Modeliranje poslovnih procesa pomoću grafičkog jezika BPMN

Martinek, Ena

Master's thesis / Diplomski rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:217:511129

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-26**

Repository / Repozitorij:

Repository of the Faculty of Science - University of Zagreb

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO – MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Ena Martinek

MODELIRANJE POSLOVNIH PROCESA POMOĆU GRAFIČKOG JEZIKA BPMN

Diplomski rad

Voditelj rada:

Robert Manger

Zagreb, 2018

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____ , predsjednik

2. _____ , član

3. _____ , član

Povjerenstvo je rad ocijenilo ocjenom _____ .

Potpisi članova povjerenstva:

1. _____

2. _____

3. _____

ZAHVALA

Htjela bih se zahvaliti svojim roditeljima Danku i Željki, sestri Lei, dečku Hrvoju i prijateljima na podršci. Hvala prof. dr. sc. Robertu Mangeru na mentorstvu, te pomoći za izradu ovog diplomskog rada.

Sadržaj

PREDGOVOR 4

1. Općenito o BPMN 5

1.1 Značenje BPMN 5

1.2. Automatizacija procesa 6

1.3. Dijagrami 7

2. Struktura BPMN 9

2.1. Osnovni elementi potrebni za razumijevanje BPMN-a 9

2.1.1. Perspektiva u analizi procesa 10

2.1.1. Modeli, instance, žetoni, korelacije, simboli i atributi 10

2.2. Jednostavan zadatak i obični događaji 11

2.3. Oblikovanje procesnih puteva s grananjem/spajanjem 13

2.3.1. Isključujuće grananje/spajanje bazirano na podacima 13

2.3.2. Paralelno grananje/spajanje 14

2.3.3. Uključivo grananje/spajanje bazirano na podacima 16

2.3.4. Uobičajeni tok i zapinjanje 17

2.3.5. Složeno grananje/spajanje 18

2.4. Oblikovanje procesnih puteva bez grananja/spajanja 20

2.5. Trake 22

2.6. Događaji 24

2.6.1. Važnost događaja u BPMN 24
2.6.2. Događaji poruke 27
2.6.3. Vremenski događaji 27
2.6.4. Događaji pogreške 29
2.6.5. Uvjeti 30
2.6.6. Signalni događaji 30
2.6.7. Događaji prekidanja 31
2.6.8. Događaji povezivanja 33
2.6.9. Događaji naknade 33
2.6.10. Višestruki događaji 35
2.6.11. Paralelni događaji 36
2.7. Potprocesi 36
2.7.1. Potproces u posebnom dijagramu 37
2.7.2. Potproces kao proširenje roditeljskog procesa 37
2.7.3. Događajni potprocesi 38
2.8. Bazeni i tokovne poruke 39
2.8.1. Dirigent i orkestar 39
2.8.2. Pravila primjene 40
2.8.3. Urušeni bazeni 42
2.8.4. Bazeni višestrukih instanca 43
2.9. Podatkovni objekti 44

2.10. Artefakti ... 45
2.10.1. Napomene i grupe ... 45
2.10.2. Prilagođeni artefakti ... 46
3. Usporedba BPMN-a s drugim notacijama ... 47
3.1. Prošireni procesni lanac vođen događajima (eEPC) ... 47
3.2. UML dijagram aktivnosti ... 49
4. Studijski primjer korištenja BPMN-a ... 51
4.1. Opis poslovnog procesa: kupovina karte za autobus ... 51
4.2. Modeliranje poslovnog procesa ... 53
4.2.1. Roditeljski proces ... 53
4.2.2. Potproces „Prijava“ ... 54
4.2.3. Potproces „Kupovina karte“ ... 55
4.2.4. Potproces „Izrada iskaznice i produljenje“ ... 57
4.2.5. Potproces „Naplata“ ... 58
4.2.6. Potproces „Ispiši artikl“ ... 60
Zaključak ... 61
LITERATURA ... 62
SAŽETAK ... 63
SUMMARY ... 64
ŽIVOTOPIS ... 65

PREDGOVOR

Ovaj rad opisuje grafički jezik BPMN (Business Process Model Notation) koji služi za modeliranje poslovnih procesa. Potrebno je naglasiti da svaki poslovni proces možemo opisati na više načina, a stil modeliranja ovisi o dizajneru te o cilju modela. Na dizajneru je odgovornost da njegov model prati željeni cilj u skladu sa standardima BPMN-a.

Rad se sastoji od četiri poglavlja. U prvom poglavlju govori se općenito o BPMN-u, kako je nastao te za što se koristi.

Drugo poglavlje je najopširnije i u njemu se opisuje struktura BPMN-a koju smo podijelili u pet kategorija. Svaka kategorija je objašnjena kroz primjere koji su preuzeti iz knjige [1] gdje se mogu naći i detaljniji opisi koji daju širi uvid u opisane kategorije. Svi primjeri (tj. slike) u ovom radu napravljeni su u programu Microsoft Visio-u.

U trećem poglavlju uspoređuje se BPMN s ostalim notacija. Nadalje poglavlje opisuje prednosti i mane BPMN-a

Četvrto poglavlje je primjena BPMN-a za modeliranje izmišljenog poslovnog procesa. Za studijski primjer odabrana je kupovina karata za autobus. Za izradu dijagrama korišten je program Microsoft Visio. Glavni dijagram, koji se još naziva i roditeljski dijagram, sadrži nekoliko potprocesa koji su prikazani u zasebnim dijagramima.

1. Općenito o BPMN

1.1 Značenje BPMN

Kako bismo bolje razumjeli pojam BPMN-a, prvo ćemo objasniti što je Business Process Management (BPM).

BPM je sistematizirani pristup za zapisivanje, projektiranje, izvođenje, dokumentiranje, mjerenje, praćenje te kontrolu automatiziranih i neautomatiziranih procesa u svrhu postizanja ciljeva i poslovne strategije tvrtke. Kroz sustavno i svjesno upravljanje procesima tvrtke postižu bolje rezultate brže i fleksibilnije. Kroz BPM, procesi se mogu uskladiti sa poslovnom strategijom te tako poboljšati performanse zahvaljujući optimizaciji procesa unutar poslovnih odjela ili čak izvan granica tvrtke. Cilj je razumjeti, a time i poboljšati, cijeli proces a ne samo njegove komponente.

Radi bolje automatizacije procesa razvijen je BPMN (Business Process Model Notation) kao grafička notacija koja služi modeliranju i opisu poslovnih procesa. Pruža standardizirani grafički jezik te olakšava komunikaciju i bolje razumijevanje poslovnih procesa. Kažemo da je BPMN „zajednički jezik“ između poslovnog svijeta i informacijske tehnologije.

Modeli pružaju osnovu za razvoj informatičkog sustava za izvršavanje i podršku poslovnih procesa. Zbog toga modeli trebaju standardiziranu strukturu te trebaju sadržavati sve potrebne informacije za razvoj sustava. Procesi podržani sustavom *(engl. System-supported processes)* sve su više kontrolirani uz pomoć BPMS-a (Business Process Management System). On sadrži procesni stroj koji direktno kontrolira radni tok *(engl. Workflow)* koristeći odgovarajuće modele procesa ili formalni opis procesa. Iz tog razloga modeli trebaju zadovoljiti vrlo stroge zahtjeve jer nisu pretvoreni u računalni program od strane čovjeka, već ih direktno obrađuje stroj. Cilj je osigurati da različiti procesni strojevi tumače i izvršavaju određeni model na isti način. Postoje različiti softveri koji izvršavaju BPMN modele. Primjeri takvih softvera su Activiti i jBPM, open-source softveri napisani u programskom jeziku Java. Više o njihovim funkcionalnostima te načinu rada se može naći na njihovim web stranicama; Activiti [4], jBPMN [5].

Ostali standardi, kao što su dijagrami aktivnosti *(engl. Activity dijagram*) iz UML-a (Unified Modeling Language), nisu prihvaćeni kao modeli poslovnih procesa u praksi.

Njihovo korištenje je ostalo ograničeno na područje objektno-orijentiranog oblikovanja softvera gdje je UML prihvaćeni standard. Zadnjih nekoliko godina BPMN je prihvaćen kao vodeći standard za modeliranje procesa. Također je važno napomenuti da preciznost, kao i formalna ispravnost procesnog modela, varira ovisno o cilju modela i očekivanjima potrošača. Danas mnoge organizacije omogućuju svojim upravljačkim timovima obuku o BPMN-u te uvode BPMN kao svoj organizacijski standard za modeliranje.

1.2. Automatizacija procesa

Da bismo objasnili automatizaciju procesa, pogledajmo prvo običan proces koji nije automatiziran:

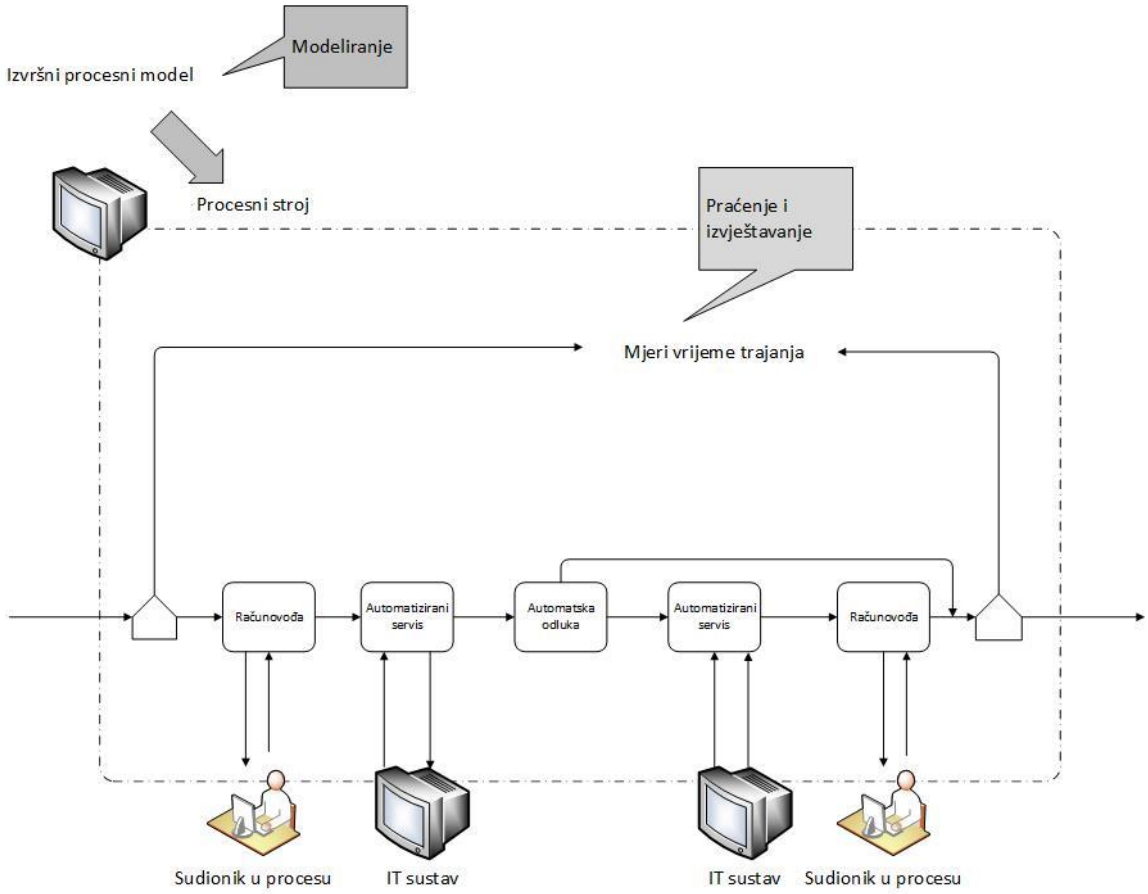
Potencijalni klijent banke predaje zahtjev za odobrenje kredita i taj zahtjev završi na stolu računovođe. Računovođa pregledava zahtjev klijenta, provjerava njegovu kredibilnost kroz njihov sustav. Ukoliko je klijent kreditno sposoban, njegove podatke pohranjuje u softver banke u kojem se čeka odobrenje zahtjeva upravitelja banke.

Zamislimo da imamo isti slučaj, ali da je ovaj proces, za razliku od gornjeg, automatiziran: potencijalni klijent banke preda zahtjev za uzimanje kredita. Službenik u banci skeniranjem prebacuje pisani zahtjev u elektronički zapis. Softver kojeg nazivamo **procesni stroj** *(engl. Process engine/Workflow engine)* preuzima elektronički zapis te ga preusmjerava na virtualnu aplikaciju računovođe. Računovođa pristupa aplikaciji, pregledava i pritiskom na dugme, uz pomoć procesnog stroja, pristupa agenciji za kreditni rejting u kojoj pronalazi sve relevantne podatke potrebne za izradu izvješća. Ukoliko je klijent kreditno sposoban, procesni stroj preusmjerava izvješće u bankovni posebni softver, koji smo gore spomenuli, i stavlja zahtjev za odobravanje na popis zadataka upravitelja.

U primjeru smo ilustrirali sljedeća načela automatizacije procesa:

- Automatizacija ne znači nužno da je cijeli proces u potpunosti automatiziran
- Glavna komponenta automatizacije je procesni stroj koji izvršava procesni model
- Procesni stroj kontrolira proces tako da informira ljude o zadacima koje moraju izvršiti
- Procesni stroj odlučuje koji zadaci se trebaju izvršiti, a koji ne, s obzirom na uvjete njihovog izvršavanja
- Ljudi mogu i dalje utjecati na konačnu odluku

Primjer automatizacije koje smo naveli, ilustriran je na slici 1.1.



Slika 1.1 Primjer automatiziranog procesa: podnošenje i odobravanje zahtjeva za kredit

1.3. Dijagrami

Najvažniji dio BPMN-a je grafički prikaz modela. BPMN pruža tri vrste dijagrama:

- ***Procesni dijagram ili dijagram suradnje*** *(engl. Process or collaboration diagram)*

U ovom tipu dijagrama možemo modelirati procesni tok, uključujući aktivnosti, razdvajanja, paralelne tokove itd. Također je moguće prikazati suradnju dva ili više procesa s njihovim međusobnim porukama. Iako procesni dijagram i dijagram suradnje pripadaju istom tipu dijagrama, dijagram sa samo jednim procesom zovemo *procesni dijagram*, dok dijagram s više interaktivnih procesa zovemo *dijagram suradnje*.

- ***Koreografski dijagram*** *(engl. Choreography diagram)*

Ovaj tip dijagrama sadržava modeliranje razmjene podataka između različitih partnera, slično kao u suradnji. Međutim, svaka izmjena podataka modelirana je kao aktivnost tako da je na ovom nivou moguće predočiti razdvajanja i petlje kako bi se prikazale složene razmjene između protokola.

- ***Razgovorni dijagram*** *(engl. Conversation diagram)*

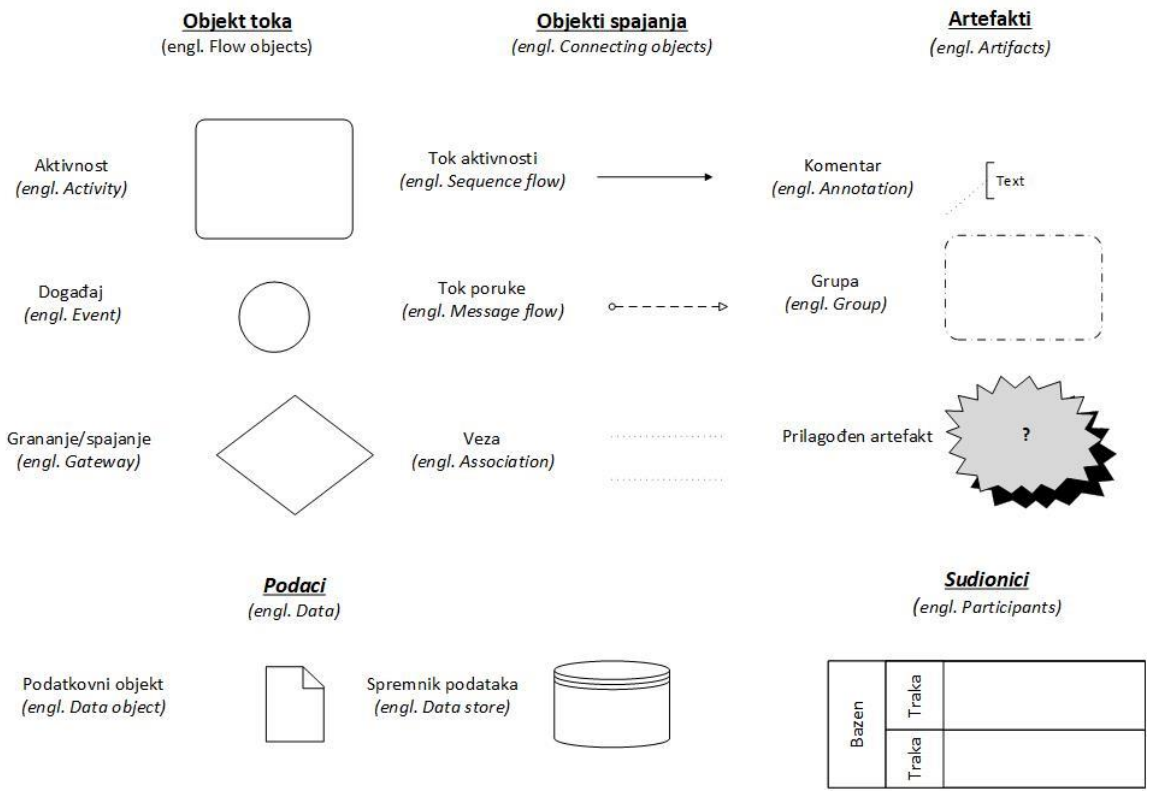
Razgovorni dijagram omogućava pregled nekoliko partnera i njihove međusobne odnose.

Najčešći tip dijagrama je procesni dijagram ili dijagram suradnje. Neki BPMN alati i knjige koriste samo ovaj tip dijagrama. Iako je nedvojbeno najznačajniji tip dijagrama, postoje korisna područja primjene i za ostale tipove dijagrama.

2. Struktura BPMN

2.1. Osnovni elementi potrebni za razumijevanje BPMN-a

Kod crtanja dijagrama u BPMN-u koristimo simbole koji su raspoređeni po kategorijama kao što je pokazano na slici 2.1.Te kategorije nazivamo osnovni elementi BPMN-a.



Slika 2.1. Osnovni elementi BPMN-a

Općenito, određeni zadaci moraju se izvršiti tijekom procesa (aktivnosti, *engl. Activities*), pod nekim uvjetima (grananje/spajanje, *engl. Gateways*) te stvarima koje se mogu dogoditi (događaji, *engl. Events*). Ono što povezuje ova tri **objekta toka** *(engl. Flow objects)* su strelice pod nazivom **tok aktivnosti** *(engl. Sequence flows)*, ali samo unutar jednog **bazena** *(engl. Pool).* Izvan granica bazena koristi *se* **tok poruke** *(engl. Message flow).*

Artefakti (*engl. Artifacts*) daju dodatne informacije o procesu, ali ne utječu direktno na redoslijed objekta toka. Za spajanje svakog objekta toka s artefaktom koriste se **veze** (*engl. Associations*). Simbol za **sudionika** *(engl. Participant)* i za njegov proces je u bazenu; svaki proces dobiva svoj vlastiti bazen.

BPMN sadrži dodatnu kategoriju s podacima *(engl. Data).* Ona se odnosi na stvaranje, obradu i spremanje podataka koji mogu postati relevantni za proces. Njezini simboli najčešće povezuju aktivnosti kroz veze. Postoje još tri aspekta potrebna za potpuno razumijevanje BPMN-a:

- Napredne ideje i pravila ponašanja iza ove jednostavne sheme
- Veliki raspon simbola
- Praktično znanje

Prve dvije stavke će biti pobliže objašnjene kasnije, dok je za treću potrebno iskustvo.

2.1.1. Perspektiva u analizi procesa

BPMN je baziran na pretpostavci da jedan ili više **sudionika** mogu postojati unutar jednog dijagrama. Sudionik je logički element na kojeg se primjenjuju sljedeća pravila:

- Svaki proces može sadržavati samo jednog sudionika (ovdje mislimo na logički element jer proces može sadržavati više ljudskih sudionika)
- Sudionik ima potpunu kontrolu nad procesom
- Sudionik je potpuno odgovoran za proces kojim upravlja
- Ostali sudionici ne mogu utjecati na proces; možda ni ne znaju kako radi

Jedan proces može izgledati potpuno drugačije za svakog sudionika, a to ovisi o njihovoj perspektivi. To rezultira drugačijim modelom procesa. U BPMN-u, simbol za sudionika i njegov proces je bazen*;* svakom procesu dodijeljen je njegov vlastiti bazen. Doduše, logički, sudionik može kontrolirati više od jednog procesa.

2.1.1. Modeli, instance, žetoni, korelacije, simboli i atributi

Model procesa *(engl. Process model)* najosnovniji je opis procesa u obliku dijagrama u kojem je moguće opisati jedan ili više procesa. **Instanca procesa** *(engl. Process instance)* je proces proveden u stvarnosti. Na primjer, jedna žalba klijenta je instanca procesa žalbi. Neki procesi mogu se instancirati samo nekoliko puta u godini, poput kvartalnog izvještaja knjigovodstvenog ureda, dok se drugi mogu češće instancirati (kao primjer možemo uzeti zahtjev za uzimanje kredita u banci).

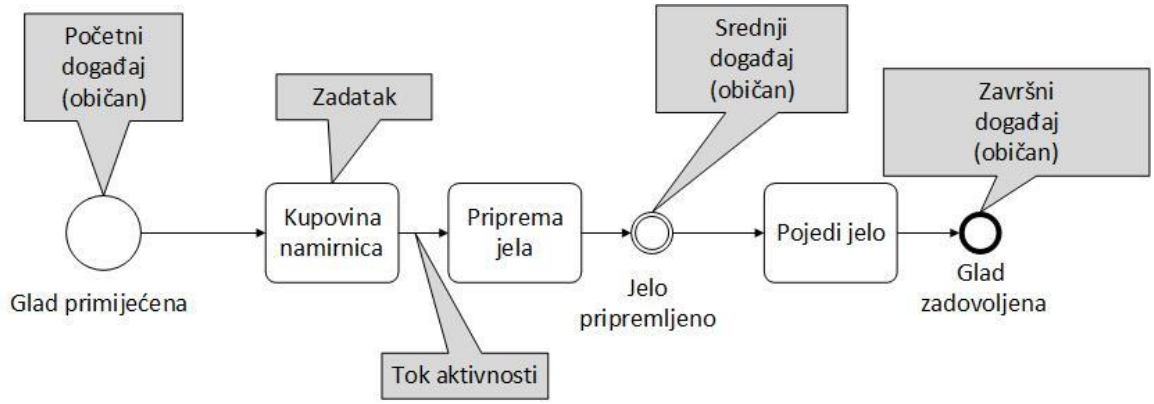
Model žetona *(engl. Token model)* može se primijeniti ukoliko imate na umu model procesa i želite saznati koji se putevi unutar procesa mogu ili moraju koristiti tijekom procesa instance. Žeton putuje po mogućim putevima u dijagramu. Ako naiđe na dva moguća puta, stvara se klon žetona koji usporedno s originalnim nastavlja put po dijagramu. Svaki put žetona u dijagramu predstavlja jednu instancu procesa.

Kako bismo bolje objasnili što je korelacija, poslužit ćemo se primjerom: prilikom podnošenja zahtjeva, dodjeljuje vam se jedinstveni broj zahtjeva. Broj zahtjeva je jedinstveni ključ koji je vezan specifično za taj objekt te se njime pozivate na zahtjev koji ste predali. Upravo ta alokacija temeljena na jednoznačnom ključu naziva se **korelacija**.

Specifikacija BPMN-a opisuje simbole predviđene za modeliranje procesa. Također opisuje mnoge atribute koji se mogu dodijeliti simbolima. Mnogi atributi ne pojavljuju se u dijagramu; takvi atributi su pohranjeni u alatima za modeliranje te se koriste kada procesni stroj izvršava modelirani proces.

2.2. Jednostavan zadatak i obični događaji

Na jednostavnom primjeru proučit ćemo jedan proces. Na slici 2.2 opisan je proces koji se događa kada je osoba gladna. Ovdje rabimo izraz **običan događaj** *(engl. None event)* jer postoje i specifični događaji koje ćemo kasnije opisati.



Slika 2.2. Proces kada je osoba gladna

Proces opisan riječima bi bio:

- (početni događaj) osoba osjeti glad
- (aktivnost) odlazi u dućan po namirnice
- (aktivnost) priprema obrok
- (srednji događaj) jelo je pripremljeno

- (aktivnost) jede
- (završni događaj) osoba više nije gladna

Da se u konačnici dogodi željeni ishod, koristimo ***zadatke*** *(engl. Tasks)*. Zadaci su srce procesa. U BPMN-u, zadaci su tehnički dio kategorije aktivnosti. Prilikom imenovanja zadataka pokušavamo se držati načela objektno-orijentiranog dizajna [glagol] + [objekt]. Na primjer, rekli bismo *kupiti namirnice* umjesto *najprije se pobrinuti za kupovinu namirnica.*

Na našem primjeru na slici 2.2. koristili smo običan događaje. Obični događaji mogu se koristiti u procesnom toku da bi se označio status ili prekretnica. Općenito, **događaji** *(engl. Events)* opisuju značenje stvari koje se događaju prije, za vrijeme i na kraju procesa:

- **Početni događaji***(engl. Start events)* prikazuju koji događaji uzrokuju pokretanje procesa
- **Srednji događaji** *(engl. Intermediate events)* predstavljaju status koji se postiže u procesu i on je eksplicitno modeliran. Često se upotrebljavaju u slučaju kad dostizanje određenog statusa smatrate prekretnicom i želite izmjeriti vrijeme potrebno za dostizanje te prekretnice.
- **Završni događaji** *(engl. End events)* označavaju status dosegnut na kraju procesa.

Čak i za jednostavnije događaje moramo napraviti daljnje razlike. Početni događaji su **događaji hvatanja** *(engl. Catching events).* To znači da se nešto dogodilo neovisno o samom procesu, ali proces mora čekati da taj događaj prođe ili reagirati na njega.

Srednji događaji mogu nastati, biti uzrokovani ili biti pokrenuti od strane samog procesa. Ukoliko ih proces sam pokrene, nazivamo ih još i **događaji bacanja** *(engl. Throwing events).* Srednji obični događaji označavaju status postignut samim procesom pa je stoga on sam događaj bacanja.

Završni događaji se odvijaju kada proces više ne reagira na ništa. Kao posljedica toga samo ih proces sam može pokrenuti.

BPMN događaji se odnose na nešto što se već dogodilo bez obzira na proces (ukoliko je riječ o događajima hvatanja) i rezultat procesa (ukoliko je riječ o događajima bacanja). Iz tog razloga koristimo prvo [*objekt*] i napravimo [*glagol*] u pasivnom značenju pa zapis u našem slučaju glasi: *glad primijećena.* U BPMN-u, nije nužno započeti s početnim događajem i završiti ga sa završnim događajem, njih možemo izostaviti. Ukoliko se pak odlučimo započeti s početnim događajem, moramo svaki puta završiti sa završnim. Vrijedi i obrnuto, završni događaji zahtijevaju početni događaj. Dobra je praksa uvijek započeti s početnim i završiti sa završnim događajem iz dva razloga:

1. Na taj način moguće je odrediti okidač procesa
2. Može se opisati konačan status za svaki kraj puta.

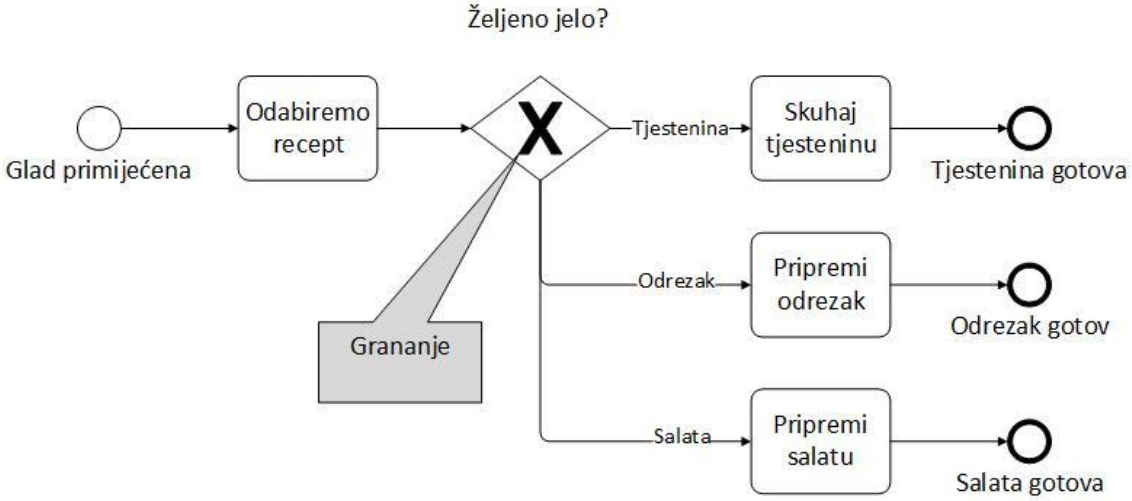
Tok aktivnosti *(engl. Sequence flow)* opisuje vremensko-logički slijed elemenata toka *(engl. Flow elements)*: zadaci *(engl. Tasks)*, događaji *(engl. Events)*, grananja/spajanja *(engl. Gateways)*. Procesni put preuzet žetonom također je jedan tok aktivnosti. Kod crtanja dijagrama je orijentacija toka proizvoljna. Doduše, najčešće orijentacija je horizontalna i ide s lijeva na desno.

2.3. Oblikovanje procesnih puteva s grananjem/spajanjem

2.3.1. Isključujuće grananje/spajanje bazirano na podacima

(engl. Data-base exclusive gateway)

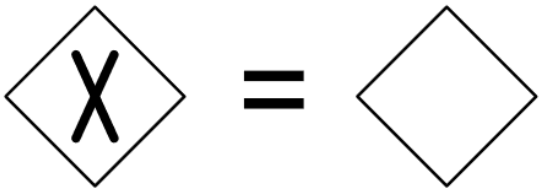
Određene stvari mogu se obavljati samo u određenim okolnostima pa se nekoliko procesa uvijek izvodi na isti način.



Slika 2.3. Isključujuće grananje

Na slici 2.3 želimo prikazati detalje našeg primjera sa slike 2.2. Sada gledamo postupak kuhanja. Potaknuti gladi razmišljamo što ćemo skuhati. Pretpostavimo da znamo napraviti samo tri jela: tjesteninu, odrezak i salatu. Pretpostavimo da od ta tri recepta odabiremo samo jedan: ***ili*** tjesteninu ***ili*** odrezak ***ili*** salatu. Tada su te opcije isključive. Točka odluke

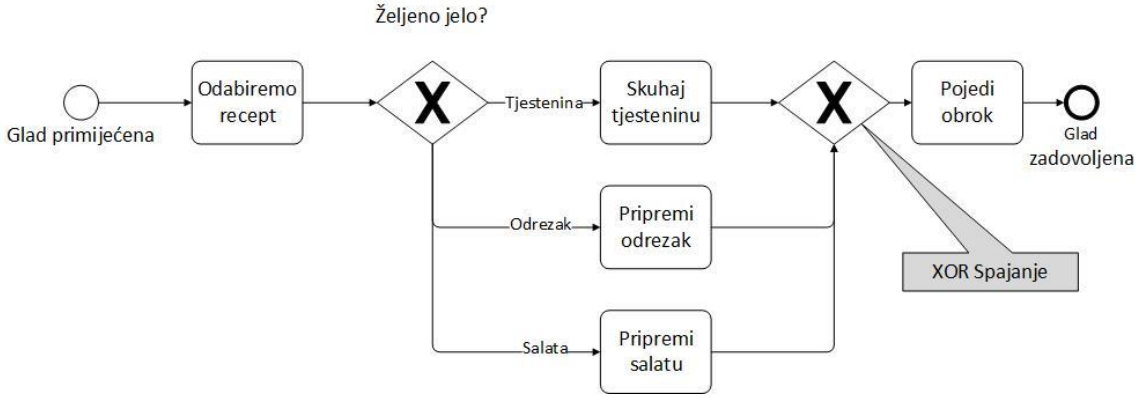
što napraviti naziva se **grananje/spajanje**, ovisno slijedi li prije ili poslije aktivnosti. Odlučujemo na temelju dostupnih podataka (u našem primjeru su to recepti) i slijedimo samo jedan put koji je isključiv (ekskluzivan). Skraćeno ga zapisujemo kao **XOR** (engleska skraćenica za *Exclusive OR*). Važno je napomenuti da grananje/spajanje nije zadatak.



Slika 2.4. oznake za XOR

Na slici 2.4. vidimo dvije oznake za XOR. U teoriji obje oznake koriste se za označivanje XOR, ali u praksi češće nailazimo na oznaku s 'X' jer jasnija.

Gledajući naš primjer možemo reći da se obrok jede neovisno o tome koji je recept odabran. U tom slučaju možemo koristiti XOR **spajanje** na način da se žetoni iz tri različita puta ujedine u jedan odlazni put. Primjer je prikazan na slici 2.5.

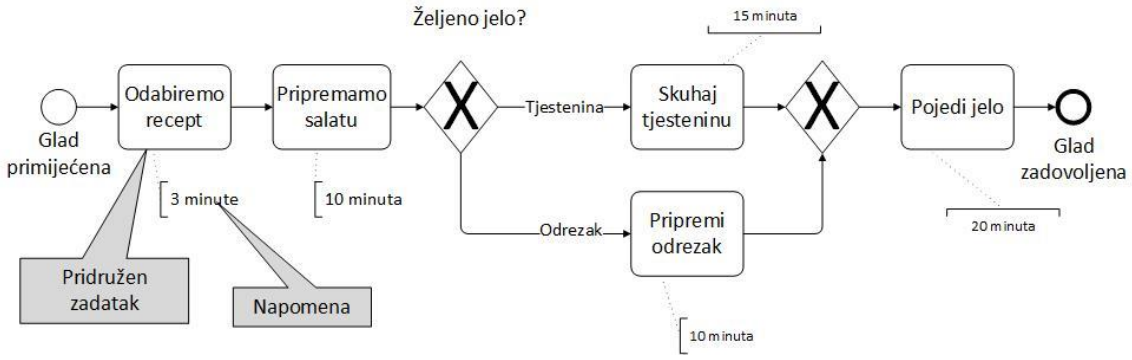


Slika 2.5. XOR spajanje

2.3.2. Paralelno grananje/spajanje

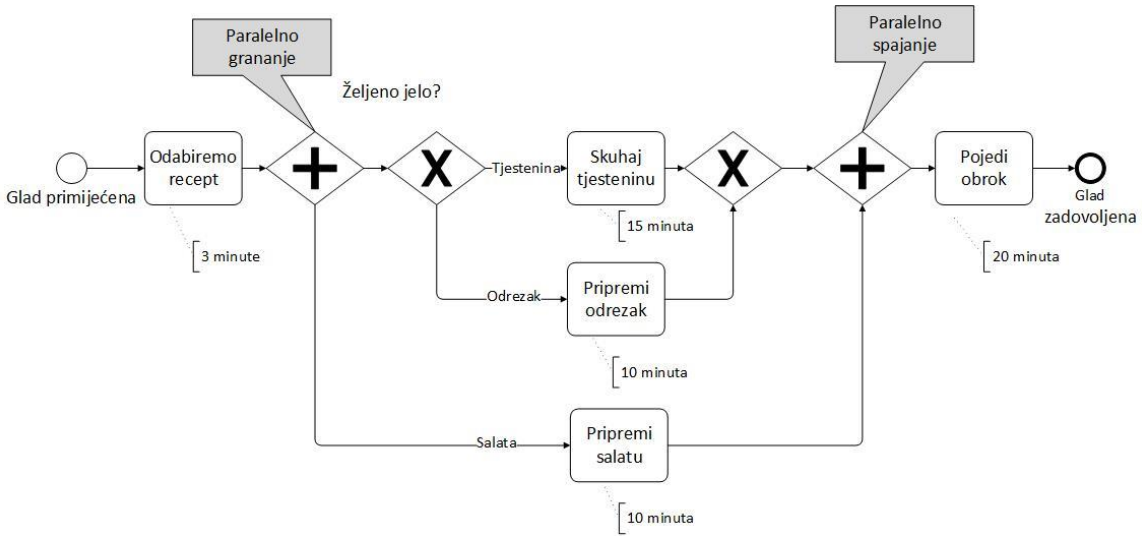
Da bismo prikazali paralelno grananje/spajanje, pretpostavimo da imamo sličan primjer, ali da želimo pokraj jela imati salatu. To znači da ćemo imati salatu neovisno o tome koje jelo odaberemo. Za potrebe tog prikaza uvodimo drukčiji simbol: napomenu *(engl. Annotation)*. Kao što možemo vidjeti na slici 1.1, pripada kategoriji artefakta kojeg

možemo povezati s bilo kojim objektom toka (u našem primjeru koji je prikazan na slici 2.6, objekt toka je zadatak). Napomena može sadržavati bilo kakav tekst. U gore opisanom primjeru, u napomenu smo stavili prosječno vrijeme za obavljanje pridruženog zadatka. Ukupno vrijeme zadatka jednako je vremenu trajanja procesa, ukupno 48 minuta za tjesteninu i 43 minute za odrezak.



Slika 2.6. Priprema salate i glavnog jela

Primijetimo da će do trenutka kada krećemo jesti proći 23 minute (u slučaju da odaberemo napraviti odrezak) ili 28 minuta (ako radimo tjesteninu). Ukoliko smo jako gladni, možemo se požuriti s pripremom jela tako da paralelno radimo salatu i glavno jelo. Za takav prikaz koristit ćemo simbol za paralelno grananje odnosno spajanje: **AND**. Prikaz je vidljiv na slici 2.7. gdje smo sa '+' označili grananje/spajanje AND.



Slika 2.7. Paralelna priprema salate i glavnog jela

Uvođenjem paralelnog grananje/spajanja, ukupno vrijeme procesa umanjili smo za 10 minuta. Zbog optimizacije trajanja procesa koriste se paralelna grananja/spajanja gdje god je to moguće.

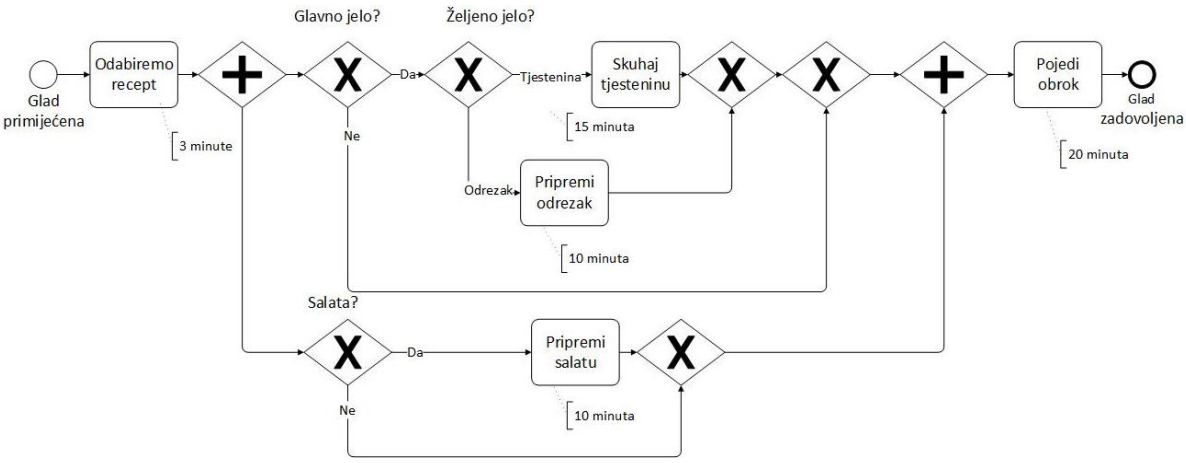
2.3.3. Uključivo grananje/spajanje bazirano na podacima

(engl. Data-base inclusive gateway)

Želimo li naš proces napraviti još fleksibilnijim, kada smo gladni, možemo jesti:

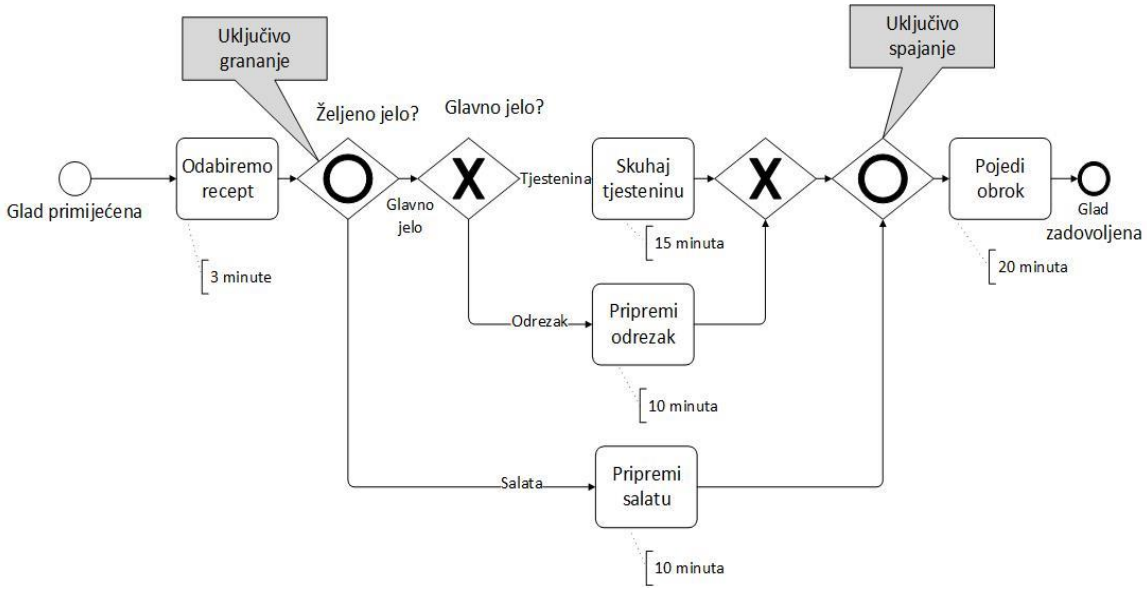
- Samo salatu
- Salatu zajedno sa glavnim jelom
- Samo glavno jelo.

Koristeći već prije prikazane simbole možemo modelirati navedenu situaciju (prikazano slikom 2.8.)



Slika 2.8. Više kombinacija jela

Ukoliko želimo kompaktniji prikaz, možemo koristiti uključivo grananje/spajanje **OR.** Njega koristimo za opisivanje **i /ili** situacije u kojima proces može teći jednim, više ili svim odlaznim putevima. Njegovim uvođenjem pojednostavljujemo prikaz. Primjer 2.8. mogli smo jednostavnije zapisati koristeći OR grananje i spajanje kao na slici 2.9.

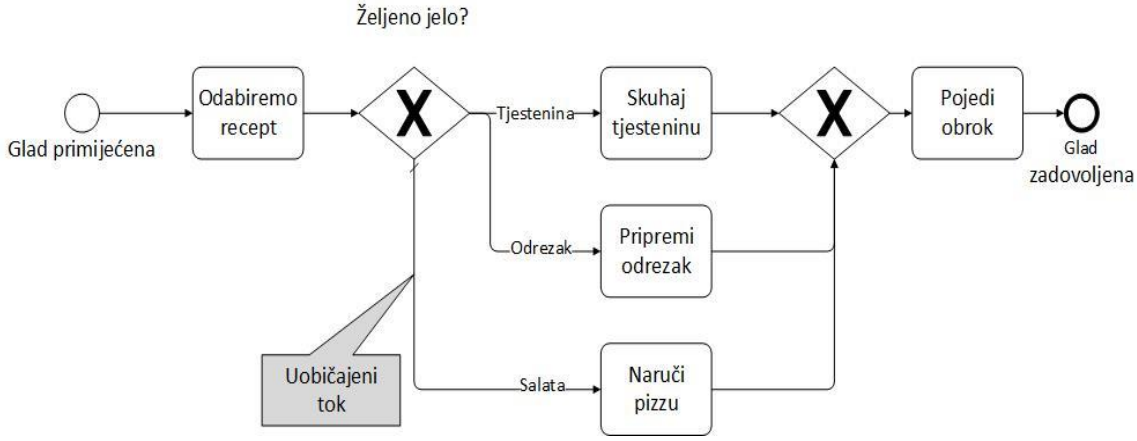


Slika 2.9. Kompaktniji prikaz slike 2.8. koristeći OR

2.3.4. Uobičajeni tok i zapinjanje

(engl. Default flow and getting stuck)

Što će se dogoditi ukoliko ne želimo niti tjesteninu niti odrezak? U prijašnjim modelima, ova situacija bi značila da žeton nikad ne bi mogao ići dalje zbog XOR grananja/spajanja. Po specifikacijama BPMN-a to bi izbacilo pogrešku*(engl. Error)*. Da bismo se zaštitili od takvih grešaka, koristimo **uobičajeni tok** *(engl. Default flow)*. Uobičajeni tok označavamo malom kosom crticom kao što je prikazano na slici 2.10.



Slika 2.10. Uobičajeni tok

Uobičajeni tok koristimo kada pri ispitivanju svih puteva nijedan nije odabran. On sprječava zapinjanje. U praksi se uobičajeni tok ne mora koristiti, ali je koristan. Češća je praksa stvoriti drugi tok aktivnosti i imenovati ga po vlastitoj želji.

2.3.5. Složeno grananje/spajanje

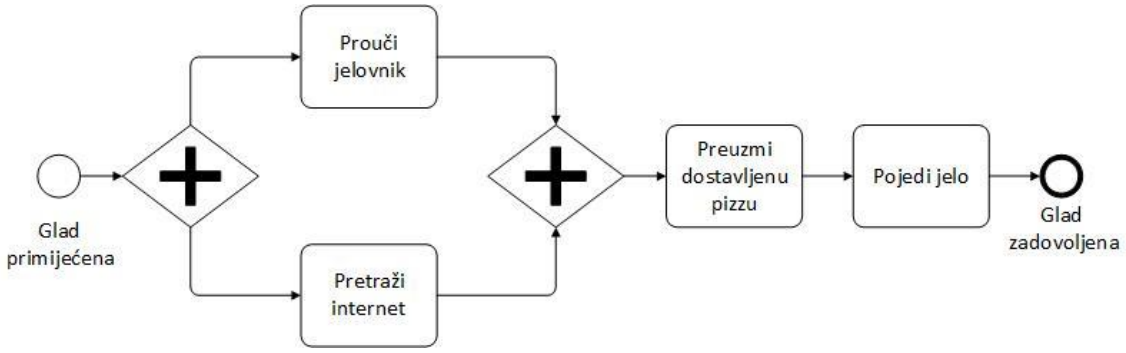
(engl. Complex gateway)

Iako se složeno grananje/spajanje ne koristi često, postoje situacije u kojima su korisni. Opišimo jedan takav primjer:

Želimo naručiti pizzu. Imamo naviku naručivati uvijek iz istog restorana, no odlučimo se na internetu potražiti još koji. Usporedimo njihove jelovnike. Nakon što se odlučimo što želimo jesti, naručujemo pizzu. U toj situaciji morali smo pregledati oba izvora (jelovnika) pa tek onda odlučiti, a ne odlučiti prije pregledavanja jelovnika.

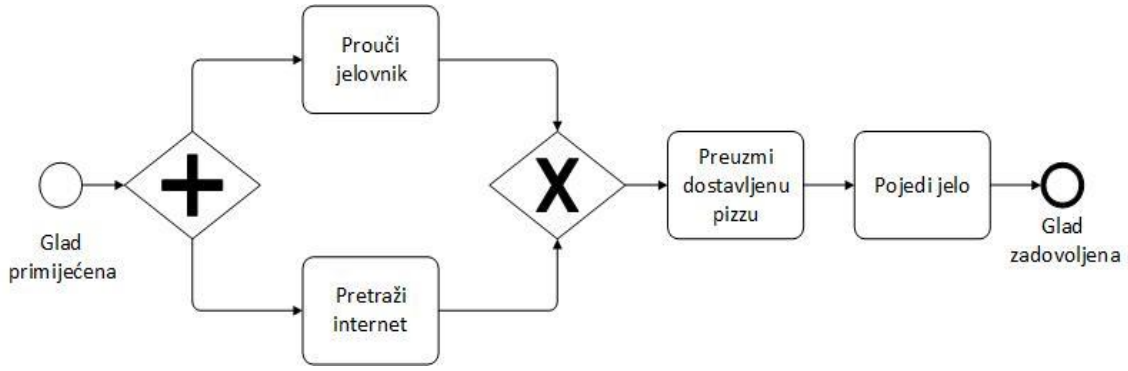
Dizajnirat ćemo 3 dijagrama koristeći grananja ili spajanja koje smo do sad upoznali: AND, XOR i OR.

Na slici 2.11. prikazan je dijagram u kojem smo koristili AND spajanje. On je rezultat naručivanje pizze samo nakon što smo oba jelovnika proučili.



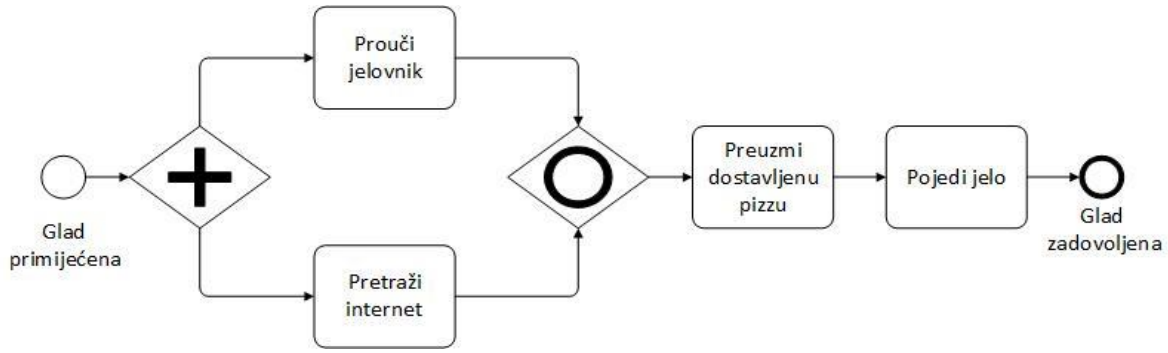
Slika 2.11. Naručivanje pizze s AND grananjem/spajanjem

Sada promotrimo dijagram na slici 2.12, s obzirom na koncept žetona, narudžba bi bila izvršena dva puta.



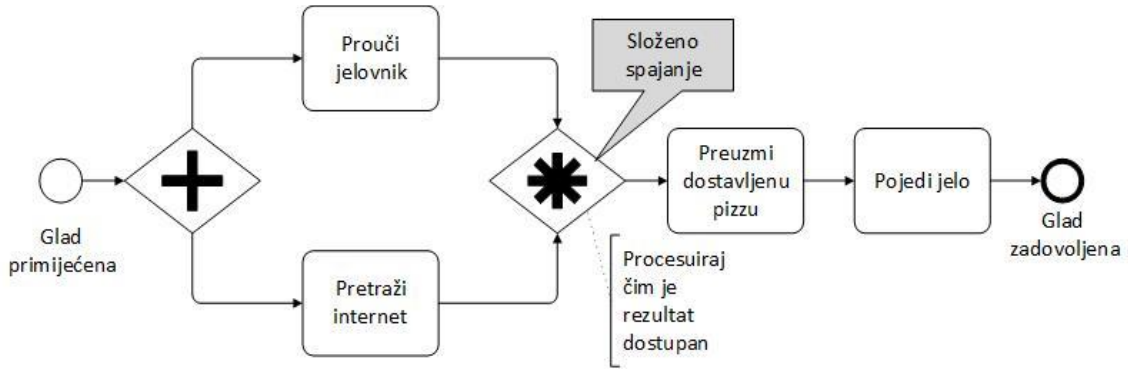
Slike 2.12. Naručivanje pizze s XOR spajanjem

Na slici 2.13. prikazan je dijagram u kojem koristimo OR grananje/spajanje. Kada žeton dođe do OR spajanja, proces čeka odgovarajući žeton koji možda nikad neće doći.



Slike 2.13. Naručivanje pizze s OR spajanjem

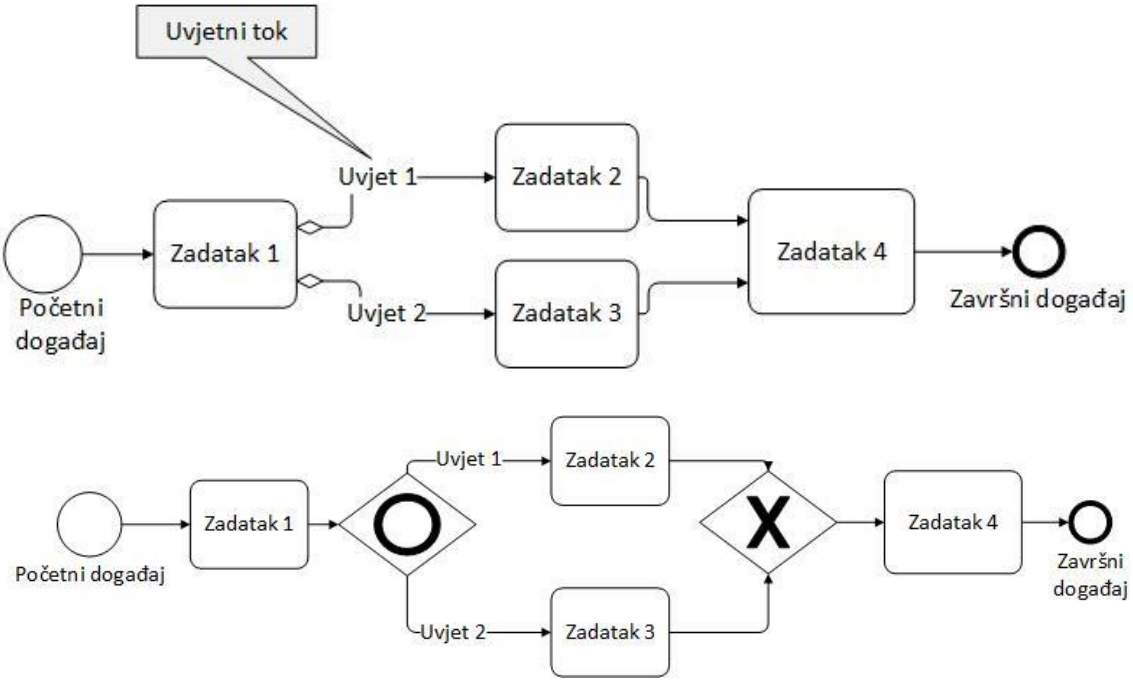
Analizirajući ta tri dijagrama, zaključili smo da nismo mogli dobiti rješenje kakvo želimo. Rješenje je u **složenom grananju/spajanju***(engl. Complex gateway)*u kombinaciji s napomenom kao što je prikazano na slici 2.14.



Slika 2.14. Naručivanje pizze s složenim spajanjem

2.4. Oblikovanje procesnih puteva bez grananja/spajanja

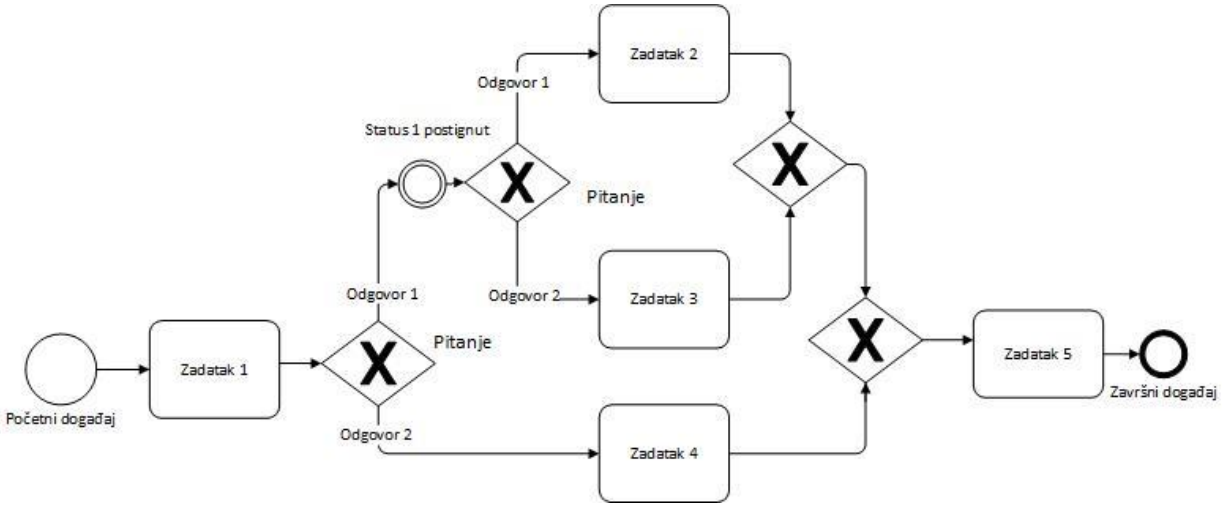
Neki ljudi ne preferiraju grananje/spajanje. Smatraju da oni kompliciraju izgled, a tako i razumijevanje dijagrama. Oni nisu nužni kod izrade modela, može se modelirati i logika grananja/spajanja XOR, AND i OR direktno sa zadacima. Ukoliko se odlučimo modelirati logiku sa zadacima, trebamo biti na oprezu.



Slika 2.15. OR razdvajanje s uvjetnim tokom i XOR spajanjem

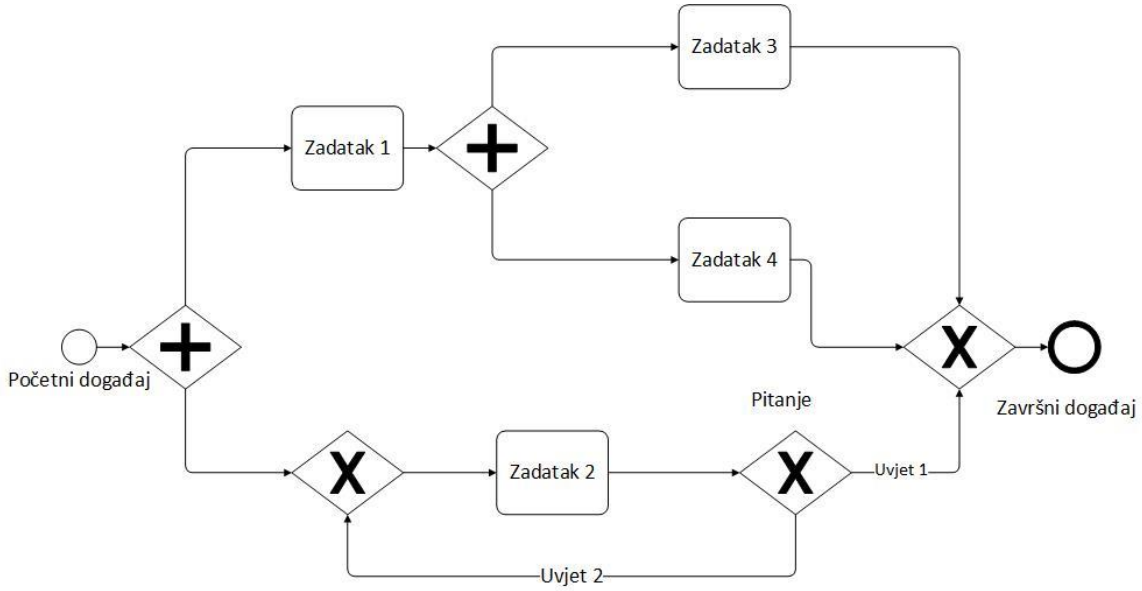
Na slici 2.15. možemo vidjeti kako smo iskoristili uvjetni tok *(engl. Conditional flow)* kao alternativu OR grananja. Donji i gornji modeli su ekvivalentni. Donji model pokazuje dva toka usmjerena izravno na Zadatak 4. Također, predstavlja OR grananje sa simbolom uvjetnog toka: mali romb koji povezuje Zadatak 1.

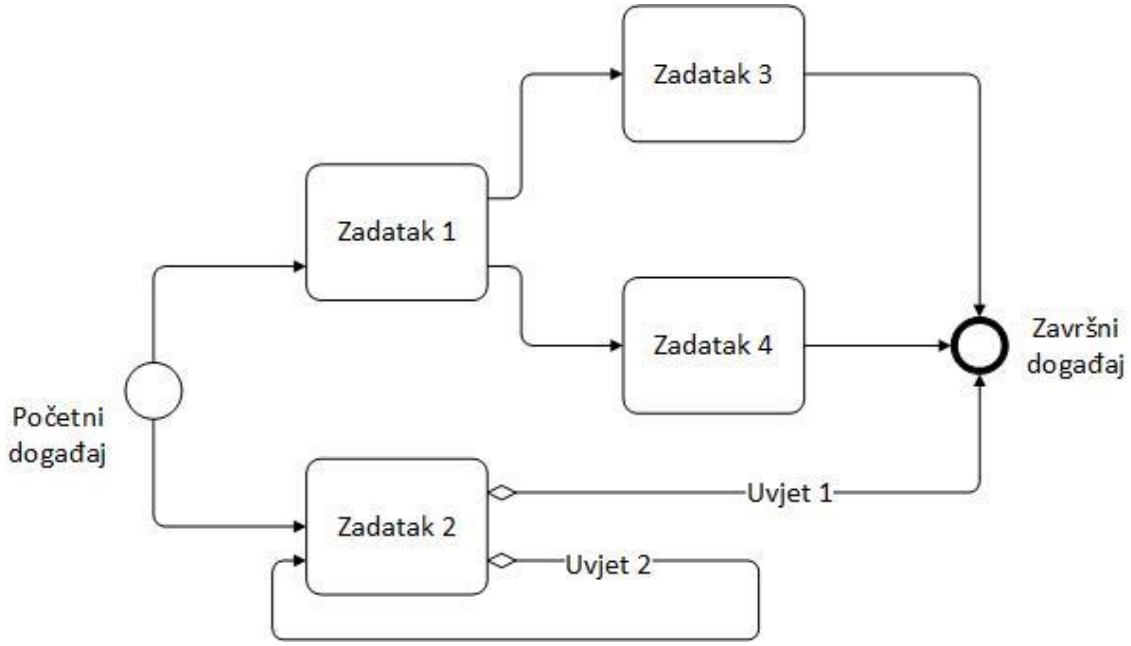
Simboli uvjetnog toka mogu se povezati samo sa zadacima ili potprocesima, i to samo kao utičnice. Ne smiju se koristiti s grananjem/spajanjem ili događajima. Iako smo uspjeli naći zamjenu za OR i XOR grananje/spajanje, ne možemo sinkronizirati model (AND spajanja) bez njih. Drugi problem izbacivanja grananja/spajanja je što ne možemo kombinirati uvjetne provjere. Na slici 2.16. prikazan je model kojeg ne možemo drugačije modelirati, a da logika ostane ista, zbog srednjeg događaja.



Slika 2.16. Kombinirano XOR grananje/spajanje

Još jedan problem je što uvjetni tok slijedi istu semantiku kao i OR grananje. Drugim riječima, definirano grananje ne smije biti isključivo. Iako bismo sad mogli zaključiti da je najbolje uvijek koristiti grananje/spajanje, to isto nije istina. Trebamo biti svjesni kako naš model izgleda nekome tko nema pretjeranog iskustva s BPMN-om. Osim što treba pratiti logiku procesa, treba što manje zbunjivati korisnika. BPMN dopušta više tokova od početnog do završnog događaja pa dijagrami mogu biti kompaktniji i pregledniji. Da bismo to prikazali, na slici 2.17. modelirali smo dijagrame sa i bez grananja/spajanja.



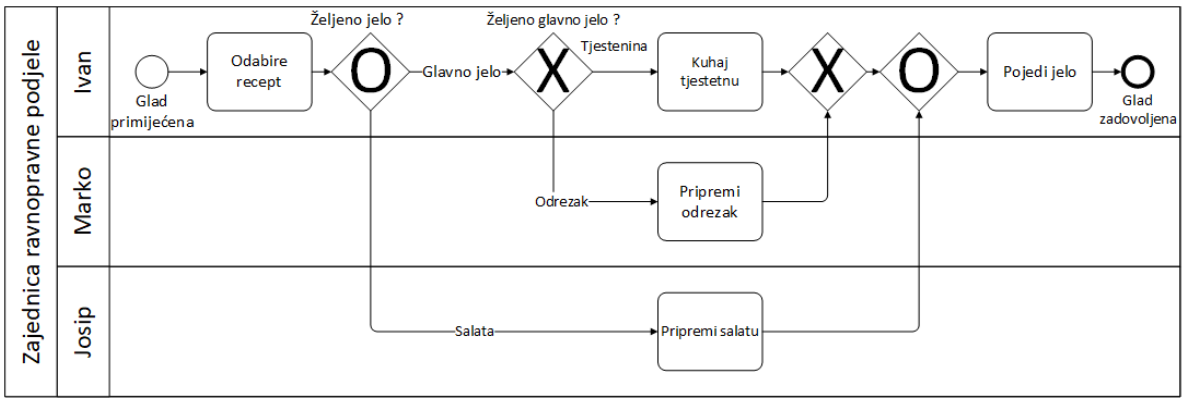


Slika 2.17. Dva modela opisuju skoro isti proces

Na slici 2.17. su dva modela koji prikazuju skoro iste procese. Razlikuju se jer u gornjem (zbog XOR grananja/spajanja) moramo proći nekoliko puteva. Zahtijeva da se uvjeti 1 i 2 nikad ne događaju istovremeno, dok se u drugom modelu uvjeti 1 i 2 mogu istovremeno događati.

2.5. Trake

U BPMN-u koristimo trake *(engl. Lanes)* kako bismo znali tko je odgovoran za izvršenje pojedinog zadatka. Da bi nam bilo malo jasnije kako trake pomažu odrediti tko je izvršitelj kojeg posla, pogledajmo sliku 2.18.



Slika 2.18. Raspodjela odgovornosti korištenjem traka

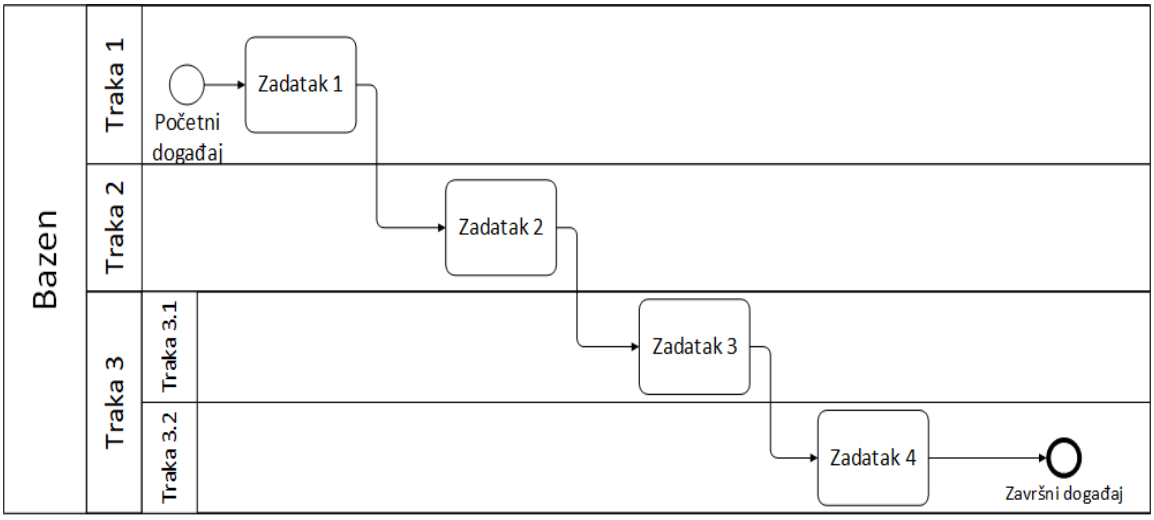
Na slici je prikazana sljedeća raspodjela:

Ako je Ivan gladan, on odabire recept kojeg želi jesti. Ovisno o tome što on odabere, može kuhati samo za sebe ili mu cimeri, Marko i Josip, mogu pomoći u pripremi. Ako će mu cimeri pomoći, Marko će pripremiti odrezak, a Robert salatu. Na kraju, Ivan jede. Tri trake (Ivan, Marko, Josip) ujedinjene su u jednom bazenu dizajniranom kao zajednica ravnopravne podjele *(engl. Flat-sharing community)*.

U našem primjeru trake su raspoređene ljudima, ali to nije pravilo. Trake možemo dodjeljivati sukladno našim željama. U praksi trake često dodjeljujemo:

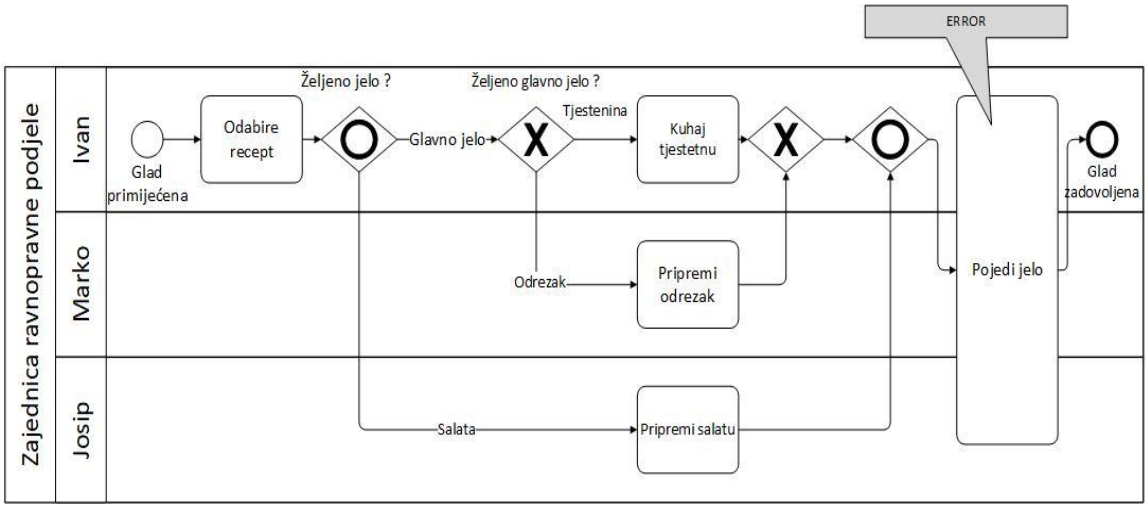
- Pozicijama u primarnoj organizaciji, npr. računovođa
- Ulogama u sekundarnoj organizaciji, npr. službenik za zaštitu podataka
- Opće uloge, npr. kupac
- Odjelima, npr. odjel prodaje
- IT aplikacijama, npr. CRM sustav

Također, imamo i trake pod trakom koje označavaju profinjenu odgovornost. Jedna takva ilustracija prikazana je na slici 2.19.

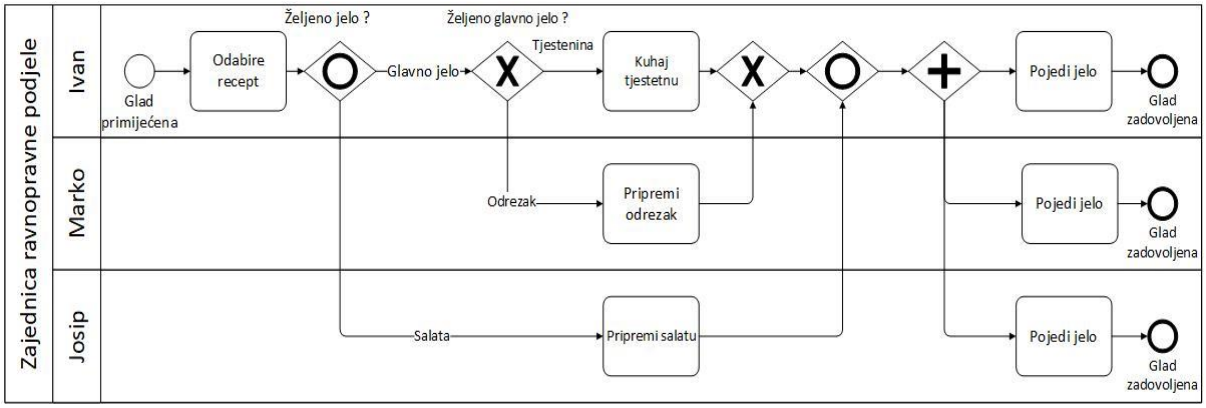


Slika 2.19. Trake pod trakom

Sjetimo se sad našeg primjera sa Ivanom, Markom i Josip. Zamislimo situaciju u kojoj i Marko i Ivan žele jesti. U BPMN-u ne smijemo imati objekt toka (aktivnost, događaj, grananje/spajanje) smješten izvan jedne trake. Rješenje za našu situaciju je duplicirati zadatak „Pojedi jelo“ i svakome dodijeliti po jedan taj zadatak. Zabranjena situacija (objekt toka preko više traka) prikazan je na slici 2.20, dok je ispravan način s dupliciranjem zadatka prikazana je na slici 2.21.



Slika 2.20. Neispravno rukovanje trakom



Slika 2.21. Ispravno rukovanje trakom

2.6. Događaji

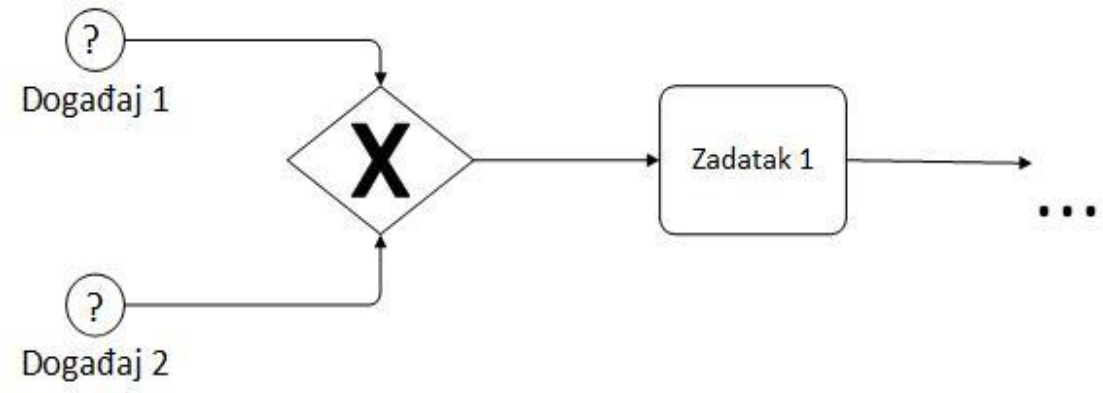
2.6.1. Važnost događaja u BPMN

Znamo da se stvari (zadaci) moraju dogoditi pod određenim uvjetima (grananje ili spajanje). Ono što nedostaje su stvari (događaji) koje se trebaju dogoditi. Do sad smo spomenuli događaje hvatanja i bacanja te početne, srednje i završne događaje.

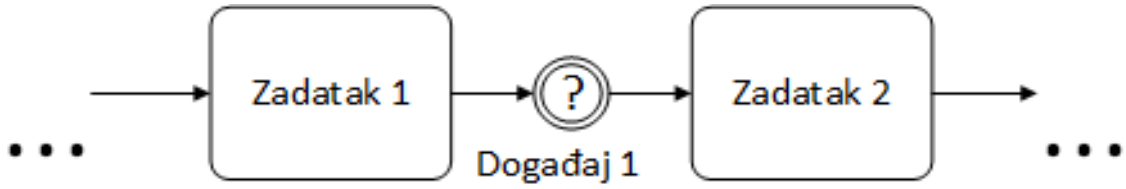
Događaji hvatanja su događaji koji imaju definiran okidač koji ga pokreće. Smatramo da se dogode nakon što se okidač pokrene ili otkaže. Ti događaji utječu na tijek procesa, stoga moraju biti modelirani. Za razliku od događaja bacanja, događaji hvatanja ne reagiraju na okidače, već sami sebe pokreću. Kažemo da su oni aktivni u usporedbi s pasivnim događajima hvatanja. Početni događaji su događaji koji se uvijek javljaju na početku. Proces ne može pokrenuti događaj prije nego što je počeo; tek kada se događaj izvrši, proces započinje. Na slici 2.22. ilustrirano je kako proces počinje čim se prvi događaj dogodi.

Slika 2.22. Čim se Događaj 1 dogodi, proces počinje

Upitnik unutar Događaja 1 označava da se događaj može dodijeliti određenoj vrsti. Drugačiji događaji mogu pokrenuti proces što se može modelirati kao na slici 2.23. Važno je da svaki događaj pokreće vlastitu procesnu instancu.



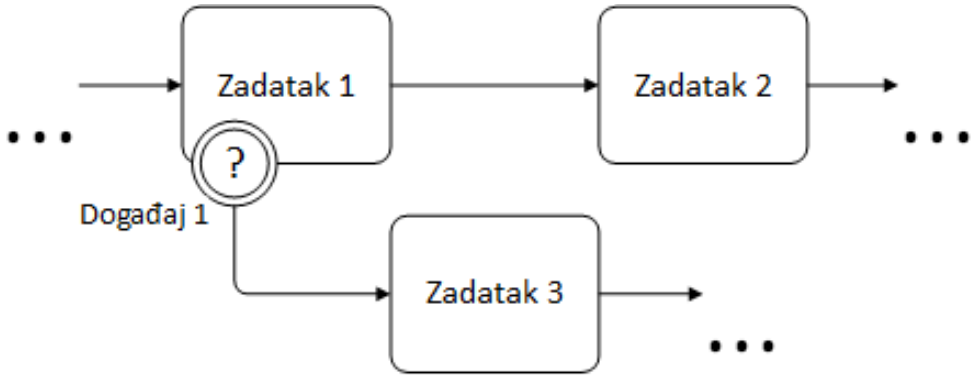
Slika 2.23. Nakon što se dogodi događaj 1 ili 2, proces započinje



Slika 2.24. Nakon Zadatka 1, proces čeka da se dogodi Događaj 1 i tek tada procesira Zadatak 2

Pretpostavimo da imamo srednji događaj. Proces može čekati nakon određenog zadatka da se događaj dogodi kao na slici 2.24 ili postoje događaji koji ne zahtijevaju eksplicitno čekanje, ali prekidaju naše aktivnosti, zadatke i potprocese. Takve srednje događaje zovemo **pridruženi događaji** *(engl. Attached events)* jer ih postavljamo na samu granicu aktivnosti koju želimo prekinuti. Žeton koji putuje kroz proces prikazan na slici 2.25 ponašao bi se na sljedeći način:

- Žeton dolazi do Zadatka 1 koji počinje u skladu s njim
- Ukoliko se Događaj 1 dogodi dok se Zadatak 1 obrađuje, zadatak se odmah prekida i žeton odlazi kroz tok iznimke *(engl. Exception flow)* do Zadatka 3
- S druge strane, ukoliko se Događaj 1 ne dogodi, obradit će se Zadatak 1, žeton se pomiče kroz redoviti tok aktivnosti do Zadatka 2
- Ako se Događaj 1 dogodi tek nakon što se završi Zadatak1, prestaje važiti



Slika 2.25. Događaj 1 zaustavlja Zadatak 1 i počinje Zadatak 3

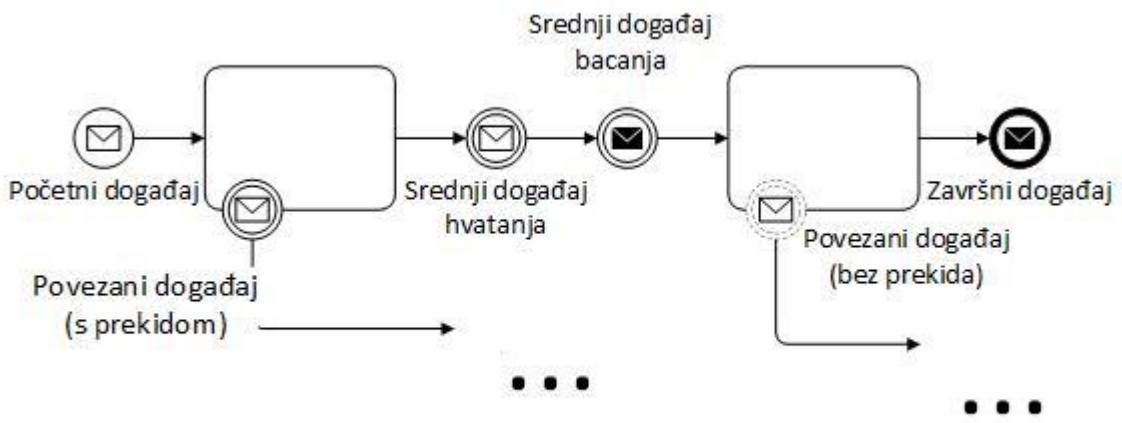
Pridruženi događaji ne moraju rezultirati otkazanim aktivnostima zbog neprekidnog srednjeg događaja.

Srednji događaji bacanja pokreću se od strane procesa. To znači da žeton koji se pojavio na takvom okidaču procesa odmah kreće dalje. Događaji bacanja ne dovode do otkazivanja aktivnosti i zbog toga nikad nisu pridruženi. Oni se pojavljuju samo u slijedu toka. Već smo ranije spomenuli srednje obične događaje koji se mogu koristiti za modeliranje ulaza u definiran status. Oni su također događaji bacanja.

2.6.2. Događaji poruke

(engl. Message events)

Prije ili kasnije, većina procesa zahtijeva neku komunikaciju koja se može prikazati kao događaj poruke u BPMN-u. Taj događaj ima simbol male omotnice unutar simbola događaja. Općenito, kako se koristi događaj poruke prikazano je na slici 2.26.



Slika 2.26. Dodavanje događaja poruke

Značenje poruke unutar BPMN-a nije nužno pismo, e-mail ili poziv. Porukom smatramo svaku radnju koja se odnosi na određenog primatelja i predstavlja ili sadrži neke podatke.

2.6.3. Vremenski događaji

(engl. Time events)

Vremenski događaji često se koriste prilikom rada s BPMN-om jer su jednostavni za primjenu te imaju više funkcija. Mogu se koristiti za pokretanje procesa:

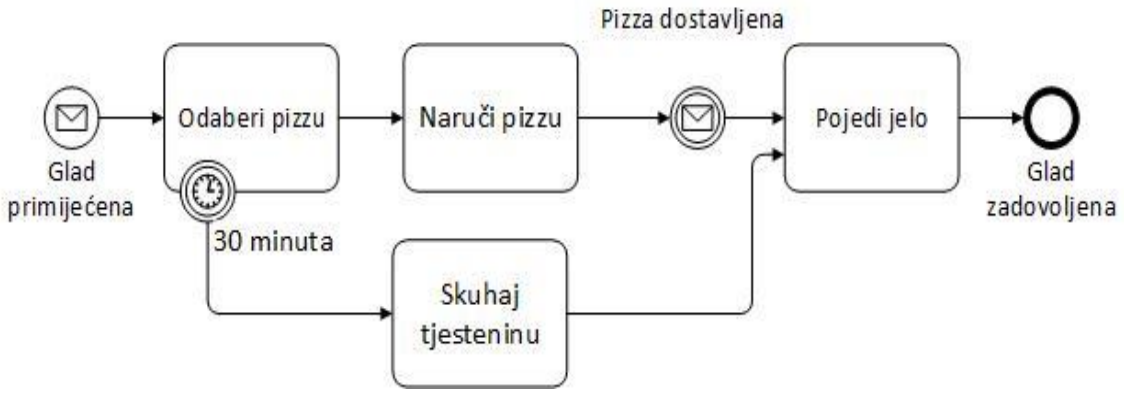
- Jednom, u točno određenom trenutku

- Kroz ponavljanje, u točno određenom trenutku
- U nekom vremenskom intervalu

Također, možemo ga koristiti kao srednji događaj. Vremenski događaj može zaustaviti proces sve dok:

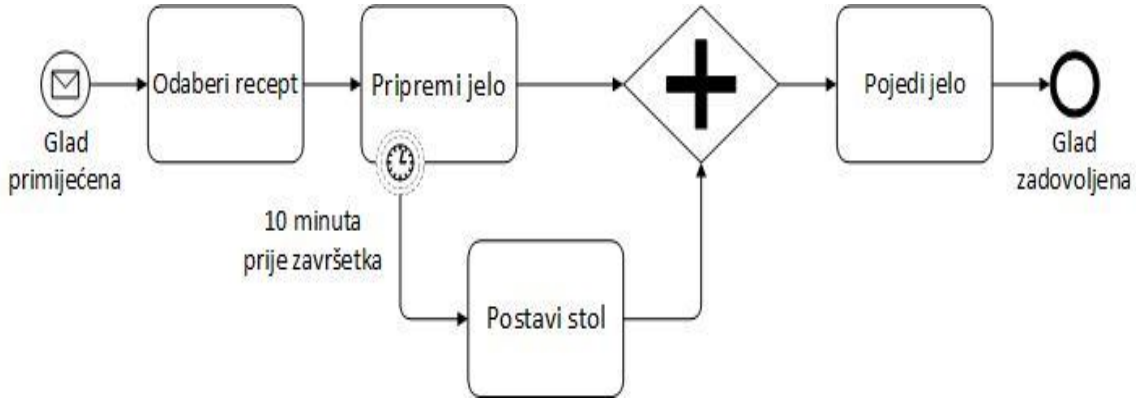
- Ne stigne zakazan trenutak
- Ne završi određeni interval
- Istekne vrijeme povezano za drugi događaj

Općenito, vrijeme se kreće bez obzira na to što mi ili naš proces radi tako da vremenski događaji mogu postojati samo kao događaji hvatanja ili srednji događaji. Također, mogu se modelirati kao odbrojavanje s pridruženim događajem sata. Takva primjena je česta u BPMN-u. Može se odrediti i gornja granica vremena odnosno maksimalno dopušteno vrijeme za obradu određenog zadatka. Takva primjena je prikazana na slici 2.27 u kojoj smo ograničili trajanje zadatka biranja pizze na maksimalno 30 minuta. Ukoliko zadatak nije izvršen u zadanom vremenskom okviru, odnosno ako nismo odabrali pizzu unutar 30 minuta ,otkazujemo zadatak „Odaberi pizzu“ i prelazimo na zadatak „Skuhaj tjesteninu“. Primijetimo, ovdje smo koristili i događaj poruke kao povratnu informaciju da je pizza dostavljena.



Slika 2.27. Odbrojavanje za zadatak „Odaberi pizzu“ je 30 minuta

Na slici smo vidjeli primjer kako možemo otkazati događaj s odbrojavanjem, ali možemo koristiti vremenski događaj tako da ne otkazujemo događaj, nego umjesto njega generiramo dodatni žeton. Primjer takvog modela prikazan je na slici 2.28.



Slika 2.28. Vremenski događaj ne dovodi do otkazivanja događaja,

već se generira novi žeton

Slika 2.28. prikazuje proces u kojem moramo pripremiti jelo i postaviti stol, ali možemo početi postavljati stol samo 10 minuta prije nego što je jelo gotovo.

2.6.4. Događaji pogreške

(engl. Error events)

Dok modeliramo proces, moramo imati na umu i moguće pogreške. Neželjene pogreške mogu se spriječiti identifikacijom potencijalnih pogrešaka. U BPMN-u, događaji pogreške označavaju se kao simbol munje. Primjer njegove primjene vidljiv je na slici 2.29

Slika 2.29. Primjena događaja pogreške

BPMN specifikacije nemaju unaprijed definirano što bi sve pogreška mogla biti. Ne možemo generalno reći što su pogreške, to ovisi o svakom procesu zasebno. Sam dizajner moram biti svjestan mogućih grešaka unutar procesa.

2.6.5. Uvjeti

Ponekad želimo da proces počne ili da se nastavi samo ako je određeni uvjet ispunjen. Za uvjet možemo staviti što god želimo, ali uvjeti moraju biti neovisni o procesu te se tako pojavljuju samo kao događaji hvatanja.

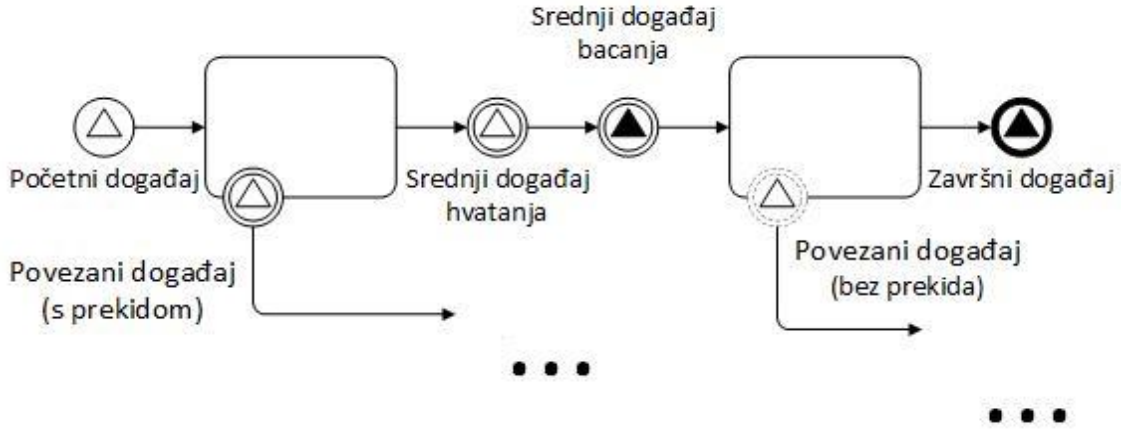
Da bismo pobliže objasnili kako se uvjeti mogu dodati u model procesa, pogledajmo sljedeći primjer. Uzimamo smrznutu pizzu iz hladnjaka i palimo pećnicu. Pizzu nećemo stavljati u pećnicu sve dok ona ne dostigne određenu temperaturu, u našem slučaju 180 Celzijevih stupnjeva. Nakon što smo je stavili u pećnicu, izvadimo je tek nakon što je gotova i tek onda je možemo pojesti. Ovaj primjer ilustriran je na slici 2.30.

Slika 2.30. Dodavanje uvjeta u model procesa

2.6.6. Signalni događaji

(engl. Signal events)

Signalni događaji su slični kao događaji poruka; iz tog razloga se mogu modelirati na jednak način kao i događaji poruka. Iako su slični porukama, postoji bitna razlika među njima: poruke su upućene određenom primatelju, dok signali nisu usmjereni prema nikome specifično. Svatko tko primi signal može reagirati na njega. Na slici 2.31 ilustrirano je dodavanje signalnog događaja.

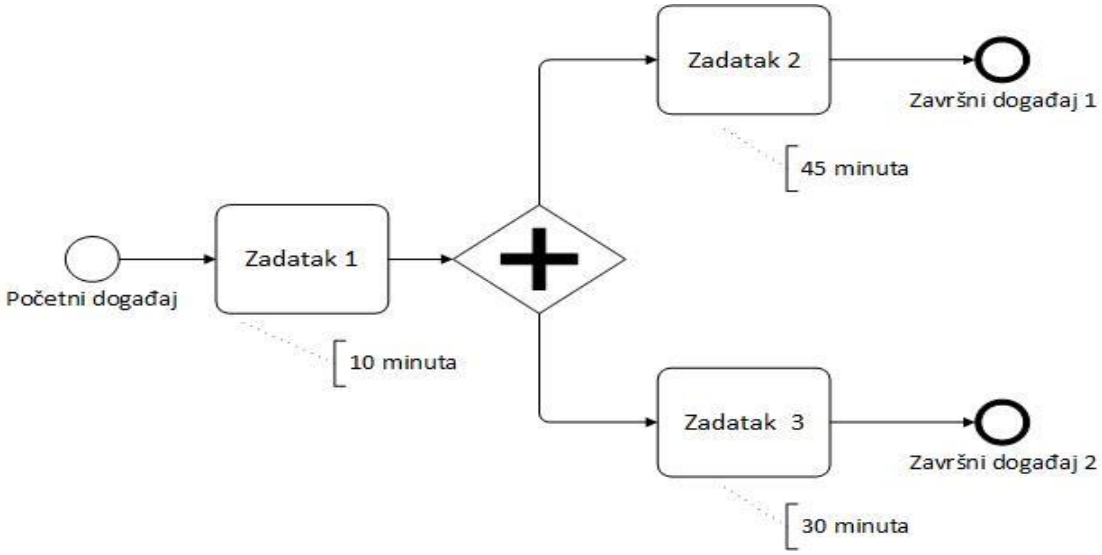


Slika 2.31. Dodavanje signalnog događaja

2.6.7. Događaji prekidanja

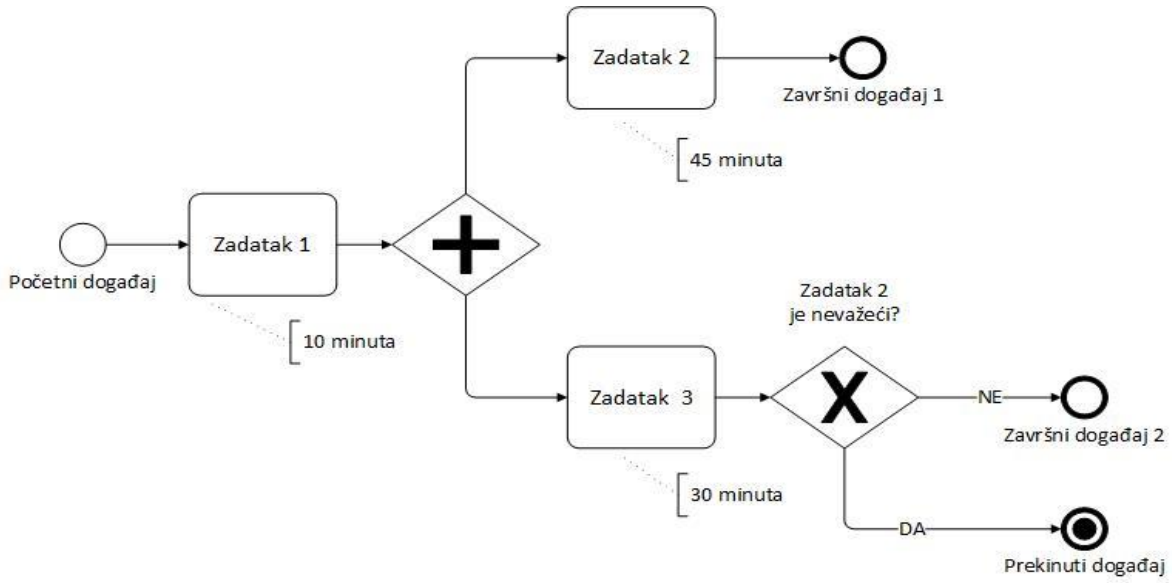
(engl. Terminate events)

Promotrimo prvo sliku 2.32. Na slici je prikazan model procesa kojemu je uvijek potrebno 55 minuta kako bi izvršio proces do kraja. Nakon što proces završi sa Zadatkom 1, Zadatak 2 i 3 mogu se izvršiti istodobno. S obzirom na duže trajanje izvršenja Zadatka 2, on određuje koliko je trajanje cijelog procesa. Naime, žeton koji prolazi procesom kloniran je kroz AND grananje. Prvi žeton se zadržava na Zadatku 2 45 minuta dok je drugi na Zadatku 3 30minuta. Nakon što drugi žeton dođe do običnog događaja- Završni događaj 2. nakon 15 minuta, prvi žeton dolazi do običnog događaja Završni događaj1. Zato što nema više dostupnih žetona, instanca procesa završava za ukupno 55 minuta koliko je trebalo prvom žetonu da završi.



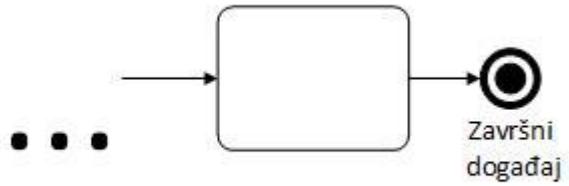
Slika 2.32. Trajanje procesa uvijek traje 55 minuta

Iako nema pogreške u našem modelu, želimo ga optimizirati i izbjeći dodatno trošenje vremena, u našem slučaju 15 minuta, na Zadatak 2 koji je nakon Zadatka 3 postao suvišan. U slučaju zaustavljanja procesa koristit ćemo prekinuti događaj. Njegovo korištenje možemo vidjeti na slici 2.33.



Slika 2.33 Proces staje odmah nakon završetka Zadatka 3

Prekinute događaje koristimo kako bismo osigurali da se svi dostupni žetoni odmah završe. Iz tog razloga dolazi do prekidanja procesne instance tako da se prekinuti događaj može pojaviti samo kao završni događaj kao što je prikazano na slici 2.34.



Slika 2.34. Dodavanje prekinutog događaja

2.6.8. Događaji povezivanja

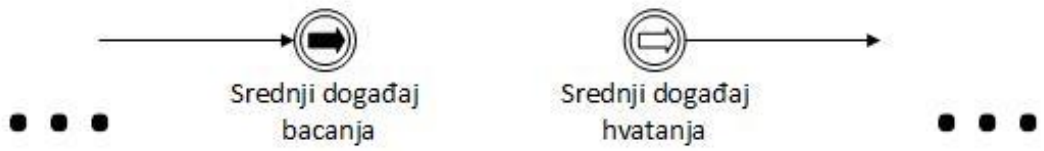
(engl. Link events)

Događaji povezivanja su posebna vrsta događaja jer ne mijenjaju sami proces. Oni ne utječu direktno na sadržaj, ali pomažu kod stvaranja procesa. Možemo ih koristiti kao alternativu za tok aktivnosti tako da napravimo dvije povezane veze gdje je prva ulazna točka *(engl. Entrance point)*, a druga izlazna točka*(engl. Exit point)* i one uvijek idu u paru.

Događaji povezivanja mogu biti korisni kod idućih situacija:

- Zbog nedovoljnog mjesta procesni dijagram trebamo razdvojiti na više stranica. Da ne bude zbunjujuće gdje smo stali i gdje počinjemo, koristimo događaje povezivanja koji su indikatori
- Imamo sveobuhvatne procesne dijagrame s puno slijeda tokova pa da bude čitljivije

Dodavanje događaja povezivanja je vidljivo na slici 2.35.



Slika 2.35. Dodavanje događaja povezivanja

2.6.9. Događaji naknade

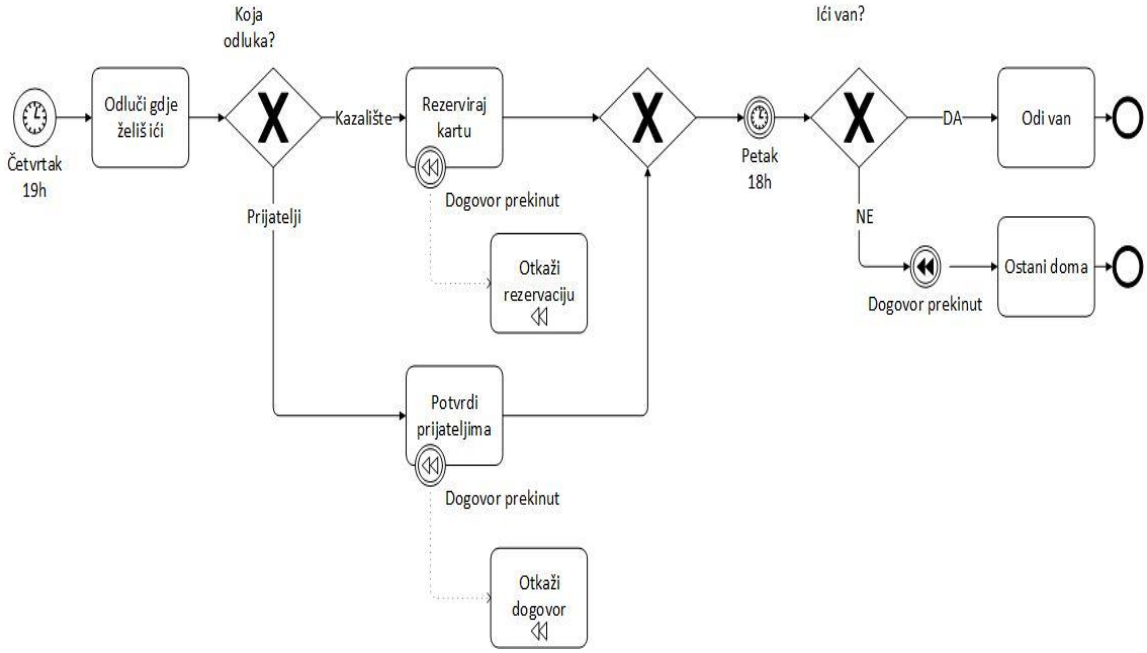
(engl. Compensation events)

Događaje naknade koristimo najčešće za transakcije jer u procesu izvršavamo zadatke koji ponekad moraju biti otkazani pod određenim uvjetima. Neki od tipičnih primjena događaja naknade su razne rezervacije (npr. vlaka ili avionske karte), punjenje kreditne kartice itd.

Postoje posebna pravila kod korištenja događaja nadoknada:

- Događaji bacanja naknade se odnose na vlastite procese tako da utječu samo unutar bazena. Zbog toga se ova vrsta događaja razlikuje od događaja bacanja poruke.
- Ostali pridruženi događaji mogu utjecati samo dok su aktivnosti kojima su pridruženi i dalje aktivne. Suprotno tome, pridružena naknada djeluje samo ako proces sam pokrene naknadu i ako je aktivnost na koju je pridružen uspješno završena.
- Događaj pridružene naknade se povezuje sa zadatkom naknade kroz veze, a ne kroz tok aktivnosti.

Zamislimo sada situaciju da je početak tjedna i trebamo se dogovoriti gdje ćemo ići ovaj vikend. Imamo dvije opcije; ići u kino s partnerom ili ići van s prijateljima. Koju god opciju odaberemo, moramo se do četvrtka navečer odlučiti i rezervirati kartu ili potvrditi prijateljima da dolazimo. Kako prolazi tjedan, shvaćamo da postoji opcija da u petak otkažemo sve i provedemo večer doma pred televizorom. Slijedeću situaciju možemo modelirati koristeći događaj naknade kako je prikazano na slici 2.36. Koristeći događaj naknade, ne moramo dodatno provjeravati koji smo događaj potvrdili da bismo ga otkazali.



Slika 2.36. Dodavanje događaja naknade

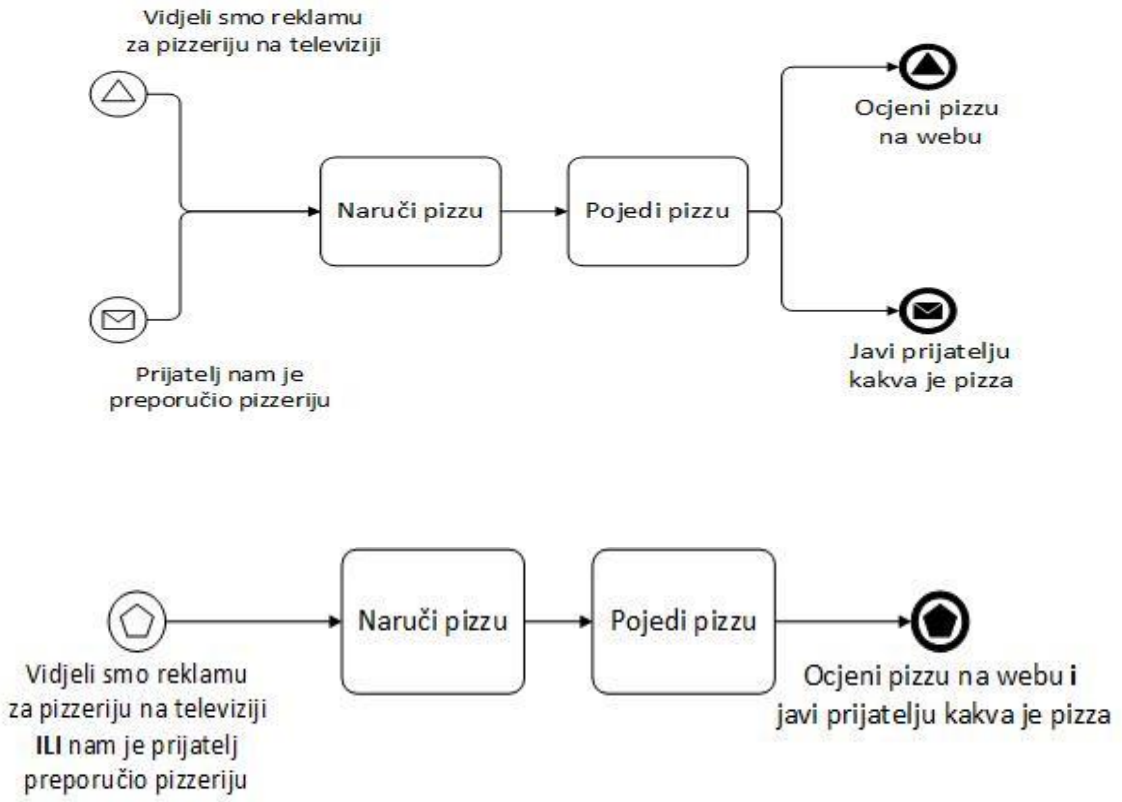
2.6.10. Višestruki događaji

(engl. Multiple events)

Višestruke događaje možemo koristiti kako bismo saželi nekoliko događaja s jednim simbolom. Semantika je jednostavna:

- Ako modeliramo višestruki događaj kao događaj hvatanja, samo jedan od svih sažetih događaja se mora dogoditi da bi počeli ili nastavili proces ili zaustavili zadatak.
- Ako pak modeliramo višestruki događaj kao događaj bacanja, svi događaji koji su sažeti moraju se dogoditi.

Na slici 2.37. vidimo dva modela istog procesa. Na donjoj slici koristili smo višestruki događaj umjesto dva zasebna događaja.

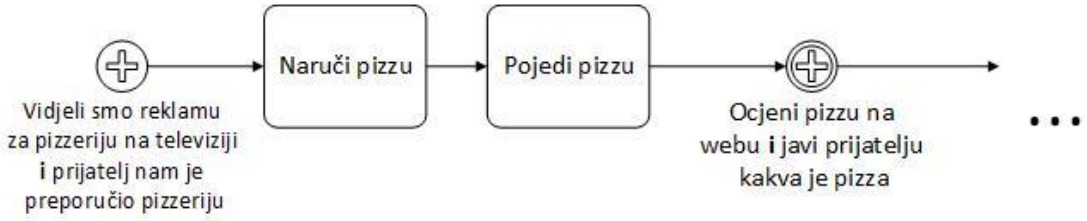


Slika 2.37. Dodavanje višestrukih događaja

2.6.11. Paralelni događaji

(engl. Parallel events)

Paralelni događaji nadopunjuju višestruke događaje. Dok višestruki događaji hvatanja imaju XOR semantiku, paralelni koriste AND semantiku. Ne dogodi se sve dok se svi događaji koji su sadržani u višestrukom događaju ne dogode. Pošto višestruki događaji bacanja imaju AND semantiku, paralelne događaje definiramo samo kao događaje hvatanja. Na slici 2.38. je prikazan model koji koristi paralelni događaj. U ovom slučaju ćemo naručiti pizzu samo ako smo vidjeli reklamu za pizzu i sjetili se da nam je prijatelj preporučio tu pizzeriju.

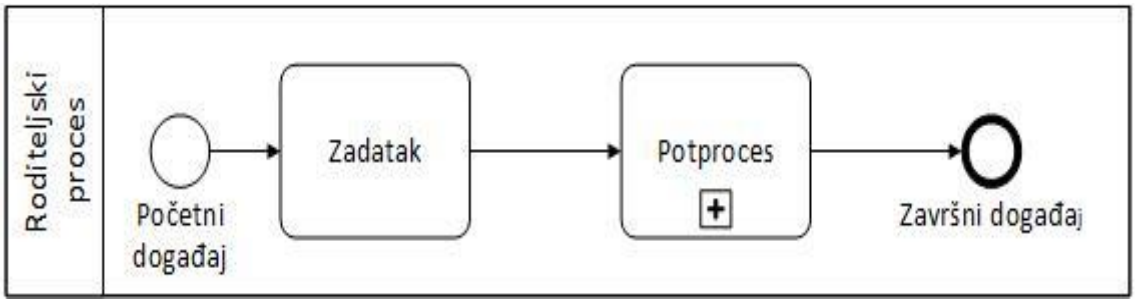


Slika 2.38. Paralelni događaji

2.7. Potprocesi

(engl. Subprocesses)

Ukoliko imamo neki kompliciraniji proces, neće nam sve stati na jedan dijagram. Da bismo vizualno pojednostavnili dijagram, koristimo potprocese. Oni nam pomažu napraviti kompaktne dijagrame. Na slici 2.39. prikazan je jedan potproces.



Slika 2.39. Dodavanje potprocesa

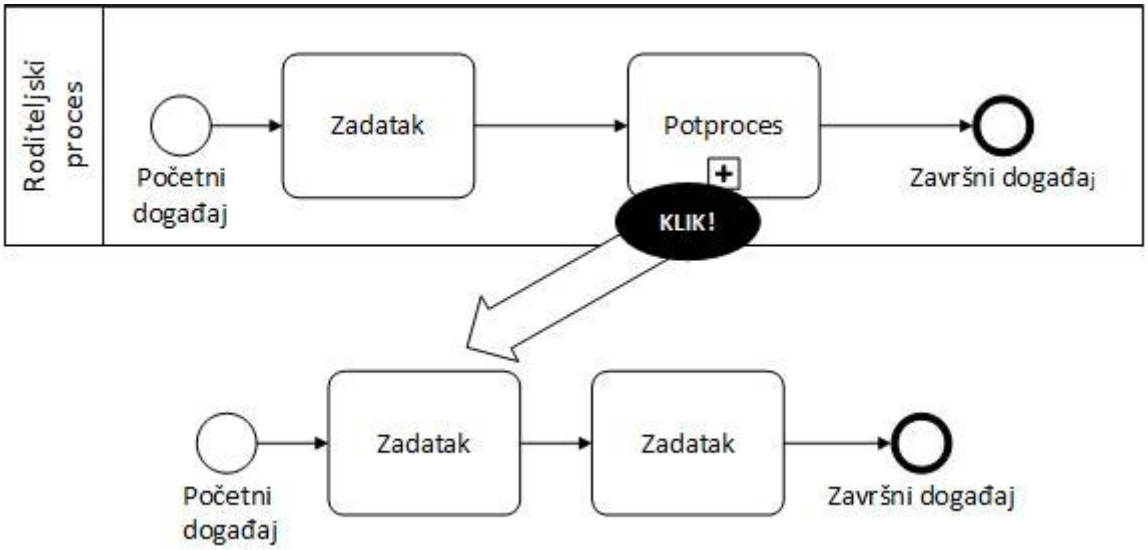
Postoje dva načina za povezivanje potprocesa s njegovim glavnim procesom kojeg još nazivamo i roditeljskim procesom:

- Prikaz u posebnom procesnom dijagramu
- Proširenje roditeljskog procesnog dijagrama

U oba slučaja važno je započeti tok aktivnosti roditeljskog procesa na lijevom rubu a završiti na desnom rubu potprocesa.

2.7.1. Potproces u posebnom dijagramu

Unutar roditeljskog procesa, simbol potprocesa povezan je s tim potprocesom. Model možemo zamisliti kao web preglednik na kojem se nalazi simbol potprocesa i pritiskom na njega otvara se nova stranica preglednika u kojoj je modeliran potproces. Tu ilustraciju prikazuje slika 2.40.

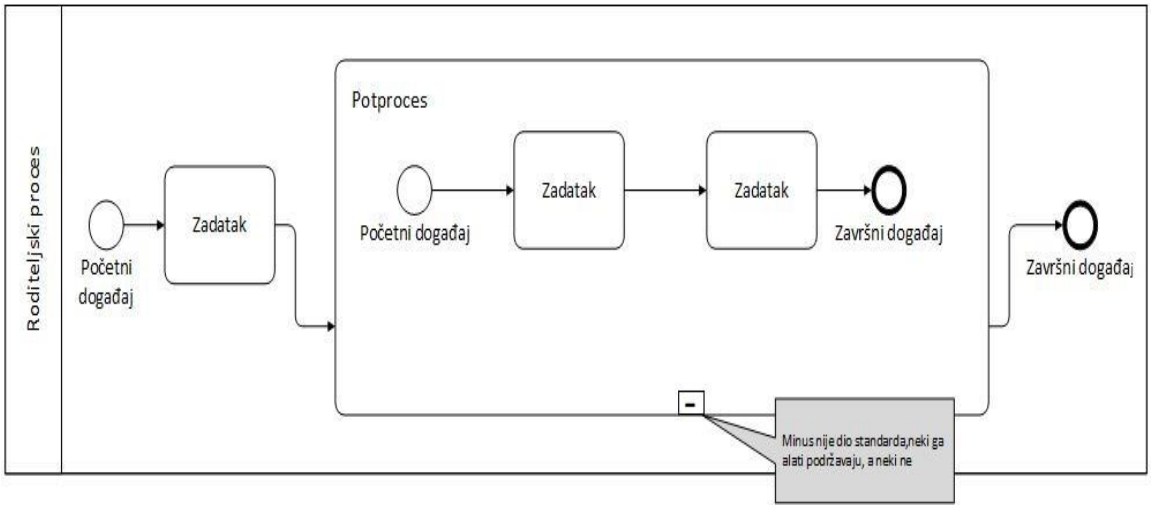


Slika 2.40. Potproces prikazan u posebnom dijagramu

2.7.2. Potproces kao proširenje roditeljskog procesa

Potproces se, kao proširenje njegovog roditeljskog procesa, označava kao aktivnost s malim „+“ koju zovemo sažeti potproces *(engl. Collapsed process).* Oznaku '+' može se pritiskom proširiti i tako prikazati potproces što doprinosi boljoj preglednosti. Iako BPMN podržava ovu opciju, ona nije implementirana u svim alatima koji koriste BPMN. Slika 2.41. prikazuje kako se potproces izravno proširio na roditeljskom dijagramu. Ukoliko

imate alat koji podržava tu opciju, omogućeno je proširenje i sažimanje izravno unutar dijagrama kako biste prikazali ili sakrili informacije.

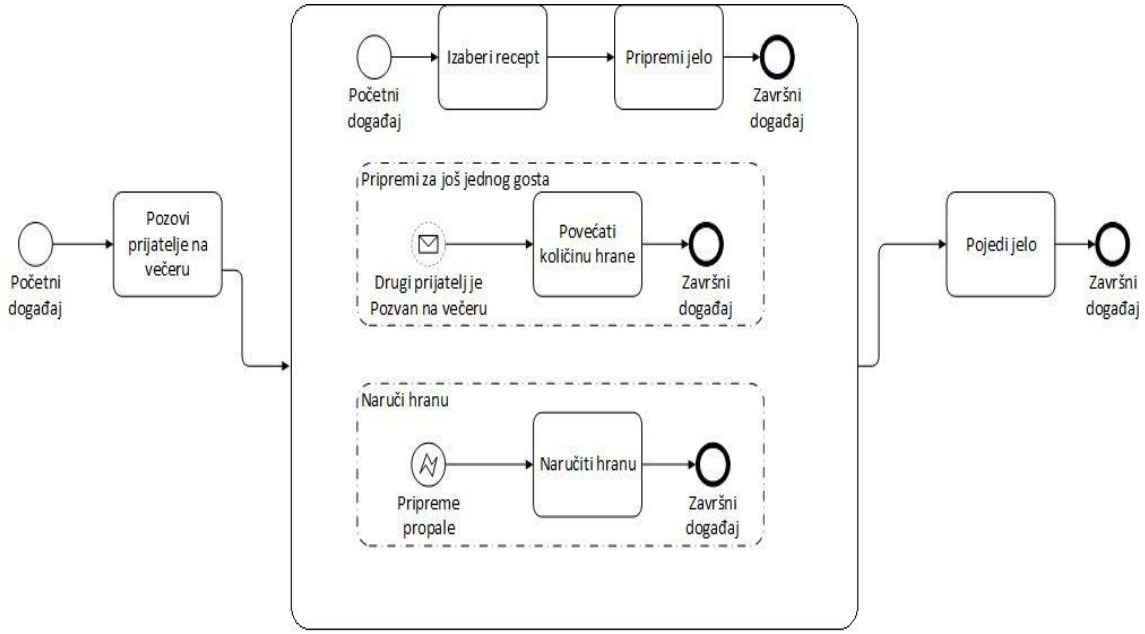


Slika 2.41. Potproces kao proširenje roditeljskog procesa

2.7.3. Događajni potprocesi

(engl. Event subprocesses)

Događajne potprocese obično smještamo u drugom procesu ili potprocesu. Oni su označeni okvirom s točkicama. Jedan početni događaj pokreće događajni potproces, a to se može dogoditi samo dok je proces zatvaranja ili potproces aktivan.



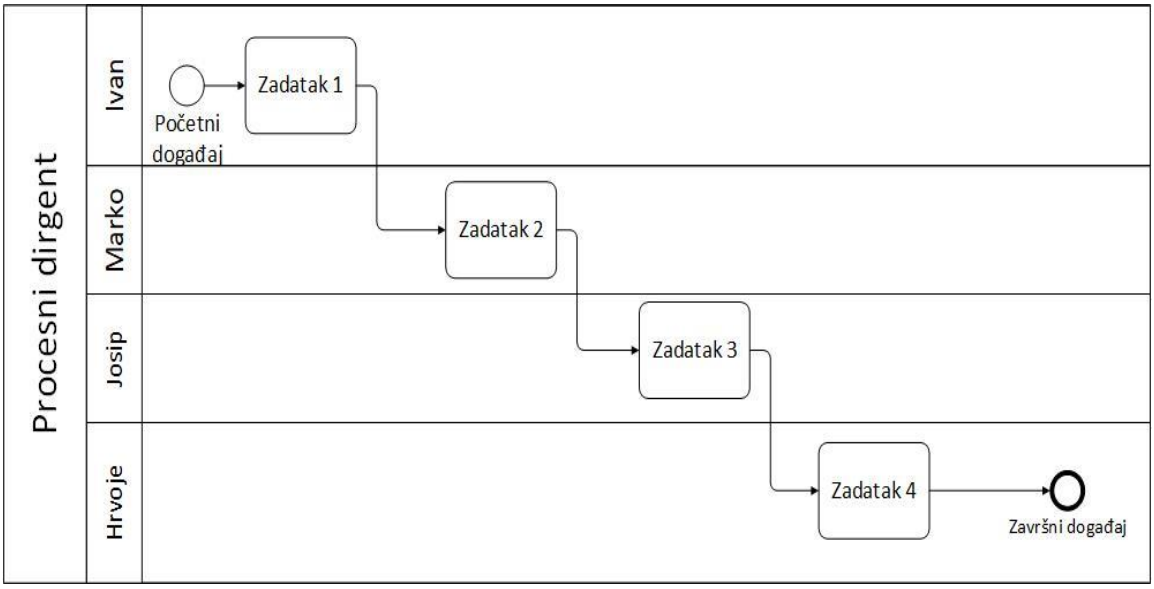
Slika 2.42. Prikaz primjera jednog potprocesnog događaja

Promotrimo sad primjer sa slike 2.42. Pozovemo nekoliko prijatelja na večeru. To pokreće potproces s odabirom recepta i zatim pripremu jela. Dok radimo, zazvoni telefon. Još jedan prijatelj nas je nazvao pa smo pozvali i njega na večeru. Sada povećamo količinu hrane za još jednu osobu ili postavljamo još jedno mjesto za stolom bez prekidanja pripreme jela. Ukoliko se dogodi neka nesreća dok pripremamo hranu, pogreška pokreće prekidanje potprocesnog događaja. Tada naručujemo hranu. Kada se taj potprocesni događaj završi, izlazimo iz potprocesa kroz redovni izlaz i krećemo jesti hranu.

2.8. Bazeni i tokovne poruke

2.8.1. Dirigent i orkestar

Već prije smo opisali uporabu traka za dodjeljivanje odgovornosti za zadatke ili poslovne potprocese različitim radnicima. Trake su unutar bazena, a granice bazena predstavljaju granice procesa od početka do kraja. Bazen kontrolira proces, odnosno dodjeljuje zadatke. Iz tog razloga bazen nazivamo dirigentom orkestra, a ovakav proces nazivamo orkestracijom. Na slici 2.43 prikazano je kako dirigent procesa organizira da Marko obavi Zadatak 2 nakon što Ivan odradi Zadatak 1. Kao što u orkestru dirigent odlučuje o svakoj melodiji te instrumentu, tako i procesni dirigent ima potpunu kontrolu nad procesom.



Slika 2.43. Dirigent kontrolira cijeli proces

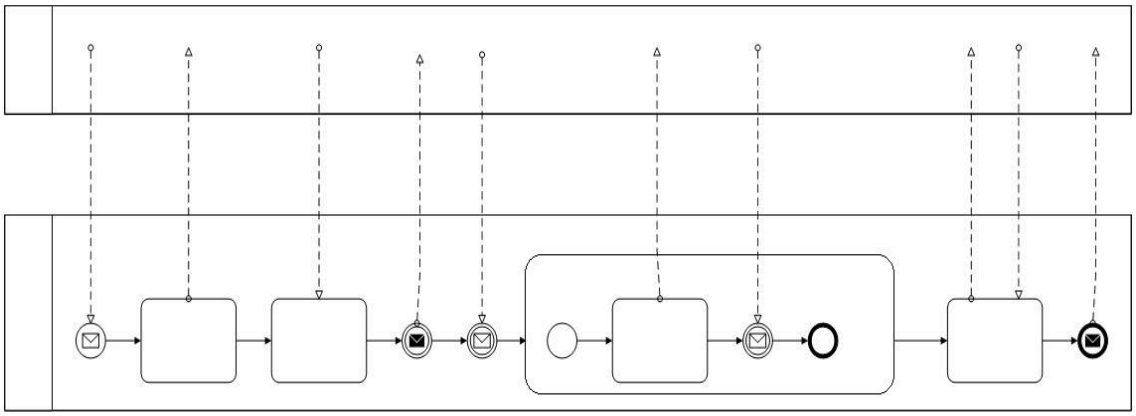
Iako je mnogima nelogično postojanje nekoga tko u potpunosti kontrolira svaki proces, upravo radi automatizacije procesa imamo ovakvu logiku zapisivanja. Naime, kod automatizacije procesa, procesni stroj kontrolira sve zadatke u tom procesu (iako ih izvršavaju različiti radnici). Upravo iz toga razloga, procesnog dirigenta poistovjećujemo s procesnim strojem.

2.8.2. Pravila primjene

Kod rada s bazenima i tokovima poruka, možemo modelirati sljedeće:

- Događaji hvatanja poruka čiji tok poruke dolazi
- Događaju bacanja poruka čiji tok poruke izlazi
- Zadatak čiji tok poruke dolazi **ili** izlazi
- (Prošireni) potproces, koji tok poruke dolazi ili izlazi

Na slici 2.44 možete vidjeti prihvatljive konstrukcije za korištenje tokovnih poruka između bazena. One se koriste samo između različitih bazena i ne smiju se koristiti za povezivanje elemenata istog bazena (za takve situacije koristimo tok aktivnosti). Prikazana su dva bazena; u donjem bazenu su elementi od kojih možemo poslati tokovnu poruku na drugi bazen, dok je gornji bazen prazan i predstavlja bilo koje elemente koje mogu primiti tokovnu poruku (događaji bacanja, potprocesi, zadaci).

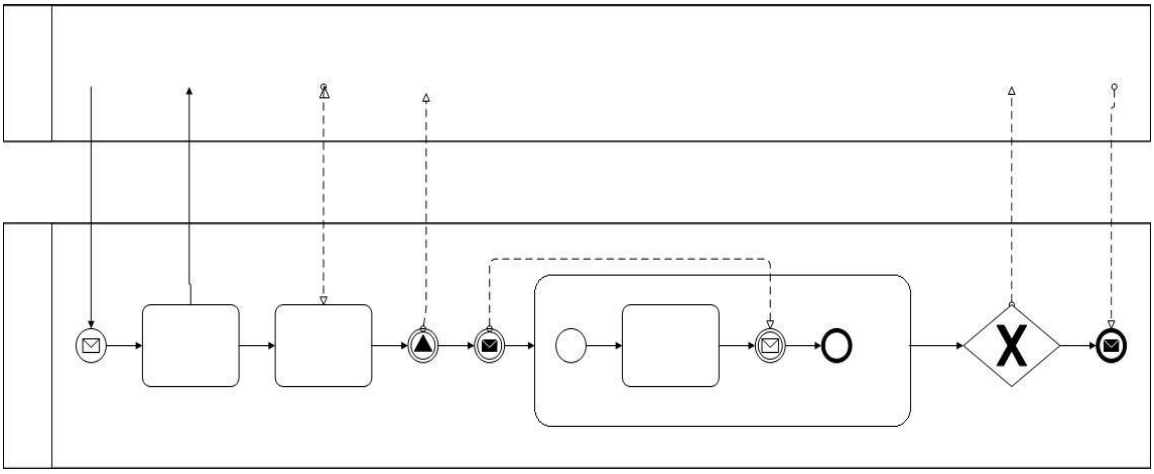


Slika 2.44. Prikaz prihvatljive konstrukcije pri korištenju bazena i tokovnih poruka

Osim što smo sad naveli koje su konstrukcije prihvatljive u BPMN-u, evo nekoliko svari koje ne pripadaju specifikaciji BPMN-a, stoga se ne smiju koristiti:

- Tok aktivnosti prelazi granicu bazena
- Tok poruke koji ne prelaze granicu bazena
- Događaji s tokom poruke koji nisu događaji poruke (tip događaja nije poruka)
- Događaji čiji tok poruke ulazi **i** izlazi
- Tok poruke koji sadrži strelicu s obje strane (na početku i na kraju)
- Grananje/spajanje s tokom poruke

Zabranjene konstrukcije za rad s bazenima i tokovima poruka prikazane su na slici 2.45.



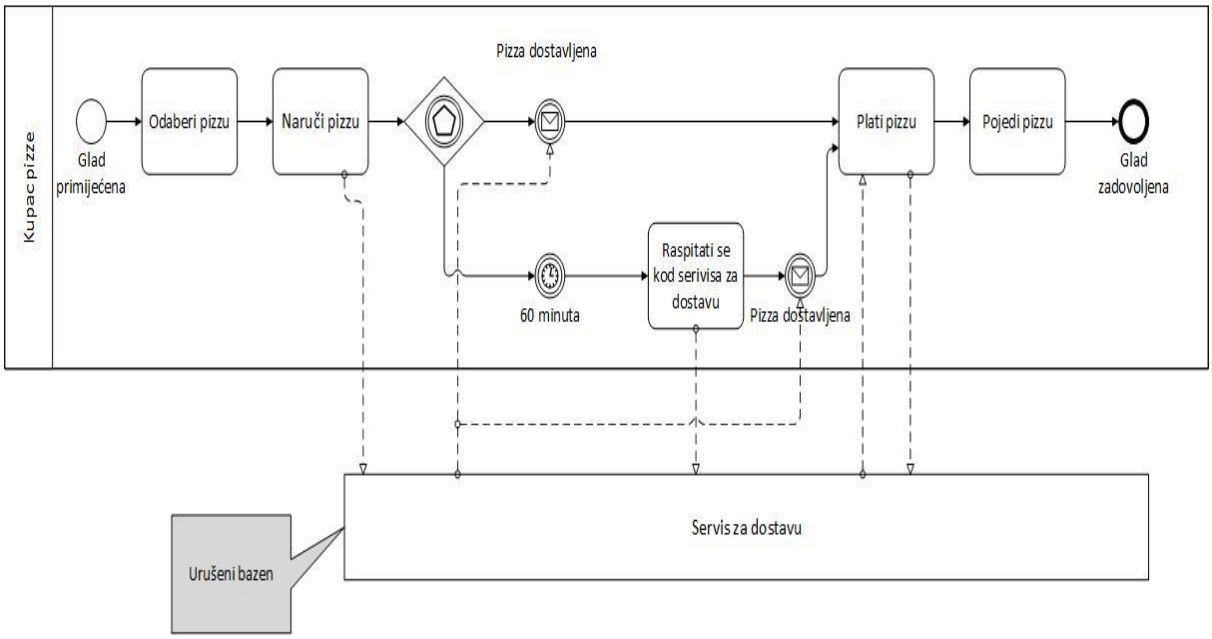
Slika 2.45. Prikaz zabranjene konstrukcije pri korištenju bazena i tokovnih poruka

2.8.3. Urušeni bazeni

(engl. Collapsed pools)

U suradnji s vanjskim tvrtkama, često se ne znaju svi detalji dijelova procesa. Zanimaju nas samo dijelovi koji su potrebni našem vlastitom procesu. Kod primjera naručivanja pizze nas ne zanima je li pizzerija ostala bez potrebnih sastojaka ni je li se dostavljač taj dan razbolio. Nas zanima samo jedna stvar, dostava pizze na našu adresu. Kod takvih situacija možemo sakriti proces dostave i urušiti ga.

Primjer takvog modeliranja prikazan je na slici 2.46.



Slika 2.46. Bazen dobavljača je urušen, što prikriva detalje procesa

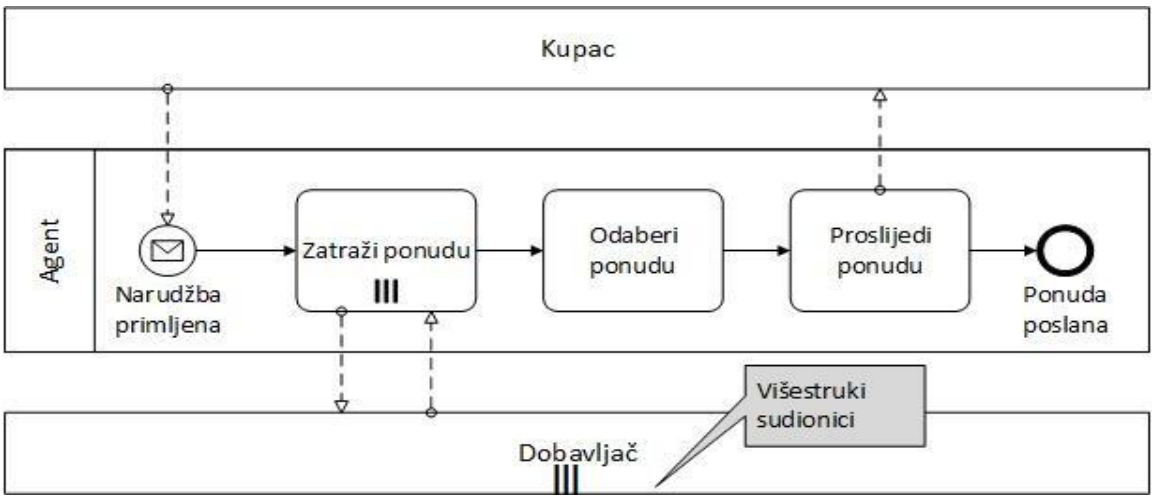
Osim što možemo urušiti proces, možemo urušiti i bazen. Ukoliko se odlučimo za to, bit će nam vidljive samo poruke koje se izmjenjuju. Kod toga je najveći problem što ne možemo prepoznati međuovisnosti. Ne možemo vidjeti prođu li uvijek zahtjevi i imaju li određene uvjete. Kako bismo izbjegli takve probleme, postoje dijagrami koreografije koji su spomenuti već u poglavlju 1.3. Primjer izgleda takvog dijagrama prikazan je na slici 2.47.

Slika 2.47. Oba bazena su urušena; samo su označeni tokovi poruka

2.8.4. Bazeni višestrukih instanca

(engl. Multiple instance pools)

Kod BPMN-a neke elemente možemo označiti kao višestruke, što znači da će se ti elementi i izvoditi više puta tokom izvođenja. Osim zadataka i potprocesa, i bazene možemo označiti kao višestruke. Budući da bazen uvijek predstavlja sudionika, tad se naziv mijenja u **višestruki sudionici***(engl. Multiple participants)*. Na slici 2.48. prikazano je kako se označava.



Slika 2.48. BPMN dopušta višestruke sudionike

Na slici 2.48 definirali smo tri sudionika: kupac, agent i dobavljač. Agentov posao se instancirao ukoliko je kupac dao narudžbu. To uzrokuje višestruko izvršenje zadatka „Zatraži ponudu“ . Bazen dobavljač ima istu oznaku kao zadatak koji se izvršio više puta. To ukazuje na to da nije uvijek isti dobavljač, odnosno da je nekoliko dobavljača uključeno u sam proces. Nakon što su dobivene sve ponude, samo jedna je odabrana i proslijeđena dalje kupcu.

Bazen višestrukih instanci nam uvijek pomaže kada želimo prikazati interakciju procesa kroz dijagram suradnje. Neki procesi se mogu instancirati samo jednom, dok se drugi mogu instancirati nekoliko puta unutar okvira suradnje.

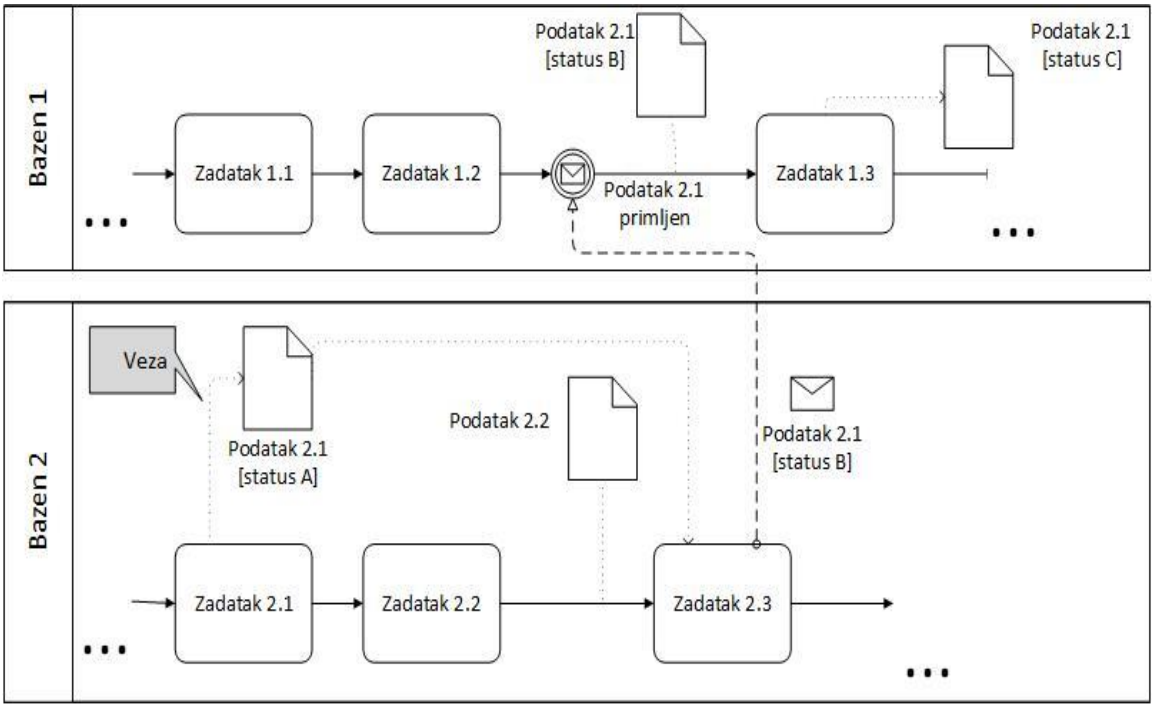
2.9. Podatkovni objekti

U opisu procesa, BPMN se usredotočuje na tok aktivnosti: slijed zadatka, grananje/spajanje i događaje. Svi ostali potencijalno relevantni aspekti tretiraju se kao manje važni. Isto tako vrijedi i za informacije ili dokumente koji se koriste (ili su generirani) u procesu. Oni se mogu prikazati unutar dijagrama modeliranjem **objekta podataka** (*engl. Data objects*). Oni mogu sadržavati sve vrste informacija: papirnate (fizičke) , apstraktne ili elektronske podatke.

Objekti podataka su povezani s objektima toka pomoću **podatkovnih veza** *(engl. Data associations)*. Osim njihove oznake, moguće im je dodijeliti određen status koji je označen unutar uglatih zagrada. Neki od tipičnih statusa su:

- Generirano
- Za provjeriti
- Provjereno
- Za preradu
- Prerađeno
- Odbijeno
- Odobreno

Primjer korištenja objekta podataka prikazan je na slici 2.49



Slika 2.49. Primjer rada s objektom podataka

Na slici 2.49. smo označili interakciju bazena 1 i 2:

Bazen 2 generira podatkovni objekt 2.1. s početnim statusom A. Zadatak 2.3. treba taj podatkovni objekt. S obzirom na to da te zadaće izravno slijede jedna drugu, možemo izostaviti direktnu udrugu te spojiti podatkovni objekt 2.2. direktno. Zadatak 2.3. transformira status A objekta podataka 2.1. na status B i šalje ga kroz tok poruke direktno u bazen 1. Bazen 1 čeka tu poruku i tada prosljeđuje primljene podatke 2.1. zadatku 1.3. koji ih transformira u status C.

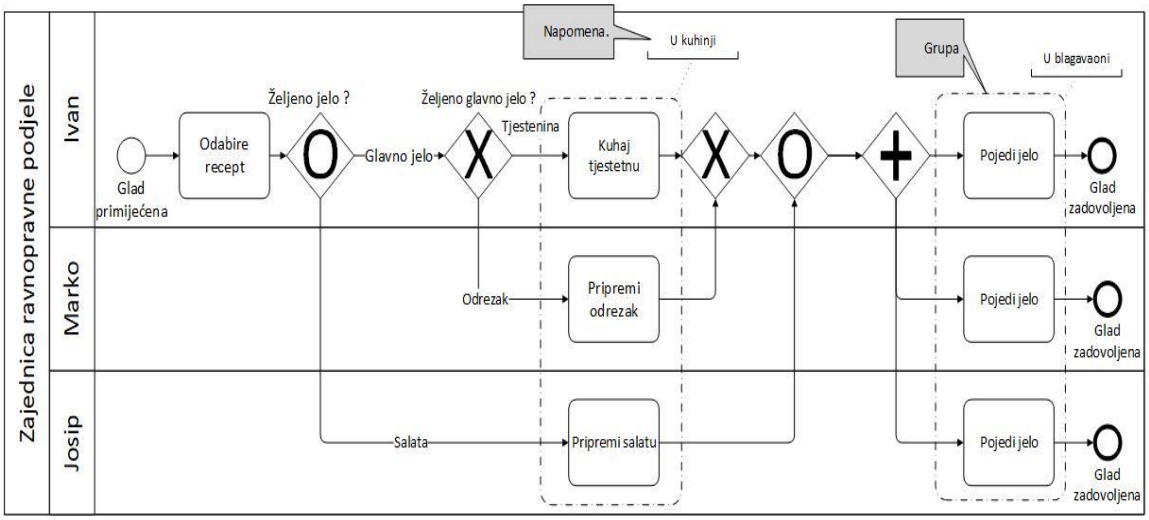
2.10. Artefakti

2.10.1. Napomene i grupe

(engl. Annotations and groups)

Napomene koristimo za nadopunjavanje dijagrama s dodatnim informacijama. Napomene mogu sadržavati skoro sve što je korisno, a možemo ih povezati s drugim elementima veza. Obično koristimo napomene kako bismo dobili uvid u izvršavanje zadatka, kao na primjer prosječno vrijeme obrade ili detaljnije objašnjenje nekog uvjeta kod grananja ili spajanja.

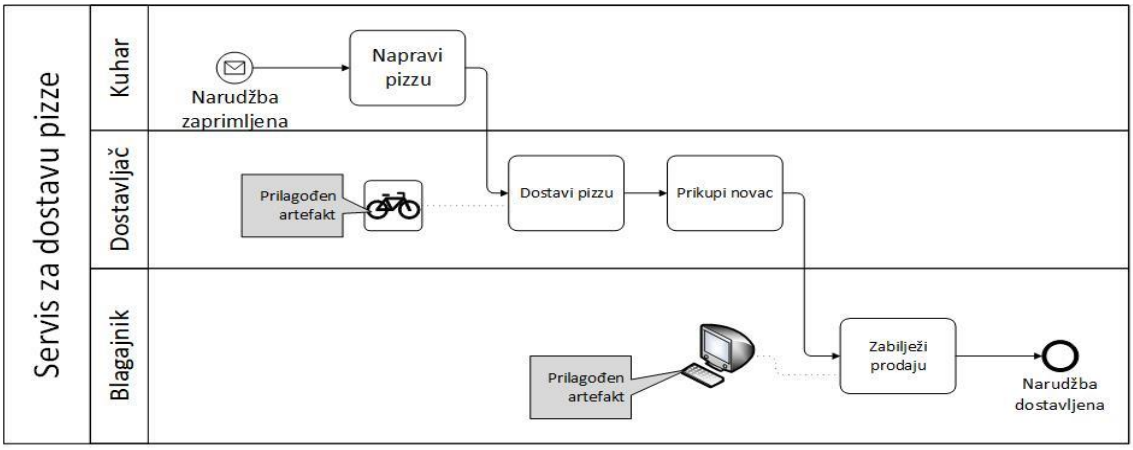
Pogledajmo sliku 2.50. Naveli smo da se priprema hrane odvija u kuhinji, ali se jede u blagovaonici. S obzirom na to da se ove informacije istodobno odnose na nekoliko elemenata u dijagramu, možemo ih grupirati. Kao i svi artefakti, grupe ne utječu na semantiku izvršenja.



Slika 2.50. Korištenje napomena i grupa

2.10.2. Prilagođeni artefakti

BPMN dopušta uvođenje simbola koji odgovaraju vašem stilu modeliranja kako biste mogli prilagoditi simbole određenih elemenata vašim preferencijama. Za njih se određuju ista pravila kao i za ostale artefakte: povežite ih kroz veze na bilo koji objekt toka (na bilo koju poziciju unutar dijagrama). Na slici 2.51. proširili smo proces dostave pizze s elementom bicikla i kompjutera.



Slika 2.51. Korištenje prilagođenih artefakata

3. Usporedba BPMN-a s drugim notacijama

Kao što smo već spomenuli, BPMN je među najrasprostranjenijim notacijama za modeliranje poslovnih procesa. Glavni problem drugih notacija jest nemogućnost modeliranja interakcije sudionika koji djeluju samostalno. Također, BPMN pruža puno veću preciznost rada s događajima od ostalih notacija.

3.1. Prošireni procesni lanac vođen događajima (eEPC)

(engl. Extender event-driven process chain)

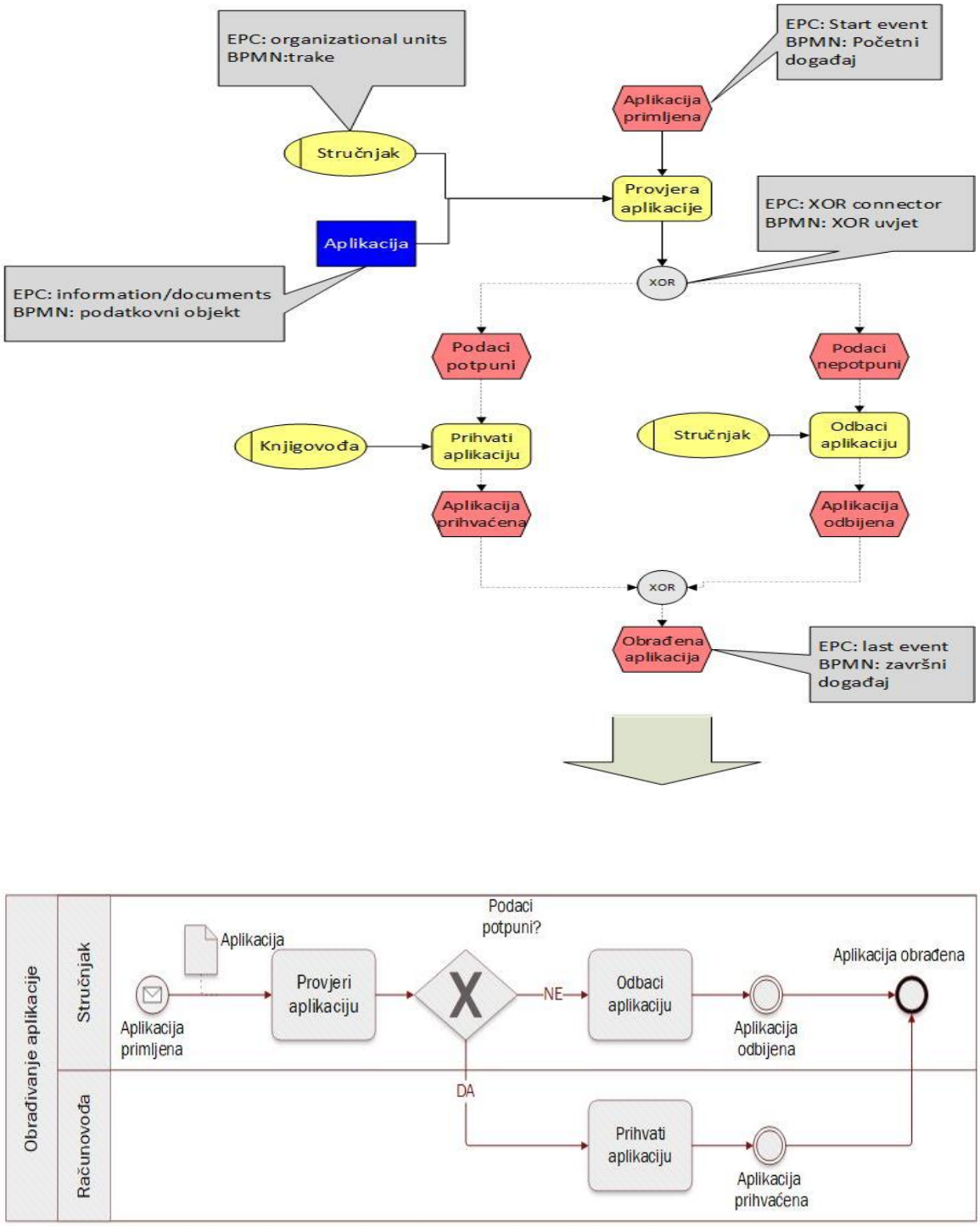
EPC (*event-driven process chain*) je sastavni dio ARIS metodologije razvoja na Saarland sveučilištu u suradnji s SAP AG. ARIS alati su integrirani s ERP (*enterprise resource planning)* rješenjima SAP-a. EPC je bio dominantna notacija 2008.godine. Danas se mnogi dizajneri prebacuju na BPMN iako to nije lako radi razlika u pristupu. U međuvremenu, ARIS je počeo nuditi i modeliranje s BPMN-om.

EPC sadrži tri temeljna elementa:

- Funkcije
- Događaje
- Konektore

Slično kao kod grananja/spajanja u BPMN-u, konektori mogu funkcionirati kao isključive grane (XOR), i/ili grane (OR) ili paralelne (AND). EEPC (prošireni EPC) je dodao simbole za opis organizacijske jedinice, podataka te aplikacije sustava.

Iako je pretvaranje EPC procesnih modela u BPMN-u relativno jednostavno (slika 3.1.), treba paziti kod pretvaranja događaja. Ne smiju se modelirati elementi bazirani na podacima *(engl. Data-based)* kao uvjete modelirane događajima *(engl. Event-based),*već je bolje koristiti grananje/spajanje bazirano na podacima. EPC ne prepoznaje razliku između početnog, srednjeg i završnog događaja ni tipove događaja kao što je događaj poruke ili sata. Također, ne podržava pridružene događaje pa je teško (ili nemoguće) modelirati funkcije praćenja, rješavanja problema ili eskalacije. Više o EPC te njihovom crtanju u Visual Paradigm-u se može pronaći na [6].



Slika 3.1. Pretvorba EPC modela u BPMN model

3.2. UML dijagram aktivnosti

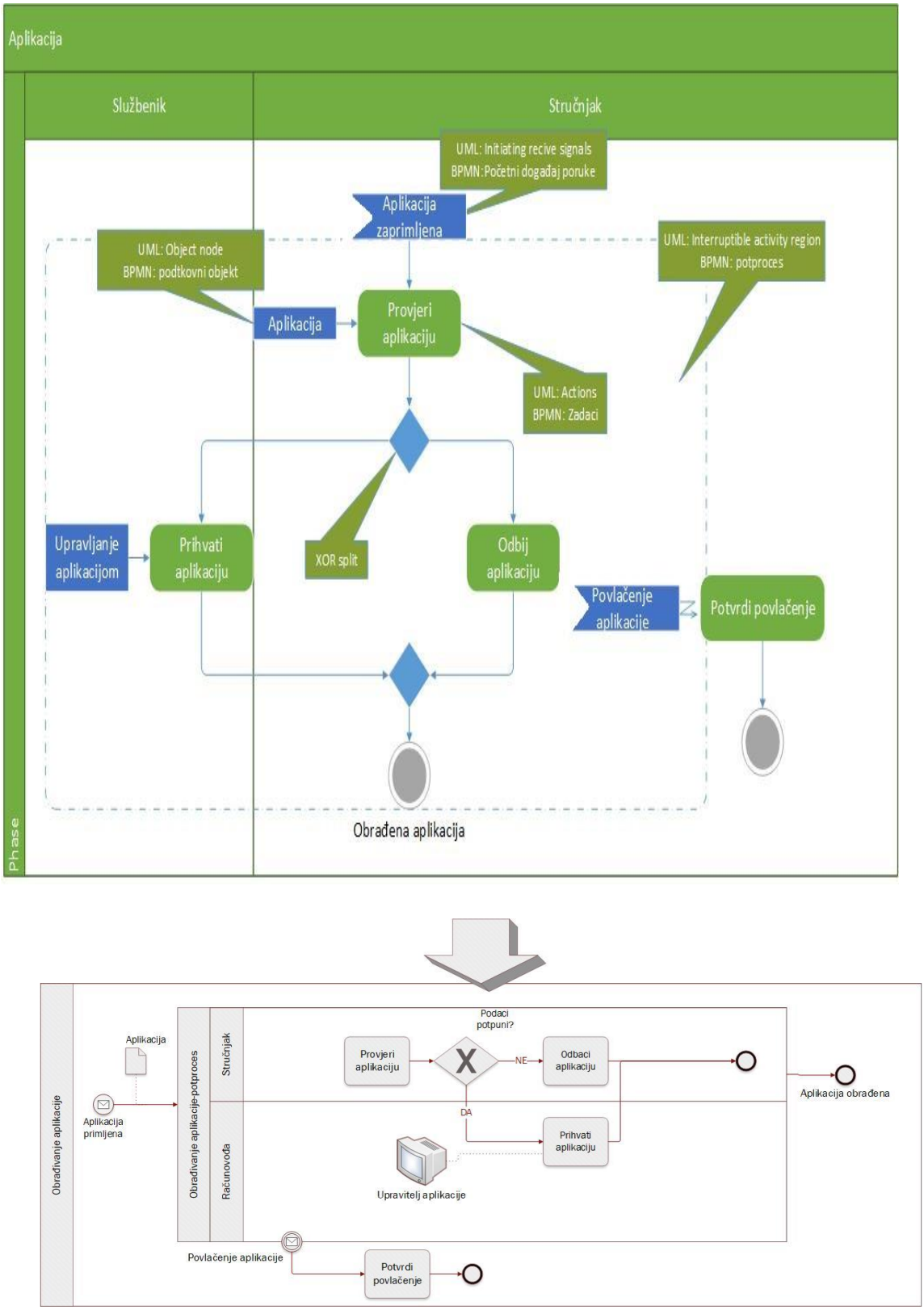
(engl. UML activity diagram)

Dijagram aktivnosti jedan je od 13 tipova dijagrama UML-a. Kao i s BPMN-om, OMG (*Object Management Group*) upravlja UML-om. Kako ne bi došlo do zabune, važno je istaknuti, BPMN nije nasljednik UML-a jer je UML opći jezik za modeliranje softverskih rješenja. UML dijagrami aktivnosti, iako nisu razvijeni za modeliranje poslovnih procesa, često se koriste u te svrhe, pogotovo za IT projekte. UML-ova notacija za dijagram aktivnosti obuhvaća više nego EPC-ova. Sadrži nekoliko simbola specifičnih za softver koji nemaju ekvivalentni simbol u BPMN-u, ali većina simbola lako se može pretvoriti u simbole podržane u BPMN-u.

Na probleme kod pretvorbe UML modela u BPMN nailazimo pri radu s prekidnim aktivnostima *(engl. Interruptible activity)* koje prelaze nekoliko traka. Bilo bi jednostavno kada bismo ih modelirali kao ugrađene potprocese, ali to je nemoguće zbog specifikacije BPMN-a: potprocesi ne smiju prijeći granicu trake. Kao jedino rješenje preostaje definirati potprocese kao globalne i modelirati bazene i trake.

BPMN je bolji za aplikacije vođene procesima, posebno ako trebate dokumentirati procesne funkcije. Određivanje procesa koji procesni stroj direktno izvršava je specijalnost BPMN-a te niti jedna druga notacija nema tako nešto. Više o UML dijagramima može se pronaći na [7].

Na slici 3.2. možemo vidjeti pretvorbu UML dijagrama u BPMN.



Slika 3.2. Pretvorba UML modela u BPMN model

4. Studijski primjer korištenja BPMN-a

4.1. Opis poslovnog procesa: kupovina karte za autobus

Kao studijski primjer, modelirat ću proces prodaje karata za autobus. Na šalteru putničke agencije na autobusnom kolodvoru mogu se kupiti karte za autobus koji vozi na raznim relacijama te se uz to mogu izraditi ili produljiti iskaznice za popust. Postoje različite cijene karata, ovisno o profilu putnika (studenti i učenici imaju popust) te ovisno o tome kupuje li se karta u jednom smjeru ili povratna. Službenik nudi sljedeće usluge:

- Prodaja karte u jednom smjeru za točan datum, vrijeme i relaciju
- Prodaja karte u oba smjera za točan datum, vrijeme i relaciju
- Izrada studentske/učeničke iskaznice
- Produženje studentske/učeničke iskaznice
- Prodaja studentske/učeničke karte u jednom smjeru uz predočenje iskaznice za točan datum, vrijeme i relaciju
- Prodaja studentske/učeničke karte u oba smjera uz predočenje iskaznice za točan datum, vrijeme i relaciju

Studenti i učenici mogu ostvariti popust samo ako posjeduju člansku iskaznicu putničke agencije. Prilikom izrade članske iskaznice trebaju pokazati studentsku iskaznicu ili indeks te službenik na šalteru provjerava točnost podataka iz vanjskog integracijskog sustava: eMatica ako se radi o učeniku i ISAK ukoliko se radi o studentu. U zahtjev za iskaznicu upisuju osobne podatke te, nakon što vanjski sustav provjeri podatke, pošalju popunjen zahtjev s novim podacima.

Izrada nove iskaznice:

- Ukoliko klijent nema prava za izradu iskaznice, javlja se poruka
- Ako klijent ima pravo, vanjski sustav vraća zahtjev s popunjenim svim poljima za izradu iskaznice
- Klijent mora platiti Cijenu 5 za naknadu koja se plaća jednom u akademskoj godini.
- Na iskaznici su napisani osobni podaci, datum izrade te datum isteka; iskaznica vrijedi do kraja tekuće akademske godine

- Za učenike školska godina računa se do 01.09, a za studente akademska godina računa se do 15.10
- Nakon što se podaci potvrde, podaci se spremaju u zahtjev za ispis iskaznice
- Službenik naplaćuje uslugu klijentu
- Naplata može biti gotovinom ili karticom.
- Nakon plaćanja ispisuje se račun.
- Nakon računa ispisuje se iskaznica koja je bila u zahtjevu

Produljuje iskaznice:

- Ukoliko klijent nema prava za produljenje iskaznice, javlja se poruka
- Ako klijent ima pravo na produljenje iskaznice, vanjski sustav vraća popunjena sva polja s novim podacima (novi datum izrade, novi datum isteka) zahtjeva za izradu iskaznice
- Klijent mora platiti Cijenu 5 za naknadu koja se plaća za produljenje
- Na iskaznici su napisani osobni podaci, datum izrade te datum isteka; iskaznica vrijedi do kraja tekuće akademske godine
- Za učenike školska godina računa se do 01.09, a za studente akademska godina računa se do 15.10
- Nakon što se podaci potvrde, podaci se spremaju u zahtjev za ispis iskaznice
- Službenik naplaćuje uslugu klijentu
- Naplata može biti gotovinom ili karticom.
- Nakon što je plaćanje izvršeno, ispisuje se račun.
- Nakon računa ispisuje se iskaznica koja je bila u zahtjevu

Prodaja karte:

- Nakon što klijent dođe na šalter, kaže za koji bi autobus kupio kartu (ovisno o relaciji, datumu i vremenu).
- Službenik pretražuje ima li slobodnih mjesta za taj određeni autobus kroz svoju aplikaciju AP1.
- Ukoliko nema mjesta, klijent može reći nove podatke ukoliko želi putovati nekim drugim autobusom.
- Ako ima mjesta, službenik provjerava iskaznicu za popust (ako je klijent ima). Nakon što provjeri iskaznicu (ako se radi o učeniku/studentu), odabire kategoriju iz cjenika.
- Odabrana karta sprema se u zahtjev za ispis te kreće naplata. Naplaćuje se cijena ovisno o kategoriji.
- Naplata može biti gotovinom ili karticom.

- Nakon što je naplata prošla, ispisuje se račun.
- Nakon računa, ispisuje se karta koja je bila u zahtjevu

Sve cijene te šifre kategorija su dane u slijedećoj tablici:

Cijene za 2018.godinu		
Kategorija	***artikl***	***cijena***
k1	karta u jednom smjeru	Cijena 1
k2	karta u oba smjera	Cijena 2
ku1	karta u jednom smjeru- učenici	Cijena 3
ks1	karta u jednom smjeru-studenti	Cijena 3
ku2	karta u oba smjera- učenici	Cijena 4
ks2	karta u oba smjera-studenti	Cijena 4
i1	Izrada/produljenje iskaznice za popust*	Cijena 5
*trajanje iskaznice vrijedi za tekuću akademsku godinu		

Slika 4.1. Cjenik s kategorijama

4.2. Modeliranje poslovnog procesa

4.2.1. Roditeljski proces

Nakon što je službenik došao na radno mjesto, prvo ulazi u aplikaciju AP1 i mora se prijaviti sa svojim podacima. Tek nakon što se prijavio prvi put, mora otvoriti svoju smjenu. Nakon što je smjena službeno otvorena, može odabrati sljedeće opcije:

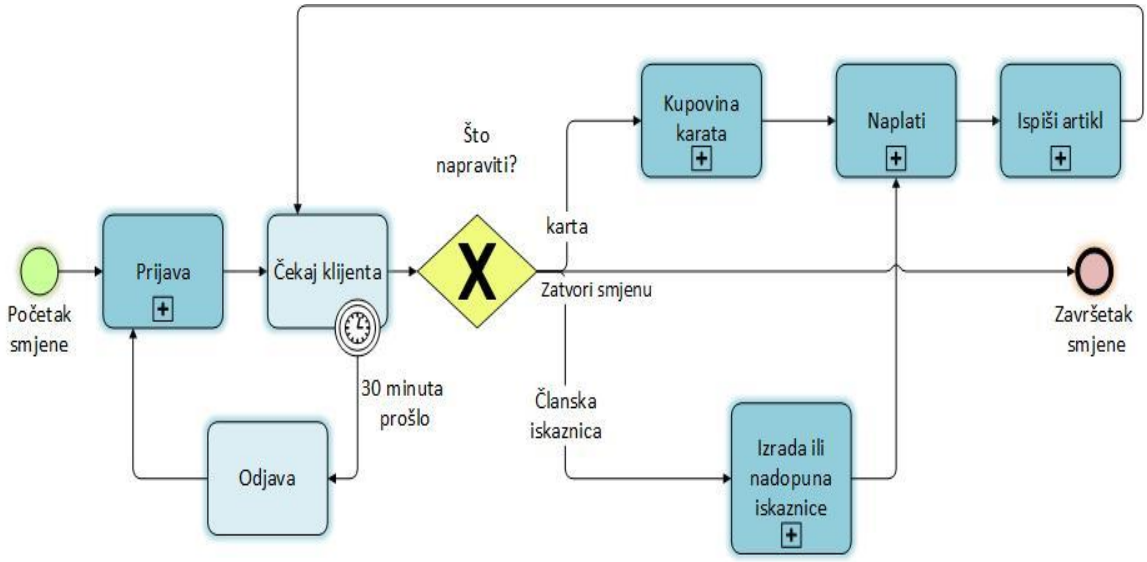
- Kupovina karte
- Izradu ili produljenje iskaznice
- Zatvaranje smjene

Ukoliko 30 minuta nije bio aktivan u aplikaciji, ona automatski odjavljuje službenika. Kada se službenik ponovo prijavi, i dalje je otvorena njegova smjena sve dok ju svjesno

sam ne zatvori. Smjena se zatvara samo nakon što službenik više ne radi i nakon toga ne može više ništa raditi (i time završava proces). Smjenu može zatvoriti nakon što je izašao iz potprocesa (kupovine karata, izrade/ produljenje iskaznice) i dok je prijavljen.

Ukoliko je klijent imao izradu/produljenje iskaznice, nakon toga može kupiti kartu.

Službenik može u bilo kojem trenutku izaći iz aplikacije (odjaviti se).

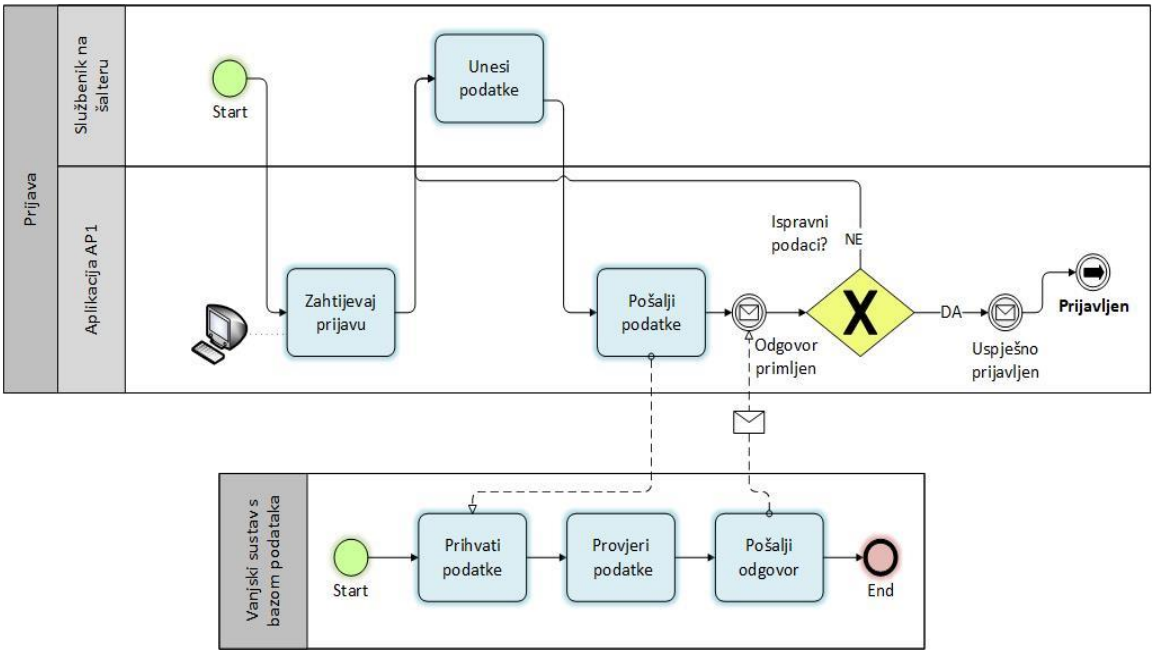


Slika 4.1. Roditeljski proces

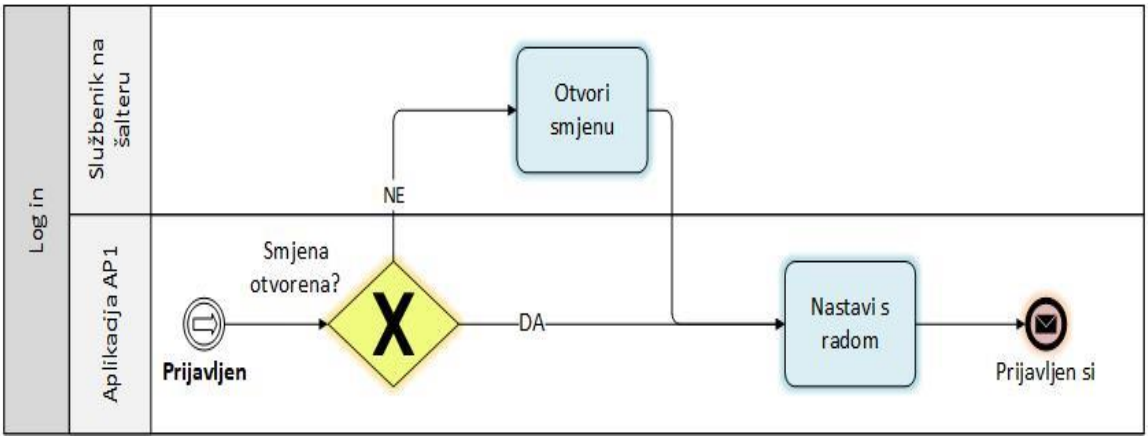
4.2.2. Potproces „Prijava“

Službenik na šalteru, nakon što dođe na posao, mora otvoriti smjenu. To radi kroz posebnu aplikaciju svoje tvrtke pod nazivom AP1. Nakon što upiše svoje podatke, podaci se šalju vanjskom sustavu koji je zadužen za održavanje aplikacije te sadrži bazu podataka svih zaposlenika. Provjerava se ispravnost podataka. Aplikacija javlja poruku ukoliko su podaci ispravni, ali i ako su neispravni. Nakon što se javila poruka o neispravnim podacima, aplikacija vraća na zahtjev za prijavu koji službenik mora opet popuniti.

Ako su podaci ispravni, aplikacija nudi opciju otvaranja smjene samo ukoliko ona nije već otvorena. Nakon toga službenik nastavlja s radom. Model potprocesa „Prijava“ ilustriran je na slikama 4.3. i 4.4



Slika 4.3. Potproces „Prijava“, prvi dio



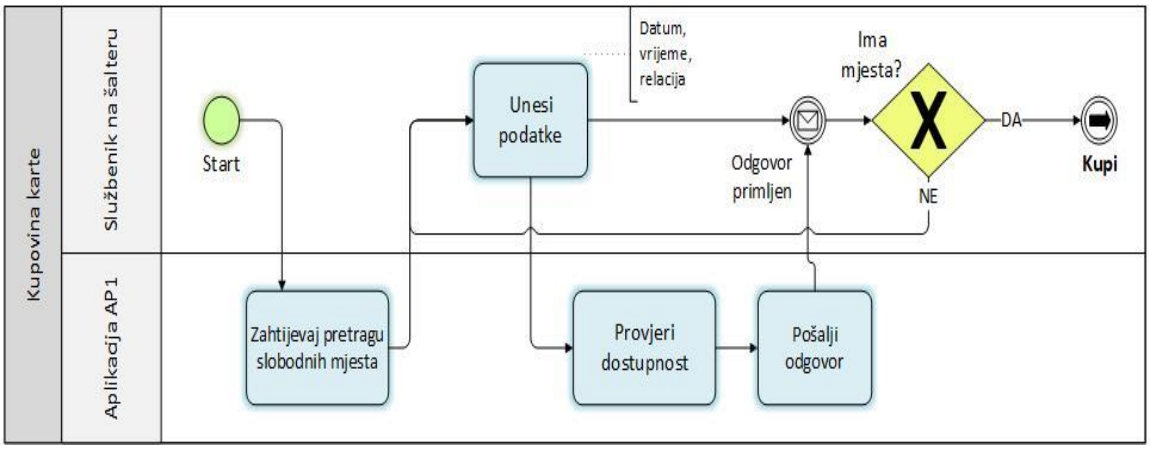
Slika 4.4. Potproces „Prijava“, drugi dio

4.2.3. Potproces „Kupovina karte“

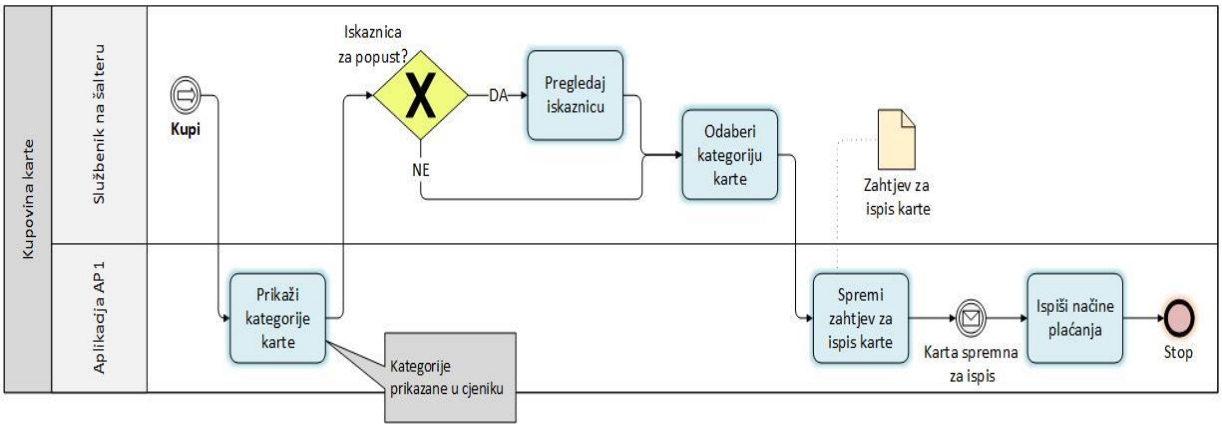
- Nakon što klijent dođe na šalter, kaže za koji bi autobus kupio kartu (ovisno o relaciji, datumu i vremenu).
- Službenik pretražuje ima li slobodnih mjesta za taj određeni autobus kroz svoju aplikaciju AP1.
- Ukoliko nema mjesta, klijent može reći nove podatke ukoliko želi putovati s nekim drugim autobusom.

- Ako ima mjesta, službenik provjerava iskaznicu za popust (ako je klijent ima). Nakon što provjeri iskaznicu (ako se radi o učeniku/studentu), odabire kategoriju iz cjenika.
- Odabrana karta sprema se u zahtjev za ispis te kreće naplata.
- Naplata kreće zahtjevom za odabir hoće li se platiti gotovinom ili karticom

Model potprocesa „kupovina karte“ ilustriran je na slici 4.5. i 4.6.



Slika 4.5. Potproces „Kupovina karte“, prvi dio



Slika 4.6. Potproces „Kupovina karte“, drugi dio

4.2.4. Potproces „Izrada iskaznice i produljenje“

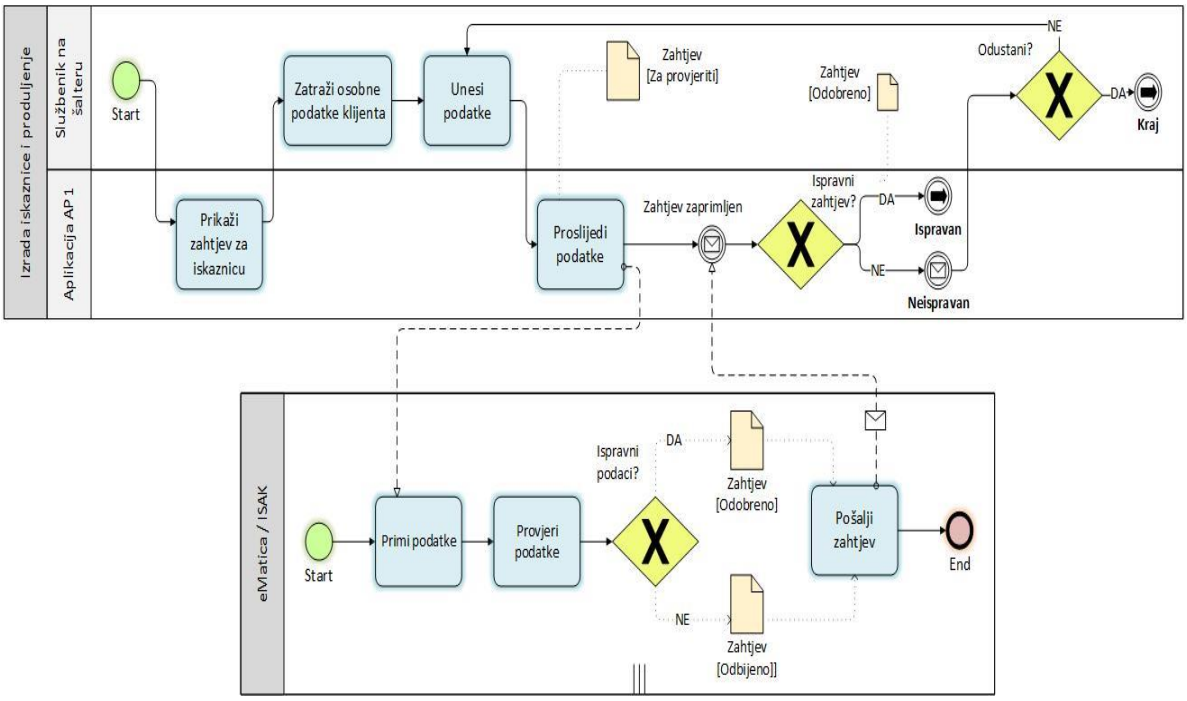
Izrada nove iskaznice:

- Ukoliko klijent nema prava za izradu iskaznice, javlja se poruka
- Ako klijent ima pravo, vanjski sustav vraća popunjena sva polja zahtjeva za izradu iskaznice
- Klijent mora platiti Cijenu 5 za naknadu koja se plaća jednom u akademskoj godini.
- Na iskaznici su napisani osobni podaci, datum izrade te do kad vrijedi; iskaznica vrijedi do kraja tekuće akademske godine
- Akademska godina se računa do 01.09. za učenike te 31.10. za studente
- Nakon što se podaci potvrde, podaci se spremaju u zahtjev za ispis iskaznice
- Službenik je spreman za naplatu
- Naplata kreće zahtjevom za odabir hoće li se platiti gotovinom ili karticom

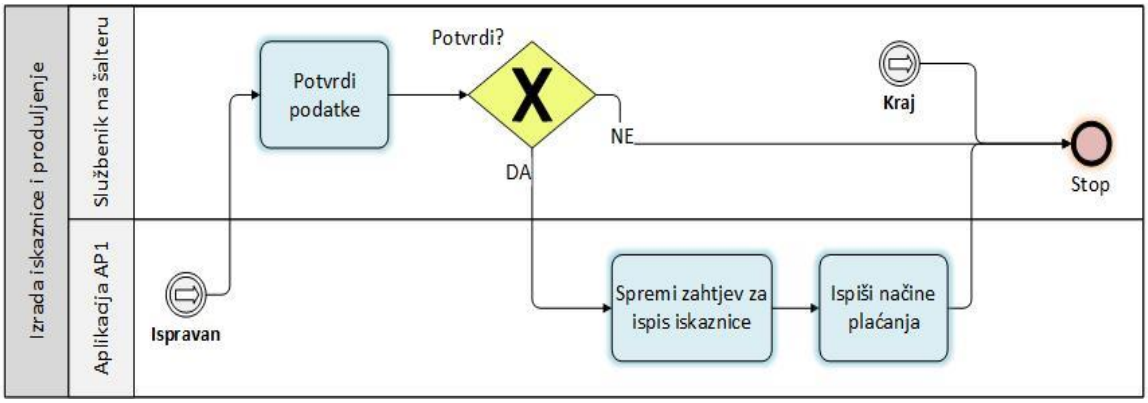
Produljuje iskaznica:

- Ukoliko klijent nema prava za produljenje iskaznice, javlja se poruka
- Ako klijent ima pravo, vanjski sustav vraća popunjena sva polja s novim podacima zahtjeva za izradu iskaznice (novi datum izrade, novi datum do kad vrijedi)
- Klijent mora platiti Cijenu 5 za naknadu koja se plaća za produljenje
- Na iskaznici su napisani osobni podaci, datum izrade te do kad vrijedi; iskaznica vrijedi do kraja tekuće akademske godine
- Akademska godina se računa do 01.09. za učenike te 31.10. za studente
- Nakon što se podaci potvrde, podaci se spremaju u zahtjev za ispis iskaznice
- Službenik je spreman za naplatu
- Naplata kreće zahtjevom za odabir hoće li se platiti gotovinom ili karticom

Model potprocesa "Izrada iskaznice i produljenje“ je prikazan na slikama 4.7. i 4.8.



Slika 4.7. Potproces „Izrada iskaznice i produljenje“, prvi dio



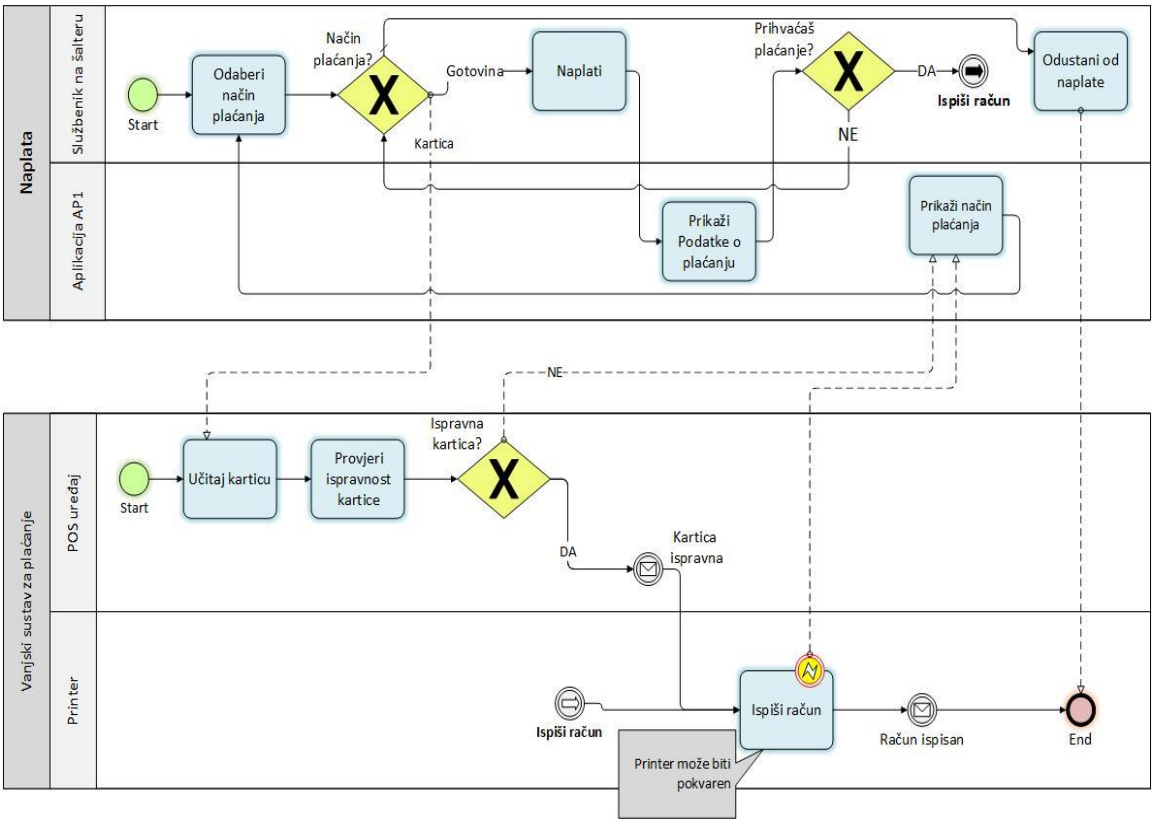
Slika 4.8. Potproces „Izrada iskaznice i produljenje“, drugi dio

4.2.5. Potproces „Naplata“

- Proces počinje nakon što se prikazao zahtjev za odabir načina plaćanja određenog artikla
- Ukoliko klijent plaća gotovinom, službenik na šalteru naplaćuje artikl. Nakon što uzme novce, potvrđuje plaćanje. POS uređaj ispisuje račun. Ako je račun dobro ispisan, javlja potvrdnom porukom i time se proces završava.

- Plaćanje karticom ide malo drugačije: nakon što je odabran način plaćanja karticom, POS uređaj je spreman za čitanje kartice. Kad je kartica učitana, podaci se šalju vanjskom suradniku za kartično plaćanje. Oni provjeravaju ispravnost kartice (vrijedi li, ima li dovoljno sredstva na računu). Ako je ispravna, javlja se poruka o ispravnosti te službenik mora potvrditi plaćanje. Čim službenik potvrdi plaćanje, POS uređaj ispisuje račun. Ako je račun dobro ispisan, javlja potvrdnom porukom i time se proces završava. Kada kartica nije ispravna, poruka javlja pogrešku te se vraća na odabir načina plaćanja (ukoliko klijent želi promijeniti način plaćanja na gotovinu ili dati drugu karticu)
- Ako službenik ne potvrdi plaćanje (nebitno je li karticom ili gotovinom), vraća se na odabir načina plaćanja
- Ukoliko tada želi odustati od kupovine, može

Model „Naplata“ je prikazan na slici 4.9.

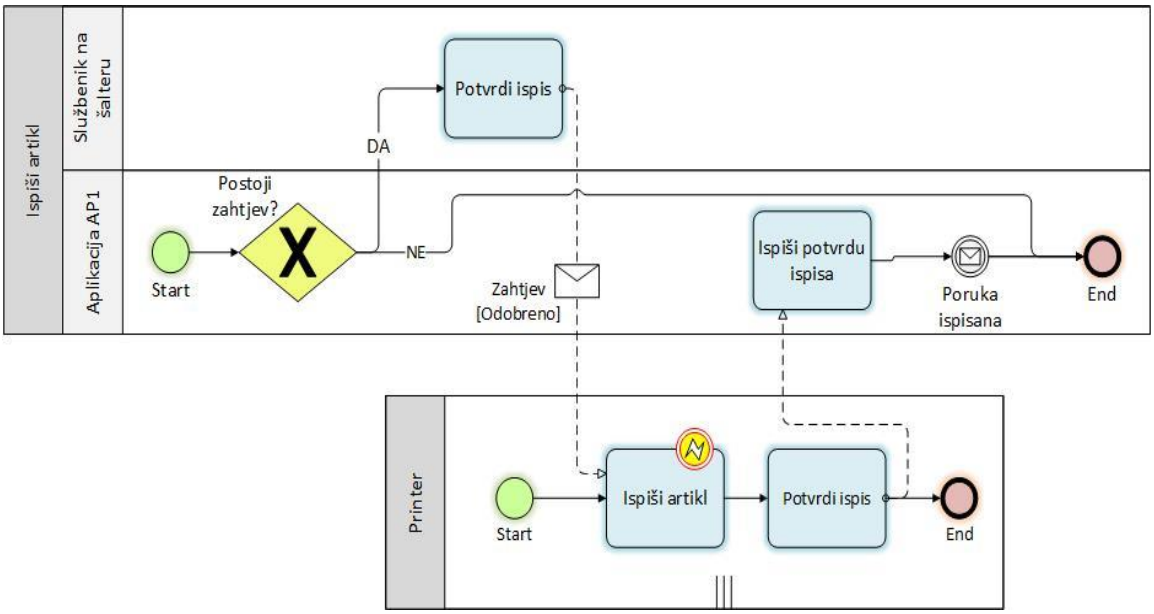


Slika 4.9. Potproces „Naplata“

4.2.6. Potproces „Ispiši artikl“

- Proces ispisa artikla počinje s gledanjem postoji li zahtjev za ispis
- Ukoliko postoji zahtjev, sukladno tome je li iskaznica ili karta, artikl za ispis se šalje printeru
- Nakon što je artikl isprintan, dobivamo poruku o potvrdi ispisa.
- Ako ne postoji nikakav zahtjev za ispis, proces odmah završava

Model potprocesa prikazan je slikom 4.10.



Slika 4.10. Potproces „Ispiši artikl“

Zaključak

Kroz lagane primjere vidjeli smo kako se na jednostavan način može grafički prikazati proces tako da bude jasan i onima koji se ne razumiju u poslovne procese. Nema specificirano kako pojedini model mora izgledati, to ovisi o dizajneru. Kao što smo vidjeli u mom studijskom primjeru, dijagrami mogu biti u boji, a i ne moraju. BPMN nema određeno koja je boja za što, ta se odluka isto prepušta dizajneru. Također, i procese koji su naizgled jednostavni, možemo modelirati tako da ga definiramo u detalje i tako jednostavan proces postane kompliciran. Na taj način možemo odabrati koji nam je dio procesa bitan da se modelira do najsitnijih detalja, a koji nam nije bitan. Radi toga, iste procese mogu interpretirati različiti modeli. Iako nema pravila oko modeliranja pojedinog procesa, postoje specifikacije kojih se moramo strogo pridržavati, ne smijemo ih ignorirati da bi stilski ljepše izgledao naš model.

Postoje procesni strojevi koji BPMN dijagrame automatski pretvaraju u računalni program. Ti softveri su specijalnost BPMN te ih niti jedna druga notacija ne koristi. Upravo iz tog razloga se sve više tvrtki odlučuje da BPMN bude njihov službeni grafički jezik Trenutno je BPMN najrasprostranjenija notacija za modeliranje poslovnih procesa i njegova uporaba samo raste.

Za negativnu stranu BPMN važno je napomenuti stroge zahtjeve kojih se mora pridržavati jer modeli nisu pretvoreni u računalni program od strane čovjeka, već ih direktno obrađuje procesni stroj.

LITERATURA

[1]. J. Freund, B, Rücker: Real-Life BPMN,3rd Edition: With introductions to CMMN and DMN

[2]. T. Allweyer: BPMN 2.0-Introduction to the Standard for Business Process Modeling

[3]. S.A. White, D. Miers: BPMN Modelling and Reference Guide - Understanding and Using BPMN. Future Strategies, Lighthouse Point FL, USA, 2008.

[4]. Alfresco software, *Activiti* , (2018), https://www.activiti.org/

[5]. Red Hat, *jBPM*, (2018), https://www.jbpm.org/

[6]. Visual Paradigm, *Visual Paradigm*, (1999), https://www.visual-paradigm.com/support/documents/vpuserguide/285/2144/56976_creatingepcd.html

[7]. uml-diagrams.org, *The Unified Modeling Language*, (2009), https://www.uml-diagrams.org/

SAŽETAK

BPMN (Business Process Model Notation) je grafička notacija koja služi za modeliranje i opis poslovnih procesa. Pruža standardizirani grafički jezik te olakšava komunikaciju i poboljšava razumijevanje poslovnih procesa.

Strukturu BPMN-a smo podijelili u 5 kategorija: objekte toka (zadaci/aktivnosti, događaji grananje/spajanje), objekte spajanja (tok aktivnosti, tok poruke, veze), artefakte, podatke te sudionike.

Određeni zadaci moraju se izvršiti tijekom procesa (aktivnosti), pod nekim uvjetima (grananje/spajanje) te stvarima koje se mogu dogoditi (događaji).

Svakom procesu je dodijeljen jedan bazen koji pripada kategoriji sudionika. Bazen kontrolira proces, odnosno sadrži trake koje predstavljaju tko je izvršitelj kojeg zadatka.

Objekte toka unutar jednog bazena međusobno povezujemo sa strelicama pod nazivom tok aktivnosti. Izvan granica bazena koristi se tok poruke.

Artefakti i podaci služe kao dodatne informacije o procesu te ne mijenjaju funkcionalnost modela.

SUMMARY

BPMN (Business Process Model Notation) is a graphical notation used for modeling and describing business processes. BPMN provides standardized graphical language, eases communication and improves understanding of business processes.

BPMN structure is divided into 5 categories: flow objects (activities/tasks, events, gateways), connecting objects (message flow, sequence flow), artifacts, data and participants.

Certain tasks have to be carried out during a process (activities), under certain conditions (gateways), and things may happen (events).

Each process has one pool which belongs to a participant category. The pool controls the process: it contains lanes which represent who is the executor of which task.

Flow objects within a pool are connected with an arrow we call sequence flow. Message flow is used to connect elements from different pools.

Data and artifacts are used as additional information about the process and they do not change the functionality of the modes.

ŽIVOTOPIS

Moje ime je Ena Martinek. Rođena sam 16.09.1991 u Zagrebu. 2010. godine završavam XIII. Gimnaziju te upisujem Prirodoslovno-matematički fakultet, smjer Matematika. 2014. godine, nakon završenog preddiplomskog studija, upisujem Diplomski sveučilišni studij Računarstvo i matematika.

Govorim engleski jezik izvrsno te osnovno njemački i talijanski.

Pored fakulteta bavim se i orijentalnim plesom. Zbog plesa često putujem u inozemstvo i do sada sam osvojila 11 medalja na međunarodnim natjecanjima u Hrvatskoj i inozemstvu.