



Upravljanje poslovnim procesima

5. Koreografija procesa

Katedra za informatiku
nastavnik: Zarić dr Miroslav



Sadržaj

- Uvod
- Motivacija i terminologija
- Faze razvoja
- Dizajn koreografije procesa
- Implementacija koreografije procesa
- Šabloni interakcija procesa
- *Let's Dance*
- Modelovanje koreografija u BPMN



Uvod

- Prethodno smo razmotrili orkestraciju procesa
- Zavisnosti između aktivnosti nisu ograničene samo na aktivnosti u jednom procesu (orkestraciji), već postoje i između aktivnosti u različitim procesima
- Kako bi izvršile ovakvu interakciju, orkestracije procesa sarađuju – najčešće razmenom poruka
- Koreografije obezbeđuju interoperabilnost orkestracija procesa koje izvršavaju određeni učesnici u B2B saradnji

Uvod (2)

- Postoje inicijative da se koreografije standardizuju u određenim domenima
 - RosettaNet za nabavke
 - SWIFTNet za finansijske usluge
 - HL7 za zdravstvene usluge
- Definišu se pravila koja učesnici moraju poštovati kako bi učestvovali u međusobnoj saradnji
- Uvođenjem pravila saradnje, smanjuju se troškovi, jer nije neophodno za svaku novu komunikaciju ugovarati modele saradnje

Uvod (3)

- Pravila saradnje se definišu koreografijama
- Iako standardizacija u određenim domenima doprinosi saradnji, nedostaje joj opšta primenljivost
- Neophodna je standardizovana definicija i implementacija koreografija

Motivacija i terminologija

- Kompanije moraju da sarađuju kako bi ponudile nove proizvode/usluge na tržištu
- B2B saradnja je od presudne važnosti za uspešno funkcionisanje kompanija
 - Dizajniranje sistema koji obezbeđuje interakciju mora biti vrlo pažljivo urađeno
- Zahtevi koji se postavljaju za razvoj koreografije procesa zavisi od broja partnera i nivoa automatizacije
 - Za poslove koji se obavljaju faksom, telefonom, detaljan dizajn koreografije nije preterano bitan
 - Za interakciju koju bi delimično ili u potpunosti trebali podržavati informacioni sistemi zainteresovanih strana, neophodan je detaljan i nedvosmislen model koreografije
 - Pogrešno definisana interakcija može dovesti do nemogućnosti izvršenja posla

Primer: prodaja automobila aukcijom

- Vlasnik nudi automobil na aukciji
- Potencijalno na hiljade ljudi može učestvovati u online aukciji
- Kupac mora prvo da se registruje
- Pogrešno definisana interakcija može dovesti do *deadlock* stanja

Primer: prodaja automobila aukcijom

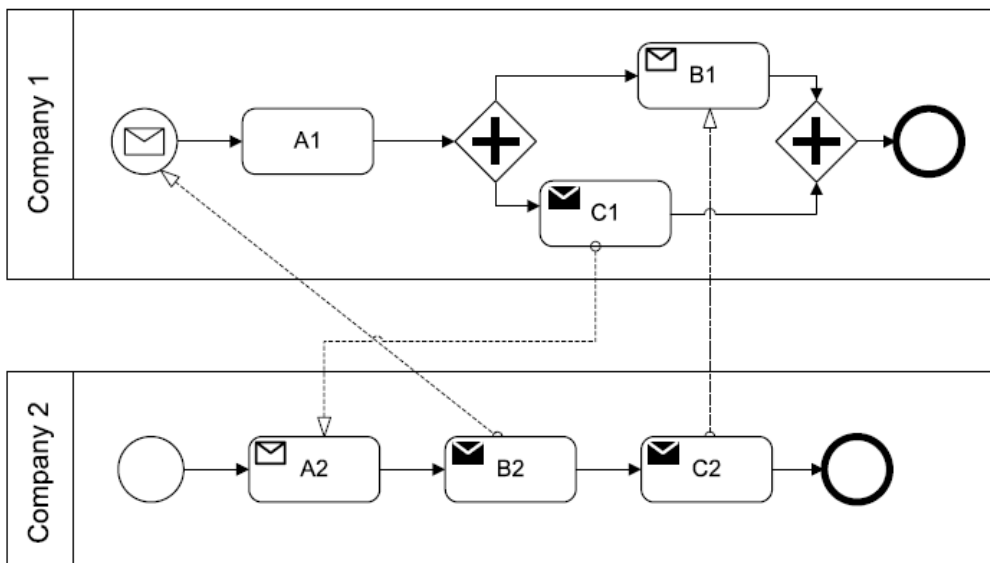


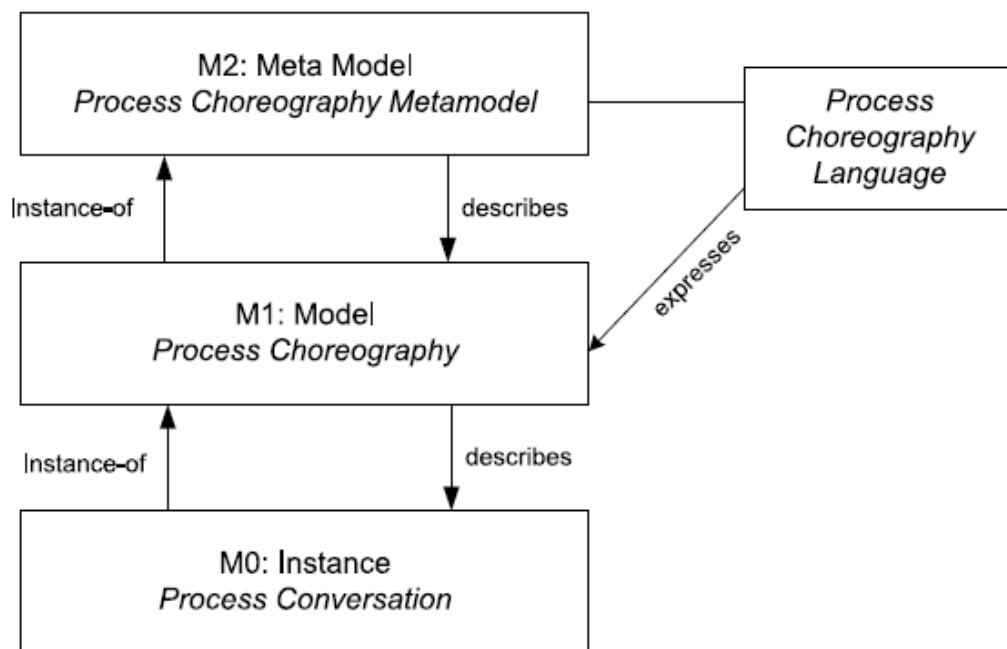
Fig. 5.1. Deadlock of interacting process orchestrations

- Ovakvom interakcijom obe orkestracije su “zaglavljene”
- Iako su formalno i jedna i druga orkestracija u redu (nema beskonačnih petlji, internih *deadlock-a*), njihova interakcija ih blokira
- Neophodno je da oba učesnika budu saglasna o koreografiji kako bi se ovakva stanja izbegla

Motivacija i terminologija

- Posmatranje izolovano samo jednog procesa nije dovoljno za korektno izvršavanje interakcije procesa
- Neophodno je modelovanje interakcija
- Za modelovanje koreografija takođe postoje već razmatrani nivoi apstrakcije i metamodeli

Motivacija i terminologija



Metamodel
koreografije procesa
definiše koncepte
pomoću kojih se kreira
model koreografije
procesa

Instanca modela
koreografije procesa je
Konverzacija procesa

Fig. 5.2. MOF levels of process choreographies

Konceptualni model koreografije procesa

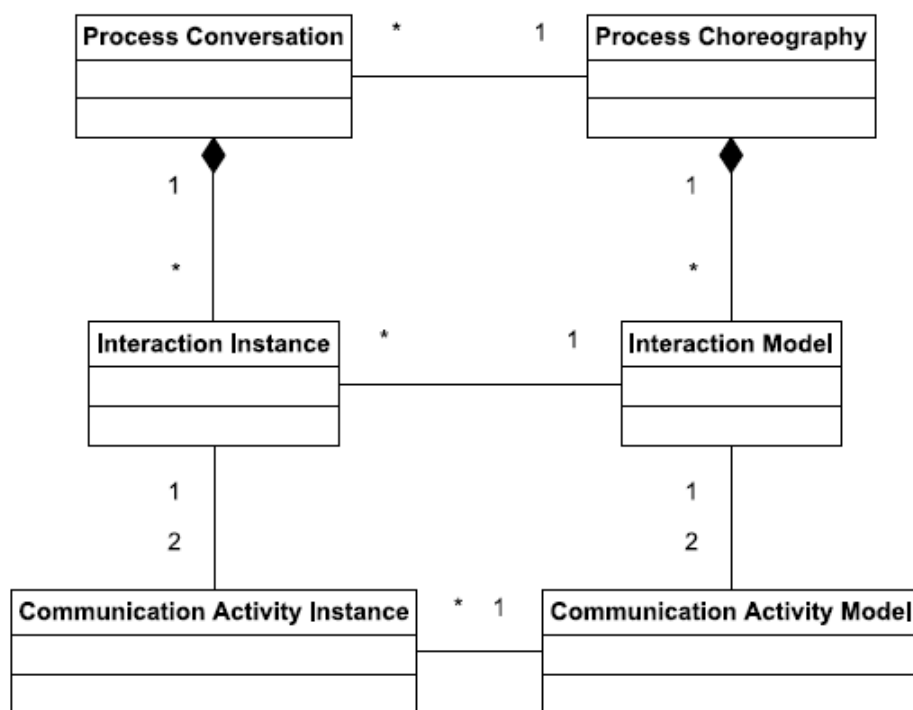


Fig. 5.3. Process choreography conceptual model

Koreografija procesa sastoji se od više **modela interakcije**. Svaki od njih povezan je sa 2 **modela komunikacionih aktivnosti**

Communication Activity Model predstavlja model aktivnosti u procesu koji ima komunikacione sposobnosti (slanje ili prijem poruka)

Svakom modelu prilikom izvršavanja koreografije odgovara određena instanca (konverzacija, instanca interakcije, instanca komunikacione aktivnosti)

Faze razvoja

- Dati su koncepti koji se razmatraju tokom razvoja - ne referentna metodologija
- Grubo - faze razvoja su organizovane u
 - Dizajn koreografije
 - Implementaciju koreografije
- U svaku od faza, na različite načine su uključeni različiti učesnici
 - Inženjeri poslovnog procesa
 - Sistem arhitekta
 - Programeri (*developeri*)

Faze razvoja (2)

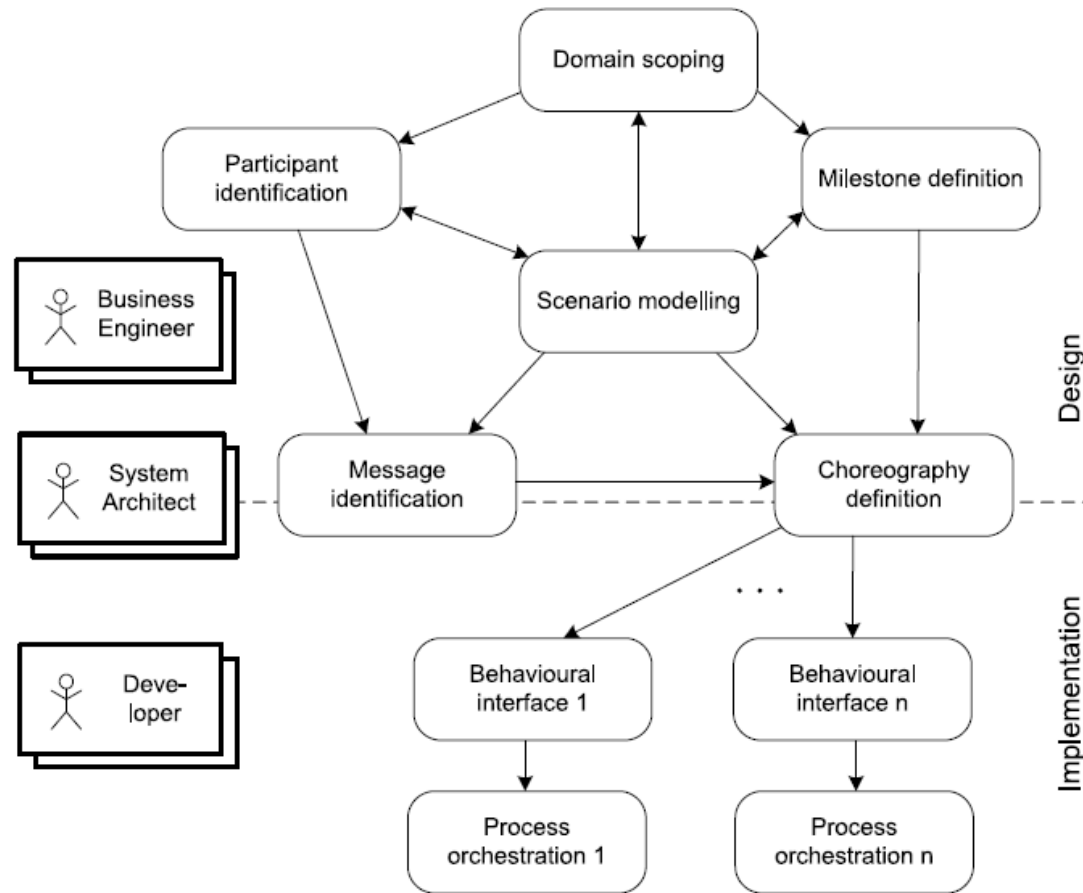


Fig. 5.4. Phases during choreography design and implementation

Faze razvoja – učesnici (3)

- Inženjeri poslovnog procesa
 - Odgovorni za poslovne aspekte saradnje, vode računa da modelovana saradnja doprinosi poslovanju
- Sistem arhitekta
 - Odgovorni za arhitekturu sistema na kome će se implementirati predloženi model koreografije
 - Učestvuju i u dizajnu i u implementaciji
- Programeri (*developeri*)
 - Zaduženi za implementaciju modela koreografije na predloženoj arhitekturi sistema

Faze razvoja (4)

■ Identifikacija učesnika

- Služi da se definišu uloge učesnika (organizacija) u koreografiji

■ Definicija kritičnih tačaka (*milestones*)

- Definišu se stanja koreografije u kojima su saradnjom učesnika postignuti određeni rezultati
- Kritične tačke i njihov redosled definišu aspekte ponašanja koreografije (*behavioural aspect*)

Faze razvoja (5)

■ Identifikacija poruka

- ☐ U ovoj fazi uočavaju se i definišu poruke koje se razmenjuju u određenim interakcijama
- ☐ U ovoj fazi neophodno je uzeti u obzir i poslovne (sadržaj poruka) i tehničke aspekte (format poruka, metode prenosa)

Faze razvoja (6)

■ Definicija koreografije

- U ovoj fazi se kombinuju se podaci iz prethodne dve faze
- I ovde se uzimaju u obzir i poslovni i tehnički aspekti (npr. format poruka)
- Formati poruka, kao i značenje moraju već biti dogovoreni
- Domenski specifični standardi obezbeđuju zajedničku i razumljivu terminologiju
 - Dodatno se poboljšavaju tehničkim specifikacijama koje definišu i strukture podataka i formate poruka koje se razmenjuju

Faze razvoja (7)

■ *Behavioural Interfaces*

- Definišu intefejse za svaku od uloga koja učestvuje u koreografiji procesa
- Služe kao osnova za orkestraciju procesa koje određeni učesnik realizuje

Dizajn koreografije procesa

- Više aktivnosti u kojima se definišu različiti artefakti neophodni za koreografiju
 - Dizajn strukture na visokom nivou (apstrakcije)
 - Identifikuju se uloge učesnika i struktura komunikacije – sprovodi se kroz fazu *identifikacije učesnika*
 - Dizajn ponašanja na visokom nivou (apstrakcije)
 - Definišu se kritične tačke i redosled kojim se one dešavaju – sprovodi se kroz fazu *definicije kritičnih tačaka*

Dizajn koreografije procesa (2)

□ Scenarija saradnje

- Koreografije visokog nivoa se poboljšavaju uvođenjem scenarija saradnje koji povezuju dostizanje određene kritične tačke sa ostvarenim komunikacijama između učesnika – sprovodi se kroz fazu *definicije koreografije*, na osnovu neformalne specifikacije scenarija dobijene kroz fazu *modelovanja scenarija*

□ Behavioural Interfaces (interfejsi ponašanja)

- Na osnovu scenarija saradnje izvodi se interfejs ponašanja svakog od učesnika

Dizajn koreografije procesa na visokom nivou apstrakcije

- Strukturni dizajn – utvrđuje uloge učesnika
- U našem primeru:

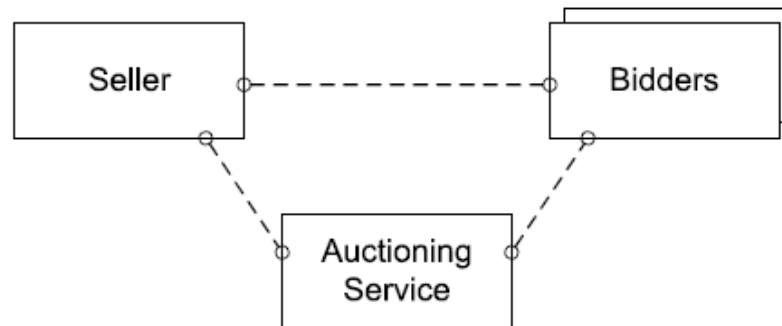
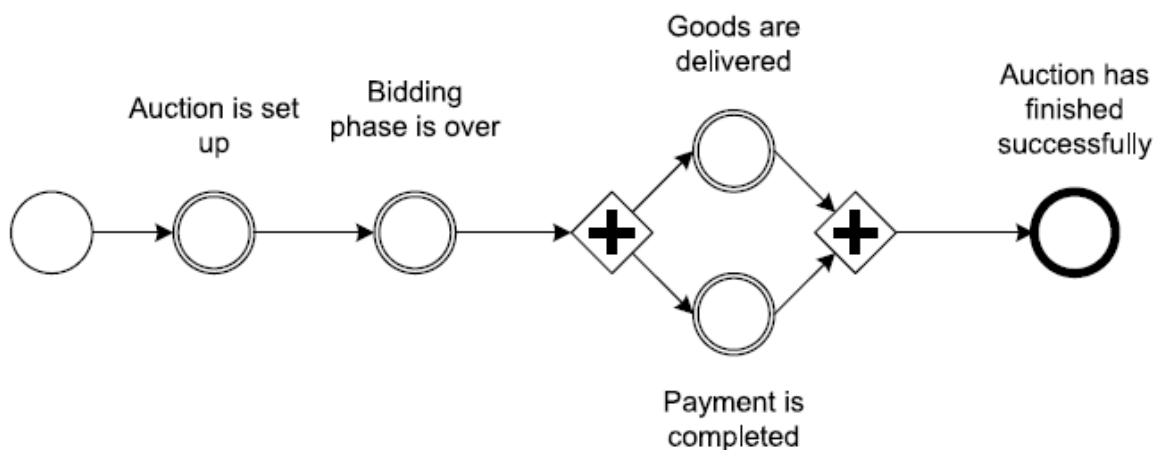


Fig. 5.5. High-level structural model of participants in bidding scenario

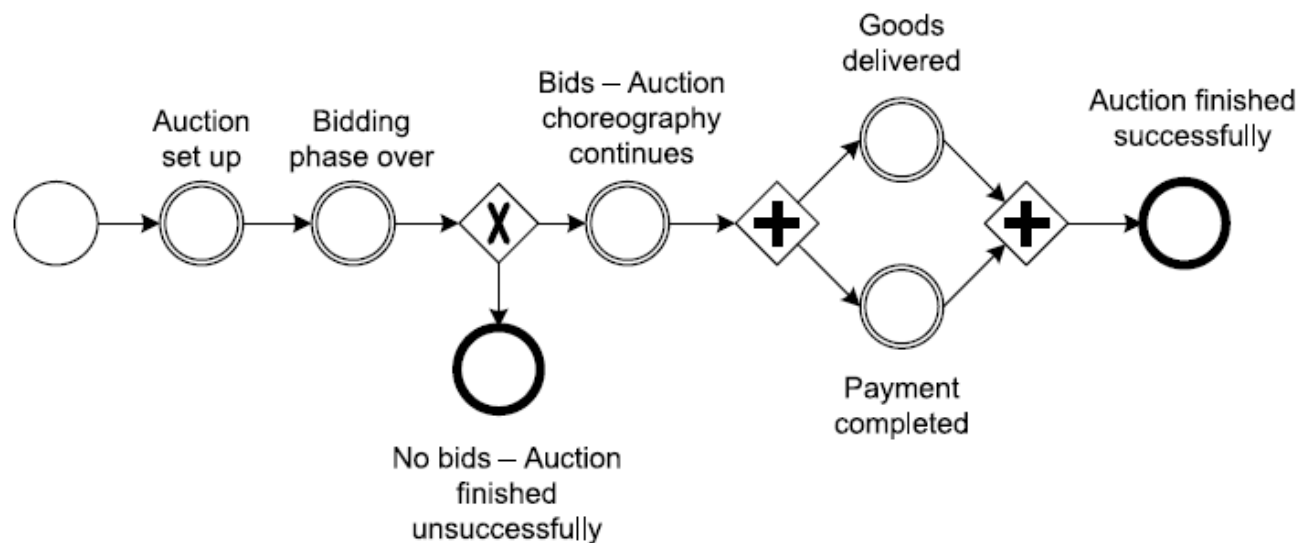
Dizajn koreografije procesa na visokom nivou apstrakcije (2)

- Dizajn ponašanja sistema – utvrđuje kritične tačke (*milestones*)
 - Može se predstaviti BPMN notacijom, gde su *mileston*-i predstavljeni kao događaji
- U našem primeru:



Dizajn koreografije procesa na visokom nivou apstrakcije (3)

- Ukoliko se neki *milestone* ne dostigne, moguće je i da cela koreografija ne obavi zadatak zbog koga je kreirana
- U našem primeru:



Scenario saradnje

- Nakon što su određene kritične tačke u koreografiji, mogu se razmatrati scenariji saradnje
 - Specificira se interakcija koja je neophodna između učesnika kako bi koreografija napredovala od jedne do druge kritične tačke
 - Interkacije koje sedešavaju između dve kritične tačke mogu biti opisane jednim ili više scenarija saradnje
 - Pojedinačni scenario treba da je jednostavan, jer je lakše postići daglasnost na nekompleksnim scenarijima

Scenario saradnje (2)

- U našem primeru – scenario saradnje pri postavljanju aukcije
 - Ključna tačka koja je terminalna za ovaj scenario je samo prelazna kritična tačka u kompletnoj koreografiji

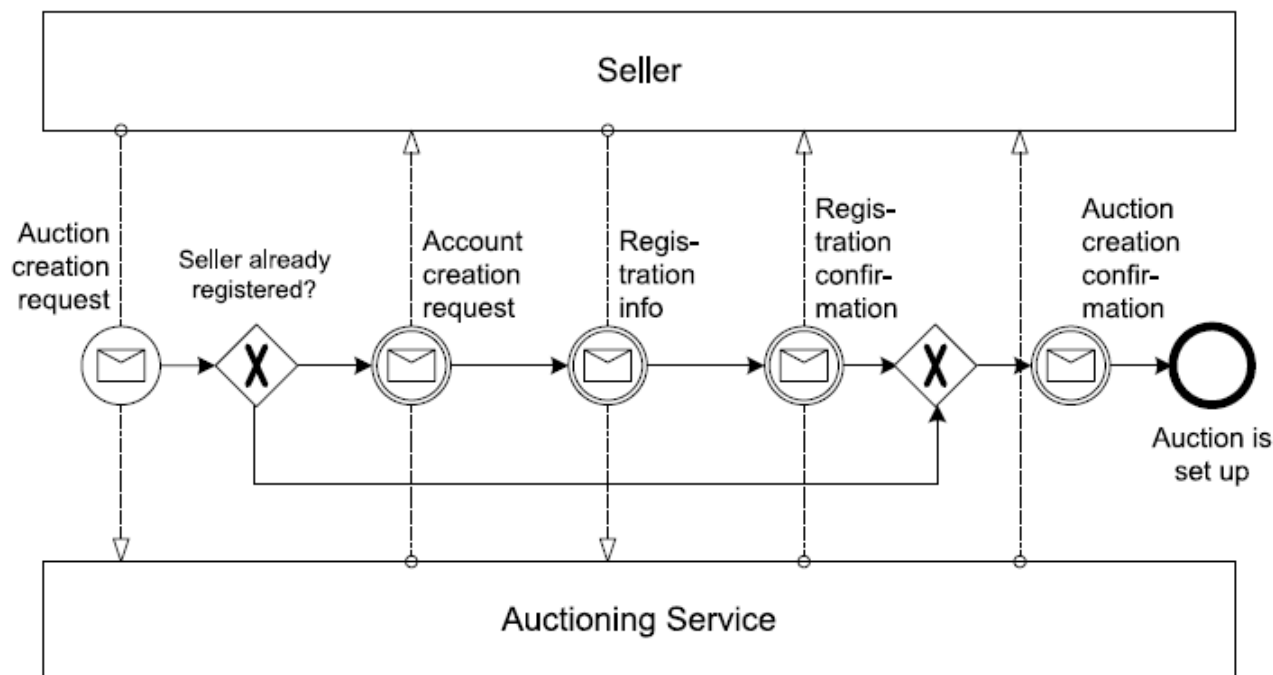


Fig. 5.8. Collaboration scenario: reaching milestones through interactions

Scenario saradnje (3)

- Model interakcije između učesnika se može (kao u primeru) predstaviti posebnim procesom koji se sastoji od modela interakcije umesto modela aktivnosti
- Na ovom nivou nedostaju tehnički detalji koji definišu strukturu i sadržaj pojedinačnih poruka

Scenario saradnje (4)

- Prethodni scenario opisuje interakcije koje nastaju, ali ne pojašnjava kako to pojedinačni učesnik “vidi” u orkestraciji svog procesa
- Interfejsi ponašanja – definišu pogled jednog učesnika na koreografiju
 - Definišu delove orkestracije procesa koje imaju spolja vidljivo ponašanje

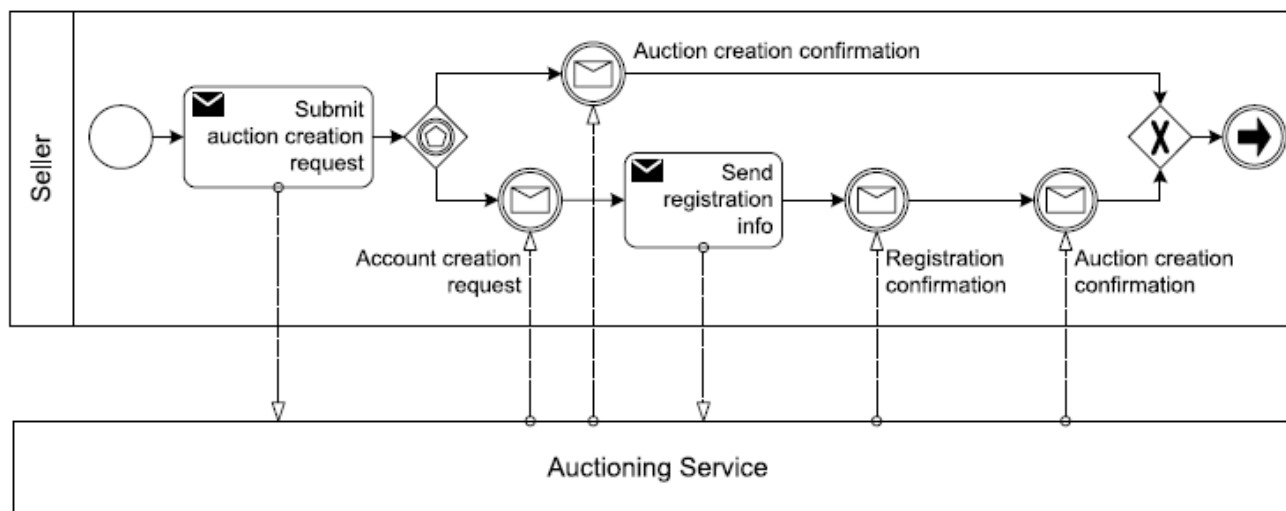


Fig. 5.9. Behavioural interface for seller

Kompatibilnost

- Sposobnost skupa učesnika da uspešno sarađuju u skladu sa datom koreografijom procesa
- Mogući uzroci nekompatibilnosti
 - Različiti formati poruka
 - Pogrešne i neusklađene interakcije
 - Npr. proces jednog učesnika očekuje notifikaciju pre nego što se proces nastavi, a nijedan proces drugih učesnika takvu notifikaciju ne proizvodi
 - Kompatibilnost upravo teži da reši ove probleme
- Strukturna kompatibilnost
- Kompatibilnost ponašanja

Strukturna kompatibilnost

- Kada se posmatra samo struktura komunikacije govori se o strukturnoj kompatibilnosti
 - Jaka strukturna kompatibilnost
 - Za svaku poruku koja može biti poslana, postoji učesnik koji će je primiti, i za svaku poruku koja može biti primljena postoji korisnik koji će je poslati
 - Slaba strukturna kompatibilnost
 - Za sve poruke koje mogu biti poslane može postojati učesnik koji će ih primiti. Za svaku poruku koja može biti primljena ne mora postojati i učesnik koji će je stvarno poslati
 - Pošto su orkestracije najčešće nastajale samostalno ovakav vid kompatibilnosti je više verovatan
 - Pretpostavlja se da interakcija može da se obavi iako učesnik može da primi i dodatne poruke. Pretpostavka je da te poruke nisu kritične za njegovu orkestraciju procesa.

Strukturna kompatibilnost (2)

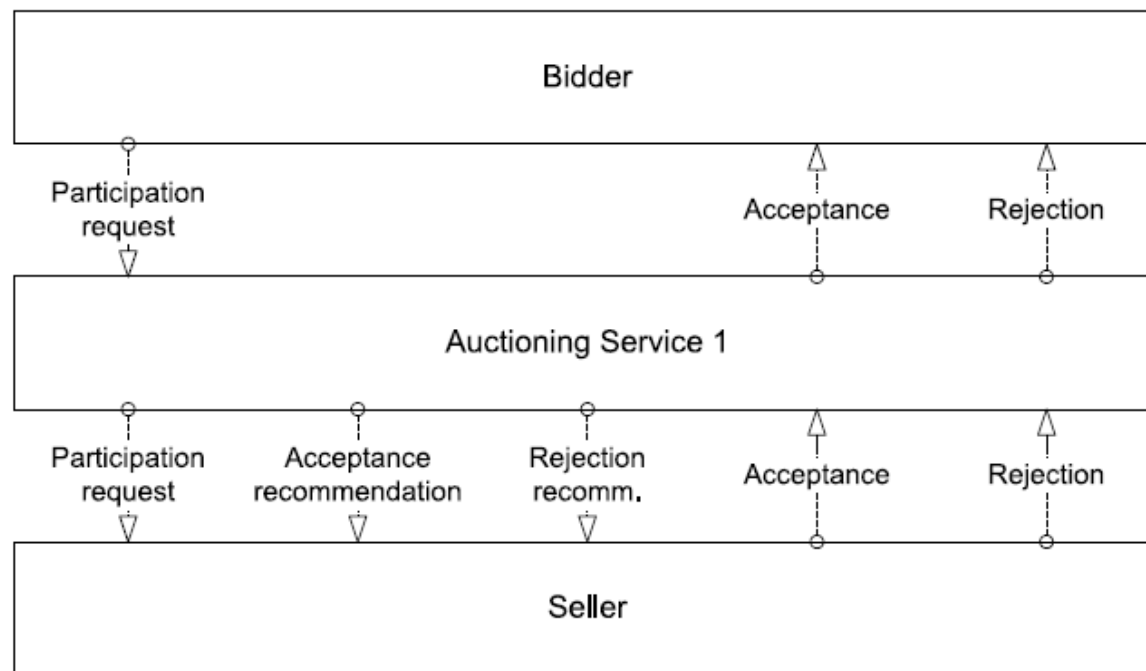


Fig. 5.10. Interactions of participants in auctioning scenario

Jaka strukturna kompatibilnost

Strukturna kompatibilnost (3)

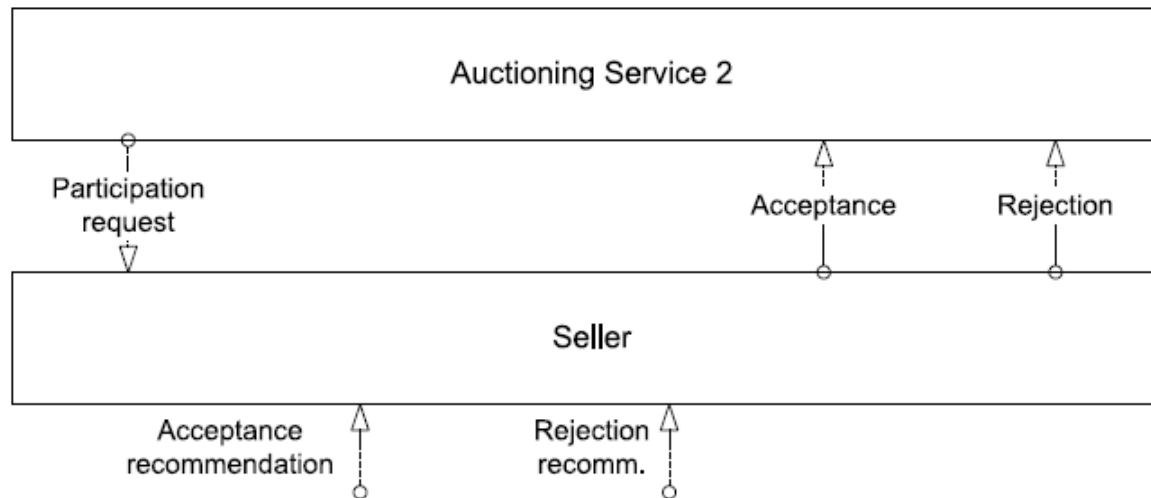


Fig. 5.11. Alternative auctioning service results in weak structural compatibility

Slaba strukturna kompatibilnost – izostavljen *bidder*

Kompatibilnost ponašanja

- Kompatibilnost ponašanja uzima u obzir kontrolu toka između instanci procesa u konverzaciji
- Ovde se uzima u obzir i sama orkestracija procesa učesnika u konverzaciji
- Neophodna je formalna, nedvosmislena reprezentacija
- *Martens* je preznetovao pristup za formalnu kontrolu komaptibilnosti ponašanja
 - Orkestracije procesa se predstavljaju kao posebna klasa Petri mreža – moduli radnih procesa (*workflow modules*)
 - *Wokflow modules* su u osnovi *workflow nets* sa specifičnim komunikacionim mestima koja reprezentuju tok poruka među učesnicima

Workflow module

- Kada učesnik šalje poruku, orkestracija njegovog procesa sadrži tranziciju sa izlaznim komunikacionim mestom koje može sadržati poslanu poruku.
- Na prijemnoj strani postoji odgovarajuće prijemno komunikaciono mesto
- Svaka orkestracija procesa se predstavlja *workflow modulom* koji definiše njeno interno ponašanje, kao i spolja vidljivo komuniciranje

Workflow module (2)

■ Definicija:

Petri mreža $PN = (P, T, F)$ predstavlja *workflow module* ako i samo ako su zadovoljeni sledeći uslovi:

- Skup mesta P particionisan je na skup P^N internih mesta, P^I skup dolaznih mesta i P^O skup odlaznih mesta
- T je neprazan skup tranzicija
- Relacija F se particioniše na relacije internog toka $F^N \subseteq (P^N \times T) \cup (T \times P^N)$, i relacije komunikacionih tokova $F^C \subseteq (P^I \times T) \cup (T \times P^O)$
- (P^N, T, F^N) je *workflow net*
- Ne postoji tranzicija t koja je spojena i na izlazno i na ulazno mesto

Workflow module (3)

- Delimični prikaz orkestracija - primer aukcije:
 - Workflow module NIJE workflow net, jer postoje druga mesta (osim start i end) koja imaju samo ulazne ili samo izlazne grane
 - Tranzicije u kojima se šalje poruka označavamo sa !
 - Tranzicije u kojima se primaju poruke označavamo sa ?

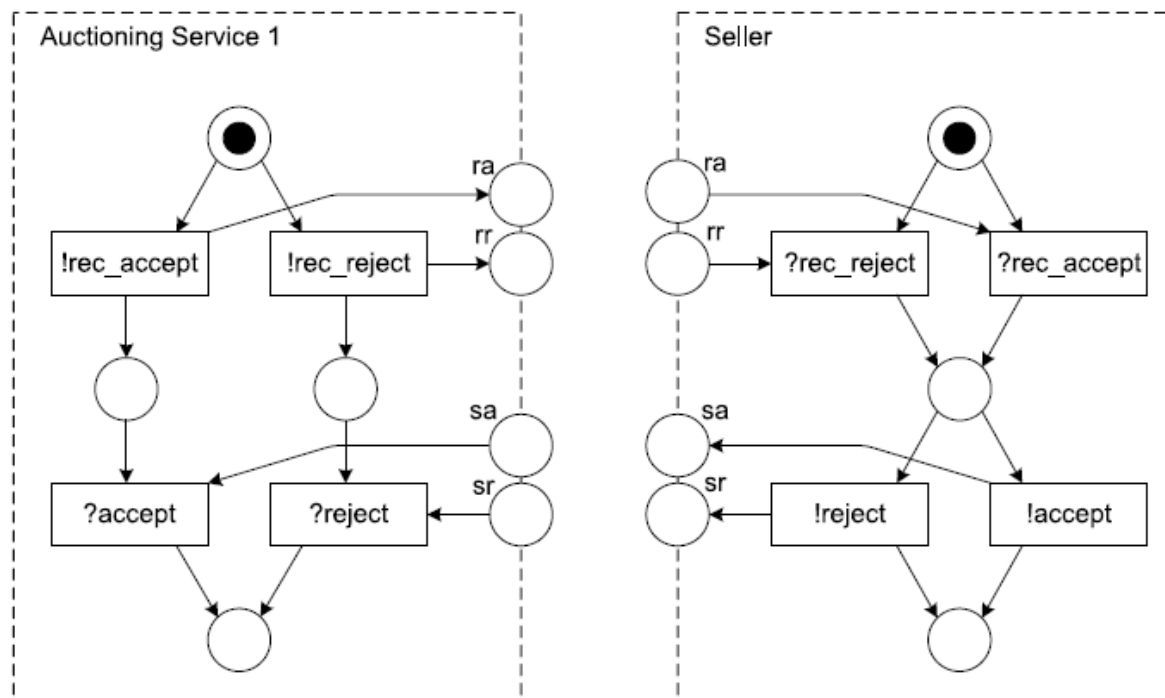


Fig. 5.12. Workflow modules as basis for checking compatibility

Workflow module (4)

- Ovakav model zahteva jaku strukturnu kompatibilnost
- Kako za svako odlazno mora postojati korespondirajuće dolazno mesto, kada se moduli spoje dobija se *workflow net*

Workflow module (5)

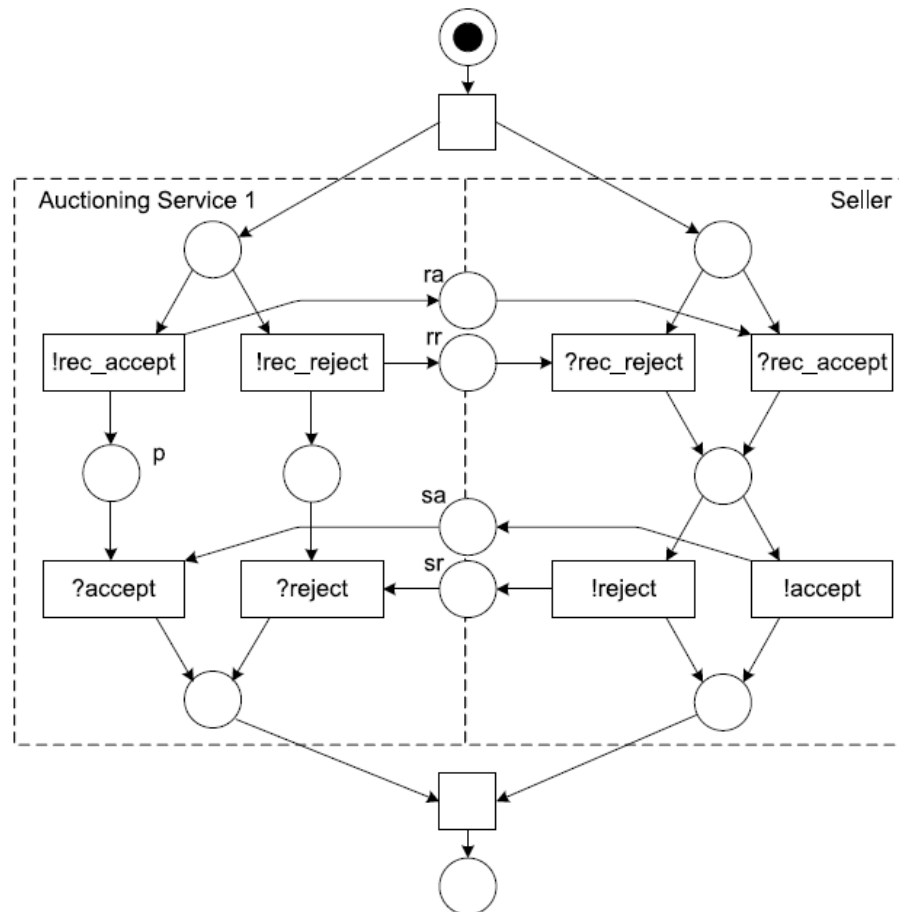


Fig. 5.13. Workflow net as composition of workflow modules

Workflow module (6)

- Iako ovo jeste *workflow net*, kompozicija modula je nezadovoljavajuća
- Ukoliko je poslana preporuka za prihvatanje, a prodavac odluči da odbije ponuđenog kupca, ova Petri mreža bi završila u *deadlock-u* (jedan token u p, drugi u sr)
- Korigovani model je prikazan na sledećoj slici

Workflow module (7)

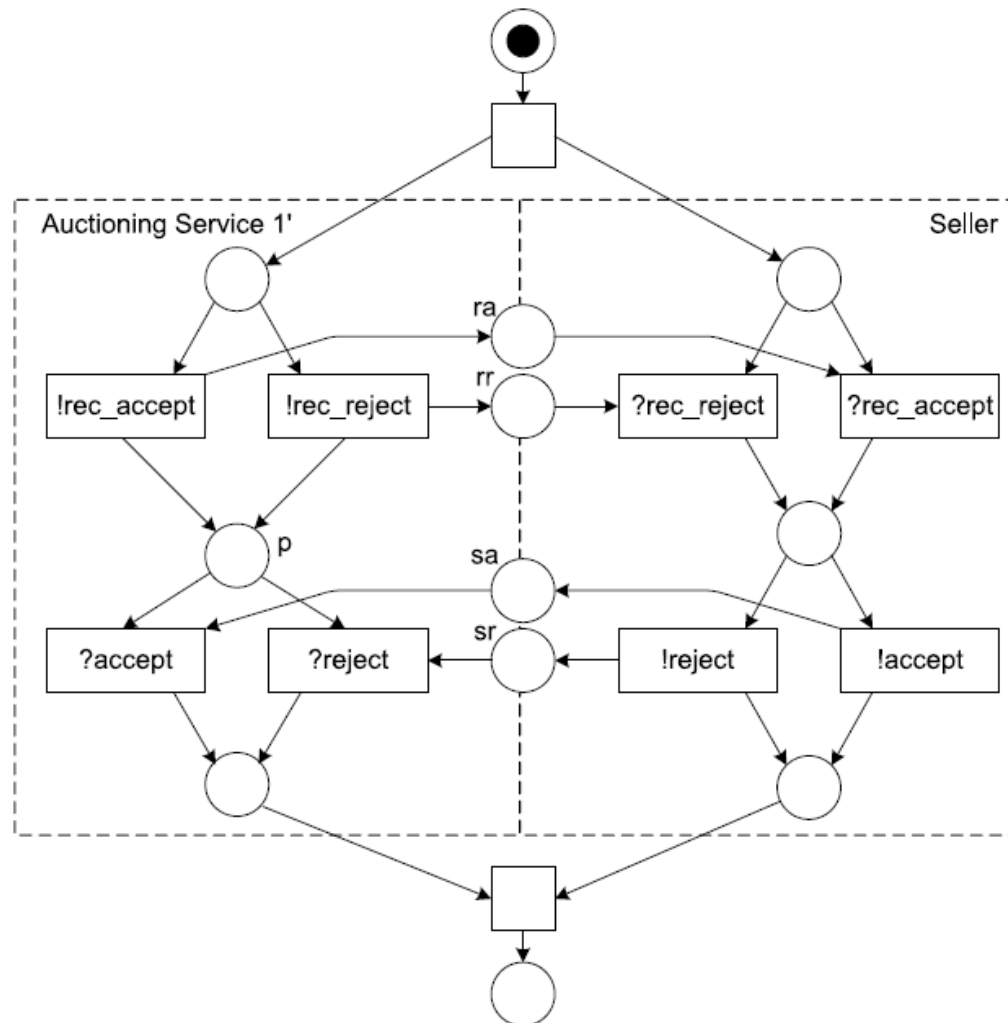


Fig. 5.14. Workflow modules that are compatible

Implementacija koreografije procesa

- Interfejsi ponašanja predstavljaju osnov za kreiranje internih orkestracija procesa, jer svaka orkestracija mora obezbediti spolja vidljivo ponašanje koje je prethodno definisano
- Ovo ponašanje sada mora biti inkorporirano u orkestraciju internog procesa
- Čak ako se interni proces naknadno menja (aktivnosti se dodaju ili preslažu), spolja vidljivo ponašanje definisano interfejsom ponašanja mora ostati očuvano

Implementacija koreografije procesa (2)

- Koristimo primer invertovane aukcije (tender), gde kupac (npr. fabrika automobila) zahteva ponude (za delove)
- Prodavci moraju da se registruju
- Kupac je upravljanje aukcijom (tenderom) prepustio posebnoj službi
- Za odabranog ponuđača bira se i prevoznik koji će dopremiti robu
- Interfejsi između korisnika moraju biti kompatibilni

Implementacija koreografije procesa (3)

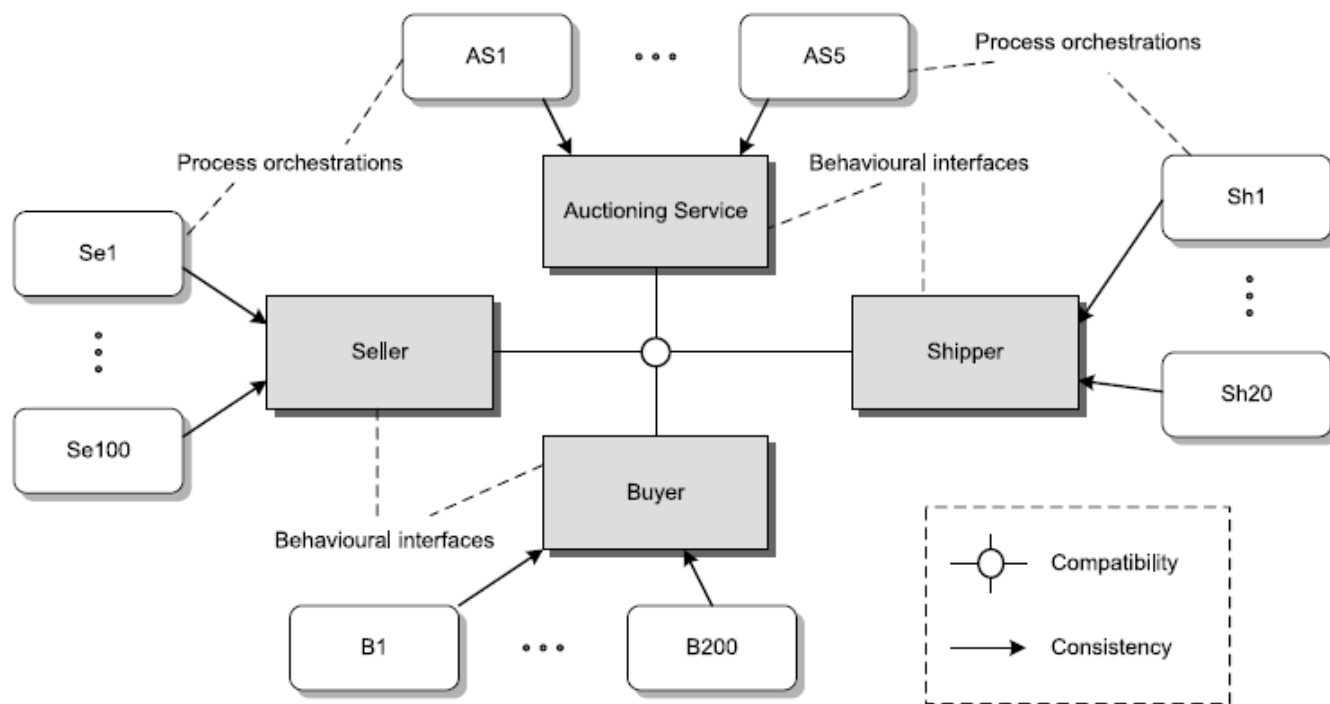


Fig. 5.16. Participant roles with compatibility and consistency relations in a reverse auctioning scenario

Implementacija koreografije procesa (4)

- U svakoj od uloga može se potencijalno pojaviti više različitih organizacija sa sopstvenim orkestracijama procesa
- Orkestracije moraju biti u skladu sa očekivanim interfejsom ponašanja za datu ulogu
- Korišćenjem pravila konzistentnosti svaki učesnik može lokalno proveriti da li je njegova orkestracija procesa u skladu sa zadatim interfejsom ponašanja

Implementacija koreografije procesa (4)

- Ukoliko su interfejsi ponašanja svih uloga međusobno kompatibilni i ako su, za svakog učesnika, interne orkestracije procesa u skladu sa zadatim interfejsom ponašanja za datu ulogu, onda će realizovana saradnja biti uspešna

Implementacija koreografije procesa (5)

- Alternativne implementacije procesa kupca

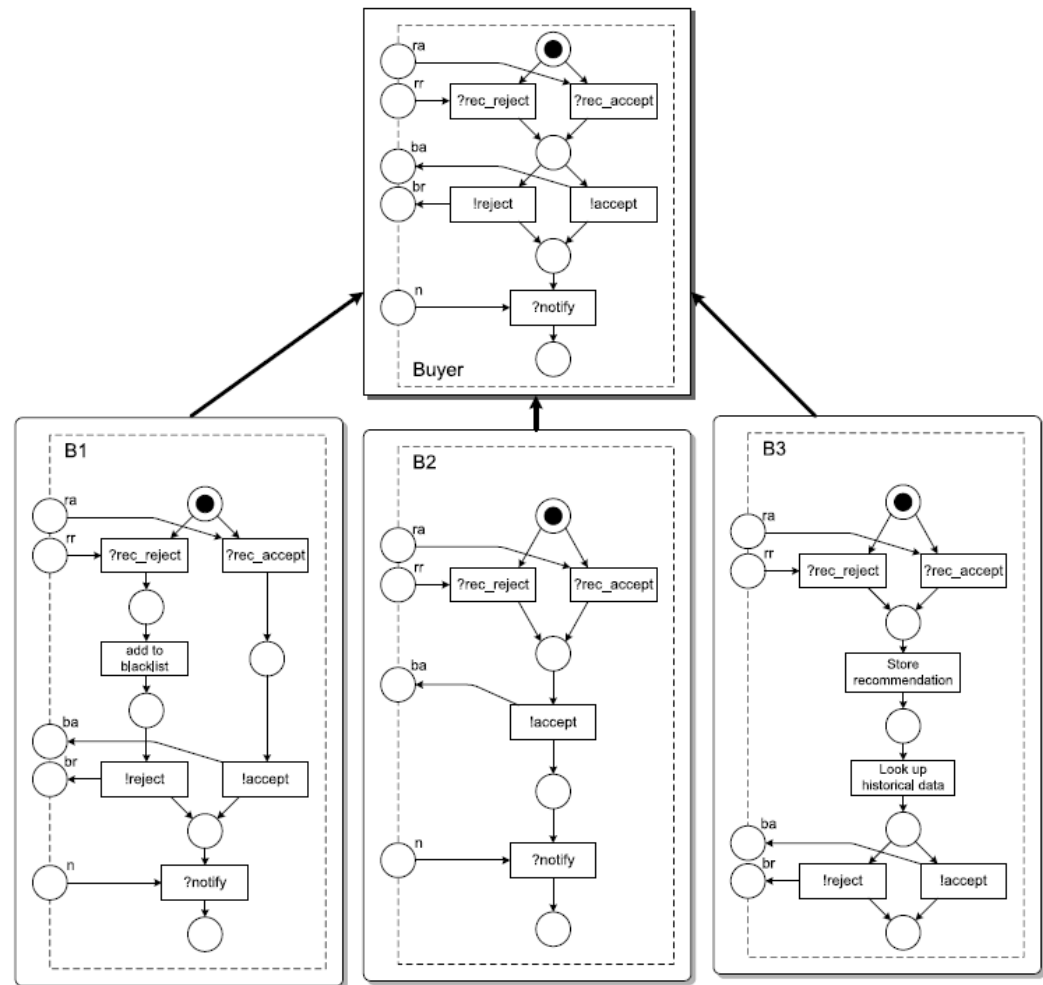


Fig. 5.17. Alternative implementations for buyer role

Kriterijum konzistentnosti

Pristup javno-privatno

- Baziran na Petri mrežama
- Koristi koncept nasleđivanja ponašanja
- Partneri se prvo dogovore o koreografiji – definisanoj pomoću Petri mreže
- Mreža se particioniše – definišući odgovornosti svakog partnera
 - Particionisanje definiše i interfejs ponašanja tj. javno spolja vidljivo ponašanje procesa posmatranog partnera
 - Rafinisanje javno vidljivog procesa kako bi se razvila interna orkestracija procesa izvodi se skupom transformacija – graf operacije nad Petri mrežom

Kriterijum konzistentnosti transformacije Petri mreže

- Petlja: Proces se može transformisati dodavanjem petlje za koje su početno i krajnje mesto isto mesto u javno vidljivom procesu
- Obilazak: Grana Petri mreže može biti zamenjena podmrežom, koja definiše obilazak originalnog toka definisanog u javno vidljivom procesu
- Konkurentna grana: može biti dodata tako da se razvija iz originalnog toka i naknadno sinhronizuje ponovo u originalni tok

Kriterijum konzistentnosti transformacije Petri mreže (2)

- Dokazano je da primenom ovih transformacija ne dolazi do promene spolja vidljivog ponašanja procesa
- Provatni procesi kreirani na ovaj način predstavljaju samo specijalizaciju javnog vidljivog procesa
- Takođe je dokazano da kombinovanje ovako nastalih privatnih procesa rezultuje korektnom koreografijom
- Privatni procesi koji mogu biti izvedeni iz javnog procesa primenom samo ovih transformacija su konzistentni sa javnim procesom
- Po ovim kriterijumima ni B1 ni B2 ni B3 nisu konzistentni sa javnim procesom

Kriterijum konzistentnosti transformacije Petri mreže - petlja

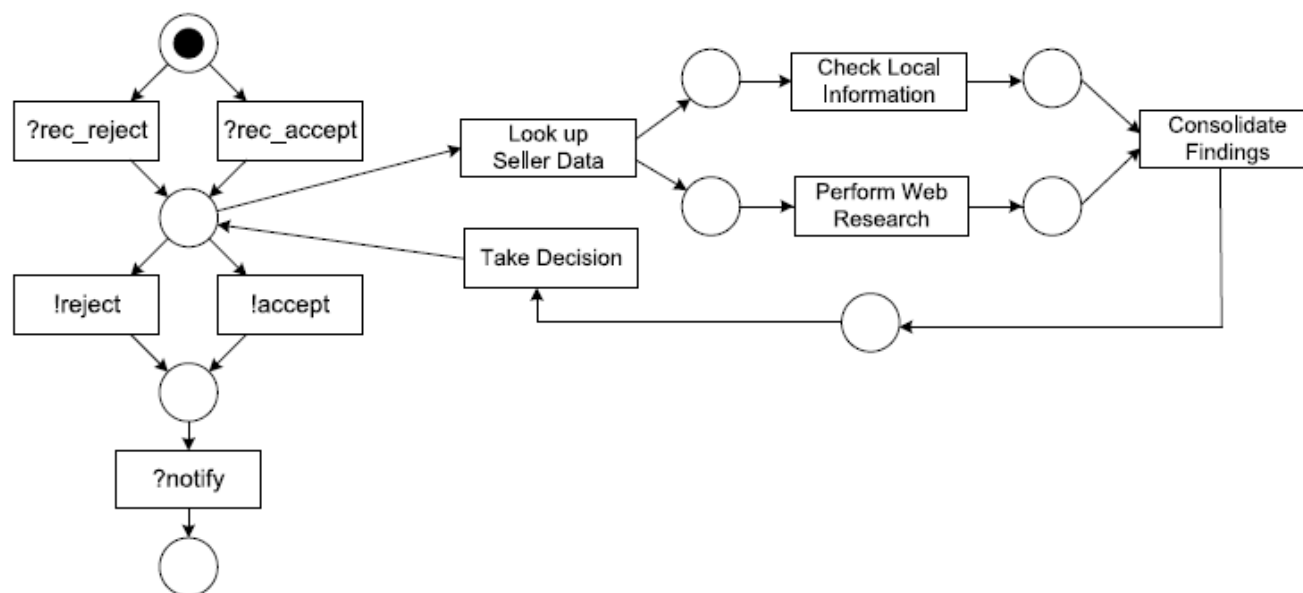


Fig. 5.18. *Loop* transformation operation of the public-to-private approach

Kriterijum konzistentnosti transformacije Petri mreže - obilazak

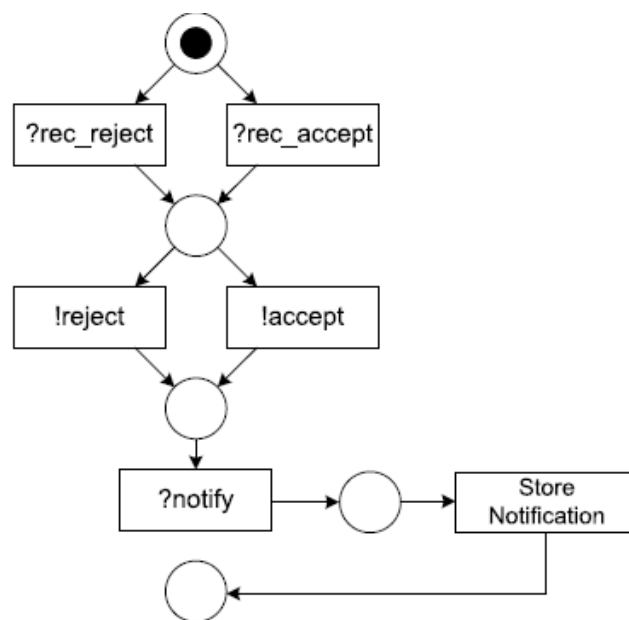


Fig. 5.19. *Detour* transformation operation of the public-to-private approach

Kriterijum konzistentnosti transformacije Petri mreže – konkurentna grana

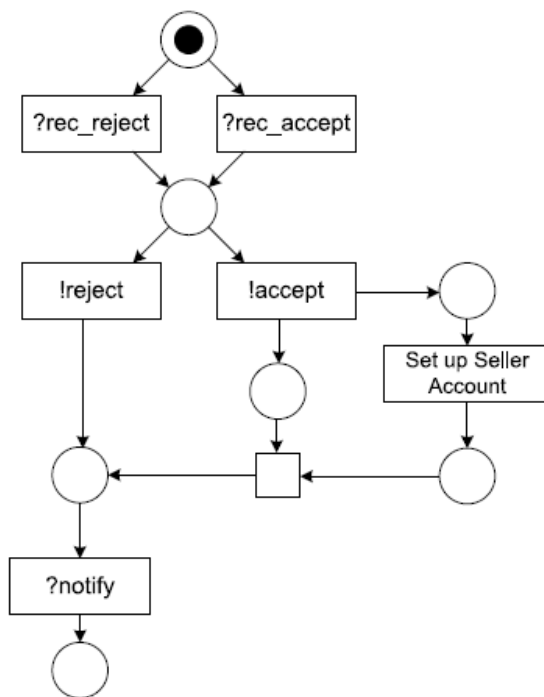


Fig. 5.20. *Concur* transformation operation of the public-to-private approach

Transformacije Petri mreže - primer

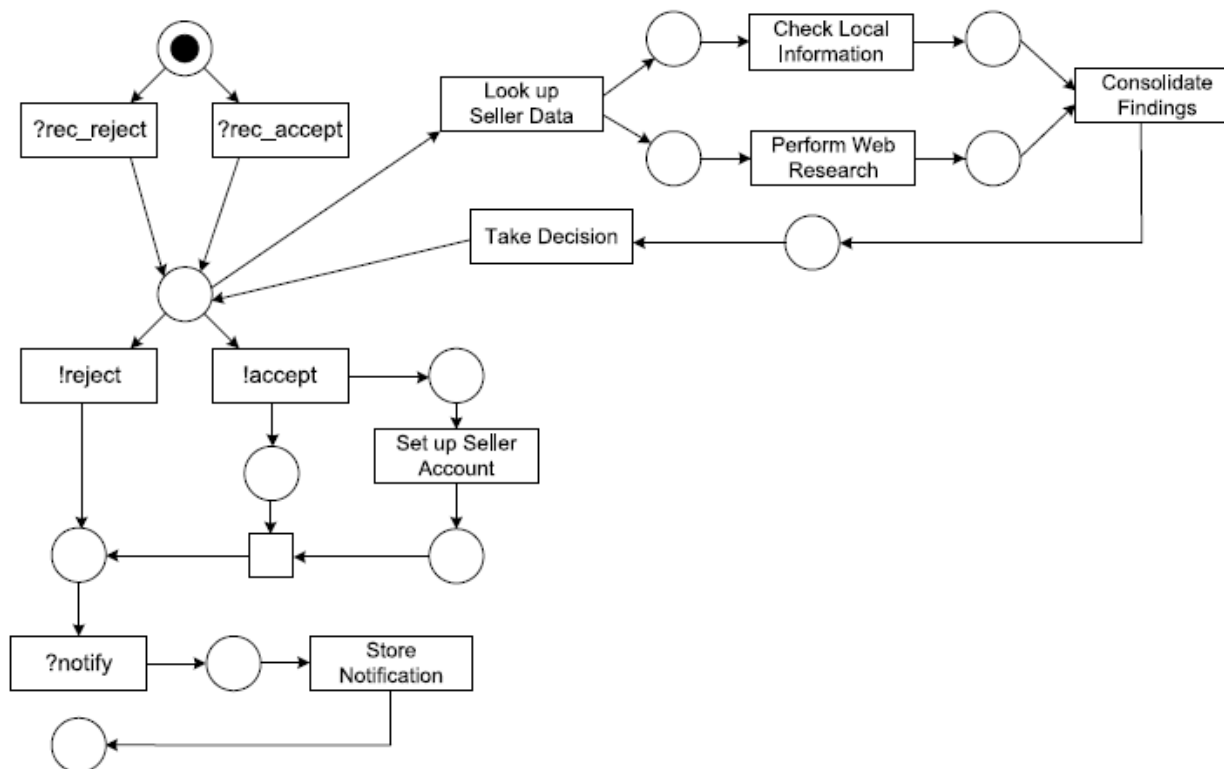


Fig. 5.21. Private process that—according to the public-to-private approach—is consistent with the public process of the buyer

Šabloni interakcije

- Iako su šabloni za kontrolu toka već pojašnjeni koreografije su specifične, zasnovane na tokovima poruka
- Šabloni interakcije servisa
 - predlažu interakcije fine granularnosti koje se mogu kombinovati u koreografije
 - Mogu se koristiti i kao sredstvo za procenu koliko dobro određeni jezik podržava napredne konverzacione modele

Šabloni interakcije (2)

■ Mogu se klasifikovati na osnovu:

□ Broja učesnika

- Bilateralni i multilateralni

□ Broju razmenjenih poruka

- Interakcije sa jednim ili više transmisija poruka

□ Varijacija u primaocu poruke

- Kod dvosmerne komunikacije

- *round trip* poruka – pošiljalac i prijemnik poruke su uvek isti
- *Routed interaction* – primalac poruke je u opštem slučaju različit od pošiljaoca

Šabloni interakcije - *Send*

- Predstavlja jednosmernu interakciju između dva učesnika gledano iz perspektive pošiljaoca
 - Primalac može biti postavljen tokom modelovanja
 - Primalac može biti postavljen tokom izvršavanja procesa

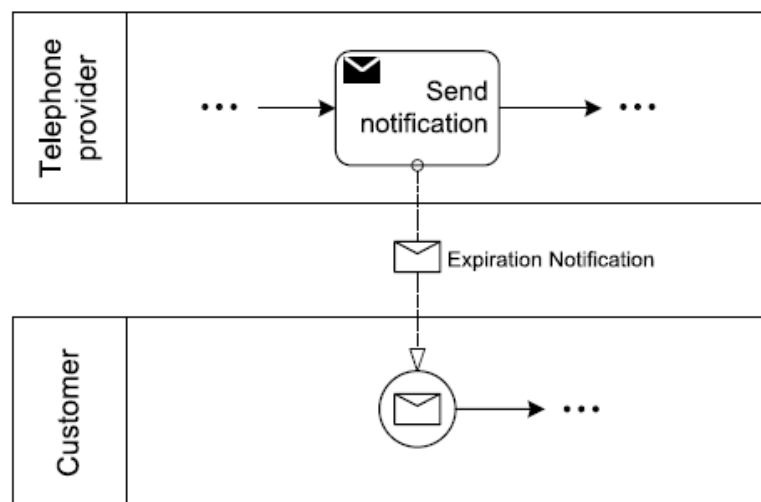


Fig. 5.22. Send pattern

Šabloni interakcije - *Receive*

- Predstavlja jednosmernu interakciju između dva učesnika gledano iz perspektive primaoca
 - Sa stanovišta baferovanja poruka moguće je
 - Odbaciti poruke koje nisu očekivane
 - Sačuvati ih i naknadno obraditi

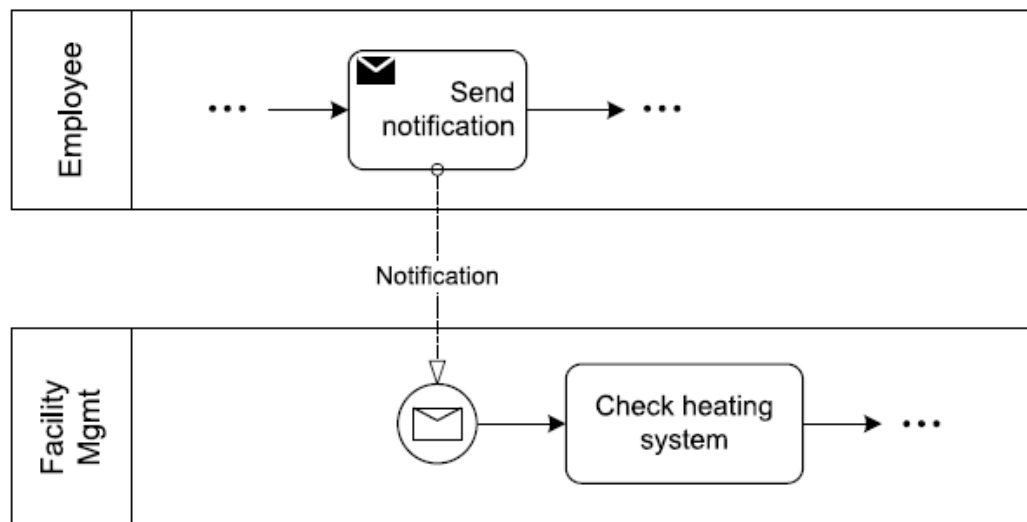


Fig. 5.23. Receive pattern

Šabloni interakcije – *Send/Receive*

- Jedan učesnik šalje poruku drugom učesniku, koji zatim na nju odgovara
 - Obe poruke pripadaju istoj konverzaciji
 - Pošto može biti više poruka, mora se znati koje poruke čine par

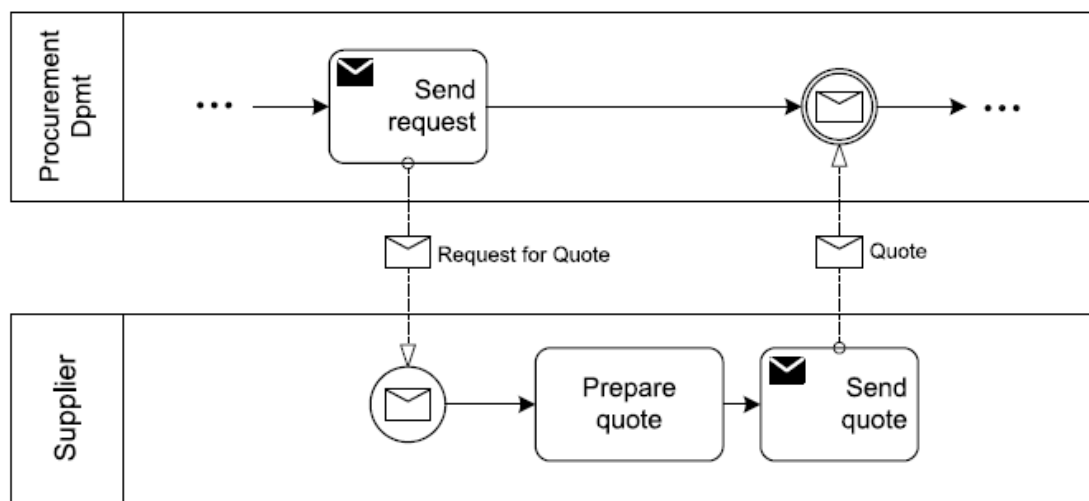


Fig. 5.24. Send/receive pattern

Šabloni interakcije – *Racing Incoming Messages*

- Jedan učesnik čeka na prijem određene poruke, ali i ostali učesnici imaju mogućnost slanja poruke
- Poruke od različitih učesnika se utrkuju
- Samo prva primljena će biti i obrađena

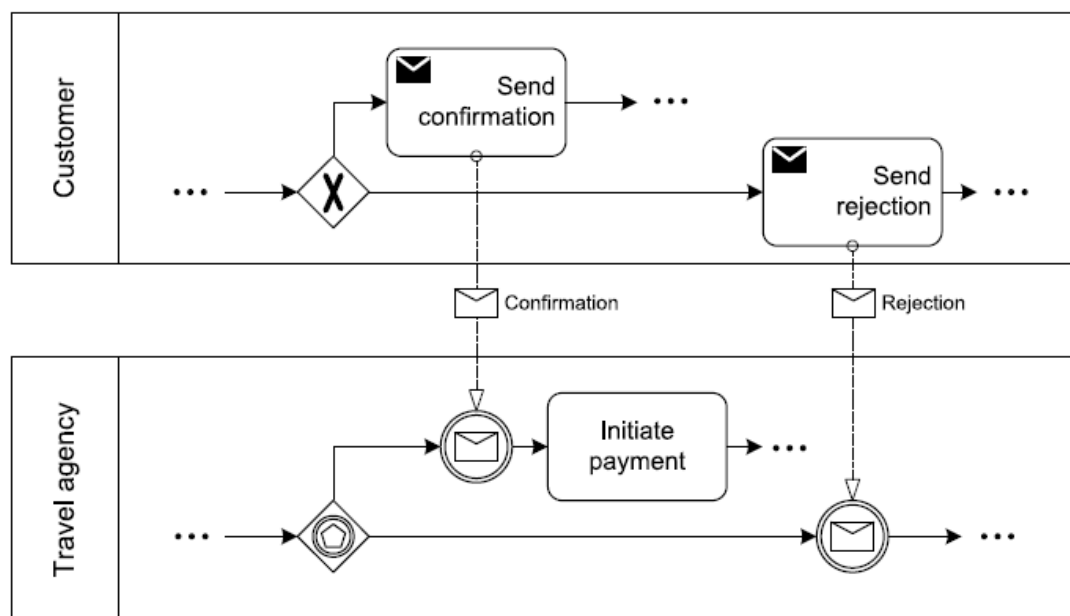


Fig. 5.25. Racing incoming messages pattern

Šabloni interakcije – *One-to-many send*

- Jedan učesnik šalje više poruka odjednom različitim korisnicima
- Lista primalaca može biti poznata za vreme dizajna ili formirana tokom izvršavanja

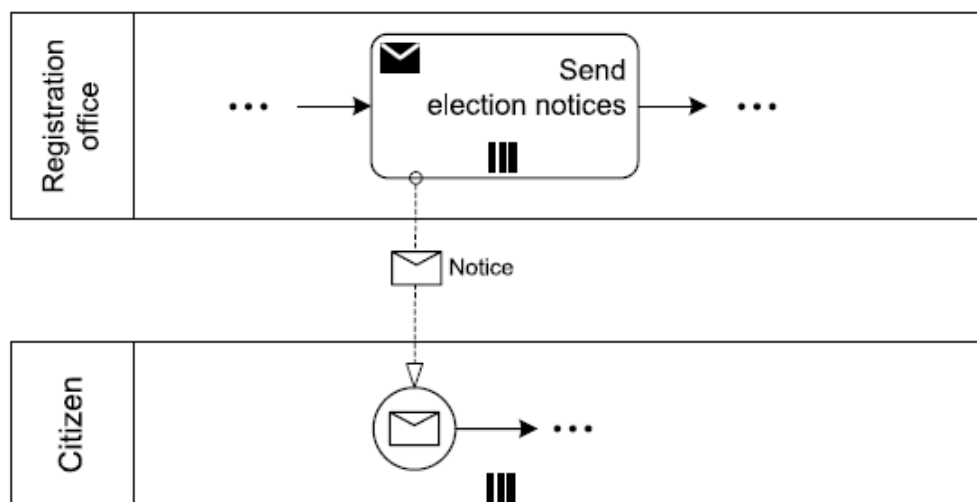


Fig. 5.26. One-to-many send pattern

Šabloni interakcije – *One-from-many receive*

- Poruke mogu biti primljene od raznih pošiljalaca
- Primalac čeka poruke, a svaki pošiljalac može poslati tačno jednu
- Tipično primalac ne zna ukupan broj poruka i čeka dok ne pristigne neki određeni broj

Šabloni interakcije – *One-from-many receive*

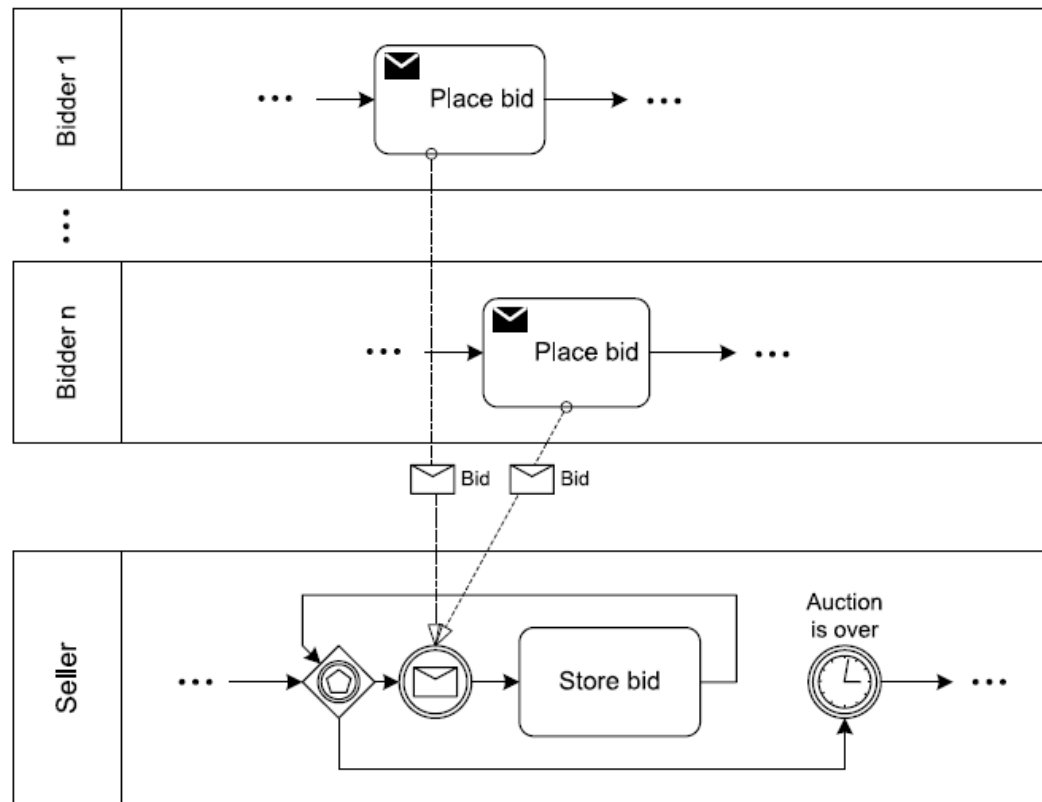


Fig. 5.27. One-from-many receive pattern

Šabloni interakcije – *One-to-many send/receive*

- Učesnik pošalje nekoliko poruka različitim korisnicima i čela na njihov odgovor
- Tipično se ne čeka da pristignu svi odgovori već se čekanje prekida nakon što je pristigao određeni broj odgovora

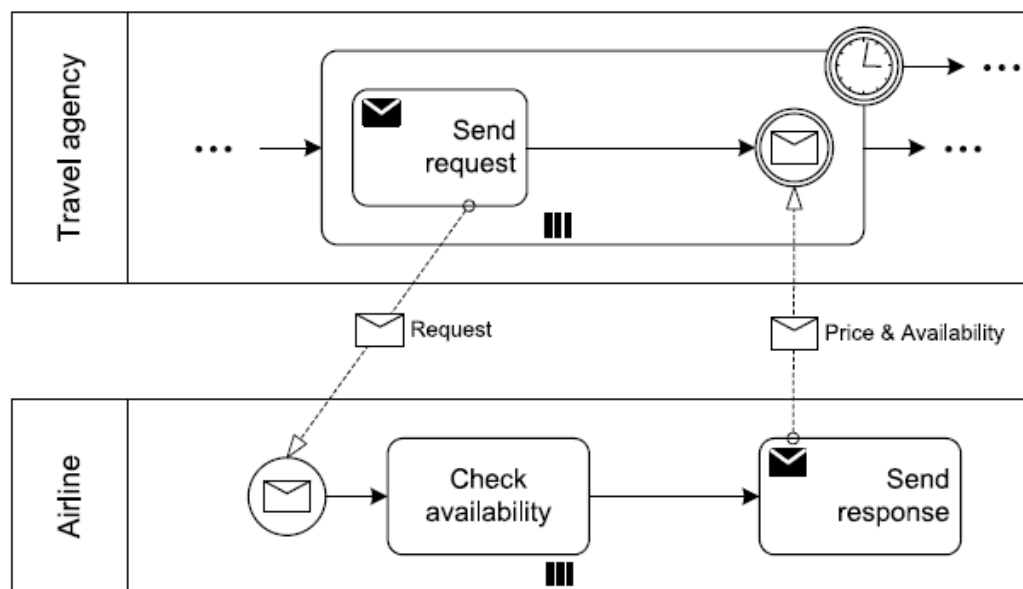


Fig. 5.28. One-to-many send/receive pattern

Šabloni interakcije – *Contingent request*

- Učesnik pošalje poruku drugom korisniku i čeka određeno vreme
- Ako nema odgovora šalje zahtev drugom korisniku

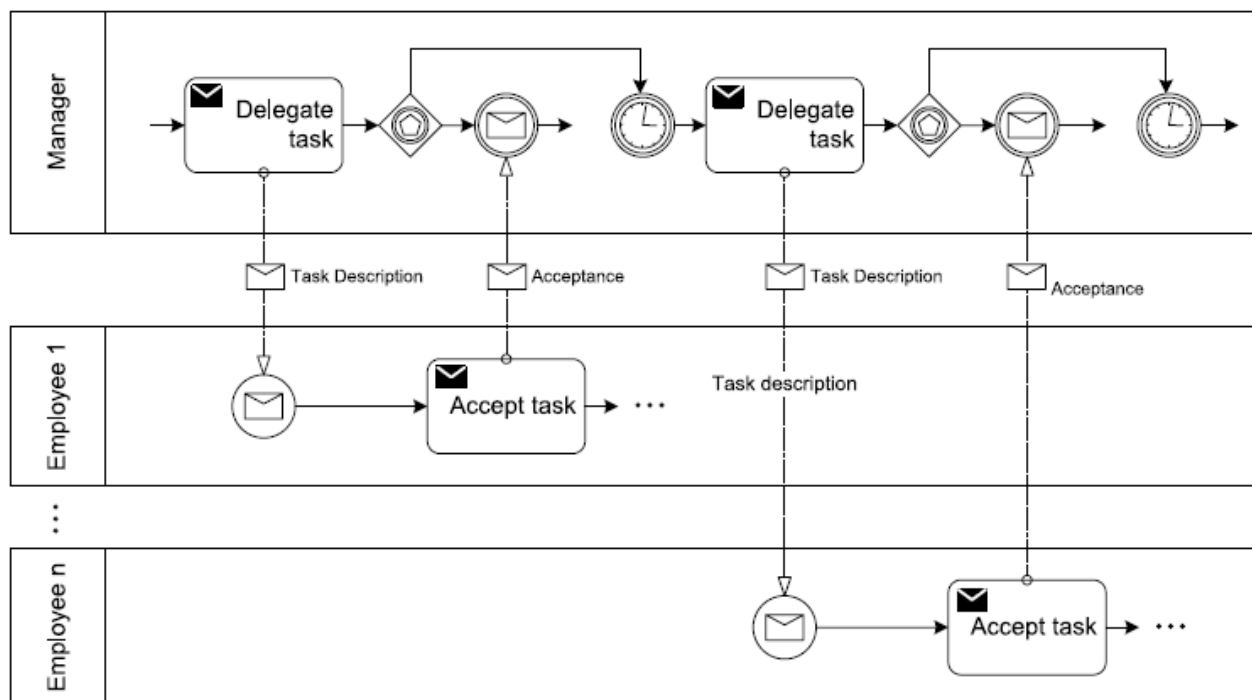


Fig. 5.29. Contingent requests pattern

Šabloni interakcije – *Atomic multicast notification*

- Učesnik šalje notifikaciju za više korisnika koji moraju da je prihvate
- U nekim slučajevima dovoljno je da jedan učesnik potvrdi notifikaciju, u drugim se zahteva određeni broj ili svi

Šabloni interakcije – *Request with referral*

- Posebno važan kod servisno orijentisanih orkuženja gde postoji registar koji dozvoljava povezivanje na odgovarajući servis tokom izvršavanja
- Može se koristiti i u sledćem slučaju: učesnik A šalje poruku učesniku B, a u poruci se referiše na učesnika C. Iako B ne zna za učesnika C unapred, na osnovu informacija iz poruke može odraditi interakciju sa njim (*link passing mobility*)

Šabloni interakcije – *Request with referral*

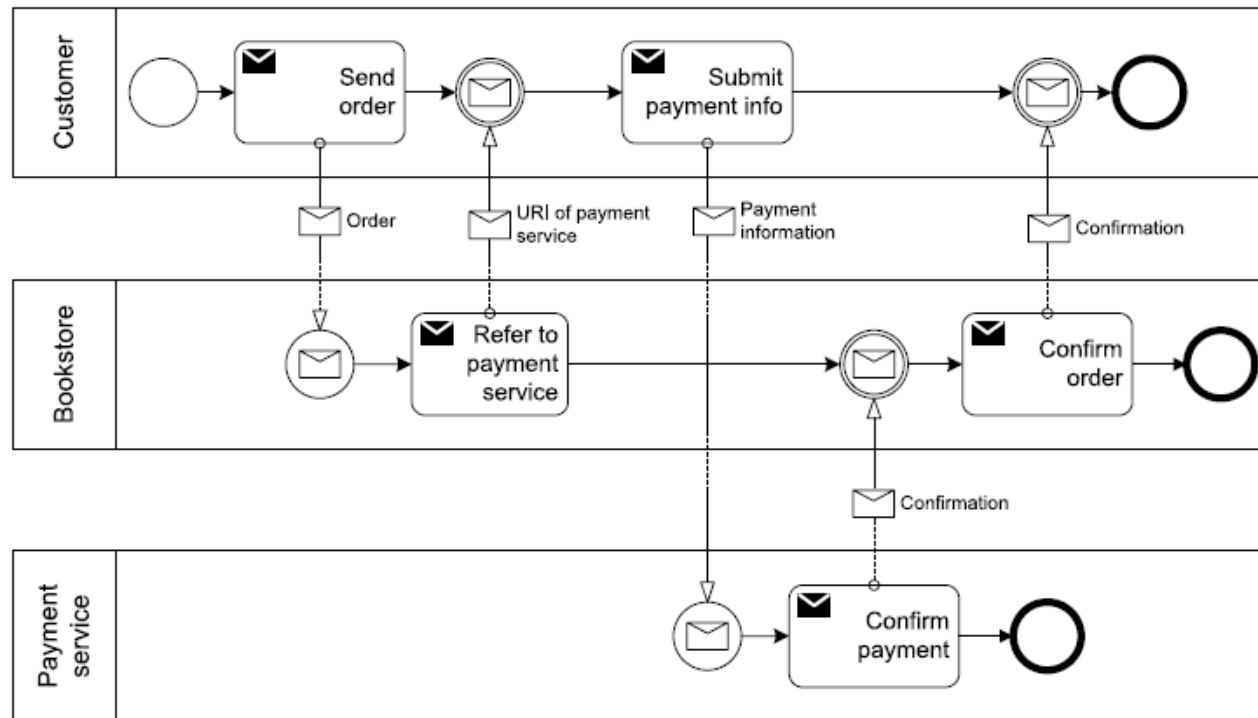


Fig. 5.30. Example involving request with referral pattern

Šabloni interakcije – *Relaying request*

- Uobičajen u sistemima u kojima se saradnja ostvaruje email porukama
- Učesnik A šalje poruku B, koji je prosleđuje učesniku C. Učesnik C nakon toga ima interakciju sa A
- B uvek dobija kopiju poruka kako bi nadgledao konverzaciju

Let's Dance

- Kao alternativa modelovanju interfejsa ponašanja, razvijeni su jezici za direktno modelovanje interakcija
- *Let's Dance* je jedan takav jezik za opis koreografija
 - Baziran je na šablonima za kontrolu toka i šablonima interakcije servisa
 - Glavni fokus je usmeren na prepoznavanje interakcija i njihovih zavisnosti
 - Elementarne interakcije su osnovni elementi ovog modela

Let's Dance - Elementarne interakcije

- Elementarne interakcije su kombinacija *send* modela aktivnosti i *receive* modela aktivnosti
- Primer elementarne interakcije sa uslovom:

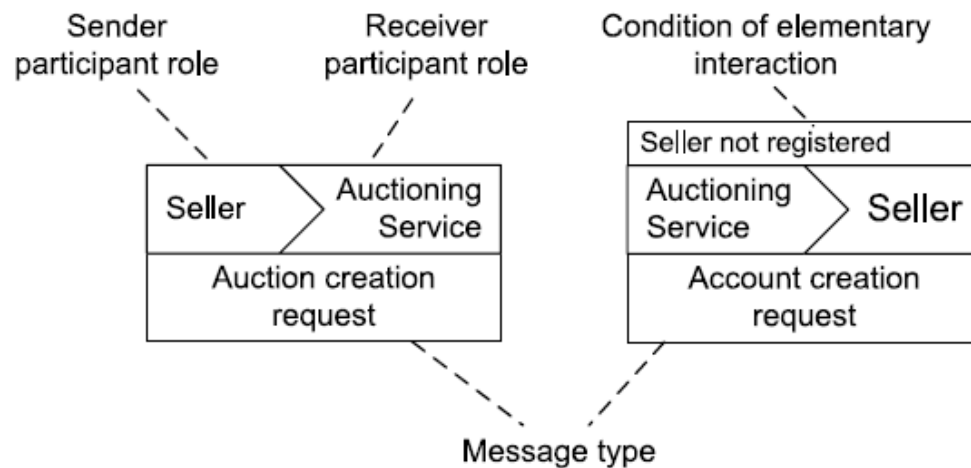
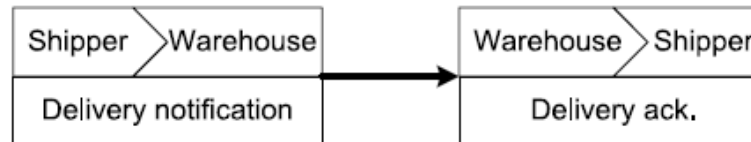


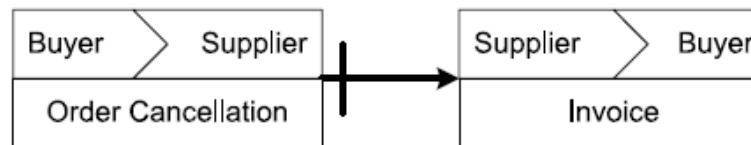
Fig. 5.31. Elementary interaction and conditional elementary interaction

Let's Dance – Osnovna ograničenja za izvršavanje

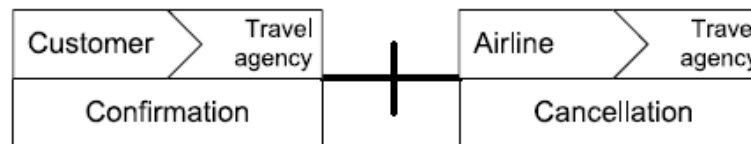
Precedes



Inhibits



Vice-versa-Inhibits



WeakPrecedes

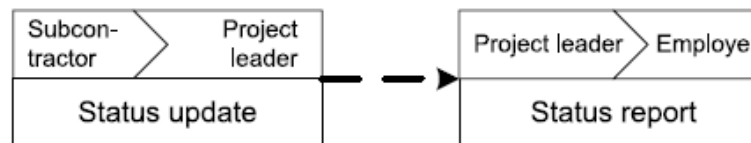


Fig. 5.32. Basic control flow structures relating interactions

Let's Dance – Životni ciklus interakcija

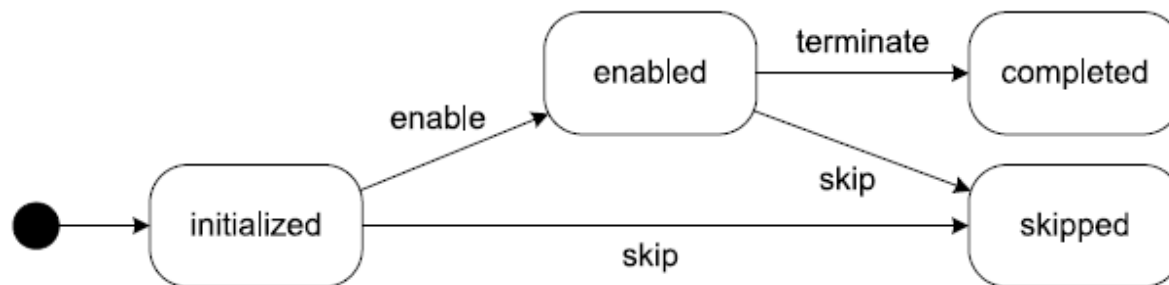


Fig. 5.33. Interaction instance lifecycle

- Instanca interakcije postaje omogućena ako nema *precede* ili *weak precede* interakcija koje kao cilj imaju posmatranu instancu (ili ako su sve interakcije koje su u relaciji *precede* završene, a one koje su u relaciji *weak precede* završene ili preskočene)
- Razmena poruka se može desiti tj. instanca interakcije izvršiti samo ako je onda omogućena
- U slučaju preskakanja, moraju se poduzeti koraci za eliminaciju putanje koje su “slepe ulice”

Let's Dance – Primer povezanih interakcija

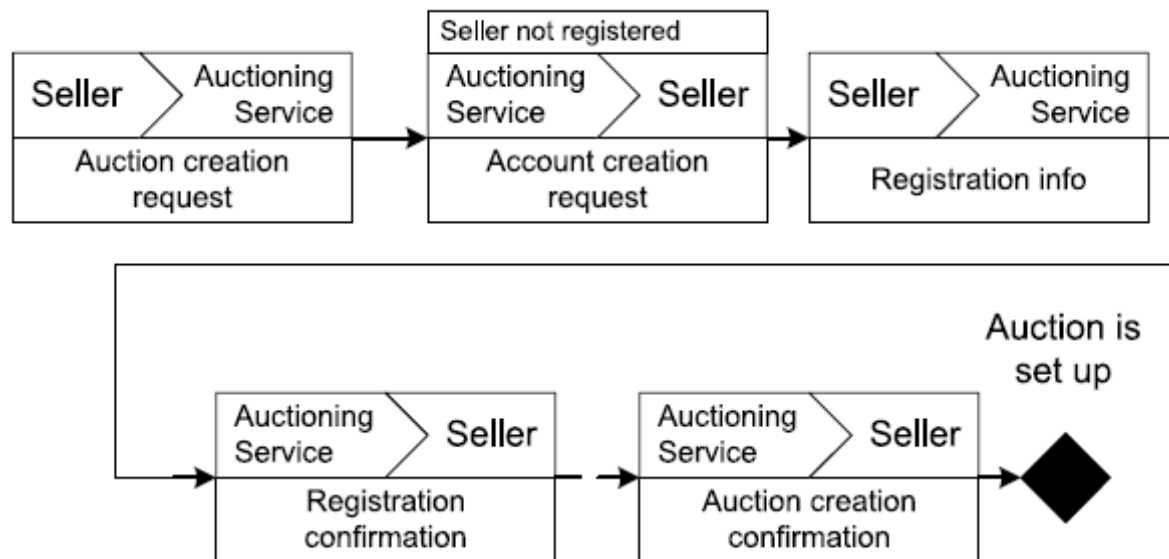


Fig. 5.34. Interaction modelling

Let's Dance – Napredne kontrole toka

- Kompozitne interakcije – više elementarnih interakcija mogu pripadati jednoj kompozitnoj. Nijedna ne može postati omogućena pre nego što to postane kompozitna koja ih sadržava. Kompozitna se završava tek kad su sve sadržane interakcije završene.
- Instance interakcija mogu biti “čuvane”
 - Da bi mogle biti omogućene mora biti zadovoljen određeni uslov ili se preskaču
- Ponavljanja i paralelna grananja sa neograničenim brojem grana se modeluju ponavljanjem interakcija
 - *while, repeat, for each (paralelni), for each (sekvencijalni)*

Let's Dance – Napredne kontrole toka

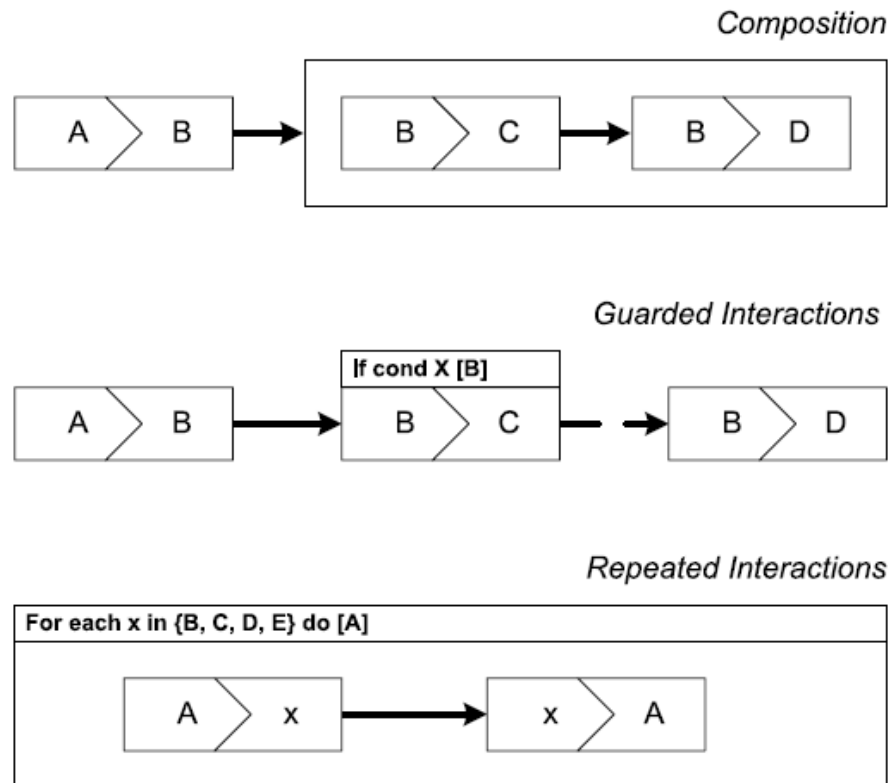


Fig. 5.35. Advanced control flow constructs

BPMN –modelovanje koreografija

- BPMN obezbeđuje bogat skup konstrukta za izražavanje koreografija procesa na različitim nivoima apstrakcije
 - *Na visokom nivou – dijagrami konverzacija*
 - *Na niskom nivou – detaljni dijagrami koreografije*

BPMN – konverzacioni dijagrami

- Daju prikaz saradnje na visokom nivou apstrakcije
 - *Ko komunicira s kim*
- Konverzacija – skup logički povezanih poruka razmenjenih između učesnika
- Kad god se razmeni neka poruka – desila se saradnja procesa
- Konverzacioni dijagrami su neformalna predstava saradnje

BPMN – konverzacioni dijagrami - elementi

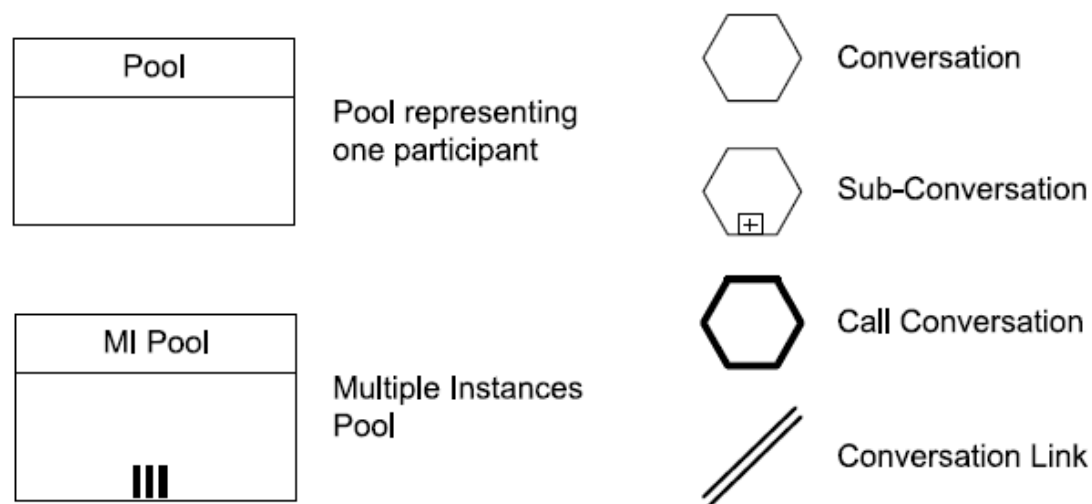


Fig. 5.36. Elements used in BPMN conversation diagrams

- Konverzacija je osnovni element dijagrama
- Mogu biti ugnežedene

BPMN – konverzacioni dijagrami - primer

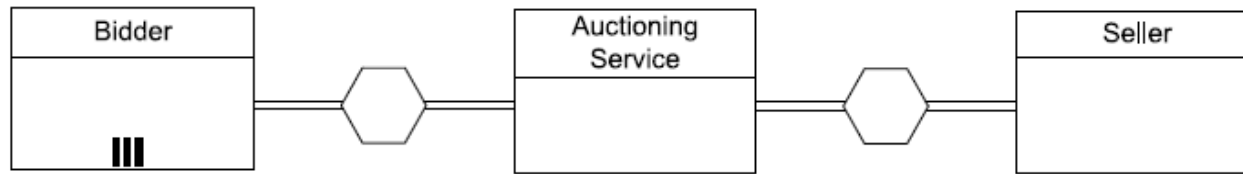


Fig. 5.37. Conversation diagram involving multiple bidders, an auctioning service, and a seller

BPMN – dijagrami koreografije

- Definišu konkretne zavisnosti ponašanja procesa koji učestvuju u koreografiji
- Elementi notacije:

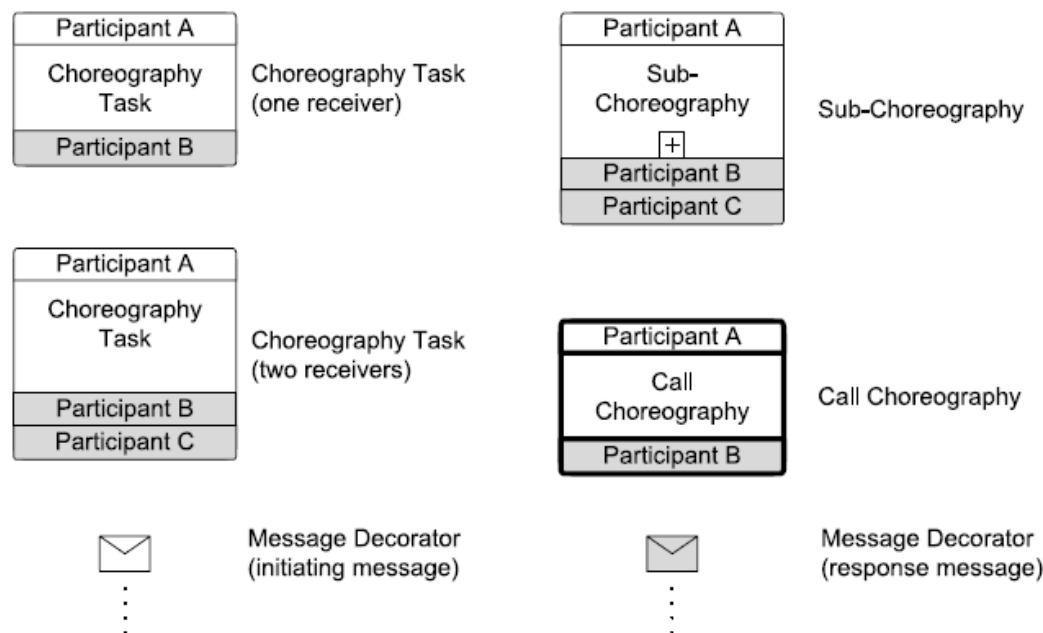


Fig. 5.38. Notational elements used in BPMN Choreography diagrams

BPMN – dijagrami koreografije

■ Osnovni elementi – zadaci koreografije

- Predstavljaju poruku ili povezane poruke koje se razmenjuju među učesnicima
- Svaki skup razmene poruka predstavljen ovim zadacima inicira tačno jedan učesnik (inicijator – prikazan belo; primatelji prikazani sivo)
- Zadatak sa dva učesnika predstavlja razmenu poruka između njih – tipično predstavlja *request/response* ponašanje
- Mogu predstavljati i razmenu poruka između više učesnika
- Podkoreografije su analogne podprocesima

BPMN – dijagrami koreografije

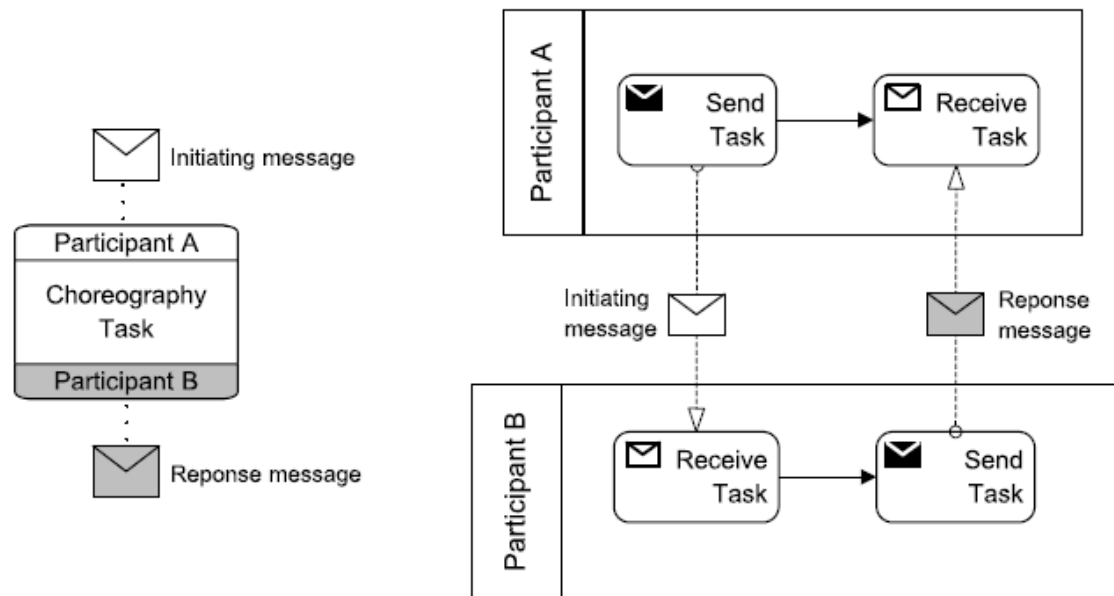


Fig. 5.39. Choreography task and corresponding process diagram

BPMN – dijagrami koreografije - primeri

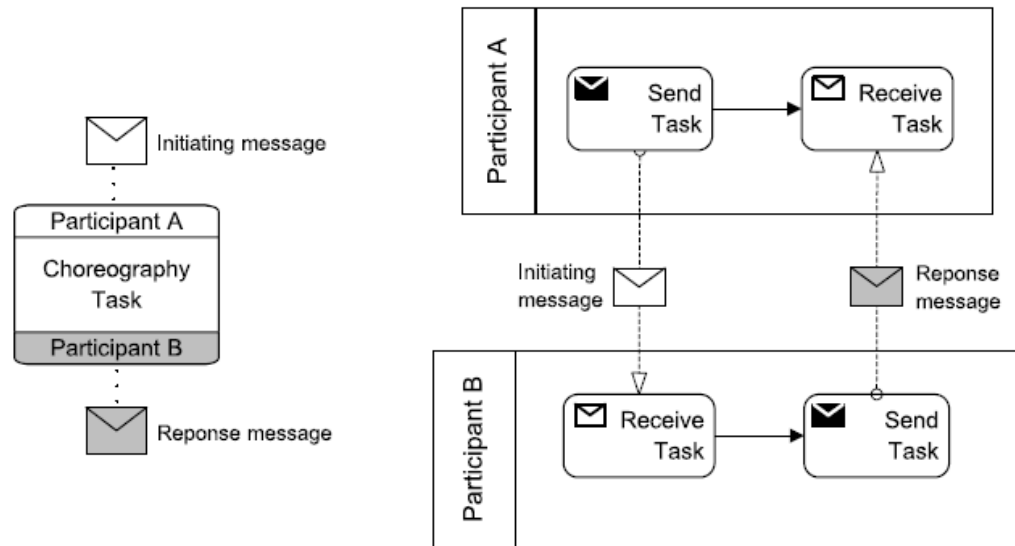


Fig. 5.39. Choreography task and corresponding process diagram

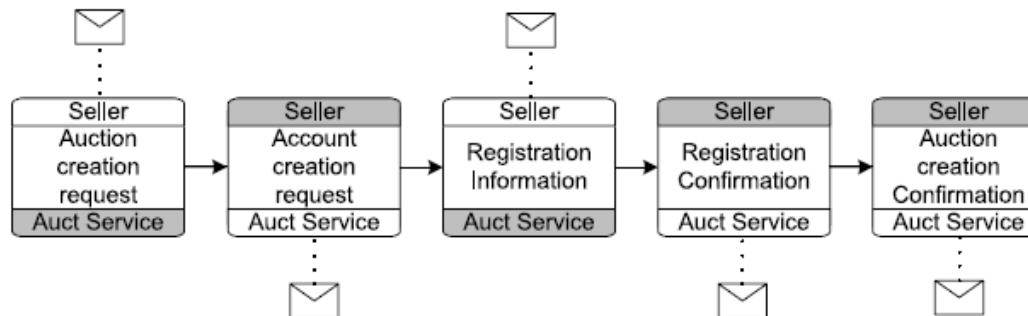


Fig. 5.40. Collaboration scenario, described by BPMN Choreography Diagrams

BPMN – dijagrami koreografije - primeri

- Mogu se koristiti *pool* i tok poruka
 - U ovom slučaju informacija o učesniku ne mora biti u samom *tasku*
 - Ovo postaje nepregledno kod složenijih koreografija

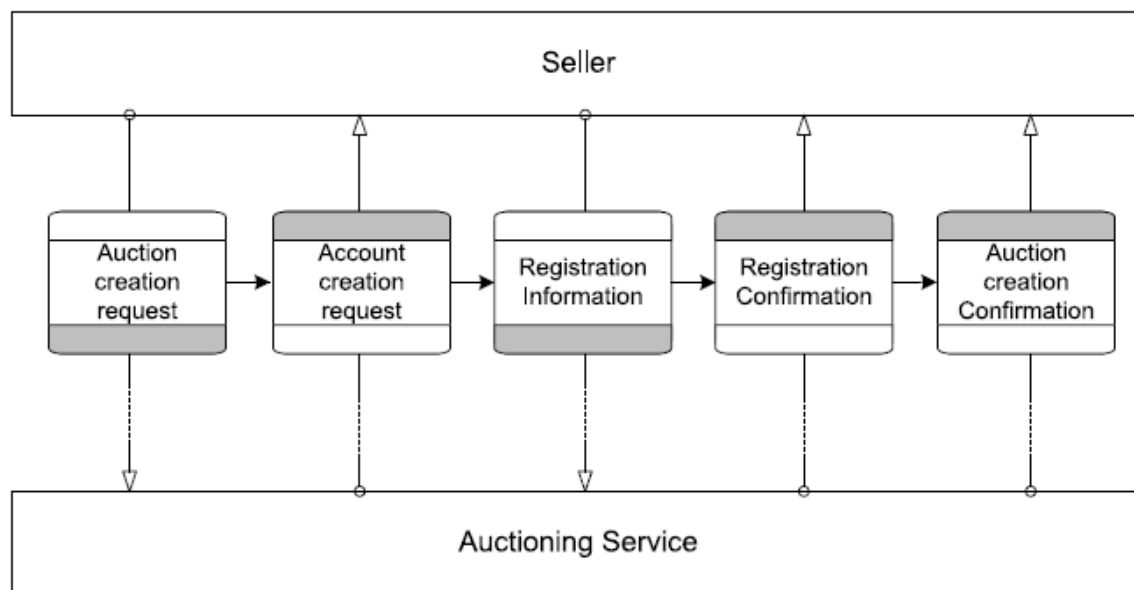


Fig. 5.41. Choreography diagram with pools

BPMN – dijagrami koreografije

- Iako liče na dijagrame procesa postoje konceptualne razlike

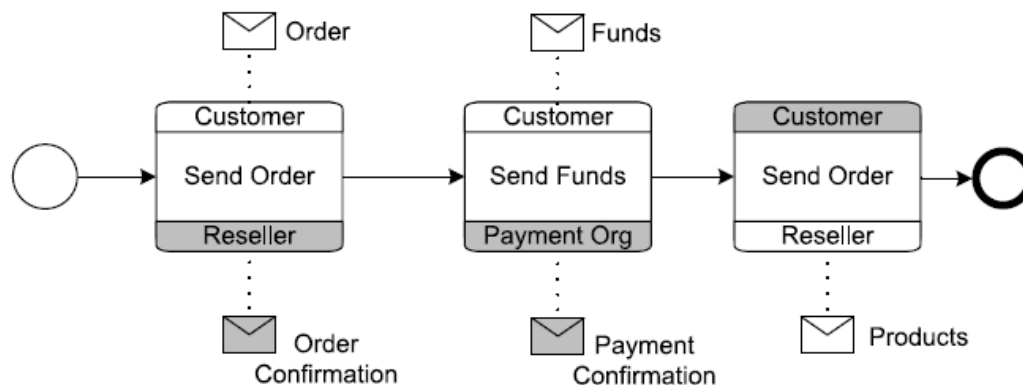


Fig. 5.42. Choreography diagram that is not enforceable, since the reseller cannot know if the customer has already sent the funds

- Problem je sa trećim taskom jer *reseller* ne zna da li se prethodna razmena poruka obavila

BPMN – dijagrami koreografije

- Koreografije sa ovakvim zadacima su *non-enforceable* (nesprovediva)
- Koreografije kod kojih u svakom tasku inicijator ima informaciju da li se prethodna komunikacija su *enforceable* (sprovedive)
- BPMN dozvoljava samo koreografije kod kojih je inicijator taska učestvovao i u prethodnom zadatku

BPMN – grananja u koreografijama

- Semantika *gateway-a* je ista kao u modelima procesa
- Implementacija je kompleksnija jer ne postoji centralna kontrola toka (kao kod pojedinačnog procesa) ni globalno dostupni podaci

BPMN – grananja u koreografijama

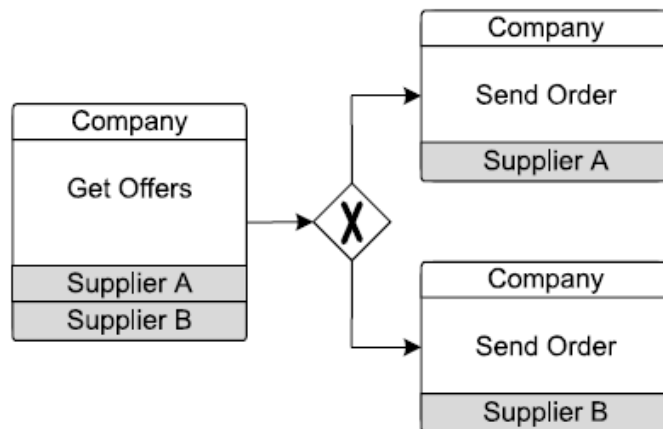


Fig. 5.44. Choreography diagram with exclusive gateway

- Kompanija prvo traži ponude
- Zatim odlučuje šalje narudžbu ili dobavljaču A ili dobavljaču B
- BPMN standard kao jedan od kriterijuma konzistencije za ILI grananje navodi neophodnost da svi učesnici na koje izbor utiče moraju imati dostupne podatke na osnovu kojih je izbor izvršen

BPMN – grananja u koreografijama

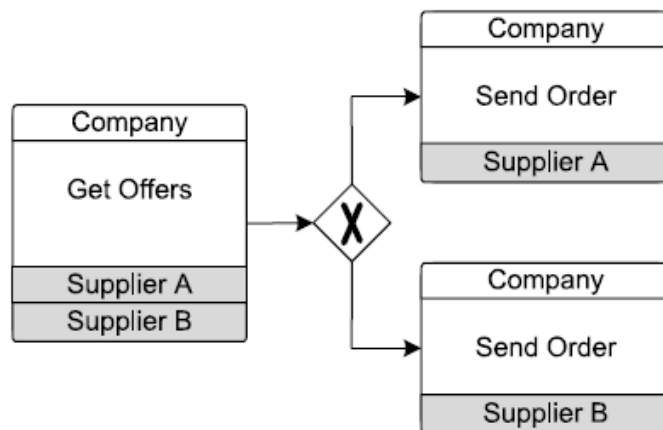


Fig. 5.44. Choreography diagram with exclusive gateway

- Kompanija prvo traži ponude
- Zatim odlučuje šalje narudžbu ili dobavljaču A ili dobavljaču B

BPMN – grananja u koreografijama

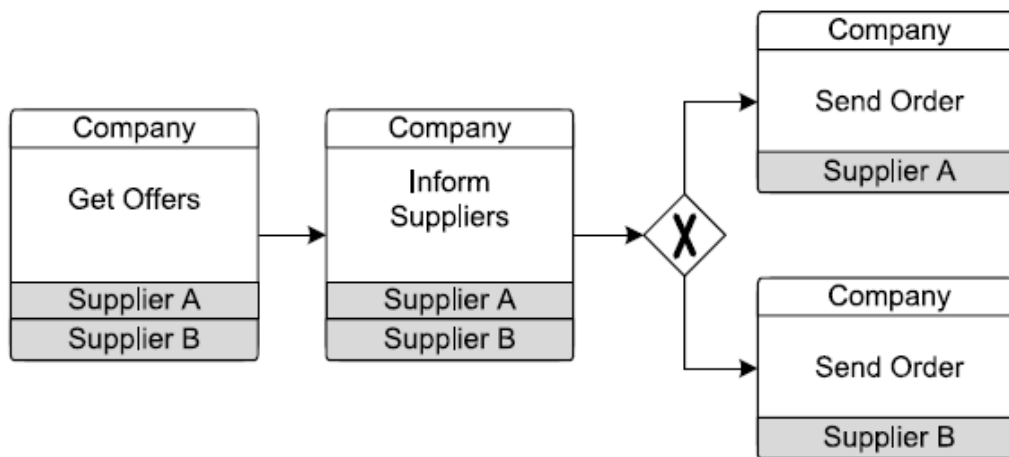


Fig. 5.45. Additional choreography task *Inform Suppliers* makes sure that receivers are informed about the decision taken by the exclusive gateway

- BPMN standard kao jedan od kriterijuma konzistencije za ILI grananje navodi neophodnost da svi učesnici na koje izbor utiče moraju imati dostupne podatke na osnovu kojih je izbor izvršen

BPMN – grananja u koreografijama

- Dijagram koreografije služi i kao polazna osnova za razvoj orkestracije procesa učesnika

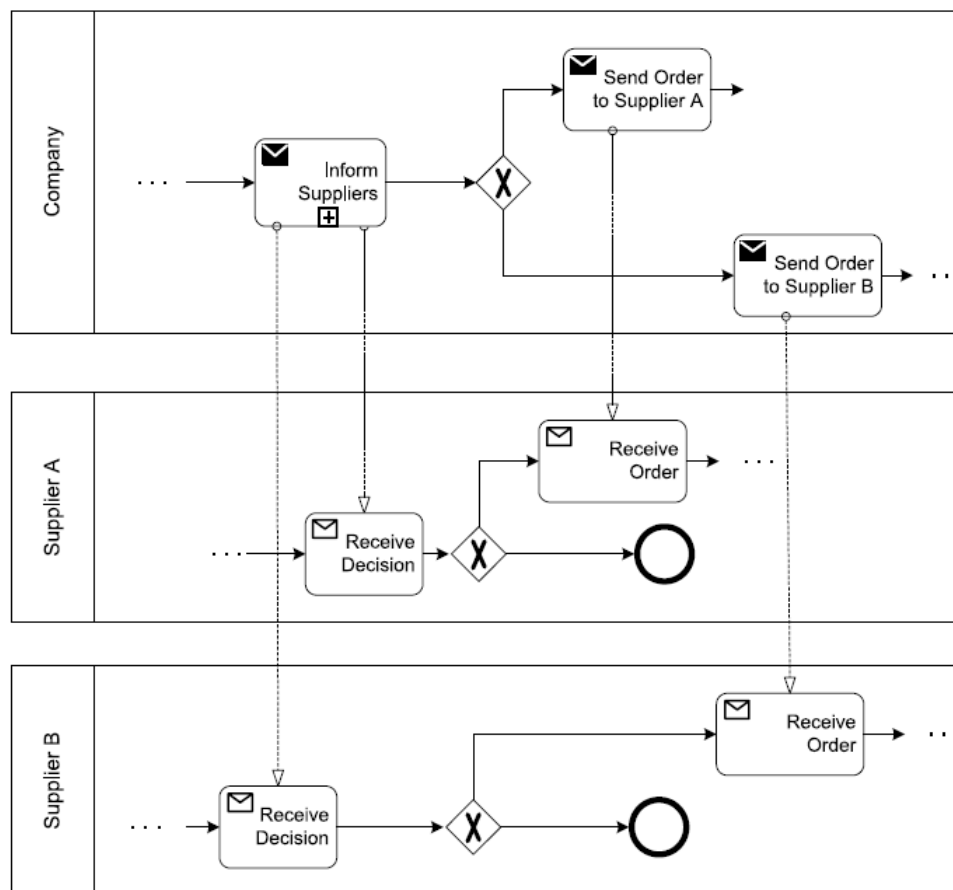


Fig. 5.46. Collaboration diagram with process orchestrations, implementing the choreography defined in Figure 5.45

BPMN – grananja u koreografijama

- Odloženi izbor – grananje uslovljeno događajima
- Partner koji donosi odluku ne mora obavestiti druge o svojoj odluci

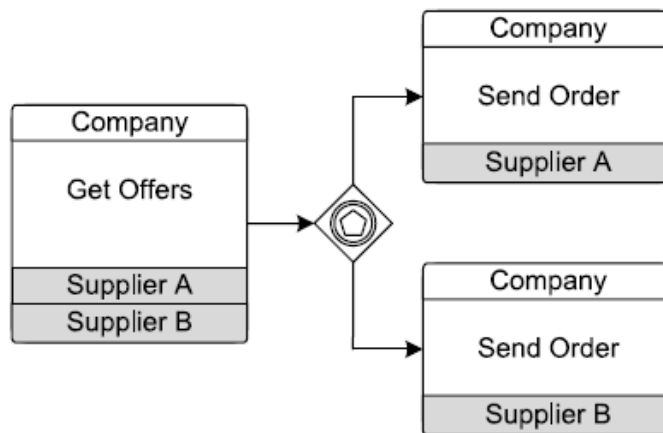


Fig. 5.47. Choreography diagram with event-based gateway

BPMN – grananja u koreografijama

- Odloženi izbor – grananje uslovljeno događajima
 - Primer orkestracije na osnovu prethodnog dijagrama koreografije

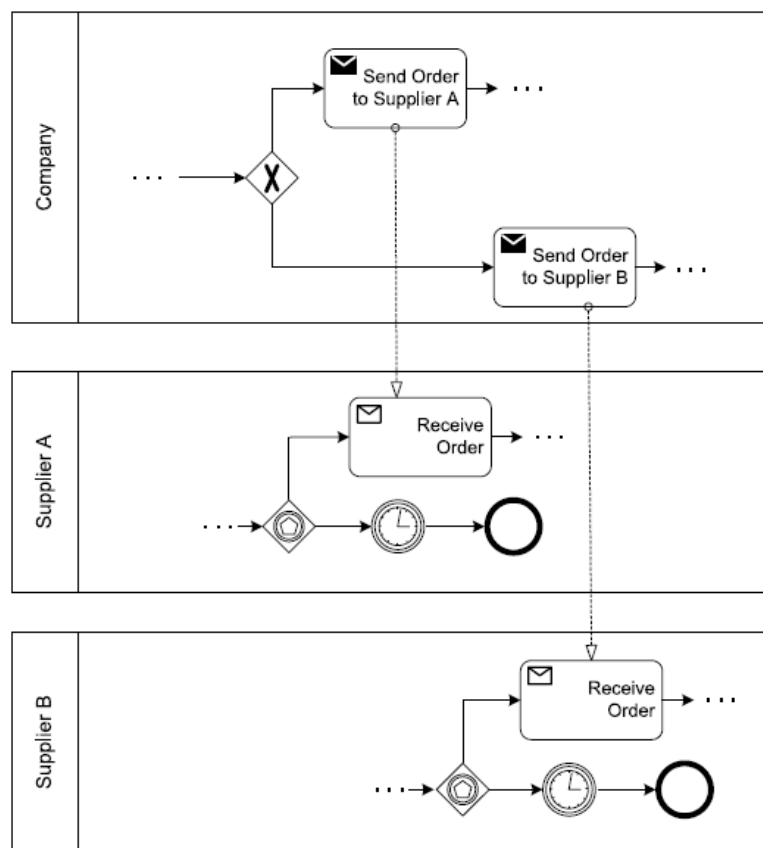


Fig. 5.48. Collaboration diagram with process orchestrations, implementing the choreography defined in Figure 5.47

BPMN – grananja u koreografijama

- BPMN podržava i paralelno i inkluzivno grananje
 - Incijatori taskova nakon grananja moraju biti učesnici u konverzaciji (tasku) pre grananja

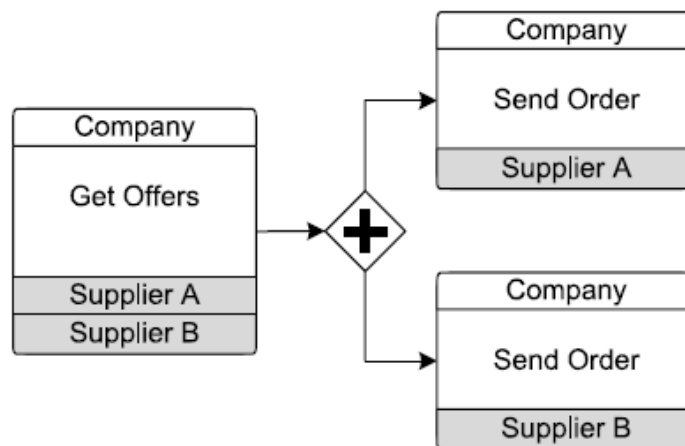


Fig. 5.49. Choreography diagram using parallel gateway

BPMN – grananja u koreografijama

- Dijagram procesa sa implementiranim prethodnim paralelnim grananjem

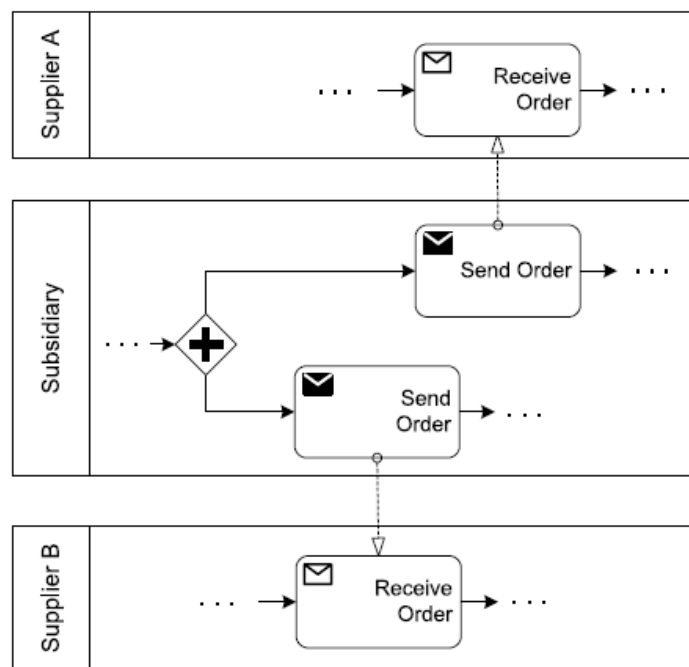


Fig. 5.50. Process diagram realizing parallel gateway of process choreography

BPMN – za kraj primer aukcije

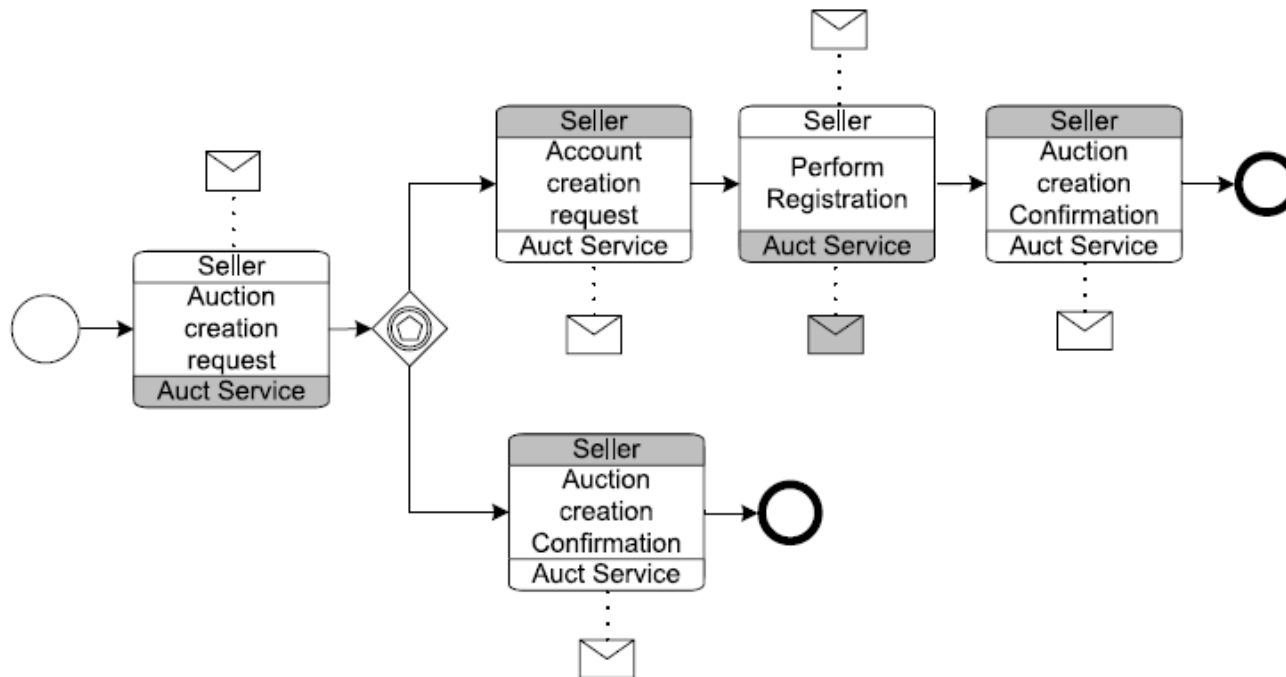


Fig. 5.51. Choreography diagram representing auctioning scenario