



DARBO SU IP RUOŠINIU METODIKA

Turinys

1	Integracinės platformos aprašymas	3
1.1	Naudojamos technologijos	3
1.2	Platformos struktūra.....	3
2	Darbo su integracine platforma aprašymas	5
2.1	Platformos konfigūravimas	5
2.1.1	Verslo procesų valdymo sistema	5
2.2	Verslo procesų specifikavimas	7
2.2.1	Integracinėje platformoje naudojami verslo procesų elementai	7
2.2.2	Verslo procesų modeliavimas.....	9
2.3	Sistemų integravimas	10
2.4	Naudotojų rolių valdymas	12
2.5	Darbo su integracinės platformos ruošiniais metodika.....	13

1 INTEGRACINĖS PLATFORMOS APRAŠYMAS

Šioje dalyje yra aprašomos integracinės platformos pagrindiniai moduliai ir technologijos, kurios buvo panaudotos integracinės platformos prototipo realizavimui.

1.1 Naudojamos technologijos

Integracinės platformos prototipui realizuoti buvo naudojamos tik atviro kodo laisvai prieinamos technologijos, kurios buvo atrinktos palyginamojo tyrimo metu.

ServiceMix - pranešimų perdavimo ir sistemų integravimo magistralė skirta sistemoms integruoti tarpusavyje į viena bendrą duomenų perdavimo ir paslaugų teikimo kanalą. Integracinėje platformoje naudojama ServiceMix 4.5.1 versija.

Maven - konfigūracijų valdymo įrankis skirtas integracinės sistemos modulinės struktūros palaikymui ir automatiniam sistemos modulių testavimui, surinkimui ir diegimui. Naudojama Maven 3.05 versija.

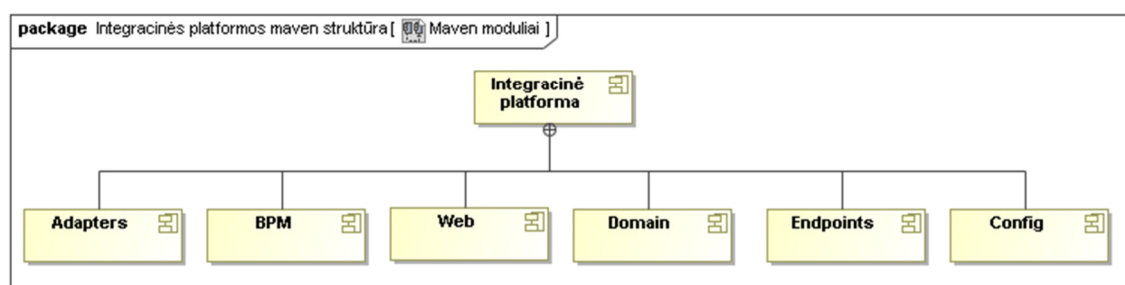
Java - integracinę platformą realizuota Java programavimo kalba. Realizuojant integracinę platformą panaudota Java 1.6 kalbos versija.

jBPM - JBoss kompanijos sukurta verslo procesų valdymo ir vykdymo sistema. Ši sistema integracinėje platformoje atlieka verslo procesų specifikavimų vykdymą. Integracinėje platformoje naudojama jBPM 5.4.0 versija.

Eclipse - programavimo aplinka naudota integracinės platformos programavimo darbams atlikti. Šios aplinkos BPM Designer įskiepis buvo naudojamas verslo procesų specifikacijoms modeliuoti. Integracinės platformos programavimo ir verslo procesų modeliavimo metu buvo naudota Eclipse Helios versija.

1.2 Platformos struktūra

Verslo procesų valdymo integracinę platformą sudaro šeši pagrindiniai moduliai (Pav 1.).



Pav. 1. Integracinės platformos modulinė struktūra

Adapters - šiame modulyje yra saugomi ServiceMix pranešimų magistralės komponentai, kurie yra naudojami integracijoms su išorinėmis sistemomis realizuoti. Kiekvienai integruotai išoriniai sistemai yra atitinkamas ServiceMix adapterio komponentas.

BPM - tai modulis atsakingas už verslo procesų specifikacijų saugojimą ir jų vykdymą. Šiame modulyje yra saugomos procesų specifikacijos aprašytos BPMN 2.0 standartu ir verslo procesų vykdymo sistemos nustatymai.

Web - šiame modulyje yra saugoma tinklalapio priemonėmis realizuota naudotojo grafinė sąsaja. Šios grafinės sąsajos pagalba naudotojai atlieka jiems skirtas duomenų įvedimo, peržiūros ir patvirtinimo procesų užduotis.

Domain - tai modulis, kuriame yra saugoma tam tikros srities koncepcinės duomenų struktūros realizacija.

Endpoints - modulis saugojantis ServiceMix magistralės tiekiamų paslaugų sąsajas. Šio modulio sąsajos naudojamos ServiceMix magistralės paslaugų iškvietimui iš verslo procesų valdymo sistemos.

Config - šiame modulyje yra saugomos bendros integracinės platformos konfigūracijos, kurios gali būti keičiamos pritaikant integracinę platformą konkrečiam integravimo sektoriui.

2 DARBO SU INTEGRACINE PLATFORMA APRAŠYMAS

Šiame skyriuje aprašomas integracinės platformos konfigūravimas ir darbo su integracine platforma ypatumai.

2.1 Platformos konfigūravimas

2.1.1 Verslo procesų valdymo sistema

Ši sistema yra konfigūruojama nustatymų failų pagalba. Pagrindinis nustatymų failas yra **jbpm.properties**, kuris yra saugomas /bpm/src/main/resources kataloge. Šio failo pagalba galima nustatyti šias sistemos savybes:

Savybės pavadinimas	Savybės aprašymas	Savybės reikšmė
persistence.datasource.name	duomenų šaltinio pavadinimas	jdbc/jbpm-ds
persistence.datasource.user	duomenų šaltinio vartotojo vardas	sa
persistence.datasource.password	duomenų šaltinio slaptažodis	
persistence.datasource.url	duomenų šaltinio adresas	jdbc:h2:tcp://localhost/~jbpm-db
persistence.datasource.driverClassName	duomenų šaltinio valdiklio klasės pavadinimas	org.h2.Driver
persistence.enabled	duomenų saugojimas bazėje įjungtas	true
persistence.persistenceunit.name	duomenų saugojimo bibliotekos pavadinimas	org.jbpm.persistence.jpaa
persistence.persistenceunit.dialect	duomenų saugojimo dialekto klasė	org.hibernate.dialect.H2Dialect
taskservice.enabled	naudotojų užduočių serveris įjungtas	true
taskservice.datasource.name	naudotojų užduočių serverio duomenų saugyklos pavadinimas	org.jbpm.task
taskservice.transport	naudotojų užduočių serverio pranešimų protokolas	hornetq
taskservice.usergroupmapping	užduočių serverio naudotojų rolių susiejimo nustatymo failas	classpath:/telco-usergroups.properties
taskservice.usergroupcallback	naudotojų rolių paslaugos tiekėjo realizacijos klasė	com.nortal.telco.controllers.SpringUserGroupCallbackImpl
application.context	Spring konteksto failas	classpath:/telco-jbpmserver-context.xml
hibernate.jndi.class	hibernate duomenų saugojimo karkaso kūrėjo klasė	bitronix.tm.jndi.BitronixInitialContextFactory
hibernate.jndi.url	nuoroda į duomenų saugojimo	

	karkaso kūrėjo klasė	
processes.dir	verslo procesų saugojimo katalogo pavadinimas	processes
processes.use.filesystem	naudoti procesus iš katalogo ar iš war archyvo	false

Verslo procesų valdymo sistemos integracija su ServiceMix paslaugomis yra konfigūruojama Spring karkaso priemonėmis. Spring karkaso kontekstas yra konfigūruojamas naudojant **jbpmserver-context.xml** failą, kuris yra saugomas /bpm/src/main/resources kataloge. Šiame faile yra aprašomi ServiceMix paslaugų Java klientų savybės: kliento identifikatorius, kliento paslaugos adresas, paslaugos sąsajos pavadinimas ir sąsajos aprašymo wsdl failas. Pavyzdinis ServiceMix kliento konfigūravimas Spring karkaso konteksto priemonėmis pateiktas žemiau.

```
<jaxws:client
id="accountService"
address="http://lvpaip:9000/telco/services/accountService"
serviceClass="com.nortal.lvpa.telco.endpoints.account.service.AccountService"
wsdlLocation="classpath:wsdl/account-service-interface/account-service-interface.wsdl"
/>
```

Integracijos sistemos pranešimų spausdinimas į žurnalą atliekamas naudojant slf4j sąsają realizuojantį Logback karkasą. Logback žurnalizavimo karkasas yra konfigūruojamas **logback.xml** failo pagalba, kuris yra saugomas src/main/resources kataloge. Šiame faile galima nurodyti norimą žurnalizavimo pranešimų lygį, žurnalų saugojimo ir išvedimo būdus. Pavyzdinis žurnalo konfigūravimas Logback karkaso priemonėmis pateiktas žemiau.

```
<configuration>
<appender name="CONSOLE" class="ch.qos.logback.core.ConsoleAppender">
<layout class="ch.qos.logback.classic.PatternLayout">
<Pattern>%p{6}: %d{HH:mm:ss} %c - %msg%n</Pattern>
</layout>
</appender>
<logger name="org" additivity="false">
<level value="ERROR"/>
<appender-ref ref="CONSOLE"/>
</logger>
</configuration>
```

2.2 Verslo procesų specifikuojimas

2.2.1 Integracinėje platformoje naudojami verslo procesų elementai

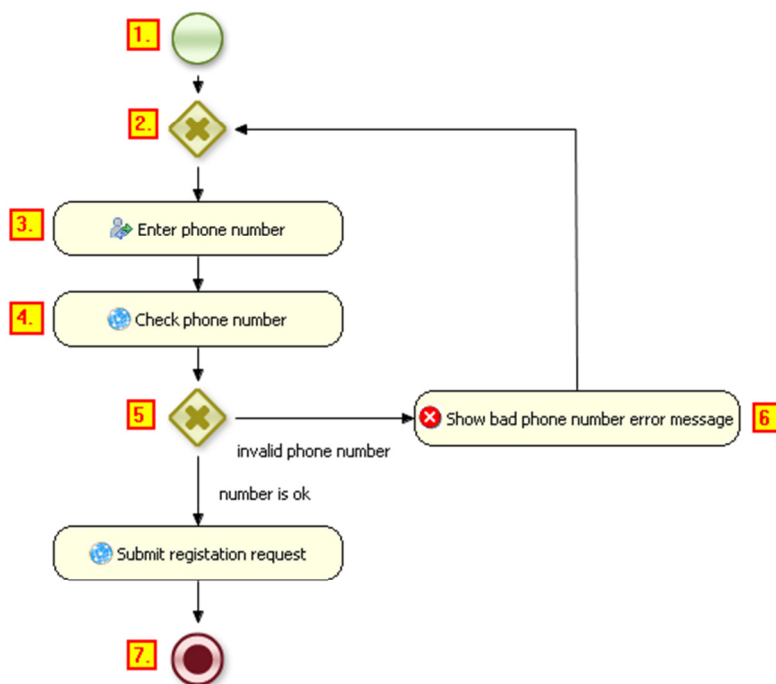
Verslo procesų valdymo sistemoje verslo procesai yra specifikuojami naudojant standartinę BPMN 2.0 notaciją. Kiekvienas verslo procesas turi tokias savybes:

- ID (proceso identifikatorius). Šis identifikatorius yra unikalus proceso pavadinimas, kuri gali būti naudojamas norint pradėti vykdyti verslo procesą.
- Variables (kintamieji). Ši savybė, tai sąrašas kintamųjų kuriuos galima naudoti šio verslo proceso vykdymo metu duomenims saugoti ir apdoroti. Kiekvienas kintamasis turi pavadinimą ir tipą. Tipas gali būti primitivus (Integer, Float, String) arba sudėtinis (Object). Nurodant kintamojo Object tipą reikalinga nurodyti ir pilną klasės pavadinimą, kuri bus naudojama kaip šio kintamojo tipo realizacija.

Integravimo verslo procesams specifikuoti naudojami BPMN notacijos elementai, kurie yra pateikti paveiksle 2:

1. Proceso pradžios taškas - žymi proceso vykdymo sekos pradžią ir pirmą veiksmą. Kiekviename verslo procese privalo būti bent vienas proceso pradžios taškas.
2. Sekos suliejimo taškas - skirtas keletui proceso vykdymo sekų sulieti į vieną seką.
3. Naudotojo užduoties veiksmas (User task) - šio veiksmo metų, naudotojas, turi atlikti tam tikrą užduotį, kuri gali būti įvesti reikiamus duomenis, peržiūrėti duomenis arba juos patvirtinti. Naudotojo užduoties atlikimui integracinė platforma pateikia naudotojui grafinę naudotojo sąsają realizuotą tinklalapio pavidalu. Verslo proceso vykdymo metu, verslo procesų valdymo sistema pasiekusi naudotojo užduoties veiksmą, sustabdo vykdomą procesą ir tęsia jo vykdymą tik tada kai naudotojas atlieką jam priskirtą užduotį. Naudotojo užduoties veiksmas turi tokias savybes:
 - ActorId (aktoriaus identifikatorius). Nurodo aktorių, kuris gali atlikti naudotojo užduotį.
 - Parameter Mapping (parametrų susiejimas). Susieja proceso kintamuosius su naudotojo užduoties įėjimo parametrais.
 - Result Mapping (rezultato susiejimas). Susieja naudotojo atliktos užduoties kintamuosius su verslo proceso kintamaisiais.
 - Skippable (praleidžiamas veiksmas). Nurodo ar naudotojo veiksmas yra privalomas, ar naudotojas gali jį praleisti ir pereiti prie kitų veiksmų.
 - TaskName (naudotojo veiksmo pavadinimas). Naudotojo užduoties veiksmo pavadinimas, pagal kurį naudotojo sąsajos modulis parenka atitinkama naudotojo sąsają veiksmui atlikti.
4. Paslaugos kvietimo veiksmas (Service Call Task) - tai veiksmas skirtas iškviešti ServiceMix pranešimų magistralės tiekiamos paslaugos metodą. Šis veiksmas nėra standartinis BPMN notacijos veiksmas, tai notacijos praplėtimas, skirtas supaprastinti paslaugų iškvietimą verslo proceso vykdymo metu. Paslaugos kvietimo veiksmas turi tokias savybes:
 - Method (paslaugos metodas). Paslaugos metodo, kuri reikia iškviešti pavadinimas.
 - Parameter Mapping (parametrų susiejimas). Susieja proceso kintamuosius su paslaugos kvietimo veiksmas įėjimo parametrais.
 - ReturnTo (rezultatų grąžinimo kintamasis). Nurodo proceso kintamąjį į kurį bus grąžintas metodo vykdymo rezultatas. Proceso kintamojo galima ir nenurodyti jeigu kviečiamas void tipo metodas.

- Service (paslaugos pavadinimas). Paslaugos, kurią reikia iškviešti pavadinimas. Paslauga su šiuo pavadinimas turi būti nurodyta Spring konteksto xml faile.
5. Sekos išsišakojimo taškas - šis taškas skirtas iššakoti vieną proceso vykdymo seką į keletą sekų. Kiekvienai iš šio taško išeinančiai proceso vykdymo sekai nurodoma sąlygą, kuri turi būti tenkinama, norint vykdyti proceso pasirinktos šakos kryptimi. Sekos išsišakojimo taškas turi tokias savybes:
 - Constraints (apribojimai). Apribojimai nurodantys sąlygas, kurias reikia tenkinti vykdant procesą išsišakojimo kryptimis.
 - Type (tipas). Sekos išsišakojimo taško tipas. Galimos XOR ir AND reikšmės. XOR atveju procesas vykdomas tik vieną atšakos kryptimi. Tuo tarpu AND atveju procesas gali būti vykdomas keliomis šakomis. AND tipa reikalinga naudoti kai reikalingas lygiagretus keletu proceso vykdymo šakų vykdymas.
 6. Informavimo apie klaidą veiksmas (Error Message Task). Tai veiksmas, kuris yra skirtas informuoti naudotoją apie proceso metu įvykusią klaidą, kurią dažniausiai sąlygoja neteisingai arba nepilnai įvestį duomenis. Klaidos pranešimas pateikiamas naudotojui grafinėje naudotojo sąsajoje, kuri yra realizuota tinklalapio pavidalu. Informavimo apie klaidą veiksmas turi tokias savybes:
 - Message (pranešimas). Pranešimo tekstas, kuris bus pateikiamas naudotojui naudotojo grafinėje aplinkoje.
 - MessageCode (pranešimo kodas). Pranešimo kodas pagal kurį sistema gali parinkti pranešimo tekstą išverstą kitą kalba.
 - ProcessVariable (proceso kintamasis). Proceso kintamasis į kurį bus įrašytas pranešimo tekstas.
 7. Proceso pabaigos taškas - pažymi proceso pabaigą, kurią pasiekus proceso vykdymas yra nutraukiamas ir procesas yra laikomas užbaigtu. Kiekviename verslo procese privalo būti bent vienas proceso pabaigos taškas.

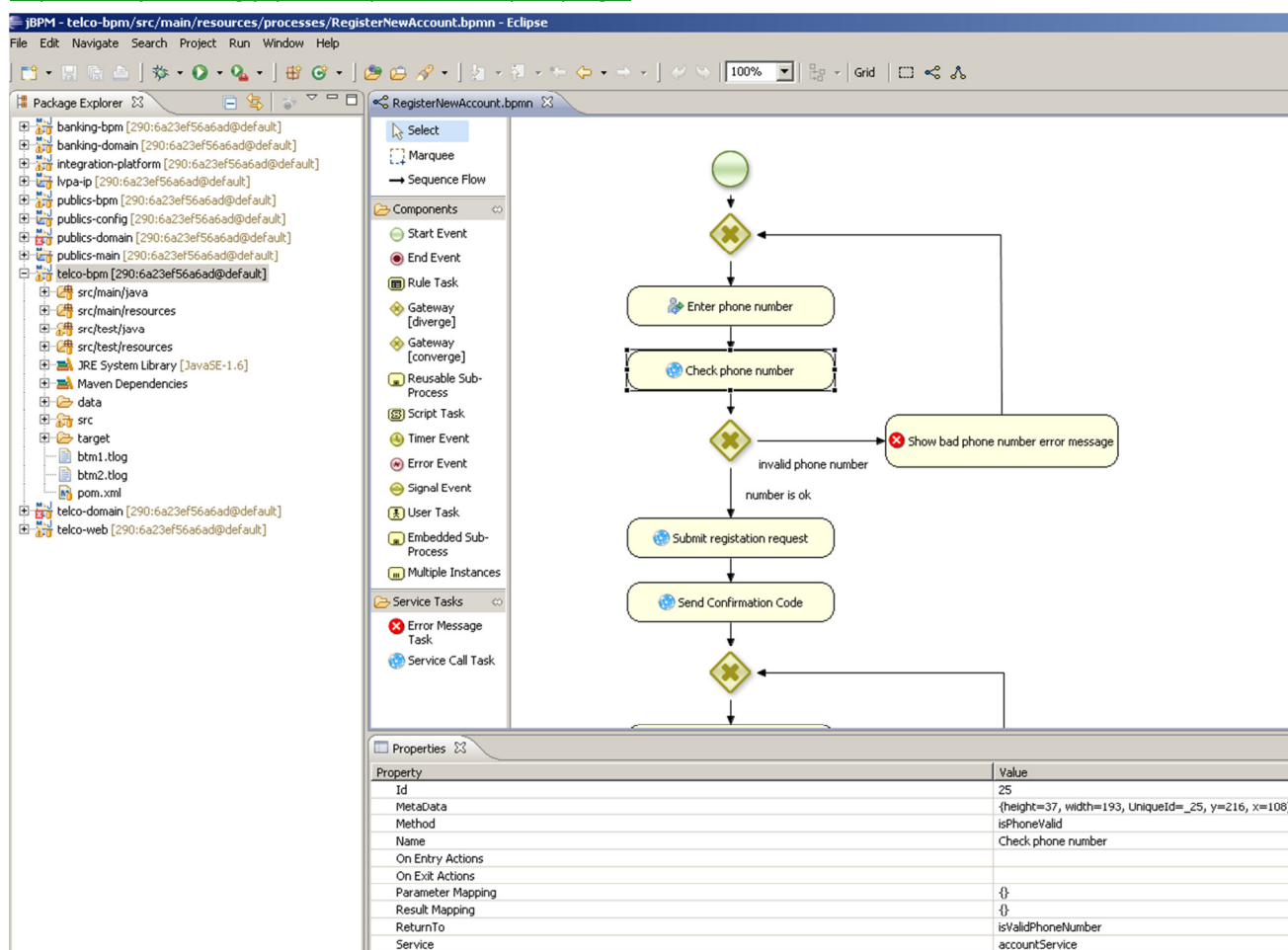


Pav. 2. Verslo proceso specifikacijos pagrindiniai elementai

2.2.2 Verslo procesų modeliavimas

Verslo procesai yra specifikuojami naudojant BPMN 2.0 notaciją. Verslo procesų specifikacijų modeliavimui galima naudoti bet kuri BPMN 2.0 notacija palaikanti modeliavimo įranki. Integracinės platformos verslo procesų modeliavimui yra rekomenduojama naudoti Eclipse BPM Designer įskiepi arba JBoss kompanijos kuriamą jBPM Designer įrankį.

Eclipse BPM Designer įrankis tai įskiepis, kuris gali būti diegiamas į esamą Eclipse aplinką (Pav. 3). Ši įskiepi patogu naudoti programuotojams kuriantiems sistemų integracijas Eclipse programavimo aplinkoje. Eclipse BPM Designer įrankis papildo Eclipse aplinką ne tik grafinio verslo proceso modeliavimo galimybe, bet turi ir papildomus įrankius vykdomo proceso stebėjimui, naudotojo užduočių sąrašo valdymo ir prisijungimo prie verslo taisyklių saugyklos funkcijas. Eclipse BPM Designer įskiepi galima parsisiųsti pasinaudojant šiuo adresu <http://www.jboss.org/jbpm/components/eclipse-plugin>.



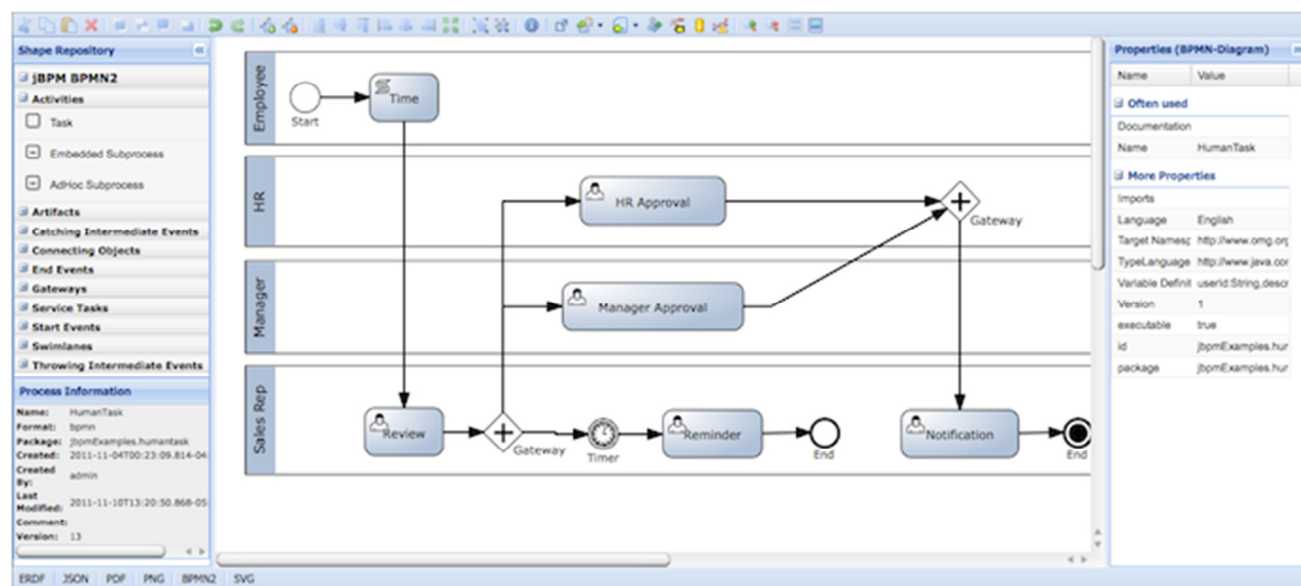
Pav 3. Eclipse BPM Designer įrankis Eclipse programavimo aplinkoje.

Verslo analitikams ir kitiems neturintiems programavimo įgūdžių nariams verslo procesų modeliavimui ir peržiūrai yra rekomenduojama naudoti jBPM Designer įrankį (Pav. 4). Šis įrankis realizuota tinklalapio pagrindu ir nereikalauja jokios papildomos programinės įrangos jo naudojimui. jBPM Designer įrankį galima parsisiųsti pasinaudojant šiuo adresu

<http://sourceforge.net/projects/jbpm/files/jBPM%20Process%20Designer/>. Informacija, apie tai, kaip naudotis

jBPM Designer įrankių yra aprašyta jo vartotojo vadove, kuri galima rasti šiuo adresu

<http://docs.jboss.org/jbpm/v5.2/userguide/ch10.html>



Pav. 4. jBPM Designer įrankis

2.3 Sistemų integravimas

Integravimo platformos sistemų integravimas yra atliekamas kuriant ServiceMix pranešimų magistralės adapterio komponentus. Kiekvienai integruojamos sistemos paslaugai kuriamas atskiras adapterio komponentas. Service Mix Adapterio komponentai yra talpinami į adapters katalogą. Adapterio komponentą turi sudaryti:

- CamelRouteBuilder klasė. Šioje RouteBuilder klasę praplečiančioje klasėje turi būti realizuotas configure() metodas, kuriame turi būti aprašyti pranešimų srautai, jų siuntėjai ir galutiniai gavėjai.

```
public class CamelRouteBuilder extends RouteBuilder {

    private static Logger log = LoggerFactory.getLogger(CamelRouteBuilder.class);

    @Override
    public void configure() throws Exception {
        log.info("AccountServiceAdapter. Configure Camel routes...");

        from("cx:bean:routeEndpoint") .recipientList(simple("direct:${header.operationName}"));

        log.info("AccountServiceAdapter. Configure Camel routes... Done");
    }
}
```

Pavyzdinė CamelRouteBuilder klasės realizacija.

- ServiceWorker klasė. Šioje klasėje turi būti realizuotas tiesioginis duomenų perdavimas iš išorinės sistemos. Šios klasės metodus yra nukreipiami pranešimai iš CamelRouteBuilder klasės.

- features.xml Tai OSGi modulinės technologijos konfigūravimo failas. Šiame faile yra nurodoma kokias bibliotekas naudoja adapterio komponentas darbui su išorinėmis sistemomis, nurodomos tiekiamos ir naudojamos paslaugos. Features.xml failas turi būti patalpintas į adapteriaus komponento projekto src\main\features katalogą.

```

<feature name="{project.artifactId}" version="{project.version}" >
<details>{project.description}</details>
<bundle start-level="90">mvn:{project.groupId}/{project.artifactId}/{project.version}</bundle>
<feature version="3.0.7.RELEASE">spring</feature>
<feature version="2.10.3">camel-core</feature>
<feature version="2.10.3">camel-spring</feature>
<feature version="2.10.3">camel-cxf</feature>
<bundle start-level="80">mvn:com.nortal.lvpa.telco.domain/telco-domain/0.0.1-SNAPSHOT</bundle>
<bundle start-level="85">mvn:com.nortal.lvpa.telco.endpoints.account.service/account-service-
interface/0.0.1-SNAPSHOT</bundle>
<bundle start-level="80">wrap:mvn:com.nortal.lvpa.telco.externals.unified-login/unified-login-
sei/0.0.1-SNAPSHOT</bundle>
</feature>

```

Pavyzdinė feature.xml failo struktūra.

- camel-context.xml. Tai Spring karkaso konteksto konfigūravimo failas. Šiame faile yra nurodoma konfigūracija skirta adapterio komponento inicializavimui Spring priemonėmis. Camel-context.xml failas turi būti patalpintas į adapteriaus komponento projekto src\main\resources\META-INF\spring\ katalogą.

```

<context:component-scan
base-package="com.nortal.lvpa.telco.adapters.account.service.adapter"
annotation-config="true" />

<cxf:cxfEndpoint id="routeEndpoint"
address="http://0.0.0.0:9000/telco/services/accountService"
serviceClass="com.nortal.lvpa.telco.endpoints.account.service.AccountService"
wsdlURL="classpath:wsdl/account-service-interface/account-service-interface.wsdl"
loggingFeatureEnabled="true" />

<jaxws:client id="unifiedLoginClient"
serviceClass="com.nortal.lvpa.externals.unifiedlogin.ws.UnifiedLogin"
address="http://lvpaip:8080/unified-login/services/unifiedlogin" />

<camelContext id="account-service-adapter-context"
xmlns="http://camel.apache.org/schema/spring">
<routeBuilder ref="accountServiceRouteBuilder"/>
</camelContext>

```

```
<bean id="accountServiceRouteBuilder"
class="com.nortal.lvpa.telco.adapters.account.service.adapter.CamelRouteBuilder" />
</beans>
```

Pavyzdinė camel-context.xml failo struktūra.

2.4 Naudotojų rolių valdymas

Verslo procesų specifikuotos užduotys yra priskiriamos tam tikrai naudotojo rolei arba jų grupei. Integracinės platformos naudotojų rolių valdymui yra naudojamas tviro kodo Spring karkaso Security modulis, kurio detalųjį aprašymą galima rasti šiuo adresu <http://static.springsource.org/spring-security/site/reference.html>. Naudotojų rolės yra konfigūruojamos naudojant security-context.xml failą kuris yra saugomas web/src/main/resources/ kataloge. Šio failo pagalba galima nurodyti paslaugas ir jomis naudotis priskirtus naudotojus, nurodyti puslapį prisijungimo duomenims įvesti. Pavyzdinis security-context.xml failas pateiktas žemiau.

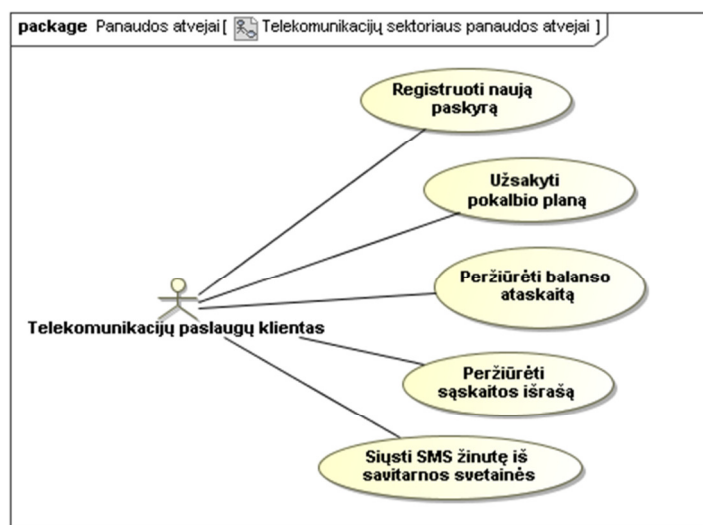
```
<security:user-service id="userDetailsService">
<security:user name="Administrator" password="" authorities="admin" disabled="true"/>
<security:user name="anonymousUser" password="test" authorities="user,admin" />
<security:user name="ruslanas" password="test"
authorities="admin,manager,user,webdesigner,functionalanalyst" />
</security:user-service>

<security:authentication-manager id="customerAuthenticationManager">
<security:authentication-provider user-service-ref="userDetailsService">
</security:authentication-provider>
</security:authentication-manager>

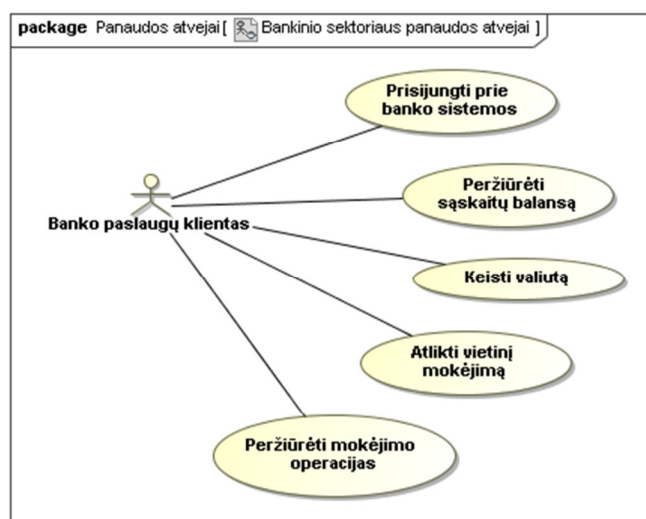
<security:http authentication-manager-ref="customerAuthenticationManager"
pattern="^(?!/images/|/styles/|/css/|/js/|/media/).*" path-type="regex">
<security:form-login password-parameter="password" username-parameter="login" default-target-
url="/user" always-use-default-target="true"
login-processing-url="/j_spring_security_check" login-page="/telco#fragment-login"/>
<security:logout invalidate-session="true" logout-success-url="/telco" logout-
url="/j_spring_customer_logout"/>
<security:anonymous enabled="true"/>
<security:intercept-url pattern="/checkout/*" access="ROLE_CUSTOMER"/>
<security:intercept-url pattern="/*" access="IS_AUTHENTICATED_ANONYMOUSLY" />
</security:http>
```

2.5 Darbo su integracinės platformos ruošiniais metodika

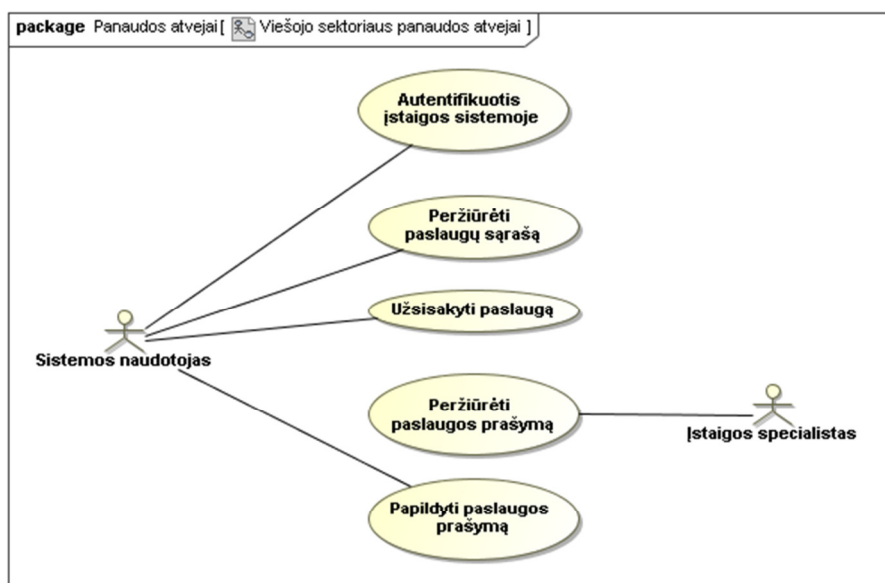
Dabartinės integracinės platformos realizacijoje yra paruošti naudoti 15 integracijų ruošinių viešajam, telekomunikacijų ir bankiniam sektoriui. Kiekvieno sektoriaus ruošiniai yra saugomi atitinkamuose maven moduluose (telco-main, banking-main, publics-main). Kiekvieną ruošinį sudaro proceso specifikaciją (bpm/processes katalogs), reikalingų paslaugų sąsajos (endpoints katalogas) ir sąsajos realizuojantys ServiceMix adapterių komponentai (adapters katalogas). Kiekvienas ruošinys realizuoja tam tikrą panaudos atvejį iš bankinio, telekomunikacijų ir viešojo sektoriaus. Realizuotų panaudos atvejų ruošiniai pateikti žemiau.



Pav. 5. Telekomunikacijų sektoriaus panaudos atvejai.



Pav. 6. Bankinio sektoriaus panaudos atvejai



Pav. 7. Viešojo sektoriaus panaudos atvejai

Norint pritaikyti esamus ruošinius specifiniam panaudos atvejui realizuoti reikalinga atlikti šiuos veiksmus:

1. **Nustatyti integravimo uždavinio aktorius.** Reikalinga nustatyti kokie aktoriai naudosis integravimo sprendimu. Aktoriai gali būti tiek žmonės tiek išorinės sistemos. Aktoriui nustatymui rekomenduojama naudoti panaudos atvejų aktorių nustatymo metodiką.
2. **Pasirinkti labiausiai tinkamą ruošinių sektorių duotam integravimo uždaviniui spręsti.** Jeigu integravimo uždavinys yra susijęs su finansinėmis sistemomis, rekomenduojama rinktis ruošinius iš bankinio sektoriaus. Jeigu integravimo uždavinyje yra tiekiamos informacinio pobūdžio paslaugos ir atsiskaitoma už jas, rekomenduojama rinktis telekomunikacijų sektorių. Jeigu integravimo uždavinyje yra keletas aktorių, kurie keičia artefakto būseną (vienas sukuria, kitas patvirtina) rekomenduojama rinktis ruošinius iš viešojo sektoriaus.
3. **Identifikuoti integravimo uždavinio panaudos atvejus.** Reikalinga identifikuoti ne tik integruojamas sistemas bet ir panaudos atvejus t.y. galutinę vertę naudotojui. Panaudos atvejų nustatymui rekomenduojama naudoti panaudos atvejų nustatymo metodiką.
4. **Parinkti labiausiai tinkamus ruošinius panaudos atvejams realizuoti.** Teisingai parinkti integracinės platformos ruošiniai gali būti lengvai pritaikyti atitinkamiems uždaviniams spręsti. Jeigu uždavinyje yra tiesiog surenkama informacija iš daugelio šaltinių ir tiesiog pateikiama naudotojui labiau tinka ruošiniai "Peržiūrėti balansą", "Peržiūrėti paslaugų sąrašą" ir pan. Jeigu integravimo užduotyje naudotojui reikalinga atlikti keletą veiksmų keliuose žingsniuose, labiau tinka tokie ruošiniai kaip "Registruoti naują paskyrą", "Užsakyti pokalbio planą" ir kiti.
5. **Pritaikyti ruošinio proceso specifikaciją integravimo panaudos atvejui realizuoti.**
 - a. Padaryti ruošinio proceso specifikacijos failo kopiją "processes" kataloge.
 - b. Atverti ruošinio proceso specifikaciją verslo procesų modeliavimo įrankio aplinkoje. Verslo proceso specifikacijos keitimui tinka bet kuris BPMN 2.0 notaciją palaikantis modeliavimo įrankis, tačiau yra rekomenduojama naudoti modeliavimo įrankius išvardintus Verslo procesų specifikavimas dalyje.

- c. Pakeisti ruošinyje proceso pavadinimą, pervadinti proceso specifikacijos failo pavadinimą ir pakeisti proceso identifikatorių (Id savybė).
 - d. Pašalinti nereikalingus proceso veiksmus ir taškus. Reikia prisiminti, kad kiekvienoje verslo proceso specifikacijoje turi būti bent vienas pradžios ir pabaigos taškas.
 - e. Pašalinti nereikalingus proceso kintamuosius.
 - f. Papildyti proceso specifikaciją reikiamais veiksmais, taškais ir proceso kintamaisiais.
6. **Sukurti tinklalapio naudotojo sąsają naudotojo užduoties veiksmams atlikti.** Pavyzdinius ruošinių tinklalapių puslapius galima rasti web/src/main/webapp/WEB-INF/jsp kataloge.
- a. Realizuoti Controller.java Spring MVC kontrolėrį kuriame susieti procesų identifikatorius su nuorodomis tinklalapio puslapiuose.
 - b. Užregistruoti į proceso vykdymo veiksmus reaguojančia TaskEventListener klasės realizaciją naudojant JBPM Server klasės addEventListener metodą. Pavyzdinė TaskEventListener sąsajos realizacija pateikta žemiau.

```
server.addEventListener(new DefaultTaskEventListener() {
@Override
public void taskCompleted(final TaskUserEvent event) {
logger.info("completing a task : " + event.toString());
super.taskCompleted(event);
new Thread() {
@Override
public void run() {
String taskName = server.getTaskName(event.getTaskId());
String userId = event.getUserId();
GenericWebResponse gwr = createGenericWebResponse();
}
}.start();
}
}
```

- c. Realizuoti tinklalapio puslapius JSP priemonėmis kiekvienam naudotojo užduoties veiksmui. Naudotojo užduoties veiksmo puslapiai turi gauti užduoties duomenis iš nurodytų parametrų verslo proceso naudotojo užduotyje (Parameter Mapping savybė). Atlikus naudotojo užduotį reikalinga iškviešti JBPM Server klasės completeTask metodą, kuriam reikalinga paduoti duomenų rinkinį su kintamaisiais aprašytais proceso naudotojo užduoties rezultatu (Result Mapping) savybėje.
 - d. Susieti TaskEventListener klasėje naudotojo užduoties veiksmų identifikatorius (TaskName savybė) su jiems sukurtiems JSP puslapių veiksmais.
7. **Sukurti paslaugų sąsajas ir jų realizacijas.** Pavyzdinės ruošiniuose naudojamos paslaugų sąsajos ir jų realizacijos yra patalpintos endpoints kataloge.
8. **Užregistruoti paslaugų sąsajų realizacijas Spring karkaso jbpmserver-context.xml faile.** Paslaugų sąsajų registravimo faile jų identifikatoriai turi sutapti su identifikatoriais naudojamais verslo proceso Paslaugos kvietimo veiksmų "Service" savybėje.
9. **Supakuoti ir įdiegti tinklalapio aplikaciją į aplikacijų serverį.** Verslo procesų specifikacijos ir tinklalapio puslapiai supakuojami į viena diegimui paruoštą paketą maven komanda "mvn package".

Supakuota aplikaciją galima diegti į bet kuri Java Servlet konteinerio savybes realizuojantį aplikacijos serverį. Aplikacijos diegimą galima atlikti naudojant maven komanda "mvn deploy".