

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Peter Brezovnik

# **Poslovna pravila v poslovnih procesih**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJ RAČUNALNIŠTVA IN  
INFORMATIKE

MENTOR: prof. dr. Matjaž B. Jurič

Ljubljana 2014



Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja. <sup>1</sup>

*Besedilo je oblikovano z urejevalnikom besedil  $\LaTeX$ .*

---

<sup>1</sup>V dogovorju z mentorjem lahko kandidat diplomsko delo s pripadajočo izvorno kodo izda tudi pod katero izmed alternativnih licenc, ki ponuja določen del pravic vsem: npr. Creative Commons, GNU GPL.





Št. naloge: 01996 / 2014  
Datum: 10.3.2014

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **PETER BREZOVNIK**

Naslov: **POSLOVNA PRAVILA V POSLOVNIH PROCESIH  
BUSINESS RULES IN BUSINESS PROCESSES**

Vrsta naloge: DIPLOMSKO DELO UNIVERZITETNEGA ŠTUDIJA

Tematika naloge:

Predstavite koncepte storitvenih arhitektur in upravljanja poslovnih procesov. Analizirajte notacijo BPMN 2 za modeliranje poslovnih procesov. Podrobno preglejte področje poslovnih pravil. Identificirajte vrste poslovnih pravil, njihovo uporabo in načine razvoja. Preglejte lastnosti sistemov za upravljanje s poslovnimi pravili in primerjajte štiri najpomembnejše izdelke. Izdelajte praktični primer poslovnega procesa, ki uporablja poslovna pravila.

Mentor:

prof. dr. Branko Matjaž Jurič



Dekan:

prof. dr. Nikolaj Zimic



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Peter Brezovnik, z vpisno številko **63080074**, sem avtor diplomskega dela z naslovom:

*Poslovna pravila v poslovnih procesih*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Matjaža B. Juriča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 7. marca 2014

Podpis avtorja:



*Zahvaljujem se prof. dr. Matjažu Juriču za mentorstvo pri izdelavi diplomske naloge.*

*Staršema, predvsem mami, da sta mi pokazala pot in omogočila hoditi po njej.*

*Hvala tudi vama, bica in deda, za vso vajino pomoč in da sta me vzpodbujala v težkih ter se veselila z mano ob lepih trenutkih.*

*Iskrena hvala tudi tebi Anže! Brez tebe čas študija ne bi bil niti približno enak.*

*Nazadnje pa se zahvaljujem tebi - Tina, ker si čudovito sonce, ki vsak dan sije in lepša moje življenje.*



Prvo načelo uspešnosti je miselna naravnost. Ljudje začnejo uspevati, ko začnejo verjeti.

J. C. Roberts



# Kazalo

## POVZETEK

## ABSTRACT

<b>1</b>	<b>UVOD</b>	<b>1</b>
<b>2</b>	<b>SOA IN BPM</b>	<b>5</b>
2.1	STORITVENO ORIENTIRANA ARHITEKTURA . . . . .	6
2.2	UPRAVLJANJE POSLOVNIH PROCESOV . . . . .	9
2.3	POVEZAVA MED SOA, BPM IN POSLOVNIMI PRAVILI . .	12
<b>3</b>	<b>MODELIRANJE POSLOVNIH PROCESOV</b>	<b>15</b>
3.1	BPMN 2.0 – PODPORA MODELIRANJU IN IZVAJANJU . .	16
3.2	OSNOVNI GRADNIKI . . . . .	17
3.3	PRIMER MODELIRANJA . . . . .	20
<b>4</b>	<b>UPORABA POSLOVNIH PRAVIL V PROCESIH</b>	<b>23</b>
4.1	VRSTE POSLOVNIH PRAVIL . . . . .	25
4.2	UPORABA POSLOVNIH PRAVIL . . . . .	27
4.3	RAZVOJ POSLOVNIH PRAVIL . . . . .	30
<b>5</b>	<b>AVTOMATIZACIJA POSLOVNIH PRAVIL</b>	<b>39</b>
5.1	SISTEM ZA UPRAVLJANJE S POSLOVNIMI PRAVILI . .	42
5.2	TEHNIKE ZA PREDSTAVITEV POSLOVNIH PRAVIL V BRMS . . . . .	45

## KAZALO

5.3	PREDNOSTI IN SLABOSTI UPORABE BRMS . . . . .	48
<b>6</b>	<b>KRATEK PREGLED BRMS TEHNOLOGIJ</b>	<b>51</b>
6.1	IBM WODM . . . . .	52
6.2	ORACLE BR . . . . .	53
6.3	OPEN RULES . . . . .	54
6.4	JBOSS DROOLS . . . . .	55
<b>7</b>	<b>IZDELAVA PRAKTIČNEGA PRIMERA</b>	<b>57</b>
7.1	NAČRTOVANJE . . . . .	57
7.2	IZVAJANJE . . . . .	64
<b>8</b>	<b>ZAKLJUČEK</b>	<b>71</b>

# SEZNAM UPORABLJENIH KRATIC

- ABRD - Agile Business Rule Development (agilna metodologija za razvoj poslovnih pravil)
- BAM - Business Activity Monitoring (nadzor in spremljanje poslovnih aktivnosti)
- BPEL - Business Process Execution Language (programski jezik za izvajanje poslovnih procesov)
- BPM - Business Process Management (upravljanje poslovnih procesov)
- BPMN - Business Process Modelling Notation (notacija za modeliranje poslovnih procesov)
- BPMN 2.0 - Business Process Modelling And Notation (notacija za modeliranje in izvajanje poslovnih procesov)
- BRM - Business Rules Management (upravljanje s poslovnimi pravili)
- BRMS - Business Rule Management System (sistem za upravljanje poslovnih procesov)
- CRM - Customer Relationship Management (upravljanje s strankami)
- DSL - Domain Specific Programming Language (domensko specifični jeziki)

## KAZALO

- EPC - Event Process Chain (splošno namenski jezik za modeliranje)
- ERP - Enterprise Resource Planning (celovita programska rešitev)
- ESB - Enterprise Service Bus (storitveno vodilo)
- http - HyperText Transfer Protocol (glavna metoda za prenos informacij na spletu)
- IT - Information Technology (informacijska tehnologija)
- KIE-WB - Knowledge Is Everything Workbench (okolje za izvajanje poslovnega procesa)
- KPI - Key Performance Indicators (ključni kazalniki poslovanja)
- OMG - Object Management Group (skupina za objektno upravljanje)
- SOA - Service Oriented Architecture (storitveno orientirana arhitektura)
- SOAP - Simple Object Access Protocol (enostaven protokol za dostop do objektov)
- UDDI - Universal Description Definition and Integration (univerzalen opis, odkritje in integracija)
- UML - Unified Modeling Language (splošno namenski modelirni jezik)
- WS - Web Service (spletne storitve)
- WSDL - Web Services Description Language (opisni jezik spletnih storitev)
- XML - eXtensible Markup Language (razširljiv označevalni jezik)
- XPDL - XML Process Definition Language (jezik XML za definicijo procesov)

# POVZETEK

Spremembam se ni mogoče izogniti – z njimi se je potrebno soočiti na pravi način. To velja tudi za združbe, katere morajo v želji po povečevanju poslovne učinkovitosti, napredku ali ohranjanju svojega položaja na trgu, znati upravljati poslovne procese (Business Process Management - BPM) in jih prilagajati okolju. Spremembe pa se ne nanašajo zgolj na poslovne procese same, pač pa tudi na podpirajoče aplikacijske sisteme. V njih je v obliki poslovnih pravil zbrana poslovna logika, ki omogoča avtomatizacijo in konsistenten pregled nad odločitvami, ki se vršijo v poslovnem procesu. Ker so poslovna pravila dinamična, je njihovo upravljanje ključnega pomena za doseganje organizacijske agilnosti. V ta namen smo v diplomski nalogi z različnih vidikov preučili pomen poslovnih pravil, jih podrobno opisali in predstavili sisteme (Business Rule Management System - BRMS), ki v povezavi s prevladujočim (Service Oriented Architecture - SOA) konceptom za razvoj programskih rešitev, omogočajo njihovo učinkovito upravljanje in vzdrževanje.

**Ključne besede:** Poslovna pravila, BRMS, BPMN 2.0, SOA, BPM, JBoss Drools



# ABSTRACT

Changes are unavoidable and as such they should be addressed appropriately. This means that the companies should know how to manage their business processes (BPM) and how to adapt them to their business environment in order to increase business efficiency, business progression or just to keep their place on the market. The changes refer not only to business processes, but to the supporting application systems as well. These systems contain business logic in the form of business rules which enable automation and consistent overview of decisions that are executing in business processes. Since these decisions change dynamically it is vital to manage them properly in order to achieve business agility. In this diploma thesis we focus on presenting the importance of business rules from different angles. We describe the rules in detail and explain the meaning of Business Rule Management System (BRMS) which together with Service Oriented Architecture (SOA) enables management and maintenance of Business Rules.

**Keywords:** Business Rules, BRMS, BPMN 2.0, SOA, BPM, JBoss Drools



# Poglavje 1

## UVOD

Vsak dan sprejemamo veliko (osebnih) odločitev. Z nekaterimi od njih se ubadamo enkrat dnevno, z drugimi pa se soočamo bolj pogosto. Pri odločanju nam pomaga naše vedenje o problemu, ki ga želimo rešiti, naše pretekle izkušnje s podobnimi situacijami ali pa kaj tretjega. Enako velja tudi za odgovorne v združbah, ki se morajo v želji po čim boljšem opravljanju svojega poslanstva vsakodnevno odločati med različnimi, včasih tudi suboptimalnimi, variantami. Posledice njihovih odločitev so dolgoročno vidne v poslovni učinkovitosti organizacije in njenem napredku, kratkoročno pa se najprej izrazijo v načinu izvajanja samega poslovnega procesa in aplikacijskih sistemih, ki takšne procese podpirajo. Da bi bil pregled nad odločitvami čimboljši in da bi bila zagotovljena njihova konsistentnost, ima vsaka organizacija definirana poslovna pravila. Ta definirajo poslovno logiko in določajo vidike poslovanja združbe. V njih so zbrani natančni podatki o podrobnostih in izvajanju poslovnega procesa. Tako lahko vsaka za to pooblaščen oseba dobi pojasnila o tem, do katerih odločitev je prišlo in zakaj. Okoliščine poslovanja pa ne ostajajo za vedno enake in se seveda spreminjajo - s tem pa tudi poslovna pravila, zato je zelo pomembno z njimi dobro upravljati. Zbrana morajo biti na enem mestu, da je dostopanje do njih hitro in enostavno. Pomembno je tudi, da so razumljiva širšemu krogu uporabnikov, tako poslovnemu, kot tudi tehničnemu osebju in da lahko z njimi upravlja vsakdo, ki ima za to pravice.

Posledice spreminjanja morajo biti zabeležene, spremembe pa se morajo na jasnem načinu odražati v poslovnem procesu in podpirajočih aplikacijskih sistemih. V ta namen smo v diplomski nalogi podrobno predstavili poslovna pravila, jih umestili v poslovni in tehnološki kontekst, ter na praktičnem primeru prikazali njihovo uporabo.

V 2. poglavju je predstavljen pomen dobrega upravljanja poslovnih procesov - BPM (angl. Business Process Management) in način, kako to doseči. Storitveno orientirana arhitektura - SOA (angl. Service Oriented Architecture) skupaj z BPM podpira celoten življenjski cikel poslovnega procesa. BPM predstavlja procesno orientirana disciplino, SOA pa je primer tehnologije, ki jo za realizacijo uporablja informacijska tehnologija - IT (angl. Information Technology). V nadaljevanju poglavja je opisana še njuna povezava s poslovnimi pravili.

V 3. poglavju je predstavljen jezik BPMN (angl. Business Process Modelling Notation) za modeliranje poslovnih procesov, ki je od verzije BPMN 2.0 tudi izvajalni jezik. Z verzijo BPMN 2.0 se spremeni tudi pomen same okrajšave, ta sedaj pomeni Business Process Modeling and Notation in nadomešča dosedanji jezik za izvajanje poslovnih procesov - BPEL (angl. Business Process Execution Language). Tako je omogočeno enostavnejše sodelovanje med poslovnim analitikom in tehničnim razvijalcem, saj oba delata na istem modelu procesa, pri tem pa ne prihaja do morebitnih napak pri pretvarjanju med modeloma.

4. poglavje nas približje seznanja s poslovnimi pravili in jih razdeli v različne kategorije. Nato nam predstavi prednosti, ki jih pristop z uporabo poslovnih pravil nudi, principe, ki omogočajo doseganje teh prednosti in metodologijo za razvoj poslovnih pravil. Poglavje se zaključuje z napotki, ki omogočajo doseganje najboljših učinkov pri uporabi pristopa s poslovnimi pravili v poslovnem procesu.

Načini implementacije poslovnih pravil v aplikacijskem sistemu so obdelani v 5. poglavju, kjer je nato podrobneje opisan eden (sistem za upravljanje s poslovnimi pravili) od naštetih načinov ter njegove prednosti in slabosti.

Opisali smo tudi tehnike za predstavitev poslovnih pravil v teh sistemih.

V 6. poglavju smo najprej predstavili konkretne rešitve različnih ponudnikov BRMS sistemov, nato pa smo s pomočjo ene od omenjenih tehnologij (JBoss Drools 6) izdelali praktičen primer. Načrtovanje in izvajanje le-tega smo prikazali v 7. poglavju.



# Poglavje 2

## SOA IN BPM

Glavni cilj informacijske tehnologije je zagotavljanje podpore poslovnim procesom. Le-ti so predstavljeni kot množica aktivnosti (izvaja jih lahko stroj ali človek), ki stremijo k istemu cilju – doseganju poslovnih rezultatov [1]. Na začetku je bila razvita podpora zgolj posameznim (računovodskim) funkcijam, kot so denimo vodenje poslovnih knjig ali podpora plačnemu sistemu. Kljub uspešnosti avtomatizacije teh aktivnosti, pa se je vseeno širilo prepričanje, da to še ni vse, kar lahko IT ponudi in da bi bilo dobro posamezne avtomatizirane funkcije tudi povezati. V odgovor na to prepričanje so informacijski sistemi vsebovali vedno večji nabor funkcionalnosti in tako so se na trgu razširili sistemi, kot so ERP (angl. Enterprise Resource Planning), CRM (angl. Customer Relationship Management) in ostali. Ob soočenju z omenjenimi sistemi so združbe spoznale, da je avtomatizacija nujna in da neposredno vpliva na poslovne rezultate. Bolj učinkovito, kot so lahko definirale in podprle poslovne procese, bolj učinkovito so združbe lahko poslovale. Prav učinkovitost pa je ob inovativnosti ključni kazalec za uspeh in povečanje konkurenčne prednosti na tržišču. Tako so zavoljo uspešnosti želele avtomatizirati vsak korak poslovnega procesa oz. razviti aplikacije, ki bi nudile pomoč in podporo poslovnemu procesu od njegovega začetka pa vse do konca.

Kljub temu, da se omenjena zahteva sliši enostavno, temu ni tako. Za to

obstajata vsaj dva objektivna razloga:

1. Vsaka združba opravlja edinstven poslovni proces, na osnovi katerega morajo biti zasnovani aplikacijski sistemi.
2. Poslovni procesi niso statični, temveč se skozi čas spreminjajo. Vsaka sprememba v poslovnem procesu mora biti izražena tudi v aplikacijskem sistemu, ki mora biti prožen in omogočati hitre odzive na spremembe.

Da bi zadostili obema pogojevema, arhitektura takšnega sistema ne sme biti toga. Čim manjši mora biti t.i. »IT Gap«, to je čas, da IT strokovnjaki prilagodijo informacijski sistem spremembam – novim potrebam, ki so se pojavile v poslovnih procesih [2]. Prav tako pa se morajo aplikacijski sistemi zelo natančno prilagajati dinamičnim poslovnim procesom, čemur doslej ni bilo tako. Ti sistemi, zaradi nejasnosti načrtovanih diagramov ali zaradi netočnosti implementacije zaradi tehnoloških preprek, niso bili vedno odraz realnega stanja poslovnega procesa združbe. Prihajalo je do semantičnega prepada in razlik med dejanskim poslovnim in podpornim aplikacijskim sistemom. Kot odgovor na omenjene težave glede težavnosti razvoja in vzdrževanja kompleksnih sistemov, se je pokazala SOA.

## 2.1 STORITVENO ORIENTIRANA ARHITEKTURA

Storitveno usmerjene arhitekture so vrste porazdeljenih sistemov, ki temeljijo na paradigmi zahtevk/odgovor (ang. request/reply). Aplikacijska poslovna logika ali posamezne funkcije so sestavljene iz komponent, imenovanih storitve. Ker so bili včasih poslovni informacijski sistemi grajeni funkcionalno, niso podpirali celotnega poslovnega procesa, ampak le posamezne funkcije. Obenem pa so bile podprte funkcije razvite v različnih časovnih obdobjih in izdelane v različnih tehnologijah. Prihajalo je do težav pri povezovanju

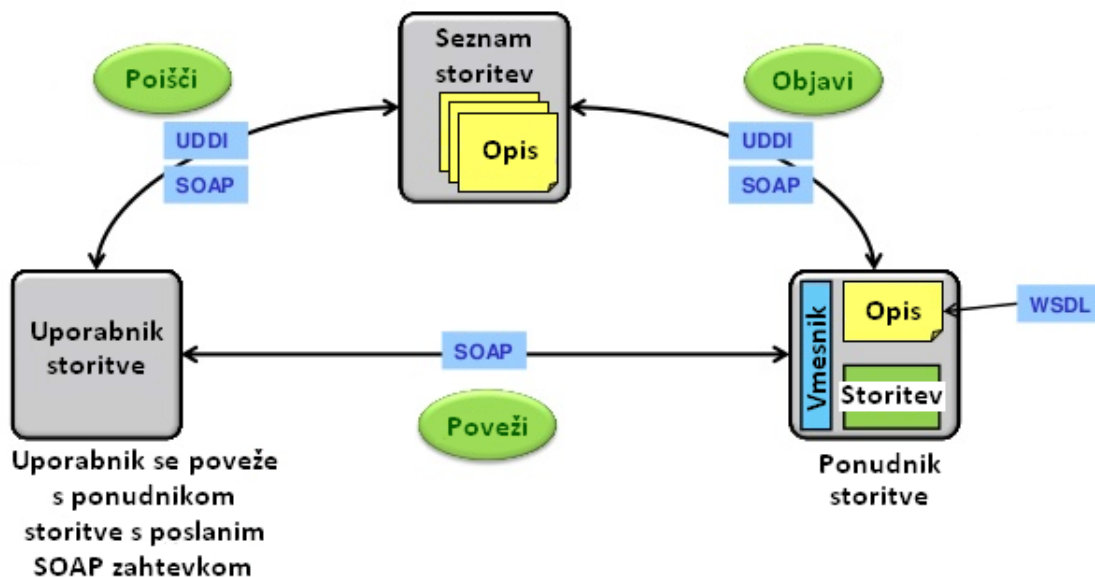
različnih aplikacij v celoto – integraciji. Ključna značilnost storitev je njihova šibka sklopljenost [4], njihovi vmesniki (pogodba med storitvijo in odjemalcem) pa so neodvisni od implementacije. To pomeni, da je lahko storitev implementirana v katerikoli tehnologiji (denimo .Net ali J2EE), aplikacija, ki uporablja to storitev, pa se lahko izvaja v drugem programskem jeziku ali pa na drugi platformi. SOA in storitve so torej industrijski dogovor za interoperabilnost tehnologij in morajo upoštevati naslednje principe [3]:

- Meje storitev so eksplicitne – Zaradi razumljivosti mora biti poudarek na preprostosti vmesnikov in upoštevanju podpiranja standardov;
- Storitve so samostojne – Ker niso povezane z drugo aplikacijsko kodo, pridobimo na enostavnosti razumevanja in razširljivosti;
- Storitve delijo med seboj sporočila s podatki o vsebini in ne implementacij – Vse kar mora aplikacija poznati o storitvi, je njena pogodba;
- Skladnost storitev temelji na politiki – Dodatna pravila med odjemalcem in storitvijo, ki se nanašajo na varnost in zanesljivost, morajo biti definirana izven dokumentacije.

### 2.1.1 STORITVE

Storitve so glavni koncept SOA. So samostojne aplikacije, ki opravljajo neko operacijo. Zaradi njihove avtonomnosti, jih lahko uporabljamo tudi izven konteksta aplikacije – lahko jih povezujemo med seboj in združujemo v večje procese [2]. Do njih dostopamo preko vmesnikov, ki vsebujejo operacije. Realizirane so preko uporabe množice tehnoloških standardov [6]:

1. UDDI (angl. Universal Description, Definition and Integration) – Omogoča iskanje ustreznih storitev v repozitoriju;
2. WSDL (angl. Web Services Description Language) – Za dodaten opis storitev sistema;



Slika 2.1: Arhitektura spletnih storitev [26].

3. SOAP (angl. Simple Object Access Protocol) – Za pošiljanje sporočil med odjemalcem in ponudnikom - uporabo storitev;
4. WS-\* standardi (angl. Web Service) – Za doseganje atributov kvalitete;
5. XML (angl. eXtensible Markup Language) – Določa obliko kodiranja podatkov, ki se prenašajo;
6. http (angl. HyperText Transfer Protocol) – Protokol, ki določa transport podatkov.

Prvi od dveh pomembnih vidikov SOA je torej izpostavljanje funkcionalnosti. Imenujemo ga tudi princip od spodaj navzgor (angl. Bottom-Up Approach) in je osnova, da lahko nadaljujemo z drugim korakom – kompozicijo storitev. Kompozicija predstavlja pristop od zgoraj navzdol (angl. Top-Down Approach).

### 2.1.2 KOMPOZICIJA STORITEV

Kompozicija predstavlja združevanje osnovnih storitev v nove, kompleksnejše in večje storitve. Ta se izvede kot [5] orkestracija, pri čemer so storitve centralno vodene – boljše za zasebne procese, ko imamo na voljo vse informacije, ali kot koreografija, kjer centralnega vodenja ni – boljše za javne procese. S kompozicijo storitev se realizira enoten nivo, imenovan procesni nivo. Kot primer navedimo storitev internetne prodaje, ki združuje osnovne storitve – izbiro izdelka, izbiro metode plačila, pregled košarice . . . Ob odločitvi, da se k metodam plačila doda nova plačilna možnost, ki doslej še ni bila na voljo, se v ta namen razvije nova osnovna storitev in doda v že obstoječ poslovni proces. Prednosti realizacije tega nivoja se odražajo v agilnosti sistema. Te so [2]:

- Realnočasovni vpogled v poslovne procese;
- Izboljšana odzivnost na poslovne spremembe;
- Enostavnost definicije in sprememb poslovnih procesov.

Procesni vidik realizacije SOA upošteva poslovne procese kot najvažnejšo sestavino, njihovo pripadajoče sodelovanje, izmenjavo informacij in aktivnosti pa izrazimo z uporabo ustreznih tehnologij, predvsem ESB (angl. Enterprise Service Bus) – ogrodje, ki ponuja transparentnost komunikacije in podporo varnosti, transakcijam in podobnim storitvam; BPEL – Za definicijo procesnega toka in koordinacijo ter BPMN – Za modeliranje in tudi izvajanje poslovnih procesov [2].

## 2.2 UPRAVLJANJE POSLOVNIH PROCESOV

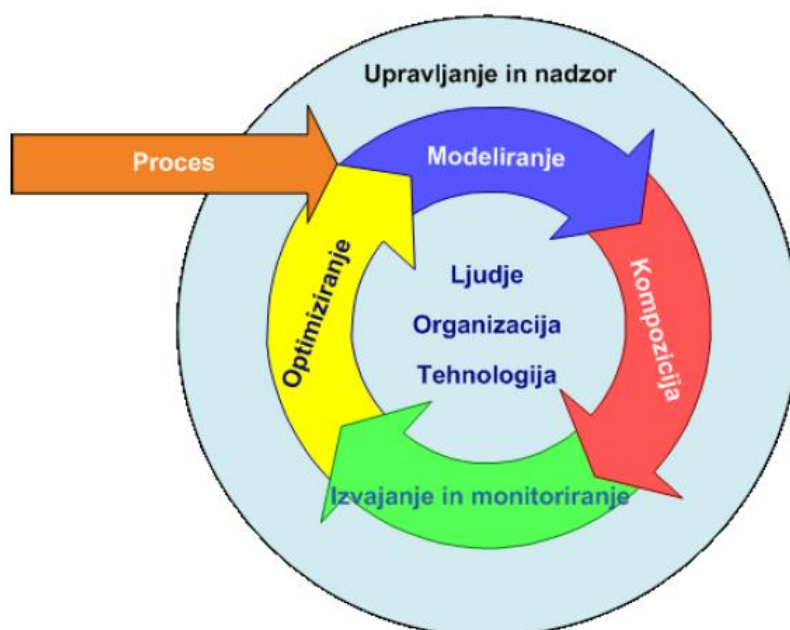
Upravljanje poslovnih procesov BPM predstavlja most med poslovno filozofijo in IT. Na dolgi rok spreminja organizacijo iz funkcionalne v procesno

naravnano. Je metoda za spodbujanje poslovne učinkovitosti in inovativnosti, agilnosti in integracije s tehnologijami [1]. Glavni cilji BPM so [7]:

- Optimizacija celotnih poslovnih procesov (privatnih in javnih), od začetka pa vse do konca;
- Zagotovitev transparentnosti poslovnih procesov v vseh stopnjah razvoja;
- Skrb za usklajevanje modela poslovnega procesa z izvajalno različico;
- Sposobnost hitre optimizacije procesov in neprestano izboljševanje;
- Možnost merljivosti prispevka novosti k uspehu združbe.

### 2.2.1 ŽIVLJENJSKI CIKEL POSLOVNIH PROCESOV

1. Procesno modeliranje - Življenjski cikel se prične s procesnim modeliranjem, v katerem poslovni analitik opazovani poslovni ali nov proces definira z v ta namen določeno notacijo. Definirajo se tudi poslovna pravila, omejitve in ključni kazalniki poslovanja – KPI (angl. Key Performance Indicators). Obstaja več vrst notacij (npr. EPC, angl. Event Process Chain ali UML, angl. Unified Modeling Language), s SOA pa se najbolj usklajuje BPMN. Njeni največji prednosti sta, da je enostavno razumljiva in da nudi možnost hitre prevedbe v izvršljivo obliko (BPEL, BPMN 2.0). Učinkovitost nastalega modela se preveri s simulacijo, ki omogoča identifikacijo možnih napak in opozori na potencialne preobremenitve;
2. Implementacija - Cilj te faze je pretvoriti nastali model poslovnega procesa v izvršljivo obliko. Ker IT nudi podporo celotnemu poslovnemu procesu in ne le določenim funkcijam, dobimo z implementacijo boljši pregled nad celotnim izvajanjem. Pri implementaciji po principu SOA



Slika 2.2: Življenjski cikel poslovnih procesov [27].

je ključna preslikava BPMN modela v izvršljivo obliko. Med preslikovanjem zaradi zahteve po boljši integraciji med poslovnim procesom in programsko kodo ne sme priti do nesoglasij in posledično do izgube informacij. Od verzije BPMN 2.0 naprej je ta strah odveč, saj predstavlja BPMN 2.0 tako jezik za modeliranje kot tudi izvajalni jezik;

3. Izvajanje in nadzorovanje procesa - V tej fazi izvajamo poslovni proces in smo pozorni na to, ali se izvaja, kot smo predvideli. Kot ključni element te faze se pokaže procesni strežnik, ki nam omogoči izvajanje. Če pride do odstopanj, lahko proces dopolnimo ali spremenimo. Z izvajanjem pridobivamo pomembne podatke o procesu (npr. o trajanju aktivnosti ali porabi virov), ki nam bodo koristili v naslednji fazi. Pri tem nam zelo pomaga BAM (angl. Business Activity Monitoring), preko katerega imamo celoten pregled nad sprotnim izvrševanjem procesa;

4. Spremljanje rezultatov in optimizacija - V tej fazi uporabimo podatke iz prejšnje faze, jih primerjamo s KPI in poskušamo ugotoviti, če in kateri del poslovnega procesa bomo izboljšali. SOA nam omogoča orodja, ki upoštevajo podatke iz BAM.

Za doseganje izboljšav moramo cikle ponavljati. Po zadnji fazi (spremljanje rezultatov in optimizacija) se zopet začne modeliranje. Le tako lahko gremo v korak s časom in dosežemo zastavljene cilje!

## **2.3 POVEZAVA MED SOA, BPM IN POSLOVNIMI PRAVILI**

### **2.3.1 POVEZAVA SOA IN BPM**

BPM predstavlja način organizacije poslovnih, SOA pa tehničnih zmožnosti [8]. SOA podpira cel BPM življenjski cikel, največji vpliv pa je viden v fazah implementacije (BPEL, BPMN 2.0) ter izvajanja in nadziranja (procesni strežnik, BAM). S SOA načinom pridemo do večje fleksibilnosti poslovnih procesov in boljših možnosti za uspešno optimizacijo sistemov. Omogoča nam bolj učinkovito poslovno prilagajanje in spremembe v več manjših in hitrejših korakih za razliko od tradicionalnega pristopa. Ker so poslovni procesi dinamični in ker ljudje ne marajo velikih sprememb, je tako omogočen lažji in učinkovitejši prehod iz trenutnega (angl. As-Is) v prihodnje (angl. To-Be) stanje.

### **2.3.2 POVEZAVA SOA IN BPM S POSLOVNIMI PRAVILI**

V želji po izvajanju nadzora in avtomatizaciji odločitev, ki se nanašajo na poslovni proces, dobimo z upravljanjem poslovnih procesov pregled nad procesnim tokom aktivnosti in informacij. Da pa bi do avtomatizacije odločitev prišlo, morajo zato obstajati konsistentna pravila in kriteriji, preko kate-

rih se lahko vrši avtomatizacija odločitev (in posledično aktivnosti same) v poslovnem procesu. Takšnim pravilom, ki vplivajo na odločitve v poslovnem procesu in omejujejo vidike poslovanja, pravimo poslovna pravila [15]. Mnogo proizvajalcev predstavlja BPM in upravljanje poslovnih pravil - BRM (angl. Business Rules Management) kot dve alternativni možnosti za doseganje fleksibilne, agilne arhitekture [30]. V resnici imata omenjeni disciplini veliko skupnega. Sta komplementarni in se lahko odlično dopolnjujeta, še posebej, če ju uporabimo v povezavi s SOA. Glavno idejo povezave BPM in poslovnih pravil s SOA predstavlja striktno ločevanje odločitvene logike od procesne logike. Procesna logika se nanaša predvsem na nadzorovanje zaporednega izvajanja procesnih aktivnosti in je implementirana kot del BPM, medtem ko pod pojem odločitvene logike spada implementacija poslovne logike, ki predstavlja politiko in principe organizacije. SOA tako spreminja način sodelovanja med BPM in BRM saj poslovna pravila, ki se navezujejo na odločanje, niso vsebovana v samem BPM, ampak so predstavljena kot storitve, dostopne vsem delom aplikacijskega sistema. Kot smo videli, so s sinergijo med omenjenimi disciplinami omogočeni pogoji za doseganje agilnosti poslovnih procesov. Več o poslovnih pravilih in sistemih za njihovo upravljanje bomo povedali v nadaljevanju.



## Poglavje 3

# MODELIRANJE POSLOVNIH PROCESOV

Glavni cilj faze procesnega modeliranja je razvoj realnega (As-Is) procesnega modela, kateri natančno prikaže procesni tok združbe. Ker je to prvi izmed zaporednih rezultatov upravljanja poslovnih procesov, je njegova natančnost še toliko pomembnejša – na njemu temeljijo nadaljnje ugotovitve in stvaritve. Pred modeliranjem si moramo odgovoriti naslednja vprašanja [1]:

- Katere aktivnosti se izvajajo v poslovnem procesu?
- Kdo izvaja aktivnosti?
- Kateri poslovni dokumenti se izmenjujejo?
- Katere vloge so potrebne v poslovnem procesu?
- Kako lahko spremenimo proces v prihodnje?

Z odgovori na omenjena vprašanja dobimo podroben vpogled v delovanje združbe ter možnost za razvoj podrobnega modela ter kasneje učinkovite aplikacijske podpore. Za modeliranje je potrebna tudi enotna in enostavno razumljiva notacija. V preteklosti so se uporabljali mnogi zapisi (denimo EPC in UML), trenutno pa je najprimernejši zapis BPMN, ki predstavlja procesno usmerjen pogled na poslovni proces.

### 3.1 BPMN 2.0 – PODPORA MODELIRANJU IN IZVAJANJU

BPMN je najbolj splošno razumljiva notacija za zapis poslovnih procesov doslej. Nastala je leta 2004, naprej pa se je razvijala pod okriljem skupine OMG (angl. Object Management Group) [1]. Njegovi največji prednosti sta, da je razumljiv širokemu krogu ljudi (od poslovnih analitikov do tehničnih razvijalcev) in da ga je mogoče (do verzije BPMN 2.0), s pomočjo ad hoc orodij (BPMS, angl. Business Process Management Systems), pretvoriti v izvajalni jezik. Trenutna različica jezika je torej 2.0 in je izšla leta 2011. S to različico se je spremenil tudi pomen kratice BPMN, ta po novem pomeni Business Process Modeling and Notation. Vizija BPMN 2.0 je imeti eno samo specifikacijo za podporo notaciji in izvajanju procesov [10]. BPMN ni več le jezik za modeliranje, ampak tudi izvajalni jezik. Ker doslej metamodela, namenjena modeliranju (BPMN) in izvajanju (BPEL), nista bila identična, je lahko prihajalo do izgube informacij pri preslikovanju enega modela v drugega. BPMN 2.0 nam zaradi operacijske semantike, ki je blizu BPEL, omogoča premoščanje teh razlik, ki bi sicer nastale pri pretvarjanju med BPMN in BPEL. Posledici sta enostavnejše sodelovanje med poslovnim analitikom in tehničnim razvijalcem procesa, saj oba delata na istem modelu procesa, ter zmanjšan semantični prepad med procesnimi modeli in aplikacijsko kodo [10]. Nekaj ostalih novosti, ki jih še prinaša BPMN 2.0 [9]:

- Podpora pogledom na model iz različnih perspektiv (diagrama koreografije in pogovora);
- Tipi skladnosti orodja za podporo zahtevam BPMN;
- Podporo izvajanju poslovnih pravil v poslovnih procesih;
- Natančen opis semantike, ki je potrebna za interpretacijo in izvajanje BPMN;

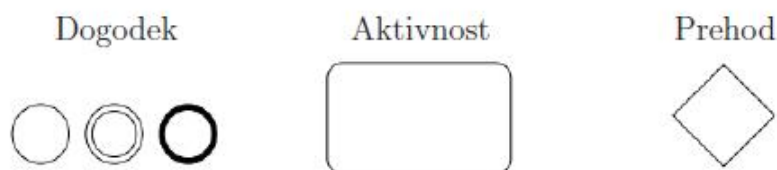
- XPDL (angl. XML Process Definition Language) – standard za izmenjavo poslovnih procesov;
- Definicija razširitvenega mehanizma, ki omogoča tako razširitev procesnega modela, kakor tudi grafične razširitve.

## 3.2 OSNOVNI GRADNIKI

Notacija BPMN definira diagram BPD (angl. Business Process Diagram), ki je zasnovan na konceptu diagrama poteka. Model poslovnega procesa je tako predstavljen kot mreža grafičnih objektov, aktivnosti in nadzornikov poteka, ki določajo vrstni red izvajanja aktivnosti [11]. Njegovi osnovni elementi so razvrščeni v štiri kategorije, katere predstavljajo:

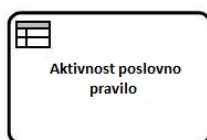
1. elementi procesnega toka,
2. povezovalni elementi,
3. steze,
4. artefakti.

**Elemente procesnega toka** predstavljajo dogodki, aktivnosti in prehodi. Dogodki vplivajo na potek procesa in imajo ponavadi nek povod (prožilec, ki dogodek sproži) in posledico. Obstajajo trije tipi dogodkov, ki se razlikujejo glede na mesto v poslovnem procesu. To so začetni, vmesni in končni dogodek. Jedro poslovnega procesa predstavljajo aktivnosti, ki označujejo enoto dela. Aktivnosti so lahko atomarne ali sestavljene. Atomarne so dokončne, sestavljene pa se delijo še naprej, na podprocesse, opravila, transakcije in klicne aktivnosti. Vejitve in združevanja zaporednega poteka aktivnosti podajajo elementi prehodov. Tudi teh obstaja več vrst, od prehoda, ki omogoča izbiro le ene poti, do takšnega, ki omogoča vzporedno izvajanje. Osnovni simboli za predstavitev gradnikov prve skupine so naslednji [12]:



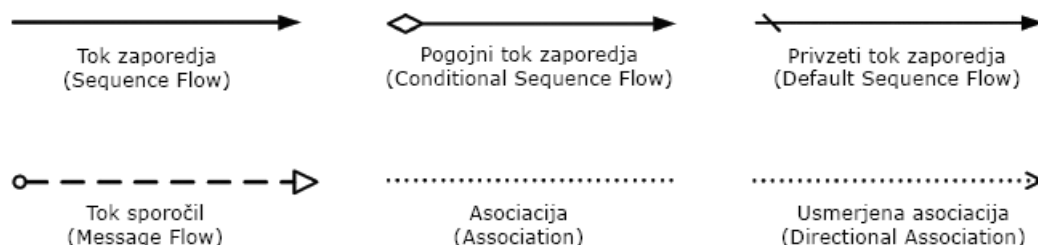
Slika 3.1: Elementi procesnega toka [28].

Na tem mestu velja izpostaviti še posebno vrsto aktivnosti, ki spada med elemente procesnega toka - poslovno pravilo (angl. Business Rule Task). Ta vrsta opravila je novost v jeziku BPMN 2.0, preko nje se izvede klic sistema za upravljanje s poslovnimi pravili (angl. Business Rule Management System). V BRMS se izvršijo odločitve, ki se navezujejo na poslovna pravila združbe. Opravilo torej deluje kot vhodna točka v BRMS in kot prejemnik izračunov ter njihov posrednik za nadaljnje izvajanje poslovnega procesa, samo po sebi pa v procesnem modelu ne izvede nobene akcije [31].



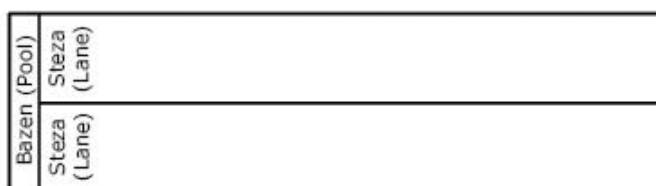
Slika 3.2: Aktivnost poslovno pravilo.

**Povezovalni elementi** imajo nalogo med seboj povezovati elemente procesnega toka, elemente steza ali artefaktov. Tok zaporedja prikazuje vrstni red izvajanja aktivnosti v procesu, tok sporočil izmenjavo sporočil med dvema različnima udeležencema procesa (poslovnima entitetama ali vlogama), pogojni in privzeti tok zaporedja se uporabljata za preverjanje pogoja pred prehodom na naslednjo aktivnost ali označevanje običajne poti, tok asociacije pa povezuje podatke, besedila in druge izdelke z elementi procesnega toka in se uporablja za prikaz vhodov in izhodov v/iz posameznih aktivnosti.



Slika 3.3: Povezovalni elementi [28].

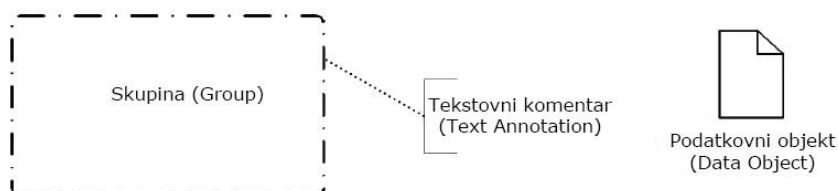
Za organizacijo aktivnosti v vizualno ločene kategorije, z namenom izpostavitve različnih funkcionalnih zmožnosti ali odgovornosti, se uporablja koncept **steze** (angl. Swimline). Diagram BPD sestavljata dva tipa stez: bazen (angl. Pool) in proga (ang. Lane). Bazeni predstavljajo udeležence (združbe) v procesu, hkrati pa deluje tudi kot grafični vsebnik za ločevanje različnih množic aktivnosti. Proga predstavlja posamezno entiteto v združbi in se uporablja za organizacijo in kategorizacijo aktivnosti. Bazeni se uporabljajo, ko diagram vključuje dve ali več ločenih entitet oziroma udeležencev.



Slika 3.4: Prikaz bazena in prog [28].

Aktivnosti, vključene v ločene bazene, se obravnavajo kot ločeni procesi, kar pomeni, da tok zaporedja ne sme prečkati meje bazena. Komunikacija med aktivnostmi v različnih bazenih je namreč prikazana z uporabo toka sporočil.

**Artefakti** se uporabljajo za podajanje dodatnih informacij o poslovnem procesu, na njegov potek izvajanja pa ne vplivajo. Za modeliranje v BPMN 2.0 so specificirani naslednji gradniki [12]:



Slika 3.5: Prikaz specificiranih artefaktov v BPMN 2.0 [28].

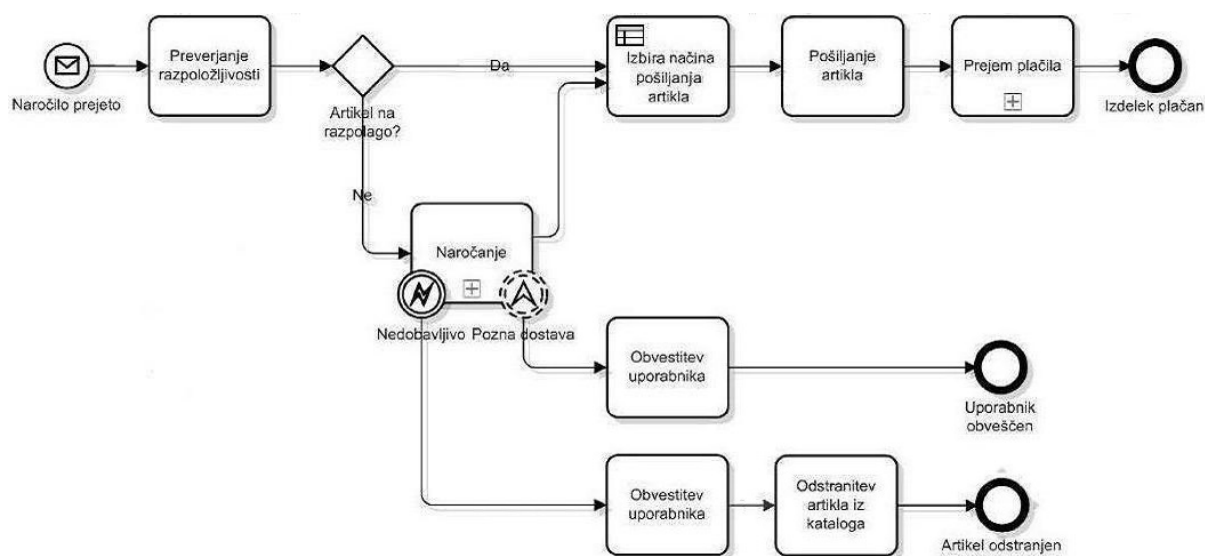
BPMN model mora biti z uporabo gradnikov čim bolj nazoren in lahko berljiv. Da bi to dosegli, se je potrebno dogovoriti o tem, do kakšnega nivoja podrobnosti ga je potrebno razčleniti, obenem pa mora biti sintaktično pravilen s pravilno uporabljenimi elementi in njihovimi povezavami. Na posameznem nivoju procesa naj bi bilo prikazanih  $7 \pm 2$  aktivnosti. Preveč obsežne aktivnosti namreč povečujejo kompleksnost poslovnega procesa in zmanjšujejo ponovno uporabljivost in fleksibilnost. Poleg osnovnega (najverjetnejšega delovnega toka) moramo predvideti še nevsakdanje situacije v obliki izjem in kompenzacij.

### 3.3 PRIMER MODELIRANJA

Kot primer poslovnega procesa navajamo proces sprejemanja naročil za spletno trgovino. Proces se začne s sporočilnim začetnim dogodkom, ki se aktivira v primeru prejetega sporočila, natančneje naročila. Nato sledi atomarna aktivnost preverjanja razpoložljivosti artiklov. Če je želeni artikel na razpolago, se v prehodu izbere zgornja veja, nakar pred zaključkom procesa v stanju "izdelek plačan", sledijo še tri aktivnosti. Prva aktivnost je podtipa opravilo in predstavlja klic za izvajanje poslovnih pravil organizacije, kjer se glede na določena poslovna pravila določi način pošiljanja izbranega arti-

kla. Druga je atomarna aktivnost "pošiljanja artikla", kateri sledi podproces "prejem plačila".

Če pa artikla ni na razpolago, se izvede podproces "naročanje". V optimalnem primeru, se bo zahtevani izdelek naročil, proces pa se bo nadaljeval v že prej opisanih stanjih od aktivnosti "izbira načina pošiljanja artikla" naprej. Ker pa se lahko v procesu naročanja zgodi kaj nepredvidenega, moramo predvideti tudi te situacije, natančneje:



Slika 3.6: Primer poslovnega procesa sprejemanja naročil.

- Izdelek je lahko nedobavljiv zato moramo to predvideti z vmesnim dogodkom napake, ki prekine podproces naročanja. Tako se proces nadaljuje z aktivnostma "obvestitev uporabnika" in "odstranitev artikla iz kataloga" in se konča s končnim stanjem "artikel odstranjen";
- Naročanje izdelka pa se lahko tudi zavleče. V tem primeru uporabimo vmesni dogodek eskalacijo, ki ne prekine naročanja, ampak v tem primeru obvesti uporabnika o nastali situaciji v aktivnosti "obvestitev uporabnika". Ta veja procesa se zaključi v končnem stanju "uporabnik obveščen".



## Poglavje 4

# UPORABA POSLOVNIH PRAVIL V PROCESIH

Človeka od ostalih živih bitij loči sposobnost abstraktnega mišljenja in svoboda v vseh pogledih, tudi pri odločanju. Ko se moramo odločati, to storimo tudi na podlagi čustev in logike. Ker imamo različne načine pogleda na situacijo, v kateri smo se znašli, ne reagiramo vsi enako ob enakih pogojih. Tako je tudi v poslovanju, kjer odločanje temelji na znanih dejstvih. Če se želimo dobro, kvalitetno odločiti, morajo naše odločitve imeti zaledje v kvalitetnih informacijah in pravilih, ki so nam pri odločanju na voljo. V povezavi s poslovnimi odločitvami, so ta dejstva imenovana poslovna pravila. Služijo nam kot vodilo, katero vpliva na skupinsko obnašanje organizacij in informacijskih sistemov. Na žalost pa se prepogosto dogaja, da so poslovna pravila neznana ali nedostopna. Lahko so izražena tacitno – v glavah odgovornih ali vgrajena v aplikacijsko kodo, za katero ni prave razlage – dokumentacije. V takih primerih, ko pravila odločanja niso eksplicitno zapisana in poznana, se morajo odgovorni ljudje v združbah odločati po svojih načelih. Tako pogosto prihaja do nekonsistentnosti odločitev, informacijski sistemi pa so težavni za nadgrajevanje in spreminjanje. V današnjem času, ko je ritem (poslovnih) sprememb zelo hiter, se moramo ljudje hitro prilagajati in učiti novih stvari. To pomeni, da moramo sprejemati veliko število odločitev oz. svoje rešitve

dopolnjevati, zato je zgoraj omenjeni način vgradnje odločitev v aplikacijsko kodo neprimeren. Zaradi tega moramo sisteme zasnovati na drugačen način, in sicer morajo biti poslovna pravila ločena od ostalih komponent. Tako vsi udeleženci vedo, da obstajajo, kakšna so, od kod izvirajo in kako se jih da izboljšati. Kaj torej so poslovna pravila? Opišemo jih lahko z naslednjimi izjavami [16]:

1. Poslovna pravila ne vplivajo zgolj na ljudi znotraj združbe, ampak tudi na ljudi zunaj posameznega sistema;
2. Poslovna pravila učijo in vlivajo samozavest; Vodijo k večji produktivnosti pri odločanju;
3. Poslovna pravila so osnova za pravilno odločanje;
4. Poslovna pravila vplivajo na obnašanje in pripomorejo h doseganju skupnih ciljev;
5. Poslovna pravila motivirajo udeležence;
6. Poslovna pravila določajo verjetnost doseganja zastavljenih ciljev;
7. Poslovna pravila so vzvodi preko katerih se organizacija lahko razvija.

Obstaja tudi več različnih definicij, ki so odvisne od avtorjev in obdobja, v katerem so nastale. Prvi je pojem definiral Daniel S. Appleton leta 1984. Poslovna pravila je predstavil kot "Eksplicitno navedene omejitve, ki obstajajo znotraj poslovne ontologije." Po drugi od definicij [13] poslovna pravila združb "definirajo poslovne omejitve. Nanašajo se na ljudi, procese, poslovanje ali informacijske sisteme in pomagajo organizacijam dosegati zastavljene cilje." Naslednja [14] definicija predstavlja poslovna pravila kot "formalne pogoje, ki upravljajo in avtomatizirajo poslovne dogodke tako, da so ti sprejemljivi za organizacije." Še ena definicija jih deklarira [15] kot "izjave, ki omejujejo nek vidik poslovanja in so namenjena opredelitvi poslovne strukture oz. vplivu ali nadzoru poslovnega obnašanja. Poslovnih pravil ne

moremo členiti dalje – so atomarna.” Kot vidimo, ne obstaja enotna definicija, vso bistvo poslovnih pravil pa nam s svojo definicijo predstavi Barbara Von Halle [14]. Bistveni pojmi njene definicije so:

- Formalni pogoji, kar pomeni definirane procese, opravila, vloge in produkte;
- Upravljanje in avtomatizacija – Upravljanje se nanaša na zbiranje, objavljanje, pregledovanje in razvoj poslovnih pravil, avtomatizacija pa na njihovo izvajanje;
- Sprejemljivost za organizacije – Pomeni, da poslovanje poteka kot je bilo zamišljeno.

## 4.1 VRSTE POSLOVNIH PRAVIL

Tako kot pri definiciji, obstaja tudi pri določanju vrst poslovnih pravil več načinov. Delitve so si večinoma podobne, praviloma pa vsebujejo vsaj tri kategorije, po katerih se klasificirajo:

- V prvo kategorijo spadajo pravila, ki predstavljajo posamezne pojme iz poslovnega okolja;
- V drugo kategorijo spadajo pravila, ki se navezujejo na razmerja med osnovnimi pojmi;
- V tretjo kategorijo pa spadajo pravila, ki se dotikajo omejitev v zvezi s strukturo in izvajanjem poslovnih procesov.

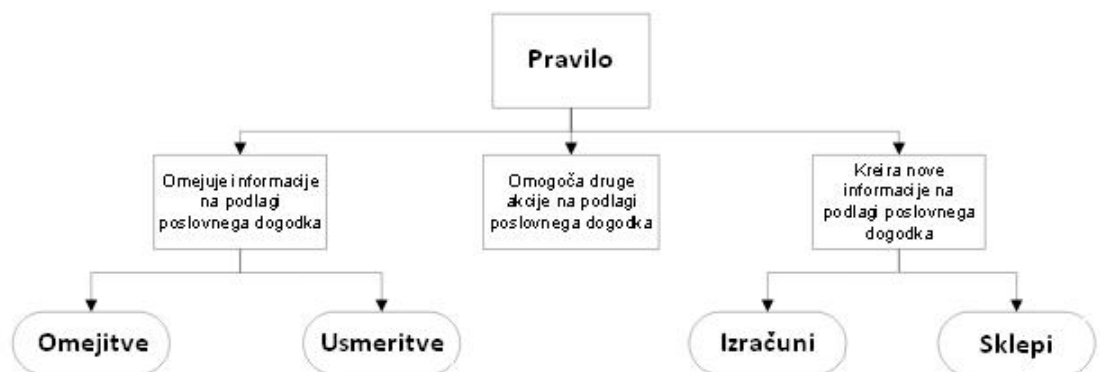
Na spodnji sliki 4.1 je prikazana ena od delitev poslovnih pravil, ki je predstavljena v [16]:

1. **Termini** (angl. Terms) – So samostalniki ali samostalniške fraze, ki predstavljajo dogovorjeno definicijo za poslovni sistem. Lahko so pojmi, njihove lastnosti ali nabor vrednosti. Kot primer termina lahko podamo pojem stranke;



Slika 4.1: Prikaz klasifikacije poslovnih pravil [16].

2. **Dejstva** (angl. Facts) – Dejstva so izjave, ki povezujejo termine v smiseln poslovni kontekst. Dejstva skupaj s termini predstavljajo semantiko, ki je uporabljena za poslovnimi pravili. Primer dejstva – Stranka lahko odda naročilo;
3. **Pravila** (angl. Rules) - Predstavljena so kot omejitve ali izvajalna logika, ki za vhode uporabi informacije in kreira izhode ter proži dogodke. Sestavljajo jih tri kategorije (kot prikazuje slika 4.2):



Slika 4.2: Prikaz nadaljnje razdelitve pravil [16].

- 3.1. Kategorija **omejitev** (angl. Constraint) in **usmeritev** (angl. Guideline) - Omejitve vodijo obnašanje poslovnih dogodkov. Izražene

so kot obvezna pravila, ki se jih v procesu mora upoštevati. V nasprotnem primeru se proces ustavi. Primer omejitve: Stranka ne sme imeti naenkrat naročenih več kot deset izdelkov. Usmeritve so predstavljene kot izjave, ki izražajo varnostna opozorila glede poslovnih dogodkov. Za razliko od omejitev ne prekinjajo procesa, ampak zahtevajo poseg odgovorne osebe. Primer usmeritve: Stranka naj ne bi imela naenkrat več kot desetih naročenih izdelkov;

- 3.2. Kategorija **prožilcev** (angl. Action Enabler) – Sestavljajo jih izjave, ki na podlagi izpolnjenih ali neizpolnjenih pogojev poslovnih dogodkov prožijo nadaljnje akcije kot so pošiljanje sporočil ali začenjanje novih aktivnosti. Primer prožilcev: Če izdelkov ni na voljo, o tem obvesti stranko;
- 3.3. Kategorija **kreiranja novih informacij na podlagi poslovnih dogodkov**. Sem spadajo izračuni (angl. Computation) in sklepi (angl. Inference). Pod izračune spadajo postopki za pridobivanje matematično pridobljenih vrednosti. Primer: Skupna cena je sestavljena iz vsote vrednosti posameznega izdelka in pripadajočega davka. Sklepi so izjave, ki na osnovi ene vrednosti določajo novo. Največkrat so izraženi v če – potem (angl. If-Then) obliki. Primer sklepa: Če ima stranka opravljen vozniški izpit, potem si lahko izposodi avtomobil.

## 4.2 UPORABA POSLOVNIH PRAVIL

V današnjem času je pritisk na organizacije velik. Največkrat se pokaže v obliki spreminjanja poslovnih procesov, do česar lahko pride zaradi pojavljanja konkurence, nove zakonodaje, želje po ohranitvi in pridobivanju novih strank, ko podporni sistemi ovirajo spremembe poslovanja... Posledica teh sprememb so tudi modifikacije v podpirajočih aplikativnih sistemih. Pristop s poslovnimi pravili se pri razvoju in vzdrževanju sistemov pokaže kot najbolj

zaželen in praktičen, saj na ta način zgradimo boljše in na spremembe bolj odporne sisteme hitreje kot z ostalimi pristopi [18, 19]. Prednosti razvoja takšnih sistemov se najbolj odražajo v [16]:

- Enostavnosti – Pristop s poslovnimi pravili je preprost za razumevanje tako za poslovno kot tudi za tehnično osebje, saj je koncept pravil precej intuitiven. Poslovni ljudje namreč niso preveč zainteresirani za razumevanje podatkovnih ali objektnih modelov, so pa definitivno željni razumevanja poslovne politike in pravil združbe. Preko le-teh namreč krmilijo v poslovnem svetu;
- Majhnem številu poznavanja potrebnih pojmov za jasno interpretacijo – Jedro pristopa s poslovnimi pravili predstavljajo pravila sama, ki jih lahko na povezujemo na osnovi odločitev. Tako dobimo vzorce pravil, ki temeljijo na podobnih izjavah (družine pravil) in omogočajo nadaljnjo analizo;
- Neodvisnosti pravil – Pravila so izražena s sintakso, ki je neodvisna od tehnologije in aplikacij, ter se nanaša na poslovno in ne na sistemsko-tehnično vsebino;
- Ponovni uporabljivosti pravil – Množico poslovnih pravil lahko implementiramo samo enkrat, ampak jo uporabimo večkrat v različne namene. Pravila je tako mogoče tudi testirati pred razvojem preostalega dela aplikacije;
- Poenostavljenem načrtovanju sistemov – Možno je ločiti osnovni procesni tok od izvrševanja poslovnih pravil;
- Dinamiki pravil – Upravljalca poslovnih pravil lahko spreminja pravila na enostaven način in tako omogoča poslovno prilagodljivost;
- Učinkovitosti – Komercialni sistemi za upravljanje s poslovnimi pravili so posebej zasnovani za to, da lahko upravljajo in izvajajo poslovna pravila. Tako vsebujejo notranjo logiko za doseganje najboljše

učinkovitosti izvajanja, prav tako pa se del aplikacijske kode generira avtomatsko, kar zmanjšuje možnost programerskih napak. Ti lahko tako več časa posvečajo drugim nalogam.

Kaj omogoča te prednosti? Pristop s poslovnimi pravili temelji na štirih enostavnih principih, ki skupaj omogočajo opisane prednosti. Ti principi so tako imenovani STEP principi in pomenijo [17]:

1. Ločevanje (angl. *Separate*) – Pomeni, da je potrebno ločevati poslovna pravila od ostalih zahtev in sistema samega. Primarno nam to omogoča ponovno uporabljivost pravil in njihovo spreminjanje neodvisno od ostalih delov sistema;
2. Sledljivost (angl. *Trace*) – Če želimo zadostiti principu sledljivosti, moramo ohranjati povezavo med poslovnim pravilom in dvema smerema. Prva smer je od pravila proti njegovemu izvoru. Pravilo ima svoj izvor v smislu poslovne motivacije, ki jo predstavljajo cilji, strategije, taktike, politike in metrike, preko katerih se izmeri učinkovitost pravila v skladu z delovanjem združbe. Tako se lahko skozi čas ugotavlja, ali je pravilo še aktualno, ali pa ga bo potrebno v prihodnje spremeniti. Druga smer pa je od pravila proti njegovi implementaciji. Ta smer nam omogoča zaznavanje vpliva spreminjanja poslovnih pravil na mestih, kjer se pravilo izvaja;
3. Zapisovanje (angl. *Externalize*) – Pomeni izražanje pravil na način, ki ni razumljiv zgolj tehničnemu, ampak tudi poslovnemu osebju. To storimo zato, da vsi udeleženci vedo, kakšna so pravila, na kaj vplivajo, kakšna je njihova učinkovitost in da jih je mogoče optimizirati;
4. Pripravljenost na spremembe (angl. *Position*) – Pomeni, da mora biti pravilom vedno omogočeno enostavno in hitro spreminjanje. To dosežemo z implementacijo pravil v tehnologiji, ki to omogoča (denimo komercialni sistemi za upravljanje s poslovnimi pravili), istočasno pa

moramo paziti na to, da spremembe pravil ne povzročijo dragih in časovno potratnih sprememb v podatkovni bazi.

<b>STEP PRINCIP</b>	<b>NAMEN STEP PRINCIPA</b>
<b>Ločevanje</b>	<ul style="list-style-type: none"> <li>- Ponovna uporaba pravil</li> <li>- Uporaba posebnih tehnik za optimizacijo kvalitete pravil</li> <li>- Spreminjanje pravil neodvisno od drugih delov sistema</li> </ul>
<b>Sledljivost</b>	<ul style="list-style-type: none"> <li>- Določanje veljavnosti pravila</li> <li>- Določanje vpliva spreminjanja poslovnih pravil</li> </ul>
<b>Zapisovanje</b>	<ul style="list-style-type: none"> <li>- Možnost seznanitve s pravili</li> <li>- Možnost preverjanja pravil</li> </ul>
<b>Pripravljenost na spremembe</b>	<ul style="list-style-type: none"> <li>- Omogočanje enostavnega spreminjanja</li> <li>- Omogočanje hitrega spreminjanja</li> </ul>

Tabela 4.1: STEP principi in njihov namen [16].

### 4.3 RAZVOJ POSLOVNIH PRAVIL

Da bi lahko izkoristili prednosti, ki jih pristop s poslovnimi pravili ponuja, je potrebno za razvoj pravil uporabiti metodologijo, ki nam to omogoča. V ta namen je organizacija ILOG leta 2003 [21, 22] izdala agilno metodologijo za razvoj poslovnih pravil ABRD (angl. Agile Business Rule Development), ki podrobno definira aktivnosti za razvoj poslovnih pravil od njihovega začetka, pa vse do konca. ABRD kot inkrementalna in iterativna metodologija za izkoriščanje prednosti pristopa s poslovnimi pravili, upošteva načela agilnega manifesta za razvoj programske opreme [20], katera določajo, da so:

- Posamezniki in interakcije pred procesi in orodji;
- Delujoča programska oprema pred vseobsežno dokumentacijo;
- Sodelovanje s stranko pred pogodbenimi pogajanja;
- Odziv na spremembe pred togim sledenjem načrtom.

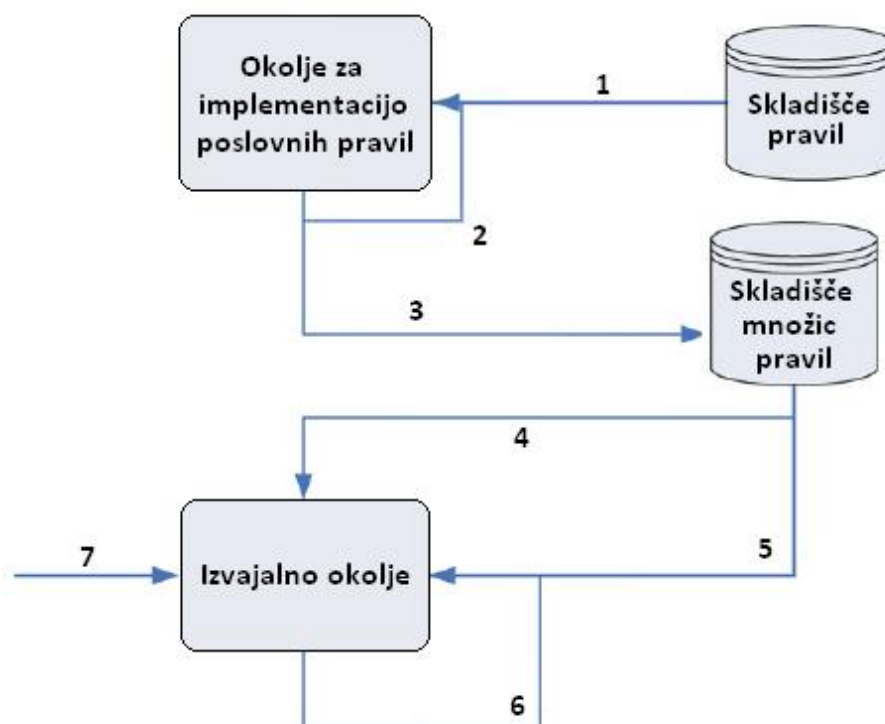
### 4.3.1 AKTIVNOSTI RAZVOJA POSLOVNIH PRAVIL

Tako metodologija ABRD aktivnosti razvoja razdeli v cikle, ki omogočajo agilni razvoj. Med aktivnosti so vključeni [14]:

1. Odkrivanje pravil (angl. Rule Discovery) – Aktivnost se redno izvaja med razvojnim procesom. Najbolj unikaten vidik te faze je, da začnemo zaznavati poslovne dogodke kot množico odločitev. Ti so bili doslej zapisani v različnih dokumentih, skriti v glavah zaposlenec, zakopani v izvajalnih kodi itd. Pravila niso bila shranjena na enem (preglednem) mestu. Izdelki te aktivnosti so tako dokumenti z zapisanimi pravili, ki se uporabljajo v sistemu;
2. Analiza pravil (angl. Rule Analysis) – Cilj te aktivnosti je razumevanje pomena napisanih pravil. V ta namen moramo odstraniti semantične nepravilnosti in vse ostale nejasnosti in tako pripraviti pravila za kasnejšo implementacijo. To pomeni, da moramo transformirati pravila na atomarne in samostojne elemente, odstraniti prekrivanja pravil in njihovo redundanco. Primer transformacije poslovnih pravil na atomarne enote: Zavarovalnica ne vrača stroškov za medicinske storitve v tujini, če je bil podan zahtevek za povračilo več kot eno leto po nastanku stroškov ali pa je vlagatelj preživel v preteklem letu v tujini več kot 182 dni. Omenjeno izjavo moramo torej razčleniti v dve atomarni poslovni pravili, ki se ju ne da razstavljati še naprej, brez da bi izgubili na svojem pomenu. Prvo pravilo je: če je bil podan zahtevek za

povračilo stroškov več kot eno leto po njihovem nastanku, potem se njihovo povračilo zavrne. Drugo pravilo: če je vlagatelj v preteklem letu preživel v tujini več kot 182 dni, potem se zahtevek za povračilo zavrne. Aktivnost analize pravil se lahko prične takoj, ko ima projektna ekipa na voljo opisana pravila, ki so odobrena in se uporabljajo v poslovnem sistemu;

3. Načrtovanje pravil (angl. Rule Design) – V sklopu te aktivnosti se moramo odločiti, na kakšen način bomo implementirali poslovna pravila v sistem. Najbolj primerna je implementacija s procesnimi stroji (angl. Rule Engine), ki nam zagotavlja največjo agilnost, ki jo predstavlja pristop s poslovnimi pravili;
4. Avtorstvo (angl. Rule Authoring) – Je zelo pomembna aktivnost v sklopu razvoja poslovnih pravil. Predstavlja implementacijo pravil v poslovni sistem v izbrani tehnologiji. Najprej je potrebno vzpostaviti okolje za implementacijo, nato pa implementirati prej pridobljena pravila v formalno, izvajalno obliko;
5. Potrjevanje pravil (angl. Rule Validation) – Aktivnost, ki zajema testiranje obnašanja pravil neodvisno od aplikacijske kode in zagotavljanje njihovega pravilnega delovanja;
6. Vpeljava pravil (angl. Rule Deployment) – V sklopu te aktivnosti vpeljemo potrjena pravila v delujoč informacijski sistem. Pri tem moramo izbrati poslovna pravila, jih združiti v obvladljive enote – množice in jih naložiti na strežnik za izvajanje. Natančnejše naloge, ki jih moramo izvesti v sklopu aktivnosti vpeljave pravil, so:
  - 6.1. Izbrati obseg pravil, katera želimo dati v izvajanje;
  - 6.2. Strniti pravila v množice, ki jih lahko vpeljemo v izvajalno okolje;
  - 6.3. Vpeljati množice v okolje, kjer so pripravljene za izvajanje;



Slika 4.3: Aktivnost vpeljave pravil v informacijski sistem [14].

- 6.4. Obvestiti procesni stroj za izvajanje, da obstajajo nova poslovna pravila;
- 6.5. Naložiti pravila znotraj izvajalnega okolja;
- 6.6. Sprožiti prevajanje pravil;
- 6.7. Izstaviti zahtevek za proženje pravil.

### 4.3.2 RAZVOJNI CIKLI POSLOVNIH PRAVIL

Izvajanje aktivnosti v kratkih ciklih nam zagotavlja, da bodo izhodi iteracij izpolnjevali prvotne zahteve in bodo relevantni [18].

1. cikel – Seznanitev (angl. Harvesting) - Zajema aktivnosti odkrivanja in analize pravil. V prvi fazi projektna ekipa s pomočjo modeliranja poslov-

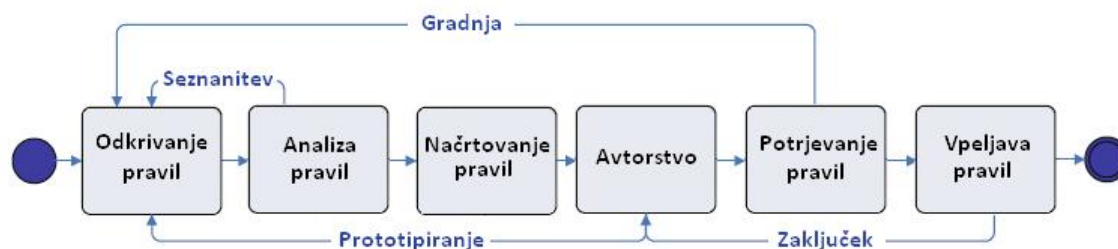
nih aktivnosti dobi vpogled nad odločitvami v procesu, njegova cilja pa sta seznanitev s poslovnimi pravili in politiko združbe, ter osnovno razumevanje poslovnih entitet in dokumentov. Ekipa najprej pridobi zelene podatke, nato pa rezultate skupaj analizira in dokumentira. Eden od rezultatov prvega cikla je dokument, ki vsebuje točke v procesu, kjer se vršijo odločitve.

2. cikel – Prototipiranje (angl. Prototyping) – Zajema aktivnosti odkrivanja, analize, avtorstva in načrtovanja pravil, njegov namen pa je storiti prvi korak k implementaciji z iterativno in inkrementalno izdelavo izvajalnega prototipa uporabe poslovnih pravil. Cikel se začne s pripravo in odločitvijo o izvajalnem okolju, ugotavljanjem organizacije poslovnih pravil ter nadaljuje z grajenjem testnih scenarijev in prototipiranjem pravil. Ko je dosežen ustrezen nivo, lahko ekipa začne z naslednjo aktivnostjo – avtorstvom pravil. Zgodnje začenjanje z implementacijo ima pozitivne vplive na potek projekta, saj lahko zgodaj odkrije morebitne pomanjkljivosti pri analizi in načrtovanju, ki se najbolj izrazijo pri implementaciji. Tako lahko ekipa popravi situacijo in izgradi poslovno relevantne množice pravil.

3. cikel – Gradnja (angl. Building) – V glavnem jo predstavljata aktivnosti avtorstva in potrjevanja pravil, cikel pa lahko v primeru potrebe po mnenju eksperta ali zavoljo izboljšav zajame tudi aktivnosti odkrivanja in analize pravil. Namen tega cikla je postopna implementacija posameznih primerov pravil z realnimi podatki in njihovo testiranje v okviru izvajalnega konteksta. Ta cikel je zelo pomemben, saj prikaže, kako projekt napreduje in kako je razvojna ekipa sprejela nove tehnologije.

4. cikel – Sestavljanje (angl. Integrating) – Namen je sklapljanje posameznih delov poslovnih pravil in njihovo skladiščenje na strežniku za namen celotnega (angl. End-To-End) testiranja. Ta pravila bodo v primerih zahtevkov aplikacij posredovana v izvajanje.

5. cikel – Zaključek (angl. Enhancing) – Zadnji cikel zajema aktivnosti avtorstva, preverjanja in vpeljave in je namenjen predvsem dokončnemu preverjanju kvalitete in izvajanju dodatnih testov s strani bolj poslovno orientiranih akterjev.



Slika 4.4: Slika prikazuje izvajanje aktivnosti v ciklih [14].

### 4.3.3 ŽIVLJENJSKI CIKEL POSLOVNIH PRAVIL

V hitre in pogoste spremembe, na katere se mora poslovni svet v najkrajšem možnem času odzvati, so vključene tudi spremembe poslovnih pravil. Tako prihaja do pojavitve novih poslovnih pravil, do njihovega preoblikovanja ali pa do prekinitve veljavnosti sedanjih pravil. Lahko torej rečemo, da imajo poslovna pravila svojo življenjsko dobo, v kateri gredo skozi različne faze. Vse se začne s prvo fazo - z rojstvom, ko se novo pravilo pojavi. Nato se pravilo razvija in bolj natančno definira. Sledi obdobje potrjevanja, ko mora pravilo prestati testiranja in potrditi, da počne tisto, kar se od njega zahteva. Tako je sposobno vstopa v naslednjo obdobje, v delovni svet, ko ga vključimo med ostala pravila v sistemu. Poslovno pravilo tako izvaja nalogo, ki se jo od njega zahteva, dokler se ne postara in izgubi na veljavi. Tedaj konča z delovnim obdobjem in ga je potrebno odstraniti iz informacijskega sistema ter nadomestiti z novim. Stanja, v katerih se lahko nahaja poslovno pravilo, so [19]:

- Pravilo je novo - Pomeni da je ustvarjeno in pripravljeno na programerjeve modifikacije;
- Pravilo je nared za pregled – Potem ko je programer končal z začetnim delom, ga mora pregledovalec pravil pregledati in preveriti njegovo pravilnost v smislu poslovnega obnašanja; Če pravilo ni ustrezno, pregle-

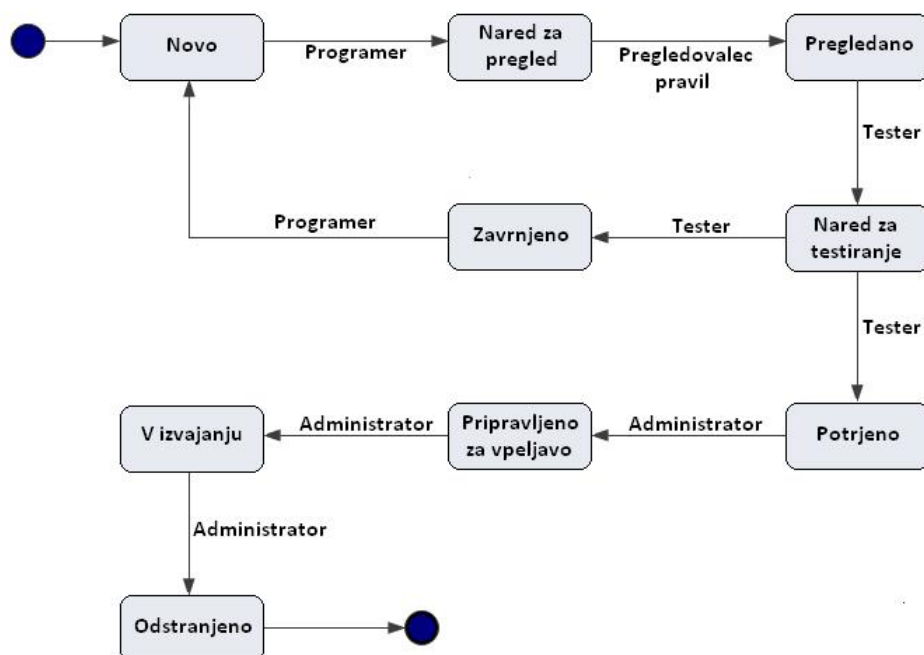
dovalec sodeluje s programerjem, dokler ni zagotovljena njegova ustreznost;

- Pravilo je pregledano – Pregledovalec pravila ga označil za ustreznega;
- Pravilo je pripravljeno na testiranje – Pravilo bo ostalo v tem stanju med obdobjem njegovega testiranja. To pomeni do takrat, ko ga bo tester sprejel ali zavrnil;
- Pravilo je zavrnjeno – Tester je zavrnil veljavnost pravila;
- Pravilo je potrjeno – Tester je potrdil veljavnost pravila;
- Pravilo je pripravljeno na vpeljavo – Zadnje stanje, preden gre pravilo v produkcijo;
- Pravilo je v izvajanju – Administrator postavi pravilo, skupaj z ostalimi, v produkcijsko okolje;
- Pravilo je odstranjeno – Administrator zaradi nerelevantnosti odstrani pravilo iz izvajalne faze.

Življenjski cikel poslovnih pravil se lahko med posameznimi organizacijami razlikuje. Pomembno je, da ga, glede na vrsto pravil, število testnih okolij, izkušnost razvojne ekipa in ostalih dejavnikov prilagodijo tako, da kar najboljše ustreza njihovim potrebam.

#### **4.3.4 NAJBOLJŠE PRAKSE**

Da bi imela uporaba poslovnih pravil čimboljši učinek na poslovni proces, moramo pred začetkom njihovega razvoja določiti standarde, katerih se bomo držali med samim razvojnim procesom. Mednje spada tudi upoštevanje najboljših praks, ki se nanašajo na pisanje in načrtovanje poslovnih pravil. Te so [1]:



Slika 4.5: Stanja v življenjskem ciklu poslovnih pravil [19].

- Upoštevanje natančnosti – Pri definiranju poslovnih pravil ne sme prihajati do dvoumnih razlag. Poslovno pravilo mora biti razumljeno le na en način, v nasprotnem primeru ga je potrebno dodelati;
- Upoštevanje konsistentnosti – Eno poslovno pravilo ne sme vplivati na ostala;
- Neodvečnost – Poslovna pravila se ne smejo prekrivati, če je pravilo vsebovano v eni področni množici, potem ne sme biti še v drugi;
- Poslovna orientiranost – Poslovna pravila morajo biti izražena z uporabo poslovnega besedišča, zato da jih lahko razume široka množica oseb. Treba se je izogibati uporabi tehničnega žargona;
- Deklarativnost – Pojem, ki pojasnjuje, da morajo zaradi razumljivosti definicije poslovnih pravil opisovati njihovo namero, ne pa načina uveljavitve.



## Poglavje 5

# AVTOMATIZACIJA POSLOVNIH PRAVIL

Kot smo že dejali, je v poslovnem svetu zelo pomembna agilnost, ki se nanaša na poslovne spremembe. Želja odgovornih je pohitrili postopek izdelave izdelkov ter zmanjšati ceno razvoja in njihovega vzdrževanja. Da bi uresničili omenjeni želji, moramo neprestano spreminjati svoj poslovni proces, ga izpopolnjevati in prilagajati razmeram na trgu. To pa ni enostavno. Raziskava [21] v letu 2004 je razkrila, da naj bi kar polovica razvojnih IT projektov v Združenih državah Amerike presegala zastavljene časovne in denarne okvirje, ali pa naj sistemi ne bi vsebovali dogovorjenih kritičnih funkcij. Eden od najpogostejših razlogov za omenjene neuspehe naj bi se skrival v nejasnosti vizije in nerazločnosti doseganja zastavljenih ciljev organizacij. Da bi to dosegli, morajo biti postopki, po katerih smo prišli do odločitev, pojasnjeni in konsistentni, zaradi hitrih sprememb v poslovnem svetu pa tudi enostavno spremljivi. Pravi problem se torej skriva v vprašanju, kako učinkovito preslikati poslovne odločitve organizacij v IT sisteme in s tem pridobiti preglednost nad odločitvami, ter možnost njihovega spreminjanja?

Obstaja več načinov, ki nam omogočajo implementacijo poslovnih pravil v aplikacijskem sistemu in njihovo kasnejše izvajanje. Najpogostejši so [14]:

- Implementacija poslovnih pravil znotraj aplikacijske kode – Komple-

ksno odločitveno logiko lahko vrinemo znotraj funkcij, podprogramov, procedur... Takšna implementacija ima, čeprav je lahko za majhno množico pravil, ob dobrem programiranju ustrezna, tudi svoje pomanjkljivosti. Težave nastanejo pri sledljivosti poslovnih pravil, ki so lahko razbita na različne dele aplikacijske kode. Tako je težko ohraniti nadzor nad spremembami in verzijami pravil. Če omenjenemu razlogu dodamo še dejstvo, da so zapisana v jeziku, ki poslovnim uporabnikom ni domač, vidimo, da takšen način upravljanja s poslovnimi pravili ni zaželen;

- Implementacija na nivoju podatkovnega modela – Mnogo poslovnih pravila se navezuje na strukturo podatkov. Kot primer navedimo izjavo: »Aplikacija za vodenje lastništev avtomobilov, mora vsebovati osebo, ki je bila prvi lastnik avtomobila, poleg prvega lastnika pa lahko vsebuje še vse morebitne nadaljnje lastnike.« Takšno pravilo bo razdeljeno na dva dela (razreda), razred »Lastnik« in »Aplikacija« in na dve povezovalni asociaciji. Poleg izraženih strukturnih pravil, pa lahko na njih uveljavljamo tudi pravila, ki zadevajo njihovo obnašanje v času nastanka ali trajanja pravila. To lahko dosežemo npr. s prožilci v podatkovnih bazah. Največji problem te vrste implementacije je njegova statičnost in nesposobnost enostavnega spreminjanja poslovne logike, saj zahteva spremembe v vseh ciklih razvoja programske opreme. Tega si v današnjem času, ko je zahteva po poslovni agilnosti na najvišji ravni, ne moremo privoščiti;
- Znotraj sistema za upravljanje poslovnih procesov – BPM orodja se pri modeliranju osredotočajo na glavne značilnosti procesa, kot so kdo in kdaj je udeležen v proces, kaj morajo posamezniki storiti itd. Takšna orodja podpirajo tako človeška opravila, kakor tudi avtomatizirana. Tem opraviom moramo na teh mestih na deklarativen način določiti pravila, na katera se med izvajanjem opirajo, ter nato zasnovati procesni tok glede na odločitve, ki se vršijo v opravilih. Problem takšne

implementacije je v spreminjanju izvajajočega procesnega toka aplikacij. Vsaka sprememba namreč zahteva ponovno definiranje procesa, obenem pa postaja z vsakim dodanim pravilom procesni tok bolj nepregleden. Takšne spremembe so lahko na dolgi rok problematične, saj ne želimo neprestano spreminjati trenutno izvajajočega procesa. Nove vrste poslovnih pravil ne smejo vplivati na samo izvajanje osnovnega poslovnega procesa;

- S pomočjo grafičnega vmesnika – Predstavlja enega od načinov implementacije poslovnih pravil znotraj aplikacijske kode. Veliko spletnih aplikacij uporabljata ta način predstavitve poslovnih pravil. Ta se predvsem nanaša na validacijo pravilnosti vnosa podatkov in na nadaljnje usmerjanje glede na doslej vnesene podatke. Ponavadi so takšna poslovna pravila implementirana znotraj strežnikovih kontrolnih razredov ali pa preko različnih skriptnih jezikov na odjemalčevi strani, ki lahko direktno (brez prevajanja) izvajajo ukaze. Ta način ima prednost v hitrosti, saj se izogne nepotrebному prometu po omrežju zaradi dodatnih prevajanj kode, pomanjkljivosti pa so podobne kot pri prvo omenjenemu načinu implementacije (prilagodljivost, sledljivost), poleg tega pa skriptni jeziki niso dovolj industrijsko močni, da bi lahko hkrati podpirali veliko število transakcij [22];
- Implementacija znotraj sistemov za upravljanje s poslovnimi pravili BRMS – V tem primeru za določanje poslovnih pravil skrbijo poslovni uporabniki. Pišejo jih v njim razumljivem jeziku, ta pravila pa so shranjena ločeno od jedra aplikacijske kode. Ta način si bomo ogledali v nadaljevanju.

## 5.1 SISTEM ZA UPRAVLJANJE S POSLOVNIMI PRAVILI

Da bi dosegali uspehe v poslovnem svetu, mora biti izvajanje poslovnih pravil in njihovo upravljanje čimbolj pregledno in enostavno. Najboljši način za doseganje teh želja je uporaba sistema za upravljanje s poslovnimi pravili - BRMS. BRMS je programska aplikacija, ki vsem udeleženiim v procesu, tako poslovnim uporabnikom kot tudi IT strokovnjakom, omogoča definiranje, izvajanje, spremljanje in posodabljanje poslovne logike znotraj posamezne organizacije [23]. V ta namen mora tipičen BRMS vsebovati vsaj naslednje komponente:

1. Repozitorij poslovnih pravil – To je nekakšno centralizirano skladišče, ki predstavlja poslovno politiko organizacije in je najvažnejša ideja BRMS [16, 21]. Vanj se shranjujejo vse informacije v povezavi s poslovnimi pravili - to so poslovna pravila sama, posamezne množice poslovnih pravil, ki se nanašajo na določeno področje uporabe ali različne verzije teh množic pravil, ki nam dajejo vpogled nad že uporabljanimi pravili v preteklosti. Pomembna funkcionalnost repozitorija je, da omogoča hranjenje poslovnih pravil ločeno od aplikacijske kode preostalega informacijskega sistema. Ker so zbrana na enem mestu in veljajo na nivoju celotne organizacije, torej niso v lasti zgolj enega procesa, omogočajo povečan nadzor in lažje vzdrževanje uporabljane poslovne logike, saj je ob spreminjanju njihovega pomena to mogoče storiti na enem mestu. Poleg tega pa zaradi spreminjanja in ponovnega prevajanja aplikacije, le-te ni potrebno ustaviti;
2. Orodja, ki omogočajo nadzor nad izvajanjem in upravljanjem poslovne logike same – V nabor teh orodij spadajo orodja, ki poslovnim uporabnikom (pa tudi IT strokovnjakom) omogočajo enostaven dostop do nabora poslovnih pravil. Njihovo upravljanje je ponavadi omogočeno preko uporabniških vmesnikov, ki so intuitivni za uporabnike in pri-



Slika 5.1: Komponente tipičnih BRMS [1].

jazni za uporabo. Tako je zagotovljena čim večja enostavnost pisanja in spreminjanja poslovnih pravil v oblikah, ki so blizu poslovnim uporabnikom. S tem se zmanjša odvisnost poslovnih uporabnikov od IT strokovnjakov pri implementaciji sprememb in omogoči poslovnim uporabnikom, da se bolj osredotočijo na svojo primarno nalogo – posel in ne posvečajo obsežnemu tehničnemu izobraževanju. Med komponente, ki sestavljajo BRMS, pa spadajo tudi orodja, ki so namenjena IT strokovnjakom in predpisujejo podrobnosti o avtomatizaciji izvajanja poslovnih pravil samih in njihovi integraciji z ostalimi deli aplikacijske arhitekture. Med drugim določajo tudi, kako procesni stroji dostopajo do repozitorija z množicami pravil, ki so pripravljene na izvajanje;

3. Procesni stroj – Predstavlja osrednji del BRMS, izvajalno okolje, ki je odgovoren za izvrševanje poslovnih pravil. Omenjena komponenta prejme zahtevo za izvršitev storitve, ki se nanaša na poslovna pravila. Na podlagi zahteve poišče in izvrši poslovna pravila, obenem pa nam

omogoči sledljivost, kako je do končnega sklepa prišlo. Rezultat nato poda nazaj aplikaciji.

Izvajanje poslovnih pravil se v večini sistemov, zaradi želje po boljšem delovanju sistema v primeru velikih množic pravil, ne izvaja zaporedno, ampak se izvede na podlagi mrežnega (lat. Rete) algoritma [1, 33]. To je učinkovit postopek za primerjavo vzorcev (dejstev in pravil), ki shrani delno ujemaajoče rezultate delovne množice v omrežje vozlišč. Zasnoval ga je Charles L. Forgy in ga leta 1979 predstavil v svoji doktorski dizertaciji [33]. Zaradi narave algoritma je število primerjav med podatki in pravili manjše. Tako ob spreminjanju, brisanju ali dodajanju novih dejstev ne prihaja do njihovega večkratnega, nepotrebnega (zamudnega) preverjanja. Rete algoritem je torej podatkovno usmerjen (angl. Data-Oriented) in deluje po principu veriženja naprej (angl. Forward-Chaining) [1, 32] - izvajanje poslovnih pravil vodijo že vsebovana dejstva. Ko se določeno pravilo sproži, lahko to privede do novega spoznanja in s tem povečanja omrežja vsebovanih dejstev, na katerih se vrši izvajanje. Ta proces je iterativen in se nadaljuje, dokler ni doseženo končno stanje in s tem izpolnjen cilj.

Rete algoritem pa ni edini (učinkoviti) postopek, preko katerega je mogoče izvajanje poslovnih pravil. Najnovejši algoritem za ta namen, uporabljen je tudi v sistemu JBoss Drools 6, se imenuje PHREAK, skupaj z BRMS pa ga je izdalo podjetje RedHat (oddelek JBoss) novembra 2013. Za razliko od rete algoritma, je PHREAK ciljno orientiran (angl. Goal-Oriented) - sklepanje se začne končnim ciljem, za katerega procesni stroj poskuša najti dejstva, ki bi tak cilj omogočala [35]. Pri tem je upoštevana ena od značilnosti PHREAK-a, in sicer odlašanje z evalvacijo delno ujemaajočih se primerjav med podatki, kateri nimajo neposrednega vpliva na izvajanje poslovnih pravil [34]. Posledica tega je večja učinkovitost izvajanja poslovnih pravil v sistemih z velikimi količinami podatkov.

## 5.2 TEHNIKE ZA PREDSTAVITEV POSLOVNIH PRAVIL V BRMS

Poslovna pravila se lahko izražajo s pomočjo različnih tehnik. Naša želja je, da bi lahko predstavili logiko poslovnih pravil na razumljiv način, ki bi ustrezal potrebam organizacije in, ki bi njeno upravljanje primarno omogočal poslovnih uporabnikom. BRMS produkti omogočajo podporo tekstovnim predstavitvam pravil, dodatno pa lahko podpirajo tudi grafične predstavitve v obliki odločitvenih dreves ali tabel poslovnih pravil [24].

### 5.2.1 DOMENSKO SPECIFIČNI JEZIKI

Domensko specifični programski jeziki (angl. Domain Specific Programming Language – DSL) so namenski jeziki, ki so zasnovani za specifično področje uporabe. V našem primeru se to področje nanaša na razvoj poslovnih pravil. Zaradi svojega ožjega domenskega področja, se razlikujejo od jezikov za splošno uporabo (angl. General-Purpose Programming Language - GPL), kot je denimo Java. Tako lahko bolj podrobno zajamejo specifično problem-sko področje. DSL jeziki se zaradi svoje opisne narave (opisujejo, kaj se bo zgodilo in ne kako) uvrščajo med izvajalne jezike. Zaradi višjega nivoja abstrakcije so bolj natančni in lažji za implementacijo. Zaradi tega, ker omogočajo zapis poslovnih pravil na način, ki je zelo podoben naravnemu jeziku, niso razumljivi zgolj programerjem, pač pa tudi specialistom na posameznih področjih. To se odraža v krajšem razvojnem času in cenejšem vzdrževanju, zmanjša pa se tudi odvisnosti od IT oddelka. Kot primer navedimo poslovno pravilo, ki se nanaša na odobritev nakupa avtomobila. Pravilo bi v programskem jeziku Java izgledalo takole:

```
public boolean checkDriverAbility (Customer customer){  
  
    boolean declineSelling = false;  
    int income = customer.getIncome();
```

```
int dangerRating = customer.getDangerRating();
if (income < 10000 || dangerRating > 5){
    declineSelling = true;
}
return declineSelling;
}
```

Dodajmo pa še njegovo alternativno obliko v DSL:

```
if the customer yearly income is less than 10000
    and the customer danger rating is higher than 5
then decline the selling;
```

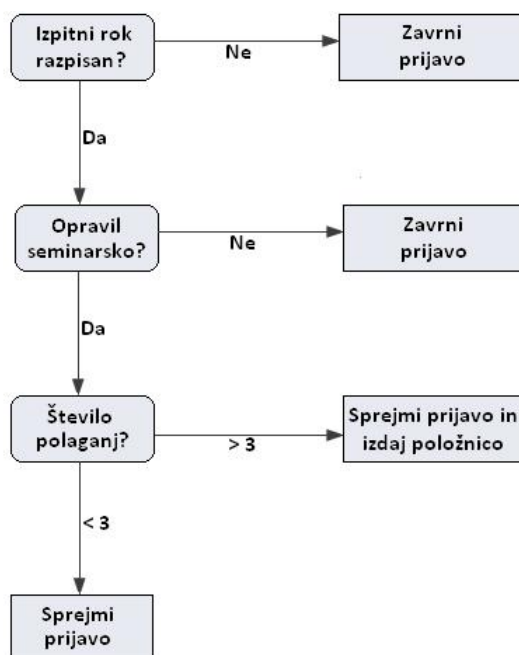
### 5.2.2 ODLOČITVENA DREVESA

Odločitvena drevesa (angl. Decision Trees) prikazujejo hierarhično odvisnost med odločitvami. Opisujejo zaporedje korakov, ki se zgodijo pri izvajanju poslovnih pravil. Procesni stroj zna prebrati poslovno pravilo iz odločitvenega drevesa in ukrepati glede na situacijo. Drevesa so sestavljena iz vozlišč in povezav med njimi. Vozlišča predstavljajo pogoje, povezave med njimi pa možne vrednosti posameznih pogojev. Posamezna pot v drevesu poslovnih pravil, od njegovega korena (začetnega) do predzadnjega vozlišča, predstavlja kombinacijo pravil. Zadnje vozlišče, imenovano tudi list drevesa, prikazuje seznam akcij, ki se prožijo s kombinacijo izvajanih pravil na poti. Primer odločitvenega drevesa, ki predstavlja poslovna pravila za (poenostavljeno) prijavo študenta na izpit, je prikazan na sliki 5.2:

### 5.2.3 ODLOČITVENE TABELE

Odločitvene tabele (angl. Decision Tables) predstavljajo tehniko za predstavitev poslovnih pravil v tabelarnem formatu. So kombinacija pogojev, domene vrednosti in akcij, ki se izvršijo glede na situacijo. Težave, ki se

## 5.2. TEHNIKE ZA PREDSTAVITEV POSLOVNIH PRAVIL V BRMS 47



Slika 5.2: Primer odločitvenega drevesa.

lahko pojavijo pri uporabi odločitvenih tabel, so vezane na njihovo velikost. V primeru velikega števila različnih variant pravil jih je težko obvladovati. Primer prikazuje poslovno pravilo, ki določa različne načine naslavljanja oseb z uporabo odločitvene tabele.

Spol	Status	Način naslavljanja
Moški		Mr.
Ženski	Poročena	Mrs.
Ženski	Samska	Miss
Ženski	Neznan	Ms.

Tabela 5.1: Primer odločitvene tabele [29].

### 5.3 PREDNOSTI IN SLABOSTI UPORABE BRMS

Kot smo videli, lahko v želji po ločevanju poslovnih pravil od ostalih delov aplikacijske kode, uporabimo BRMS. Njegova uporaba je za upoštevanje pristopa s poslovnimi pravili zelo priporočljiva, če pa za ključen vidik poslovanja smatramo njegovo agilnost, pa je uporaba BRMS sistemov obvezna [14]. Sistemi za upravljanje poslovnih pravil upoštevajo manifest poslovnih pravil iz leta 2002, [25] v katerem je določeno, da:

1. Posvečanje poslovnim pravilom je primarna in ne sekundarna naloga;
2. Poslovna pravila morajo biti ločena od procesov in ne vsebovana v njih samih;
3. Morajo biti poslovna pravila za poslovne uporabnike izražena v opisnem in ne proceduralnem jeziku;
4. Morajo biti poslovna pravila dobro formirana, da lahko poslovni uporabniki validirajo njihovo pravilnost;
5. So BRMS sistemi primarno namenjeni poslovnim uporabnikom;
6. Morajo biti BRMS sistemi arhitekturno zasnovani tako, da omogočajo konstantne spremembe pravil in ne programskih ali strojnih platform. Prav tako morajo omogočati razlago, zakaj je do določenih akcij prišlo;
7. BRMS sistemi morajo biti podrejeni poslovanju in ne tehnologiji.

Uporaba BRMS sistemov tako organizacijam omogoča precej prednosti [21, 22, 23]:

- Zmanjšana odvisnost od IT oddelka;
- Poslovni uporabniki imajo večji nadzor nad razvojem upravljanja poslovnih pravil;

- Možnost izražanja poslovne logike z večjo natančnostjo in možnost hitrejšega razvoja;
- Večja učinkovitost poslovnega procesa zaradi avtomatizacije;
- Lažje vzdrževanje in večja možnost ponovne uporabe;
- Zmanjšan semantični prepad med poslom in IT;
- Večja konsistenca pri odločitvah znotraj organizacije;
- Jasnejša revizija.

Kljub vsem prednostim, ki jih s seboj prinašajo sistemi za upravljanje poslovnih pravil, pa sami ne zagotavljajo uspešnosti. Vedeti moramo, da je to samo en del kompleksne aplikacije oz. rešitve, ki pomaga pri učinkovitosti izvajanja poslovnih procesov. Uporaba BRMS sistemov pa ima tudi nekatere pomanjkljivosti [21, 22]:

- Velika navezanost na določen proizvod BRMS sistema. V primeru zamenjave sistema lahko nastanejo stroški zaradi prevelike navezanosti na implementacijo prejšnjega;
- Slabo razviti vmesniki, ki ne omogočajo popolnoma neomejenega pristopa k reševanju problemov;
- Prihaja lahko do tehničnih težav pri prevajanju sistema z ogromno in slabo urejenimi množicami pravil.



## Poglavje 6

# KRATEK PREGLED BRMS TEHNOLOGIJ

BRMS sistemi omogočajo poslovnim in tehničnim uporabnikom upravljanje s poslovnimi pravili, katera kot že vemo, vodijo posel. Pri upoštevanju SOA načina za načrtovanje arhitekture, ki omogoča neodvisnost med aplikacijami ali procesi, je poslovna logika zajeta v obliki storitev znotraj BRMS sistemov. Različne aplikacije oz. procesi lahko po potrebi kličejo te storitve, preko katerih se izvršijo zahtevani izračuni in posredujejo končne odločitve nazaj aplikacijam. Na tak način je mogoče tudi sodelovanje med BPM in BRMS sistemi. Na trgu obstaja veliko ponudnikov BRMS sistemov, zato je konkurenca precej zaostrena. Na to vpliva kar nekaj dejavnikov [14], eden od teh je uspešen preboj BRMS sistemov na tržišče in zavedanje strank o njihovem pomenu za doseganje zelenih poslovnih rezultatov. Drugi razlog je, da so na tržišču že druge ali tretje generacije začetnih BRMS produktov, ki vsebujejo nove izboljšave, izboljšano pa je tudi delovanje omenjenih sistemov. Tretji razlog pa se skriva v svetovnem trendu zategovanja pasu, ki ga je s seboj prinesla gospodarska kriza - tako vsaka združba dvakrat premisli, kam bo investirala svoj denar in obenem zahteva več za nižjo ceno. Nekaj ponudnikov BRMS sistemov na trgu [36]: FICO, IBM, InRule Technology, JBoss, OpenRules, Oracle, SAP, Versata.

## 6.1 IBM WODM

IBM Websphere Operational Decision Manager je lastniški BRMS sistem, ki je nasledil ILOG JRules. Prva verzija (v7.5) je izšla leta 2011, trenutna različica sistema pa je v8.5, ki je luč sveta ugledala junija 2013 [37]. Sestavljajo ga naslednji produkti, s katerimi predstavlja platformo za upravljanje in izvrševanje poslovnih pravil [39]:

- IBM Decision Center V8.5 - Odločitveni center nudi integriran repozitorij in upravljalne komponente, ki vsem namenskim ekspertom omogočajo sodelovanje pri definiranju in upravljanju odločitvene logike poslovnih pravil;
- IBM Decision Server V8.5 - Odločitveni strežnik priskrbi izvajalne in razvojne komponente za avtomatizacijo poslovne logike. Omogoča zaznavanje poslovnih situacij, ki zahtevajo izvrševanje poslovnih pravil, izvrševanje poslovnih pravil in odgovarjanje na zahteve;
- Decision Server Rules Edition for Integration Bus V8.5 - Omogoča enostavnejšo integracijo med WODM in vodilom, ki je potrebna za povezavo med izvajajočo aplikacijo in odločitvenimi storitvami;
- IBM ODM Express V8.5 - Predstavlja vstopno točko za pomoč pri avtomatizaciji in upravljanju poslovnih odločitev.

IBM WODM se ponavadi uporablja skupaj s preostalima IBM rešitvama za upravljanje poslovnih procesov [38]:

- IBM BlueWorks Live - Je orodje za modeliranje poslovnih procesov in skupnost za poslovne uporabnike. Temelji na storitvah v oblaku (angl. Cloud-Computing), do katerih se dostopa preko spleta;
- IBM Business Process Manager - IBM BPM je vodilna rešitev za upravljanje poslovnih procesov na svetu. Z njo lahko modeliramo poslovne procese, sprejemamo modele iz okolja BlueWorks Live, izvajamo

stroškovne simulacije, izvršujemo procese v produkciji, spremljamo realno-časovne KPI metrike, optimiziramo procese ipd.

## 6.2 ORACLE BR

Oracle Business Rules je produkt, ki v poslovne procese vnaša agilnost, poslovno kontrolo in transparentnost. Trenutna verzija sistema je 11g R1 11.1.1.7 in je izšla aprila 2013 [42]. Rešitev je bila predstavljena kot del Oracleovih infrastrukturnih produktov, imenovanih Oracle Fusion Middleware[40]. Ti so integrirani v aplikacijah in tehnologijah za pohitritev implementacij ter zmanjšanja stroškov upravljanja sprememb. Izdelek Business Rules sestavljajo naslednji izvajalni in načrtovalni elementi [41]:

- Decision Component in an SOA Composite Application - Odločitvena komponenta za SOA kompozitne aplikacije (te aplikacije direktno podpirajo poslovne procese v podjetjih) je mehanizem za objavljanje množice pravil v obliki ponovno uporabljivih storitev. Tako so pravila hkrati na voljo večim poslovnim procesom. V njej je vsebovan tudi stroj za izvajanje poslovnih pravil, imenovan Oracle Business Rules Engine, ki skrbi za učinkovito izvajanje poslovnih pravil;
- Rules Engine with Oracle Business Rules in a Java EE Application - Stroj za izvajanje poslovnih pravil, ki ni uporaben le za izvajanje v SOA kompozitnih aplikacijah, ampak je na voljo tudi kot knjižnica za Java Enterprise Edition aplikacije;
- Rules Designer - Dodatek k orodju Oracle JDeveloper, ki tako poslovnim analitikom kot tudi razvijalcem aplikacij na enostaven način omogoča kreiranje in urejanje poslovnih pravil ter njihovo shranjevanje v slovar poslovnih pravil;
- RL Language - Visokonivojski programski jezik, ki predstavlja alternativo Rules Designer-ju za definiranje in urejanje poslovnih pravil;

- SOA Composer Application - Omogoča upravljanje s slovarjem poslovnih pravil.

OBR je komponenta, ki je vsebovana in se uporablja v naslednjih izdelkih [42]:

- Oracle SOA Suite - Predstavlja programsko opremo za grajenje, definiranje in upravljanje storitveno usmerjene arhitekture. Sistemskim razvijalcem tako omogoča rokovanje s storitvami in njihovo povezovanje s poslovnimi oz. kompozitnimi aplikacijami;
- Oracle BPM Suite - Vanj so vključena orodja za modeliranje, avtomatizacijo, spremljanje, optimizacijo in izvajanje poslovnih procesov ter njihovo sistemsko integracijo.

### 6.3 OPEN RULES

OpenRules je korporacija, ki jo je februarja leta 2003 ustanovil Jacob Feldman [44]. Njeno prvotno ime je bilo Intelligent ChoicePoint, decembra istega leta pa se je ob izdaji prvega produkta preimenovala v ime, pod katerim jo poznamo danes. To ime nosijo tudi njihovi BRMS sistemi, trenutna verzija pa je 6.2.6. in je izšla decembra 2013. Odprtokodni Open Rules sistem je sestavljen iz naslednjih večjih komponent [43]:

- Rule Repository - Služi organizaciji in vzdrževanju repozitorija poslovnih pravil. Poslovna pravila so v njem organizirana po principu preglednic, shranjena pa v xls ali xml formatu;
- OpenRuleEngine - Komponenta, ki omogoča izvajanje množic poslovnih pravil in metod;
- Rule Learner - S pomočjo algoritmov strojnega učenja omogoča analizo preteklih podatkov, na podlagi katerih lahko generira nova odločitvena pravila. Ta se lahko dodajo že obstoječim, ki so v izvajanju;

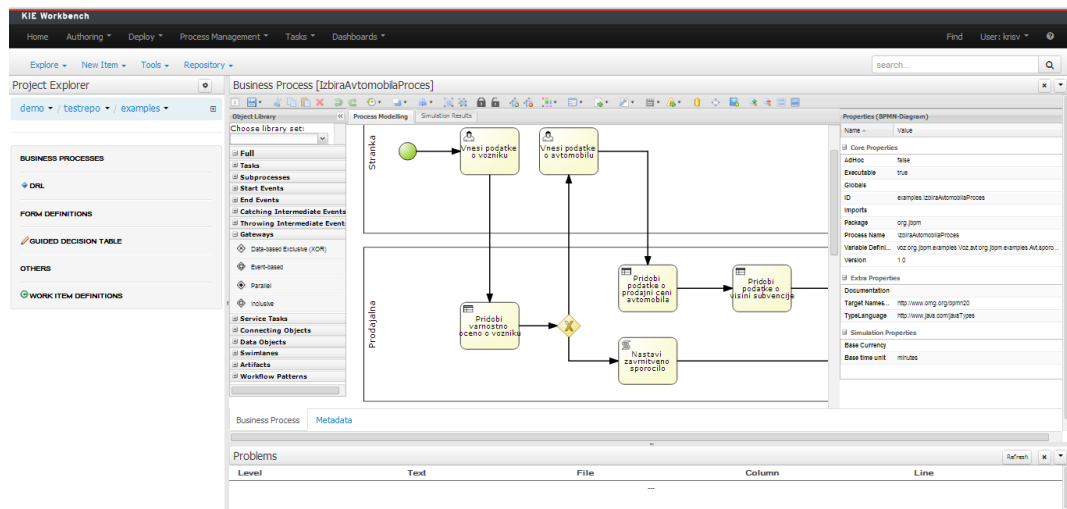
- Rule Solver - Njena naloga je posredovanje v primerih, kjer je veliko različnih možnosti izvajanja. Tako je omogočeno optimalno delovanje sistema;
- Rule Dialog - Predstavlja programsko komponento, ki tudi netehničnemu osebjemu omogoča izdelavo spletnih anket za pridobivanje strukturiranih informacij brez potrebe po učenju kompleksnih tehnik spletnega programiranja.

Poslovna pravila lahko v povezavi z OpenRules definiramo in vzdržujemo v znanih orodjih kot so Microsoft Excel, OpenOffice ali Google Spreadsheets. Projekte lahko integriramo v obstoječo programsko infrastrukturo, ki temelji na Java ali .NET tehnologiji, uporablja pa se jih lahko tudi kot spletne storitve pri SOA načinu ali pa kot storitve v oblaku.

## 6.4 JBOSS DROOLS

Še eden od BRMS produktov na trgu je odprtokodni JBoss Drools. Izdala ga je organizacija JBoss, njen ustanovitelj je bil Marc Fleury 1999. leta [45], od leta 2006 pa je JBoss divizija znotraj ameriške multinacionalke Red Hat. Trenutna verzija sistema je Drools 6.0 in je izšla novembra 2013, sestavljena pa je iz naslednjih projektov [46]:

- Drools Guvnor - Predstavlja centraliziran repozitorij za shranjevanje različnih verzij in kategorij poslovnih pravil. Vanj so vključena tudi orodja, ki omogočajo grafično upravljanje s pravili;
- Drools Expert - Stroj za izvajanje poslovnih pravil Drools Expert je deklarativno kodirno okolje, katerega temelj je izvajanje poslovnih pravil. Integriramo ga lahko z Java okoljem znotraj aplikacije ali pa kot storitev;
- jBPM 6.0 - Je fleksibilno BPM okolje, ki premošča semantični prepad med poslovnimi analitiki in razvijalci procesov. Poslovnim uporabni-



Slika 6.1: Prikaz modelirnega in izvajalnega okolja KIE.

kom preko spletnih aplikacij omogoča modeliranje, izvajanje in spremljanje poslovnih procesov - podpira celoten življenjski cikel poslovnih procesov;

- Drools Fusion - Omogoča zaznavanje velikih količin dogodkov, relacije med njimi in proži akcije na podlagi predpisanih pravil. Dogodek je definiran kot vsaka sprememba stanja ali zaznava stanja, ki proži neke akcije [47]. Tako se poveča zavedanje o pomembnih poslovnih dogodkih in omogoči hitrejšo odzivanje na njih;
- OptaPlanner - Za optimiziranje uporabe poslovnih virov (zaposlencev, sredstev, časa in denarja).

Vsi omenjeni projekti so z verzijo Drools 6 združeni v delovnem okolju KIE-WB (angl. Knowledge Is Everything Workbench), kar je prikazano je na sliki 6.1. Okolje temelji na orodju Apache Maven in primarno omogoča grajenje javanskih projektov [48]. Maven naslavlja dva vidika grajenja programske opreme - opisuje zgradbo programske opreme in pa tudi odvisnosti med njenimi deli. Za prikaz praktičnega primera smo tudi sami uporabili to okolje.

## Poglavje 7

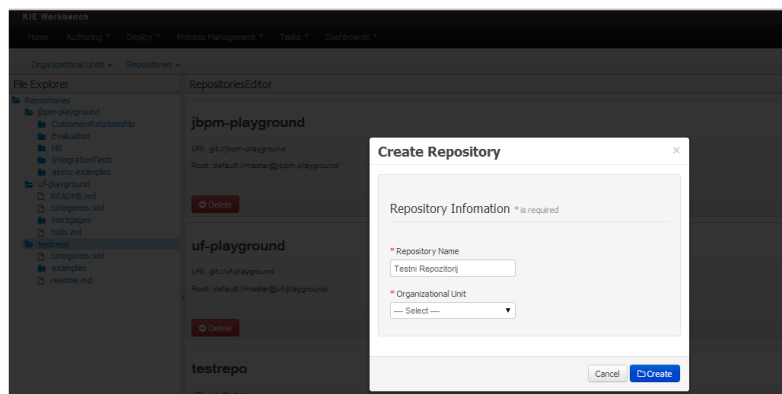
# IZDELAVA PRAKTIČNEGA PRIMERA

Poslovni proces, ki smo ga izdelali za prikaz funkcionalnosti orodja BRMS, je shematska informativna spletna konfiguracija avtomobila. Stranka v poslovnem procesu "IzdelavaAvtomobila" preko spletne aplikacije vnaša podatke, s katerimi generira avtomobil v nadaljevanju pa se opravi še informativni izračun o ceni sestavljenega avtomobila. Proces "IzdelavaAvtomobila" vsebuje aktivnost poslovno pravilo, skriptni opravili (angl. Script Task), človeška opravila (angl. User Task), dva elementa prehodov za vejitev procesnega toka in elementa za začetni in končni dogodek. Implementiran je z razvojnim okoljem KIE-WB, v jeziku BPMN 2.0, izvaja pa se na JBoss aplikacijskem strežniku verzije 7.1.1.

### 7.1 NAČRTOVANJE

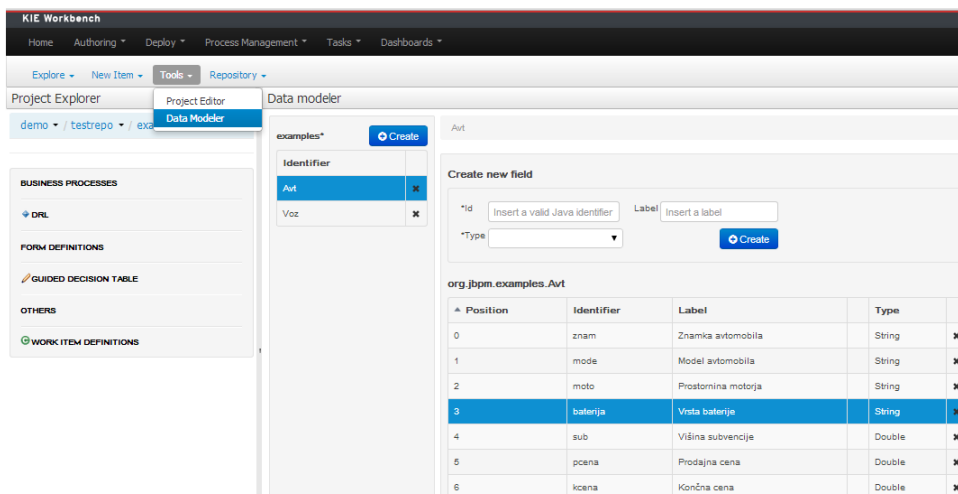
Pred začetkom načrtovanja procesnega toka v sistemu KIE-WB smo postorili dve stvari. Kot prvo smo pripravili projektni repozitorij za shranjevanje poslovnega procesa, pravil in podatkovnega modela.

Druga stvar pa je bila definicija podatkovnega modela z dvema entitetama in eno procesno spremenljivko, ki smo jih uporabili v našem primeru. Ti enti-



Slika 7.1: Izdelava novega projektnega repozitorija.

teti sta "Avt", ki predstavlja avtomobil, njeni atributi pa vsebujejo podatke o avtomobilu; ter "Voz", ki predstavlja voznika in njegove lastnosti. Lastnosti, ki opisujejo avtomobil so znamka in model avtomobila, prostornina motorja, vrsta baterije, višina državne subvencije ter prodajna in končna cena avtomobila. Še več podrobnosti o entiteti je razvidnih iz slike 7.2. Enako smo



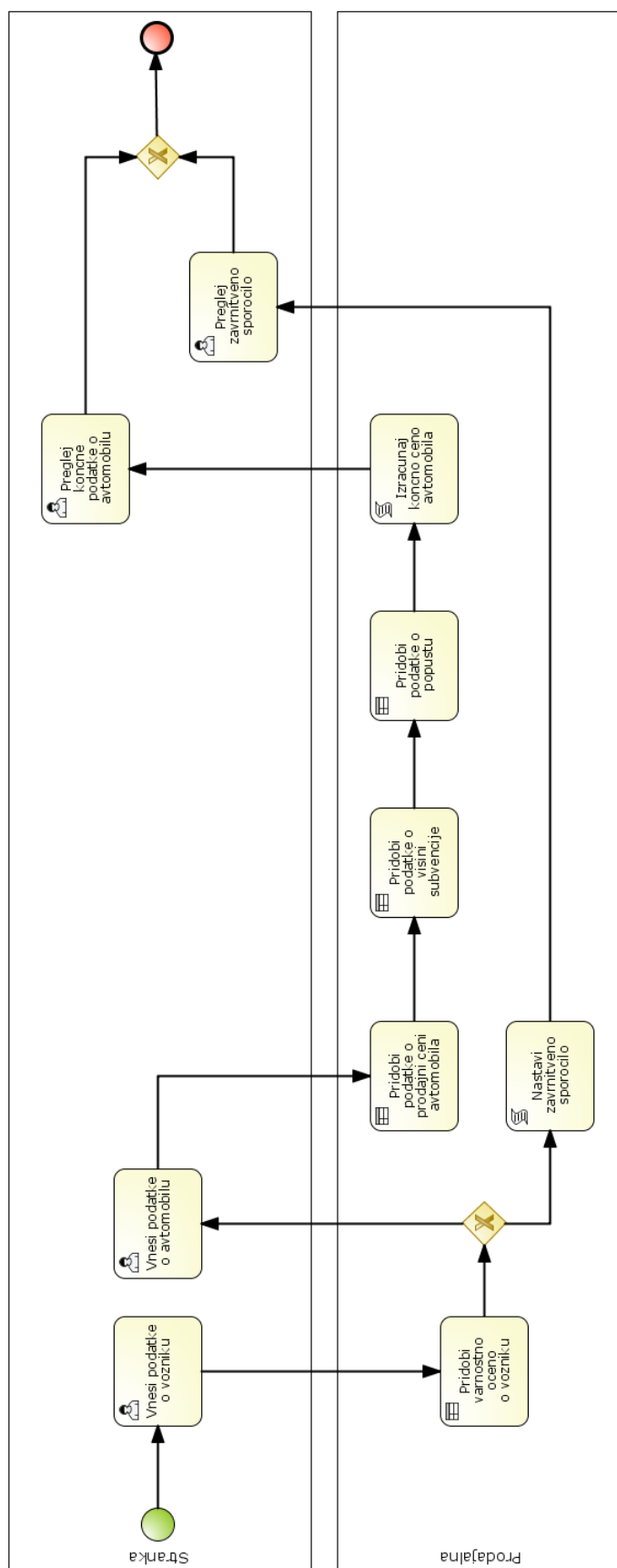
Slika 7.2: Predstavitev entitete Avt, ki predstavlja avtomobil.

storili tudi za entiteto, ki predstavlja voznika, njeni atributi pa so število kazenskih točk in let od pridobljenega vozniškega dovoljenja, število dosedanjih

prometnih nesreč, nakupovalni status, višina popusta in pa lastnost, ki hrani informacijo o upravičenosti nakupa. Poleg omenjenih entitet pa smo definirali še procesno spremenljivko tipa boolean in jo poimenovali "Sporočilo".

Po tem, ko smo definirali vse potrebno, smo začeli z modeliranjem procesnega toka. Kot prikazuje slika 7.3, se poslovni proces prične z uporabniškim opravilom "Vnesi podatke o vozniku", kjer stranka vnese zahtevane podatke o svojih dosedanjih voznških izkušnjah.

Ti podatki so število kazenskih točk, število let od pridobitve voznškega dovoljenja, število dosedanjih prometnih nesreč in nakupovalni status. Po zaključenem vnosu pri prvi aktivnosti sledi naslednje opravilo - "Pridobi varnostno oceno o vozniku", ki predstavlja izvrševanje vnaprej določenih poslovnih pravil organizacije. Poslovna pravila za omenjeno aktivnost so definirana v obliki odločitvene tabele, njihov izgled pa je predstavljen na sliki 7.4. Pravila se prožijo glede na vnešene podatke o voznških izkušnjah v prvi aktivnosti. Če voznikovo spričevalo ustreza pogojem, ki jih je za nakup postavila združba, se atribut, ki hrani voznikovo upravičenost do nakupa, postavi v stanje odobritve. Voznik je tako upravičen do nadaljevanja konfiguracijskega procesa, vrednost omenjenega atributa pa se preverja v naslednjem elementu - gradniku prehoda, v katerem se izbere ena od dveh možnih vej procesa. V omenjenem primeru se izbere zgornja. Proces se nadaljuje z novim uporabniškim opravilom - "Vnesi podatke o avtomobilu". Stranka izpolni obrazec s podatki, ki opisujejo avtomobil in po zaključenem vnosu so na vrsti tri opravila za izvajanje poslovnih pravil. Prvo se imenuje "IzbiraAvtomobila" in je tudi predstavljeno v obliki odločitvene tabele. Kot je prikazano na sliki 7.5, se na podlagi vnešenih parametrov o avtomobilu izračuna njegova prodajna cena. Pomembna stolpca v odločitveni tabeli sta poleg vseh atributov za opis variant, ki jih lahko izberemo za konfiguracijo avtomobila tudi stolpca z metapodatki "date-effective" in "date-expires". Preko njiju lahko določimo začetni in končni datum veljavnosti poslovnih pravil določene skupine poslovnih pravil. Prodajna cena avtomobila pa še ni enaka končni ceni. V naslednjih dveh aktivnostih, to sta "Pridobi podatke



Slika 7.3: Procesni diagram IzdelavaAvtomobila.

The screenshot shows the KIE Workbench interface with a 'Guided Decision Table [VarnostnaOcenaVoznika]'. The table has 16 rows and 9 columns. The columns are: #, Description, Kazenske tobe >=, Kazenske tobe <, Vejavnosti spita >=, Vejavnosti spita <, Število prometih, Število prometih, and Omogoči nakup. The rows contain data for various 'Varen' categories with numerical values in the decision columns and checkboxes in the final column.

#	Description	Kazenske tobe >=	Kazenske tobe <	Vejavnosti spita >=	Vejavnosti spita <	Število prometih	Število prometih	Omogoči nakup
1	Varen	0	7	0	6	0	6	<input checked="" type="checkbox"/>
2	Varen	0	7	6	21	6	11	<input checked="" type="checkbox"/>
3	Varen	0	7	6	21	0	6	<input checked="" type="checkbox"/>
4	Varen	0	7	21		11		<input checked="" type="checkbox"/>
5	Varen	0	7	21		6	11	<input checked="" type="checkbox"/>
6	Varen	0	7	21		0	6	<input checked="" type="checkbox"/>
7	Varen	7	13	0	6	0	6	<input checked="" type="checkbox"/>
8	Varen	7	13	6	21	6	11	<input checked="" type="checkbox"/>
9	Varen	7	13	6	21	0	6	<input checked="" type="checkbox"/>
10	Varen	7	13	21		6	11	<input checked="" type="checkbox"/>
11	Varen	7	13	21		0	6	<input checked="" type="checkbox"/>
12	Varen	13	19	0	6	0	6	<input checked="" type="checkbox"/>
13	Varen	13	19	6	21	6	11	<input checked="" type="checkbox"/>
14	Varen	13	19	6	21	0	6	<input checked="" type="checkbox"/>
15	Varen	13	19	21		6	11	<input checked="" type="checkbox"/>
16	Varen	13	19	21		0	6	<input checked="" type="checkbox"/>

Slika 7.4: Poslovno pravilo za pridobitev varnostne ocene voznika v obliki odločitvene tabele.

The screenshot shows the KIE Workbench interface with a 'Guided Decision Table [IzbiraAvtomobila]'. The table has 10 rows and 10 columns. The columns are: #, Description, ruleflow-group, date-effective, date-expires, Znamka avtomobila, Model avtomobila, Procentna motorje, Vrsta hibridne baterije, and Prodajna cena. The rows contain data for various car models like 'Eieler-Eni 1', 'Eieler-Eni 2', 'Eieler-Eni 3', 'Toyota', 'Honda', 'Tesla', 'ChevroletA', 'ChevroletB', 'ChevroletC', and 'ChevroletD' with associated dates and prices.

#	Description	ruleflow-group	date-effective	date-expires	Znamka avtomobila	Model avtomobila	Procentna motorje	Vrsta hibridne baterije	Prodajna cena
1	Eieler-Eni 1	izbiraavtomobila	01-Feb-2014	28-Feb-2014	Yogomo	MAE		ife	9900
2	Eieler-Eni 2	izbiraavtomobila	01-Feb-2014	28-Feb-2014	Yogomo	LU-EV-02		ife	9800
3	Eieler-Eni 3	izbiraavtomobila	01-Feb-2014	28-Feb-2014	Yogomo	LU-EV-03		m1	11800
4	Toyota	izbiraavtomobila	01-Feb-2014	28-Feb-2014	Toyota	Prus	1.8	plug-in	98000
5	Honda	izbiraavtomobila	01-Feb-2014	28-Feb-2014	Honda	Accord	2.0	plug-in	30000
6	Tesla	izbiraavtomobila	01-Feb-2014	28-Feb-2014	Tesla	Roadster	1.8	m1	84000
7	ChevroletA	izbiraavtomobila	01-Feb-2014	28-Feb-2014	Chevrolet	Aveo	1.2		11820
8	ChevroletA	izbiraavtomobila	01-Feb-2014	28-Feb-2014	Chevrolet	Aveo	1.4		12390
9	ChevroletC	izbiraavtomobila	01-Feb-2014	28-Feb-2014	Chevrolet	Cruze	1.6		14680
10	ChevroletC	izbiraavtomobila	01-Feb-2014	28-Feb-2014	Chevrolet	Cruze	1.4		17980

Slika 7.5: Poslovno pravilo za pridobitev prodajne cene avtomobila v obliki odločitvene tabele.

o višini subvencije” in ”Pridobi podatke o popustu”, se pridobijo podatki o višini državne subvencije, ki jo je mogoče unovčiti pri nakupu ustreznega avtomobila in podatki o višini popusta glede na nakupovalni status stranke. Podatki o višini subvencije za avtomobile, ki jih poganja (tudi) elektropogonski motor, so definirani v skripti ”PridobiVisinoSubvencije”, ki se izvede ob izvajanju opravila. Del skripte za določanje višine subvencije izgleda tako:

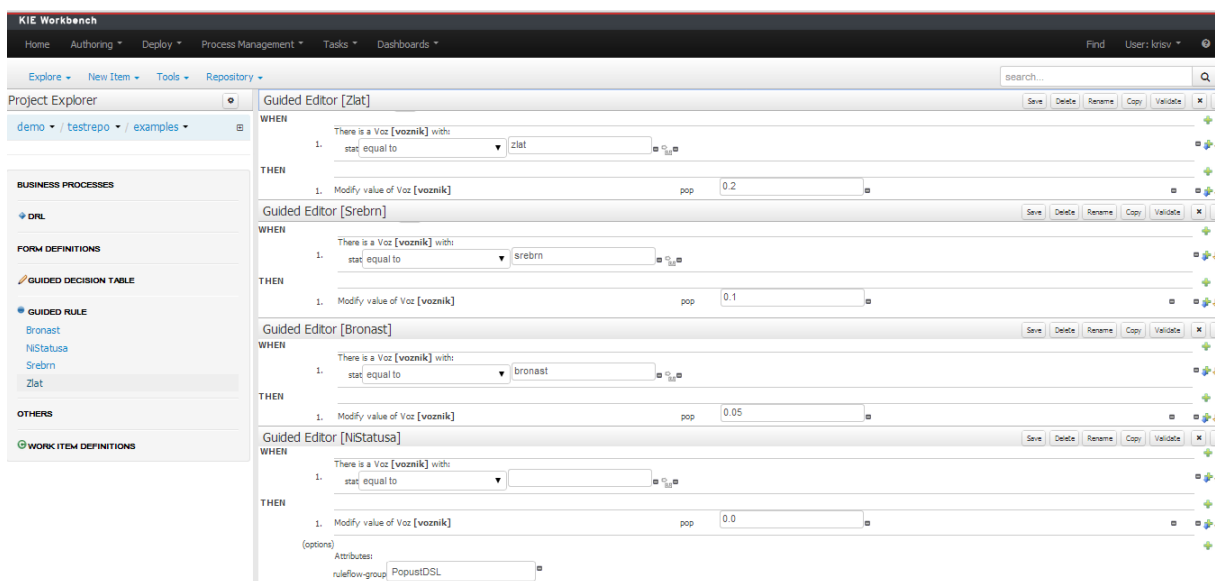
```
import org.jbpm.examples.Avt

// pravila in skupina poslovnih pravil
rule "set as m1" ruleflow-group "PridobiVisinoSubvencije"

// datum zacetka in konca veljavnosti skupine pravil
date-effective "01-Jan-2011 00:00"
date-expires "31-Dec-2014 23:59"

// del s pogoji
when
    $a : Avt(baterija== "m1")
// del s posledicami
then
    modify($a){
        setSub(5000.0);
    }
end

rule "set as n1orl7e" ruleflow-group "PridobiVisinoSubvencije"
when
    $a : Avt(baterija== "n1" || baterija== "l7e")
then
    modify($a){
        setSub(3000.0);
    }
end
```



Slika 7.6: Poslovna pravila, ki definirajo višino popusta.

}

end

Kot vidimo se glede na vnešene podatke o vrsti avtomobilske baterije iz prejšnjega uporabniškega opravila proži ustrezno poslovno pravilo, obenem pa se v entiteti, ki predstavlja voznika, nastavi atribut o višini subvencije.

V aktivnosti "Pridobi podatke o višini popusta" pa se s proženjem ustreznega poslovnega pravila pridobi podatek, s katerim je definirana višina popusta glede na voznikov nakupovalni status. Ta poslovna pravila so definirana v DSL urejevalniku. Njihov izgled je prikazan na sliki 7.6.

Tako smo pridobili vse podatke za izračun končne cene avtomobila. Ta se avtomatsko izračuna v skriptnem opravilu Izračunaj končno ceno avtomobila. Definicija za izračun končne cene avtomobila je v skriptnem opravilu definirana s formulo:

```
// inicializacija in posredovanje vrednosti nastavljenih
// spremenljivk
double prodCena = avt.getPcena();
```

```
double popust = voz.getPop();
double subvencija = avt.getSub();
double koncnaCena = 0.0;

// izracun koncne cene in nastavitvev atributa, ki hrani
// koncno ceno
koncnaCena = (prodCena - subvencija) * (1 - popust);
avt.setKcena(koncnaCena);
```

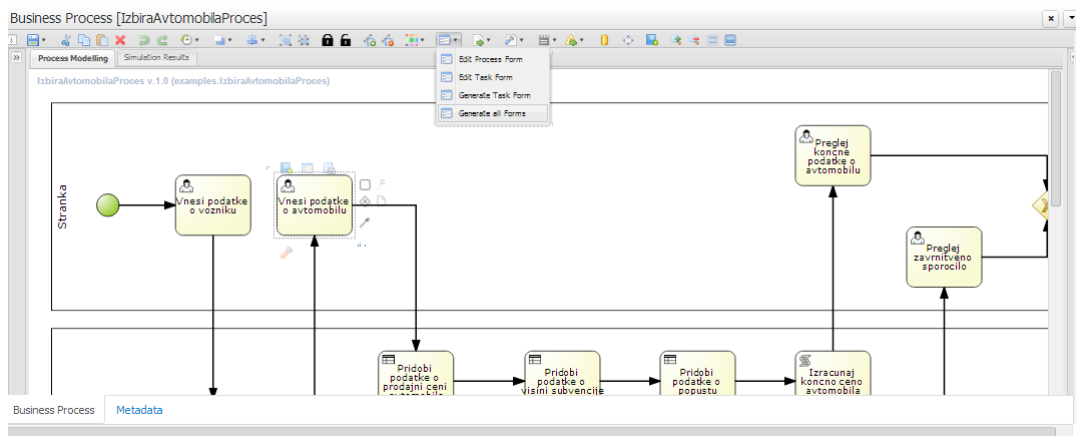
Končna cena se skupaj z vsemi preostalimi parametri izpiše v zadnji aktivnosti "Preglej končne podatke o avtomobilu". Stranka s potrditvijo, da je dobila končne podatke zaključi proces.

Obstaja pa tudi alternativni procesni tok, in sicer v primeru da voznik ne ustreza pogojem za nakup avtomobila. V tem primeru atribut, ki hrani upravičenost do nakupa, ostane v zavrnitvenem stanju. Posledično se izbere spodnja veja procesa, kjer se v naslednjem skriptnem opravilu "Nastavi zavrnitveno sporočilo" generira sistemsko sporočilo z obrazložitvijo okoliščin, nato pa se to sporočilo v opravilu "Preglej zavrnitveno sporočilo" posreduje na uporabnikov zaslon. Ko uporabnik potrdi, da je sporočilo prebral, se proces "IzbiraAvtomobila" zaključi.

Pred izvajanjem samega procesa moramo postoriti še eno stvar. To je izdelava obrazcev za vnos potrebnih podatkov v uporabniških opravilih. Lahko se odločimo za samostojno izdelavo obrazcev, okolje KIE-WB pa nam omogoča tudi njihovo avtomatsko generiranje. To prikazuje slika 7.7.

## 7.2 IZVAJANJE

Izvajanje procesa začnemo s pritiskom na gumb "Build & Deploy", ki nam omogoči izgradnjo procesa in njegov prenos na interni Maven repozitorij. To nam prikazuje slika 7.8. Sedaj lahko zaženemo naš proces. Izberemo ga med možnimi procesi in ga sprožimo z definiranjem nove instance (slika 7.9). Pojavi se začetni procesni obrazec. S pritiskom na gumb se začne izvajanje



Slika 7.7: Gumb za avtomatsko izdelavo obrazcev.

našega poslovnega procesa s prvo aktivnostjo "Vnesi podatke o vozniku".

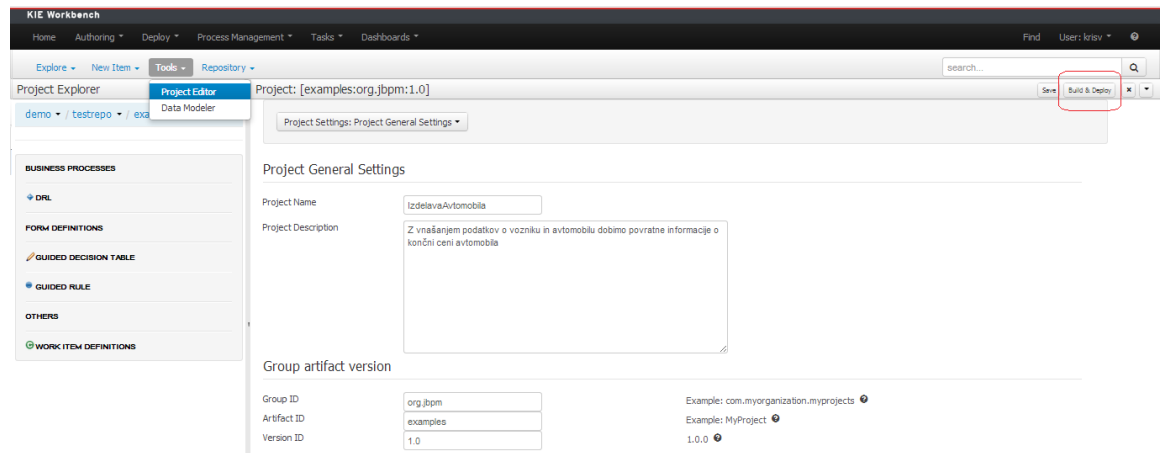
Izpolnimo obrazec z vnosom podatkov in pritisnemo na gumb "Complete". Potek je prikazan na sliki 7.10. Temu sledi naslednja aktivnost "Vnesi podatke o avtomobilu", ki je prikazana na sliki 7.10. Tudi tu vnesemo zelene podatke in nadaljujemo s procesom.

Sledi še zadnja aktivnost "Preglej končne podatke o avtomobilu", kjer se nam izpišejo podatki o končni ceni in lastnostih avtomobila. Prikaz le-teh je na sliki 7.12.

Zelo pomembna lastnost pristopa s poslovnimi pravili je njegova enostavna spremenljivost. Če želimo spremeniti katero od poslovnih pravil, to naredimo na zelo enostaven način brez vsakega poseganja v poslovni proces. Denimo, da bi se povečala subvencija za avtomobile, ki imajo vgrajen elektropogonski motor in jih poganja avtomobilska baterija tipa m1 - to storimo z enostavno spremembo v skripti "PridobiVisinoSubvencije", ki definira poslovna pravila o višini subvencij.

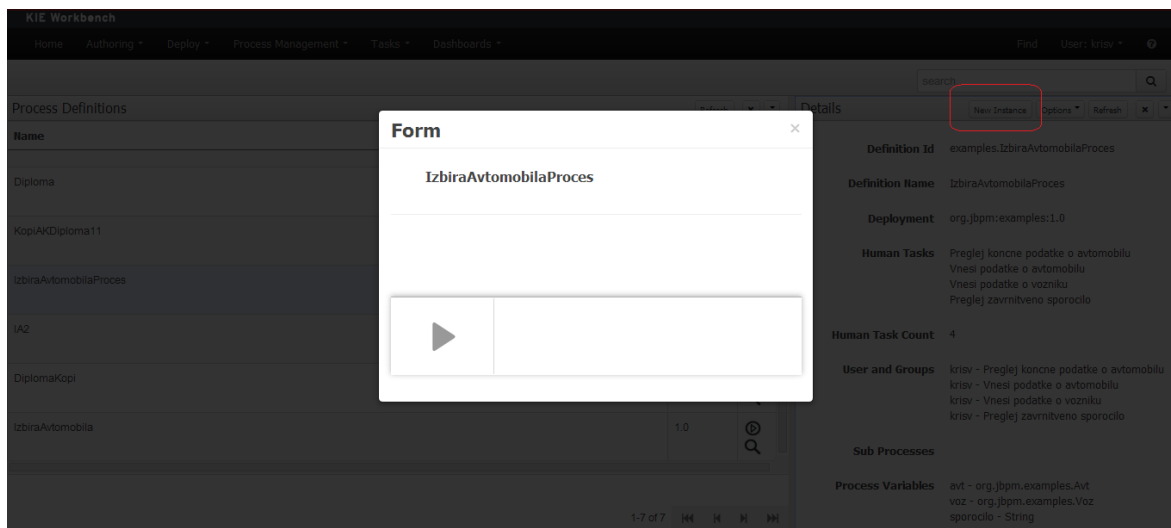
```
// spremenimo visino subvencije iz 5000 na 10000 enot
setSub(10000.0);
```

Zatem moramo samo še enkrat zgraditi naš projekt, ga naložiti na repozitorij in izvesti. Ob vnašanju enakih podatkov dobimo pri končnem pregledu

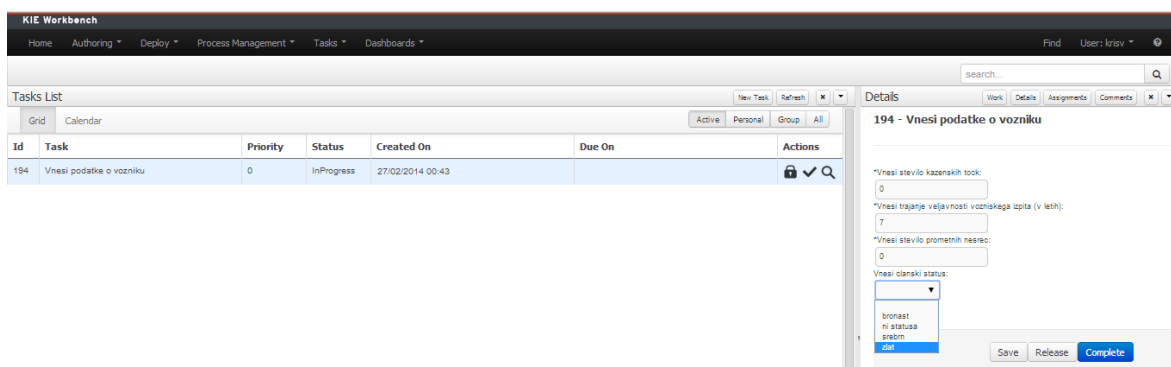


Slika 7.8: Priprava projekta na izvajanje.

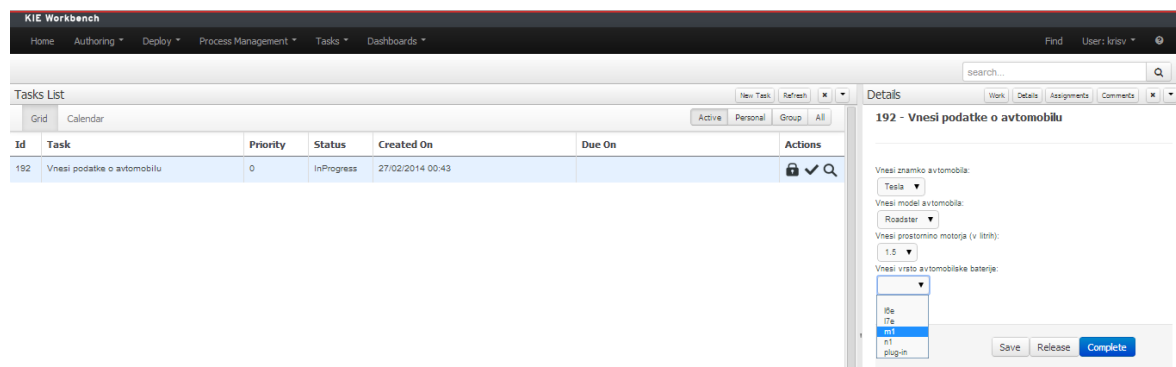
drugačno vrednost kot v prvotnem primeru. Končni podatki so prikazani na sliki 7.13.



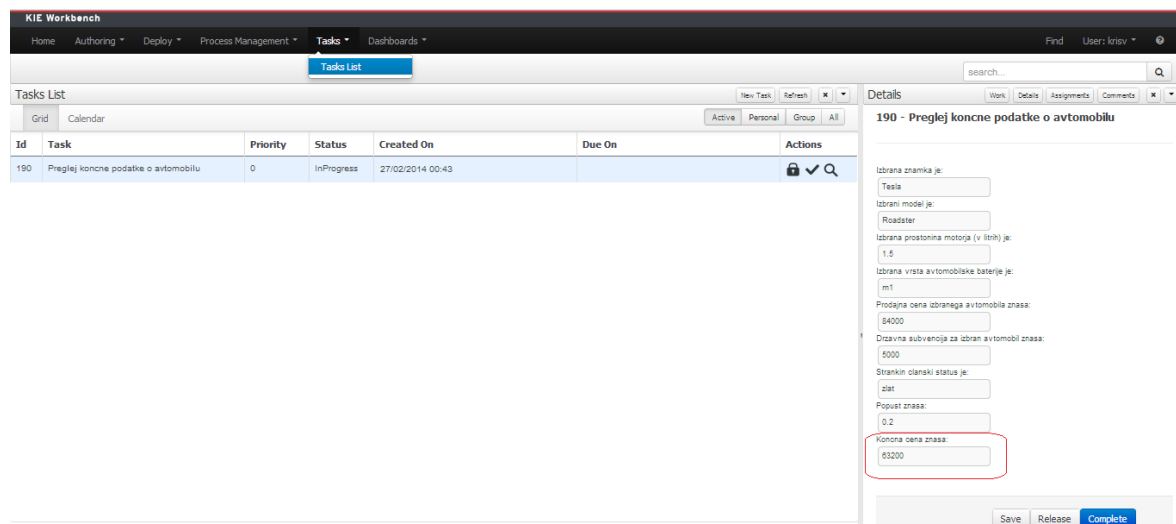
Slika 7.9: Izbira procesa in proženje nove instance.



Slika 7.10: Aktivnost Vnesi podatke o vozniku.



Slika 7.11: Aktivnost Vnesi podatke o avtomobilu.



Slika 7.12: Aktivnost Preglej končne podatke o avtomobilu.

The screenshot displays the KIE Workbench interface. The top navigation bar includes 'Home', 'Authoring', 'Deploy', 'Process Management', 'Tasks', and 'Dashboards'. A search bar is located in the top right corner. The main area is divided into two sections: 'Tasks List' on the left and 'Details' on the right.

**Tasks List:**

Id	Task	Priority	Status	Created On	Due On	Actions
199	Preglej končne podatke o avtomobilu	0	InProgress	27/02/2014 00:57		🔒 🔍

**Details:**

**199 - Preglej končne podatke o avtomobilu**

Izbrana znamka je:

Izbrani model je:

Izbrana prostornina motorja (v litrih) je:

Izbrana vrsta avtomobilske baterije je:

Prodajna cena izbranega avtomobila znasa:

Dizajna subvencija za izbran avtomobil znasa:

Stavka obdavčila status je:

Popust znasa:

Končna cena znasa:

Buttons: Save, Release, Complete

Slika 7.13: Aktivnost Preglej končne podatke o avtomobilu po spremembi višine subvencije.



## Poglavje 8

# ZAKLJUČEK

Organizacije morajo svoje poslovne procese prilagajati okoliščinam v katerih se nahajajo. Za preživetje morajo znati slediti razmeram na trgu in izkoristiti ponujene poslovne priložnosti. Stremeti morajo k poslovni agilnosti, ki s seboj prinaša večjo prilagodljivost in učinkovitost informacijske infrastrukture. Pomembno vlogo pri tem igrajo povezave med ljudmi, procesi in tehnologijo. Spoznali smo, da je mogoče do zastavljenih ciljev priti z dobrim definiranjem in avtomatizacijo poslovnih procesov. Ta pojma skupaj zajemata njihovo upravljanje in uporabo pristopa s poslovnimi pravili v povezavi s storitveno orientirano arhitekturo. Pri doseganju zastavljenih ciljev nam zelo pomagata jezik za modeliranje in izvajanje poslovnih procesov - BPMN 2.0 in definiranje poslovnih pravil z uporabo BRMS sistemov, kateri omogočijo preglednost nad odločitvami in njihovo preprosto spremenljivost. Zaradi splošne razumljivosti poslovnih pravil in dejstva, da se jezik BPMN 2.0 uporablja tako za modeliranje kot tudi za izvajanje poslovnih procesov, je olajšano sodelovanje med poslovnimi strokovnjaki in tehničnim osebjem. Vsi namreč delajo z istim modelom procesa, kar zmanjša prepada med aplikacijsko kodo in poslovnimi zahtevami. Definicija procesa in odločitev je tako v rokah za to pristojnih poslovnih strokovnjakov. S tem je združbam omogočena enostavnejša in hitrejša realizacija zastavljenih poslovnih ciljev. Na trgu obstaja veliko ponudnikov BRMS sistemov, vendar je še bolj kot za katerega

pomembno, da se organizacija sploh odloči za njegovo uporabo. V osnovi so njihovi najpomembnejše deli repozitorij poslovnih pravil, ki omogoča shranjevanje poslovnih pravil ločeno od aplikacijske kode preostalega sistema, orodja, ki omogočajo definiranje in upravljanje poslovne logike ter komponenta, ki je odgovorna za izvrševanje poslovnih pravil.

Za konec moramo poudariti še to, da zgolj uvedba BRMS sistemov ne bo rešila vseh obstoječih poslovnih težav. Ti sistemi so le del celotne rešitve in pomagajo zmanjšati kompleksnost in izboljšati učinkovitost izvajanja poslovnega procesa, bo pa njihova uporaba vsekakor maksimizirala možnosti za uspeh in omogočila sledenje zastavljenim ciljem združbe.

# Slike

2.1	Arhitektura spletnih storitev [26]. . . . .	8
2.2	Življenjski cikel poslovnih procesov [27]. . . . .	11
3.1	Elementi procesnega toka [28]. . . . .	18
3.2	Aktivnost poslovno pravilo. . . . .	18
3.3	Povezovalni elementi [28]. . . . .	19
3.4	Prikaz bazena in prog [28]. . . . .	19
3.5	Prikaz specificiranih artefaktov v BPMN 2.0 [28]. . . . .	20
3.6	Primer poslovnega procesa sprejemanja naročil. . . . .	21
4.1	Prikaz klasifikacije poslovnih pravil [16]. . . . .	26
4.2	Prikaz nadaljnje razdelitve pravil [16]. . . . .	26
4.3	Aktivnost vpeljave pravil v informacijski sistem [14]. . . . .	33
4.4	Slika prikazuje izvajanje aktivnosti v ciklih [14]. . . . .	35
4.5	Stanja v življenjskem ciklu poslovnih pravil [19]. . . . .	37
5.1	Komponente tipičnih BRMS [1]. . . . .	43
5.2	Primer odločitvenega drevesa. . . . .	47
6.1	Prikaz modelirnega in izvajalnega okolja KIE. . . . .	56
7.1	Izdelava novega projektnega repozitorija. . . . .	58
7.2	Predstavitev entitete Avt, ki predstavlja avtomobil. . . . .	58
7.3	Procesni diagram IzdelavaAvtomobila. . . . .	60

---

7.4	Poslovno pravilo za pridobitev varnostne ocene voznika v obliki odločitvene tabele. . . . .	61
7.5	Poslovno pravilo za pridobitev prodajne cene avtomobila v obliki odločitvene tabele. . . . .	61
7.6	Poslovna pravila, ki definirajo višino popusta. . . . .	63
7.7	Gumb za avtomatsko izdelavo obrazcev. . . . .	65
7.8	Priprava projekta na izvajanje. . . . .	66
7.9	Izbira procesa in proženje nove instance. . . . .	67
7.10	Aktivnost Vnesi podatke o vozniku. . . . .	67
7.11	Aktivnost Vnesi podatke o avtomobilu. . . . .	68
7.12	Aktivnost Preglej končne podatke o avtomobilu. . . . .	68
7.13	Aktivnost Preglej končne podatke o avtomobilu po spremembi višine subvencije. . . . .	69

# Tabele

4.1	STEP principi in njihov namen [16]. . . . .	30
5.1	Primer odločitvene tabele [29]. . . . .	47



# Literatura

- [1] M. B. Jurič, P. Kapil (2008). Business Process Driven SOA using BPMN and BPEL. Birmingham.
- [2] M. B. Jurič. Storitvena arhitektura - zgolj kompozicija spletnih storitev? Avgust 2005. Dostopno na:  
[http://www.soa.si/juric/soa\\_ss.pdf](http://www.soa.si/juric/soa_ss.pdf) (oktober 2013).
- [3] I. Gorton (2010). Essential Software Architecture Second Edition. Springer.
- [4] R. R. Kodali. What is service-oriented architecture? Junij 2005. Dostopno na:  
<http://www.javaworld.com/javaworld/jw-06-2005/jw-0613-soa.html?page=1> (oktober 2013).
- [5] The Open Group. Service Oriented Architecture : SOA Features and Benefits. Dostopno na:  
[http://www.opengroup.org/soa/source-book/soa/soa\\_features.htm](http://www.opengroup.org/soa/source-book/soa/soa_features.htm) (oktober 2013).
- [6] T. Erl (2010). SOA Design Patterns. Prentice Hall.
- [7] Gartner. Hype Cycle for Business Process Management. Julij 2011. Dostopno na:  
[http://www.adeptia.com/products/Hype\\_cycle\\_BPM\\_2011.pdf](http://www.adeptia.com/products/Hype_cycle_BPM_2011.pdf) (november 2013).

- [8] N. Malik. How is Business Process Management related to Service Oriented Architecture? Julij 2007. Dostopno na:  
<http://blogs.msdn.com/b/nickmalik/archive/2007/07/26/how-is-business-process-management-related-to-service-oriented-architecture.aspx> (november 2013).
- [9] Wikipedia. Business Process Model and Notation. Dostopno na:  
[http://en.wikipedia.org/wiki/Business\\_Process\\_Model\\_and\\_Notation](http://en.wikipedia.org/wiki/Business_Process_Model_and_Notation) (oktober 2013).
- [10] F. Leymann. BPEL vs BPMN 2.0: Should you care? December 2009. Dostopno na:  
<http://leymann.blogspot.com/2009/12/bpel-vs-bpmn-20-should-you-care.html> (oktober 2013).
- [11] BPMLab. BPMN. Dostopno na:  
[http://www.bpmlab.si/index.php?Itemid=65&catid=43:modeliranje&id=72:bpmn&option=com\\_content&view=article](http://www.bpmlab.si/index.php?Itemid=65&catid=43:modeliranje&id=72:bpmn&option=com_content&view=article) (oktober 2013).
- [12] Berliner BPM-Offensive. BPMN 2.0 Poster. Dostopno na:  
[http://www.bpmb.de/images/BPMN2\\_0\\_Poster\\_EN.pdf](http://www.bpmb.de/images/BPMN2_0_Poster_EN.pdf) (oktober 2013).
- [13] Wikipedia. Business rule. Dostopno na:  
[http://en.wikipedia.org/wiki/Business\\_rules](http://en.wikipedia.org/wiki/Business_rules) (oktober 2013).
- [14] J. Boyer, H. Mili (2011). Agile Business Rule Development. Springer.
- [15] Business Rules Group. Defining Business Rules - What Are They Really? Dostopno na:  
<http://www.businessrulesgroup.org/home-brg.shtml> (oktober 2013).
- [16] B. von Halle (2002). Business Rules Applied - Business Better Systems Using the Business Rules Approach. John Wiley & Sons.

- 
- [17] The Data Administration Newsletter. What is a Business Rules Approach? Januar 2002. Dostopno na:  
<http://www.tdan.com/view-articles/4983/> (oktober 2013).
- [18] EPF Wiki. Concept: Cycle Approach to Rules Development. November 2012. Dostopno na:  
[http://epf.eclipse.org/wikis/epfpractices/practice.tech.abrd.base/guidances/concepts/cycle\\_approach\\_7FA44D8F.html](http://epf.eclipse.org/wikis/epfpractices/practice.tech.abrd.base/guidances/concepts/cycle_approach_7FA44D8F.html) (november 2013).
- [19] Primatek. Business Rules Governance and management. November 2009. Dostopno na:  
<http://www.primatek.ca/blog/2009/11/23/business-rules-governance-and-management-part-vi-rule-life-cycle/> (november 2013).
- [20] Agile Alliance. Agile manifesto. Februar 2001. Dostopno na:  
<http://agilemanifesto.org/iso/sl/> (november 2013).
- [21] I. Graham (2007). Business Rules Management and Service Oriented Architecture: A Pattern Language. John Wiley & Sons.
- [22] Hartmann Software Group. Business Rule Management System. Maj 2012. Dostopno na:  
<http://www.hartmannsoftware.com/pub/Enterprise-Rule-Applications/brms> (november 2013).
- [23] Wikipedia. Business rule management system. Dostopno na:  
[http://en.wikipedia.org/wiki/Business\\_rule\\_management\\_system](http://en.wikipedia.org/wiki/Business_rule_management_system) (november 2013).
- [24] InfoQ. Implementation of business rules and business processes in SOA. Marec 2007. Dostopno na:  
<http://www.infoq.com/articles/business-rules-processes> (november 2013).

- [25] Business Rule Group. The Business Rules Manifesto. November 2003. Dostopno na: <http://www.businessrulesgroup.org/brmanifesto.htm> (november 2013).
- [26] Arhitektura spletnih storitev. Dostopno na: <http://portal.odines.ws/?p=2292> (december 2013).
- [27] Oracle. Process-Driven SOA Development. Avgust 2009. Dostopno na: <http://www.oracle.com/technetwork/articles/soa/process-driven-soa-100743.html> (december 2013).
- [28] OMG. Business Model Process and Notation, version 2.0. Januar 2011. Dostopno na: <http://www.omg.org/spec/BPMN/2.0/PDF/> (december 2013).
- [29] Open Rules. Rule Engine - Rule Execution Logic with OpenRulesEngine. Dostopno na: <http://openrules.com/ruleengine.htm#Execution%20Logic> (december 2013).
- [30] Wolfgang Martin Team. Role of Business Rules in SOA – Industrializing Business Process in Agile Enterprises. December 2006. Dostopno na: [http://www.bosch-si.com/media/en/bosch\\_software\\_innovations/documents/white\\_paper/brm/2006\\_role\\_business\\_rules\\_in\\_soa.pdf](http://www.bosch-si.com/media/en/bosch_software_innovations/documents/white_paper/brm/2006_role_business_rules_in_soa.pdf) (december 2013).
- [31] BPMInstitute.org. BPMS Watch: Five Things to Love About BPMN 2.0. Dostopno na: <http://www.bpminstitute.org/resources/articles/bpms-watch-five-things-love-about-bpmn-20> (december 2013).
- [32] Wikipedia. Forward chaining. Januar 2014. Dostopno na: [http://en.wikipedia.org/wiki/Forward\\_chaining](http://en.wikipedia.org/wiki/Forward_chaining) (januar 2014).

- 
- [33] Wikipedia. Rete algorithm. December 2013. Dostopno na:  
[http://en.wikipedia.org/wiki/Rete\\_algorithm#Overview](http://en.wikipedia.org/wiki/Rete_algorithm#Overview) (januar 2014).
- [34] M. Proctor. R.I.P. RETE time to get PHREAKY. November 2013. Dostopno na:  
<http://blog.athico.com/2013/11/rip-rete-time-to-get-phreaky.html> (januar 2014).
- [35] JBoss Community. Drools Documentation. Dostopno na:  
[http://docs.jboss.org/drools/release/6.0.0.Final/drools-docs/html\\_single/index.html#d0e23](http://docs.jboss.org/drools/release/6.0.0.Final/drools-docs/html_single/index.html#d0e23) (januar 2014).
- [36] BizRules. BRE Family Tree update shows IBM ILOG acquisition. Julij 2008. Dostopno na:  
[http://bizrules.info/weblog/bre\\_brms\\_vendors/](http://bizrules.info/weblog/bre_brms_vendors/) (januar 2014).
- [37] Wikipedia. IBM Operational Decision Management. November 2013. Dostopno na:  
[http://en.wikipedia.org/wiki/IBM\\_Operational\\_Decision\\_Management](http://en.wikipedia.org/wiki/IBM_Operational_Decision_Management) (januar 2014).
- [38] IBM. IBM Business Process Management and Operational Decision Management. Julij 2013. Dostopno na:  
<http://www.youtube.com/watch?v=bJHov82TXEc> (januar 2014).
- [39] IBM. What's New in IBM Operational Decision Manager V8.5. Dostopno na:  
<http://www-01.ibm.com/software/websphere/subscriptionandsupport/compare-odm-versions.html> (januar 2014).
- [40] Oracle. Oracle Fusion Middleware. Dostopno na:  
<http://www.oracle.com/technetwork/middleware/>

[fusion-middleware/overview/index.html?ssSourceSiteId=ocomen](http://fusion-middleware/overview/index.html?ssSourceSiteId=ocomen)  
(januar 2014).

- [41] Oracle. Oracle® Fusion Middleware User's Guide for Oracle Business Rules 11g Release 1 (11.1.1.5.0). Dostopno na:  
[http://docs.oracle.com/cd/E21764\\_01/integration.1111/e10228/intro.htm#BHCJGCBG](http://docs.oracle.com/cd/E21764_01/integration.1111/e10228/intro.htm#BHCJGCBG) (januar 2014).
- [42] Oracle. Oracle Business Rules. Dostopno na:  
<http://www.oracle.com/technetwork/middleware/business-rules/overview/index.html> (januar 2014).
- [43] Open Rules. Product Components. Dostopno na:  
<http://openrules.com/components.htm> (januar 2014).
- [44] Bloor. OpenRules release new version. Oktober 2012. Dostopno na:  
<http://www.bloorresearch.com/analysis/openrules-release-version/> (januar 2014).
- [45] Wikipedia. JBoss (company). November 2013. Dostopno na:  
[http://en.wikipedia.org/wiki/JBoss\\_%28company%29](http://en.wikipedia.org/wiki/JBoss_%28company%29) (januar 2014).
- [46] JBoss Community. Drools - The Business Logic integration Platform. Dostopno na:  
<http://www.jboss.org/drools> (januar 2014).
- [47] K. M. Chandy, W.R. Schulte (2010). Event Processing - Designing IT Systems for Agile Companies. Mcgraw-Hill.
- [48] Wikipedia. Apache Maven. Februar 2013. Dostopno na:  
[http://en.wikipedia.org/wiki/Apache\\_Maven](http://en.wikipedia.org/wiki/Apache_Maven) (februar 2014).