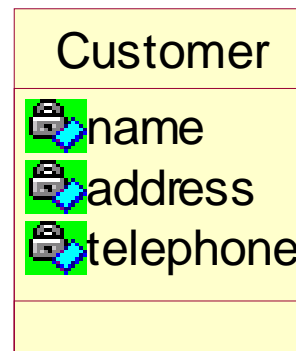


[Opšti mehanizmi: opšte podele]

- Apstraktno / konkretno

- klasa / objekat

- Primer:



Marko:Customer

Jelena

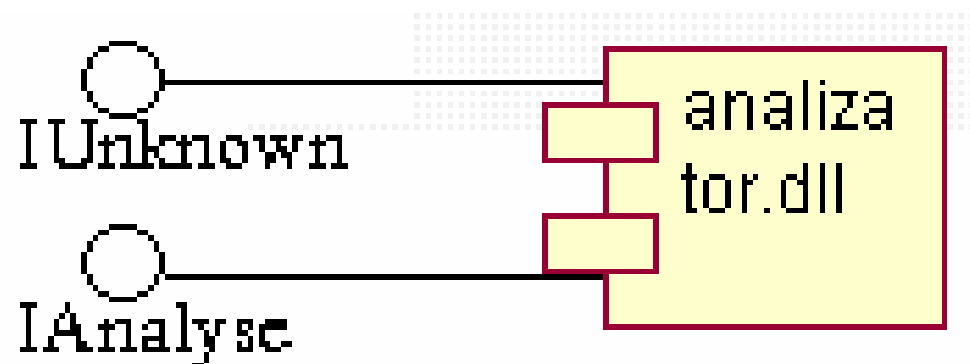
:Customer

- Slučaj korišćenja / instanca slučaja korišćenja

- komponenta / instanca komponente, itd.

Opšti mehanizmi: opšte podele (nast.)

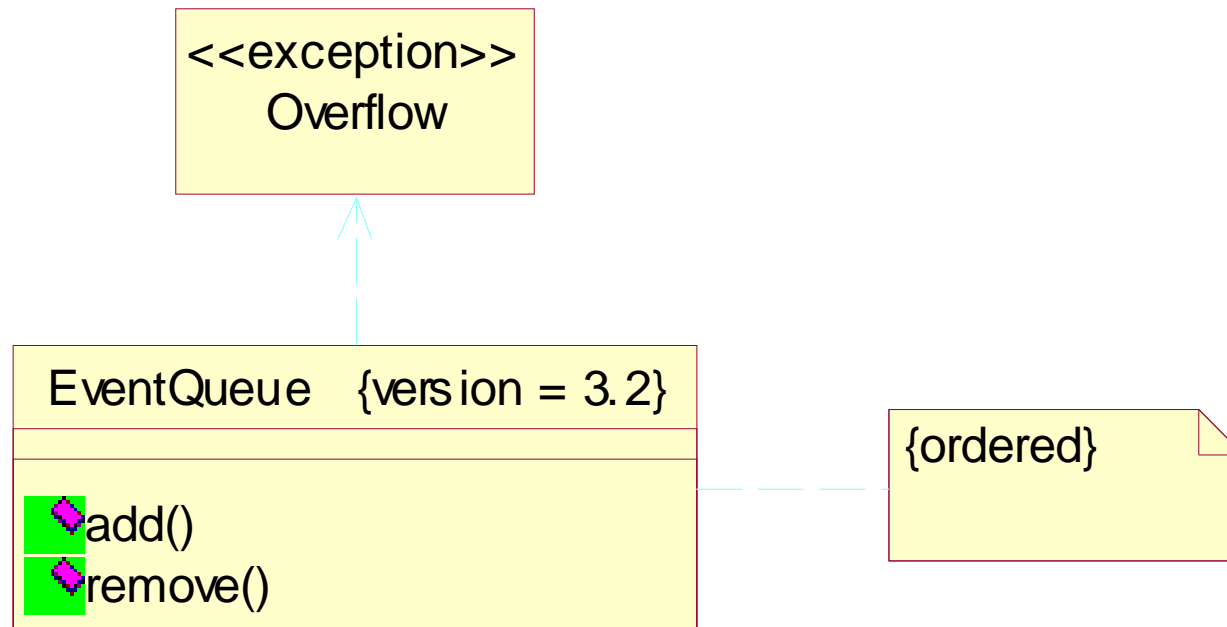
- interfejs / implementacija
 - ugovor / realizacija ugovora
 - operacija / metoda



Opšti mehanizmi: mehanizmi proširivosti

- Stereotipi
 - Proširuju rečnik UML-a
 - Primer: izuzetak – specijalna vrsta klase
- Označene vrednosti values
- Ograničenja

Opšti mehanizmi: mehanizmi proširivosti: primer



Proces (metodologija) razvoja softvera: RUP

- UML **NIJE** proces razvoja softvera (SDP)
- Nezavisan od izabranog SDP
- UML – Najprilagođeniji procesu koji je
 - Vođen slučajevima korišćenja
 - Orijetisan prema arhitekturi
 - Iterativan i inkrementalan
- RUP (Rational Unified Process)

Proces razvoja softvera: RUP (nast.)

- Vođen slučajevima korišćenja
 - Željeno ponašanje sistema
 - Verifikovanje i validacija arhitekture sistema
 - Testiranje
 - Komuniciranje

Proces razvoja softvera: RUP (nast.)

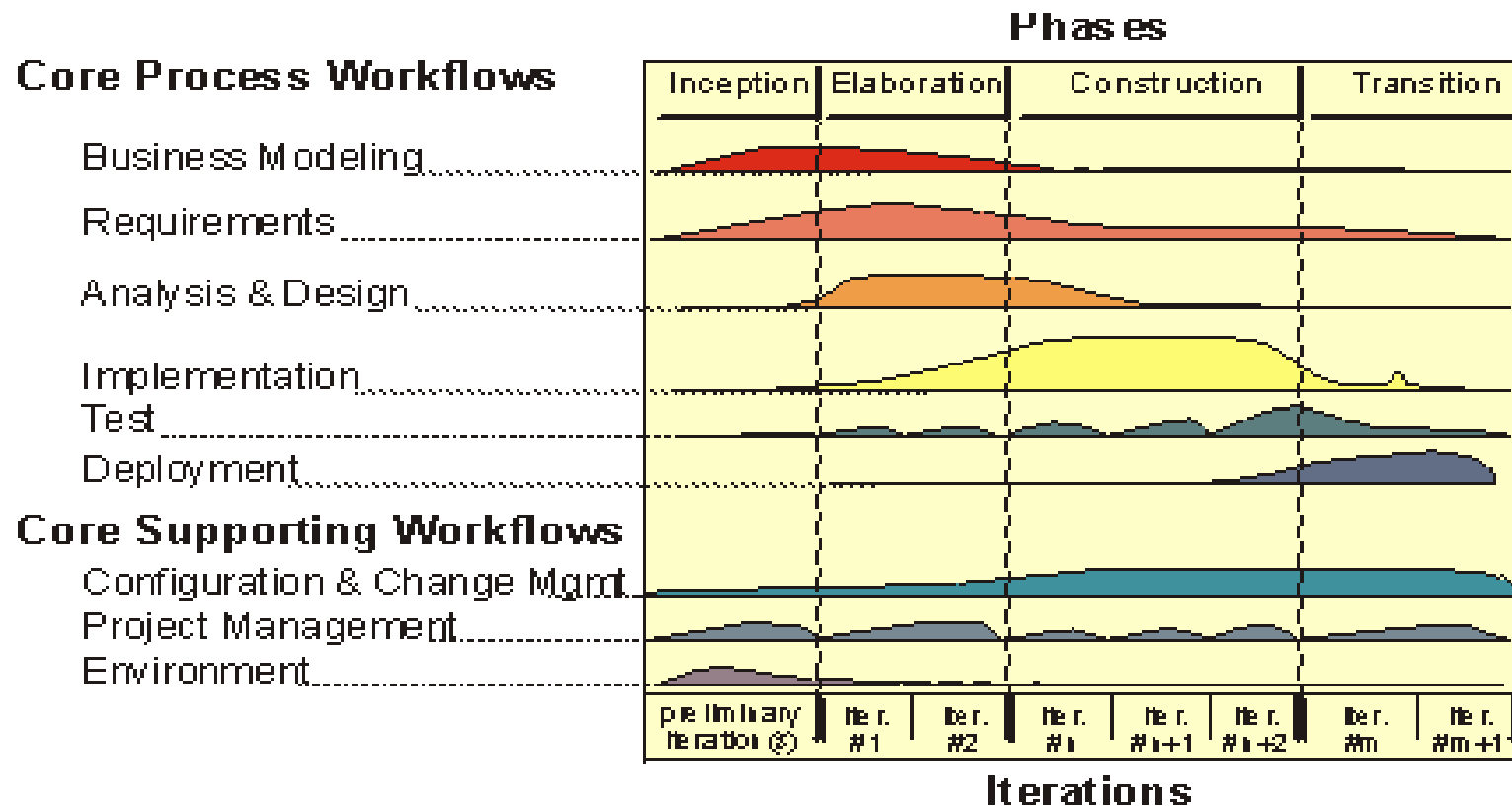
- Orijentisan prema arhitekturi
 - Izbor strukturnih elemenata od kojih se sistem sastoji, i njihovih interfejsa
 - Izbor elemenata ponašanja
 - Grupisanje strukturnih elemenata i elemenata ponašanja u podsisteme koji progresivno rastu
 - Ponovna upotreba

Proces razvoja softvera: RUP (nast.)

- Iterativan i inkrementalan
 - Kontinualna integracija systemske arhitekture
 - Prototipovi, verzije
 - Vođen procenom rizika

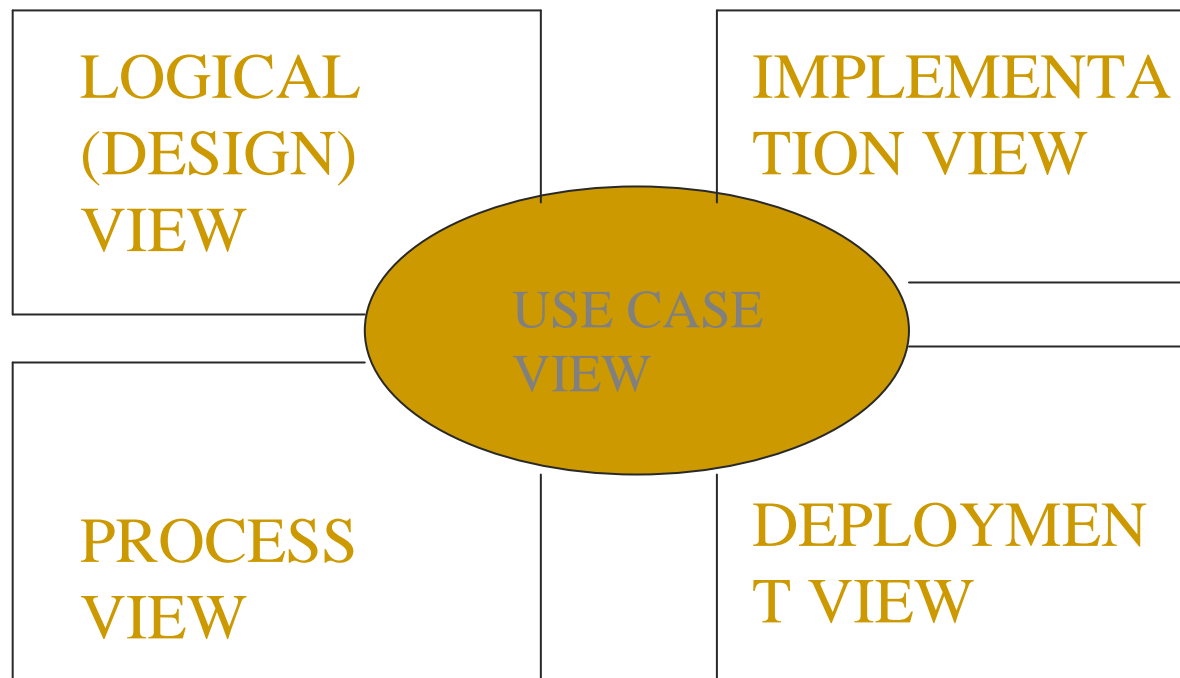
Proces razvoja softvera: RUP (nast.)

- Faze životnog ciklusa razvoja razvoja



Modeliranje arhitekture

- Arhitektura softverskog sistema
- Pogledi: samostalni / interaguju



Modeliranje arhitekture: pogled slučajeve korišćenja

- Krajnji korisnici, analitičari, testeri
- Statički aspekt sistema:
 - Dijagrami slučajeve upotrebe
- Dinamički aspekti sistema
 - Dijagrami interakcije
 - Dijagrami stanja
 - Dijagrami aktivnosti

Modeliranje arhitekture: logički pogled

- Klase, interfejsi
- Funkcionalni zahtevi
- Statički aspekt:
 - Dijagrami klasa, objekata
- Dinamički aspekti:
 - Dijagrami interakcije
 - Dijagrami stanja, aktivnosti

Modeliranje arhitekture: pogled procesa

- Konkurentnost i sinhronizacija
- Procesi
- Performanse
- Propusnost (throughput) sistema
- Dijagrami kao u logičkom pogledu
- Aktivne klase
- Multitasking okruženje

Modeliranje arhitekture: pogled implementacije

- Komponente
- Datoteke
- Fizički sistem
- Statički aspekt: dijagram komponenti
- Dinamiški aspekt: kao u prethodnim pogledima

Modeliranje arhitekture: pogled razmeštaja

- Čvorovi
- Hardverska topologija sistema
- Distribucija
- Isporuka
- Instalacija delova fizičkog sistema
- Statički aspekt: dijagrami razmeštaja
- Dinamički aspekt: dijagrami interakcije, stanja i aktivnosti

Modeliranje sistema korišćenjem UML-a

- Produktivnost, efikasnost, kvalitet
- Informacioni sistemi velikih preuzeća
- Bankarstvo i finansije
- Transport
- Odbrana / vazdušni prostor
- Maloprodaja
- Medicinska elektronika
- Nauka
- Distribuirane Web usluge, itd.

Modeliranje sistema korišćenjem UML-a: primeri

- Integralni komorski informacioni sistem
- PKJ / regionalne komore
- Sagledavanje, elaboracija I deo faze konstrukcije
- UML notacija

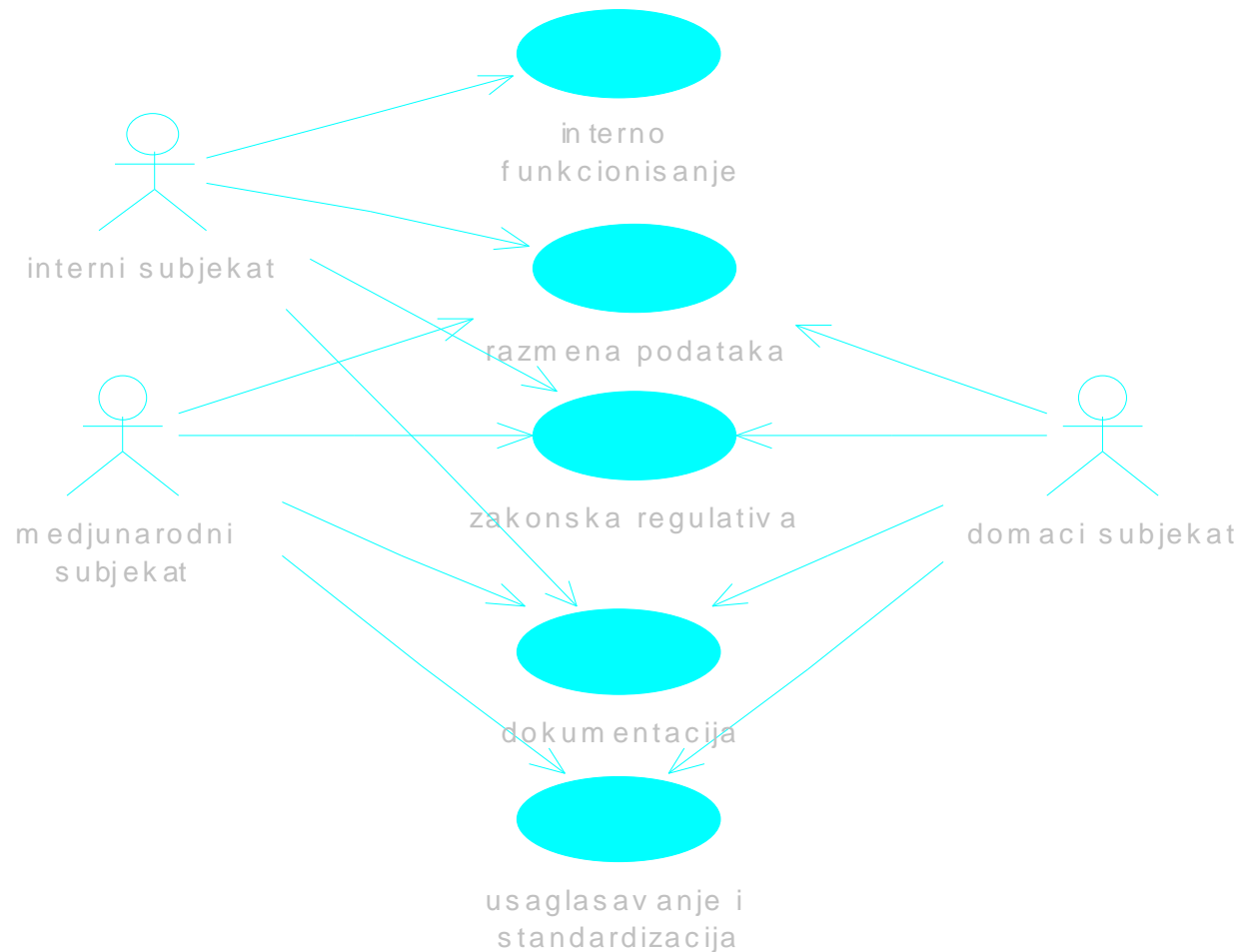
Modeliranje sistema korišćenjem UML-a: primer

- Trinaest podsistema, npr. ekonomski odnosi sa inostranstvom, poslovni podsistem, pravni podsistem, itd.

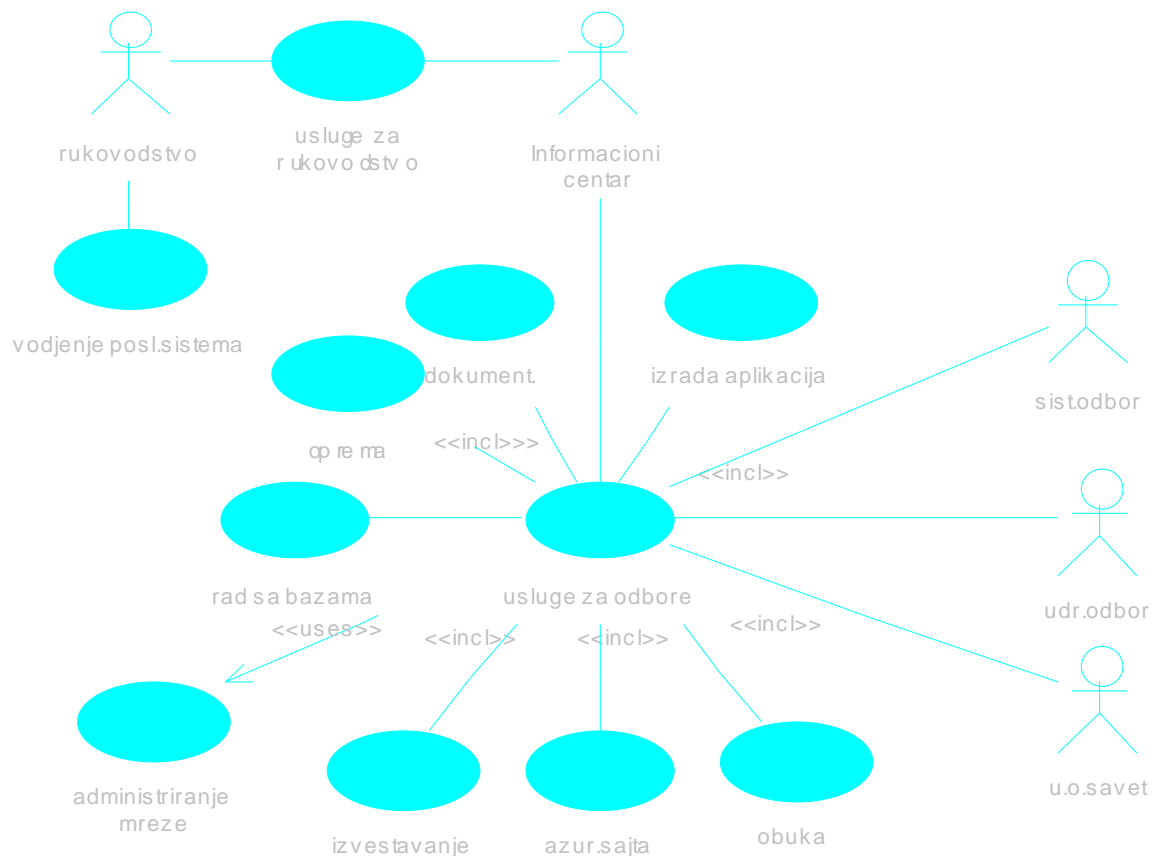
Modeliranje sistema korišćenjem UML-a: primer (nast.)

- Specifikacija zahteva, analiza:
 - Pogled slučajeve korišćenja (dijagrami slučajeve korišćenja, sekvenci, aktivnosti)
 - Logički pogled
- Projektovanje
 - Logički pogled (dijagrami klasa, saradnje)
- Implementacija (dijagrami komponenti, saradnje)

Modeliranje sistema korišćenjem UML-a: primer (nast.)



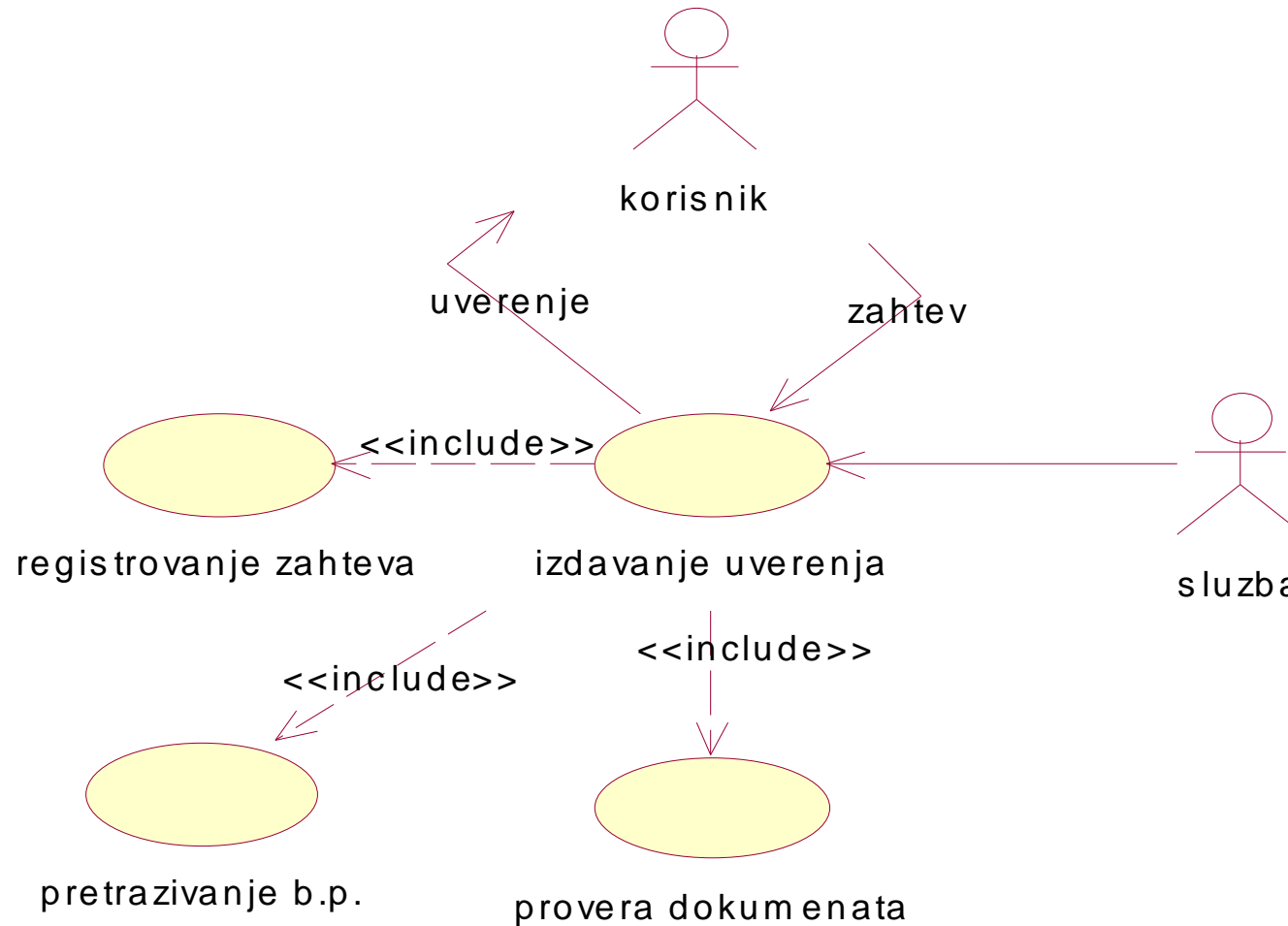
Modeliranje sistema korišćenjem UML-a: primer (nast.)



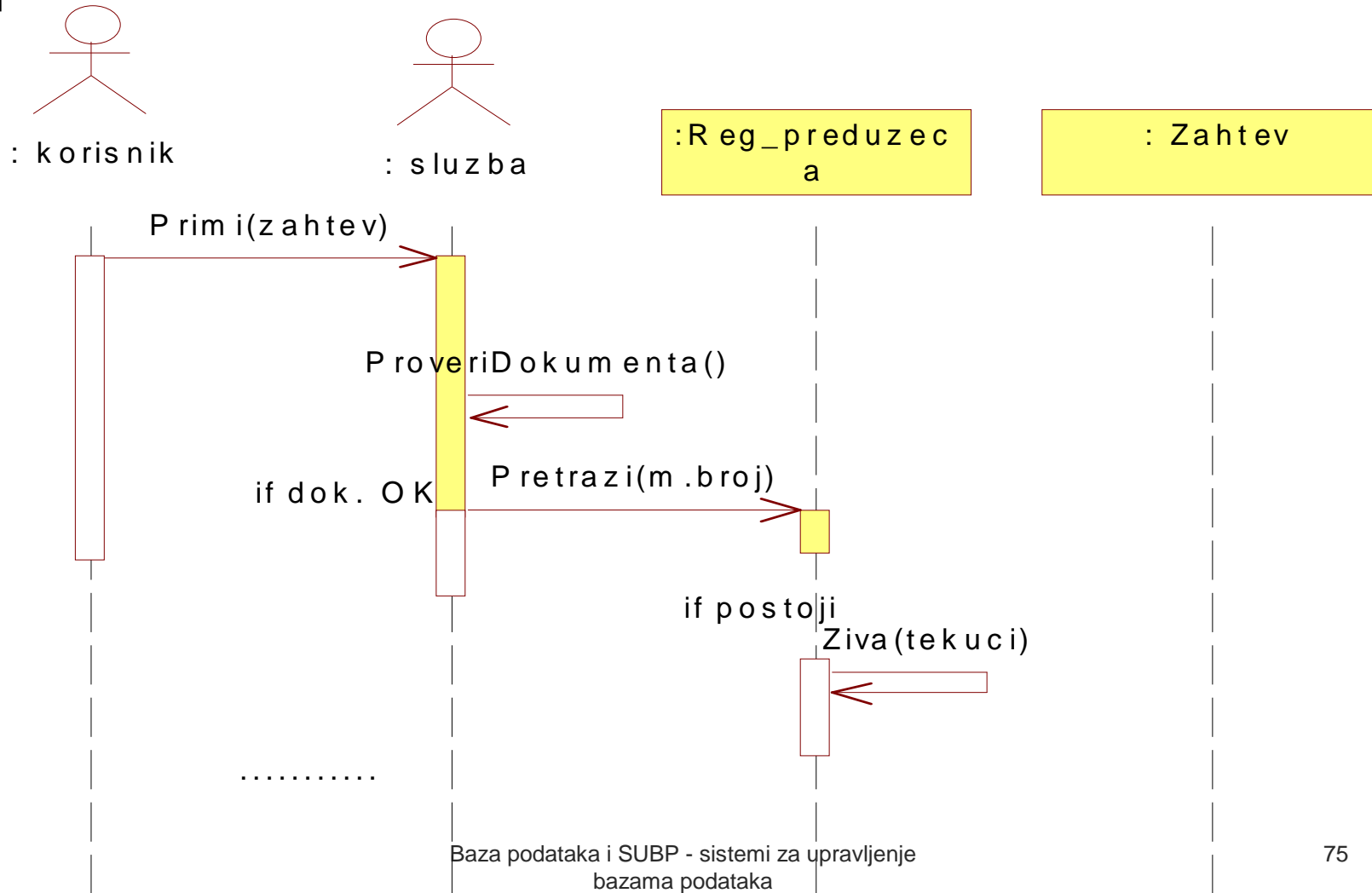
Modeliranje sistema korišćenjem UML-a: primer (nast.)

- Spoljna trgovina
- Izdavanje sertifikata da je kompanija registrovana za posao
- Analiza:
 - Pogled slučajeve korišćenja (dijagrami slučajeve korišćenja, sekvenci, aktivnosti)
 - Logički pogled

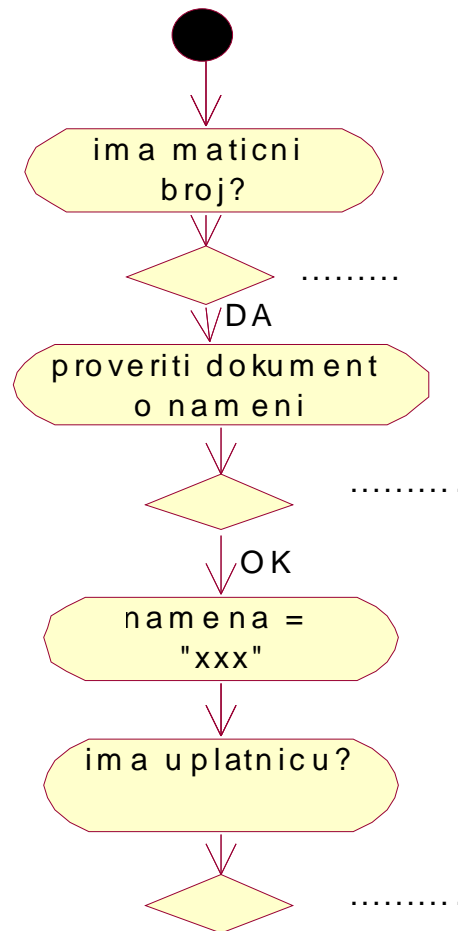
Modeliranje sistema korišćenjem UML-a: use case view – use case dijagram



Modeliranje sistema korišćenjem UML-a: use case view - dijagram sekvenci

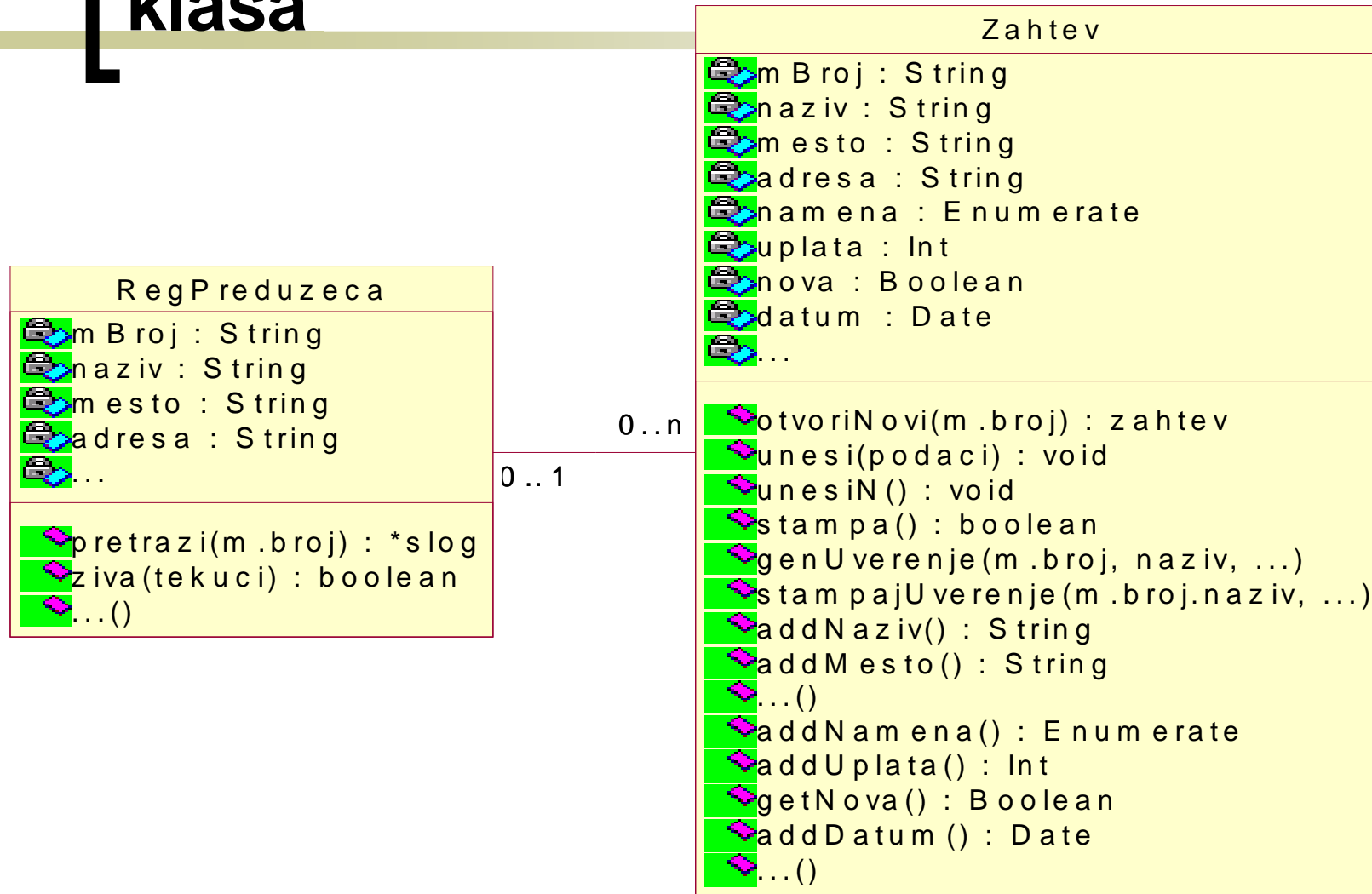


Modeliranje sistema korišćenjem UML-a: use case view – dijagram aktivnosti



Baza podataka i SUBP - sistemi za upravljenje bazama podataka

Modeliranje sistema korišćenjem UML-a: logical view – dijagram klasa



Modeliranje sistema korišćenjem UML-a: primer (nast.)

- Projektovanje
 - Logički pogled (dijagrami saradnje, klasa)

UML – ZAKLJUČAK

- Vizuelizacija
 - Laka komunikacija
 - Grafički jezik
 - Dobro definisana semantika
- Specifikacija
 - Zahtevi
 - Odluke u analizi, projektovanju i implementaciji

UML – ZAKLJUČAK (nast.)

Konstrukcija

- NIJE vizuelni programski jezik
- Direktna veza modela i raznih jezika (Java, C++, Visual Basic, RDB, OO DB)
- Generisanje koda (Forward engineering)
- Generisanje modela (Reverse engineering)

UML – ZAKLJUČAK (nast.)

■ Dokumentovanje

- Izvršni kod
- Zahtevi, arhitektura, projekat, izvorni kod;
- Prototipovi, verzije, projektni planovi