

Modelovanje poslovnih procesa

- slajdovi sa predavanja za II kolokvijum 2012/13 (by Stepke) -

1. UML2.0 - Dijagram Aktivnosti

UML

- UML (Unified Modeling Language)
- OMG (Object Management Group) - organizacija zadužena za brigu o standardizaciji UML
- UML - de facto standardni jezik za modelovanje
- Sadrži skup dijagrama za opis strukture i dinamike svih vrsta sistema
- Najviše se koristi u razvoju informacionih sistema
- Aktuelna verzija - UML 2.0

UML 2.0 dijagrami

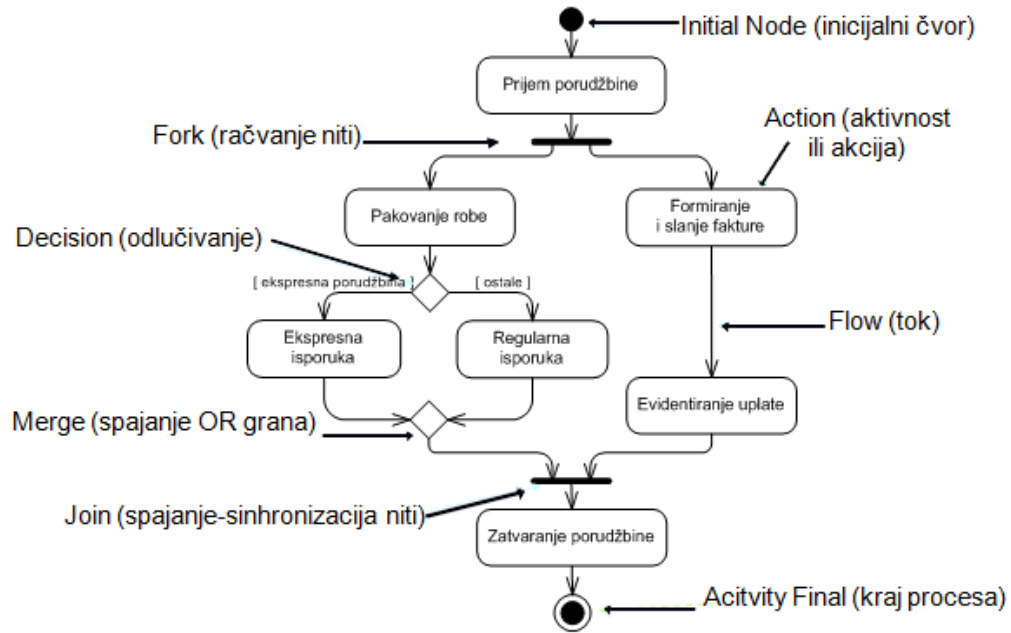
- UML 2.0 definiše 13 vrsta dijagrama

UML 2.0 Dijagrami aktivnosti

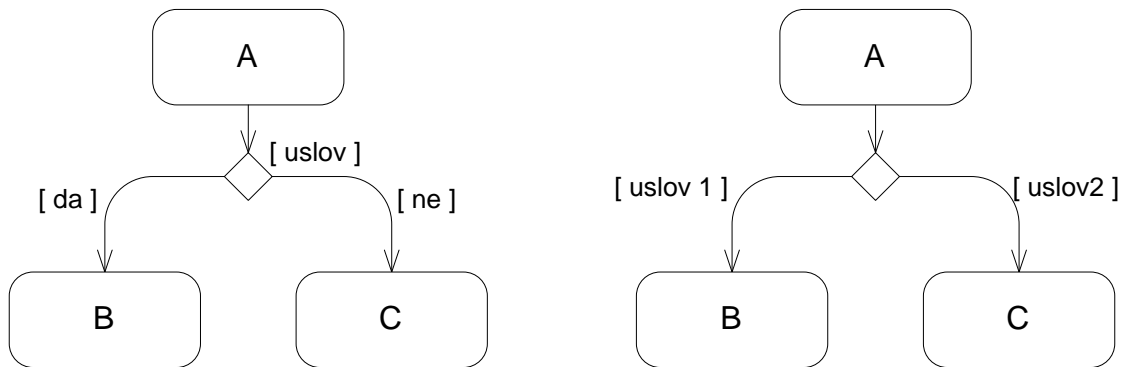
Dijagrami aktivnosti se koristi na najrazličitijim nivoima projektovanja softvera i to za opis:

- Poslovnih procesa
 - Workflow-a
 - Složenih use-case tokova
 - Proceduralne logike
 - Algoritama
-
- U prethodnim verzijama, Dijagrami aktivnosti (DA) su bili zasnovani na konceptima dijagrama prelaza stanja (teorija konačnih automata) a od verzije UML 2.0 zasnovani su na teoriji Petrijevih mreža.
 - Token - koncept preko kojeg se prati izvršenje dijagrama aktivnosti
 - Dijagrami aktivnosti predstavljaju tehniku za opis proceduralne logike i poslovnih procesa. Sa tog aspekta su slični staroj tehnici flowchart ali glavna razlika je što DA podržava modelovanje paralelnog izvršavanja aktivnosti.

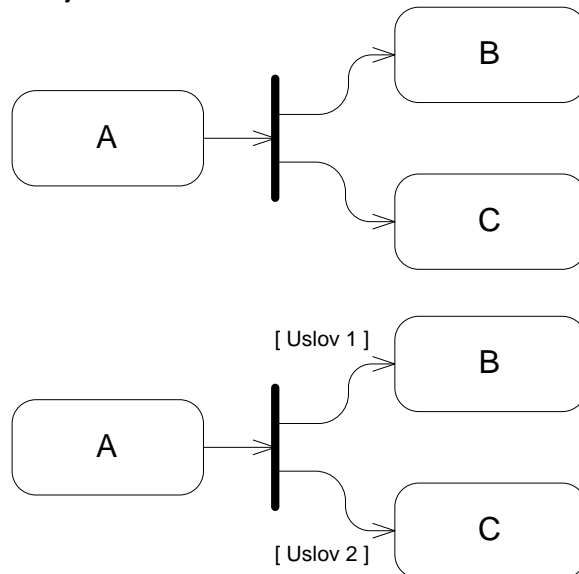
Osnovni grafički simboli DA



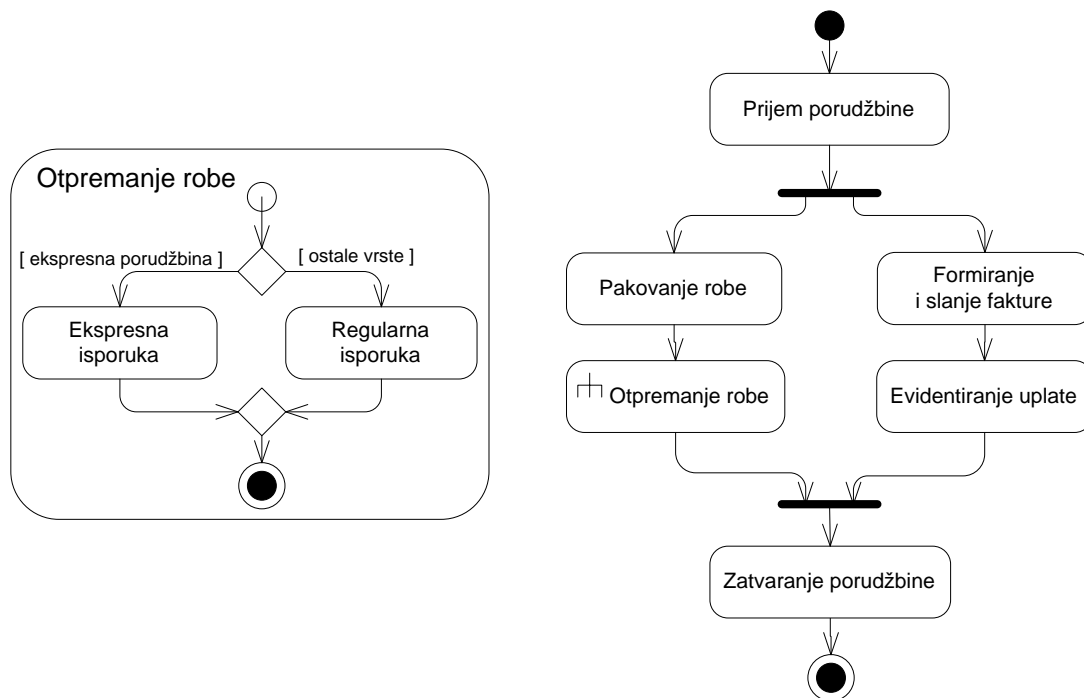
Odlučivanje - OR grananje (Decision)



Račvanje - AND grananje (Fork)

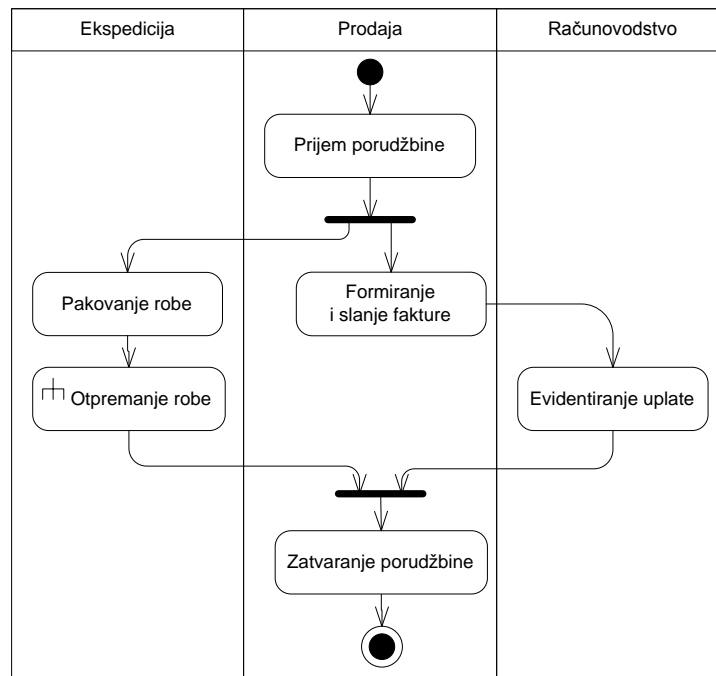


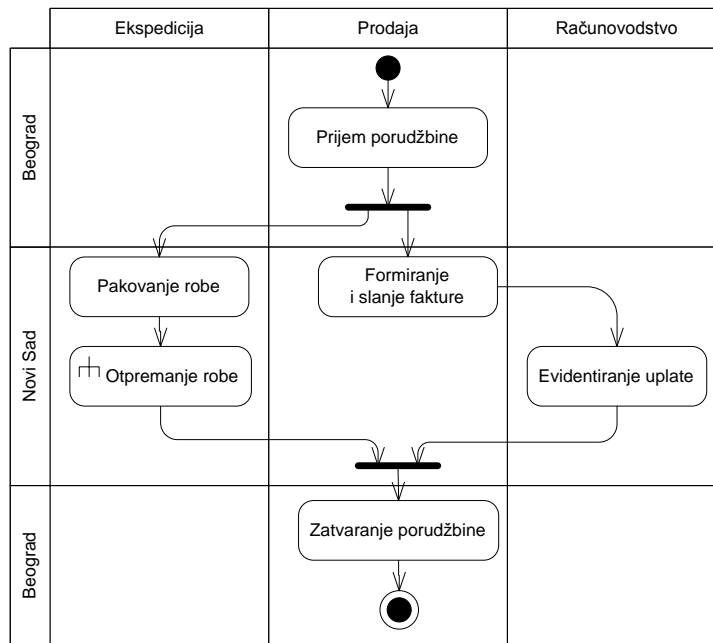
Podprocesi (Subactivities)



Particije (Partitions)

- **Particije** predstavljaju uloge (izvršioce) u modelu odvijanja poslovnog procesa
 - **Uloga** (izvršilac) može predstavljati poslovnog partnera, organizacioni deo, radno mesto ...
 - Uključivanje organizacionog aspekta
 - Od verzije UML 2.0 moguće je konstruisati dvodimenzionalni grid (particije po kolonama i redovima)
- Moguće je particije deliti na podparticije i tako prikazati hijerarhiju particija



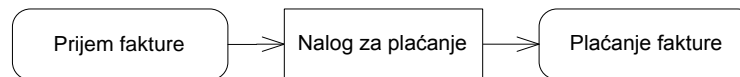


Tokovi (Flows)

- **Kontrolni tok (Control flow)**

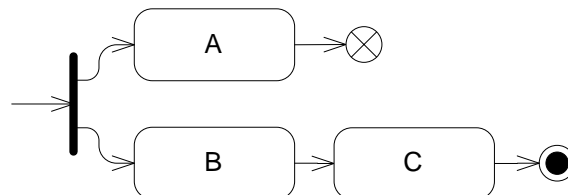


- **Tok objekta (Object flow)**



Kraj toka (Flow final)

- **Kraj toka (Flow final)** - služi za označavanje kraja jednog toka ili jedne niti, dok neka druga grana ili nit imaju završetak celog procesa.



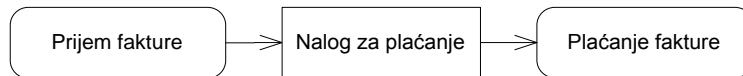
Konektori (Connectors)

- **Konektori (Connectors)** služe kao pomoćno sredstvo da se ogroman dijagram aktivnosti podeli na više delova. Tačka prekida prvog dela se označi sa imenovanim konektorom a onda se isti simbol koristi kao početna tačka narednog dela dijagrama.

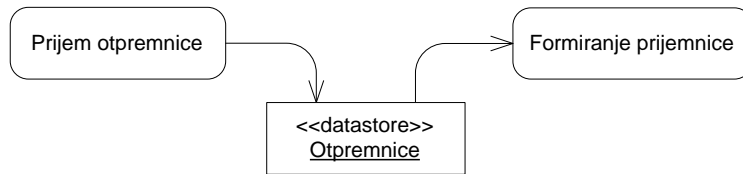


Objekat (Object)

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



- Skladište (Datastore) - skladište podataka (dokumenata)

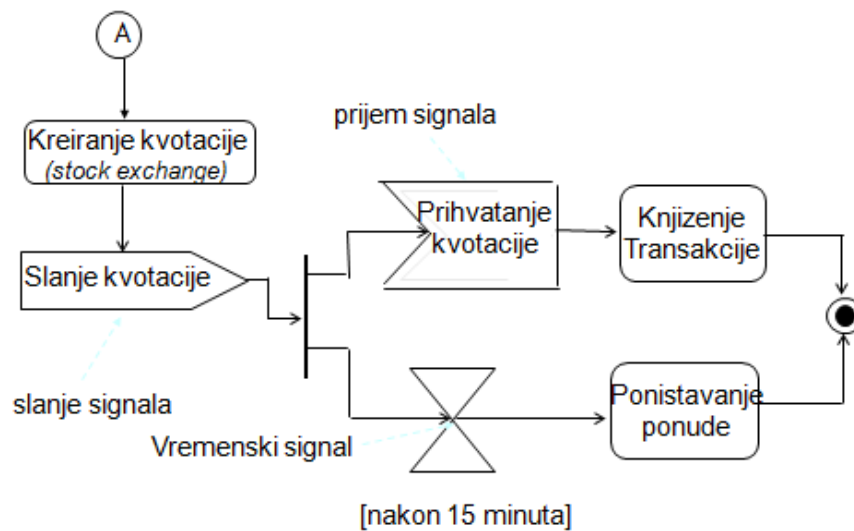


Događaji - Signali (Signals)

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

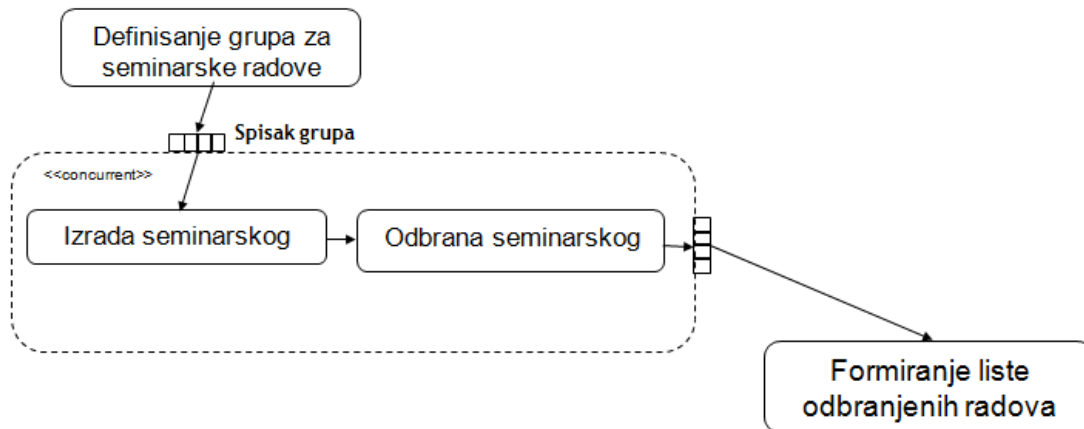


Primer:



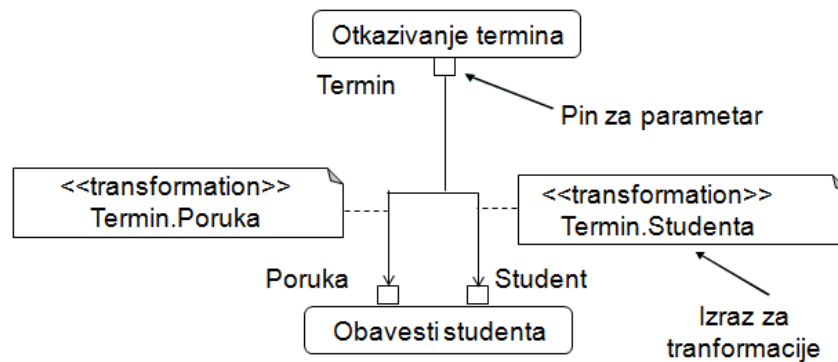
Ekspanzioni region (Expansion region)

- Koriste se za opis interakcije nad nekom kolekcijom
 - interne akcije se izvršavaju jednom za svaki element kolekcije
 - Interne akcije mogu biti sekvencijalne ili paralelne (što se prikazuje preko stereotipa <<concurrent>>) ili

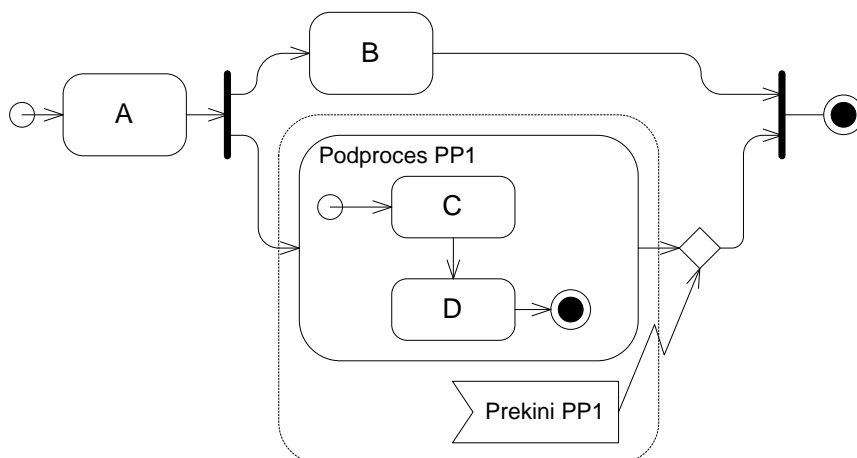


Parametri (Pins)

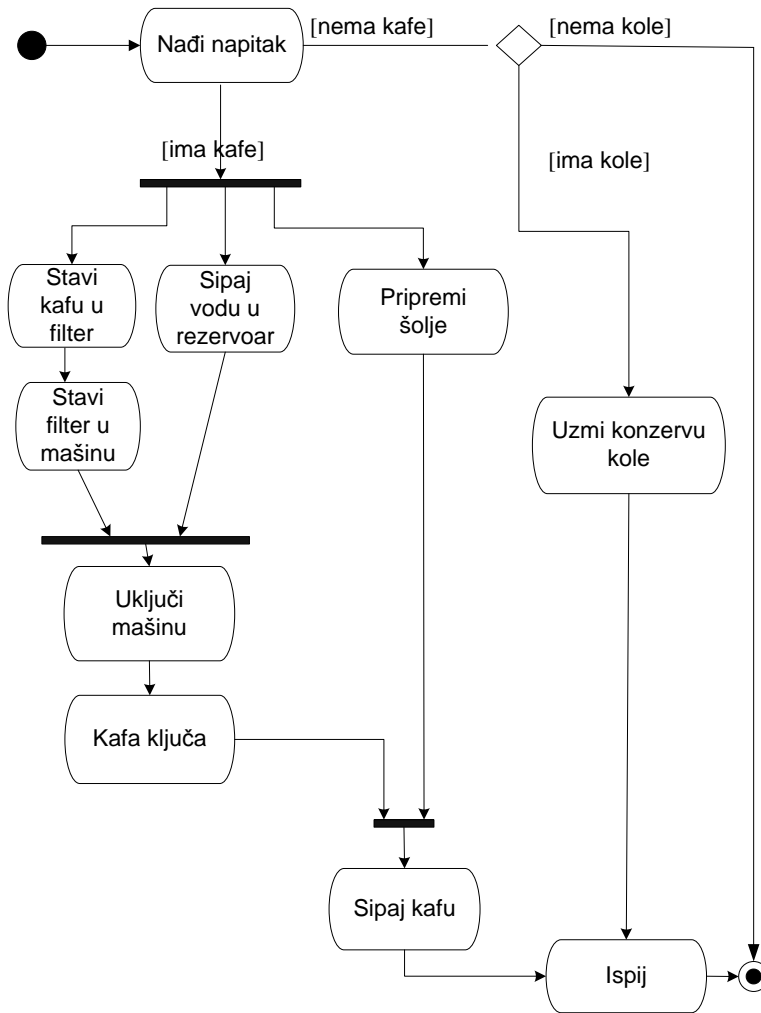
- Koncept pina se koristi za definisanje ulaznih i izlaznih parametara akcije
 - Akcije mogu imati parametre (kao metode operacije)
 - Izlazi iz neke akcije moraju odgovarati ulazima sledeće akcije
 - *Ukoliko je izlazni parametar različit od ulaznog parametara mora se prikazati preko transformacije*



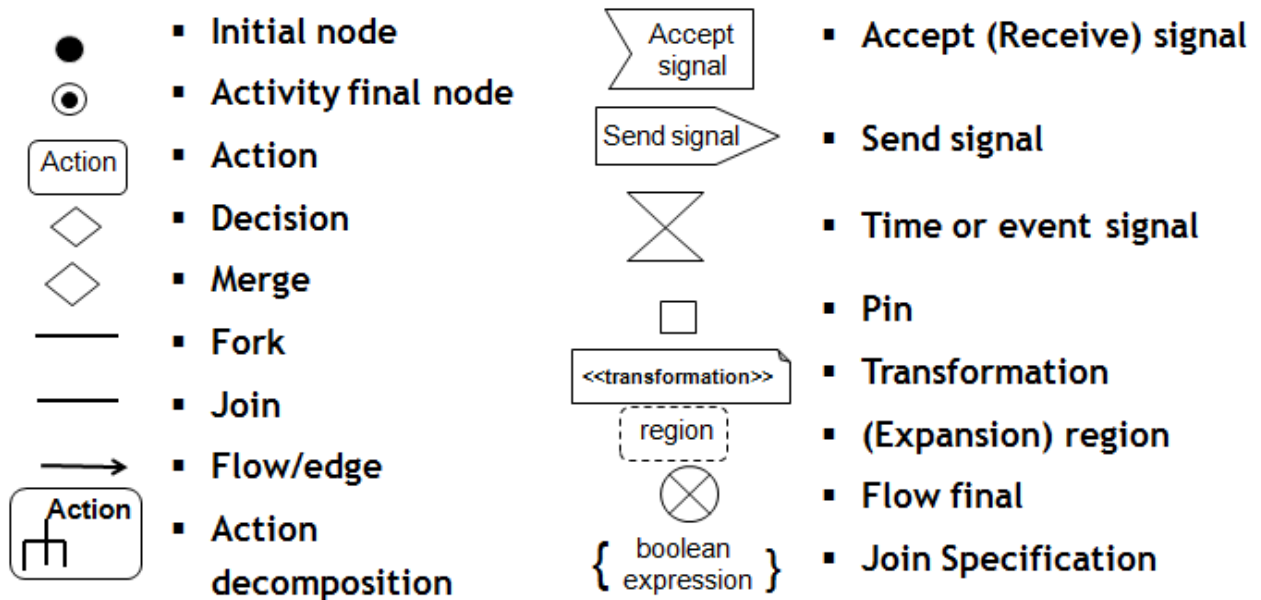
Region mogućeg prekida (Interruptible region)



Primer 1



Dijagram aktivnosti - Pregled elemenata



2. Specifikacija poslovnih procesa

Model slučajeva korišćenja

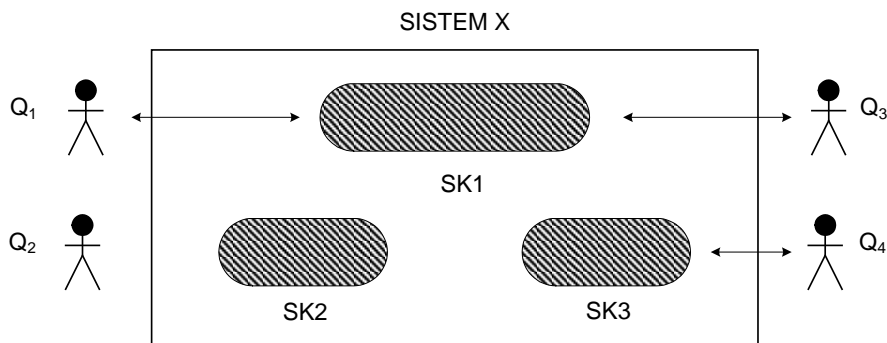
Sa tačke gledišta analize sistema i definisanja poslovnog modela "slučaj korišćenja" se definiše kao specifikacija interakcije između sistema i jednog ili više aktera i sistema zajedno sa opisom akcija sistem u ovoj interakciji.

Model slučajeva korišćenja je graf sa dve vrste čvorova:

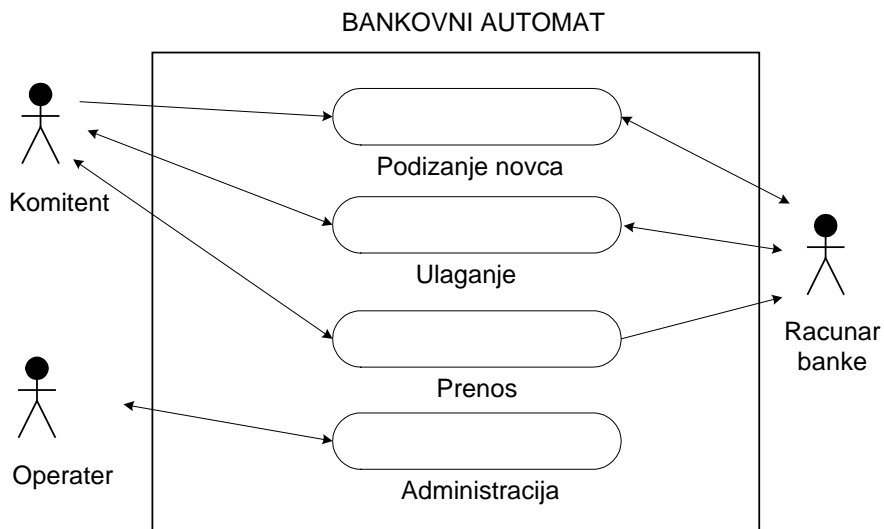
- čvorovima koji predstavljaju slučajeve korišćenja
- čvorovima koje predstavljaju aktere
 - ◆ Akter je bilo šta što stupa u interakciju sa sistemom.
 - ◆ Akter je objekat van sistema koji predstavlja tip (vrstu korisnika).
 - ◆ Akter može biti korisnik (čovjek) ili neki drugi sistem. (*Treba praviti razliku između korisnika i aktera. Korisnik je čovek koji koristi sistem, dok je akter specifična uloga koju korisnik ima u komunikaciji sa sistemom*).

Opšti model slučajeva korišćenja

Direktna komunikacija između dva aktera i dva konkretna (oni sa kojima komuniciraju akteri) slučaja korišćenja se ne može predstaviti na modelu (grafu). Međutim, kako će kasnije biti prikazano, moguće je definisati asocijaciju između klasa slučajeva korišćenja i klasa aktera (apstraktni akteri i apstraktni slučajevi korišćenja), da bi se jednostavnije prikazao neki složeni model.



Primer dijagrama slučajeva korišćenja



Opis slučaja korišćenja - SCENARIO

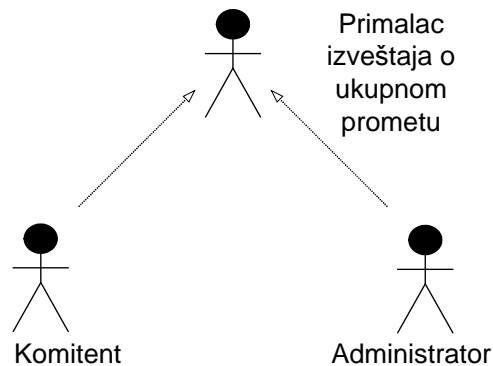
- Svaki slučaj korišćenja treba da bude detaljno opisan. Preporučuje se da se u prvoj fazi koristi strukturirani verbalni opis, jer je on neophodan čak i ako se da neki formalni opis (dijagrami kolaboracije, dijagram promene stanja).
- Uobičajeno je da se posebno daje opis normalnog toka događaja u slučaju korišćenja, a posebno mogući izuzeci.
- Jedan slučaj korišćenja predstavlja **skup sekvenci događaja**. Jedna sekvenca događaja se naziva **scenario**. Postoji osnovni scenario i skup mogućih izuzetaka i alternativnih funkcionisanja.

Veze u dijagramu slučajeva korišćenja

- **ASOCIJACIJA** - prikazana veza između aktera i slučaja korišćenja
- **GENERALIZACIJA** - veza opštijeg i specifičnijeg slučaja korišćenja koji nasleđuje opis opštijeg
- **<<extend>>** - stereotip veze zavisnosti koja referencira(ubacuje) moguće dodatno "ponašanje" opisano u posebnom apstraktnom SK, u osnovni SK
- **<<include>>** - stereotip veze zavisnosti koja eksplicitno ubacuje dodatno "ponašanje" opisano u posebnom apstraktnom SK, u osnovni SK.

Apstraktni akter

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



Kolaboracija i slučaj korišćenja

Kolaboracija je asocijacija elemenata koji u međusobnoj saradnji realizuju neki zahtev.

3. UMM 2.0

UMM (UN/CEFACT Modeling Methodology)

- Cilj UMM je da omogući definisanje razmene poslovne logike između poslovnih partnera i podršku standardizaciji poslovnih procesa.
- Metodologija koja omogućava spoznaju poslovnog znanja nezavisno od implementirane tehnologije npr. web servisa ili ebXML-a
- UMM koji omogućava modelovanje globalne koreografije B2B (business to business) scenarija.
- Pristup zasnovan na modelima. Koristi proceduru sličnu procesu razvoja softvera (od prikupljanja zahteva do projektovanja procesa). Baziran Unified Modeling Language (UML)

Globalna poslovna koreografija

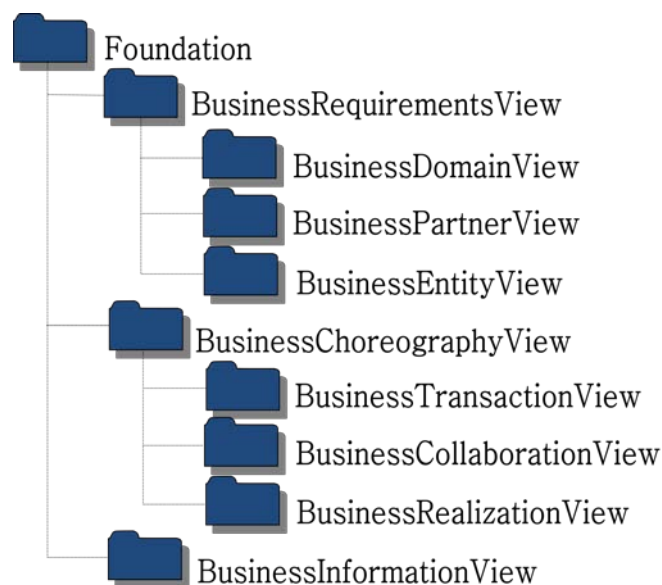
- Ukoliko svaka organizacija definiše sopstvenu koreografiju tada nije izvesna njihova interoperabilnost



- UMM opisuje kolaborativne poslovne procese sa globalnog i neutralnog gledišta



Struktura paketa - UMM 2.0



Business Domain View (BDV)

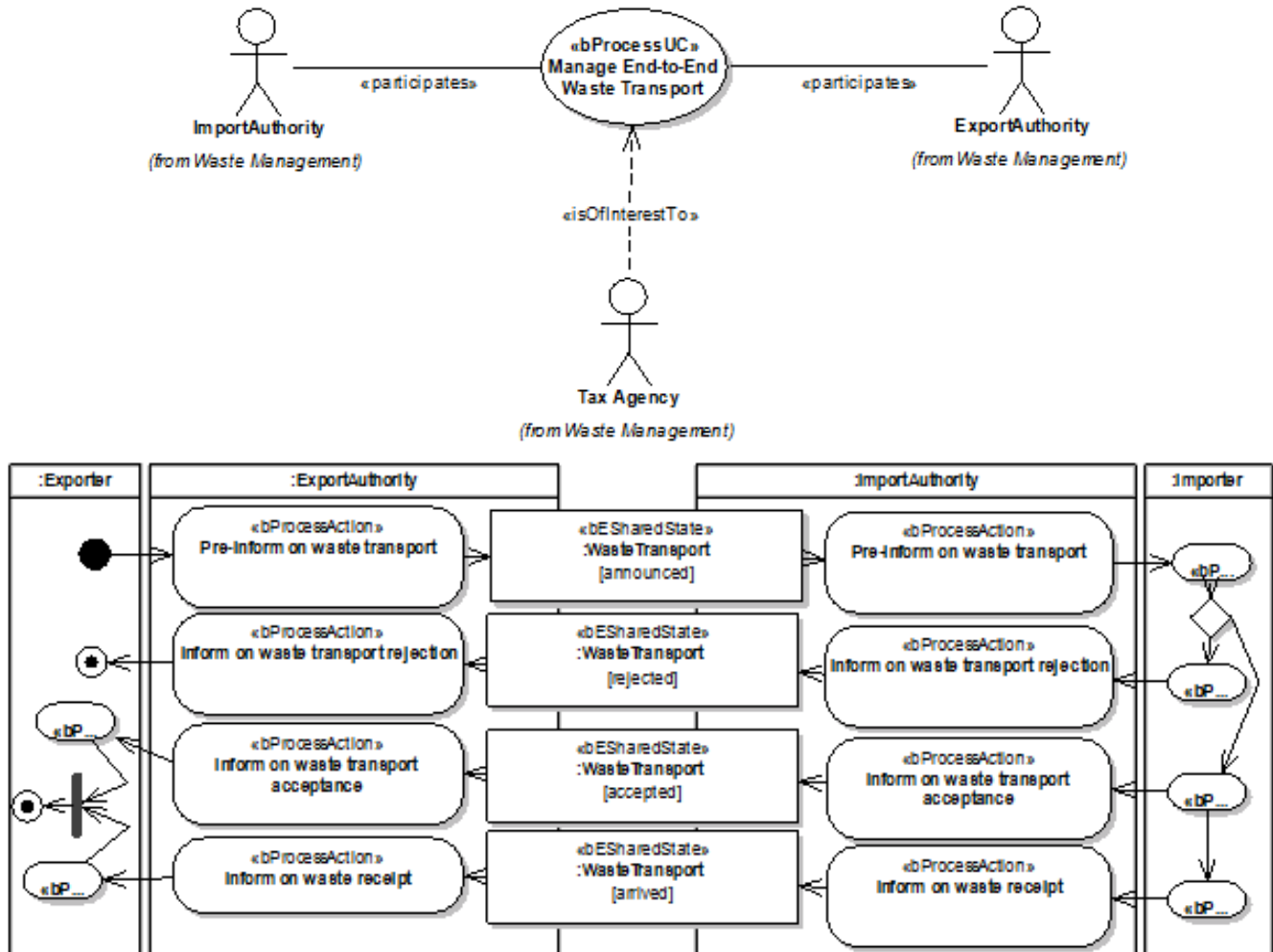
Svrha

- Razumeti domen koji se razmatra
- Otkriti postojeće poslovne procese
- Gruba identifikacija željenih kolaboracija

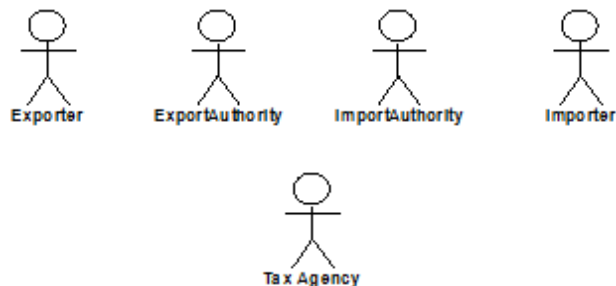
Koriste se

- Use case diagrams
- Activity diagrams

Business Domain View - primer:



Business Partner View - Primer



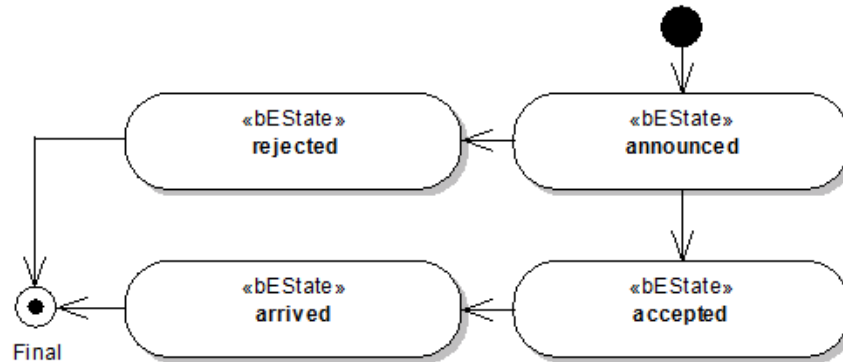
Business Entity View

Svrha

- Identifikovanje relevantnih poslovnih entiteta
- Opis životnog ciklusa poslovnih entiteta
- Stanje poslovnih entiteta se razmenjuje između poslovnih partnera

Koristi se

- Dijagram promene stanje



Business Choreography View

Business Transaction View

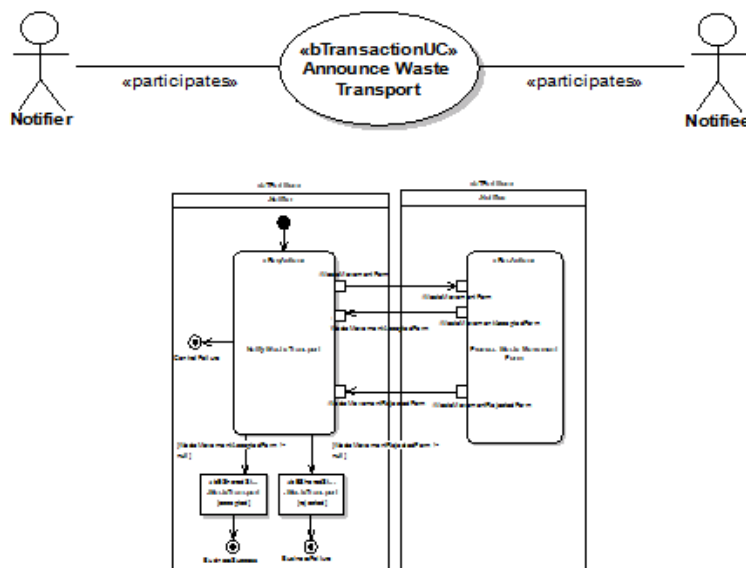
Svrha

- Poslovna transakcija opisuje poruke koje se razmenjuju između tačno dva poslovna partnera
- Razmena poruka sinhronizuje (shared) stanje poslovnih partnera

Koristi se

- Slučaj korišćenja za obuhvatanje zahteva poslovne transakcije
- Dijagram aktivnosti za opis poslovne transakcije

Business Transaction View - primer:



Business Collaboration View

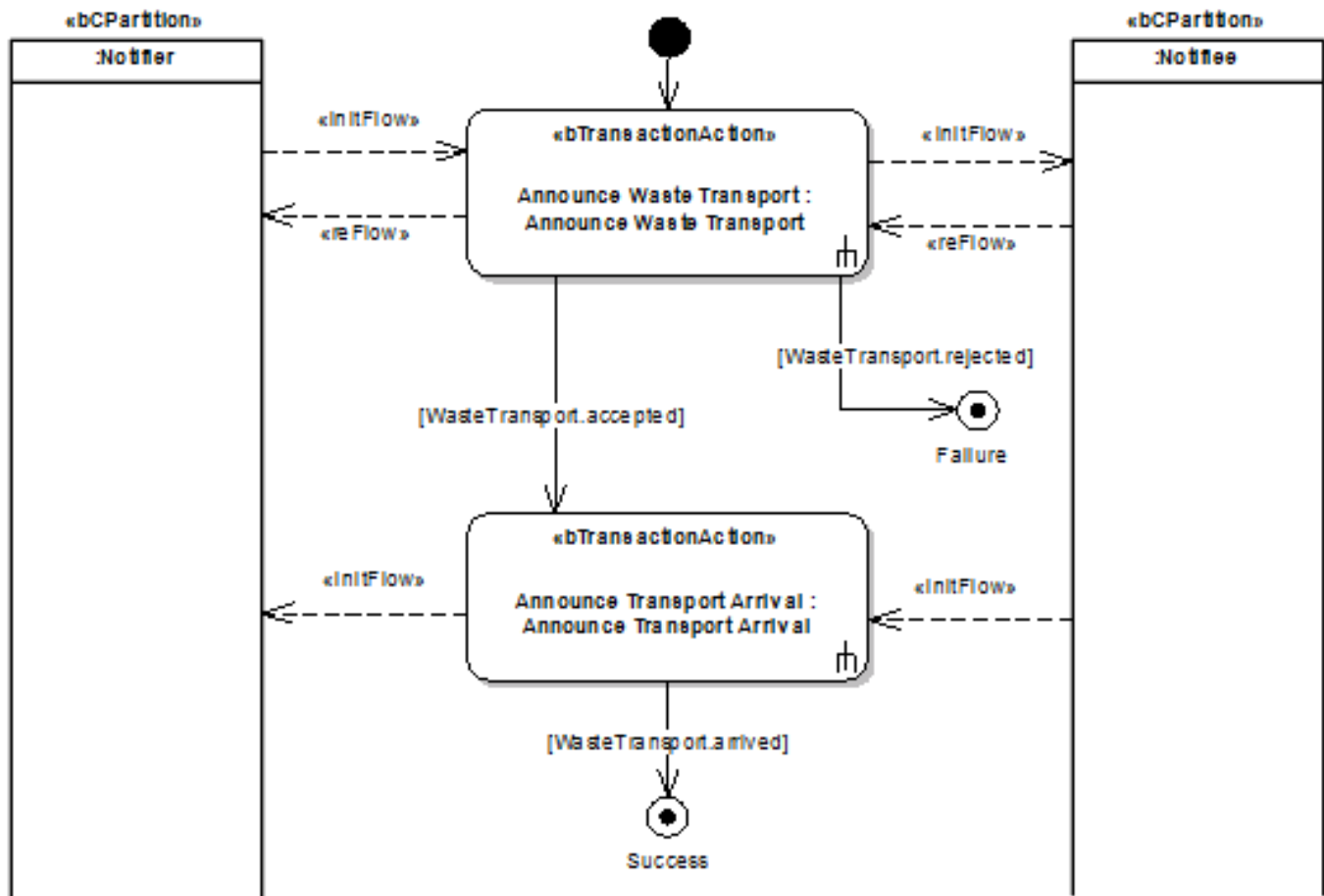
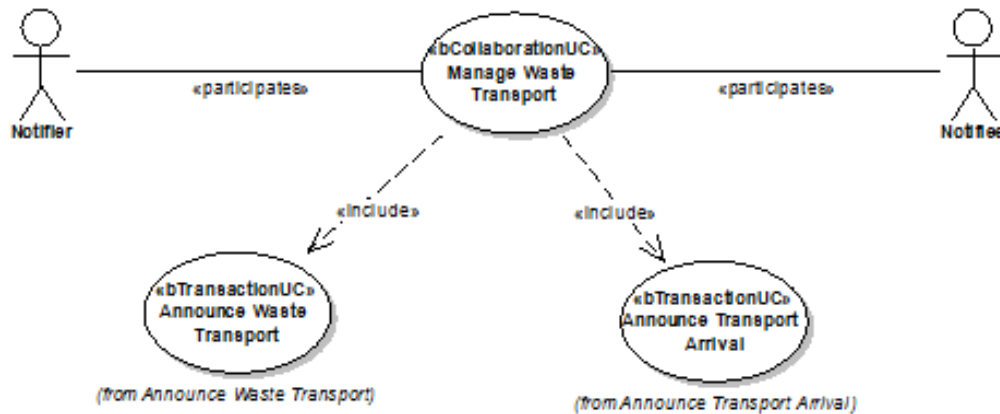
Svrha

- Opisuje poslovne kolaboracije u kojima se izvršavaju poslovne transakcije
- Poslovna kolaboracija opisuje globalnu koreografiju poslovnih procesa

Korisiti se

- Dijagram slučajeja korišćenja za indentifikovanje zahteva
- Dijagram aktivnosti za opis poslovne kolaboracije

Business Collaboration View - primer:



Business Realization View

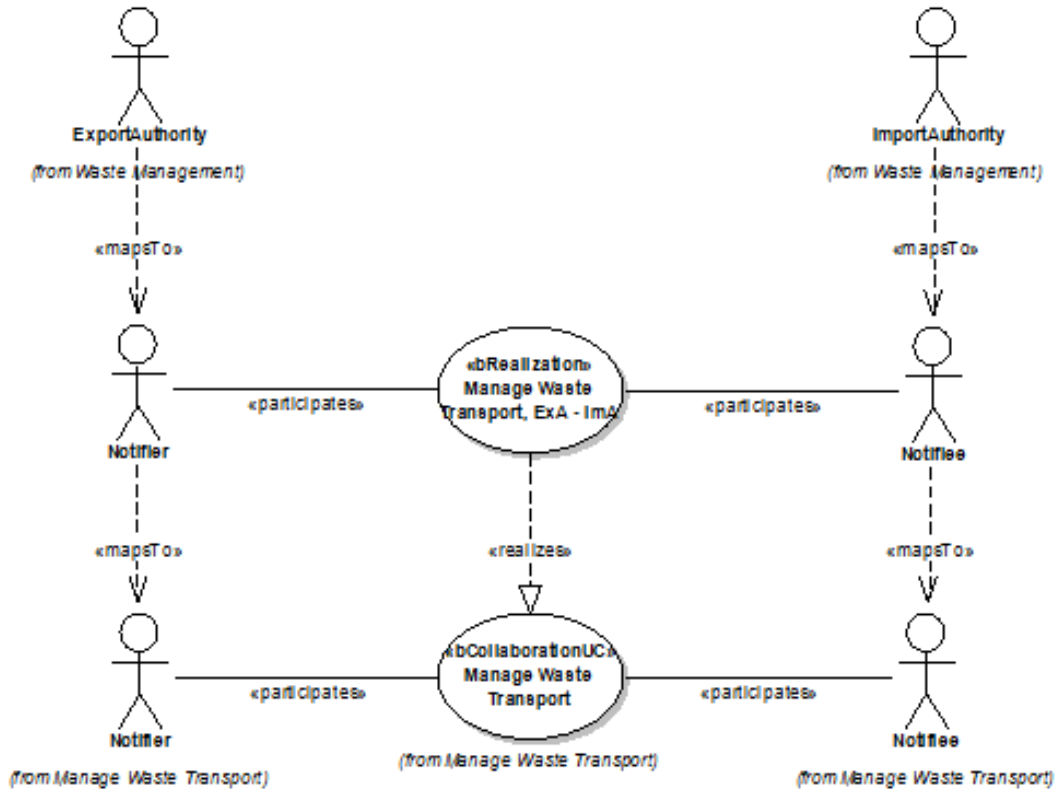
Svrha

- Dodeliti poslovnim partnerima autorizovane uloge u kolaboraciji
- Collaboration Realizations allow that different sets of business partners perform the same collaboration

Koriste se

- Slučaj korišćenja, Poslovni partneri, Autorizovane uloge

Business Realization View - primer:



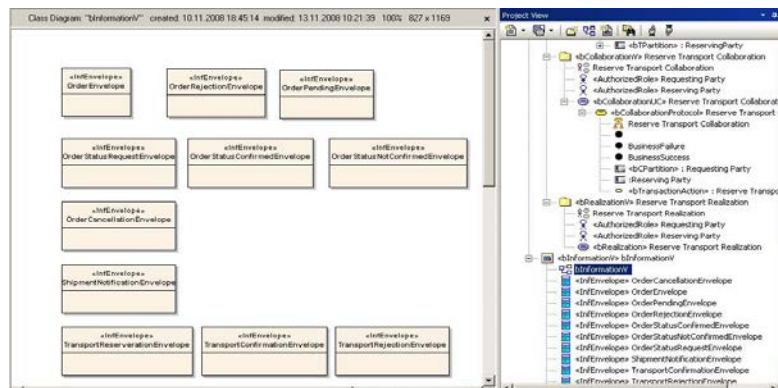
Business Information View

Svrha

- Modelovanje poslovnih informacija
- Predlaže se korišćenje UML Profile for Core Components (UPCC)

Koristi se

- Dijagram klasa



Šta omogućava UMM

- Specifikacija IT sistemih zahteva za distribuiranih servisno orjentisanih sistemom
- Definiše ugovora za globalne koreografije između poslovnih partnera
- Generisanje globalne apstraknte BPEL sintakse za implementaciju (automatizaciju) B2B poslovnih procesa
- Definiše modele koji omogućavaju kreiranje internih orkestracija procesa (UMM ne primazuje interni pogled poslovnog partnera za poslovni proces)

4. SOA

Servisno orijentisana arhitektura

Razvoj softverskih arhitektura

Vrednost softvera se sve više definiše preko povezanosti, a ne funkcionalnosti.

Potrebe za povezanošću komponenti su uticale na zbljšavanje tehnologija.

SOA i BPM (Business Process Management)

SOA se opisuje kao mogućnost konstruisanja softverski komponenti koje mogu biti korišćene u kontekstu koji nije poznat u vreme dizajna.

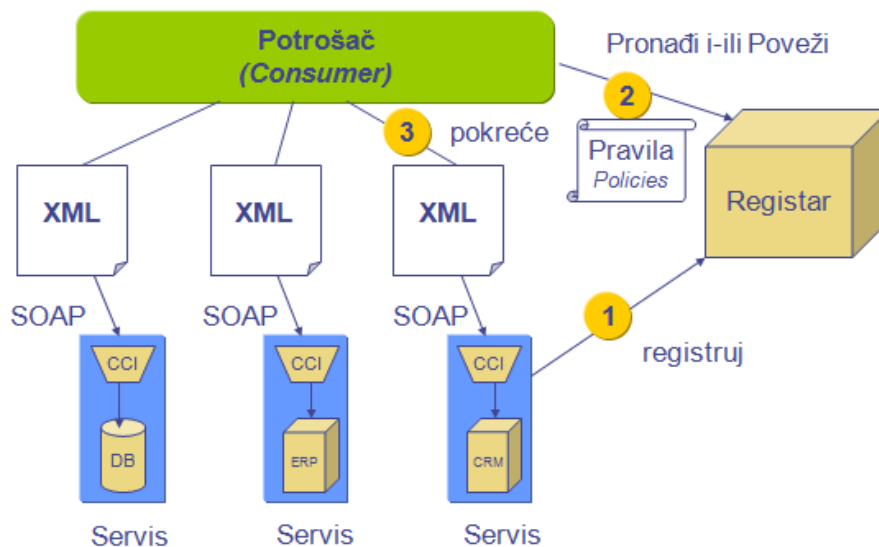
BPM se opisuje kao mogućnost preciznog modelovanja i promene konteksta u kojem će komponente preduzeća biti korišćene.

Kako se posmatra BPM u SOA zajednici

Dva pristupa u BPM

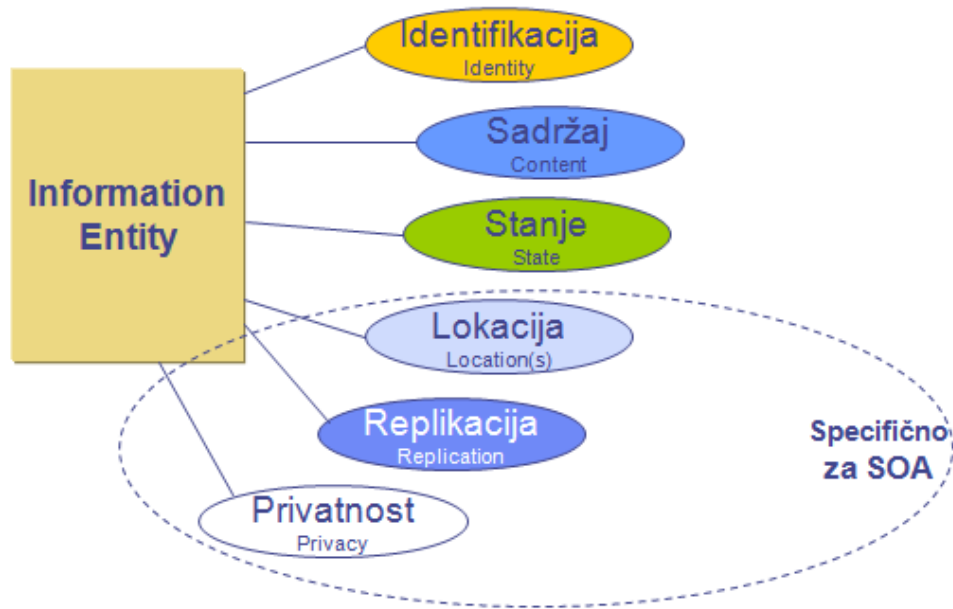
- **Zasnovan na događajima**
 - ◆ BPML, BPEL
 - ◆ Pi-Calculus (takođe Event Calculus)
- **Orijentisan ka aktivnostima**
 - ◆ WfMC
 - ◆ Petrijeve mreže (Petri nets)

Komponente postaju servisi koji se izvršavaju izvan granica potrošača



Information Entities u SOA

Karakteristike entiteta u SOA



Poslovni proces može se posmatrati kao više akterska koreografija servisa.

Servisi u SOA su orkestrirani (BPEL) - ovaj model koji omogućava najbolje ponovno korišćenje poslovne logike.

Koreografija omogućava definisanje model toka događaja između aktivnosti

SOA zahteva potpunu odvojenost poslovne logike i korisničkog interfejsa (UI - user interface)

Standardi Web servisa

Osnovni standardi za web servise su:

- XML (Extensible Markup Language)
- SOAP (simple object access protocol)
- WSDL (web services description language)
- UDDI (universal description, discovery and integration)

Arhitektura Web servisa

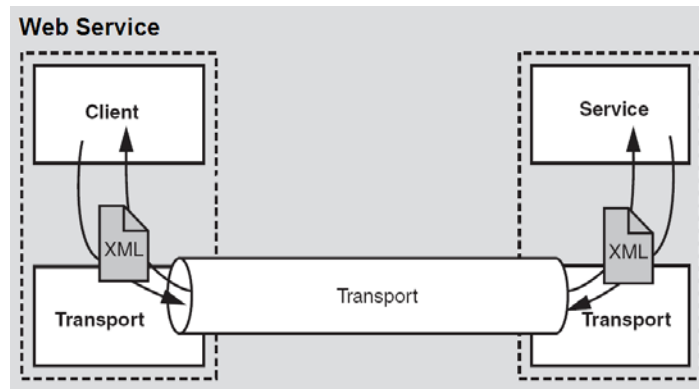
Uključuje tri glavna uloge

- Providera servisa
- Registra servisa
- Potrošača servisa

Obuhvata tri glavne operacije

- Publikovanje – registrovanje web servisa za korišćenje
- Pronalaženje – lociranje web servisa
- Izvršavanje – korišćenje web servisa

Komponente Web servisa



Dostupnost servisa

Da bi neko mogao da koristi servis mora da zna da on postoji i da se omogući:

- ◆ Otkrivanje registrovanih (publikovanih) servisa (UDDI).
- ◆ Opis interfejsa servisa (metode i argumenti) kako bi korisnik mogao da ga koristi (WSDL).
- ◆ Lokacija na kojoj se nalazi servis (web server ili specifične aplikacije za komunikaciju)

Osovne karakteristike WS

- ◆ Sva WS dokumenta su napisana u XML
- ◆ XML Schema se koristi za definisanje elemenata koji se koriste u WS komunikaciji

SOAP

- ◆ Koristi se za komunikaciju sa WS
- ◆ Zasnovan na paradigmi poruka zahtev-odgovor (SOAP poruke)
- ◆ Telo poruke (čija je gramatika data preko WSDL) se nalazi u SOAP "envelope"
- ◆ "Povezuje" korisnika i WS

SOAP poruke

- ◆ SOAP definiše standardi 'koverat' u okviru kojeg poruka može biti poslata.
- ◆ SOAP je mehanizam (protokol) za prenos informacija (poruka) između distribuiranih aplikacija.
- ◆ SOAP ne govori ništa o sadržini poruke već pošiljalac i primalac moraju međusobno da je razumeju.
- ◆ SOAP je deo komunikacionog sloja.

Struktura SOAP-a

Svaka SOAP poruka sadrži:

- Envelope - obavija ceo dokument
- Header (opciono - sadrži dodatne informacije za npr. autentifikaciju)
- Body - sadržaj poruke koji je sama XML poruka ili RPC (sadrži XML koji predstavlja poziv metode)
- Body može sadržati Fault element kojim se definišu informacije o izuzecima

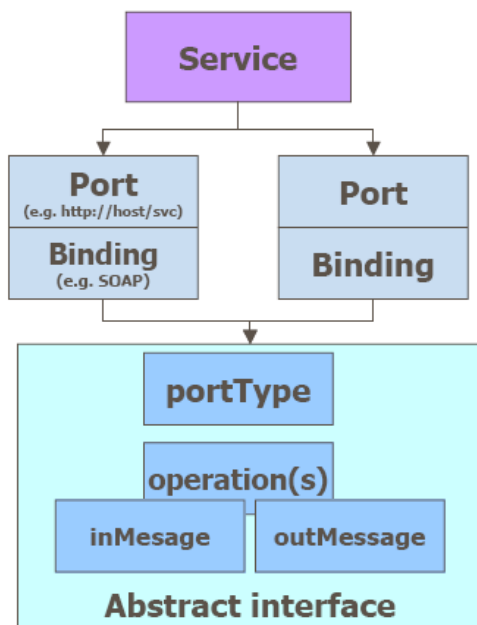


WSDL

- ◆ Opisuje WS i definiše funkcije koje su dostupne na njemu
- ◆ Definiše XML gramatiku koja se koristi u porukama

Struktura WSDL-a

- ◆ portType - apstraktna definicija servisa (skup operacija)
- ◆ Binding - načini pristupa izvršavanju servisa
- ◆ Port - gde se pristupa servisu



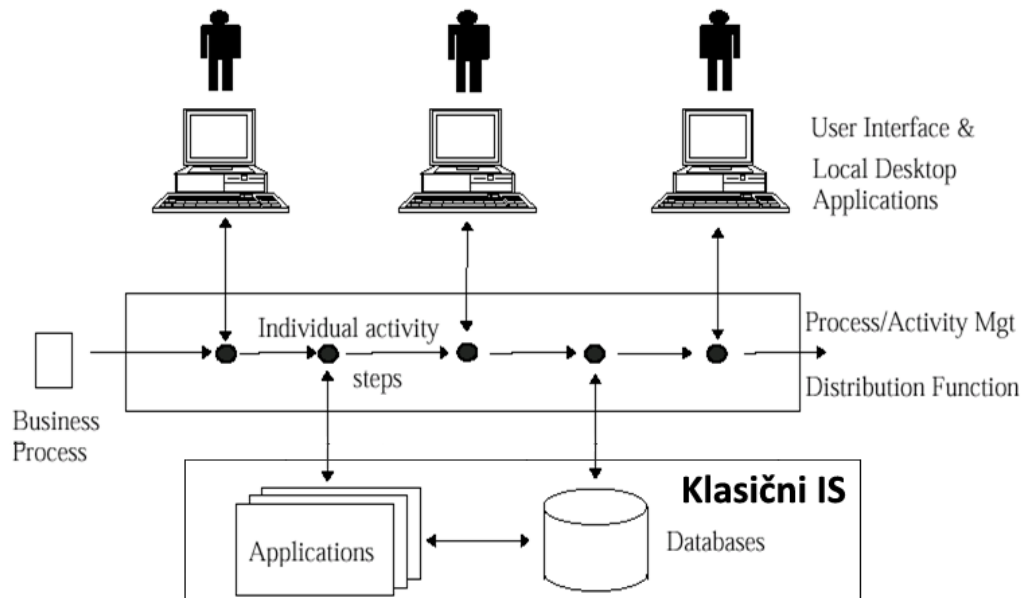
UDDI

- ◆ Koristi se za registrovanje i pronalaženje servisa u nekom centralnom registru
- ◆ Provajderi servisa mogu publikovati informacije o njihovim poslovima i servisima koje nude
- ◆ Korisnici servisa mogu pretraživati servise organizovane po poslu kategoriji servisa ili kao specifične servise

5. Sistemi za upravljanje poslovnim procesima

Automatizacija poslovnih procesa

- radni tok (eng. Workflow)



Nedostatci klasičnih IS

- Integracija poslovnih procesa preko baze podataka
 - ◆ Problem zajedničkog funkcionisanja (eng. interoperability) kod poslovnih procesa koji su geografski distribuirani i koriste različita tehnološka rešenja
- Ne postoji automatizacija radnog toka (workflow)
 - ◆ Korisnici sami moraju da "znaju" koju aplikaciju kada da pokrenu
- Ne postoji podrška za upravljanje poslovnim procesima
 - ◆ Rukovodioci nemaju automatizovanu podršku za upravljanje
 - ◆ Rezultati i efekti rada izvršilaca su samo posredno vidljivi preko efekata aplikacija koje koriste

Softverska infrastruktura

Klasifikacija sistema

- Alati za kancelarijsko poslovanje (automatizacija radnog mesta: obrada teksta, email, kalendar, ...)
- Sistemi za skeniranje i arhiviranje (Imaging and archiving systems)
- Sistemi za pretraživanje (Digital Libraries and Search engines)
- Korporativni DMS (Enterprise DMS)
- Sistemi za grupni rad (Groupware Systems)
- Sistemi za rad sa geografskim kartama i mapama (GIS)
- Sistemi za upravljanje poslovnim procesima (Workflow Systems)

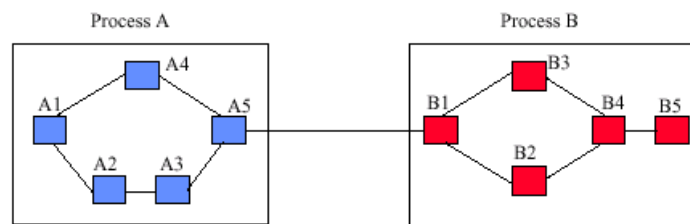
Osnovne karakteristike WfMS

Evolucija u razvoju

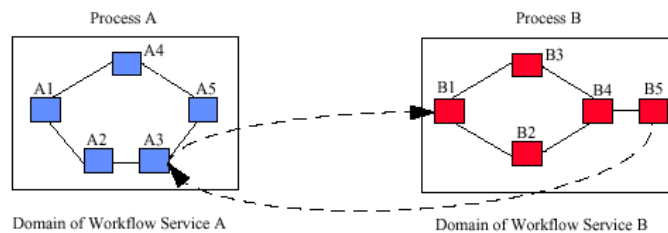
- Document based workflow
 - ◆ Lotus Notes, FileNet, Documentum, ...
- Message based workflow
 - ◆ MS Exchange
- Activity based workflow
 - ◆ Staffware, IBM MQSeries, Oracle Workflow
 - ◆ Workflow management Coalition (WfMC) standard

Integracija poslovnih procesa

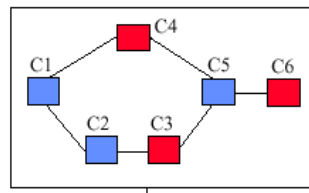
■ Olančani procesi



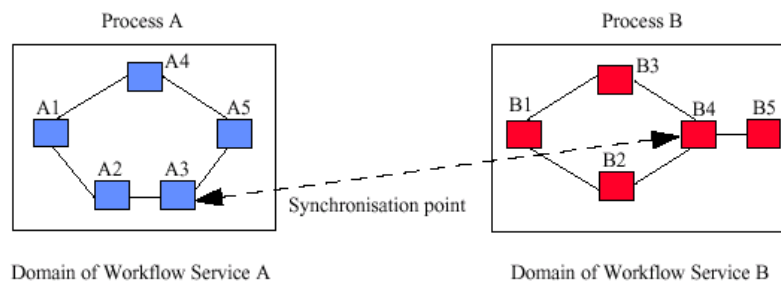
■ Ugnježeni (podproces)i



■ Ravnopravni (Peer-to-peer)



■ Paralelni



Problemi i izazovi

- ◆ Slabo strukturirani poslovni procesi
 - Neohodna fleksibilnost za poluformalno i slabo strukturirane procese
 - Postojeće tehnike previše “krute”
- ◆ Integracija poslovnih procesa

Integracija poslovnih procesa

- ◆ Organizacioni-ekonomski aspekt
 - Integracija procesa različitih poslovnih partnera
 - Podrška za “outsourcing” procesa
- ◆ Tehničko-tehnološki aspekt
 - Heterogena SW infrastruktura
 - Dinamičko otkrivanje i povezivanje
- ◆ Metodološki aspekt
 - Modelovanje i specifikacija
 - Različiti modeli – ontološki problem

6. BPEL

Karakteristike i zahtevi poslovnih sistema

Danas poslovni sistemi zahtevaju **brzu** prilagodljivost potrebama korisnika i uslova tržišta

- EAI i B2B interakcije (pomoću web servisa)

Potrebe za **internom** and **externom** fleksibilnošću

Bez zajedničkog skupa standarda, svaka organizacija svoje **specifične** poslovne protokole

- Mala fleksibilnost za pravu web servis kolaboraciju

Kompozicija Web servisa

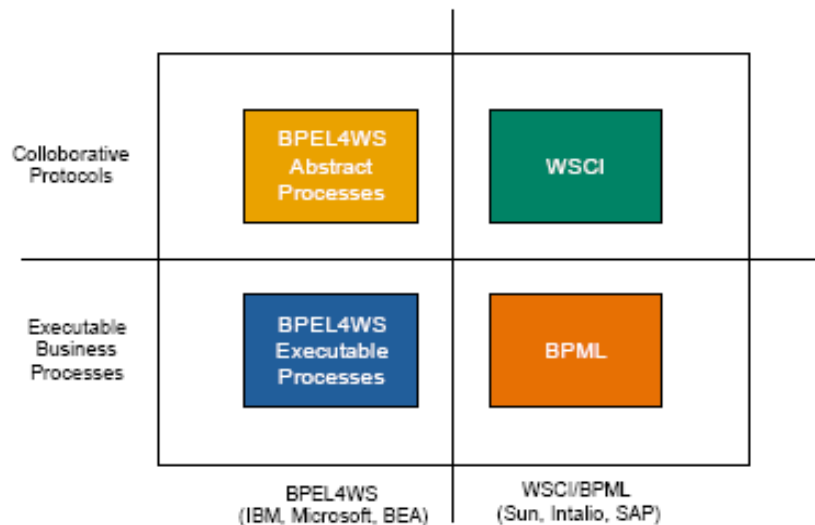
Definicija:

- Obezbeđuje **otvoren, baziran na standardu** pristup povezivanju web servisa da bi se kreirali poslovni procesi **višeg nivoa**.
- Standardi su definisani da bi **smanjili složenost** pri kompoziciji web servisa, kako bi **smanjili vreme i troškove**, i **povećali ukupnu efikasnost** u poslovanju

Standardi

- **BPEL4WS** (a.k.a. BPEL) – Business Process Execution Language for Web Services
 - ◆ IBM and Microsoft
- **WSCI** – Web Services Choreography Interface
 - ◆ Sun, SAP, BEA, and Intalio
- **BPML** – Business Process Management Language
 - ◆ BPMI.org (chartered by Intalio, Sterling Commerce, Sun, CSC, and others)

Standardi za kompoziciju servisa u odnosu na dva modela procesa (apstraktni i izvršni)



BPEL procesi i interkcije su stogo tipizirani

- ◆ BPEL se gradi pomoću WSDL opisa servisa
 - Stogo tipiziran
- ◆ Interkacija BPEL procesa preko skupa tipiziranih kanala:
 - Partner link = par WSDL port types
 - BPEL tokovi su usko povezani sa specifičnim interfejima i tipovima poruka
- ◆ BPEL podržava tradicionalne WfMS osobine:
 - Statically analyzable
 - Transakcije, pouzdanost, monitoring izvršavanja, itd.
 - Odgovara (SOA) modelu aplikacija za integraciju aplikacija preko dobro definisanih interfejsa servisa
- ◆ BPEL je proširen za podršku interkacije sa ljudima kao akterima
 - BPEL4People: servisi koje izvršavaju ljudi

BPEL4WS

- ◆ Baziran na XML
- ◆ Opisuje **kontrolnu logiku** za koordinaciju web servisa u okviru nekog poslovnog procesa
- ◆ **Interpretira** se i **izvršava** od strane BPEL engine-a

BPEL

Kao osnovnu koristi Web servis standard

1. Svaki BPEL se **predstavlja kao web servis** korištenjem WSDL. WSDL opisuje javne ulazne i izlaze tačke procesa
2. **Interakcija preko WSDL interfejsa** sa eksternim web servisima
3. **WSDL tipovi podataka** se koriste za opis tokova informacija u okviru BPEL procesa

Koncepti u BPEL4WS

- Partner Links:
 - ◆ Bilateralna konverzacija (my Role, Partner Role)
 - ◆ Interfjes zahtevi partnera
- Podaci i poruke:
 - ◆ Variable
 - ◆ Osobine
 - ◆ Skup korelisanih podatka
- Aktivnosti:
 - ◆ Osnovne aktivnosti
 - ◆ Strukturirane aktivnosti (control flow)

Partner Links

Partner link: tipizirane instance konektora

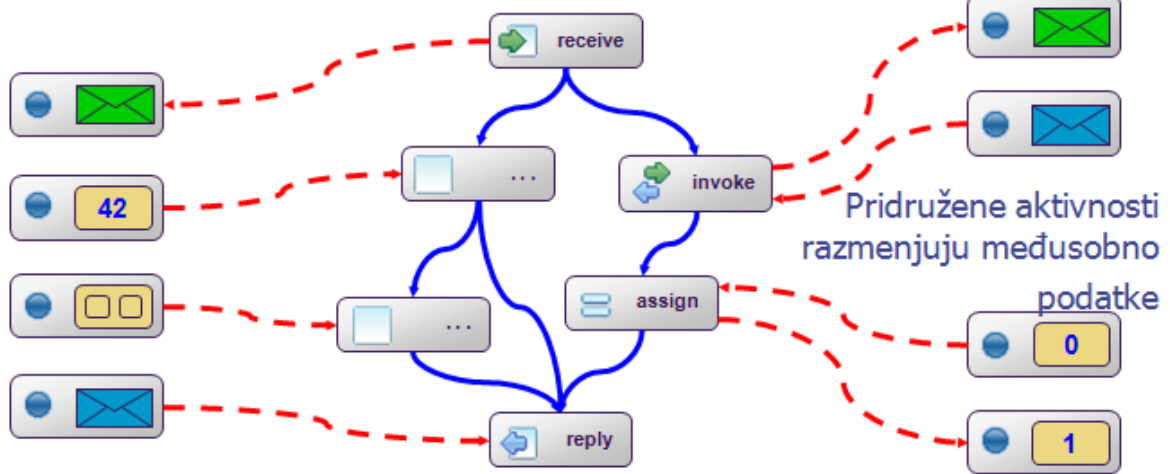
- Partner link type definiše zahtevane tipove portova
- Kanal preko kojeg se vrši peer-to-peer konverzacija

BPEL model: Variable

Variable definisanje preko

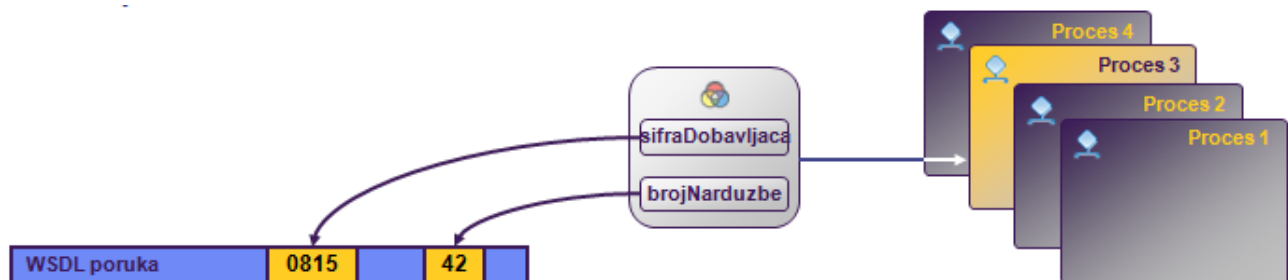
WSDL poruka ili

XML Schema elemenata/tipova



Osobine i korelisani skup podataka

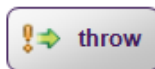
- Poruke u vremenski dugoj konverzaciji su povezane sa odgovarajućom instancom procesa
 - ◆ Tipizirane osobine definisane u WSDL su imenovane i mapirane sa delovima WSDL poruka koje se koriste u procesu



Osnovne aktivnosti



Prihvatanje poruka koje stižu (čekanje na odgovora)



Generisanje greške unutar poslovnog procesa



Slanje poruke kao odgovor na primljenu poruku



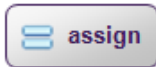
Prosledjivanje greške kad obradivaču greške



Pokretanje jednostrane ili dvostrane request-response operacije



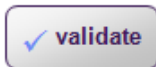
Trenutni prekid izvršavanja instance poslovnog procesa



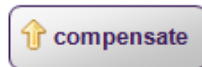
Izmena vrednosti varijabli ili partner linkova sa novim podacima



Čekanje neki definisani vremenskom period ili dok taj period ne istekne



Validacija XML podataka u varijablama



Pokretanje alternative u unutrašnjem opsegu koji se već izvršio



"prazna" instrukcija za poslovni proces

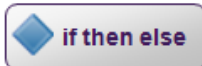
Strukturirane aktivnosti



Sadrži aktivnosti koje se paralelno izvršavaju



Blokiranje i čekanje na odgovarajuću poruku



Selekcija jedne grupe aktivnosti iz skupa



Sadrži aktivnosti koje se izvršavaju sekvencijalno ili paralelno kontrolisano preko brojača



Sadrži aktivnosti čije se izvršavanje ponavlja sve dok je zadovoljen dati uslov



Sadrži aktivnosti koje se izvršavaju sekvencijalno



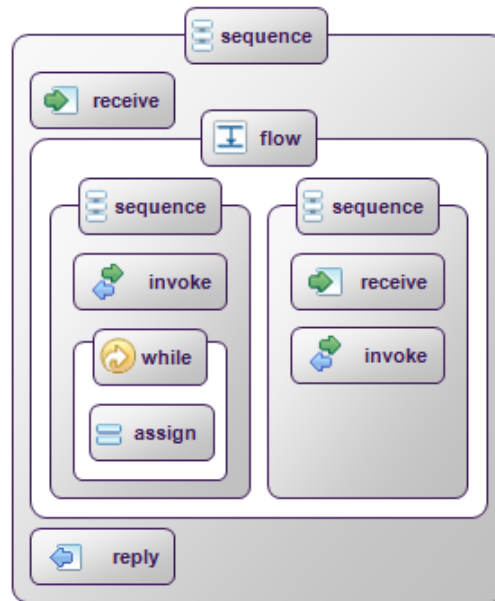
Iteracija aktivnosti dok nije zadovoljen dati uslov



Povezivanje aktivnosti sa lokalnim varijablama, obradivačima grešaka i obradivačima događaja

Ugnježdavanje strukturanih aktivnosti

```
<sequence>
  <receive .../>
  <flow>
    <sequence>
      <invoke .../>
      <while ... >
        <assign>...</assign>
      </while>
    </sequence>
    <sequence>
      <receive .../>
      <invoke ... >
    </sequence>
  </flow>
  <reply>
</sequence>
```



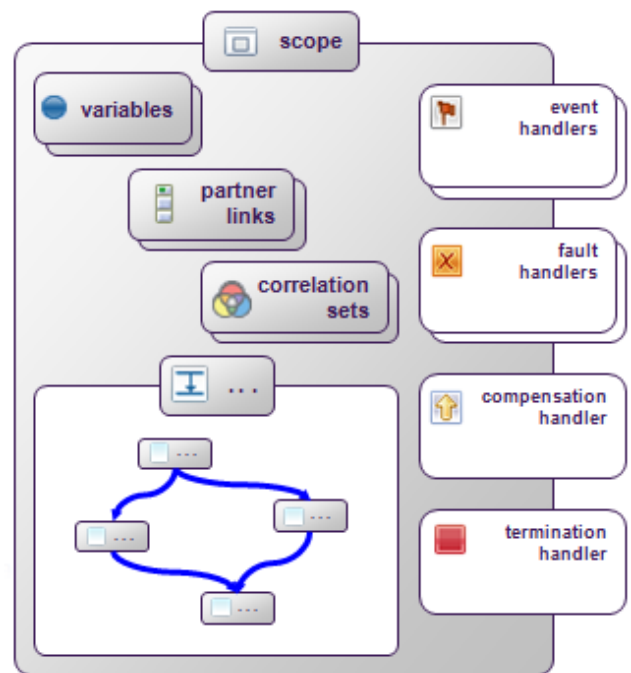
Opseg i obrađivači

Opseg (Scope)

- Lokalne varijable
- Lokalne partnerske linkove
- Lokalne skup korelisanih osobina (correlation sets)
- Skup aktivnosti (osnovne i strukturiane)

Obrađivači (Handlers)

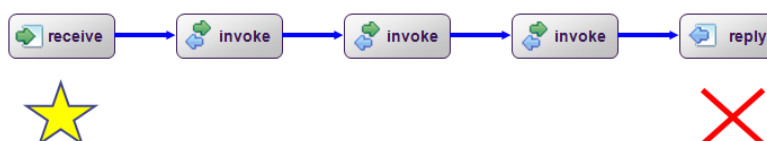
- Obrađivači događaja
 - ♦ Poruke događaji ili vremenski događaji
- Obrađivači grešaka
 - ♦ Obrada različitih internih izuzetaka (internih grešaka)
- Kompenzacioni obrađivač
 - ♦ Poništavanje trenutnih efekata kompletiranih aktivnosti
- Obrađivač prekida
 - ♦ Obrada forsiranog prekida (eksterne greške)



Životni ciklus instance procesa

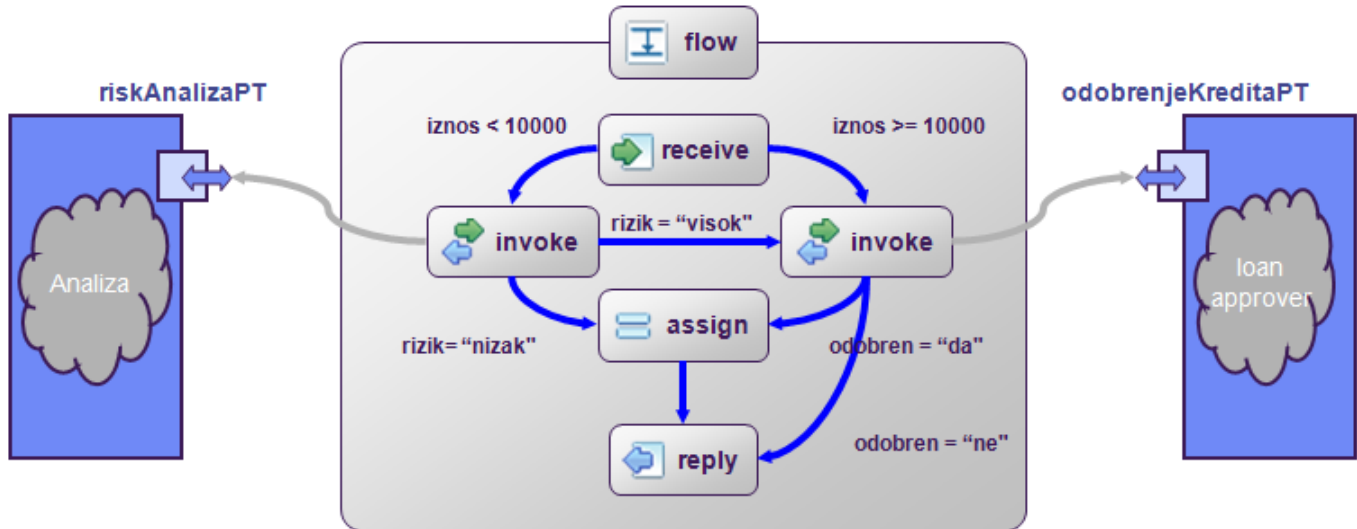
Poslovni procesi u BPEL predstavljaju Web servise koji čuvaju stanje (statefull)

- ♦ Kada se proces pokrene, kreira se nova instanca procesa
- ♦ Kreiranje i uništavanje instance procesa je implicitno



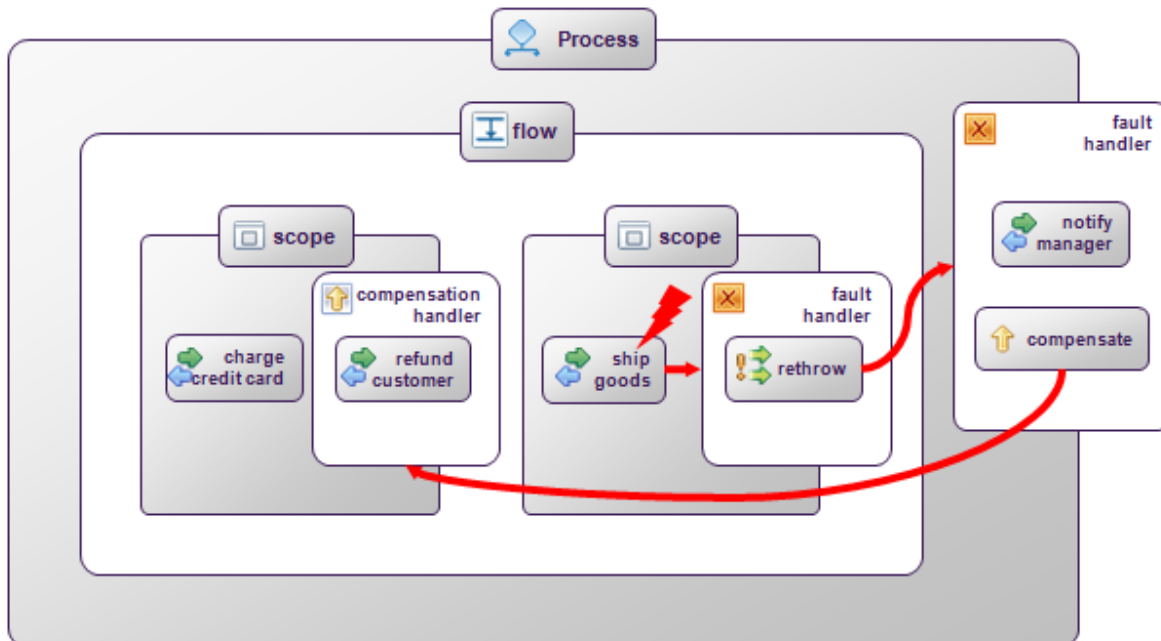
BEPL - primer

1. Klijent podnosi zahtev za kredit koji sadrži podatke o klijentu i zahtevanom iznosu kredita
2. Dva servisa su uključena:
 - a) Risk analiza koja definiše stepen rizika
 - b) Odobravanje kredita proverava ime i odobrava/neodobrava zahtev za kredit
3. Odgovor se vraća klijentu



Obrada grešaka

- ◆ Kada se generiše greška, bilo implicitno od strane BPEL engine ili eksplicitno preko **throw** aktivnosti, kontrola toka se prebacuje na obrađivač greške za dati tip greške.
- ◆ Obradivač grešaka je **catch** struktura



WS-BPEL

- ◆ Interoprebilan i prenosiv model poslovnih procesa
- ◆ Fleksibilna integracija Web servisa
 - WSDL apstraktni interfejsi se kombinuju da bi definisali kompoziciju
 - Apstraktni partneri mogu biti povezani sa aktuelnim servisima u vreme izvršavanja
 - Proces može izabrati protokol za komunikaciju sa servisom u vreme izvršavanja
 - Servisi čija definicija podataka nije odgovarajuća mogu biti korišćeni u kompoziciji
 - ◆ Transformacija podataka može biti data zajedno sa definicijom procesa

WS-BPEL alati (engine)

- ◆ Active Endpoints ActiveWebflow Server
- ◆ ActiveBPEL Engine (open source)
- ◆ bexee BPEL Execution Engine (open source)
- ◆ Cape Clear Orchestrator
- ◆ FiveSight PXE
- ◆ IBM WebSphere Business Integration – Server Foundation 5.1
- ◆ IBM WebSphere Process Server 6.0
- ◆ OpenLink Virtuoso Universal Server
- ◆ OpenStorm ChoreoServer
- ◆ Oracle BPEL Process Manager
- ◆ Parasoft BPEL Maestro
- ◆ SeeBeyond eInsight BPM
- ◆ Twister (open source)

BPEL - Procesni modeli

- ◆ Obezbeđuje podršku za dva procesna modela
 - **Izvršne**
 - ◆ Modeli ponašanja učesnika u specifičnoj poslovnoj interakciji, **privatni workflow**
 - **Apstarktne**
 - ◆ Poslovni protokoli u BPEL, specificiraju javne **razmene poruka** između aktera (organizacija)

BPEL - Objektni model

BPEL-SPE (Sub-Process Extension)

- ◆ Proširenje BPEL sa podprocesima
 - Pokretanje nekog poslovnog procesa kao podprocesa unutar drugog tako da životni ciklus zavisi od nadređenog procesa
 - Definisanje poslovnog procesa u kontekstu drugog tako da se omogućava pristup podacima iz nadređenog procesa
 - Mogućnost pokretanje podprocesa koji se izvršava unutar drugog BPEL engine-a.

BPEL4People

- ◆ Proširenje BPEL koje uključuje u poslovne procese specijalne implementacije aktivnosti koje možemo nazvati ljudske aktivnostima (aktivnosti koji se izvršavaju od strane ljudi). Ljudske aktivnosti nazivaju se poslovi (tasks)
- ◆ Definisanje korisnika kao inicijatora procesa

Zaključak

- ◆ BPEL se fokusira na kreiranje poslovnih procesa
- ◆ Opisuje **kontrolnu logiku** za koordinaciju web servisa u okviru nekog poslovnog procesa
- ◆ Baziran na XML
- ◆ **Interpretira** se i **izvršava** od strane BPEL engine-a

7. MDA

OMG Model Driven Architecture (MDA)

- ◆ Visko nivo - platformski nezavisni modeli
- ◆ Tehnološki modeli
- ◆ Preslikavanja
 - Specifična
 - Standardna
- ◆ Standardi modeli proizvode tehnološki specifične standardne modele

Cilj MDA

- ◆ Uz pomoć UML modela viskog nivoa, definisanih precizno preko profila
- ◆ Sa specifičnim preslikavanjima u tehnološko okruženje
- ◆ Da bi se kreirali delovi izvršnog sistema

Prednosti MDA

- ◆ Razdvaja domenske specifičnosti od platformskih detalja
 - Samnjuje složenost
 - Zadržava semantiku domenskog modela
 - Povećava stabilnost i životni vek
 - Generiše izabrano platformsko okruženje
- ◆ Smanjuje vreme razvoja
 - Brz iterativan razvoj
 - Razdvojenost poslovnih i inženjerskih zahteva
- ◆ Povećava kvalitet.
- ◆ Zasnovan na industrijskim standardima

MPP metodologije i tehnologije

Platformski nezavisni modeli

UMM

BPMN

BOV spec

FSV spec

**ebXML
[BPSS]**

BPEL

WS-CDL

Platformski specifični modeli

Preslikavanje BPMN u BPEL

- ◆ Preslikavanje internih procesa je direktno (preko anotacija)
- ◆ Preslikavanje kolaborativnih procesa nije podržano u trenutnoj verziji
- ◆ Definirano je i preslikavanje u ebXML BPSS

Specifična softverska rešenja BizTalk i WWF

Microsoft BizTalk server

- ◆ softverski proizvod koji se fokusira na integraciju poslovnih procesa preko orkestracije
- ◆ Omogućava dva scenarija integracije:
 - 1) EAI – enterprise application Integratio
 - 2) B2B – business-to-business integration
- ◆ Povezuje različite aplikacije koje se izvršavaju na različitim platformama i implementirane različitim tehnologijama
- ◆ Adapteri za komunikaciju sa ostalim sistemima
- ◆ Podržava BPEL samo kao mehanizam razmene opisa procesa, tj. nije izvršan
- ◆ Implementacija poslovnog procesa iz 3 koraka:
 1. Definisanje šema poruka (BizTalk Editor)
 2. Mapiranje šema poruka (BizTalk Mapper)
 3. Definisanje poslovne logike (Orchestration Designer)

Window Workflow Fondation

- Implementacija poslovnih procesa koja zahteva učešće korisnika (objedinjuje system i human workflow)
- Proces se izvršava na klijentu

Windows Workflow Fondation - WWF

- ◆ Komponente
 - Base Activity Library: nezavisne aktivnosti i osnova za izgradnju specifičnih
 - Runtime Engine: Izvršavanje i upravljanje stanjem workflow-a
 - Runtime Services: Omogućavaju fleksibilnost i komunikaciju
 - Visual Designer: grafička konstrukcija procesa