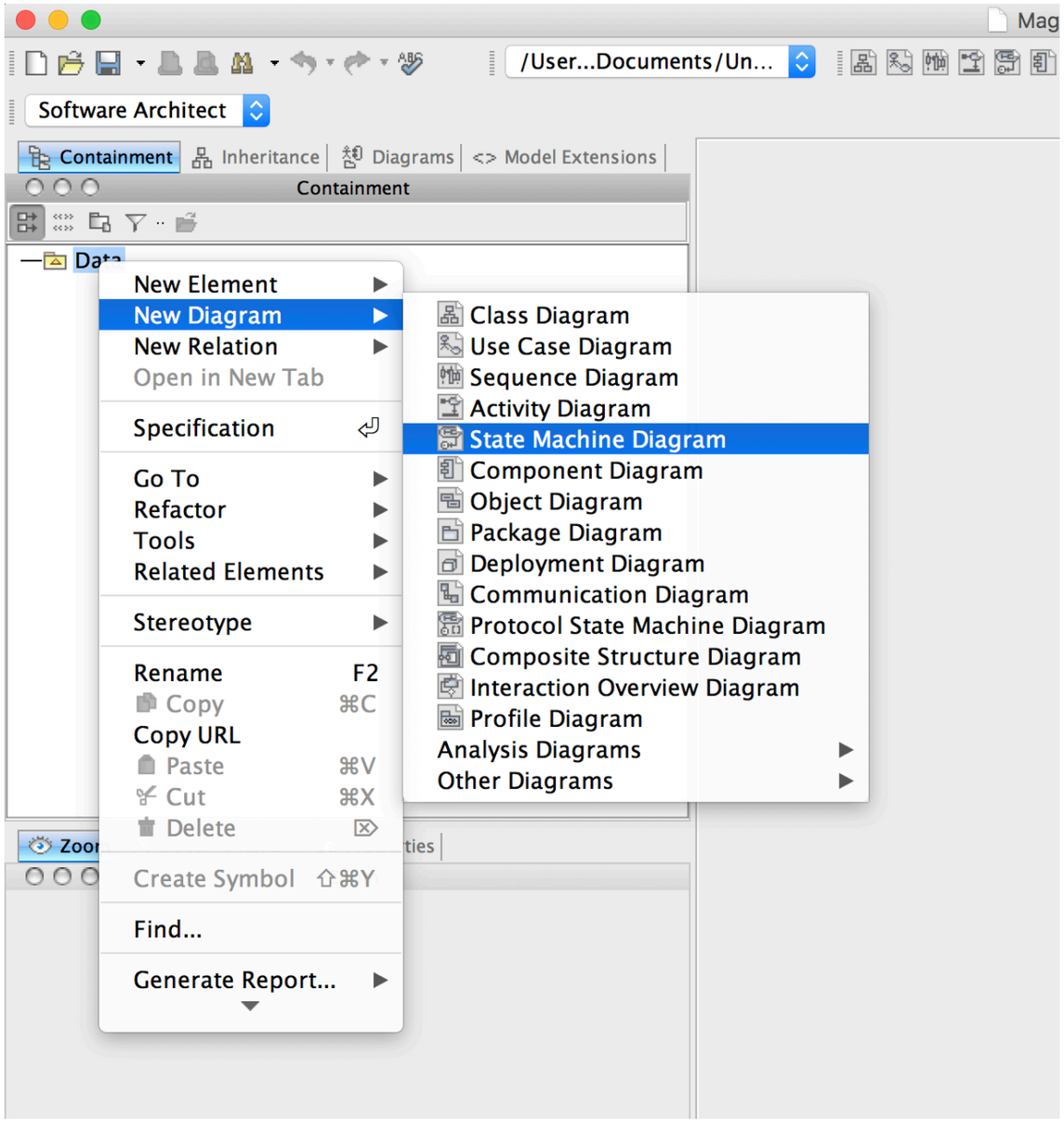
DIJAGRAM PRELAZA STANJA

OSNOVNI KONCEPTI

- Dijagram prelaza stanja (dijagram konačnih automata) koristi se za projektovanje sistema koji se mogu nalaziti u konačnom skupu stanja
- Dijagram stanja definiše **konačan skup stanja** u kojima objekat neke klase može da se nađe, **događaje** koji iniciraju prelaz objekta iz jednog stanja u drugo, kao i **akcije** koje se izvršavaju kao rezultat tog prelaza
- Prelasci iz jednog stanja u drugo su uslovljeni događajima
- Modeluje deo sistema ili objekat čije se ponašanje želi prikazati i koje je u datom trenutku relevantno
- Prikazuje se kao graf čiji su čvorovi stanja ili pseudo-stanja, a tranzicije predstavljaju grane
- Link ka delu dokumentacije Magic Draw-a koji se tiče dijagrama prelaza stanja.

KREIRANJE NOVOG DIJAGRAMA PRELAZA STANJA



OSNOVNI ELEMENTI DIJAGRAMA KLASA

- Dijagrami prelaza stanja kao osnovne elemente obično sadrže:
    1. Stanja (states)
    2. Tranzicije (transitions)
- Dijagram prelaza stanja kao dodatne elemente (pseudo-stanja) mogu da sadrže:
    1. Inicijalno stanje (initial state)
    2. Čvor odluke (decision node)
    3. Razdelnik (fork)
    4. Spoj (join)
    5. Krajnje stanje (final state)

STANJE

- Predstavlja uslove u kojima se neki objekat nalazi u određenom trenutku vremena (objekat u nekom stanju ostaje konačno dugo)
- Objekat se u svakom trenutku nalazi u nekom od stanja sa dijagrama, pri čemu stanja tokom životnog ciklusa bivaju aktivna i neaktivna
  - aktivno je kad objekat prelazi u neko stanje
  - neaktivno kad objekat izađe iz posmatranog stanja
- Stanja se na dijagramima predstavljaju pravougaonicima sa zaobljenim ivicama i nazivom stanja
- Pravougaonik je podeljen u 2 dela:
  - Prvi deo sadrži naziv stanja
  - Drugi deo definiše interne akcije i aktivnosti u zavisnosti od njemu prosleđenog događaja

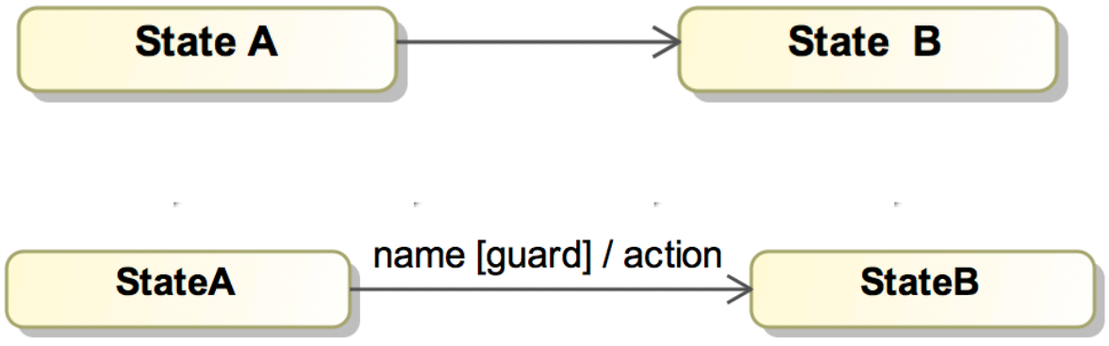
Simbol stanja

TRANZICIJE

- Modeluje reakciju na događaj, koji može izazvati prelazak iz jednog stanja u drugo ili povratak u isto stanje
- Prilikom izvođenja tranzicije, može doći do izvršenja njoj pridruženih akcija
- Specifikacija tranzicije:

  događaj [uslov] / akcija

  - Događaj – naziv događaja koji okida tranziciju
  - Uslov – skup uslova koji moraju biti zadovoljeni kako bi se tranzicija izvršila
  - Akcija – posledica okidanja transakcije



Simbol za prelaz iz Stanja A u Stanje B

SPECIFIKACIJA TRANZICIJE

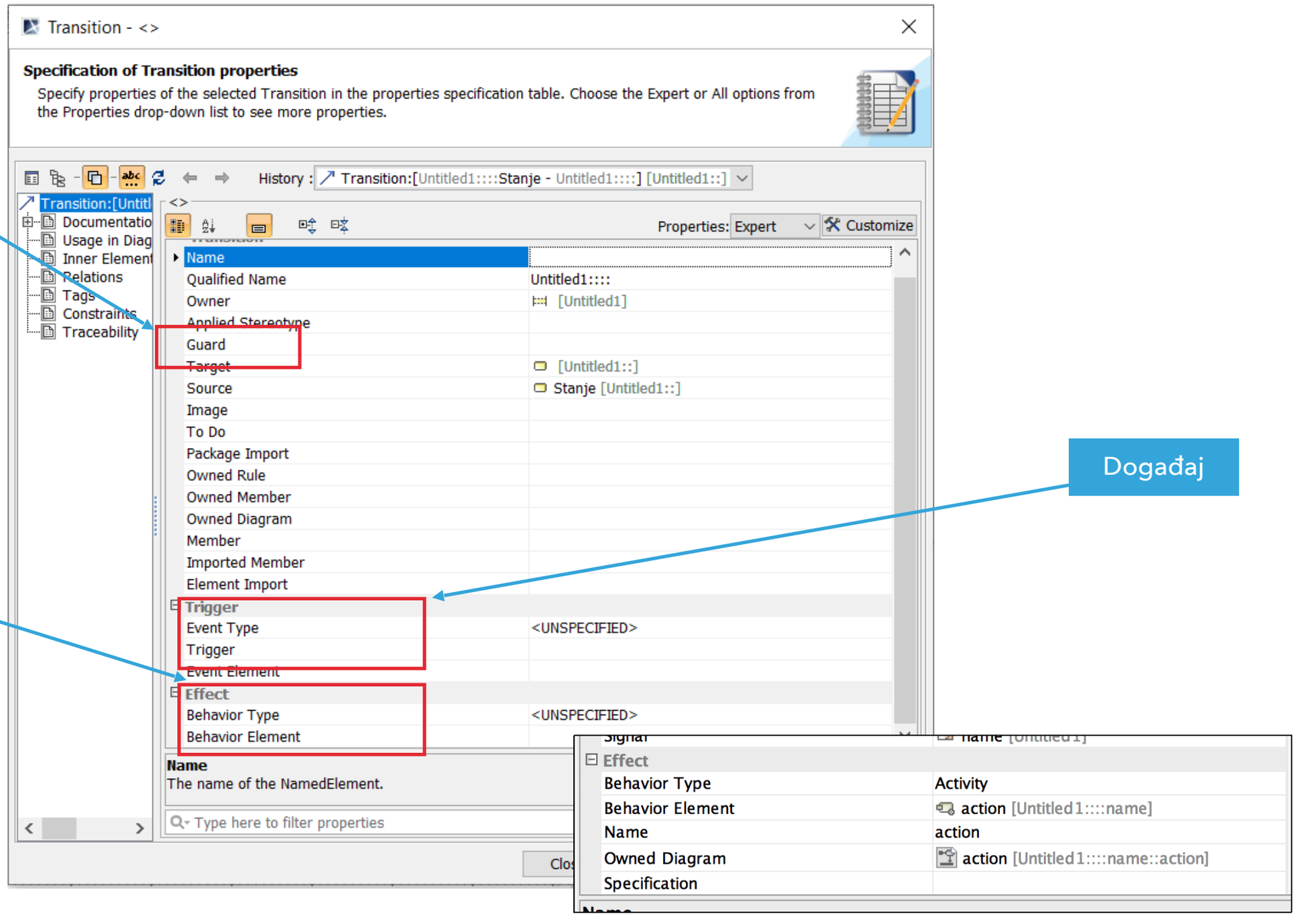
- Događaj nije obavezan – tranzicija bez specifikacije (predstavljena samo strelicom na dijagramu) se izvršava bezuslovno kada se izvrše sve akcije u stanju u kojem se objekat trenutno nalazi
- Uslov je obavezno navesti ukoliko postoji više tranzicija koje izaziva isti događaj i koje polaze iz istog stanja
- U situaciji u kojoj uslov nije zadat, tranzicija se dešava kada se završe sve aktivnosti objekta u stanju u kojem se trenutno nalazi
- Akcije pridružene tranziciji se pridružuju iza kose crte u specifikaciji tranzicije i označavaju šta sistem treba da preduzme prilikom izvršavanja tranzicije

SPECIFIKACIJA TRANZICIJE

Uslov

Akcija

Događaj

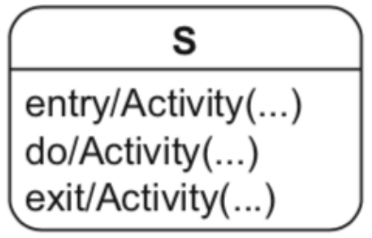


VRSTE TRANZICIJA

- Tranzicije mogu biti:
  - Spoljne tranzicije (external transitions) – dovode do promene stanja objekta (jedno stanje se napusti, dok drugo postane aktivno)
  - Unutrašnje tranzicije (internal transitions) – tranzicije koje se dešavaju unutar jednog stanja, predstavljaju reakcije na obradu događaja dok se objekat nalazi u jednom konkretnom stanju

AKCIJE

- Nakon što se specificiraju stanja koja objekat može da ima i događaji koji izazivaju tranzicije, potrebno je analizirati akcije koje sistem treba da preduzme kao reakcije na događaje
- Akcije se specificiraju u okviru tranzicija i u okviru stanja
- Akcije koje se dese prilikom okidanja tranzicije se nalaze u specifikaciji tranzicije
- Načini na koje stanje izvršava akciju:
    1. U trenutku ulaska u stanje (*entry*)
    2. U trenutku izlaska iz stanja (*exit*)
    3. Tokom boravka u stanju (*do*)
    4. Aktivnosti pri internim tranzicijama

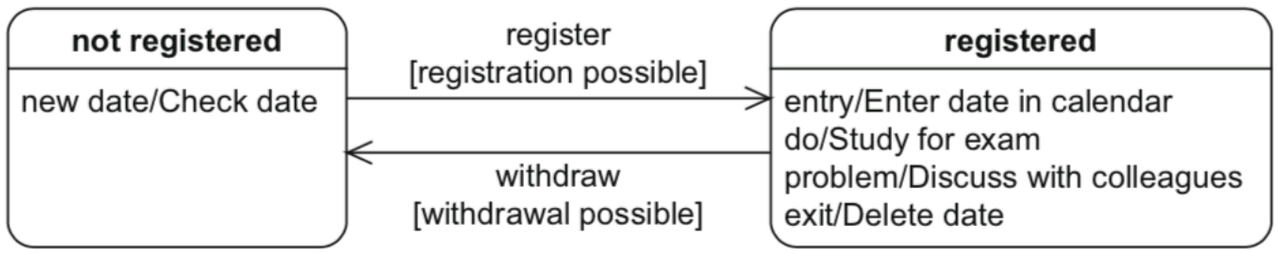


Primer stanja sa pridodatim akcijama

ENTRY, EXIT, DO

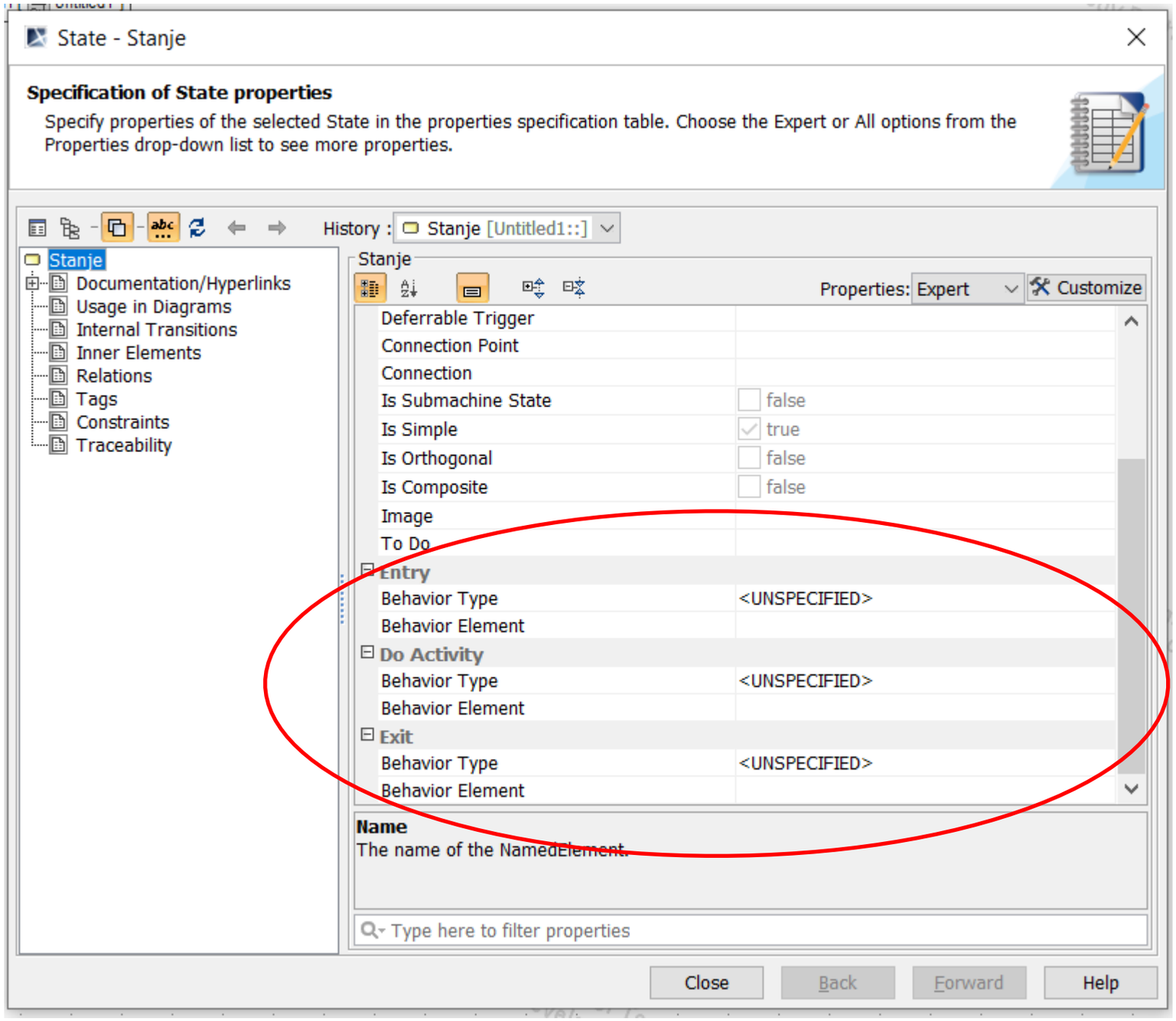
- Akcija pridružena *entry* događaju se izvršava prilikom svake tranzicije koja omogućava ulazak u stanje
- Akcija pridružena *exit* događaju se izvršava prilikom svake tranzicije koja dovodi do izlaska iz trenutnog stanja.
- Važi i za tranzicije koje izlaze i vraćaju se u isto stanje (self-transactions).
- Akcija pridružena *do* događaju se izvršava tokom boravka u stanju, a aktivira se posle izvršenja entry akcije tog stanja
- NAPOMENA: akcije koje se navode u okviru tranzicija, kao i entry i exit akcije stanja, treba da traju kratko i ne smeju se prekidati. Do akcija može da traje dugo i dozvoljeno ju je prekidati

ENTRY, EXIT, DO



- Slika prikazuje dva stanja u kojima se može nalaziti student koji vrši prijavu za ispit - a to su da je student već prijavljen, odnosno da nije prijavljen
- Sve dok student nije prijavio ispit, a pojavi se novi datum za polaganje ispita, poziva se metoda Check date
- Ukoliko se desi događaj *register* i uslov *registrationPossible == true* student prelazi u stanje *registered* i datum ispita biva dodat u njegov kalendar
- Sve dok je stanje studenta *registered* on uči za ispit (*Study for exam)*
- Ukoliko se pojavi *problem* student ga rešava u razgovoru sa kolegama
- Prilikom događaj *withdraw,* moguća su dva scenarija:
  - Ako je uslov *withdrawalPossible == true* , prekida se izvršavanje *Study for exam* i prelazi se u stanje *not registered i* datum ispita se briše iz kalendara studenta
  - U suprotnom, student ostaje u stanju *registered* i nastavlja sa učenjem

SPECIFICIRANJE ENTRY, EXIT I DO AKCIJA

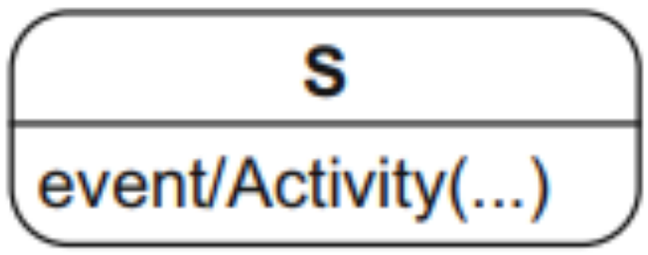


ANALIZA AKCIJA

- Ako sve tranzicije koje ulaze u neko stanje izvršavaju istu akciju, možemo tu akciju vezati za *entry* događaj tog stanja
- U slučaju da samo jedna ulazna tranzicija ne treba da vrši neku akciju, ne bi trebalo koristiti *entry* događaj, nego ostaviti akciju u specifikaciji tranzicija
- Ako sve tranzicije koje izlaze iz nekog stanja obavljaju istu akciju, treba je vezati za *exit* događaj datog stanja. Ako to nije slučaj, akcije treba da ostanu u tranzicijama
- Tranzicija ne mora da ima događaj koji je aktivira jedino ako izlazi iz stanja koje ima *do* akciju, što znači da se stanje napušta kada se *do* akcija završi, odnosno po obavljenom poslu u trenutnom stanju

INTERNE TRANZICIJE

- Interne tranzicije modeluju reakciju na događaj prilikom čega se ostaje u trenutnom stanju
- Ne dolazi do aktiviranja *entry*, *exit* i *do* akcija datog stanja
- Mogu imati iste elemente kao i obične tranzicije: događaj, uslov i akciju

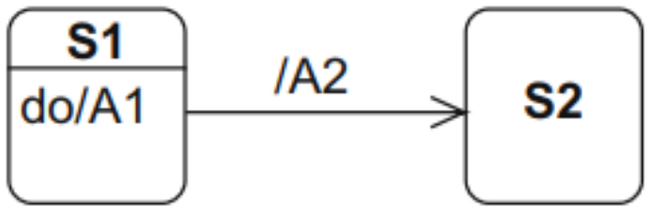


Primer stanja sa pridodatom internom tranzicijom

PRIMERI

S1 nema uslov ili event, što znači da se tranzicija izvršava cim se aktivnost A1 izvrši

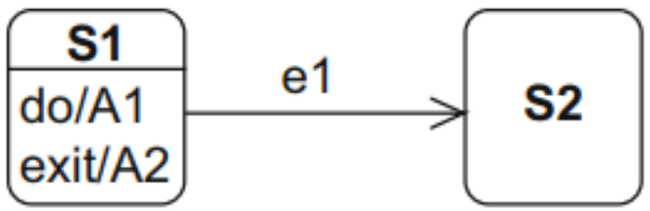
(a)



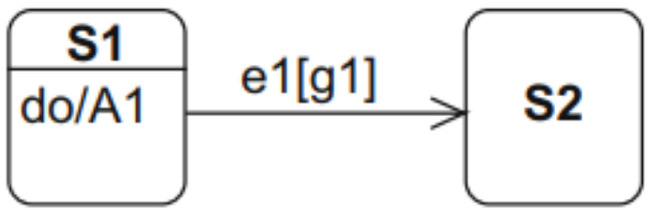
(b)

A2 se izvršava prilikom tranzicije

Tranzicija se izvršava prilikom pojave e1 događaja. Tom prilikom izvršavanje A1 aktivnosti biva prekinuto i prelazi se na stanje S2. Prilikom izlaska iz stanja S1, aktivnost A2 se izvršava



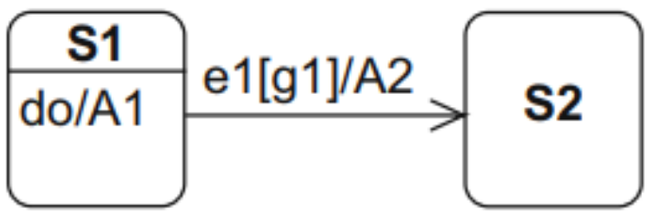
(c)



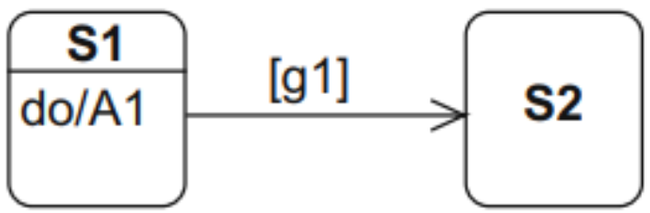
(d)

Uslov g1 se proverava čim se desi događaj e1. Ukoliko je uslov zadovoljen, prelazi se na stanjee S2, a izvršavanje A1 se prekida. U suprotnom se ne prekida izvršavanje A1

Slično kao d, s tim da se izvršava i aktivnost A2 prilikom tranzicije



(e)



(f)

Sistem ostaje u stanju S1 sve dok se A1 ne izvrši i tek tad se proverava uslov g1. Ukoliko jee g1 tačno, prelazi se u stanje S2. U suprotnom, ostaje se u stanju S1 iz koga ne može da se izađe.

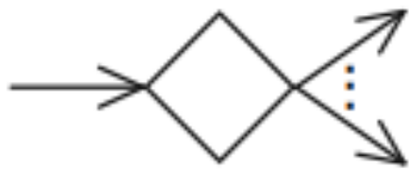
INICIJALNO STANJE

- Početak dijagrama prelaza stanja
- Nema ulaznih grana, uglavnom jedna izlazna koja vodi do prvog stanja u kojem se objekat nalazi
- Događaji se ne specificiraju na izlaznim granama, dok akcije mogu
- Postoji samo jedan, kako bi se objekat nalazio u konzistentnom stanju kada se kreira

Simbol inicijalnog stanja

ČVOR ODLUKE

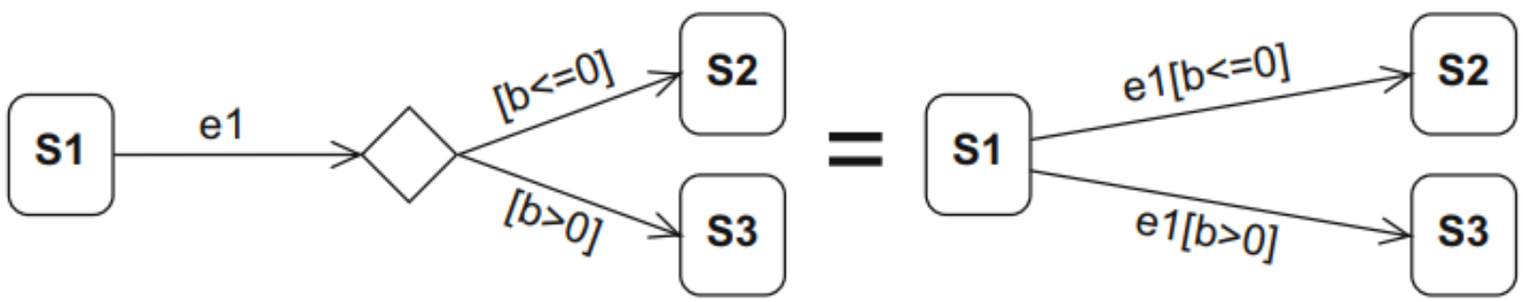
- Modeluje alternativne tranzicije
- Poseduje jednu ulaznu tranziciju sa događajem koji je okida i bar dve izlazne tranzicije
- Izlazne tranzicije moraju biti međusobno isključive po uslovima datim u njihovoj specifikaciji
- Konceptualno ne zauzima nikakvo vreme, samo se evaluiraju uslovi i objekat prelazi u naredno stanje određeno odabranom tranzicijom.



Simbol čvora odluke sa ulaznom i izlaznim tranzicijama

PRIMERI SA ČVOROM ODLUKE

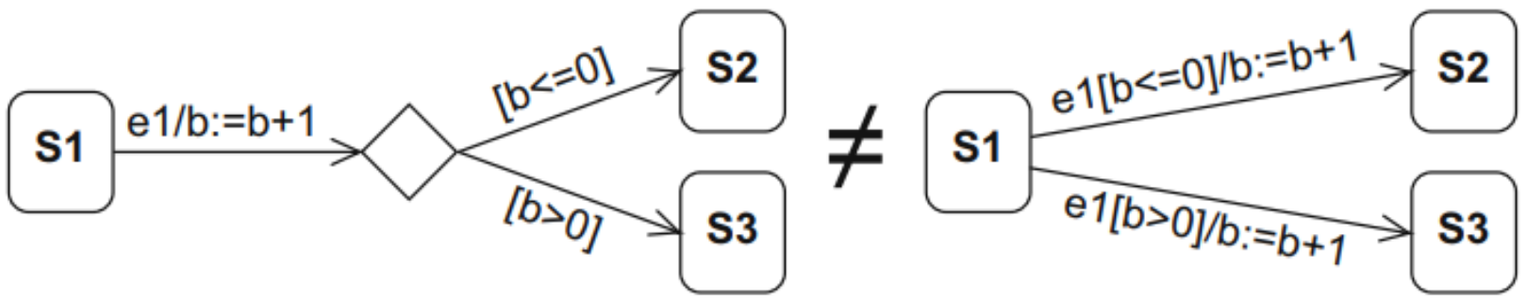
Ukoliko se desi e1, izvršava se tranzicija, nakon čega se proveravaju uslovi i sistem prelazi u jedno od mogućih stanja



(a) (b)

Isto kao i b

Ukoliko se desi e1, izvršava se tranzicija i b biva uvećano za 1, nakon čega se proveravaju uslovi i sistem prelazi u jednom od mogućih stanja

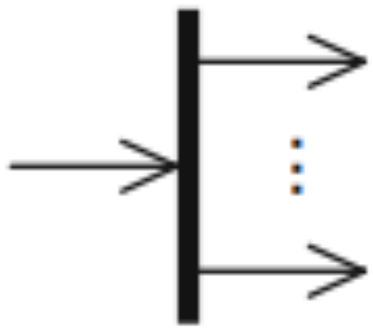


(c) (d)

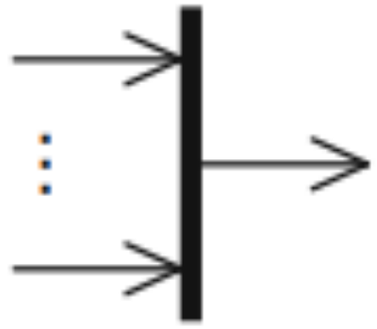
U ovom slučaju b se uvećava za 1, nakon što su provereni uslovi.

RAZDELNIK I SPOJ

- Razdelnik – modeluje scenario konkurentnih tranzicija koje iz njega izlaze
- Spoj – modeluje spajanje više konkurentnih tranzicija



Simbol razdelnika



Simbol spoja

KRAJNJE STANJE

- Reprezentuje krajnje konzistentno stanje u kojem objekat može ostati trajno, tj. ne briše se

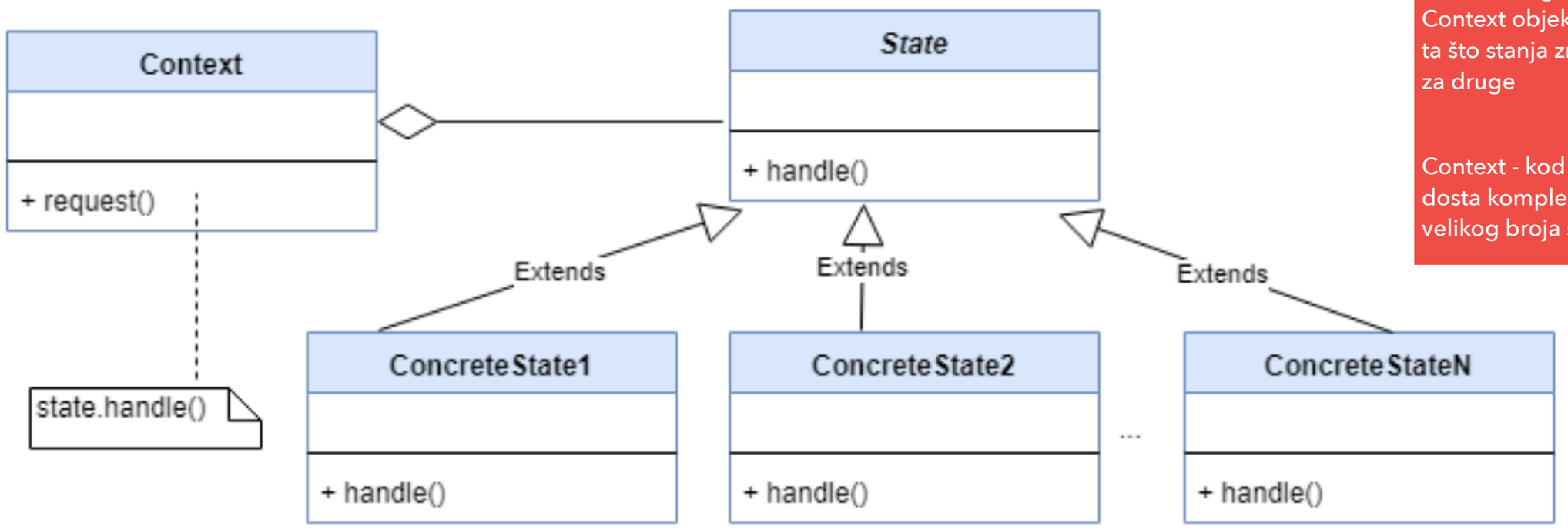
Simbol finalnog stanja

STATE PATTERN

Klasa koja ima referencu na trenutno aktivno stanje i koja predstavlja kontekst za izvršavanje konačnog automata

Apstraktna klasa ili interfejs. Ima referencu ka Context klasi.

Kreiranje i uništavanje State objekata: jednom pri inicijalizaciji ili pri svakoj promeni stanja.



Ko menja stanje?

ConcreteStateX - mora mu se omogućiti pristup Context objektu. Mana je ta što stanja znaju jedna za druge

Context - kod može biti dosta kompleksan kod velikog broja stanja

ZADATAK 1

- Na slici prikazan je automat mašine za žvake ili "gumball machine". Krugovi predstavljaju stanja, a strelice predstavljaju prelaze stanja.
- Na osnovu slike modelovati state dijagram. Isti nakon toga i implementirati u Javi. Kreirati main metodu i testirati izmene stanja.