

Upravljanje poslovnim procesima (UPP)

Nositelj: izv. prof. dr. sc. Darko Etinger

Asistent: Luka Blašković, mag. inf.

Ustanova: Sveučilište Jurja Dobrile u Puli, Fakultet informatike u Puli



Fakultet informatike u Puli

(2) Dopunski atributi elemenata modela procesa

#2

UPP

Elementi modela poslovnog procesa prikazuju se pomoću karakterističnih simbola vidljivih na grafičkom dijagramu. Oblik svakog simbola otkriva vrstu objekta i njegovu osnovnu funkciju u procesu. Uz to, pojedinim elementima mogu se pridružiti tekstualni, brojčani ili logički podaci koji detaljnije opisuju ponašanje objekta u složenom modelu procesa. U ovoj ćemo skripti nadograditi osnovne objekte koje smo prethodno obradili i istražiti koje dodatne informacije možemo definirati unutar njih.

 **Posljednje ažurirano:** 20.11.2025.

- ispravak na Slici 7

Sadržaj

- [Upravljanje poslovnim procesima \(UPP\)](#)
- [\(2\) Dopunski atributi elemenata modela procesa](#)
 - [Sadržaj](#)
- [1. Uvod](#)
- [2. Nadogradnja poslovnog procesa PRODATI ROBU](#)
- [2.1 Osnovne vrste događaja](#)
- [2.2 Osnove vrste aktivnosti](#)
 - [2.2.1 Radni korak \(eng. Task\)](#)
 - [2.2.2. Potproces \(eng. Subprocess\)](#)
 - [2.2.3 Kako ispravno koristiti send i receive aktivnosti?](#)
- [Zadaci za Vježbu 2](#)

- [1. Proces upisa studenata na Sveučilište](#)
- [2. UPPTech - Implementacija softvera](#)

1. Uvod

Modeliranje poslovnih procesa širok je pojam koji obuhvaća izradu različitih vrsta modela za različite namjene. U literaturi ima više klasifikacija modela poslovnih procesa po različitim kriterijima. U pravilu možemo definirati 3 kriterija za klasifikaciju modela poslovnih procesa:

1. **Razina detaljnosti:** model procesa može biti opisni, analitički ili izvršivi (eng. *executable*)
2. **Faze razvoja:** model se može odnositi na postojeći proces (As is) ili na budući optimizirani proces (To Be) koji tek treba uspostaviti
3. **Pretežiti korisnici:** model može biti namijenjen poslovnim stručnjacima (među kojima je i menadžment), informatičarima ili korisnicima koji su izvan struke

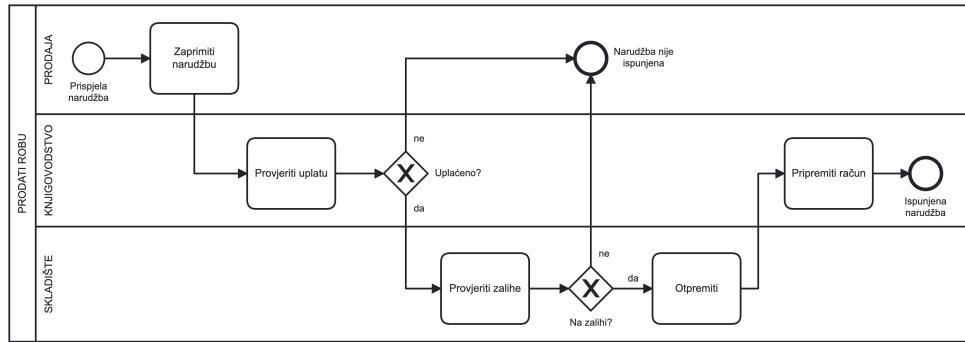
Rekli smo da su objekti toka podataka (eng. *data flow objects*) osnovni elementi modela poslovnog procesa i u pravilu ih možemo podijeliti na **događaje, aktivnosti i skretnice**. To su osnovni elementi svakog modela poslovnog procesa. Ako neki model nema ta tri elementa, onda ga i ne možemo smatrati modelom procesa u smislu definicije dane u prošloj skripti.

Prema normi BPMN 2.0, formalno je moguće da model procesa nema nijedne skretnice, ali to onda znači da je više aktivnosti (ako ih ima) povezane slijednim tokom, odnosno da aktivnosti slijede, jedna iza druge bez ikakvih uvjeta i zastoja. Tada model nije ništa drugo nego popis poslova koje treba obaviti, redom kako su navedeni u popisu.

Također, moguće je da model procesa nema nacrtan nijedan događaj, ali se tada podrazumijeva da postoji jedan početni i jedan završni događaj.

2. Nadogradnja poslovnog procesa **PRODATI ROBU**

Prisjetimo se poslovnog procesa **PRODATI ROBU** koji smo modelirali u prošloj skripti.



Slika 1. Poslovni proces **PRODATI ROBU** koji smo definirali u prošloj skripti

Kroz ovaj proces naučili smo definirati spomenute osnovne elemente. Model smo nadogradili i dodali smo **polje** i dodatne **staze** kojima smo definirali aktore koji sudjeluju u procesu.

Za početak, idemo nadograditi model različitim vrstama **događaja**.

2.1 Osnovne vrste događaja

Rekli smo da događaje definiramo **kružnicama** koje su povezane s tokom. Događaji su važni jer označavaju početak ili kraj neke aktivnosti ili procesa. Definirali smo dva osnovna tipa događaja:

- **Početni događaj** (eng. *Start Event*)
- **Završni događaj** (eng. *End Event*)

Kružnica (za početni i završni događaj)

Početni događaj je **istovjetan početku prve aktivnosti**, dok završni događaj **nastaje kada završi zadnja aktivnost**.

U interpretaciji nekog dijagrama poslovnog procesa, **nastupanje nekog događaja naziva se okidanje** jer se pojmom događaja omogućuje izvođenje aktivnosti koja slijedi poslije njega.

Pri određivanju nekog događaja uvijek treba razmotri što je bio **uzrok** njegova nastanka i gdje je nastao te kakva je **posljedica** njegova pojavljivanja i gdje se ona vidi.

Evo nekoliko različitih primjera događaja:

- referent prodaje primio je e-mail sa zahtjevom za ponudu. Događaj je ovdje **trenutak kad je zahtjev zaprimljen** (koristimo *Message start event*)
- dostignuta je neka vremenska točka u kojoj se mora obaviti aktivnost, npr. **istjecanje roka za dostavu ponude**. Ovdje je događaj **istjecanje roka** (koristimo *Timer start event*)
- odzvonoilo je 6:30 ujutro, a to je **trenutak početka radnog vremena**. Ovdje je događaj **početak radnog vremena** (koristimo *Timer start event*)
- isteklo je neko vrijeme koje smo čekali, npr. postavili smo *timer* na 60 minuta za pečenje kolača. Ovdje je događaj **istjecanje vremena pripreme kolača** (koristimo *Intermediate Timer event ako je usred procesa ili Start Timer event ako je na početku procesa*)
- **roba** je stigla na skladište. Ovdje je događaj **dostava robe** (koristimo običan *start event* ili *message start event*, ovisno o tome kako dobivamo potvrdu)
- ostvaren je neki uvjet koji smo postavili, npr. **kupac je potvrdio narudžbu ili kupac je platio račun**. Ovdje su događaji **potvrda narudžbe i plaćanje računa** (koristimo običan *abstract start event* ili *Intermediate event* ako je usred procesa)
- pojavila se pogreška tijekom provedbe neke aktivnosti. Na primjer, otkazao je stroj u proizvodnoj liniji treba se **pojaviti događaj kvara stroja** (koristimo *Error event*)
- itd.

U BPMN notaciji događaja ima puno, a možemo ih podijeliti na više načina. Mi smo do sad spomenuli samo početne i završne događaje, ali postoje i **međudogađaji** (eng. *Intermediate events*) koji se javljaju tijekom izvođenja aktivnosti.

Međudogađaji su važni jer omogućuju da se aktivnost prekine i nastavi kasnije, ili da se aktivnost ponovi, ili pak da se aktivnost prekine i prebaci na neku drugu aktivnost. Više o međudogađajima ćemo vidjeti kasnije, na vježbama UPP3 i UPP4.

Međudogađaje (eng. *Intermediate events*) definiramo kružnicom s unutarnjom koncentričnom kružnicom.



Početne i završne događaje koje smo do sad definirali bili su **neoznačeni**, u literaturi ih još nazivamo i **none** ili **blank** events.

U BPMN notaciji postoji više vrsta početnih i završnih događaja ovisno o njihovoj ulozi u procesu.

Camunda 8 podržava sljedeće vrste događaja:

- **Neoznačeni događaji** (eng. *None Event*)
- **Obavijest** (eng. *Message Event*)
- **Mjerač vremena** (eng. *Timer Event*)
- **Greška** (eng. *Error Event*)
- **Eskalacija** (eng. *Escalation Event*)
- **Ukidanje ili opoziv** (eng. *Terminate Event*)
- **Kompenzacija** (eng. *Compensation Event*)
- **Signal** (eng. *Signal Event*)
- **Priklučna točka** (eng. *Link Event*)

Međutim, u standardnoj BPMN notaciji [još vrsta događaja](#). Obzirom da se mi prvenstveno bavimo Camunda platformom, koristit ćemo samo one vrste događaja koje Camunda podržava.

Dva koja ćemo često koristiti su **obavijest** (eng. *Message event*) i **mjerač vremena** (eng. *Timer event*).



Dakle, odabiremo **početni događaj** i dodajemo mu oznaku (npr. `početak prodaje`) te dodajemo vrstu događaja. *Message event* je označen pismom unutar kružnice, dok je *Timer event* označen satom unutar kružnice.

Početni Message i Timer događaji

Završni događaj također možemo označiti *Message eventom* koji onda označava da je proces završio i moramo nekoga obavijestiti o tome, primjerice šaljemo e-mail s potvrdom ili računom. Varijanta završnog *Message eventa* izgleda ovako:

Završni Message događaj

Kako interpretirati ove događaje?

Početne događaje još nazivamo *prijamnim* (eng. *catch event*), budući da reagiraju na neki vanjski događaj koji se dogodio.

- *primjer: zaprimanje e-maila* s narudžbom
- *primjer: pristigla je ponuda* za izradu web stranice
- *primjer: pristigla je dokumentacija* za natječaj na interni IT sustav
- *primjer: istekao je rok za dostavu ponude* nekoj javnoj ustanovi

Završne događaje još nazivamo *predajnim* (eng. *throw event*) budući da oni indiciraju da se nešto dogodilo (čime aktivni proces završava), ali i postoje neke posljedice tog događaja - rezultat izvršavanja procesa preusmjerava se nekom vanjskom sustavu, započinje se neki drugi proces - procese se *predaje* (prenosi) drugom procesu.

- *primjer: slanje e-maila* s potvrdom
- *primjer: slanje računa* kupcu
- *primjer: slanje obavijesti* nadređenom o dovršenom zadatku
- *primjer: slanje izvještaja* upravi o dovršenom projektu
- *primjer: dostava materijala* za potrebne građevinske radove
- *primjer: podmirenje obveza* prema dobavljaču

Postoje i **međudogađaji obavijesti** (eng. *Intermediate Message Events*) koji onda mogu biti **catch** ili **throw** ovisno o tome reagiraju li na neki vanjski događaj ili pak generiraju neki vanjski događaj.

Varijanta prijamnog međudogađaja obavijesti izgleda ovako, npr. **primanje e-maila s obavijesti usred procesa**:

Prijamni Međudogađaj obavijesti (eng. *Intermediate catch message event*)

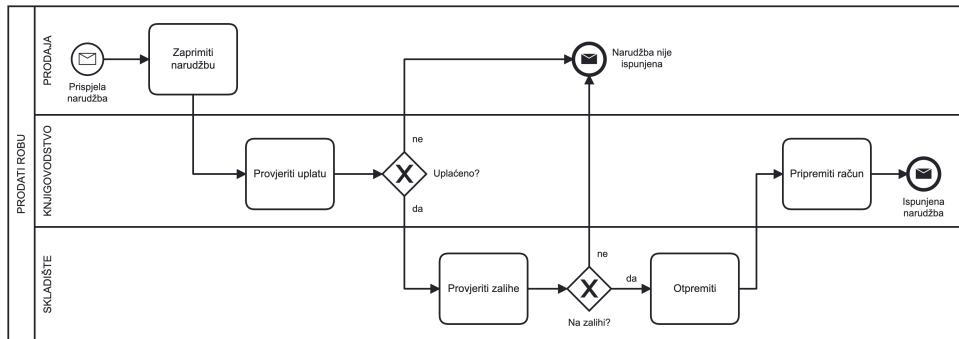
Postoji i varijanta **predajnog** međudogađaja, npr. **slanje poruke usred procesa, najčešće u svrhu pokretanja nekog drugog procesa**:

Predajni Međudogađaj obavijesti (eng. *Intermediate throw message event*)

Idemo nadograditi naš proces **PRODATI ROBU** ažuriranjem događaja.

Kako imamo događaj "Prispjela narudžba", možemo zaključiti da je vjerojatno riječ o *Message eventu* budući da je prispjela ili putem e-maila ili putem nekog drugog sustava koji šalje poruke na određeni komunikacijski kanal (ne mora biti e-mail).

Također, imamo 2 završna događaja, jedan za uspješno **ispunjenu narudžbu**, drugi za **neuspješno ispunjenu narudžbu**. Oba završna događaja možemo označiti *Message eventima* jer će se u oba slučaja poslati e-mail ili obavijestiti klijent na neki drugi način.

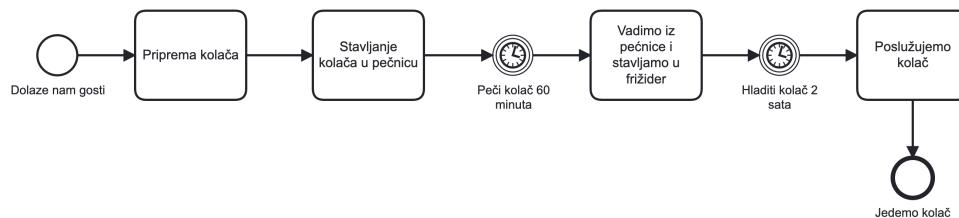


Slika 2. Poslovni proces PRODATI ROBU s dodanim Message eventima

Prije nego se prebacimo na vrste aktivnosti, pogledajmo još jedan primjer korištenja *Timer eventa*.

Timer eventi mogu biti korisni za definiranje vremenskih ograničenja ili rokova, zamislimo jednostavni proces **PRIPREMA KOLAČA**. Zamislimo da radimo kolač koji se peče 60 minuta. Nakon tog vremena, moramo ga izvaditi iz pećnice i pustiti da se ohladi u frižideru barem 2 sata. Nakon toga, kolač možemo poslužiti gostima.

Vidimo da u definiciji procesa imamo vremenska ograničenja, a budući da se ona nalaze za vrijeme izvođenja aktivnosti, možemo koristiti *Intermediate Timer evente*.



Slika 3. Proces PRIPREMA KOLAČA s dodanim Message eventima

2.2 Osnove vrste aktivnosti

Osim različitih vrsta događaja, imamo i različite vrste aktivnosti.

Aktivnost je prema normi BPMN opći (generički) pojam koji se koristi za opis svakog rada izvedenog u poslovnom procesu. Osnovni je simbol pravokutnik sa zaobljenim uglovima, unutar kojeg je opisan **naziv posla koji treba obaviti**.

Za izvođenje aktivnosti potrebni su neki **resursi i vrijeme**. Resursi se uzimaju iz organizacije ili iz okruženja, a vrijeme se odnosi na izvođenje ili trajanje aktivnosti.

Dakle, svaka **aktivnost** označava neku elementarnu radnju koja se u razmatranom dijagramu poslovnih procesa **ne raščlanjuje** i naziva se često i **radni korak**.

Složene aktivnosti u BPMN notaciji često definiramo kao **potprocese** (eng. *subprocess*).

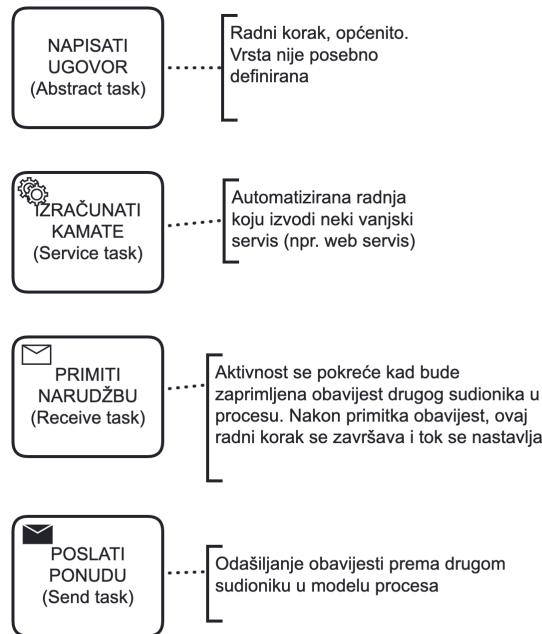
2.2.1 Radni korak (eng. Task)

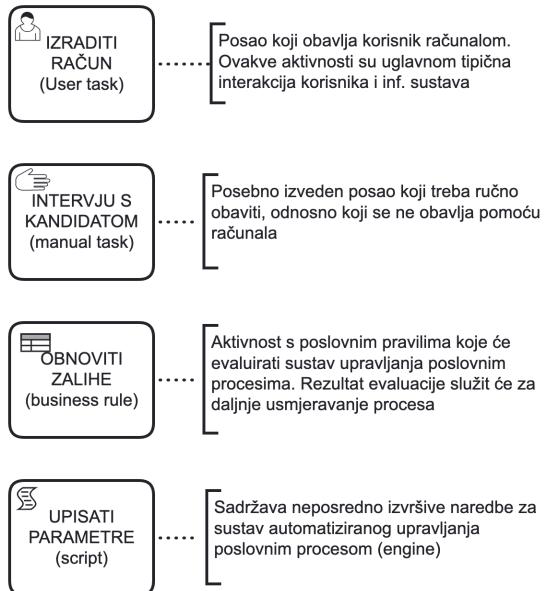
Radnih koraka u BPMN notaciji ima više vrsta, mi smo do sada koristili samo opći (eng. *abstract*) radni korak. U BPMN notaciji postoji više vrsta radnih koraka:

- **Opći radni korak** (eng. *abstract task*)
- **Servisni radni korak ili servis** (eng. *service task*)
- **Prijamni radni korak** (eng. *receive task*)
- **Otpremni radni korak** (eng. *send task*)
- **Korisnički radni korak** (eng. *user task*)
- **Ručni radni korak** (eng. *manual task*)
- **Poslovno pravilo ili pravilo** (eng. *business rule task*)
- **Naputak** (eng. *script task*)

U pravilu ćemo koristiti sve radne korake osim naputka, jer je naputak vrlo specifičan (koristi se pri razvoju procesno-orientiranih Camunda aplikacija).

U nastavku ćemo objasniti svaku vrstu radnog koraka:





Ono što vas za sad može zbunjivati je razlika između **događaja obavijesti** (eng. *message events*) i prijamnih i otpremnih radnih koraka (eng. *send and receive tasks*). Razlike mogu biti zbunjujuće i zato ćemo ih izvježbati u sljedećoj skripti detaljno, međutim sada trebate zapamtiti da su događaji tzv. **okidači** koji pokreću neki proces ili vanjski potproces, ili pak završavaju određeni proces. Međudogađaji se često koriste kao pasivni elementi koji čekaju na neki događaj da se dogodi, dok radni koraci definiraju slijed aktivnosti koji se izvršava i često su vidljivi (eng. *trackable*) korisniku ili sustavu.

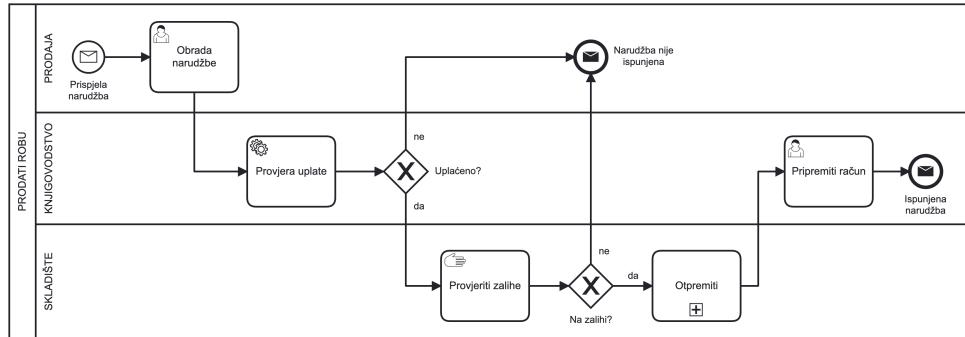
Vrsta radnje obavezno se navodi u analitičkom i izvršivom modelu procesa radi točnog opisa ponašanja aktivnosti u procesnim aplikacijama, gdje će se za svaku vrstu radnoga koraka generirati specifična programska procedura.

Idemo vidjeti kako možemo nadograditi naš proces **PRODATI ROBU** s različitim vrstama radnih koraka.

- Obzirom da poslovni proces **PRODATI ROBU** započinje **message eventom Prispjela narudžba** (eng. *message start event*) koji predstavlja okidač našeg procesa, možemo preimenovati aktivnost koja slijedi nakon toga iz "Zaprimiti narudžbu" u "Obrada narudžbe" budući da je to radnja koja nam pobliže opisuje što se događa u toj aktivnosti.
- Aktivnost "Obrada narudžbe" možemo definirati kao **korisnički radni korak** (eng. *user task*) budući da je to radnja koju obavlja djelatnik poslovnice ili referent prodaje **na računalu kroz neki informacijski sustav**
- Aktivnost "Provjeriti uplatu" možemo također definirati kao **korisnički radni korak** (eng. *user task*) jer se radi o radnji koju obavlja djelatnik poslovnice ili referent prodaje **na računalu kroz neki informacijski sustav**, ali ako je provjera automatska, odnosno ako se uplata provjerava automatski u računovodstvu nakon obrade narudžbe, onda možemo koristiti **servisni radni korak**.
- "Provjeriti zalihe" možemo definirati kao **servisni radni korak** (eng. *service task*) ako se provjera zaliha radi automatski, međutim kako je to neuobičajeno i manje vjerojatno za naš zadatak, možemo ga definirati kao **korisnički radni korak** ako korisnik provjerava zalihe kroz neki informacijski sustav, ili kao **ručni radni korak** ako korisnik provjerava zalihe fizički, npr. odlaskom u skladište.
- "Otpremiti robu" možemo definirati također kao **ručni radni korak** (eng. *manual task*) jer se radi o fizičkoj radnji, odnosno odlasku u skladište i pakiranju robe, međutim uskoro ćete vidjeti kako

preciznije definirati ovu aktivnost kroz **potproces**.

- "Pripremiti račun" možemo definirati kao **korisnički radni korak** (eng. *user task*) jer se radi o radnji koju obavlja djelatnik iz računovodstva kroz neki IS.



Slika 4. Poslovni proces **PRODATI ROBU** s različitim vrstama radnih koraka

2.2.2. Potproces (eng. Subprocess)

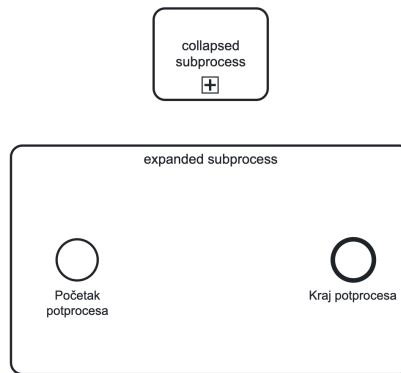
Potproces je posebna vrsta složene aktivnosti koju **ne možemo nazvati radnim korakom** budući da on nije elementarna radnja, već je **složenija radnja koja se sastoji od više manjih radnji**. Potproces se koristi za **grupiranje i organizaciju** aktivnosti koje se **ponavljaju** u procesu ili su **složene i ne mogu se opisati jednostavno**.

Primjer potprocesa iz našeg poslovnog procesa PRODATI ROBU je postojeća aktivnost "Otpremiti".

Ako usporedimo tu aktivnost, s npr. aktivnošću "Provjeriti uplatu" ili "Pripremiti račun", vidimo da je "Otpremiti robu" složenija radnja koju trebamo malo preciznije definirati. Za to koristimo **potproces**.

Potproces je doslovno **proces unutar procesa** te može biti **sklopljen** (eng. *collapsed*) ili **proširen** (eng. *expanded*).

- **Sklopljeni potproces** je onaj koji je skriven, odnosno ne vidimo unutrašnjost potprocesa, dok je
- **Proširen potproces** onaj koji je vidljiv i unutar kojeg možemo vidjeti sve aktivnosti, skretnice, *flow...* (stvarni proces).

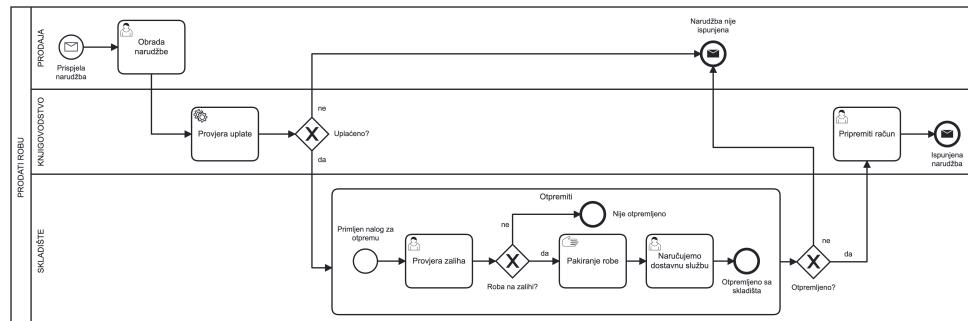


Proširen potproces mora imati **početni i završni događaj** koji označavaju početak i kraj potprocesa!

Kako ćemo definirati potproces "Otpremiti"?

Proces započinje **primitkom naloga za otpremu** u skladište, nakon čega je **potrebno provjeriti zalihe**. Ako je roba na zalihamu, možemo spakirati robu i dogovoriti prijevoz s dostavnom službom. Nakon toga, šaljemo robu i proces završava otpremom robe sa skladišta. Ako roba nije na zalihi, proces završava bez otpreme.

Nakon potprocesa, vraćamo se na glavni proces kroz XOR skretnicu "Otpremljeno?". Ako je otpremljeno, pripremamo račun i završavamo s procesom. Ako nije, narudžba nije ispunjena.



Slika 5. Poslovni proces PRODATI ROBU s dodanim potprocesom "Otpremiti"

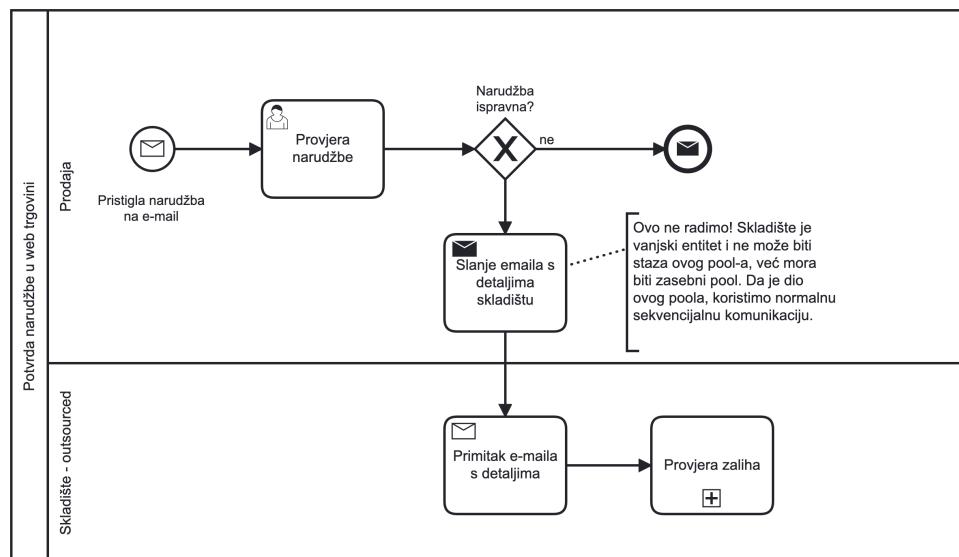
2.2.3 Kako ispravno koristiti **send** i **receive** aktivnosti?

Prijamni (eng. *receive*) i otpremni (eng. *send*) radni koraci koriste se za komunikaciju između više procesa, procesa i vanjskog sustava ili procesa i vanjske jedinice. **Česta greška** je koristiti ove aktivnosti unutar jednog procesa između dviju staza (npr. dvaju odjela u tvrtki). U tom slučaju komunikacijske veze potrebno je definirati kroz slijedne tokove.

Recimo da imamo primjer procesa potvrde web trgovine koja nema svoje skladište, već koristi vanjsko skladište za otpremu robe. Recimo da skladište posluje s mnogo trgovina i komunikacija se odvija prvenstveno putem e-maila. U tom slučaju, koristimo **send** i **receive** između ta entiteta.

Međutim, kako se radi o udaljenim entitetima, ne želimo ih definirati unutar istog polja.

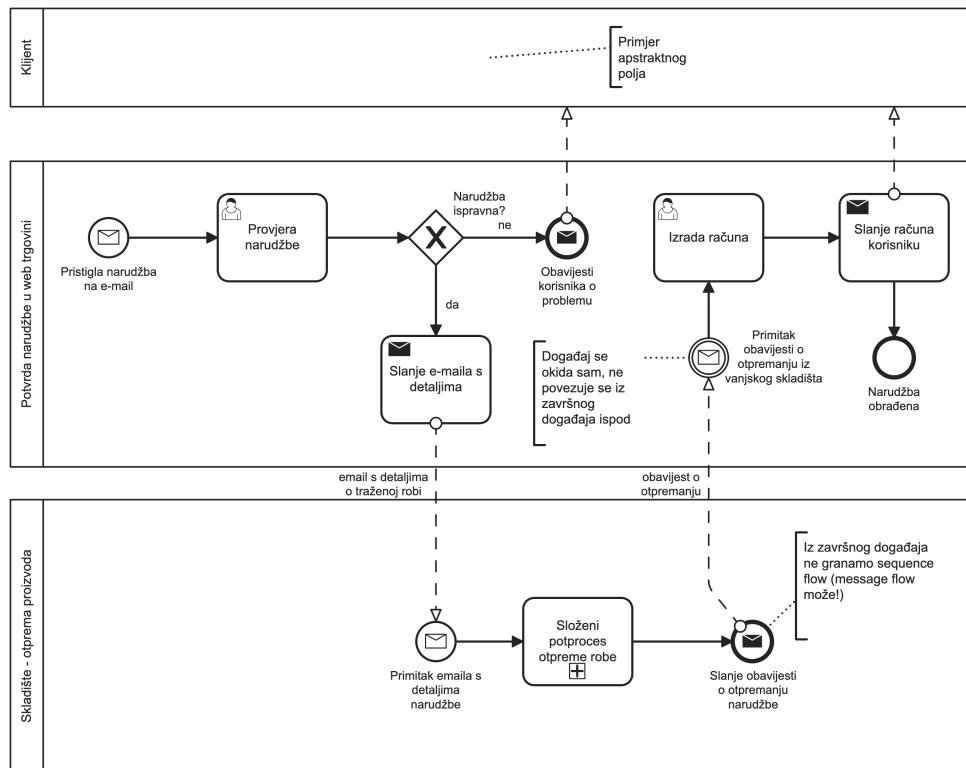
Dakle, sljedeće je pogrešno:



Slika 6. Pogrešan način korištenja **send** i **receive** aktivnosti unutar istog polja

Da se radi o nekoj malo trgovini koja ima svoje malo interno skladište, onda bi to mogli ovako prikazati međutim umjesto **send** i **receive** koristili bismo **obične sekvencijalne veze**.

S druge strane, ako se radi o **vanjskom skladištu**, onda je to ispravno **pokazati na sljedeći način**:



Slika 7. Ispravan način korištenja **send** i **receive** aktivnosti između dva polja.

Note: U modelu postoje lebdeći (floating) elementi (međudogađaj - "Primitak obavijesti o otpremanju iz vanjskog skladišta") koji nije povezan sekvencijalnim tokom s ostatkom modela. Pokazat ćemo kako rješavati ovakve situacije na vježbama UPP3.

Zapamti!

- U pravilu želimo koristiti polje i staze za komunikaciju između entiteta unutar jednog procesa (npr. odjeli unutar iste tvrtke). Rijetko želimo klijenta prikazati stazom unutar procesa, osim ako klijent nije jedan od ključnih dionika u procesu (npr. u procesu korisničke podrške), međutim ako je samo vanjski inicijator procesa, onda ga možemo prikazati vanjskim poljem, ili samo navesti u opisu početnog događaja.
- Dva ili više polja koristimo kada naši entiteti nisu dio iste cjeline, već su udaljeni i komuniciraju prvenstveno kroz komunikacijske kanale (npr. e-mail) ili nemaju povezani interni informacijski sustav. Npr. tvrtka koja koristi vanjsko skladište za otpremu robe, tvrtka koja koristi vanjskog dobavljača za nabavu sirovina, dostavnu službu koja dostavlja robu iz skladišta do kupca, itd.

Ove detalje ćemo detaljno obraditi na budućim vježbama.

Zadaci za Vježbu 2

Na temelju sljedećih opisa poslovnih procesa i do sada obrađene BPMN2.0 notacije, **izradite modele poslovnih procesa u alatu po vlastitom izboru**.

Modele *exportajte* u **png** formatu ili napravite screenshot, *zippajte* zajedno datoteke i učitajte rješenja na **Merlin**.

Slobodno dodajte napomenu ako želite dobiti povratnu informaciju za vaša rješenja.

1. Proces upisa studenata na Sveučilište

Poslovni proces započinje objavom natječaja za upise na studije u određenoj akademskoj godini. Prijave za upis i odabir studija kandidati podnose online putem Nacionalnog informacijskog sustava za upise na visoka učilišta (NISpVU), poznatijeg kao Postani student. Po završetku roka za prijave, sustav generira rang-liste kandidata za upis na pojedine studijske programe. U slučaju posebnih okolnosti, primjerice ponavljanje ispita iz matura iz pojedinog predmeta, upisi se mogu odgoditi za određeni broj dana.

Svako sveučilište (ili druga javna obrazovna ustanova) zaprima rang-liste, nakon čega se provodi ručna provjera dostavljene dokumentacije. Ako postoje nedostaci u dokumentaciji, sveučilište o tome obavještava NISpVU, a problematična dokumentacija šalje se na doradu. U slučaju ispravne dokumentacije (ili nakon ispravaka), sveučilište započinje interni proces evaluacije kandidata na temelju njihovih rezultata i vlastitih kriterija za upis, poput provođenja prijamnih ispita, intervjeta i sličnog.

Nakon završetka evaluacije, sveučilište generira konačne rang-liste kandidata za upis na pojedine studijske programe, obavještava potencijalne studente te započinje teći rok za eventualne žalbe kandidata. Nakon isteka roka, sveučilište razmatra pristigle žalbe (ako ih ima) i svaku rješava pojedinačno. Žalba se može prihvatiti ili odbiti. U slučaju složenijih žalbi, primjerice onih vezanih uz bodovanje na prijamnom ispitu, pravna služba sveučilišta može donijeti odluku o ponovnom bodovanju kandidata ili primjeni neke druge evaluacijske metode ako ista (npr. intervju) nije prikladna za tog pojedinca (npr. u slučaju invaliditeta).

Nakon što su sve žalbe riješene, sveučilište objavljuje konačne rang-liste kandidata za upis na svoje studijske programe te započinje proces upisa studenata. Upisi se provode putem interne aplikacije sveučilišta, a nastala dokumentacija šalje se studentima putem e-maila te arhivira u internom sustavu sveučilišta. Sve navedene radnje provodi Ured za studente i upise.

Po završetku jesenskog upisnog roka, proces je završen, a studenti započinju svoje akademsko putovanje na sveučilištu.

2. UPPTech - Implementacija softvera

Tvrtka UPPTech odlučila je implementirati novi softver za upravljanje poslovnim procesima. Tvrtka se sastoji prvenstveno od management tima, koji donosi poslovne odluke, i malog internog IT tima, koji održava informacijski sustav tvrtke.

Proces započinje kada management tim ustanovi potrebu za implementacijom novog softvera. IT tim tada provjerava dostupne softvere na tržištu i donosi odluku postoji li rješenje koje zadovoljava potrebe tvrtke.

Ako takav softver postoji, management tim dogovara njegovu kupnju, a IT tim ga instalira na server tvrtke. Time proces završava.

Ako odgovarajući softver ne postoji, IT tim odlučuje outsourcati razvoj novog rješenja i kontaktira vanjsku tvrtku CrazyTech. CrazyTech zaprima ponudu putem e-pošte i procjenjuje troškove razvoja. O tome obavještava management tim UPPTecha. Ako je ponuda prihvaćena, management tim šalje potvrdu, čime započinje složeni proces razvoja softverskog rješenja. Ako je ponuda odbijena, proces završava.

Složeni proces razvoja softverskog rješenja započinje definiranjem zahtjeva, dizajnom i razvojem softvera. Nakon razvoja, softver se testira. Ako testiranje prođe uspješno, softver se implementira (deploya) na server tvrtke UPPTech i time proces završava. Ako testiranje ne prođe uspješno, postupak se vraća na fazu razvoja.