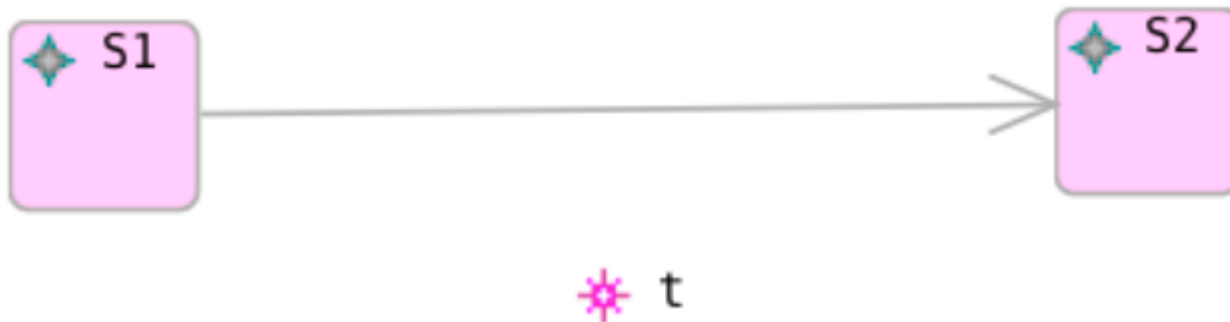


Dijagrami stanja

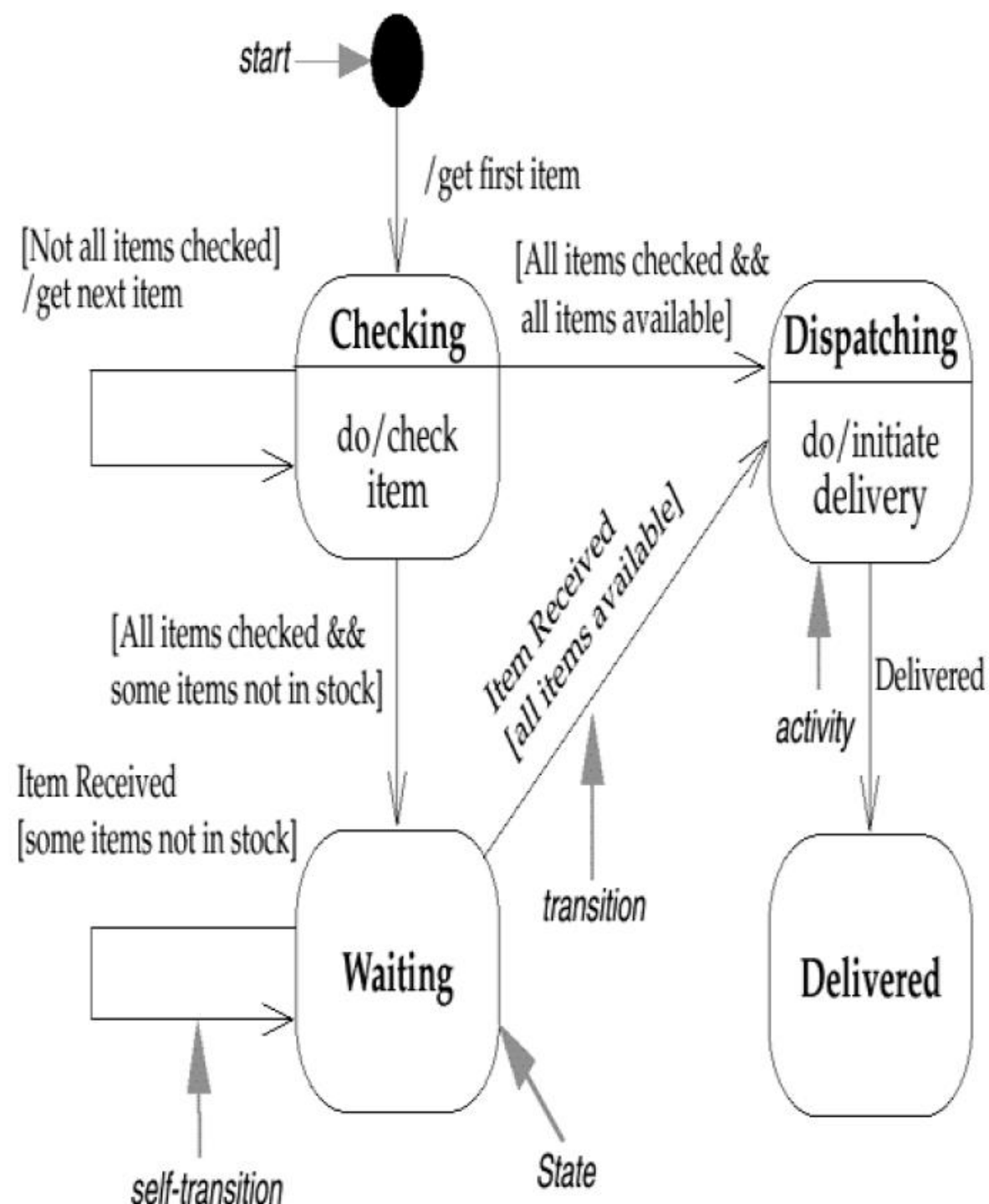
Dijagrami stanja

- Modeluju ponašanje sistema
- Opisuju moguća stanja objekta kao i tranzicije
 - Kontekst klase



Primjer

- Dijagram prikazuje različita stanja narudžbenice (order processing system)
- Start point
- Naziv inicijalne tranzicije je /get first item
 - Sintaksa *Event [Guard] / Action*
- Specifikacija aktivnosti unutar stanja Checking
 - *do/activity*



Action vs Activity

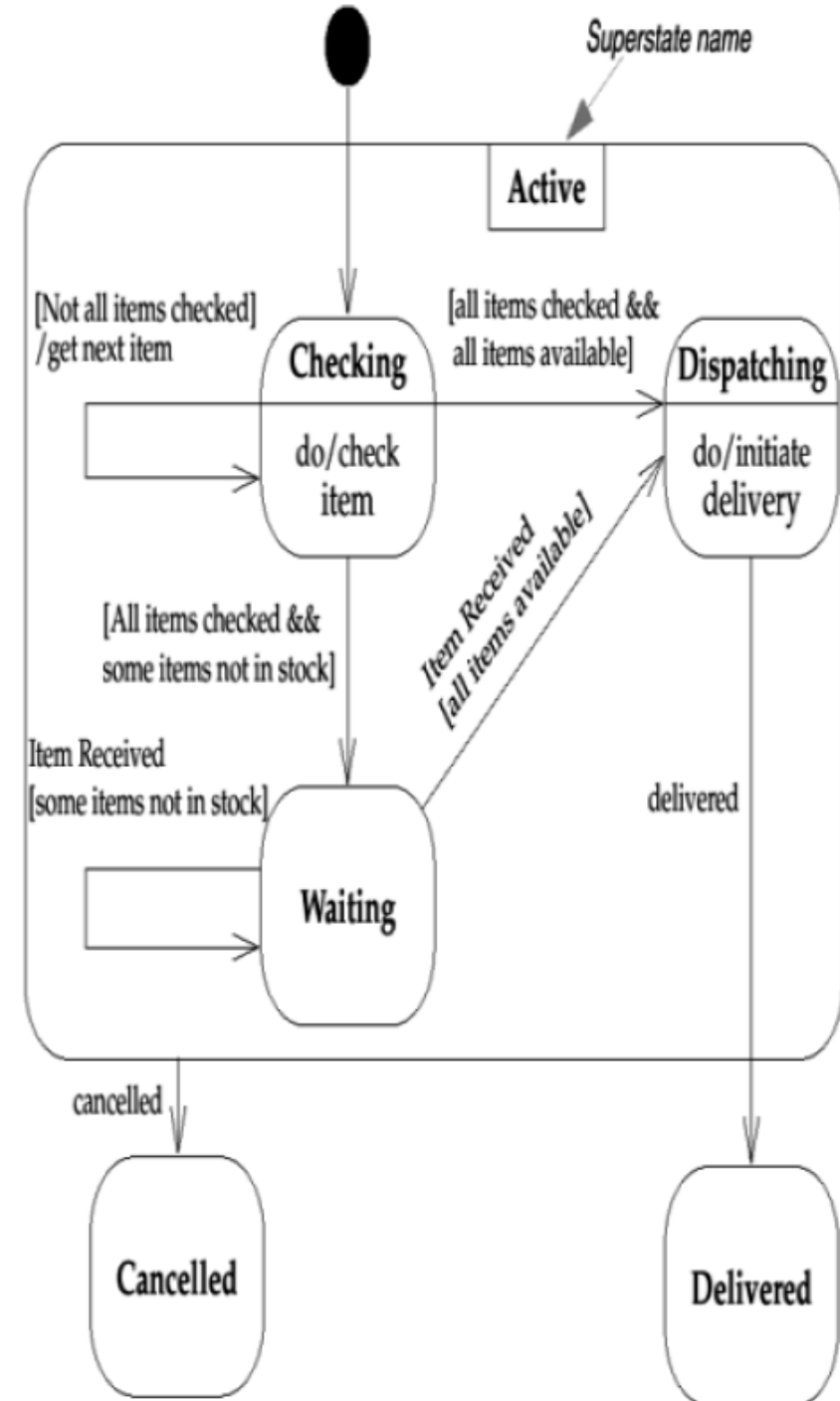
- Implementirane kao metode odgovarajuće klase, ali se ipak tretiraju posebno
 - Akcije se pridružuju tranzicijama, tretiraju se kao kratke i neprekidne operacije
 - Aktivnosti su se pridružuju stanjima, imaju duže trajanje i mogu da budu prekidane
- Kada tranzicija nema definisani *Event* dešava se implicinto poslije završetka bilo koje aktivnosti unutar tekućeg stanja
- Guard je logički uslov za tranziciju,
 - Moraju biti međusobno isključivi kada se odnose na isto stanje

Primjer 2

- Waiting state
 - Nema definisanih aktivnosti, za izlazak iz ovog stanja potreban je događaj *Item Received*
 - Provjeravaju se uslovi i prelazi se u Dispatching stanje ili se ostaje u Waiting stanju
- Dispatching state
 - Aktivnost *initiate delivery*
 - Tranzicija bezuslovna triggerovana sa *Delivered* događajem, ali ne i sa završetkom aktivnosti *initiate delivery*

Super-stanje

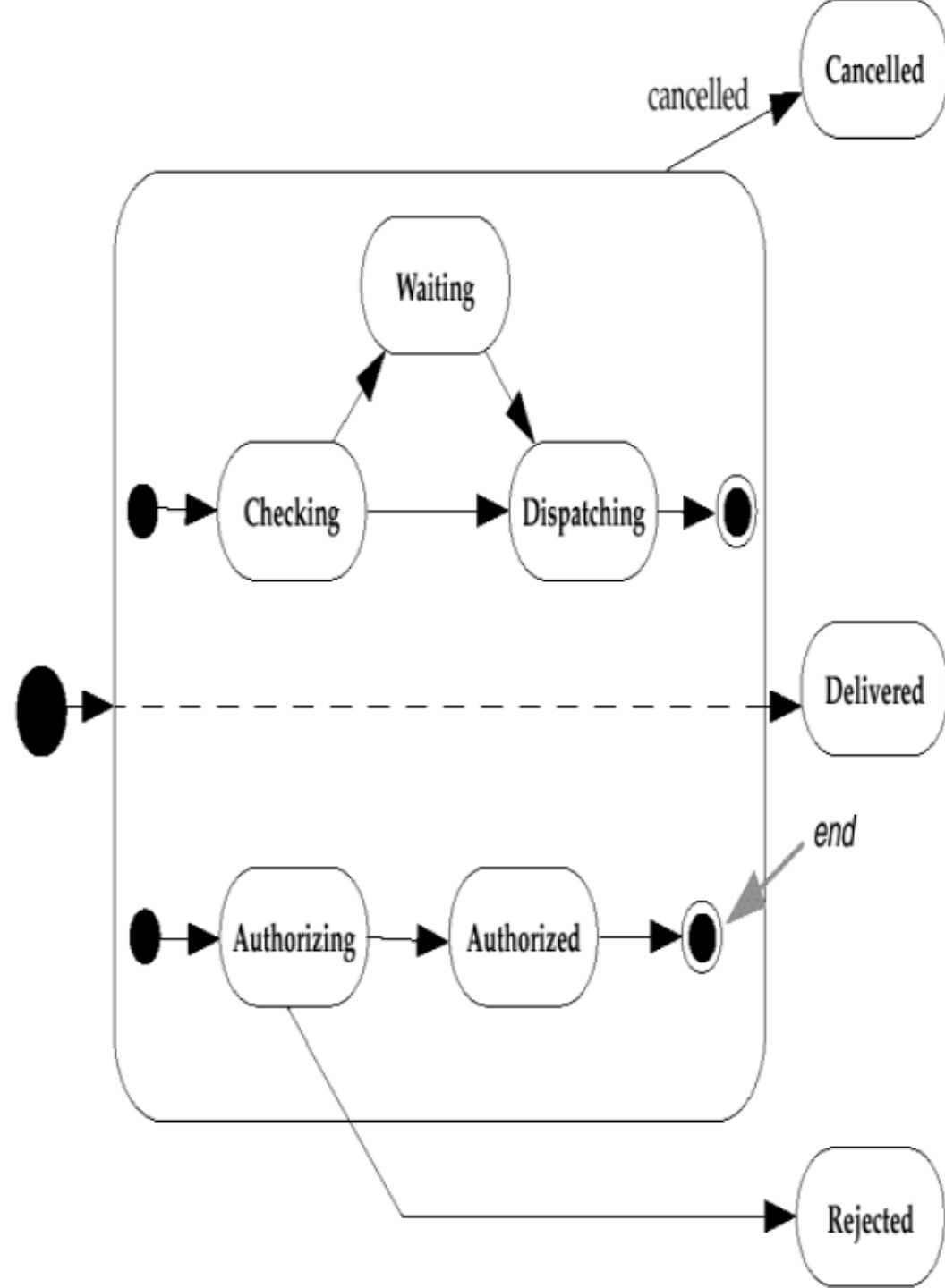
- Podstanja nasljeđuju tranzicije odgovarajućeg nadstanja



Dopunski elementi unutar stanja

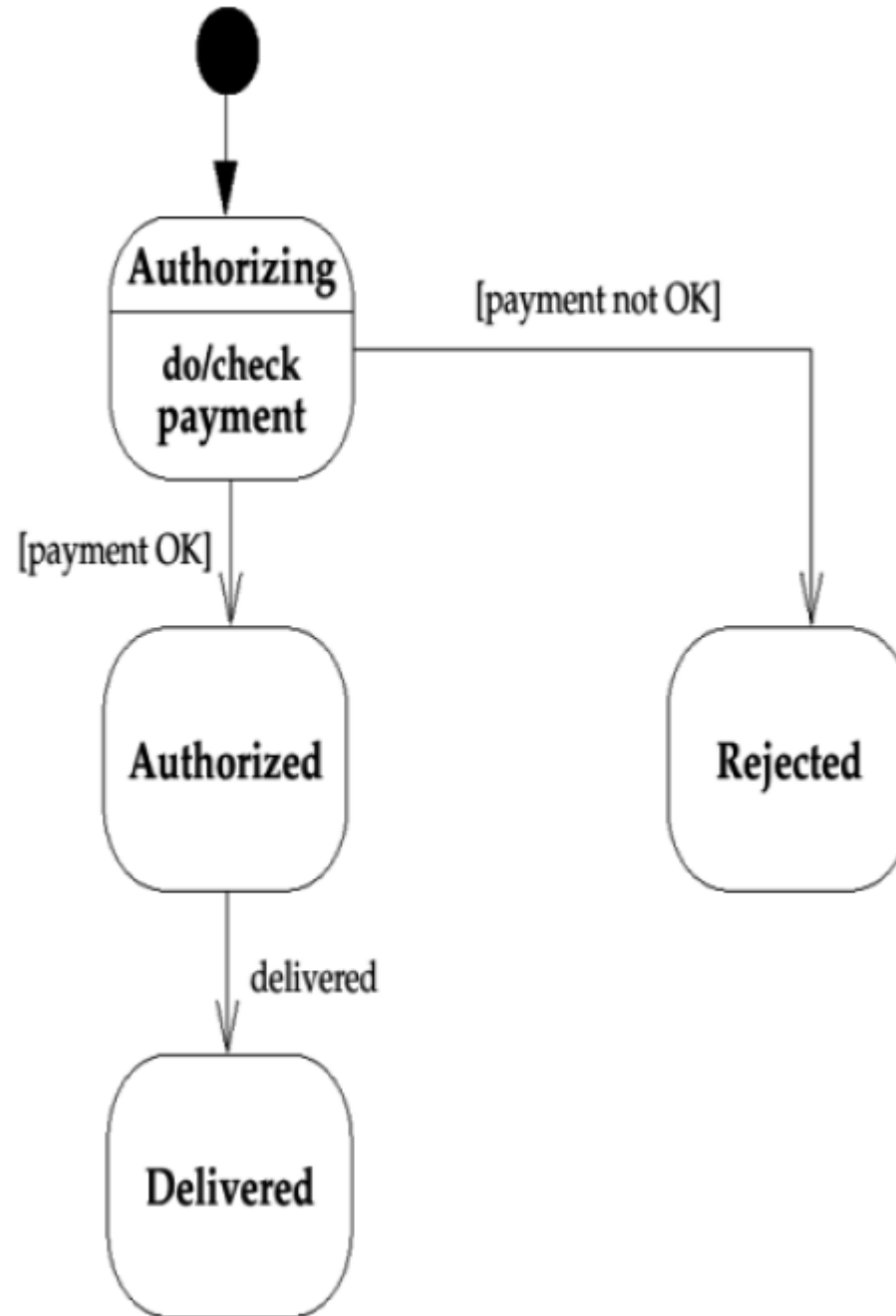
- Akcije koje ne prouzrokuju tranzicije, reakcija na događaj spolja
 - *eventName / actionName*
- Događaji
 - Dešava se poslije određenog vremena, *after (20 minuta)*
 - Dešava se kada specifični uslov bude ispunjen, *when (temperature > 100)*
 - *Entry event / action*
 - *Exit event / action*
 - Za self-transition redosljed je: exit action, transition action, entry action

Konkurentnost 2



Konkurentnost

- Motivacija



Preporuke

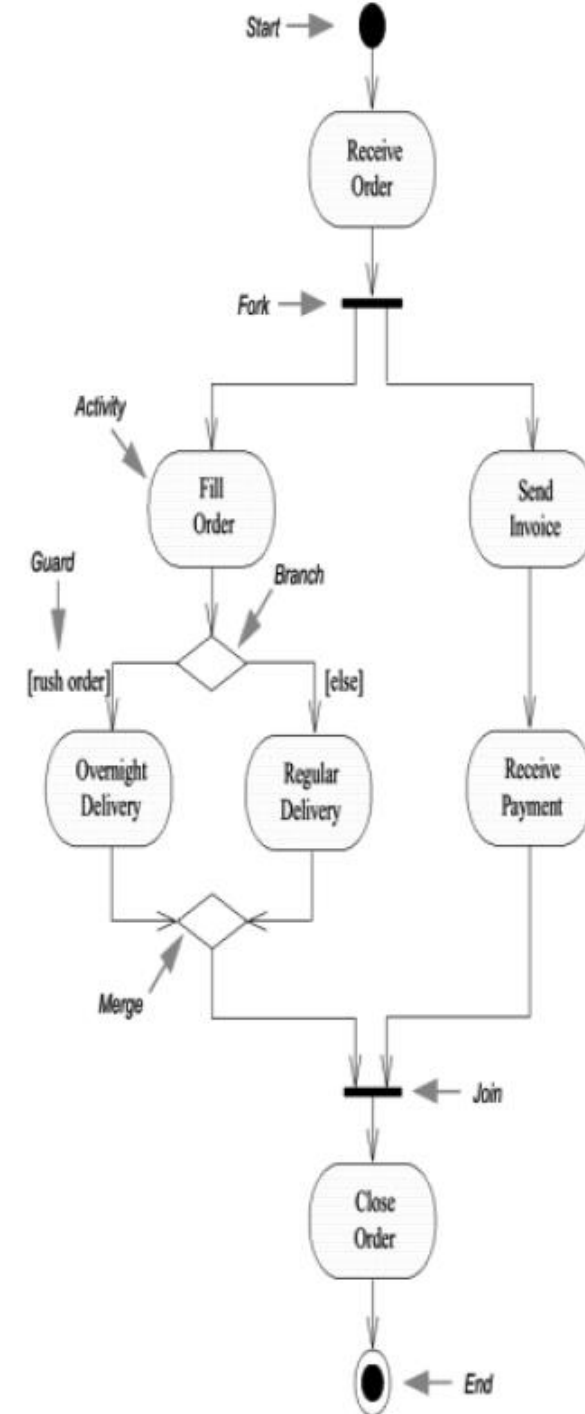
- Dijagrami stanja modeluju ponašanje objekta u više use case-ova
- Kolaboracija se predstavlja dijagramima interakcije (jedan use case)
- Dijagrami aktivnosti modeluju ponašanje više objekata u više različitih use case-ova

Zadatak

- Nacrtati dijagram stanja koji modeluje upravljanje tajnom pregradom u srednjevjekovnom zamku
 - Dragocjenosti se čuvaju u dobro skrivanom sefu u zamku
 - Da bi se otkrila bravi sefa potrebno je ukloniti specijalnu svijeću iz svjećnjaka, ali pri tome vrata moraju da budu zatvorena
 - Kada se otkrije bravi, ona se otključava i otvara se sef
 - Radi dodatne sigurnosti sef se otvara samo ako se u svjećnjak stavi druga svijeća
 - Ukoliko lopov otvori sef a ne stavi drugu svijeću, oslobađa se čudovište koje će ga pojesti

Dijagrami aktivnosti

- Prikazuju logiku procedura, poslovnih postupaka i toka posla
- Moguće je prikazati paralelna ponašanja
- Varijanta dijagrama stanja u kojoj su aktivnosti predstavljene stanjima



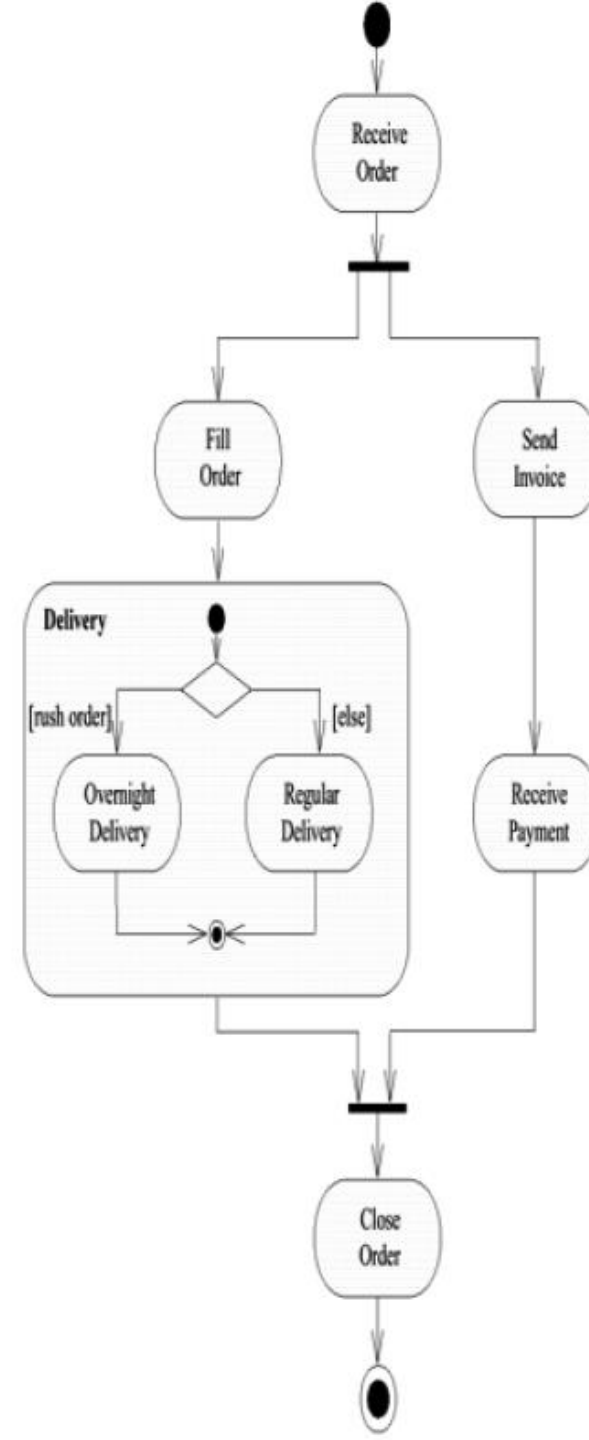
Elementi

- Branch
 - Jedna ulazna tranzicija, više uslovnih (uzajamno isključivi) izlaznih tranzicija
 - [else], kao default iz switch komande
- Merge
 - Jedna izlazna tranzicija, više ulaznih
 - Predstavlja kraj branch-a

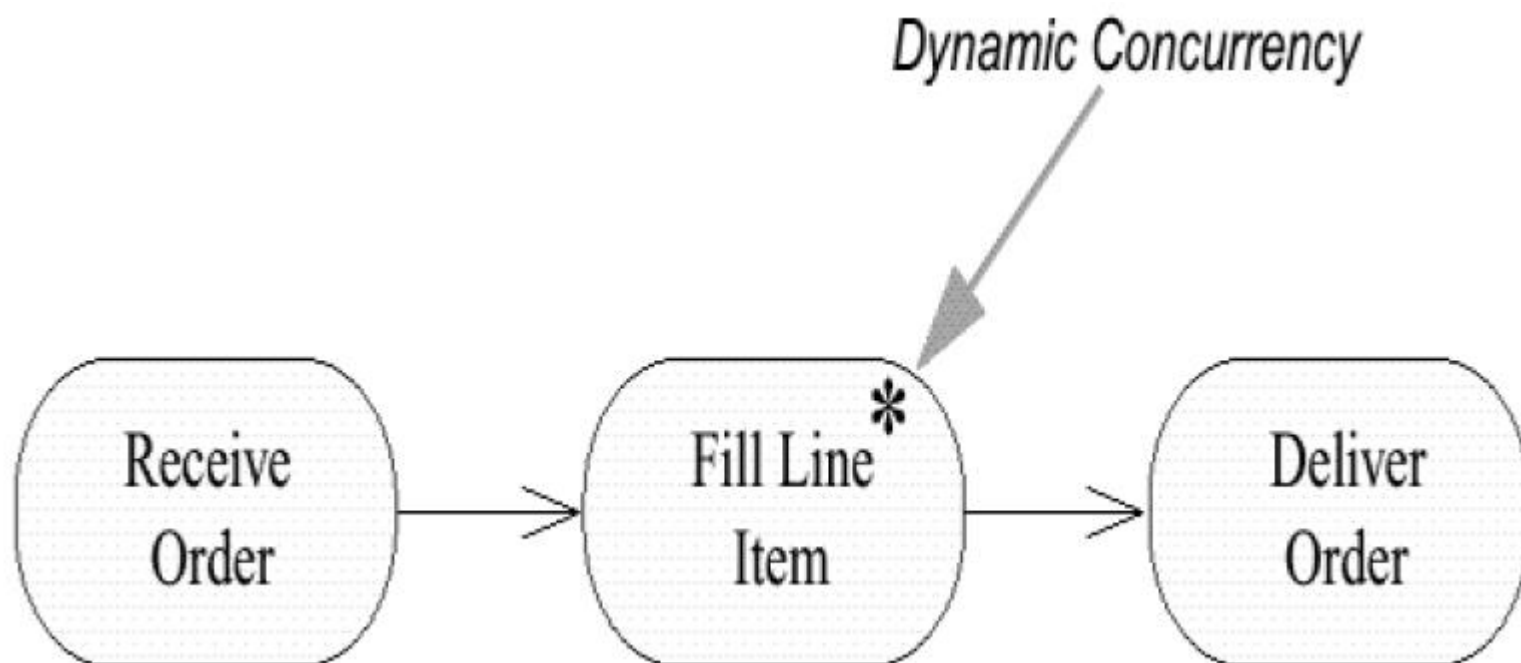
Elementi 2

- Paralelno ponašanje, fork i join
 - Konkurentno programiranje
- Fork
 - Jedna ulazna tranzicija, više izlaznih
 - Dešavaju se paralelno, odnosno ne postoji nije bitan poredak izvršavanja, odnosno moguće je i međusobno preklapanje
- Join
 - Izlazna tranzicija je moguća samo kada sve ulazne tranzicije budu kompletirane

Dekompozicija aktivnosti

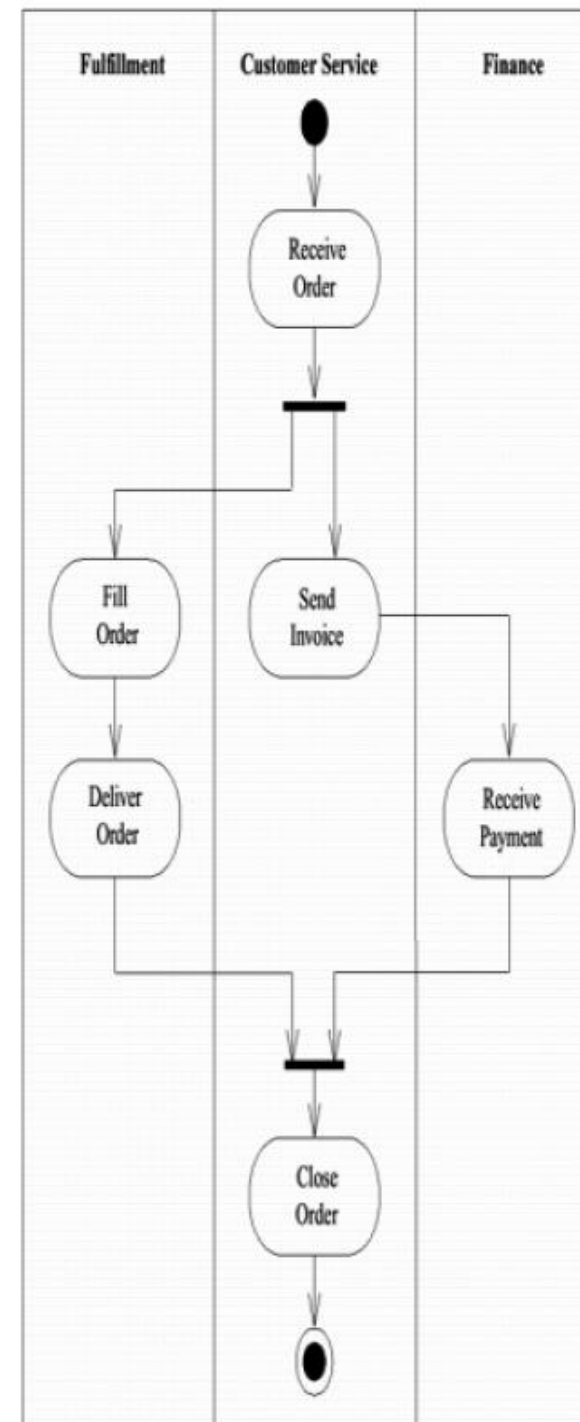


Petlja



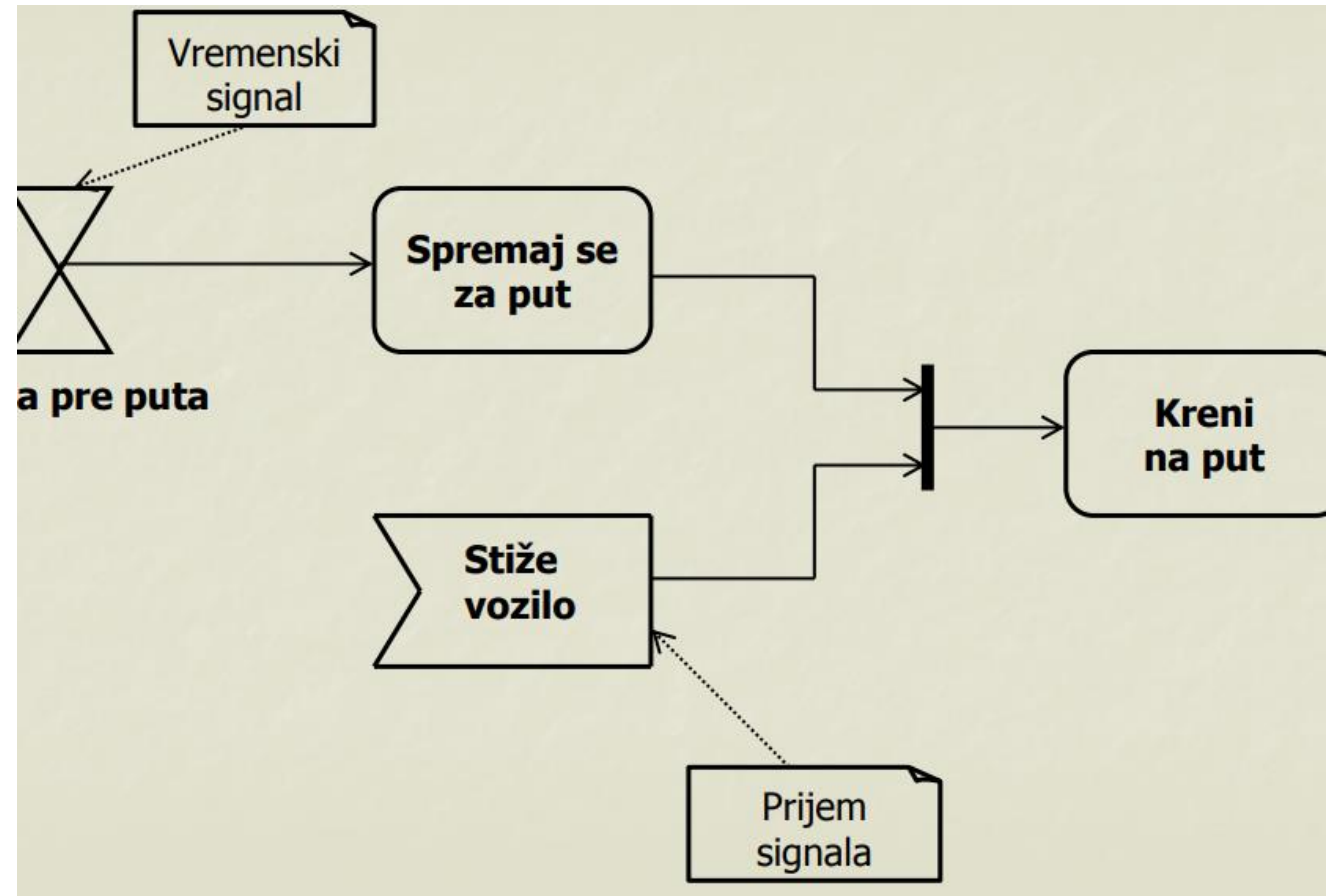
Particije – swimline

- Specificiraju odgovornosti klase



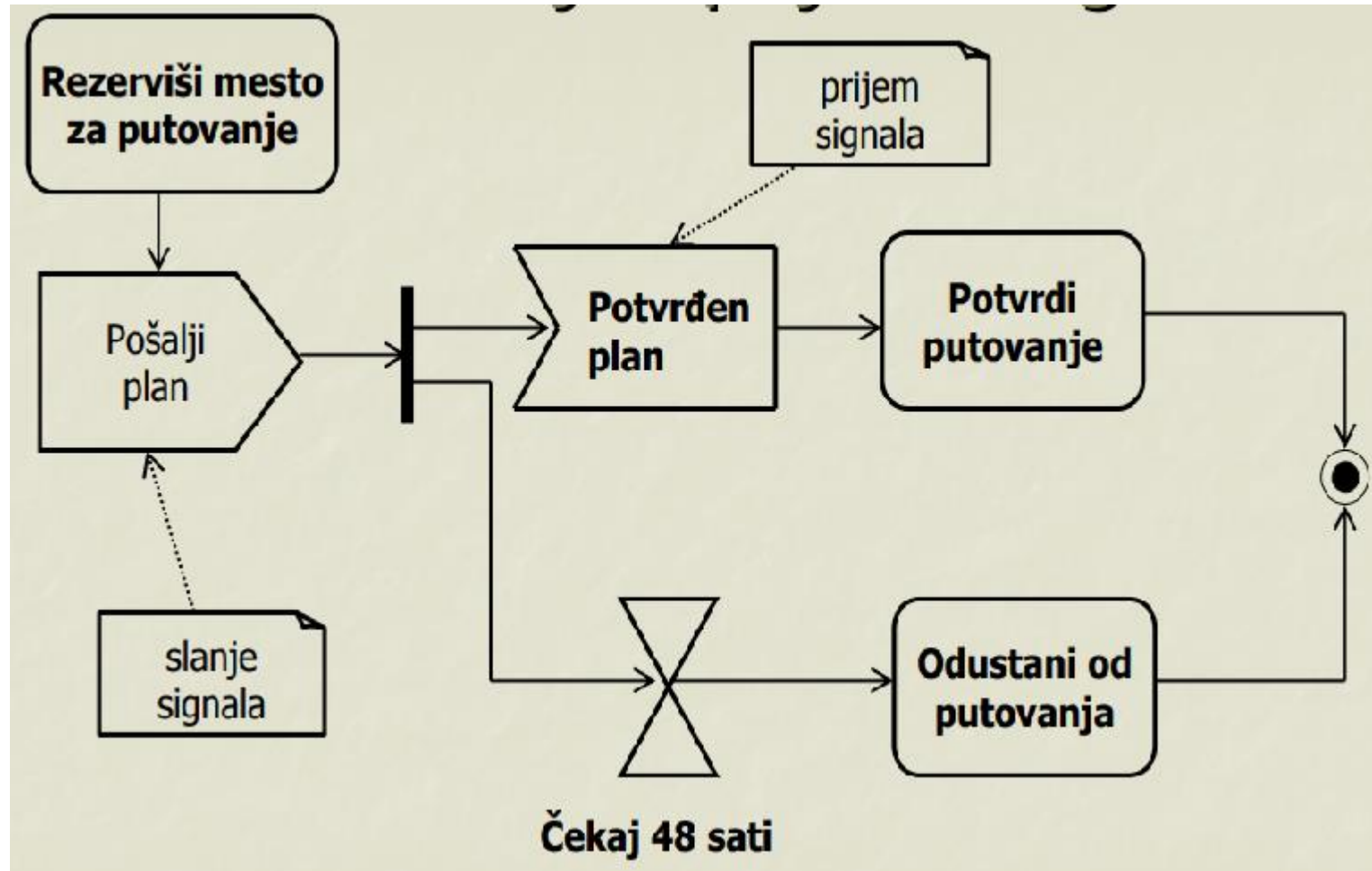
Signali

- Signal ukazuje da aktivnost prima događaj iz spoljnjeg procesa
 - Aktivnost neprekidno osluškuje događaje
 - Dijagram pokazuje kako aktivnost reaguje
- Vremenski signal nastaje zbog protoka vremena

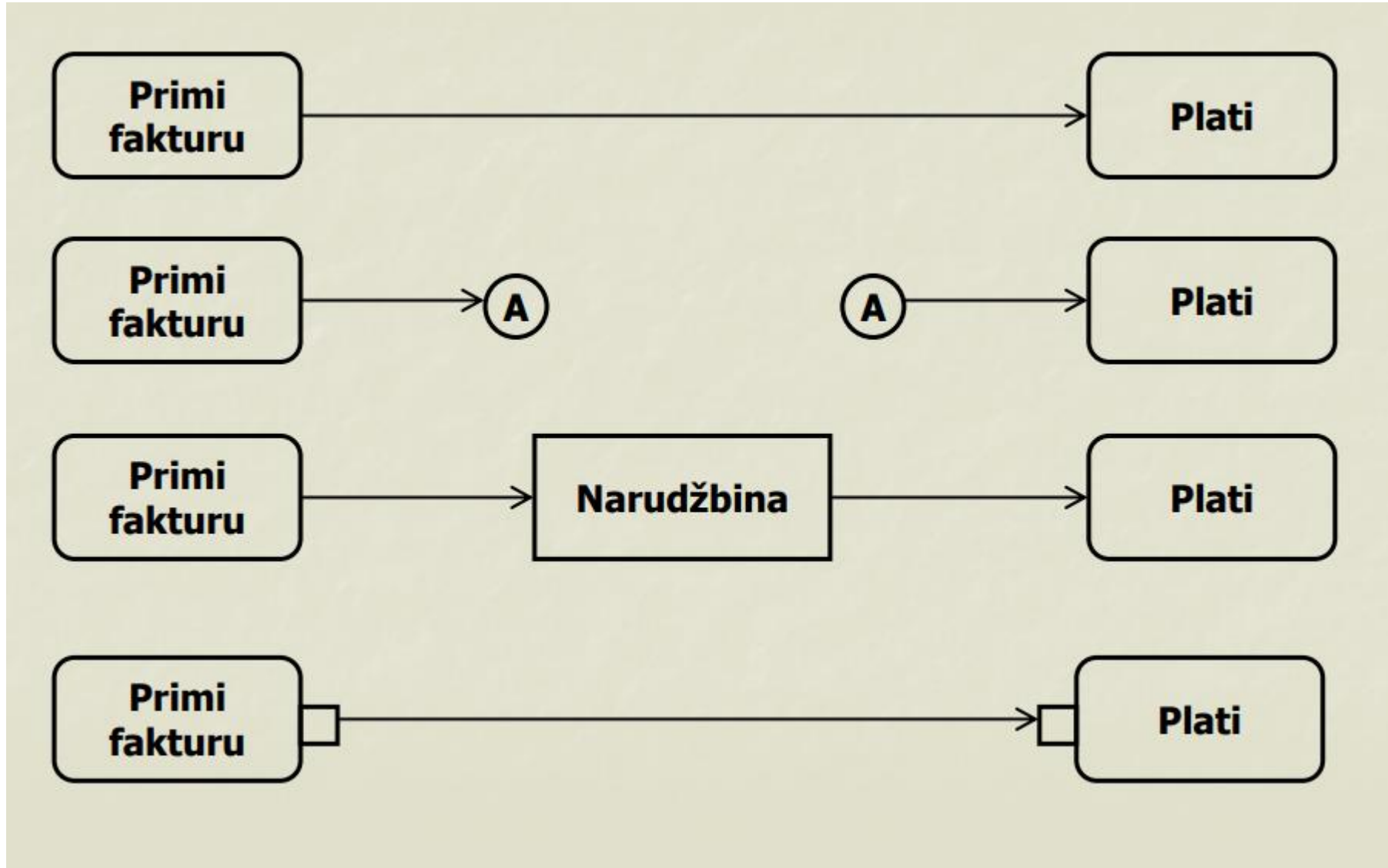


Signali 2

- Signali mogu da se šalju
 - Kada je potrebno poslati poruku i sačekati odgovor prije daljeg rada

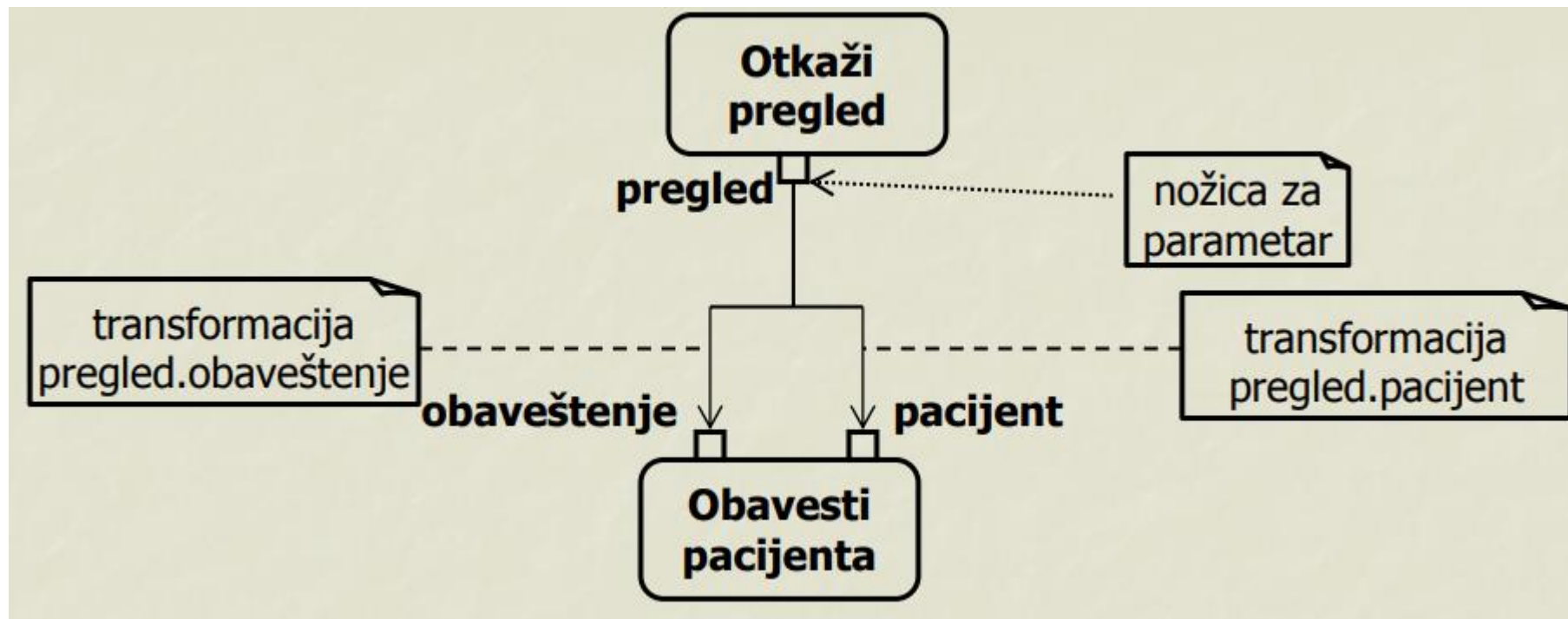


Tokovi i ivice



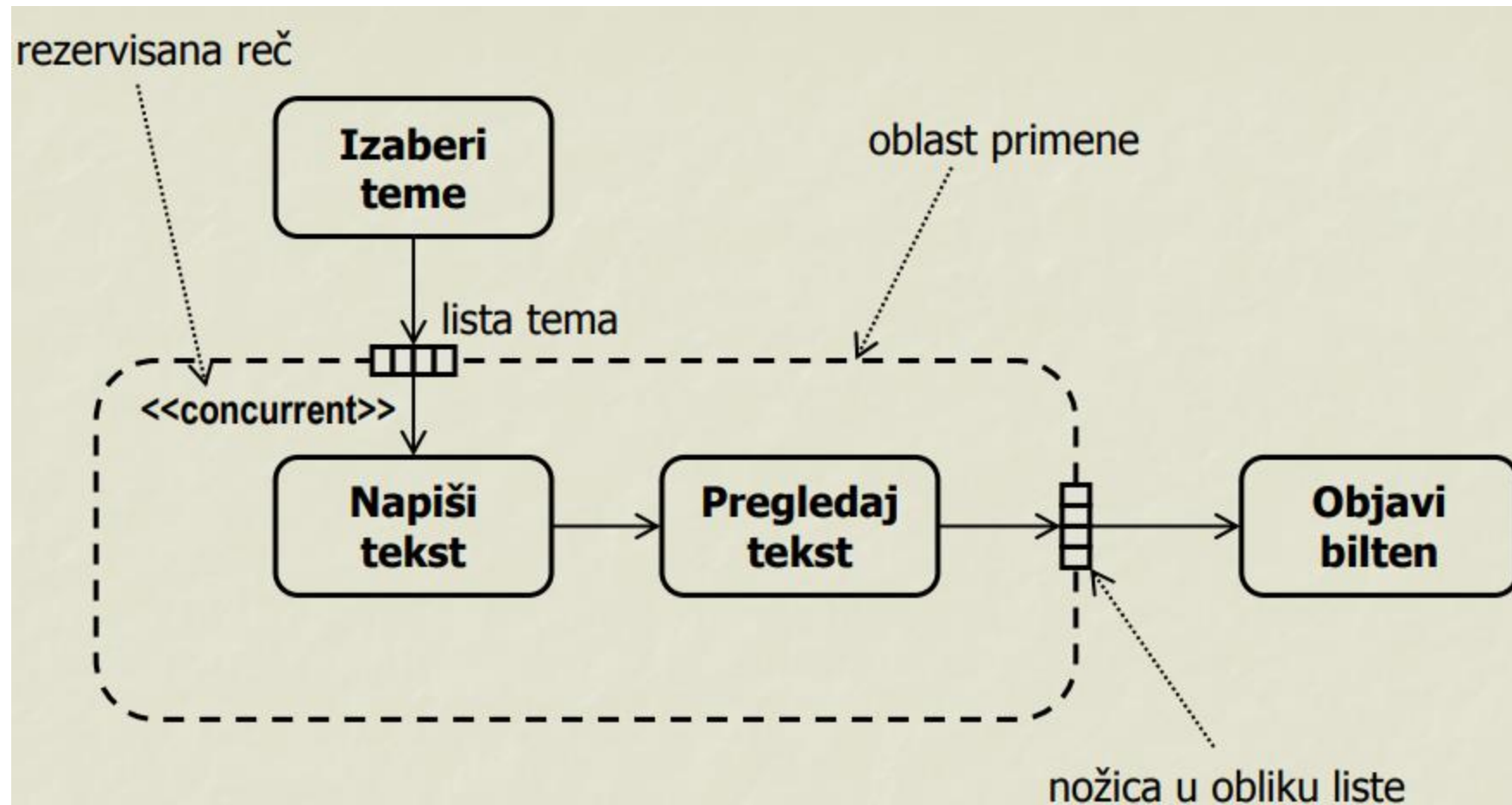
Nožice i transformacije

- Nožice prikazuju parametre akcija
 - Izlazni parametri jedne akcije moraju da odgovaraju ulaznim parametrima sljedeće akcije, inače se mora naznačiti potrebna transformacija

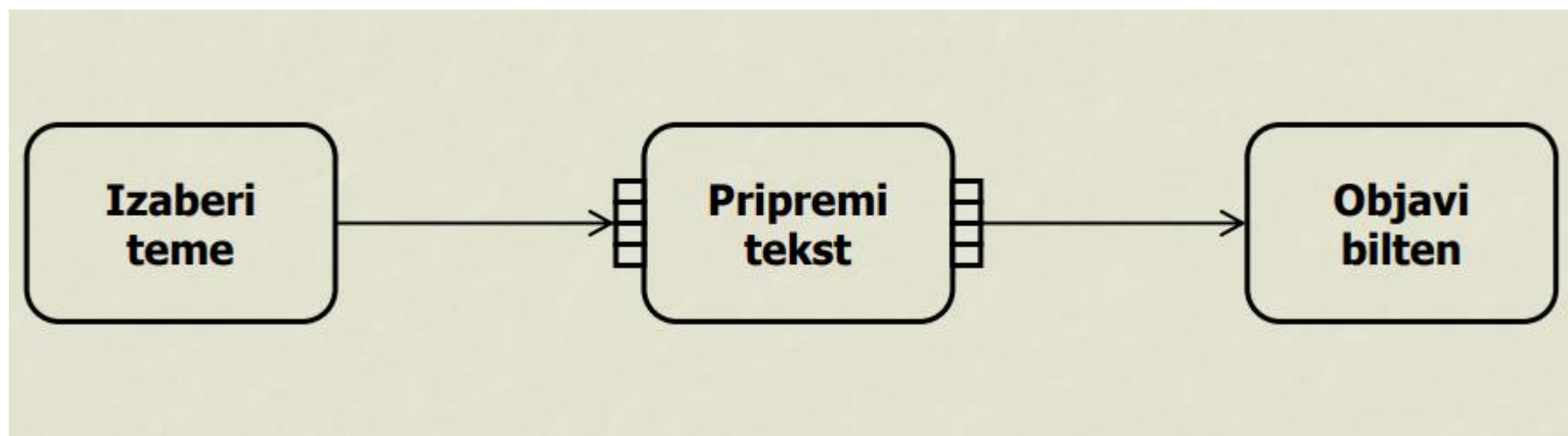


Oblasti primjene – expansion region

- Kada poslije jedne akcije više puta treba izvršiti drugu
- Oblast primjene je dio dijagrama u kome se akcije izvršavaju po jednom za svaki element kolekcije

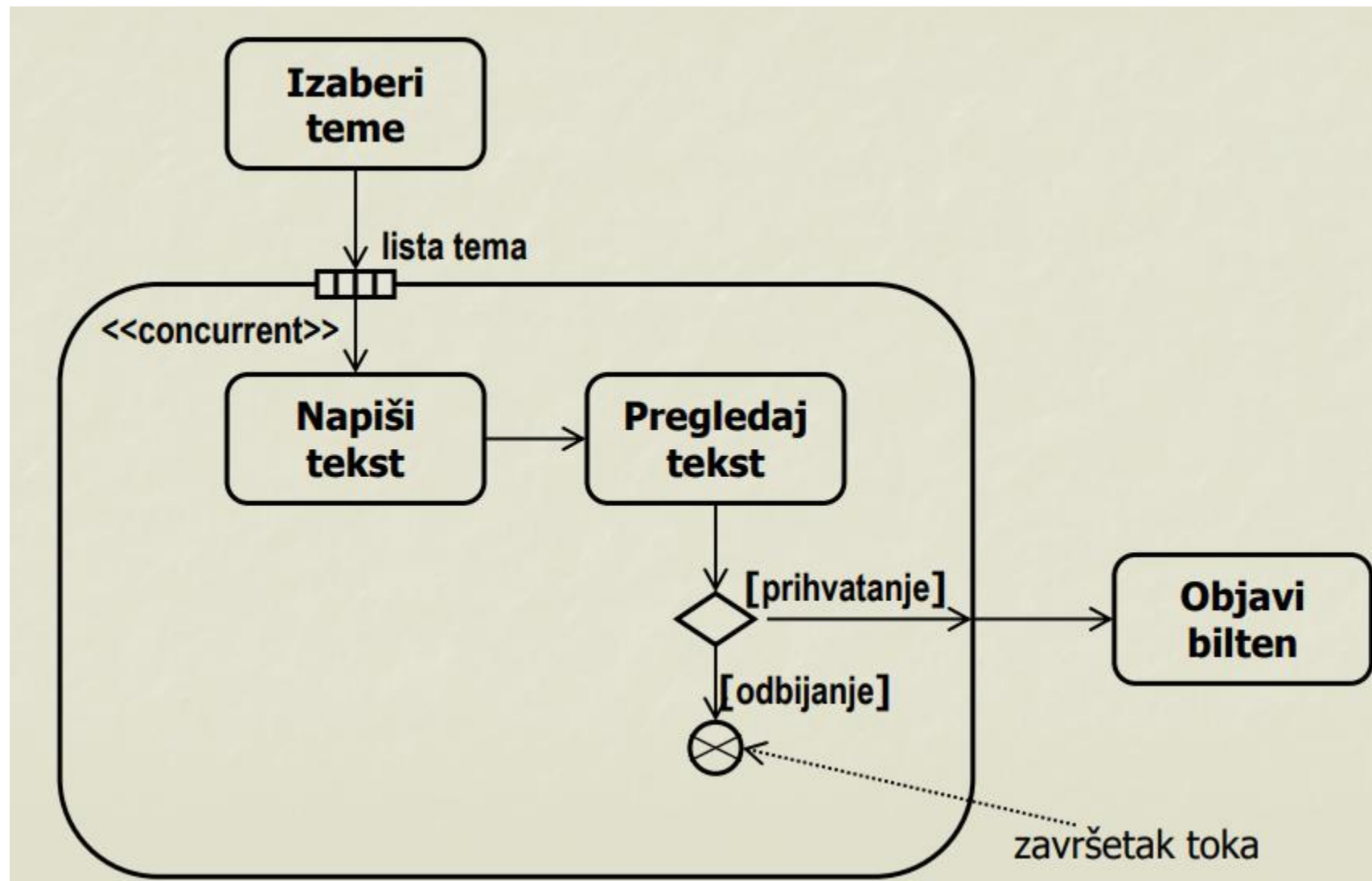


Dinamička konkurentnost – jedna akcija



Završetak toka

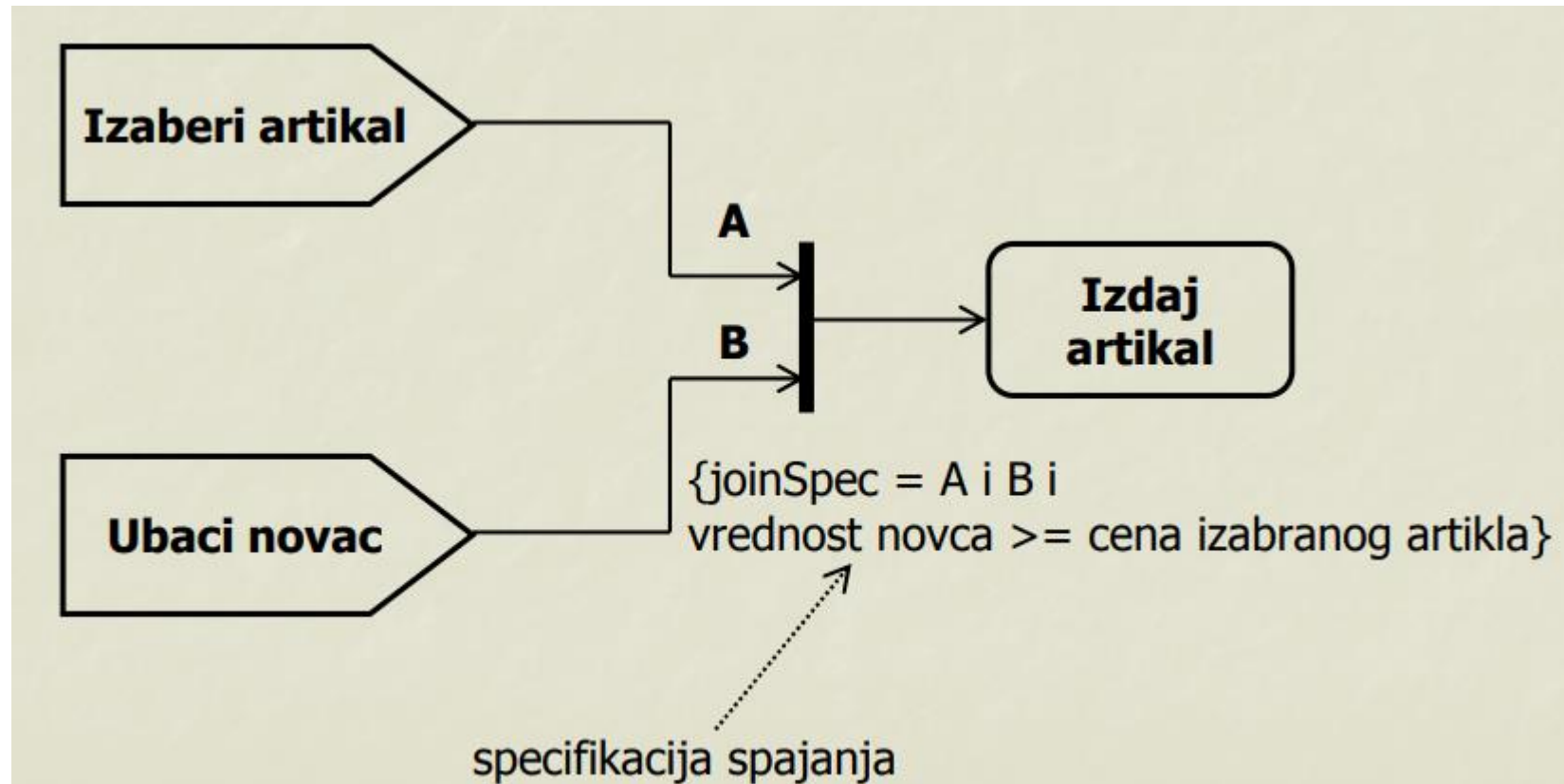
- Završetak toka bez prekidanja čitave Aktivnosti
- Ulazna kolekcija je veća od izlazne



Specifikacija spajanja

- Podrazumijevano je da spajanje dozvoljava izvršavanje izlaznog toka kada svi ulazni tokovi dođu do tačke spajanja

- Specifikacija spajanja je logički izraz koji se pridružuje spajanju



Upotreba

- Analiza use case-ova
- Razumijevanje poslovnih procesa
- Opisivanje algoritama
- Multithreading
- ...