

**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA**

Mykolas Medžionis

**Paslaugomis pagrįsto verslo proceso modeliavimas**

Magistro darbas

Darbo vadovas  
doc. dr. B. Paradauskas

**Kaunas 2008**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA**

Mykolas Medžionis

**Paslaugomis pagrįsto verslo proceso modeliavimas**

Magistro darbas

Recenzentas

doc. dr. A. Lenkevičius  
2008-01-

Vadovas

doc. dr. B. Paradauskas  
2008-01-

Atliko

IFM-2/4 gr. stud.  
Mykolas Medžionis  
2008-01-10

**Kaunas 2008**

## **Summary**

### **Service-Oriented Business Process Modeling**

Since business-to-business solutions are rapidly gaining popularity, there is a need for new measures, which would allow more robust development of such applications than existing UML-based CASE tools. To define long-running, service-oriented business process fully and correctly, it is necessary to specify following properties: data flows, business partners along with their actions, and state dynamics of entire process. Also, it is required to define mechanics which can be used to compensate failed long-running transactions. This can be achieved by combining SAGA and Communicative Action Loop models. To gain robustness, it was suggested to use two types of strictly formatted diagrams. A formal metamodel was proposed as a basis for a graphical editor which can be used to define a service-oriented business process and its properties. Finally, a prototype of graphical editor was developed and used to specify example business processes.

# TURINYS

<b>Lentelių sąrašas.....</b>	<b>6</b>
<b>Paveikslų sąrašas.....</b>	<b>7</b>
<b>Įvadas .....</b>	<b>8</b>
<b>1. Verslo procesų modeliavimas.....</b>	<b>9</b>
1.1. Tyrimo sritis ir sprendžiamos problemos .....	9
1.2. Tyrimo objektai .....	9
1.3. Sistemos vartotojai.....	11
1.4. Egzistuojantys problemos sprendimai.....	11
1.4.1. Projektas EFFICIENT .....	11
1.4.1.1. Sprendžiamos problemos.....	11
1.4.1.2. Apžvalga.....	11
1.4.2. Verslo bendradarbiavimo modelių registravimas ebXML saugykloje.....	14
1.4.2.1. Sprendžiamos problemos .....	14
1.4.2.2. Apžvalga.....	14
1.4.3. SAGA modelis.....	16
1.4.3.1. Sprendžiamos problemos.....	16
1.4.3.2. Apžvalga.....	16
1.4.4. Komunikacinių kilpų modelis.....	18
1.4.4.1. Sprendžiamos problemos.....	18
1.4.4.2. Apžvalga.....	18
1.5. Egzistuojanti programinė įranga .....	20
1.5.1. MagicDraw UML ir EFFICIENT.....	20
1.5.2. Grafinis BPEL dokumentų redaktorius.....	22
1.6. Darbo tikslas ir siekiami privalumai .....	24
<b>2. Paslaugomis pagrįsto verslo proceso modelis.....</b>	<b>25</b>
2.1. Pagrindiniai principai.....	25
2.2. Modelio elementai .....	25
2.2.1. Etapas .....	26
2.2.2. Aktorius.....	26
2.2.3. Veiksmas .....	27
2.2.4. Būsena .....	27
2.2.5. Duomenų srautas.....	29
2.2.6. Procesas .....	29
2.3. Priklausomybės.....	30
2.4. Grafinė notacija .....	31
2.4.1. Verslo proceso diagrama .....	31
2.4.2. Proceso etapo diagrama.....	33
2.5. Apibendrinimas .....	34
<b>3. Verslo procesų redaktorius.....</b>	<b>35</b>
3.1. Reikalavimai.....	35
3.1.1. Siekiamos sistemos apibrėžimas .....	35
3.1.2. Kompiuterizuojamos funkcijos .....	36
3.1.3. Reikalavimai duomenims .....	37
3.1.4. Nefunkciniai reikalavimai ir apribojimai.....	38
3.1.4.1. Reikalavimai sąsajai.....	38
3.1.4.2. Reikalavimai kokybei.....	38
3.1.5. Rizikos veiksniai .....	39
3.1.6. Rezultato kokybės kriterijai .....	39
3.2. Grafinio redatoriaus projektas .....	40
3.2.1. Architektūra ir realizavimo priemonės .....	40
3.2.2. Statinė struktūra .....	41
3.2.2.1. Loginė architektūra .....	41
3.2.2.2. Modelio elementus realizuojančios klasės .....	42
3.2.2.3. MVC architektūra.....	43
3.2.3. Realizacijos modeliai .....	44
3.3. Darbo redaktoriumi principai .....	46
3.3.1. Darbo aplinka.....	46
3.3.2. Redagavimo priemonės .....	47

<b>4. Eksperimentinis verslo proceso modelis.....</b>	<b>50</b>
4.1. Verslo procesas.....	50
4.2. Verslo proceso etapų specifikavimas .....	52
<b>5. Išvados.....</b>	<b>55</b>
<b>Literatūra.....</b>	<b>57</b>
<b>Terminai ir santrumpos .....</b>	<b>59</b>
<b>1 Priedas. Verslo proceso modelis.....</b>	<b>60</b>
<b>2 Priedas. Straipsnis. Ilgalaikių verslo B2B transakcijų modeliavimas .....</b>	<b>65</b>

## Lentelių sąrašas

1 lentelė.	Verslo proceso modelio elementai .....	26
2 lentelė.	Panaudojimo atvejo „Sukurti modelį“ aprašas .....	36
3 lentelė.	Panaudojimo atvejo „Redaguoti modelį“ aprašas .....	36
4 lentelė.	Panaudojimo atvejo „Išsaugoti modelį“ aprašas .....	37
5 lentelė.	Panaudojimo atvejo „Užkrauti modelį“ aprašas.....	37
6 lentelė.	Reikalavimai vartotojo sąsajai.....	38
7 lentelė.	Reikalavimai kokybei .....	38
8 lentelė.	Projekto rizikos faktoriai.....	39
9 lentelė.	Programinės įrangos kokybės kriterijai .....	39
10 lentelė.	Paketuose realizuojamas funkcijos .....	41
11 lentelė.	Modelį realizuojančių klasių paskirtis .....	42
12 lentelė.	Pagrindinių MVC architektūros klasių paskirtis .....	43

## Paveikslų sąrašas

1 pav.	Įprastinės grafinio redaktoriaus funkcijos.....	10
2 pav.	Verslo transakciją specifikuojanti veiklos diagrama.....	12
3 pav.	Verslo pranešimą specifikuojanti klasių diagrama.....	13
4 pav.	UMM verslo transakcijos modelis .....	15
5 pav.	SAGA modelis .....	16
6 pav.	Komunikacinė veiksmų kilpa.....	18
7 pav.	MagicDraw aplinkoje su EFFICIENT plėtiniais sukurta veiklos diagrama .....	20
8 pav.	Pranešimą specifikuojanti klasių diagrama.....	21
9 pav.	Grafinis BPEL dokumentų redaktorius .....	22
10 pav.	Verslo proceso modelį sudarančių elementų priklausomybės .....	25
11 pav.	Modelio elementų priklausomybės.....	30
12 pav.	Verslo procesą specifikuojanti diagrama.....	32
13 pav.	Verslo proceso etapą specifikuojanti diagrama.....	33
14 pav.	Sistemos aplinka.....	35
15 pav.	Kompiuterizuojamos funkcijos .....	36
16 pav.	Redaktoriaus loginė architektūra.....	41
17 pav.	Modelio elementus realizuojančių klasių diagrama .....	42
18 pav.	MVC architektūros realizacija .....	43
19 pav.	Grafinio redaktoriaus komponentų modelis.....	44
20 pav.	Grafinio redaktoriaus diegimo modelis .....	45
21 pav.	Verslo procesų redaktorius „Eclipse“ aplinkoje.....	46
22 pav.	Verslo proceso modelio realizacija .....	47
23 pav.	Proceso etapo specifikacija .....	48
24 pav.	Būsenos požymių redagavimo langas.....	49
25 pav.	Paslaugos užsakymo modelis .....	51
26 pav.	Verslo proceso etapo specifikacija .....	52
27 pav.	Verslo proceso etapo, kurį inicijuoja vykdytojas, specifikacija.....	53
28 pav.	Etapo kompensaciją specifikuojanti diagrama.....	54

## **Ivadas**

Elektroninės verslo transakcijos vis dažniau keičia tradicinius verslo procesus. Tokie pokyčiai vyksta siekiant pasiūlyti geresnes ir patogesnes paslaugas klientams ir verslo partneriams. Siekiama mažinti procesų kaštus, juos darant efektyvesnius ir mažiau imlius daug laiko reikalaujančiam žmogaus darbui.

Pirmieji duomenų mainai tarp skirtingų verslo partnerių taikomųjų programų pradėti vykdyti 1960 – 1970 m. Nuo to laiko svarbiausi pokyčiai elektroninio verslo sferoje įvyko tik atsiradus XML technologijoms. XML sudarė geras galimybes integruoti duomenis.

Tačiau paaiškėjo, kad duomenų integravimas nesukuria tokios naudos, kokios buvo tikėtasi. Atsirado naujas požiūris į integravimą – pasiūlyta integruoti ne tik duomenis, bet ir verslo procesus. Tuo pačiu metu suprasta, jog būtina kurti atvirus, visiems prieinamus metodus, kurie leistų prie integravimo proceso prisijungti visiems to norintiems.



## **1. Verslo procesų modeliavimas**

Šios darbo tikslas – nubrėžti kuriamos programinės įrangos gaires.

Tam reikia atlikti dalykinės srities – verslo procesų modeliavimo – tendencijų tyrimą ir apžvelgti literatūros šaltiniuose siūlomas metodologijas bei su jomis siejamus PĮ sprendimus.

Dalykinės srities ir egzistuojančių sprendimų analizės rezultatų pagrindu bus galima pasiūlyti konceptualų sprendimą, o vartotojų keliami nefunkciniai reikalavimai sudarys prielaidas atlikti galimos architektūros analizę, įvertinti galimas kūrimo platformas bei konkrečias priemones ir parinkti tinkamiausius realizavimo įrankius.

### **1.1. Tyrimo sritis ir sprendžiamos problemos**

Tyrimo sritis – paslaugomis pagrįstų verslo procesų ir su jais susietų ilgalaikių verslo transakcijų, kurioms negali būti taikomi atominio valdymo principai, modeliavimas.

Šiuo metu rinkoje yra daug projektavimo įrankių – nuo paprastų schemų redaktorių iki galingų integruotų paketų, tačiau trūksta specializuotų priemonių, skirtų greitai ir suprantamai modeliuoti verslo procesus. Tai sudaro prielaidas šio darbo tikslui – pasiūlyti konceptualų sprendimą ir jo pagrindu suprojektuoti grafinį verslo procesų redaktorių bei realizuoti jo prototipą.

### **1.2. Tyrimo objektai**

Verslo transakcija (B2B transakcija) apibrėžiama kaip verslo proceso dalis, vykdoma dviejų arba daugiau partnerių ir sugeneruojanti suskaičiuojamą galinę būseną (sėkmė arba nesėkmė) [15]. Paskirstytiems verslo procesams būdingos ilgos trukmės (lyginant jas su duomenų bazių transakcijų trukmėmis), jie realizuojami, įvykdant daugelį duomenų bazių transakcijų ir kaupiant duomenų srautus paskirstytosiose DB.

Vykdam B2B verslo transakcijas kaip duomenų bazių transakcijų seką, nuosekliai kinta verslo proceso ir jo dalyvių būsenos. Esant tam tikroms aplinkybėms, pvz., nutrūkus transakcijos ryšiui arba verslo aktoriui nusprendus atšaukti savo veiksmus, reikalingas mechanizmas kompensuoti ilgalaikį verslo procesą ir su juo susietus pakeitimus duomenų bazėse. Būtinai sėkmingo kompensavimo požymis – verslo partnerių būsenos sutampa su prieš kompensuojamą B2B transakciją buvusiomis būsenomis arba yra joms ekvivalenčios.

Esminė išskylanti problema – verslo transakcijos yra sudėtinės, o jų trukmės yra ilgos, todėl taikyti tradicinį atominį transakcijų valdymą yra neefektyvu dėl galimo ilgalaikio resursų blokavimo [14]. Sprendžiant šią problemą būtina formaliai specifikuoti su verslo procesais susietas

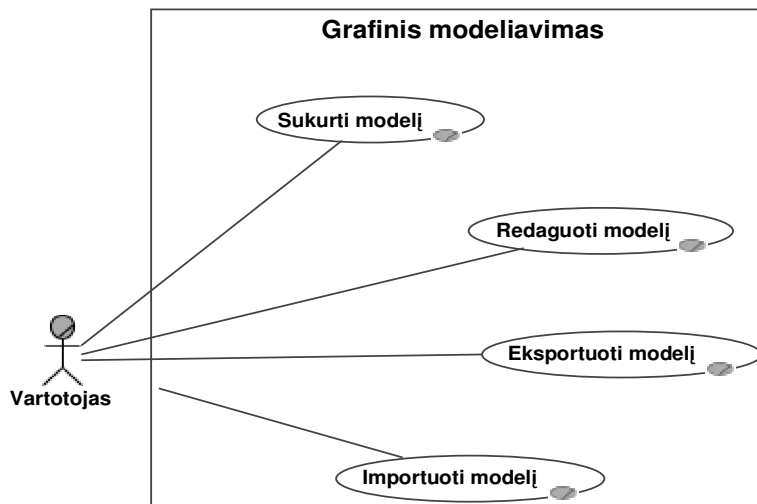
ilgalaikes transakcijas ir jų valdymą. Nors šaltiniuose siūloma verslo procesus ir transakcijas specifikuoti įprastinėmis priemonėmis (UML, XML) [7, 8, 13], tiesiogiai pritaikant šias priemones išskyla sunkumų formaliai aprašant taisykles ir apribojimus, todėl tradicinių CASE įrankių panaudojimas yra ribotas [7].

Esant tendencijoms verslo paslaugas perkelti į žiniatinklį, standartizavimo ir integravimo problemos sprendžiamos specializuotomis technologijomis (WSDL, SOAP, BPEL). Tačiau verslo procesų metu įvykdytų pasikeitimų monitoringas ir kompensavimas nėra detalai specifikuojami ir yra aktualūs, nes, vykdant bendradarbiavimą nehomogenine ir nepatikima aplinka, būtina įvertinti galimas kritines situacijas ir jų sukeltų padarinių kompensavimo galimybes.

Modeliuojant verslo procesus ir transakcijas, nepriklausomai nuo metodikos, būtina, bet dažnai nepakankama, specifikuoti šiuos objektus:

1. aktorius;
2. aktorių veiksmus;
3. duomenų srautus.

Tipinės funkcijos, kurių vartotojas gali reikalauti dirbdamas grafiniais redaktoriais yra pateiktos 1 pav.



**1 pav.** Įprastinės grafinio redaktoriaus funkcijos

### 1.3. Sistemos vartotojai

Tikėtini kuriamo įrankio vartotojai – sistemų architektai, analitikai ir verslo ekspertai, projektuojantys naujas arba pertvarkantys palikuonines informacines sistemas taip, kad jos veiktų šiuolaikinėje, nehomogeniškoje elektroninių paslaugų erdvėje.

CASE priemonės vartotojai turės suprasti verslo procesų valdymo principus ir būti gerai susipažinę su kuriamos ar atnaujinamos sistemos teikiamomis ir projektuojamomis paslaugomis.

### 1.4. Egzistuojantys problemos sprendimai

#### 1.4.1. Projektas EFFICIENT

Pierre Brimont, Eric Dubois, Rik Eshuis, Michael Schmitt ir kt. Luxembourg, 2003 – 2006.

##### 1.4.1.1. Sprendžiamos problemos

B2B transakcijų modeliavimo ir validavimo metodologijos tyrimas. Įrankiai ebXML transakcijoms modeliuoti ir validuoti.

##### 1.4.1.2. Apžvalga

Šaltiniuose išskiriami bent trys verslo proceso analizės sluoksniai [3, 7]:

1. Verslo lygmuo (angl. *business layer*).
2. Specifikacijų lygmuo (angl. *specification layer*)
3. Techninis lygmuo (angl. *technical layer*) – šiame darbe neapžvelgiamas, nes jame detalizuojamas transakcijų modelių validavimas panaudojant projekto EFFICIENT metu sukurtą įrankį.

Verslo lygmuo yra skirtas bendrai verslo transakcijų apžvalgai. Verslo proceso, kuriame dalyvauja transakcija, struktūrą nusako panaudos atvejų diagrama kartu su panaudos atvejų scenarijais, o globali klasių diagrama specifikuoja verslo procesų apdorojamą informaciją [3]. Pažymėtina, jog šiame analizės lygmenyje nėra pranešimo sąvokos [8].

Specifikacijų lygmenyje verslo transakcijos yra specifikuojamos UML veiklos diagramomis (angl. *activity diagram*), o pranešimai yra aprašomi atskiromis klasių diagramomis. Kiekviena iš šių klasių diagramų yra atskiras globalios klasių diagramos atvaizdas. Be to, pranešimai gali būti susieti su verslo taisyklių rinkiniu, aprašytu formalia arba neformalia kalba [3].

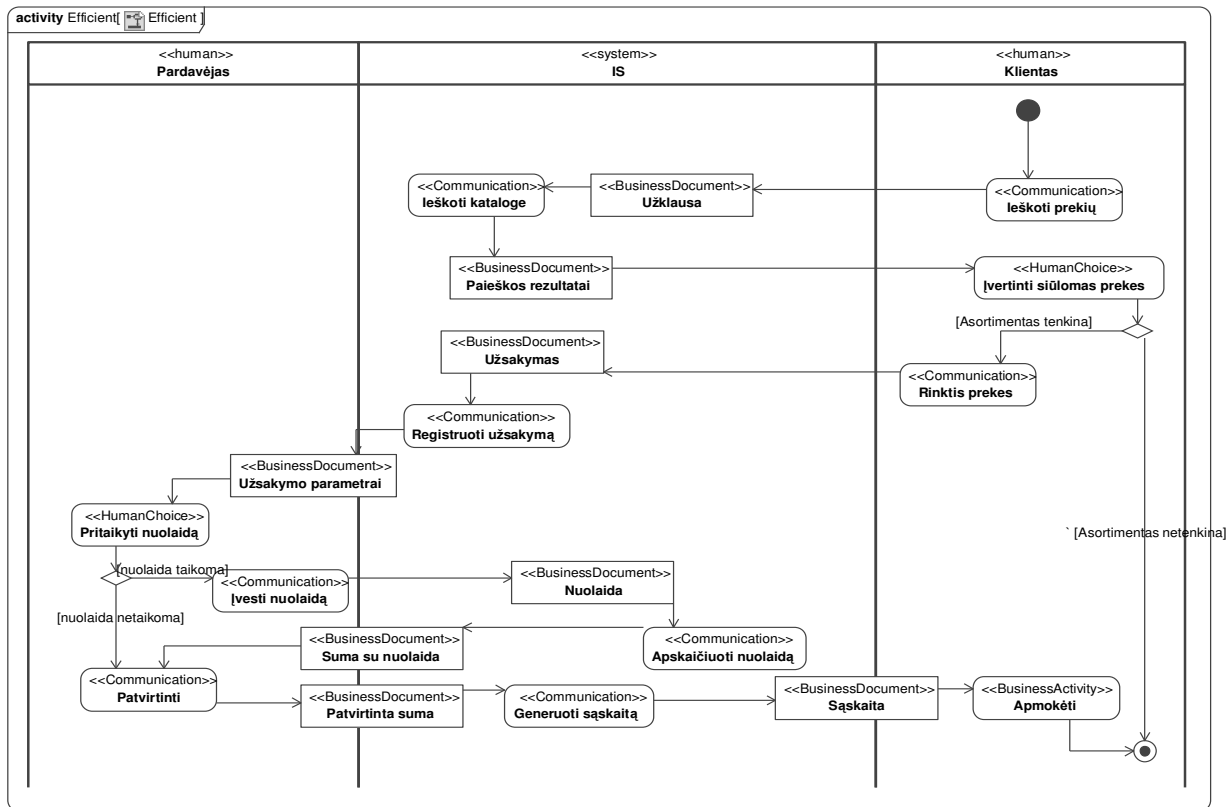
Šaltinyje [7] veiklos diagramos papildomos naujais stereotipais:

<<BusinessDocument>> – atskiras verslo pranešimas;

<<Communication>> – veikla, kurios rezultatas yra vienas pranešimas;

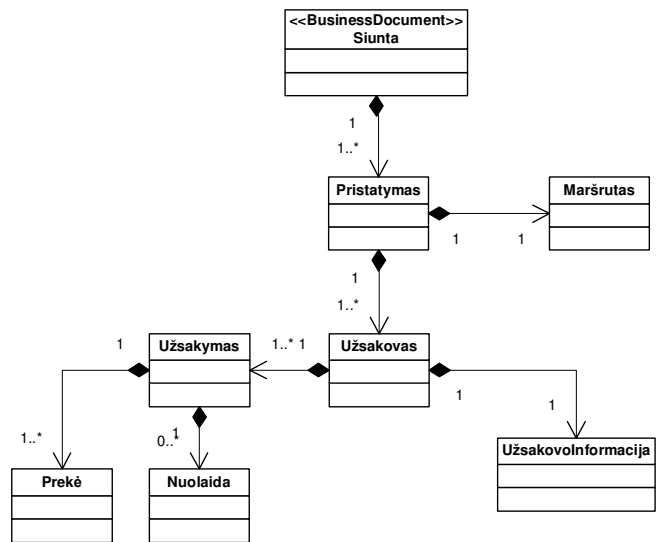
<<HumanChoice>> – veikla, kurios metu vartotojas turi įvertinti gautą pranešimą;

<<BusinessActivity>> – kita veikla.



2 pav. Verslo transakciją specifikuojanti veiklos diagrama

Kiekvienas pranešimas (<<BusinessDocument>> veiklos diagramoje) yra aprašomas atskira klasių diagrama (3 pav.). Straipsnyje pateikiamas glaustas šių klasių diagramų išsaugojimo XML schemomis algoritmo aprašas. Tame pačiame šaltinyje aprašomas metodas transakcijas, specifikuojančias veiklos diagramas išversti į XML procesų aprašymo kalbą bei pateikiami svarbiausi sąryšiai tarp veiklos diagramų ir XPDL (angl. *XML Process Definition Language*) kalbos elementų.



**3 pav.** Verslo pranešimą specifikuojanti klasių diagrama

## **1.4.2. Verslo bendradarbiavimo modelių registravimas ebXML saugykloje**

Birgit Hofreiter, Christian Huemer, Marco Zapletal. Vienna, 2006.

### **1.4.2.1. Sprendžiamos problemos**

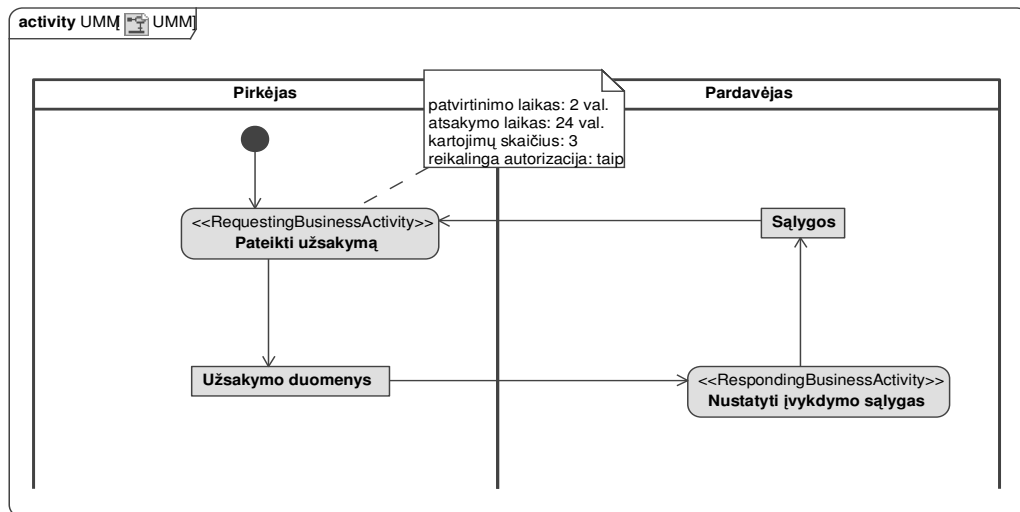
UMM (UN/CEFACT modeliavimo metodologijos) bendradarbiavimo modelio konvertavimas į ebXML saugyklos informacijos modelį (RIM) ir registravimas ebXML saugykloje.

### **1.4.2.2. Apžvalga**

Šaltinyje [11] skiriami du galimi bendradarbiavimo taisyklių tarp organizacijų nustatymo atvejai:

1. Vienas iš verslo partnerių nustato veiklos scenarijų (angl. choreography), o visi kiti norintys bendradarbiauti privalo be išlygų priimti siūlomas roles ir atitinkamai pritaikyti savo sąsajas.
2. Taikomi visuotinai priimtini (angl. well accepted) veiklos scenarijai. Šie scenarijai gali būti apibrėžti standartus reglamentuojančių organizacijų, pramonės konsorciūmų, rinkos lyderių ir pan. Organizacija prisiima standartines roles ir realizuoja reikalingas sąsajas. Tokiu atveju partnerių paieškos procesas supaprastinamas iki organizacijų, kurios atlieka reikalingas standartines roles, paieškos.

Straipsnyje nagrinėjamas antrasis atvejis, didžiausią dėmesį skiriant veiklos scenarijų registravimui ebXML saugyklose.



**4 pav.** UMM verslo transakcijos modelis

.Darbe detalai aprašomas grafinio modelio (4 pav.) pertvarkymas į kompiuteriniam apdorojimui tinkamą formatą. Kadangi UMM pagrindas yra UML modeliavimo kalba, siūloma UMM diagramas išsaugoti kaip XML metaduomenų apsikeitimo standarto (angl. *XML Metadata Interchange – XMI*) dokumentus.

Vienas svarbiausių straipsnio akcentų yra galimybė UMM verslo transakcijų modelį išsaugoti XMI formatu neatliekant jokių papildomų pertvarkymo veiksmų. Tam iliustruoti pateikiamas 4 pav. esančią diagramą atitinkantis su ebXML RIM suderinamo XMI dokumento pavyzdys ir detalai aprašoma, kaip įvairūs diagramos elementai (veiklos, ryšiai, pranešimai ir kt.) aprašomi XMI dokumente.

### 1.4.3. SAGA modelis

Hector Garcia-Molina, Kenneth Salem, Stanford, 1987-1991

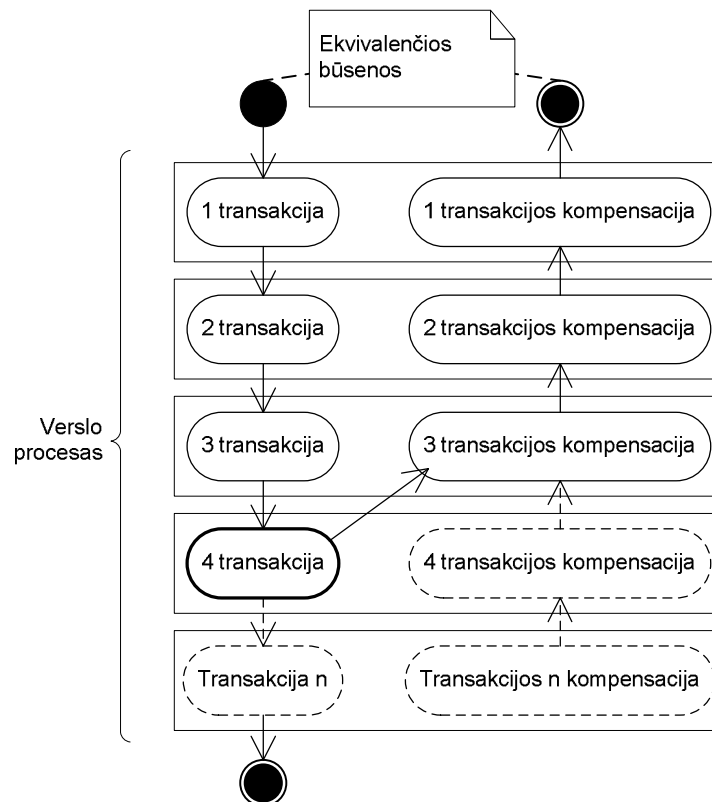
#### 1.4.3.1. Sprendžiamos problemos

Verslo procesų modeliavimas ilgalaikių transakcijų seka. Ilgalaikių transakcijų kompensavimas.

#### 1.4.3.2. Apžvalga

Ilgalaikius verslo procesus SAGA modelyje siūloma skaidyti į trumpų transakcijų, kurioms taikyti atomiškumo principus tampa tikslinga, sekas [9, 12]. Kiekvienam sekos etapui yra sukuriamas kompensacinis veiksmas. Kompensacinių veiksmų sekų įvedimas leidžia:

1. nutraukti verslo procesą, to pageidaujant verslo partneriams;
2. valdyti ilgalaikio B2B proceso etapuose išskylančias kritines situacijas: dėl techninių priežasčių nutraukus duomenų bazės transakciją, galima atšaukti tik paskutinius etapus, kuriuose įvyko klaida ir juos pakartotinai įvykdyti.



5 pav. SAGA modelis



Remiantis SAGA modeliu, kompensuojantys veiksmai atliekami šia tvarka (5 pav.):

1. Nutraukiamas vykdomas verslo proceso etapas (paveiksle – 4 transakcija). Kadangi verslo procesas yra išskaidytas į etapus, kuriems galioja ACID principai, veiksmas atšaukiamas panaudojant DB atstatymo priemones (angl. rollback), todėl nutraukiamai DB transakcijai nėra vykdomas kompensacijos veiksmas.
2. Nuosekliai vykdomi kompensaciniai veiksmai (3 transakcijos kompensacija ÷ 1 transakcijos kompensacija).

Vykdamt kompensaciją duomenų bazės lygmenyje, t.y. tiesiogiai atšaukiant pakeitimus duomenų bazėje ir neatliekant papildomos patikros, iškyła pavojus pažeisti verslo proceso logiką: verslo transakcijos įvykdymas elektroninėje erdvėje gali inicijuoti neatšaukiamus procesus, kurių vykdomi pakeitimai nebūtinai yra apriboti elektrone erdve. (Pvz., išsiunčiamas krovinyš užsakovui.). Atsiranda pavojus kilti situacijoms, kai atstačius duomenų bazės būseną, verslo dalyvio būsena nesutaps su buvusiaja prieš pradedant vykdyti verslo procesą ar proceso etapą. Tokiais atvejais bus pažeidžiamas būtinas korektiško kompensavimo požymis. Analogiškai pagrindiniam verslo procesui, kompensuojantys veiksmai nėra apribojami duomenų bazės atstatymu – tai gali būti naujos transakcijos ar fizinė veikla.

SAGA modelio trūkumas yra modelio bendrumas. Šiuo modeliu neapibrėžiamas duomenų bazių transakcijų vykdyimas ir kompensacinių veiksmų generavimas. Be to, nėra įvertinami atvejai, kai verslo proceso etapas įvykdomas tik iš dalies. Be to, pažeidžiamas izoliacijos reikalavimas, nes tuo metu, kei nevykdomos sekos transakcijos, duomenys nėra apsaugoti, ir juos gali pasiekti bei modifikuoti išoriniai procesai [3].

#### 1.4.4. Komunikacinių kilpų modelis

Remigijus Gustas. Karlstad, 1996-2006

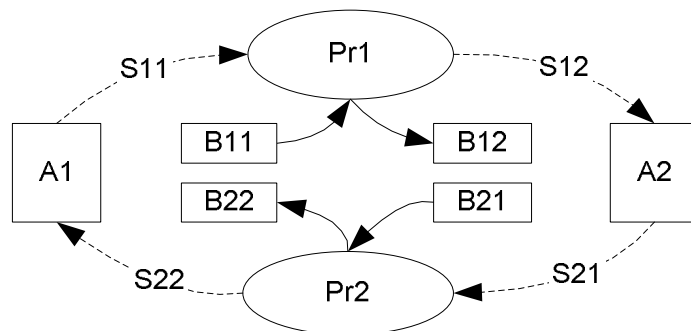
##### 1.4.4.1. Sprendžiamos problemos

Verslo proceso modeliavimas; proceso gyvybingumo pagrindimas procese dalyvaujančių veikėjų tikslais.

##### 1.4.4.2. Apžvalga

Publikacijose [1, 15] pasiūlyta verslo transakcijas modeliuoti komunikacinių veiksmų kilpomis, kurios sudaro organizacijos modeliavimo (angl. *Enterprise Modeling* – EM) metodo pagrindą [9]. EM metodo idėja yra ta, kad organizacijos modeliui specifikuojami ne tik veiklos veiksmai, bet ir pranešimų srautai.

6 pav. pateikta komunikacinė (binarinė) veiksmų kilpa sudaryta iš dviejų aktorių – kliento (A1) ir atlikėjo (A2), pranešimų srautų S11÷S22, ir dviejų veiksmų – kliento veiksmo (Pr1) ir atlikėjo veiksmo (Pr2). Dinaminiai situacijos pasikeitimai prieš ir po veiksmų pateikiami kaip perėjimai nuo būsenų B11, B21 į būsenas B12, B22 ir aibė pragmatinių bei semantinių priklausomybių tarp jų.



6 pav. Komunikacinė veiksmų kilpa

Tam tikros srities versle vykdomos transakcijos yra tarpusavyje susijusios. Kiekviena transakcija vienaip ar kitaip keičia verslo santykių būseną tarp bendradarbiaujančių partnerių. Verslo transakcijų metu bendradarbiaujančios šalys keičiasi verslo informacija (pranešimais), kuri vis papildo vykstantį verslo procesą ir veda jį į užbaigimo būseną, atitinkančią verslo partnerių siekiamus tikslus.

Bendru atveju verslo transakcijos veiksmas turi „prieš“ ir „po“ būseną. Binarinė kilpa susideda iš dviejų veiksmų, kurių kiekvienas turi „prieš“ ir „po“ būsenas (iš viso 4 būsenos).

Pirmojo transakcijos veiksmo „po“ būseną gali sutapti su antrojo veiksmo „prieš“ būseną. Būsenomis galima išreikšti aktorių tikslus ir nustatyti, ar komunikacinė kilpa yra gyvybinga. Pagrindinį gyvybingumo kriterijų galima išreikšti taip: jei aktorius A2 galimybė neigiamai veikia aktorius A1 problemą, tuomet A2 galimybė teigiamai veikia A1 tikslą (patekti į būseną B22) ir A1 tikslas neigiamai veikia A2 problemą (A2 padeda A1 pasiekti tikslą – būseną B22, A1 sumažina arba panaikina A2 problemą).

## 1.5. Egzistuojanti programinė įranga

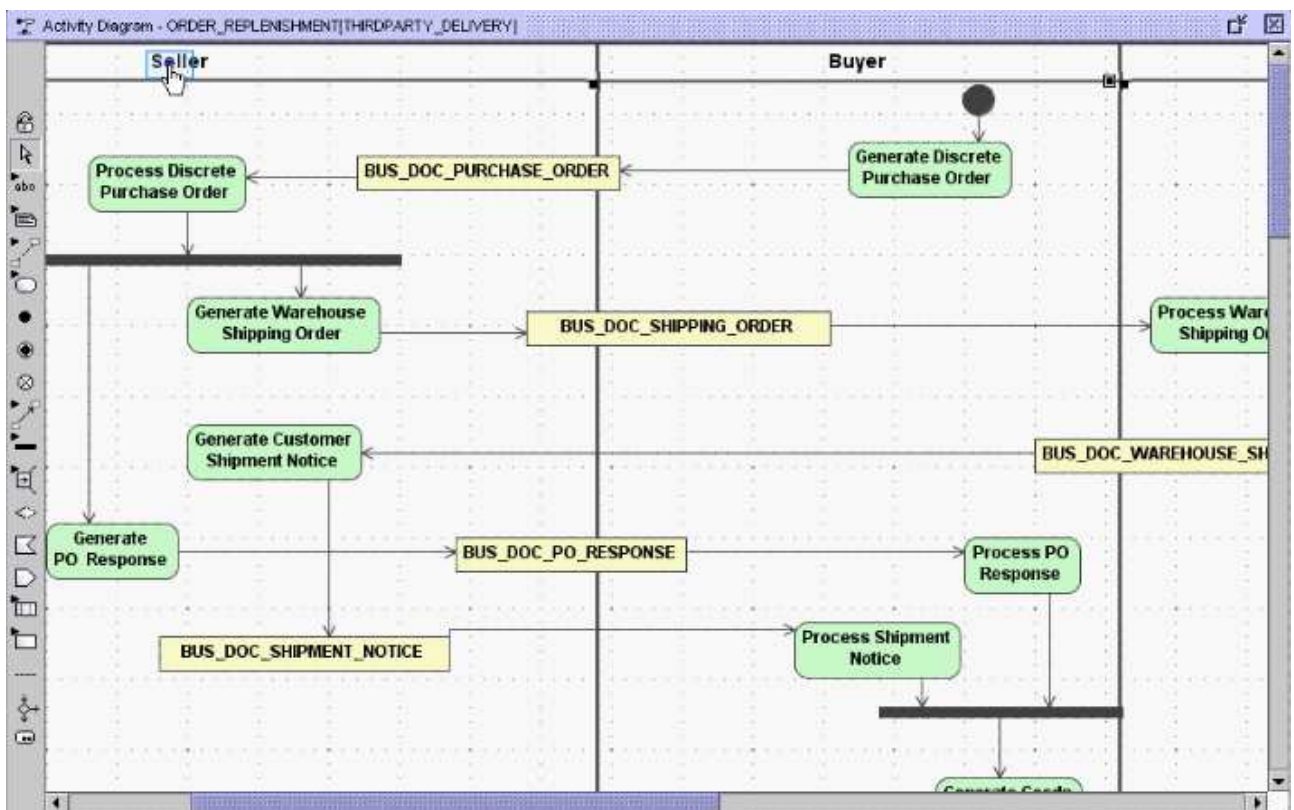
Nors šiuo metu yra daug UML įrankių – nuo paprastų schemų redaktorių iki galingų integruotų paketų, specializuotų priemonių verslo procesams modeliuoti yra mažai.

### 1.5.1. MagicDraw UML ir EFFICIENT

„MagicDraw UML“ yra populiarus, apdovanojimų pelnęs UML redaktorius, kurį sukūrė kompanija *No Magic, Inc.* Redaktoriui skirtas tinklapis yra adresu <http://www.magicdraw.com>.

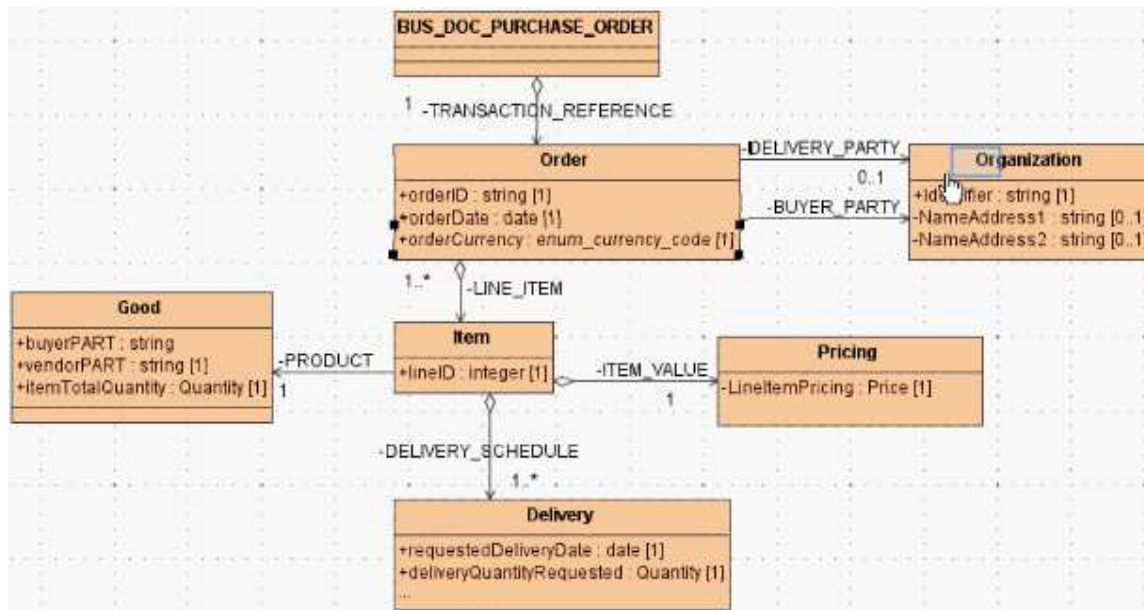
Šiuo paketu galima modeliuoti visas UML 2.0 diagramas. Nors gamintojai nepateikia jokių specifinių priemonių, tačiau trečiosios šalys gali sukurti ir įdiegti savo įskiepius (angl. *plug-ins*).

EFFICIENT projekto metu sukurtas MagicDraw UML įskiepis verslo procesų modeliavimui ir kodo generavimui. 7 pav. yra pagal EFFICIENT metodologiją papildytos veiklos diagramos fragmentas, sukurtas MagicDraw UML paketu.



7 pav. MagicDraw aplinkoje su EFFICIENT plėtiniais sukurta veiklos diagrama

Kiekvienas diagramoje esantis duomenų srautas (veiklos diagramoje žymima stačiakampiu) yra specifikuojamas atskira klasių diagrama (8 pav.)



**8 pav.** Pranešimą specifikuojanti klasių diagrama

Svarbus šios diagramos akcentas – pranešimo struktūra yra medžio formos, todėl sekų diagramoje užtenka nurodyti tik medžio šaknį atitinkančią klasę. Likusios pranešimą sudarančios klasės yra agregacijos ryšiais susietos su šaknimi.

Be grafinės notacijos įrankių, į UML redagavimo paketą įtrauktas modulis generuoti dokumentams: XPDL aprašams iš veiklos diagramų ir XML schemoms iš pranešimus specifikuojančių klasių diagramų.

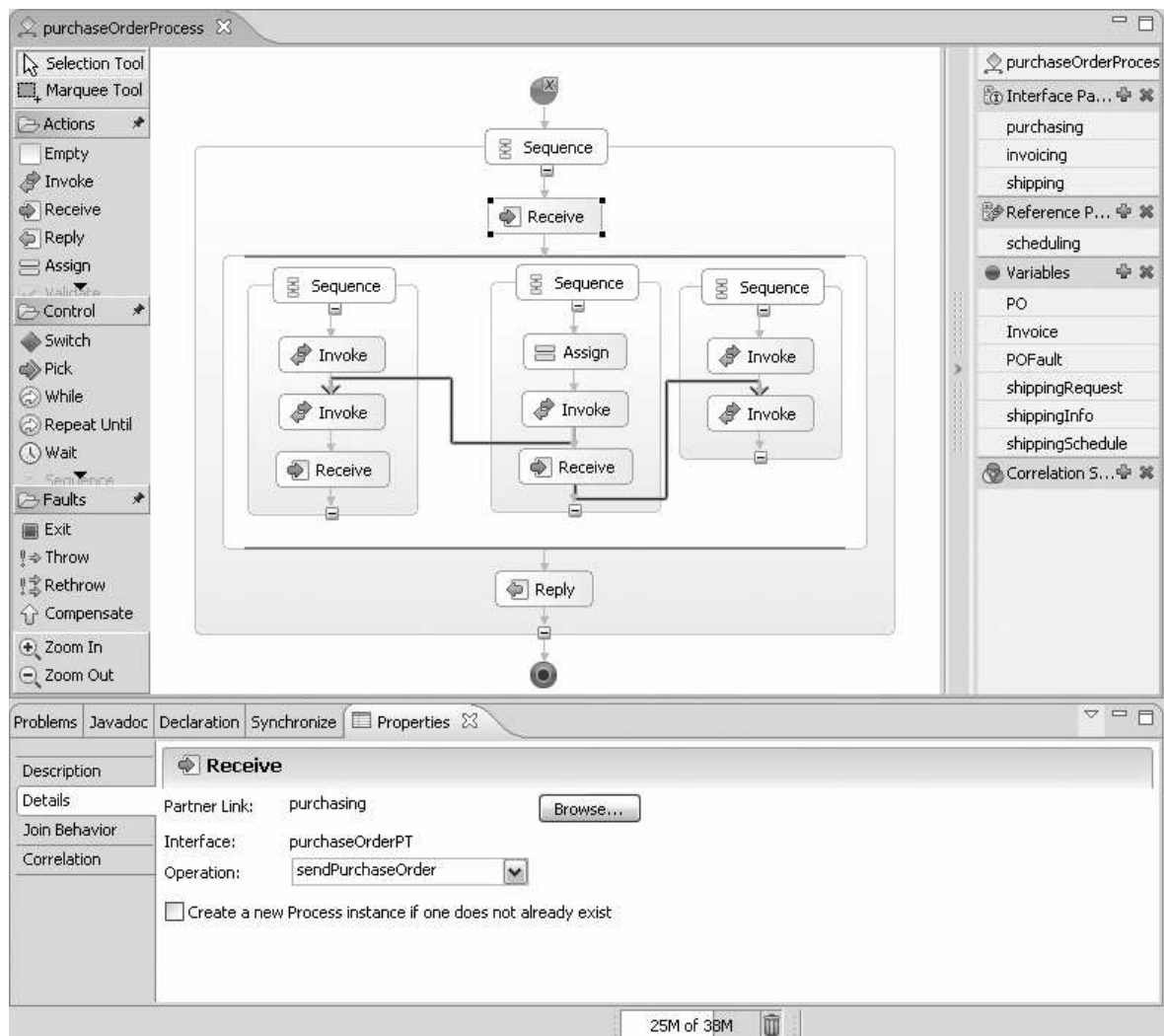
## 1.5.2. Grafinis BPEL dokumentų redaktorius

„Eclipse SDK“ – atvira platforma, sudaryta iš „Eclipse Europa“ redaktoriaus, redaktoriui skirtų išskiepių ir Java bibliotekų.

Vienas tokių išskiepių yra skirtas grafiniam BPEL dokumentų redagavimui. BPEL dokumentuose saugomi verslo procesų aprašai, kuriais į vientisą ilgalaikį procesą yra apjungiamos įvairios tinklo paslaugos. BPEL dokumentus vėliau gali tiesiogiai įvykdyti verslo procesų valdymo varikliai (*Apache ODE*, *bexee*, ir kt.)

9 pav. yra pateiktas grafinio BPEL dokumentų redaktoriaus vaizdas. BPEL dokumentus vaizduojančios schemas yra sudarytos iš dviejų pagrindinių elementų tipų:

1. bloku, vaizduojančių naudojamą žiniatinklio paslaugą;
2. proceso valdymo priemonių (sąlygų tikrinimo operatorių, lygiagrečio, klaidų apdorojimo ir kt.).



9 pav. Grafinis BPEL dokumentų redaktorius

Redaktoriuje yra realizuotos įvairios pagalbinės funkcijos, tokios kaip žiniatinklio paslaugų paieška, WSDL aprašų išgavimas ir kt. Įrankis yra suderinamas su *Apache Ant* automatizavimo priemonėmis [2], todėl yra galimybė tiesiogiai atnaujinti proceso vykdymo aplinką.

Bandymų metu pastebėtos dvi redaktoriaus problemos:

1. Paini vartotojo sąsaja. Sąsaja yra perkrauta – daug langų, įvairių parametru reikšmėms įvesti skirtų laukų, o paaiškinimai ne visais yra atvejais pakankami.
2. Nestabilus darbas. Dirbant paketu, pastebėti programos darbo sutrikimai bei nekorektiškai apdoroti vartotojo įvedami duomenys.

Atkreiptinas dėmesys į tai, jog redaktorius yra ankstyvoje kūrimo stadijoje. Naujausia išleista versija – v0.3.

Naujausiai įskiepio versijai skirtos tinklapio adresas:

<http://download.eclipse.org/technology/bpel/update-site/>

## **1.6. Darbo tikslas ir siekiami privalumai**

Darbo metu siekiama sukurti grafinį aukšto lygmens verslo proceso modelį ir juo pagrįstą redaktorių, kuriuo būtų galima greitai ir efektyviai modeliuoti verslo proceso parametrus. Tam būtina per kuo trumpesnį laiką apibrėžti:

1. verslo procesą sudarančius veiklos etapus;
2. verslo partnerius;
3. proceso būsenų kaitą;
4. proceso dalyvių siunčiamus duomenų srautus.

Kad siūlomas sprendimas įgytų pranašumą prieš egzistuojančias standartines UML priemones, bus siekiama, jog verslo procesų modelis ir jo pagrindu projektuojama programinė įranga taptų suprantama ir aiški galutiniam vartotojui. Siekiant šio tikslo, kuriama CASE priemonė bus specializuojama, ją grindžiant pasiūlytu konceptualių modelių ir derinant prie srities specifikos. Naudodami sukurtą programinę įrangą vartotojai galės susitelkti ties sprendžiamomis problemomis, skirdami kuo mažiau pastangų priemonės įsisavinimui.



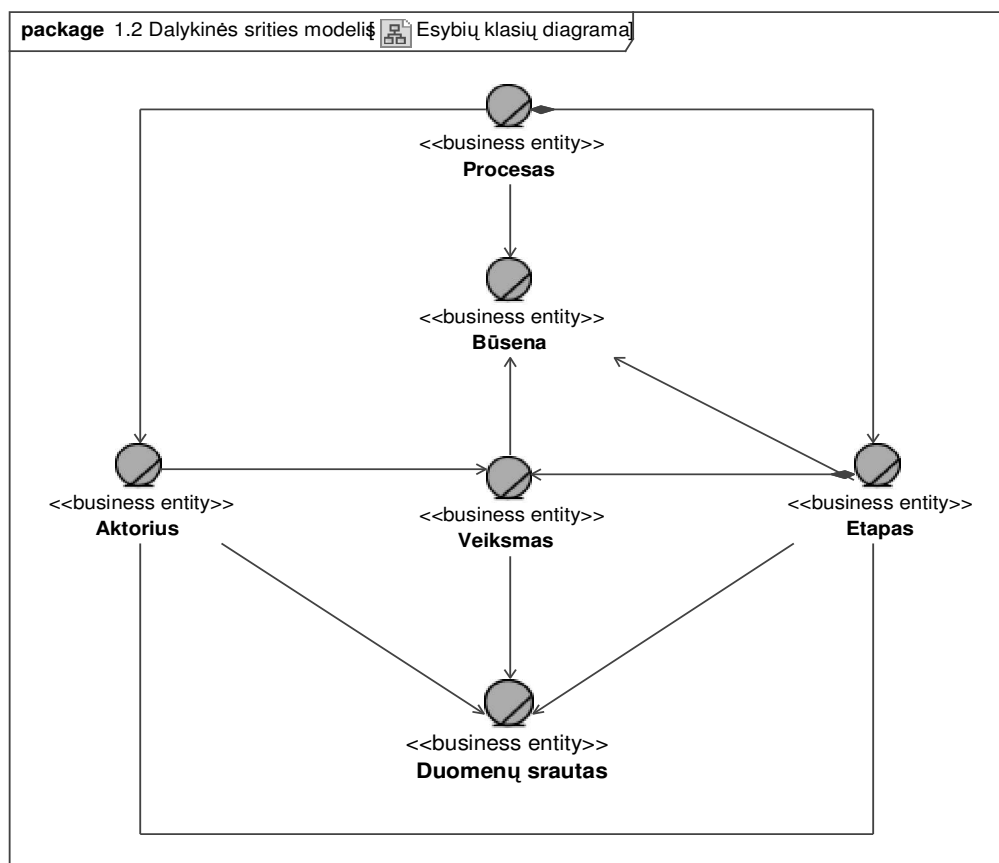
## 2. Paslaugomis pagrįsto verslo proceso modelis

### 2.1. Pagrindiniai principai

Paslaugomis pagrįsto verslo proceso modelis sudarytas remiantis 1.6 sk. išdėstytais principais. Siekiama parinkti tokią modelio elementų aibę, kuria būtų galima visapusiškai apibrėžti verslo procesą ir jo sudedamąsias dalis. Be to, būtina nustatyti tokias modelio elementų priklausomybės, kurios, būdamos griežtai apibrėžtos, taip pat yra būtų aiškios ir intuityvios vartotojui. Realizavus šiuos apribojimus, galimų vartotojo sprendimų aibė taps minimalia. Tokiu būdu bus ribojama vartotojo klaidų galimybė ir išpildomas suprantamumo reikalavimas.

### 2.2. Modelio elementai

10 pav. yra pateiktas dalykinės srities modelis. Šioje diagramoje yra vaizduojama verslo proceso modelį sudarančių elementų aibė ir baziniai elementų sąryšiai.



10 pav. Verslo proceso modelį sudarančių elementų priklausomybės

Atskirų modelio elementų paskirtis yra pateikta 1 lentelėje.

1 lentelė. Verslo proceso modelio elementai

Modelio elementas	Paskirtis
Procesas	Modelio konteineris, skirtas patalpinti kitus modelio elementus.
Būsena	Skirta modeliuoti ilgalaikio verslo proceso būsenų kaitą.
Etapas	Atskira ilgalaikio verslo proceso dalis, turinti pradžios ir pabaigos būsenas.
Aktorius	Verslo procese dalyvaujantis verslo partneris.
Veiksmas	Verslo proceso metu aktorių vykdoma veikla.
Duomenų srautas	Duomenys ir pranešimai, siunčiami vieno verslo partnerio kitam.

Toliau esančiuose poskyriuose atskirai detalizuojami modelio elementai.

### 2.2.1. Etapas

Verslo procesas yra sudarytas iš etapų, kuriuos galima traktuoti kaip ilgalaikes verslo transakcijas, sekos. Atskiri etapai specifikuojami komunikacinėmis veiksmų kilpomis (žr. 1.4 sk.). Kadangi modelis yra skirtas projektuoti verslo partnerių tarpusavio komunikaciją, nedetalizuojant procesų, vykdomų vidinėse partnerių IS, nėra tikslinga išplėsti bazines komunikacines veiksmų kilpas. Verslo procesą sudarančius etapus ir jų kompensacijas į vientisas sekas jungia proceso būsenų kaita (žr. 2.2.4 sk.).

Kiekvienam verslo proceso etapui gali būti priskiriamas kompensacinis veiksmas. Kompensacija, analogiškai kompensuojamajam veiksmui, taip pat yra specifikuojama komunikacine veiksmų kilpa.

Proceso vykdymo metu susidarius išimtinėms situacijoms, kai reikia nutraukti procesą ir kompensuoti sukeltus pokyčius [4], proceso valdymas perduodamas su verslo proceso etapu susietu kompensacinės sekos veiksmui. Kompensaciniai veiksmai jungiami į seką remiantis tais pačiais principais kaip ir modeliuojant pagrindinį verslo procesą.

Verslo proceso etapas yra žymimas ovalu (12 pav.). Etapų kompensacijos papildomai pažymimos uždara rodykle ovalo viduje.

### 2.2.2. Aktorius

Kiekviename verslo procesą sudarančiame etape dalyvauja du aktoriai – iniciatorius ir vykdytojas. Vykdam atskirus verslo proceso etapus, iniciatorius gali kreiptis į skirtingus vykdytojus ir naudotis jų paslaugomis.

Verslo proceso gyvybingumo požymis yra iniciatoriaus tikslai, kuriuos galima susieti su galine proceso būsena, t.y. sėkmingos verslo proceso pabaigos požymis yra būsena, kuriai esant išpildomi iniciatoriaus tikslai.

Modelyje aktorių žymintis simbolis yra pavaizduotas 13 pav.

### 2.2.3. Veiksmas

Vykdamas verslo proceso etapą, iniciatorius ir vykdytojas atlieka po veiksmą, atitinkamai – inicijuojantį ir atsakantį. Šiuos veiksmus galima traktuoti kaip specifinius verslo partnerių vidinių IS panaudojimo atvejus. Įvertinant SOA principus, į komunikacinę kilpą įtraukiami partnerių IS panaudojimo atvejai yra standartizuojami ir vykdomi tokia tvarka:

1. sugeneruojamas duomenų srautas (žr. 2.2.5 sk.),
2. iškviečiama partnerio teikiama paslauga ir jai perduodamas proceso valdymas,
3. proceso būseną pakeičiama iš veiksmo prieš į veiksmo po būseną (žr. 2.2.4 sk.).

Modelyje veiksmas yra žymimas elipse (13 pav.).

### 2.2.4. Būseną

Vykdamas verslo proceso etapus, nuosekliai kinta verslo proceso būseną. Siekiant, jog ilgalaikis procesas būtų vykdomas nuosekliai, įvedama *exist* priklausomybė tarp etapo pabaigos būsenos ir kito etapo pradžios būsenos, t.y. iškeliamą sąlygą, draudžiantį verslo proceso etapą pradėti anksčiau nei sėkmingai pabaigiamas ankstesnysis proceso etapas.

Būsenas išreiškus elementarių požymių rinkiniais, būtina proceso *i*-tojo etapo pradžios sąlygą:

$$S_{i-1}^f \cap S_i^s \neq \emptyset \text{ kai } i > 1. \quad (1)$$

Čia:

$S_{i-1}^f$  – (*i*-1)-tojo etapo pabaigos būseną nusakančių požymių rinkinys.

$S_i^s$  – *i*-tojo etapo pradžios būseną nusakančių požymių rinkinys.

Sėkmingai vykdamas verslo proceso etapus, jų pabaigos būsenos yra kaupiamos bendroje proceso būsenoje:

$$S_p = \bigcup_{i=1}^n S_i^f \quad (2)$$

Čia:

$S_p$  – proceso būseną.

*n* – sėkmingai įvykdytų proceso etapų skaičius;

$S_i^f$  – *i*-tojo etapo pabaigos būseną ( $i \leq n$ );

Įvertinus (2) išraišką, (1) sąlygą galima pertvarkyti:

$$S_p \cap S_i^s \neq \emptyset; i > 0. (3)$$

Čia:

$S_p$  – proceso būseną.

$S_i^s$  –  $i$ -tojo etapo pradžios būseną.

Analogiškai, su  $i$ -tuoju kompensacinės sekos etapu susiejama kompensacijos pabaigos būseną  $S_i^c$ , tokia, kad:

$$S_i^f \cup S_i^c \cong S_i^s. (4)$$

T.y. sėkmingai įvykdžius  $i$ -tojo proceso etapo kompensaciją, proceso būseną tampa ekvivalenti  $i$ -tojo proceso etapo pradžios būsenai. Iš to išplaukia, jog, įvykdžius  $n$  proceso etapų, būtina įvykdytos proceso dalies kompensavimo sąlyga yra:

$$S_p = \left( \bigcup_{i=1}^n S_i^f \right) \cup \left( \bigcup_{i=1}^n S_i^c \right) \cong S_1^s. (5)$$

Galimos situacijos, kai tampa neįmanoma išpildyti (5) sąlygos. Tokių atvejų pakankama sąlyga – verslo proceso metu atliekami negrįžtami pakeitimai, kurių neįmanoma kompensuoti arba juos kompensuojant daromas neigiamas poveikis verslo partneriams.

Pradėjus vykdyti verslo proceso etapą (iniciatoriui atlikus savo panaudojimo atvejį), procesas perduodamas į tarpinę būseną. Iškilus kritinėms situacijoms, tarpinė būseną gali būti panaudota kaip identifikatorius, skirtas nustatyti kompensacinį veiksma, kuriam reikia perduoti proceso valdymą.

Modelyje proceso būseną yra žymima stačiakampiu (žr. 12 ir 13 pav.).

(1) sąlyga žymima punktyrine rodykle su *exist* priklausomybės kryptį nurodančiu užpildytu trikampiu (žr. 12 pav.)

### 2.2.5. Duomenų srautas

Vykstant verslo procesui, jo dalyviai tarpusavyje keičiasi pranešimais. Kadangi šiuo metu pranešimų, perduodamų žiniatinkliu, vyraujantis formatas yra XML, duomenų srautams modeliuoti yra tikslinga taikyti priemones, kurias nesudėtinga transformuoti į XML schemas.

Duomenų srautai gali būti specifikuojami įvairiomis priemonėmis:

- XSD redaktoriais;
- klasių diagramomis;
- ERD diagramomis;
- įvairiomis blokinėmis schemomis.

Jei verslo procesas yra sudaromas iš jau egzistuojančių žiniatinklio paslaugų (angl. *web services*), duomenų struktūrų modeliavimas netenka aktualumo, nes tinklo paslaugų aprašuose (WSDL) yra apibrėžiami pranešimų formatai ir duomenų tipai.

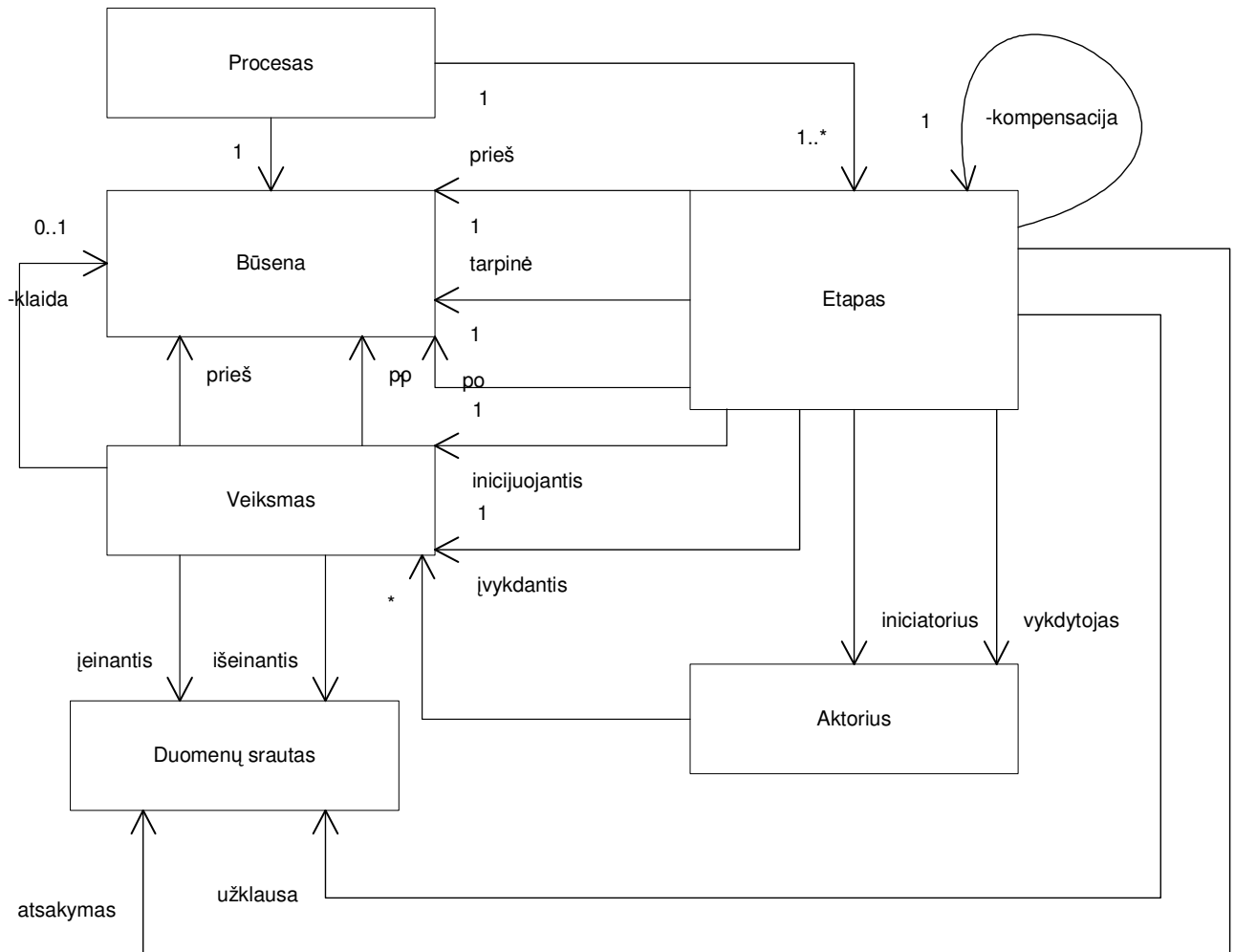
Modelyje duomenų srautas yra žymimas punktyrine rodykle su srauto pavadinimu (13 pav.).

### 2.2.6. Procesas

Procesas yra modelio konteineris, t.y. yra pagalbiniis objektas, nesusijęs su modelio logika, ir yra naudojamas kaip modelio aplinka. Visi modelio objektai tiesiogiai arba netiesiogiai priklauso proceso tipo objektui.

## 2.3. Priklausomybės

2.2 skyriuje aprašytų modelio elementų būtinos tarpusavyo priklausomybės yra pateiktos 11 pav. esančioje diagramoje.



**11 pav.** Modelio elementų priklausomybės

Korektiškai realizavus visas 11 pav. pateiktas priklausomybes, modelis tampa reikalaujamos – griežtos – formos (žr. 2.4 sk. esančius pavyzdžius).

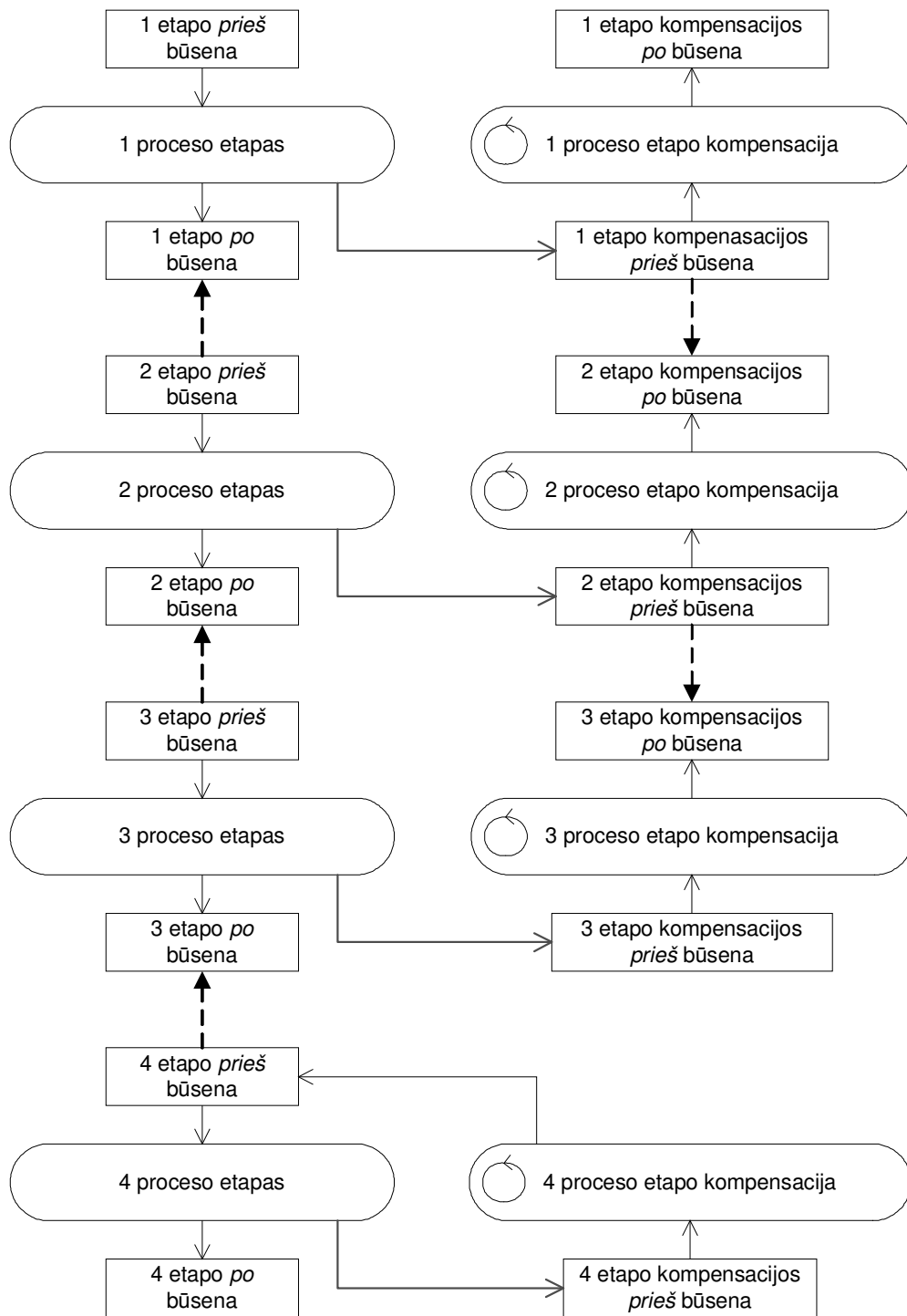
## **2.4. Grafinė notacija**

Verslo proceso modelis yra specifikuojamas į dvių lygių diagramomis – bendra viso verslo proceso diagrama (2.4.1 sk.) ir atskirus proceso etapus specifikuojančiomis žemesnio lygmens diagramomis (2.4.2 sk.).

### **2.4.1. Verslo proceso diagrama**

Verslo procesą vaizduojančią diagramą (pavyzdys pateiktas 12 pav.) sudaro šie modelio elementai:

1. Verslo proceso ir kompensacinių veiksmų sekos (atitinkamai – kairioji ir dešinioji sekos).
2. Verslo proceso etapai ir jų kompensaciniai veiksmai (ovalai).
3. Etapų prieš ir po būsenos (stačiakampiai).
4. Būsenos, į kurias pereinama įvykus klaidoms (su etapais susietos raudonomis rodyklėmis).
5. Etapų vykdymo pradžios sąlygos (punktyrinės rodyklės).



**12 pav.** Verslo procesą specifikuojanti diagrama

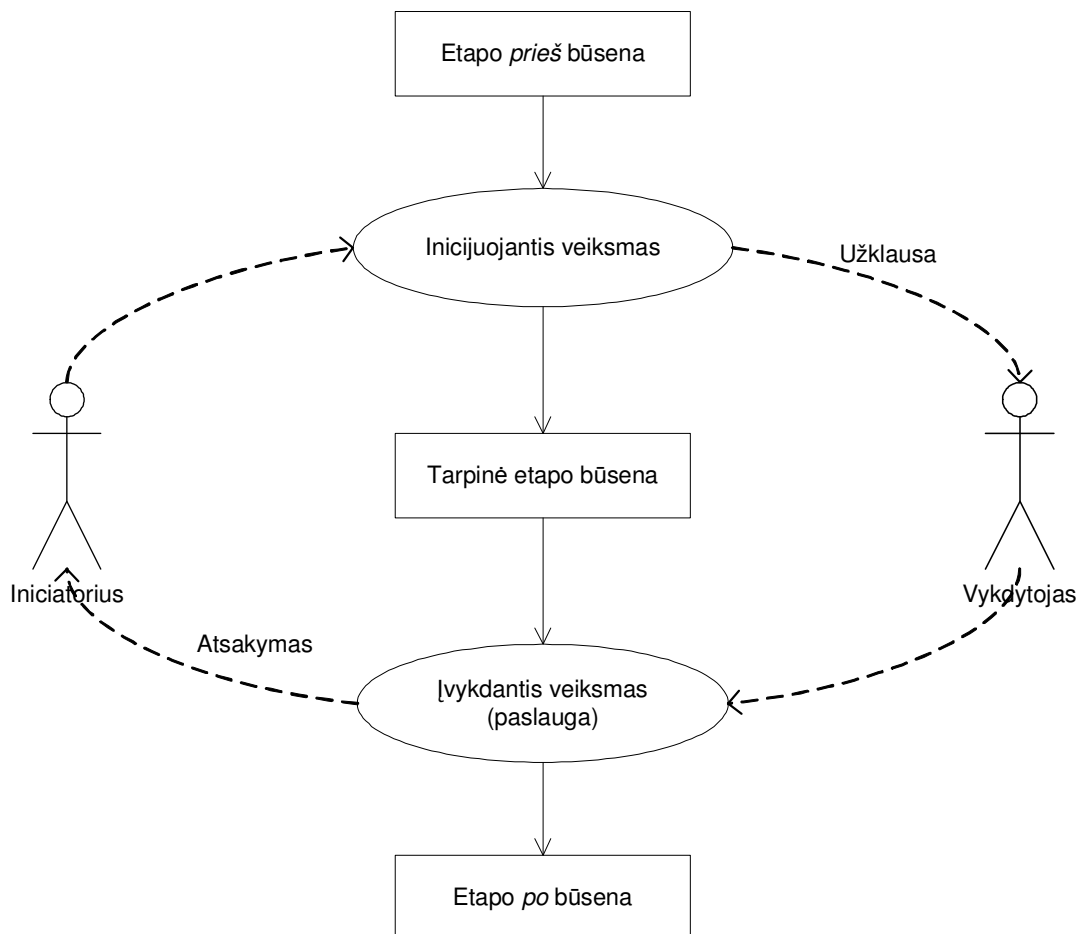
Verslo proceso etapai yra detalizuojami atskiromis diagramomis (žr. 2.3.2 sk.).



## 2.4.2. Proceso etapo diagrama

Verslo proceso etapą vaizduojančią diagramą (pavyzdys pateiktas 13 pav.) sudaro šie modelio elementai:

1. Etapo prieš, po ir tarpinė būsenos (stačiakampiai);
2. Būsenų kaita (vertikalios rodyklės).
3. Proceso etape dalyvaujantys verslo partneriai;
4. Inicijuojantis ir įvykdantis etapo veiksmai (elipsės);
5. Duomenų srautai (rodyklės su srauto pavadinimu; jei pavadinimo nėra – rodyklės nurodo yra proceso valdymo perdavimą).



**13 pav.** Verslo proceso etapą specifikuojanti diagrama

13 pav. pavaizduota diagramos forma yra griežta, t.y. visi proceso etapai privalo būti specifikuojami vienodo formato diagramomis su identiškais elementais ir jų sąryšiais.

## **2.5. Apibendrinimas**

Pagal reikalavimus sudarytas grafinis verslo proceso modelis yra pakankamas specifikuoti paslaugomis paremtą verslo proceso eigą, proceso dalyvius ir jų tarpusavio bendradarbiavimą. Siekiant sumažinti galimų proceso projektavimo klaidų skaičių, nustatyta:

1. minimali modelio elementų aibė (žr. 2.2 sk.).
2. griežtos elementų tarpusavio priklausomybės (žr. 2.3, 2.4 sk.).
3. griežta diagramų forma ir jų sąryšiai (žr. 2.4 sk.).

Remiantis 2.1 – 2.4 sk. išdėstytais verslo proceso modeliavimo principais, suprojektuotas grafinis paslaugomis pagrįstų verslo procesų grafinis redaktorius. Redaktoriaus specifikacija yra pateikta šio darbo 3 skyriuje.

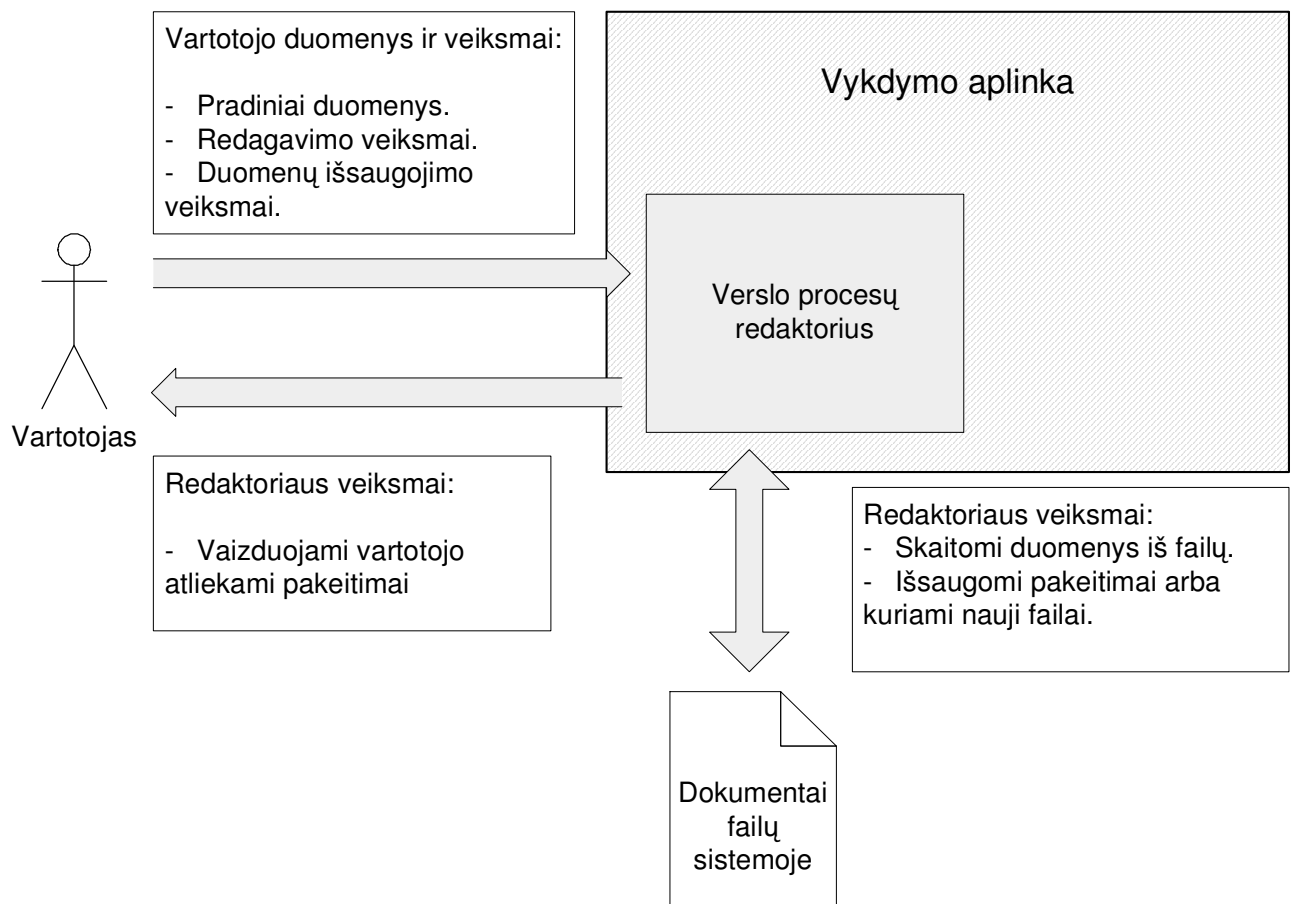
### 3. Verslo procesų redaktorius

#### 3.1. Reikalavimai

##### 3.1.1. Siekiamos sistemos apibrėžimas

Verslo procesų redaktorius yra izoliuota sistema, veikianti uždaroje vykdymo aplinkoje ir yra skirta autonominiam darbui. Sistemos darbą ir rezultatus vienareikšmiškai apibrėžia pradiniai duomenys ir vartotojo veiksmai.

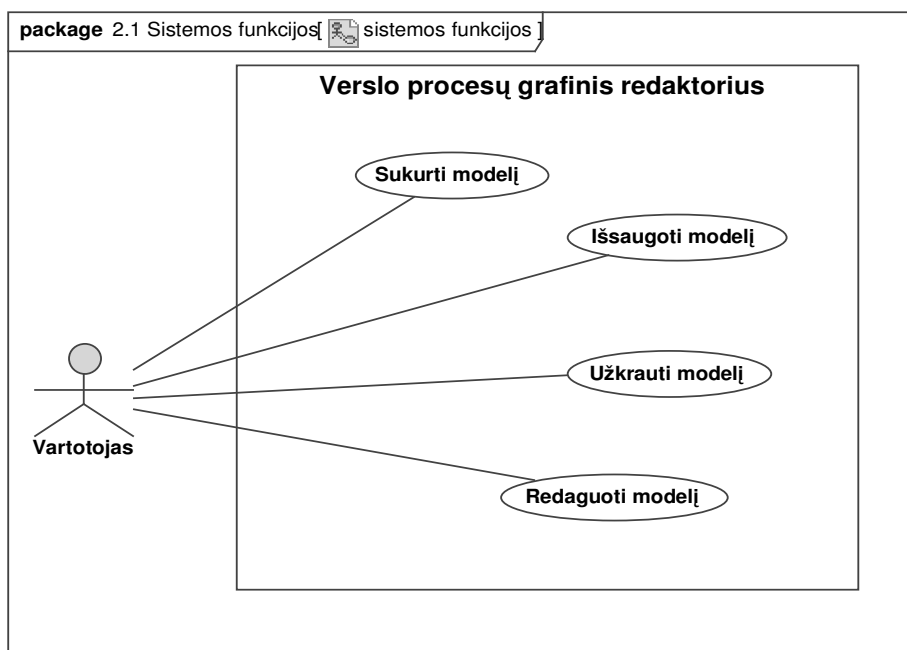
Verslo procesų redaktoriaus aplinka yra pavaizduota 14 pav.



14 pav. Sistemos aplinka

### 3.1.2. Kompiuterizuojamos funkcijos

Kompiuterizuojamos funkcijos yra pateiktos 15 pav. kaip panaudojimo atvejų diagrama.



15 pav. Kompiuterizuojamos funkcijos

Visi sistemos vartotojai turi vienodas teises, ir jie gali naudotis visomis sistemos teikiamomis funkcijomis.

2 – 5 lentelėse pateikiami sistemos panaudojimo atvejų aprašai.

2 lentelė. Panaudojimo atvejo „Sukurti modelį“ aprašas

<b>Panaudojimo atvejo pavadinimas</b>	Sukurti modelį		
<b>Galimos būsenos prieš įvykdymą</b>			
<b>Būsena po įvykdymo</b>	Modelis nėra išsaugotas		
<b>Veiksmų seka</b>	<b>Nr.</b>	<b>Vartotojo veiksmai</b>	<b>Sistemos atsakas</b>
	1.	Spaudžia sukūrimo mygtuką	
	2.		Išveda modelio parametrų langą
	3.	Įveda pradinis parametrus	
	4.		Apdoroja įvestus parametrus
	5.		Atidaro redagavimo langą

3 lentelė. Panaudojimo atvejo „Redaguoti modelį“ aprašas

<b>Panaudojimo atvejo pavadinimas</b>	Redaguoti modelį		
<b>Galimos būsenos prieš įvykdymą</b>	Modelis yra išsaugotas Modelis nėra išsaugotas		
<b>Sistemos būsena po įvykdymo</b>	Modelis nėra išsaugotas		
<b>Veiksmų seka</b>	<b>Nr.</b>	<b>Vartotojo veiksmai</b>	<b>Sistemos atsakas</b>
	1.	Keičia modelį	
	2.		Vaizduoja pakeitimus

4 lentelė. Panaudojimo atvejo „Išsaugoti modelį“ aprašas

<b>Panaudojimo atvejo pavadinimas</b>	Eksportuoti modelį		
<b>Galimos būsenos prieš įvykdymą</b>	Modelis yra išsaugotas Modelis nėra išsaugotas		
<b>Sistemos būsena po įvykdymo</b>	Modelis yra išsaugotas		
<b>Veiksmų seka</b>	<b>Nr.</b>	<b>Vartotojo veiksmai</b>	<b>Sistemos atsakas</b>
	1.	Spaudžia išsaugojimo mygtuką	
	2.		Išveda standartinį išsaugojimo failę langą
	3.	Įveda failo vardą	
	4.		Išsaugo modelį failę

5 lentelė. Panaudojimo atvejo „Užkrauti modelį“ aprašas

<b>Panaudojimo atvejo pavadinimas</b>	Užkrauti modelį		
<b>Galimos būsenos prieš įvykdymą</b>	Modelis nėra užkrautas		
<b>Sistemos būsena po įvykdymo</b>	Modelis yra išsaugotas		
<b>Veiksmų seka</b>	<b>Nr.</b>	<b>Vartotojo veiksmai</b>	<b>Sistemos atsakas</b>
	1.	Spaudžia užkrovimo mygtuką	
	2.		Išveda standartinį failo atidarymo langą
	3.	Nurodo failą	
	4.		Užkrauna modelį iš failo
	5.		Vaizduoja modelį ekrane

### 3.1.3. Reikalavimai duomenims

Duomenys bus saugomi išoriniuose duomenų failuose, kurie turės tenkinti šiuos reikalavimus:

1. Duomenų suderinamumo. Duomenų failuose turės būti su kuriama programine suderinami dokumentai, t.y. jie turės atitikti programinės įrangos palaikomą duomenų formatą.
2. Duomenų korektiškumo. Duomenų failuose išsaugoti modelių aprašai turės būti atitikti realizuotas modelio taisykles, elementus ir jų sąryšius.

### 3.1.4. Nefunkciniai reikalavimai ir apribojimai

#### 3.1.4.1. Reikalavimai sąsajai

Reikalavimai vartotojo sąsajai yra išdėstyti 6 lentelėje.

6 lentelė. Reikalavimai vartotojo sąsajai

Reikalavimas	Paiškinimas
Funkcinių reikalavimų realizavimas	Visi funkciniai reikalavimai turi būti realizuoti vartotojo sąsajoje. Negali būti ne grafinėje vartotojo sąsajoje realizuotų reikalavimų.
Sąsajos panaudojamumas	Sąsaja turi būti intuityvi. Turi būti galimybė pasiekti vartotojo vadovą tiesiai iš programos aplinkos nesinaudojant papildomomis failų paieškos programomis.
Vartotojo veiksmų kontrolė	Sąsaja turi atlikti vartotojo įvedamų duomenų ir veiksmų kontrolę.

#### 3.1.4.2. Reikalavimai kokybei

Reikalavimai programos kokybei yra išdėstyti 7 lentelėje.

7 lentelė. Reikalavimai kokybei

Reikalavimas	Paiškinimas
Patikimumas	Programa turi dirbti stabiliai, operacijų rezultatai turi atitikti vartotojų laukiamus rezultatus. Identiškų veiksmų sekų su tais pačiais duomenimis rezultatai privalo sutapti
Saugumas	Programa neturi atlikti neleistinių veiksmų operacinėje ir failų sistemose. Su sistema nesusijusios programinės įrangos darbui sistemos veikla negali turėti jokios įtakos, išskyrus racionalių dalinimąsi resursais.
Parengtumas	Kuriama sistema turi dirbti visada vartotojui pareikalavus, jei darbo aplinka tenkina būtinus reikalavimus.

### 3.1.5. Rizikos veiksniai

Kuriamos programinės įrangos funkcijos nėra trivialios, todėl egzistuoja rizika, kad funkciniai ir nefunkciniai reikalavimai nebus korektiškai įgyvendinti.

Rizikos veiksnių apžvalga yra pateikta 9 lentelėje.

8 lentelė. Projekto rizikos faktoriai

Rizikos veiksnys	Metodai rizikai sumažinti
Pasikeitę arba nauji reikalavimai	Prieš įvedant reikalavimų pakeitimus būtina įsitikinti pokyčių būtinumu. Jei pakeitimai yra motyvuoti, būtina juos dokumentuoti.
Vartotojui nesuprantama ar nepriimtina sąsaja	Bendradarbiaujant su užsakovu sukurti vartotojo sąsajos prototipą. Sukurti integruotą vartotojo žinyną ir/arba pagalbos sistemą.
Pasikeitę standartai	Atnaujinti esamus arba sukurti papildomus su naujais standartais suderinamus programos modulius.

### 3.1.6. Rezultato kokybės kriterijai

Sukurtos programinės įrangos kokybė bus nustatoma pagal tai, kaip sistema išpildo funkcinis ir nefunkcinius reikalavimus.

Rezultato kokybės kriterijai yra išdėstyti 9 lentelėje.

9 lentelė. Programinės įrangos kokybės kriterijai

Kokybės kriterijus	Aprašas
Funkcinių reikalavimų realizacija	Programinė įranga turi įgyvendinti visus projektavimo metu nustatytus funkcinis reikalavimus ir jų pakeitimus.
Nefunkcinių reikalavimų realizacija	Programinė įranga turi atitikti išskeltus stabilumo, saugumo ir parengtumo reikalavimus.
Vartotojo sąsajos kokybė	Vartotojo sąsaja turi būti intuityvi ir suprantama visiems dalykinę sritį išmanantiems vartotojams. Pagalbos sistema turi būti struktūrizuota ir lengvai pasiekama tiesiogiai iš programos.
Techninė dokumentacija	Techninėje dokumentacijoje pateikiamos informacijos turi būti pakakti norint suprojektuoti tolimesnius sistemos patobulinimus ir numatyti darbų apimtis.

## 3.2. Grafinio redaktoriaus projektas

### 3.2.1. Architektūra ir realizavimo priemonės

„Eclipse“ platformoje yra sukurtas priemonių rinkinys (GEF) grafinės informacijos vaizdavimui [5]. GEF – grafinio redagavimo aplinka (angl. *Graphical Editing Framework*) yra Java klasių biblioteka, skirta kurti taikomąsias programas darbui su diagramomis ir grafiniais modeliais. Ši priemonė sudaryta iš „lengvų“ (angl. *lightweight*) komponentų, t.y. parašyta tik Java programavimo kalba ir yra nepriklausoma nuo kompiuterinės platformos.

GEF yra paremta MVC (angl. *Model-View-Controller*) architektūra, kurioje loginis modelis (angl. *model*) yra atskirtas nuo jį reprezentuojančio grafinio vaizdo (angl. *view*). Ryšius tarp modeliuojamų objektų ir vartotojo sąsajoje vaizduojamų elementų kontroliuoja valdiklis (angl. *controller*). Tokios architektūros privalumas yra galimybė atskirti loginio modelio ir grafinės aplinkos realizacijas. Taikant MVC principus, pakeistus grafinę sąsają nereikia keisti modelio logikos.

Tam, kad modelis galėtų būti prijungtas prie valdiklio objektų, reikalingas įvykių mechanizmas, perduodantis valdikliui apie modelio pasikeitimą informuojančius pranešimus. Gavęs tokį pranešimą valdiklis keičia arba sukuria modelio objektą reprezentuojančią figūrą.

GEF bibliotekoje vartotojo grafinių figūrų pakeitimai modeliui perduodami panaudojant užklausų mechanizmą: užklausos yra priimamos atitinkamų figūrų valdiklių ir yra transformuojamos į modelio objektų pakeitimus. Komandų vykdymo metu pakeičiamas modelis ir komanda užregistruojama „Eclipse“ aplinkos komandų steke. Šios komandas gali būti panaudojamos atstatant originalią modelio būseną prieš pakeitimus.

Duomenų ir jų grafinės reprezentacijos atskyrimas yra ir didžiausias MVC architektūros trūkumas – parandami trivialūs ryšiai tarp vartotojo veiksmų ir redaguojamo modelio pokyčių. Augant duomenų apimtims ir modelių kompleksiskumui, gali ženkliai padidėti programos kodo apimtys, o valdiklio sluoksnius tapti neracionaliai sudėtingas.

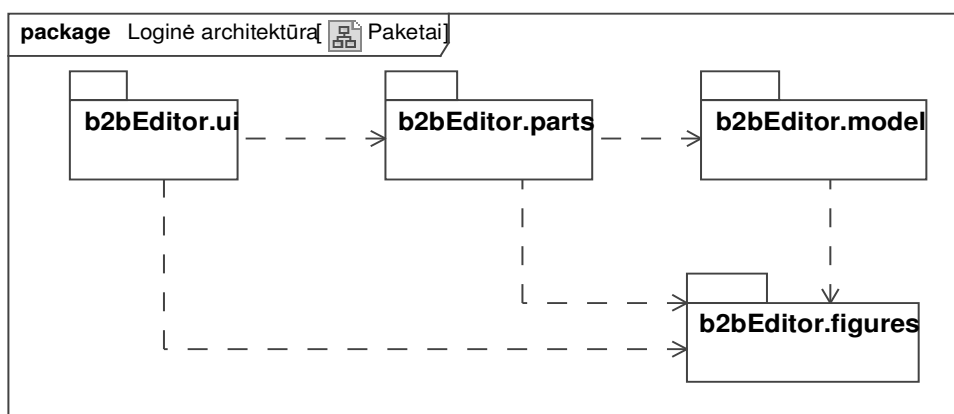


### 3.2.2. Statinė struktūra

Šiame poskyryje pateikiama kuriamos programinės įrangos statinė struktūra: bendra loginė architektūra bei klasių diagramos.

#### 3.2.2.1. Loginė architektūra

Loginės architektūros modelis (16 pav.) parodo, kaip sistemoje bus paskirstyta funkcijų realizacija į paketus ir kokios yra paketų tarpusavio priklausomybės.



16 pav. Redaktoriaus loginė architektūra

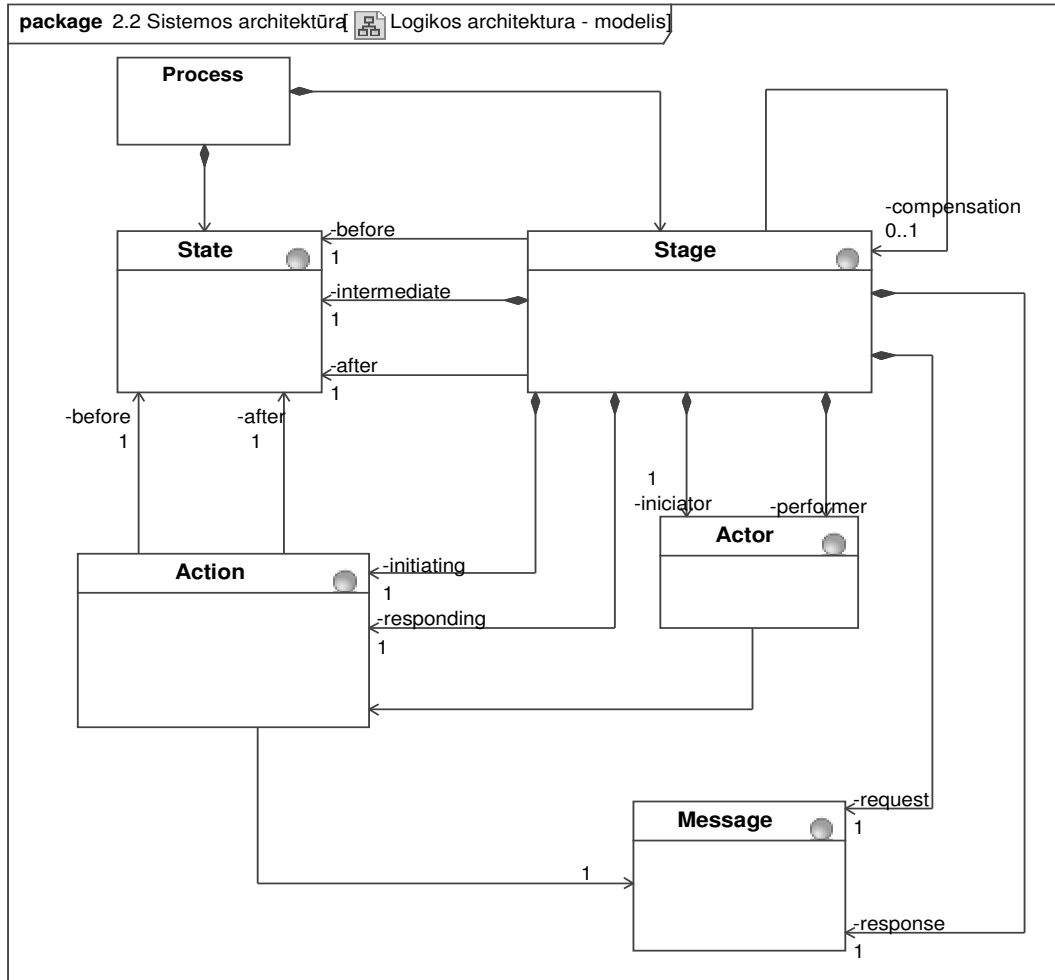
10 lentelėje yra pateikiamos atskiruose paketuose realizuojamos funkcijos.

10 lentelė. Paketuose realizuojamas funkcijos

Paketas	Paskirtis
b2bEditor.ui	„Eclipse“ grafinės aplinkos plėtiniai.
b2bEditor.model	Modelio elementus realizuojančios klasės.
b2bEditor.parts	Valdikliai, susiejantys modelio elementus, elementų grafines reprezentacijas ir grafinę vartotojo aplinką.
b2bEditor.figures	Modelio elementų grafinės reprezentacijos.

### 3.2.2.2. Modelio elementus realizuojančios klasės

Verslo proceso elementus realizuojančios klasės ir jų sąryšiai yra pavaizduoti 17 pav. pateiktoje klasių diagramoje.



17 pav. Modelio elementus realizuojančių klasių diagrama

11 lentelėje yra išdėstyti sąryšiai tarp klasių diagramoje vaizduojamų klasių ir atitinkamų modelio elementų.

11 lentelė. Modelių realizuojančių klasių paskirtis

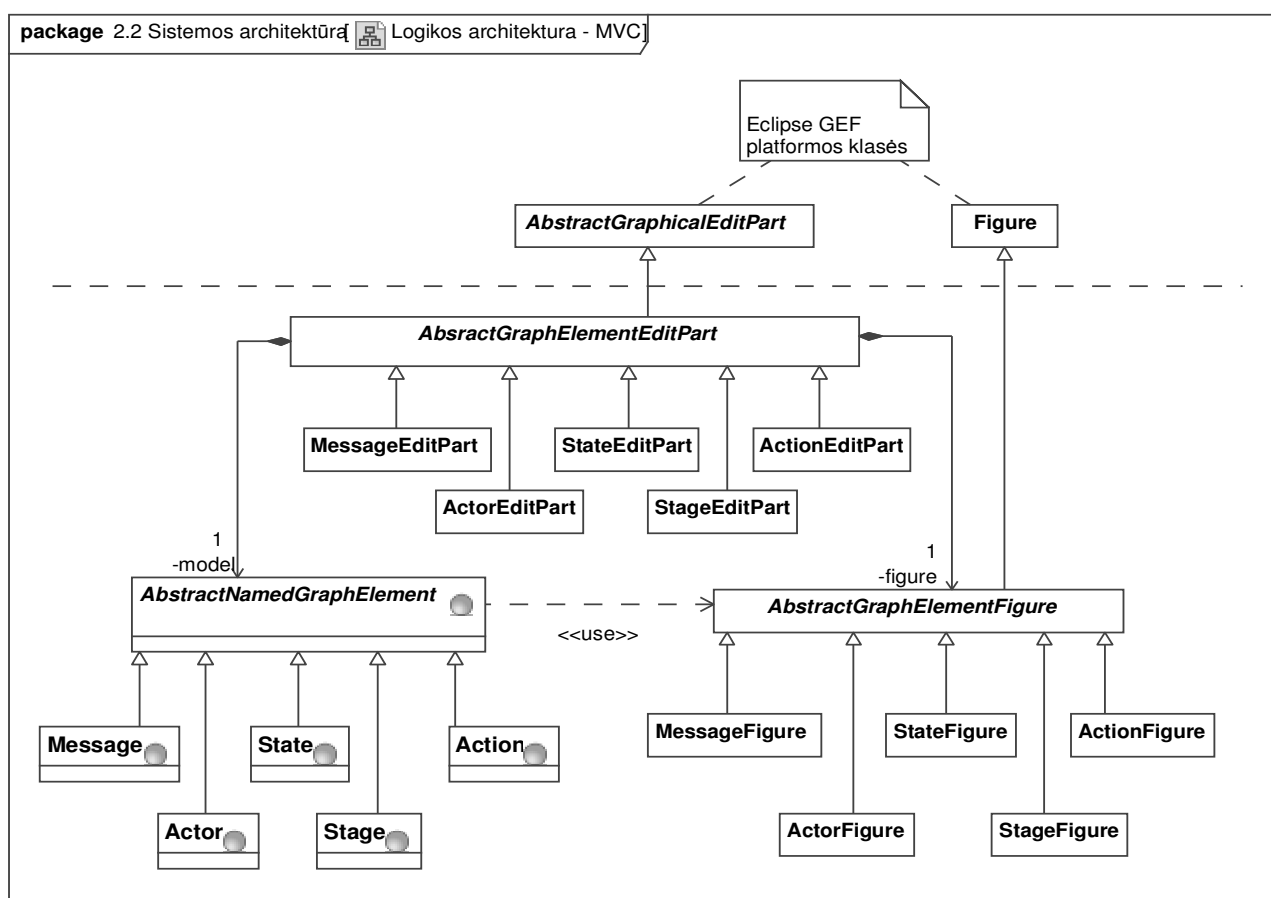
Klasė	Modelio elementas
Process	Modeliuojamas ilgalaikis verslo procesas
State	Verslo proceso būseną
Stage	Verslo proceso etapas
Actor	Verslo partneris
Action	Proceso etapo veiksmas
Message	Duomenų srautas

### 3.2.2.3. MVC architektūra

Pagrindinė grafinio redagavimo problema yra susieti redaguojamo modelio objektus su grafine šių objektų reprezentacija. Taip pat yra būtina, kad modelio pakeitimus būtų galima dinamiškai atvaizduoti reprezentacijoje, o modelio grafinės reprezentacijos pakeitimai atsispindėtų modelyje.

MVC architektūroje modelį ir grafinę reprezentaciją siejantis ryšys yra abstrahuojamas į valdiklio objektą, kurio bazinė GEF realizacija yra `AbstractGraphicalEditPart` klasė.

Grafinio redaktoriaus MVC architektūros realizacija yra pateikta 18 pav. esančioje klasių diagramoje.



18 pav. MVC architektūros realizacija

Struktūrizuojant projektą, įvestos abstrakčios, su MVC architektūra suderintos, klasės: `AbstractGraphElement`, `AbstractGraphElementFigure` ir `AbstractGraphElementEditPart`. Šių klasių paskirtis yra išdėstyta 12 lentelėje.

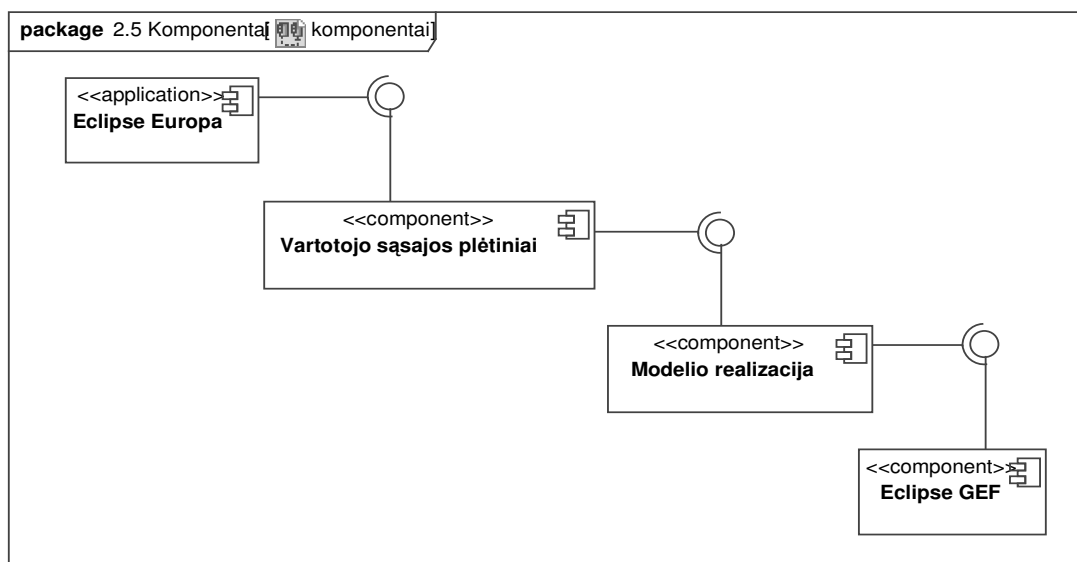
12 lentelė. Pagrindinių MVC architektūros klasių paskirtis

Klasė	Paskirtis
<i>AbstractGraphElement</i>	Modelio elementus realizuojančių klasių prototipas
<i>AbstractGraphElementFigure</i>	Modelio elementus vaizduojančių figūrų prototipas.
<i>AbstractGraphElementEditPart</i>	Modelio elementus ir juos vaizduojančias figūras siejančių valdiklių prototipas.

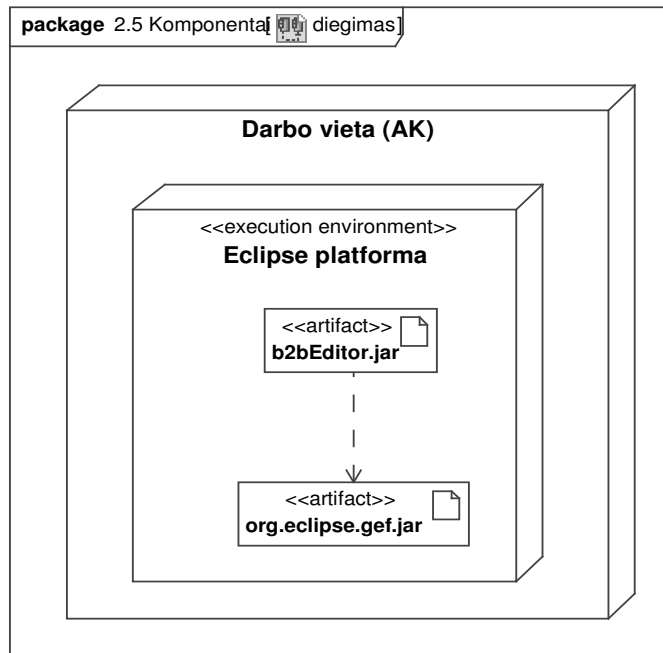
Kiekvienas klasės *AbstractGraphElement* vaikas yra susiejamas su atitinkamais klasių *AbstractGraphElementFigure* ir *AbstractGraphElementEditPart* vaikais (pvz., *Message* tipo objektas susiejamas su *MessageFigure* ir *MessageEditPart* tipo objektais). 18 pav. šie ryšiai nėra detalizuoti, nes yra trivialūs.

### 3.2.3. Realizacijos modeliai

Verslo procesų redaktoriaus realizacija yra pavaizduota 19 ir 20 paveiksluose esančiais komponentų ir diegimo modeliais.



19 pav. Grafinio redaktoriaus komponentų modelis



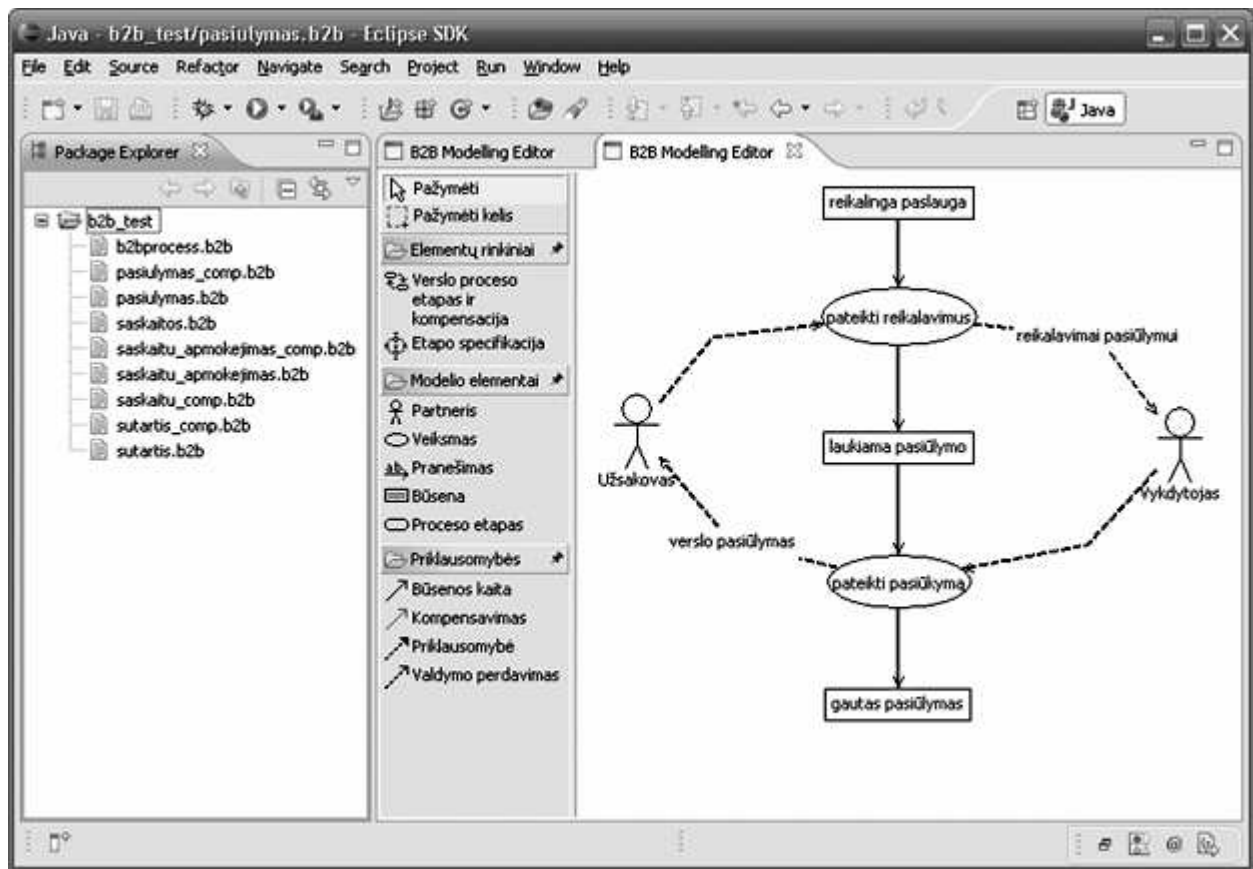
**20 pav.** Grafinio redaktoriaus diegimo modelis

„Eclipse“ platforma yra paremta įskiepių sąsaja, kuri leidžia paprasčiau sukurti, integruoti ir panaudoti programinius įrankius, taupant laiką ir išlaidas. „Eclipse“ platforma realizuota Java priemonėmis ir veikia įvairios architektūros kompiuteriuose bei operacinėse sistemose, palaikančiose Java: Windows, Linux, HP-UX, AIX, Solaris, QNX, Mac OS, ir kt.

### 3.3. Darbo redaktoriumi principai

#### 3.3.1. Darbo aplinka

Į „Eclipse“ aplinką įdiegus verslo procesų redaktoriaus įskiepi (b2bedit), atsiranda papildoma galimybė kurti bei redaguoti verslo proceso modelio failus (\*.b2b). Pradėjus darbą „Eclipse“ aplinkoje, sukuriamas naujas arba atidaromas jau egzistuojantis projektas ir įtraukiamas naujas verslo proceso diagramą aprašantis failas. Taip pat galima užkrauti jau egzistuojantį modelį, pasinaudojus pagrindinio meniu *File* ⇒ *Import* funkcija.



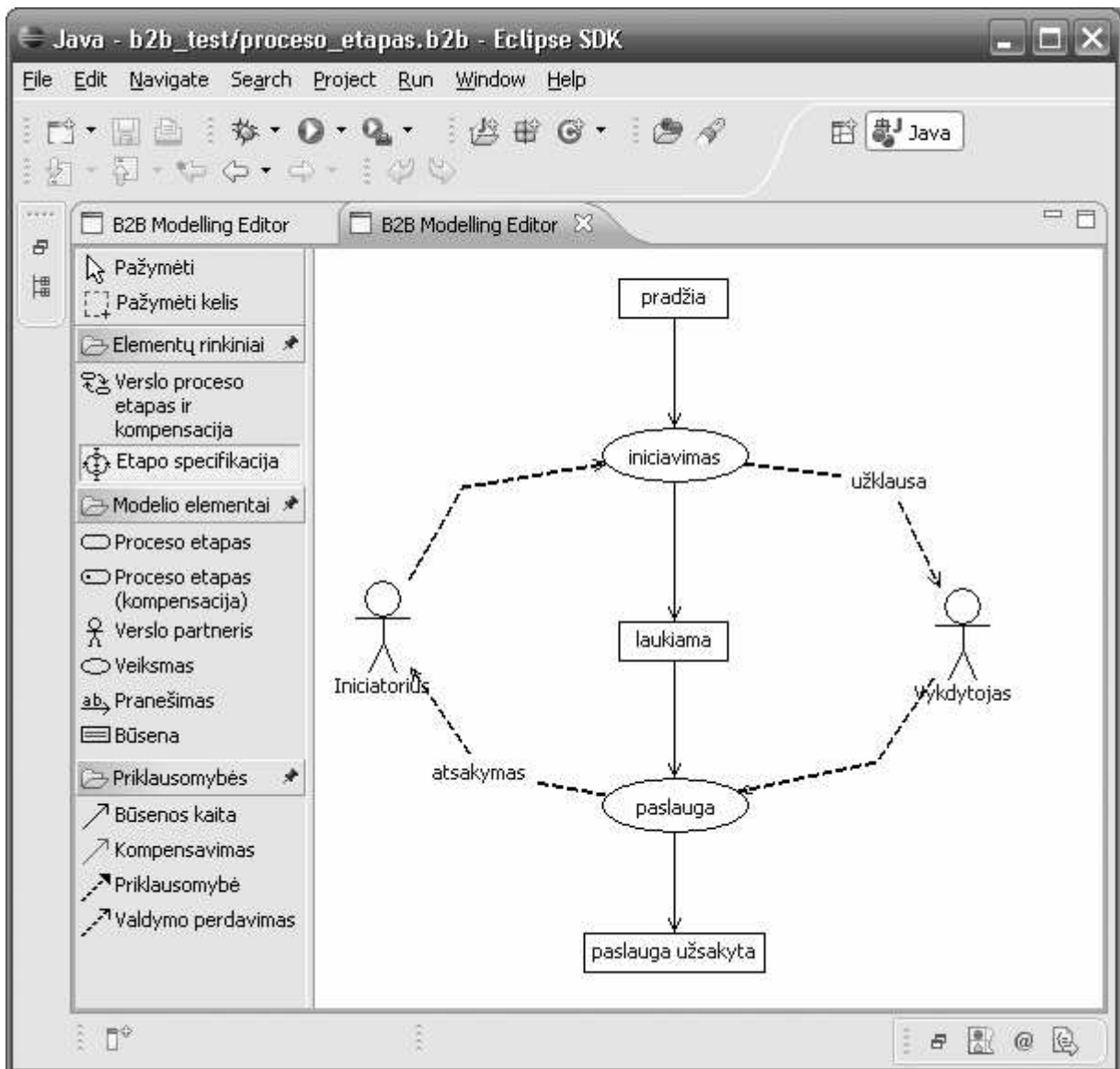
21 pav. Verslo procesų redaktorius „Eclipse“ aplinkoje

Bendras verslo procesų redaktoriaus vaizdas yra pateiktas 21 pav. Tai – tipinė „Eclipse“ pagrindu sukurtų priemonių išdėstymo schema, kurią sudaro 3 pagrindinės dalys:

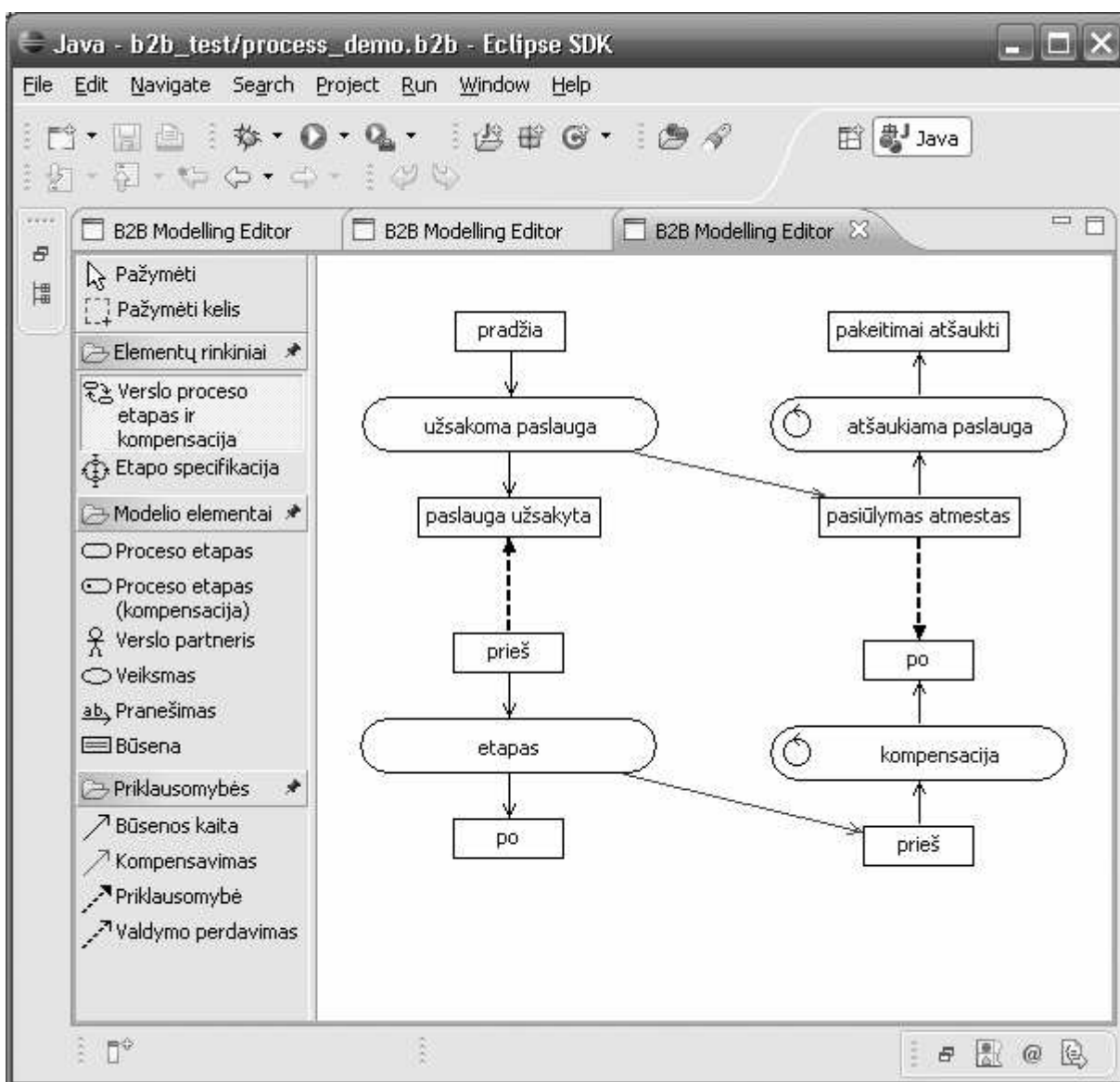
1. į projektą įtrauktų failų sąrašas (kairėje);
2. įrankių juosta (viduryje);
3. redagavimo laukas (dešinėje).

### 3.3.2. Redagavimo priemonės

Vartotojui rekomenduojama į modelį įtraukti jau parengtus šablonus, sukuriamus įrankių juostos skilties „Elementų rinkiniai“ priemonėmis ir juos toliau redaguoti. Šioje juostoje yra įrankis „Verslo proceso etapas ir kompensacija“, skirtas į verslo proceso modelį įtraukti naujus etapus, jų kompensacinius veiksmus, *prieš* ir *po* būsenas bei būtinas priklausomybes. 22 pav. yra pateikiamas proceso modelio pavyzdys, į kurį yra įtraukti du tokie rinkiniai.



22 pav. Verslo proceso modelio realizacija



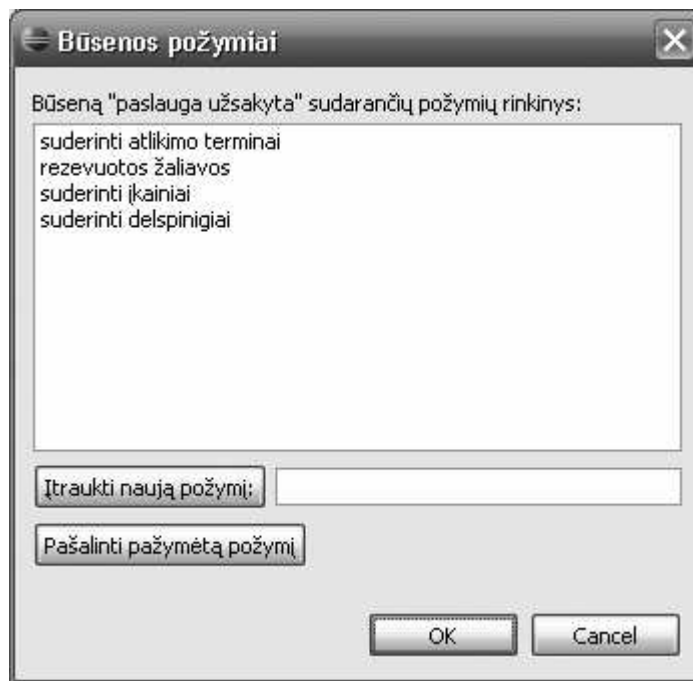
**23 pav.** Proceso etapo specifikacija

23 pav. yra pateikiama verslo proceso etapą specifikuojanti diagrama, sugeneruota panaudojus įrankį „Etapo specifikacija“.

Vartotojas taip turi galimybę į modelį įtraukti pavienius elementus ir jų priklausomybes pasirinkdamas atitinkamus įrankius iš įrankių juostos skilčių „Modelio elementai“ ir „Priklausomybės“.

Proceso būsenas aprašantys požymių rinkiniai yra sudaromi atskirame lange, kuris yra iškviečiamas kontekstiniame būseną vaizduojančio elemento meniu pasirinkus punktą „Būsenos požymiai...“ (24 pav.).





**24 pav.** Būsenos požymių redagavimo langas

## 4. Eksperimentinis verslo proceso modelis

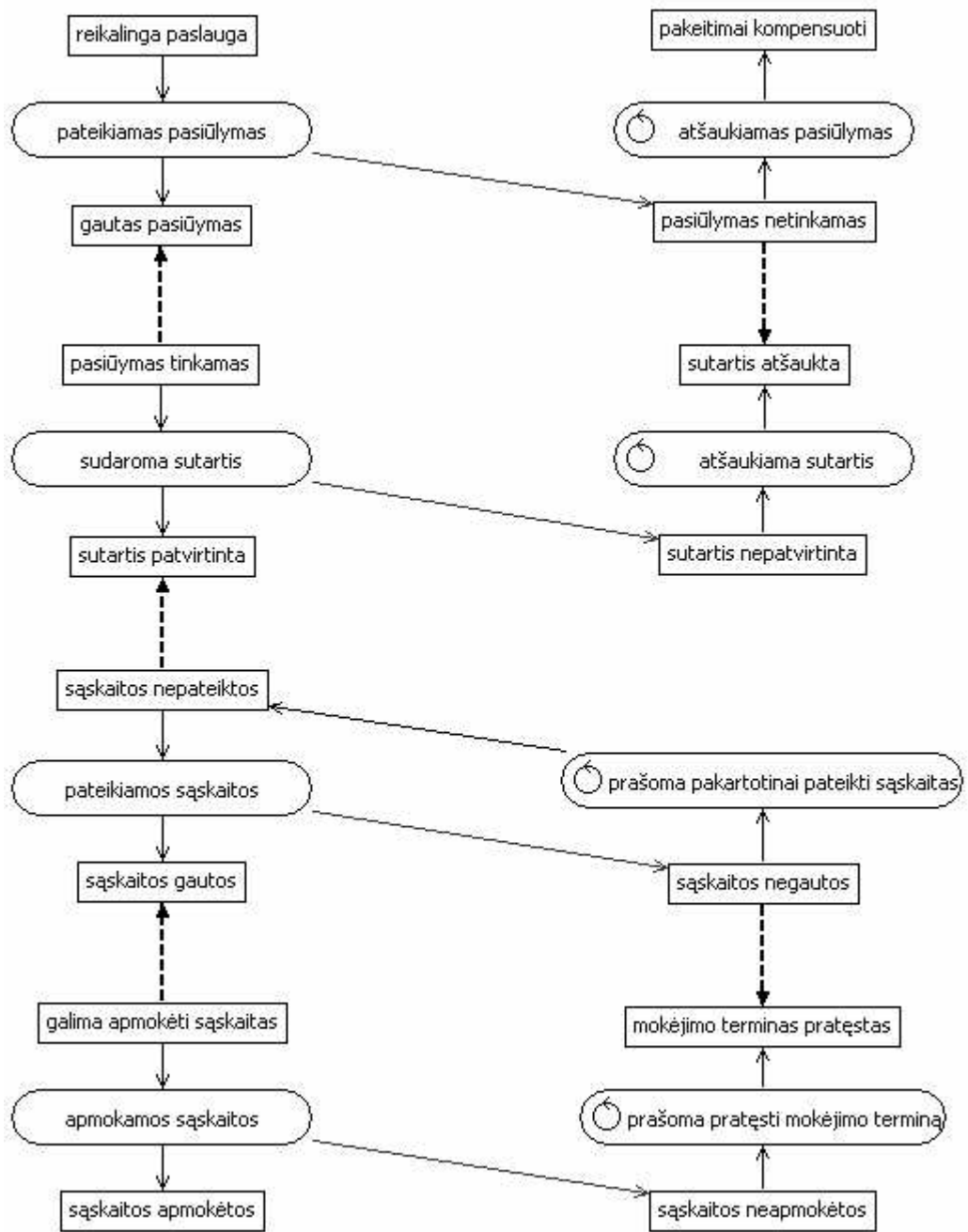
Šiame skyriuje yra pateikiamas bandomasis paslaugomis pagrįsto verslo proceso ir juo paremto grafinio redaktoriaus pritaikymas.

### 4.1. Verslo procesas

Modeliuojamas verslo procesas sudarytas iš šių veiklų:

1. Pateikiamas verslo pasiūlymas.
2. Sudaroma ir patvirtinama sutartis
3. Pateikiamos sąskaitos
4. Apmokamos sąskaitos.

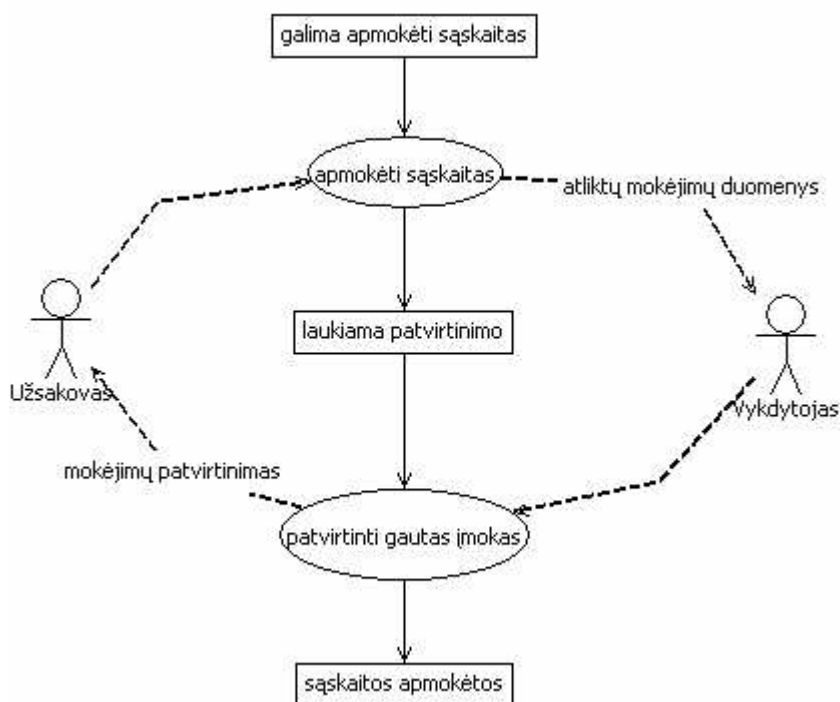
Sėkmingai patvirtinus sutartį, verslo proceso būseną tampa nekompensuojama, t.y. parandama galimybė procesą grąžinti į pradinę ekvivalenčią būseną. 25 pav. esančiame modelyje tai pavaizduota kompensacinio veiksmo „prašoma pakartotinai pateikti sąskaitas“ *po* būseną sutapatinant su veiksmo „pateikiamos sąskaitos“ *prieš* būseną ir panaikinant galimybę kompensacinės sekos metu įvykdyti veiksmą „atšaukiama sutartis“. Dėl šios priežasties kompensacinio veiksmo „atšaukiama sutartis“ *prieš* sąlyga „sutartis nepatvirtinta“ nėra susieta *exist* priklausomybe su veiksmo „prašoma pakartotinai pateikti sąskaitas“ *po* būseną.



25 pav. Paslaugos užsakymo modelis

## 4.2. Verslo proceso etapų specifikavimas

26 – 28 pav. pateikiami galimi verslo proceso etapus specifikuojančių diagramų atvejai.



26 pav. Verslo proceso etapo specifikacija

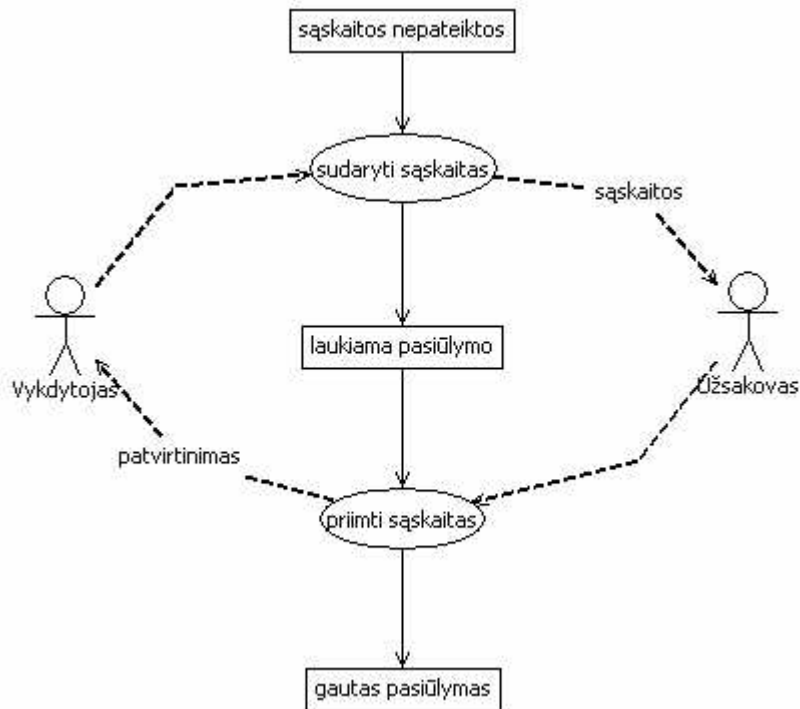
26. pav. yra pateikiama tipinio verslo proceso etapo specifikacija. Proceso etapą inicijuoja verslo partneris „Užsakovas“ įvykdydamas savo vidinės IS panaudojimo atvejį „apmokėti sąskaitas“. Vykdydamas šį panaudojimo atvejį, iniciatorius galimai atlieka vykdytojiui nematomus veiksmus (jungiasi prie banko sistemos ir pan.). Juos įvykdžius, tęsiamas panaudojimo atvejis ir sugeneruojamas bei išsiunčiamas duomenų srautas „atliktų mokėjimų duomenys“ partneriui „Vykdotojas“, t.y. modelis apribojama iki abiejų verslo partneriams aktualių veiksmų, nedetalizuojant vidinių subprocesų.

Analogiškai, paslaugą teikiantis partneris („Vykdotojas“), priėmęs duomenų srautą iš inicijuojančio partnerio, įvykdo savo vidinės IS panaudojimo atvejį „patvirtinti gautas įmokas“, kurio metu sugeneruojamas ir išsiunčiamas duomenų srautas „mokėjimų patvirtinimas“.

Inicijuojančiam partneriui atlikus su verslo proceso etapu susietą savo panaudojimo atvejį, proceso būsena pakeičiama į „sąskaitos apmokėtos“, kuri indikuoja, jog verslo proceso valdymas perduotas vykdančiam partneriui.

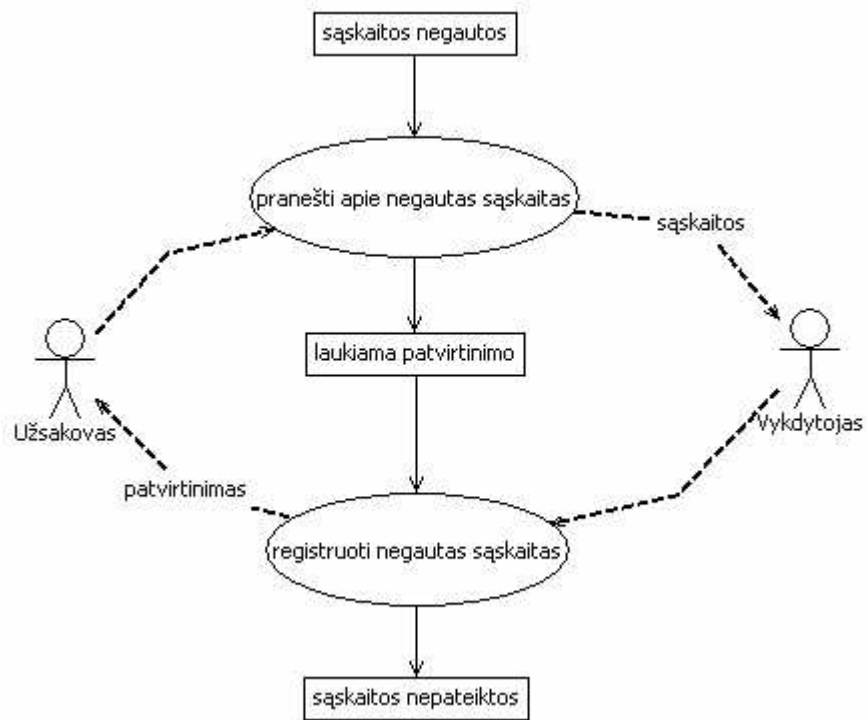
27 pav. pateikiama proceso etapą „pateikiamos sąskaitos“ specifikuojanti diagrama. Šį etapą inicijuoja aktorius „Vykdotojas“, t.y. tarpusavyje sukeičiamos verslo procese dalyvaujančių

partnerių rolės. Toks specifikuojamas galimas, nes verslo proceso modelyje (žr. 2 sk.) nėra reikalaujama, jog partnerių rolės nekistų ilgalaikio verslo proceso metu.



**27 pav.** Verslo proceso etapo, kurį inicijuoja vykdytojas, specifikacija

28 pav. esančia diagrama specifikuojama etapo „pateikiamos sąskaitos“ kompensacija „prašoma pakartotinai pateikti sąskaitas“. Ši diagrama yra analogiška 26 pav. esančiai diagramai, t.y. specifikuojant nėra esminio skirtumo tarp verslo proceso etapo ir kompensacijos.



**28 pav.** Etapo kompensaciją specifikuojanti diagrama.

Visos verslo procesą specifikuojančios diagramos yra pateiktos 1 priede.

## 5. Išvados

1. Siekiant užtikrinti patikimą verslo informacijos sistemų funkcionavimą, būtina aprašyti ilgai trunkančius verslo procesus. Ši problema tampa aktuali dėl sparčiai populiarėjančio SOA pagrindais paremta informacijos sistemų kūrimo.
2. Projektuojant paslaugomis pagrįstas verslo informacijos sistemas ir jų tarpusavio sąveikas, paslaugų, teikiamų nehomogeninėje elektroninėje aplinkoje, panaudojimą patogų specifiškai komunikaciniams veiksniams.
3. Projektuojant ilgalaikius verslo procesus, būtina su atskirais verslo proceso etapais susieti kompensacinius veiksmus. Kompensacinių veiksnių specifikavimas tampa paprastesnis įvedus taisykles, kurias galima vienodai pritaikyti ir verslo proceso etapams, ir tų etapų kompensacijoms.
4. Į verslo procesą įtraukiamus duomenų srautus yra tikslinga projektuoti priemonėmis, kurias nesudėtinga transformuoti į XML schemas.
5. Apibrėžus verslo procesams modeliuoti būtinų elementų rinkinį bei nustatius griežtas modelio elementų priklausomybes ir jas realizavus grafiniame redaktoriuje, sumažinama vartotojo klaidų rizika.
6. Kad projektavimo priemonė būtų suprantama vartotojui, reikalinga praplėsti standartinį modeliavimo funkcionalumą. Tai realizuoti galima prie standartinių įrankių įtraukiant papildomas priemones – parengtus diagramų šablonus, įvairius vedlius ir pan.
7. Realizuojant grafinio redagavimo įrankius yra tikslinga atskirti modelio logiką nuo grafinės reprezentacijos. Kad programinės įrangos architektūra būtų aiški, rekomenduojama pritaikyti jau sukurtus objekcinio programavimo šablonus, tokius kaip MVC.
8. Greitam grafinių redaktorių prototipų sukūrimui yra paranku panaudoti trečiųjų šalių pateikiamas kūrimo platformas ir į jas įtrauktas bibliotekas. Tačiau projektuojant ir realizuojant komercinius grafinio modeliavimo paketus šių priemonių panaudojamumas tampa ribotas dėl įvairių rizikos veiksnių:
  - programinės įrangos architektūra gali tapti sudėtinga, o poreikis minimaliai modifikuoti funkcionalumą pareikalautų neadekvačių resursų ir pastangų;
  - paaiškės, jog nepakanka trečiųjų šalių pateikiamo funkcionalumo;
  - gali būti nutrauktas priemonių tobulinimas ir klaidų taisymas.

9. Toliau darbą galima tobulinti šiomis kryptimis:

- Išplėsti verslo proceso modelį, įtraukiant naujus valdymo operatorius (sąlygų tikrinimo, lygiagretinimo, išimtinių situacijų apdorojimo).
- Realizuoti BPEL ir WSDL dokumentų generatorių.
- Realizuoti arba įtraukti vieną iš trečiųjų šalių siūlomų XML schemų redaktorių.
- Įtraukti papildomas vartotojo veiksmų kontrolės priemones.



## Literatūra

1. Ambrazevičius E., Jasiukevičius A., Paradauskas B., Komunikacinių kilpų informacijos išsaugojimo kriterijus. Informacijos mokslai, 2003, T. 24. pp. 98-107.
2. Apache Ant. Interneto prieiga: <http://ant.apache.org/>. Žiūrėta 2006-10-22.
3. Bertrand G., Incoul Ch., Ramel S., Schmitt M., Gautheron L., Brimont P., Dubois E. Efficient: A Framework for Animating and Validating e-Business Transactions. ERCIM News. Nice, No. 57, 2004
4. Butler M., Chessel M., Ferreira C., Griffn C., Henderson P., Vines D. Extending the Concept of Transaction Compensation. IBM Systems Journal, IBM Technical Journals, 2002, Volume 41, No. 4, pp. 743 - 756.
5. Eclipse. GEF Description. Interneto prieiga: [http://wiki.eclipse.org/index.php/GEF\\_Description](http://wiki.eclipse.org/index.php/GEF_Description). Žiūrėta 2007-02-18.
6. Efficient Project. EFFICIENT toolset demonstration: <http://efficient.citi.tudor.lu/download/Efficient.htm>. Žiūrėta 2006-10-14.
7. Eshuis R., Brimont P., Dubois E., Gregoire B., Ramel S. Animating ebXML Transactions with a Workflow Engine. CoopIS 2003, vol. 2888 of Lecture Notes in Computer Science. Catania, 2003.
8. Eshuis R., Brimont P., Dubois E., Gregoire B., Ramel S. EFFICIENT: A Tool Set for Supporting the Modelling and Validation of ebXML Transactions. Joint 9th European Software Engineering Conference (ESEC) & 11th SIGSOFT Symposium on the Foundations of Software Engineering (FSE-11). Helsinki, 2003, pp. 359-362.
9. Garcia-Molina H., Salem K. „Sagas“. Proceedings of ACM SIGMOD. San Francisco, 1987, pp. 249-259.
10. Gustas R. Towards a Communication Based Approach for Enterprise Modelling Integration. International Conference on Integration and Reuse. Las Vegas, 2000.
11. Hofreiter B., Huemer Ch., Zapletal M. Registering UMM Business Collaboration Models in an ebXML Registry. IEEE Joint Conference on E-Commerce Technology (CEC'06) & Enterprise Computing, E-Commerce and E-Services(EEE'06). San Francisco. 2006.
12. Young Ch., BizTalk Server 2006: The Compensation Model. Interneto prieiga: <http://geekswithblogs.net/cyoung/articles/100424.aspx>. Žiūrėta 2007-04-09.

13. Mertz D. Understanding ebXML: Untangling the business Web of the future. Interneto prieiga: <http://www-128.ibm.com/developerworks/xml/library/x-ebxml/>. Žiūrėta 2006-10-14.
14. Naujokaitis G, Paradauskas B. Verslo transakcijų atvaizdavimas duomenų bazės transakcijomis. Informacinės technologijos '2007, 2007.
15. Nemuraitė L. Elektroninių verslo procesų modeliavimo metodų tendencijos. Informacijos mokslai, 2003, T. 21. pp. 77-88.
16. Nemuraitė L., Paradauskas B., Salelionis L. Extended Communicative Action Loop for Integration of New Functional Requirements. Information technology and control, Technologija, 2002, No. 24, pp. 18 - 26.

## Terminai ir santrumpos

ACID – Transakcijų valdymo principai, užtikrinantys patikimą vykdymą ir klaidingų situacijų valdymą

B2B – Verslas verslui (angl. *business-to-business*).

BPEL – Verslo procesų modeliavimo ir vykdymo kalba (angl. *Business Process Execution Language*)

CASE – Kompiuterizuotas sistemų kūrimas (angl. *Computer Aided System Engineering*)

DB – Duomenų bazė

ebXML – XML pagrindu sukurta standartų šeima, skirta suteikti plėtoti verslo procesus elektroninėje erdvėje

ebXML RIM – ebXML numatytas verslo paslaugų katalogo standartas

GEF – „Eclipse“ platformos priemonės, skirtos kurti grafinius redaktorių ir kitą programinę įrangą (angl. *Graphical Editing Framework*)

IS – Informacijos sistema

MVC – Programinės įrangos architektūra, atskirianti modelio objektus nuo jų grafinės reprezentacijos (angl. *Model-View-Controller*)

SOA – Paslaugomis pagrįsta architektūra (angl. *Service Oriented Architecture*)

SOAP – protokolas, skirtas žiniatinkliu perduoti XML duomenimis

UML – programinės įrangos modeliavimo ir projektavimo kalba (angl. *Unified Modeling Language*)

UMM – UN/CEFACT sukurta modeliavimo metodologija

UN/CEFACT – Jungtinių tautų prekybos skatinimo ir elektroninio verslo centras

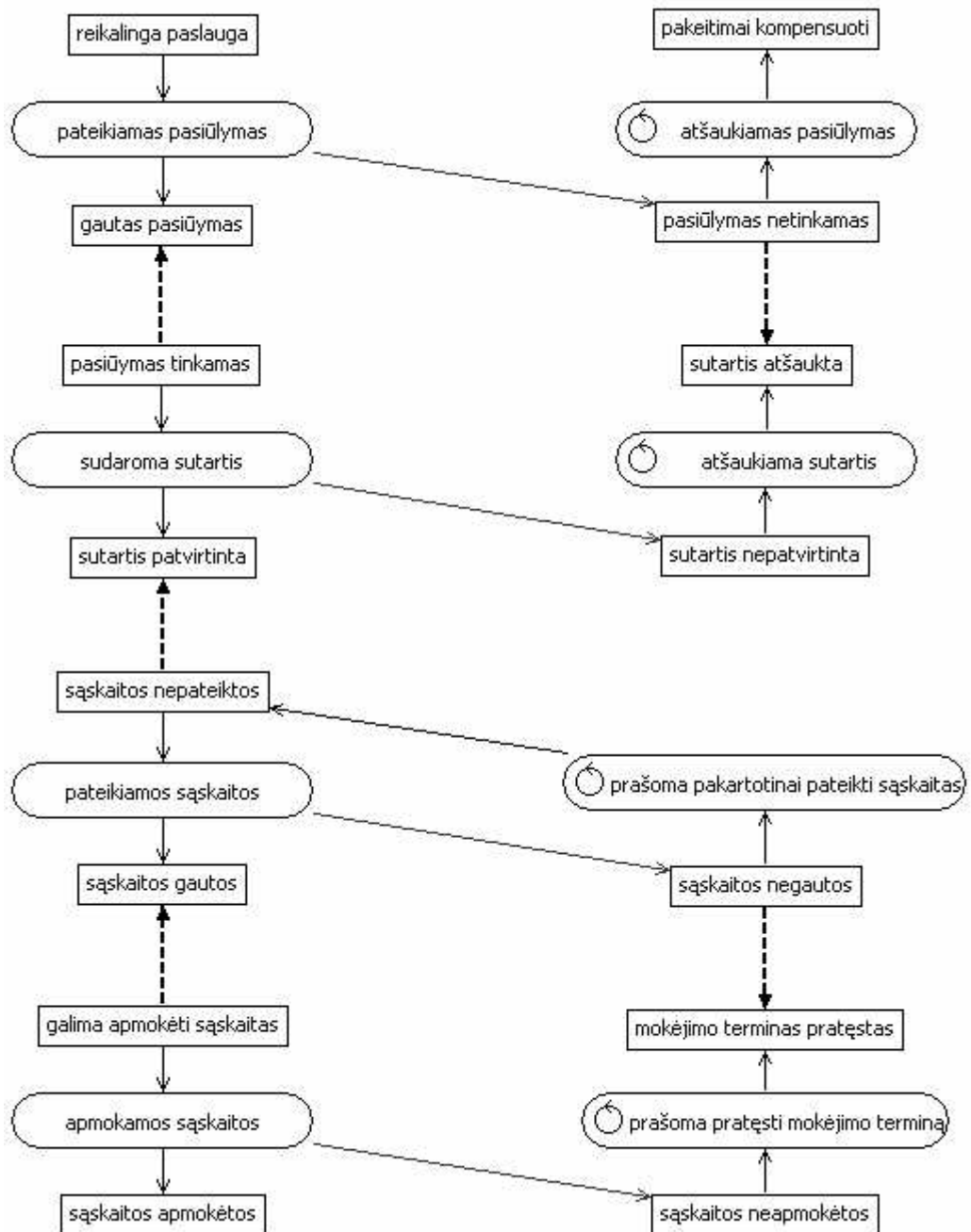
WSDL – Žiniatinklio paslaugų aprašų standartas

XML – bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba

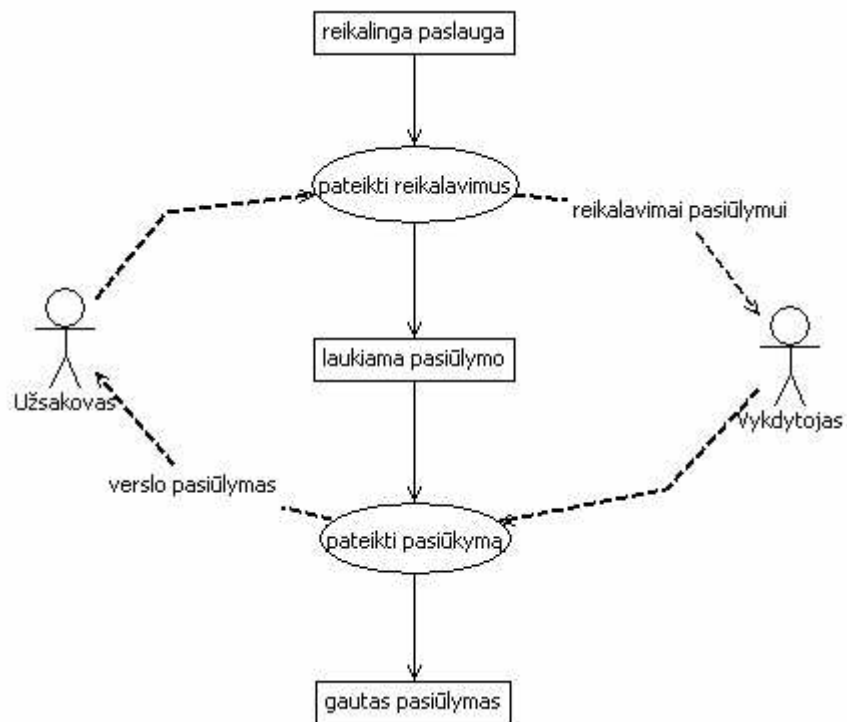
XSD – XML dokumentų struktūra aprašanti kalba

# 1 Priedas. Verslo proceso modelis

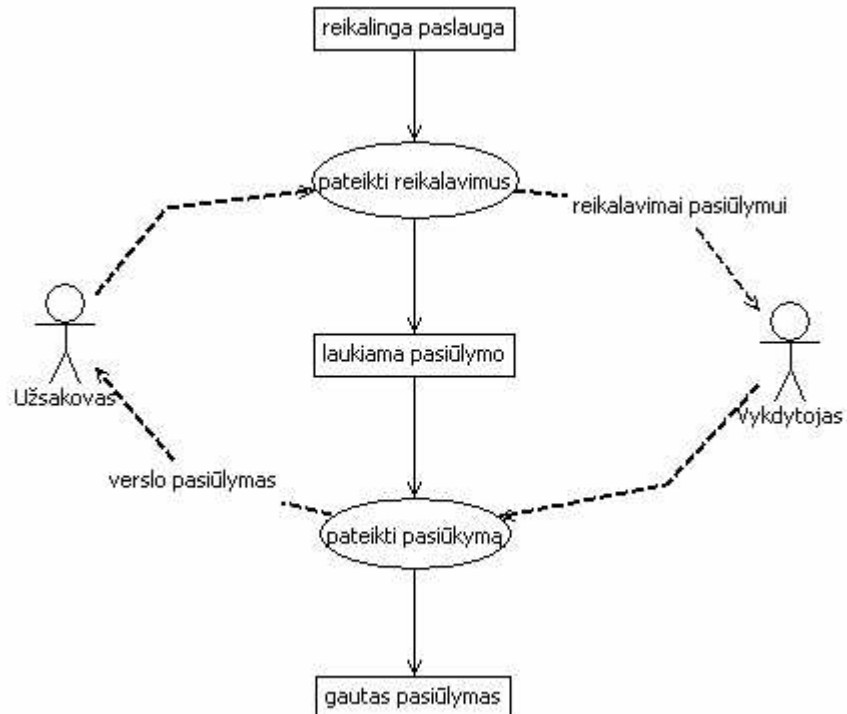
## 1. Verslo proceso diagrama



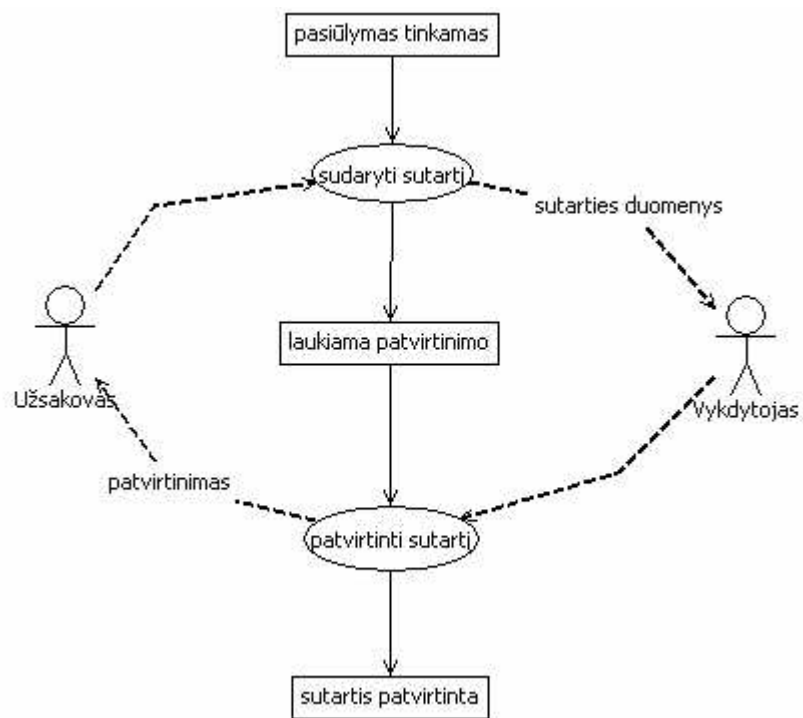
## 2. Etapo „pateikiamas pasiūlymas“ diagrama



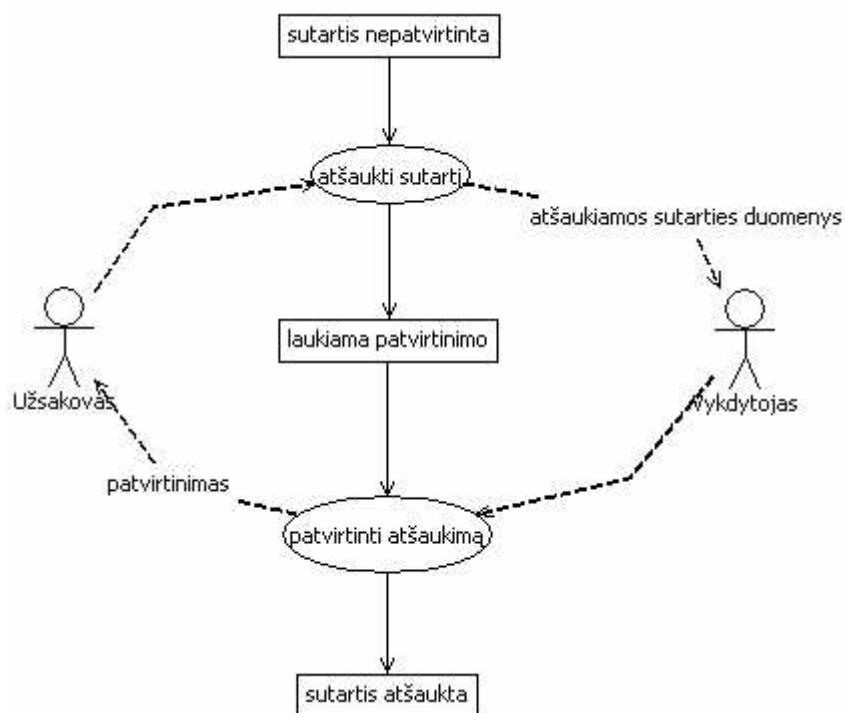
## 3. Kompensacinio veiksmo „atšaukiamas pasiūlymas“ diagrama



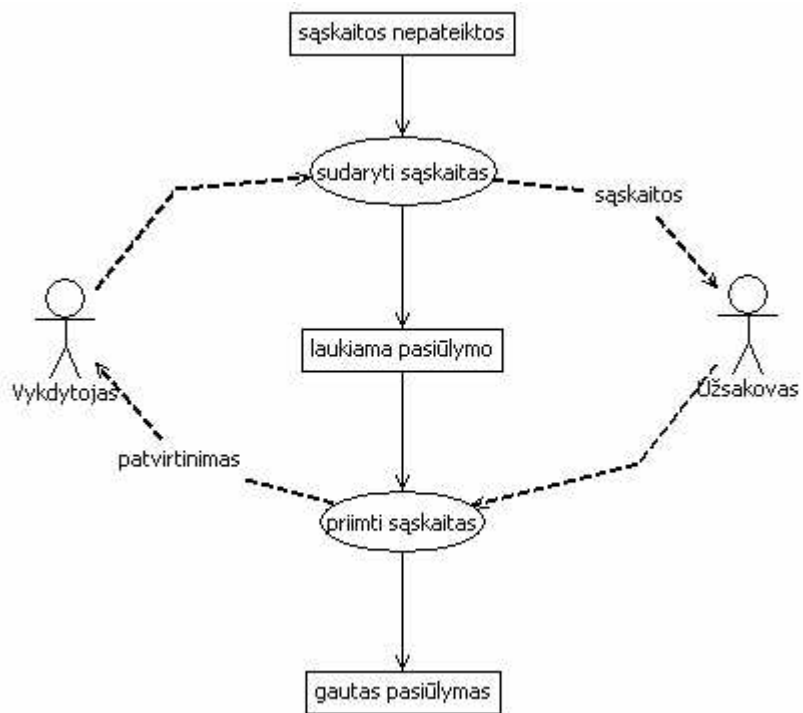
#### 4. Etapo „sudaroma sutartis“ diagrama



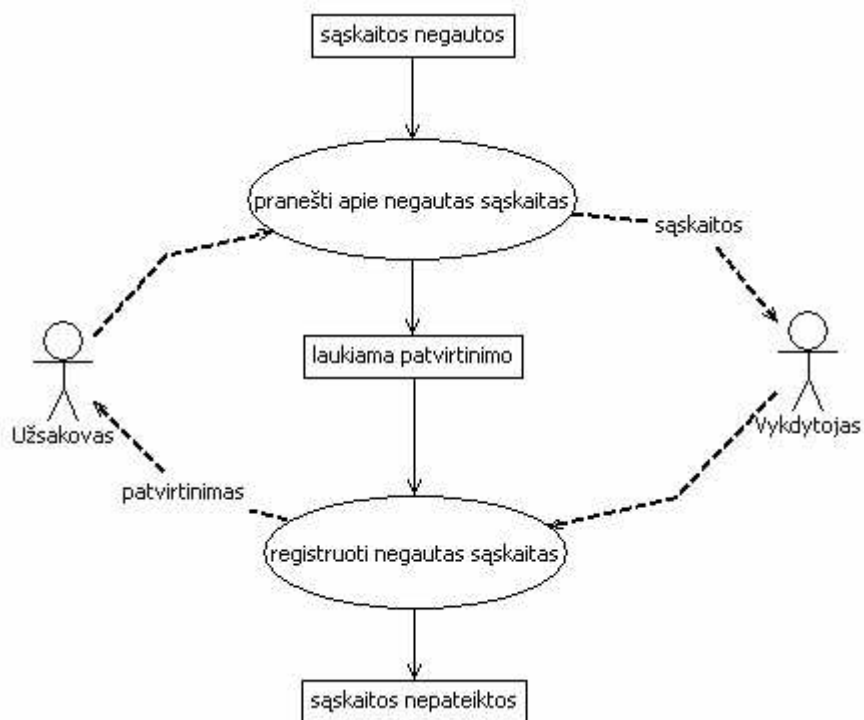
#### 5. Kompensacinio veiksmo „atšaukiama sutartis“ diagrama



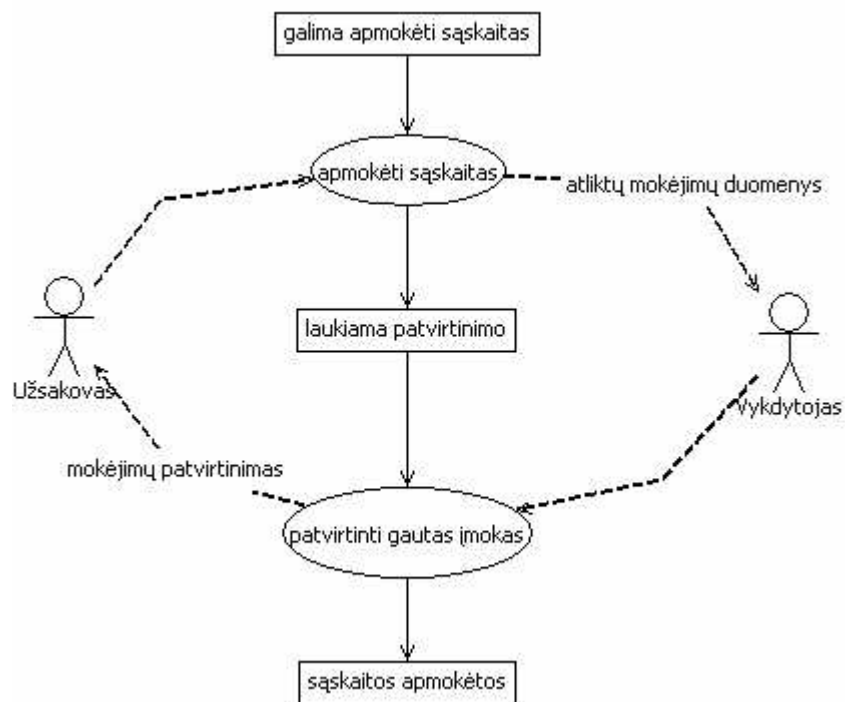
## 6. Etapo „pateikiamos sąskaitos“ diagrama



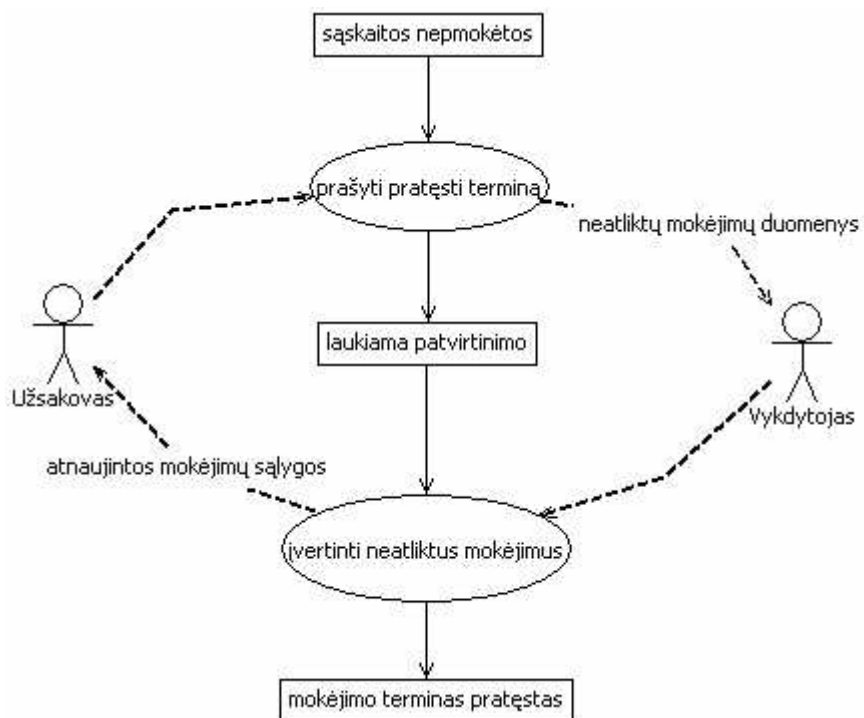
## 7. Kompensacinio veiksmo „prašoma pakartotinai pateikti sąskaitas“ diagrama



## 8. Etapo „apmokamos sąskaitos“ diagrama



## 9. Kompensacinio veiksmo „prašoma pratęsti mokėjimo terminą“ diagrama





## 2 Priedas. Straipsnis. Ilgalaikių verslo B2B transakcijų modeliavimas

### Ilgalaikių verslo B2B transakcijų modeliavimas

Mykolas Medžionis, Bronius Paradauskas

*Kauno technologijos universitetas, Informacijos sistemų katedra*

*Studentų 50, LT-51368 Kaunas, Lietuva*

Šiame darbe apžvelgiamos priemonės ilgalaikėms verslo transakcijoms modeliuoti ir projektuoti su jomis susietas duomenų bazių transakcijas. Įvertinamos galimybės modeliuoti ilgalaikių transakcijų dalyvius, dalyvių veiksmus, būsenas ir duomenų srautus. Parenkama metodika ilgalaikių verslo transakcijų valdymui ir kompensavimui. Sudarytas verslo procesų metamodelis, kuriuo paremtas projektuojamas B2B transakcijų redaktorius.

#### 1. Įžanga

Elektroninės verslo transakcijos vis dažniau keičia tradicinius verslo procesus. Šie pokyčiai vyksta siekiant pasiūlyti geresnes ir patogesnes paslaugas klientams ir partneriams. Siekiama mažinti procesų kaštus, juos darant efektyvesnius ir mažiau imlius daug laiko reikalaujančiam žmogaus darbui.

Pirmieji duomenų mainai tarp skirtingų verslo partnerių taikomųjų programų pradėti vykdyti 1960 – 1970 m. Nuo to laiko svarbiausi pokyčiai elektroninio verslo sferoje įvyko tik atsiradus XML technologijoms. XML sudarė geras galimybes integruoti duomenis. Tačiau paaiškėjo, kad duomenų integravimas nesukuria tokios naudos, kokios buvo tikėtasi. Atsirado naujas požiūris į integravimą – be duomenų pasiūlyta integruoti ir verslo procesus. Kartu su elektroninio verslo plėtra iškilio verslo procesų modeliavimo metodologijų ir jomis paremtų CASE priemonių poreikis.

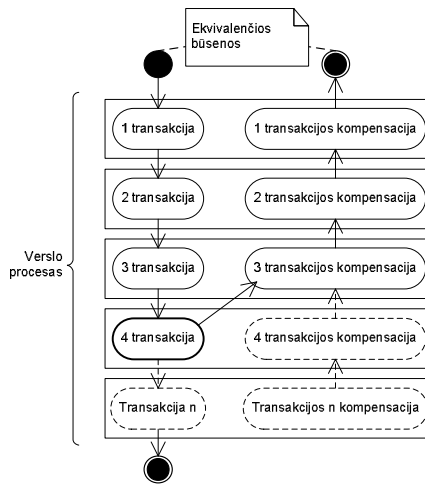
#### 2. Ilgalaikių transakcijų valdymas

Verslo transakcija (B2B transakcija) apibrėžiama kaip verslo proceso dalis, vykdoma dviejų arba daugiau partnerių ir sugeneruojanti suskaičiuojamą galinę būseną (sėkmė arba nesėkmė) [7]. Paskirstytiems verslo procesams būdingos ilgos trukmės (lyginant jas su duomenų bazių transakcijų trukmėmis), jie realizuojami, įvykdant daugelį DB transakcijų ir kaupiant duomenų srautus paskirstytosiose DB.

Vykdam B2B verslo transakcijas kaip duomenų bazių transakcijų seką, nuosekliai kinta verslo partnerių būsenos. Esant tam tikroms aplinkybėms, pvz., nutrūkus transakcijos ryšiui arba verslo aktoriui nusprendus atšaukti savo veiksmus, reikalingas mechanizmas kompensuoti B2B procesą ir su juo susietas duomenų bazių transakcijas. Būtinasis sėkmingo kompensavimo požymis – verslo partnerių būsenos sutampa su prieš kompensuojamą B2B transakciją buvusiomis būsenomis arba yra joms ekvivalenčios.

Kadangi verslo transakcijos yra daugiapakopės, o jų trukmės yra ilgos, taikyti tradicinį atominį transakcijų valdymą yra neefektyvu dėl ilgalaikio resursų blokavimo. Ilgalaikius verslo procesus SAGA modelyje siūloma skaidyti į trumpų transakcijų, kurioms taikyti atomiškumo principus tampa tikslinga, sekas [6]. Kiekvienam sekos etapui yra sukuriamas kompensacinis veiksmas. Kompensacinių veiksmų sekų įvedimas leidžia:

1. nutraukti verslo procesą, to pageidaujant verslo partneriams;
2. valdyti ilgalaikės B2B transakcijos etapuose iškylančias kritines situacijas: dėl techninių priežasčių nutraukus duomenų bazės transakciją, galima atšaukti tik paskutinius etapus, kuriuose įvyko klaida ir juos pakartotinai įvykdyti.



**1 pav. SAGA modelis**

Remiantis SAGA modeliu, kompensuojantys veiksmai atliekami šia tvarka (1 pav.):

1. Nutraukiamas vykdomas verslo proceso etapas (paveiksle – 4 transakcija). Kadangi verslo procesas yra išskaidytas į etapus, kuriems galioja ACID principai, veiksmas atšaukiamas panaudojant DB atstatymo priemones (*rollback*), todėl nutraukiamai DB transakcijai nėra vykdomas kompensacijos veiksmas.
2. Nuosekliai vykdomi kompensaciniai veiksmai (3 transakcijos kompensacija ÷ 1 transakcijos kompensacija).

Vykdam kompensaciją duomenų bazės lygmenyje, t.y. tiesiogiai atšaukiant pakeitimus duomenų bazėje ir neatliekant papildomos patikros, iškyla pavojus pažeisti verslo proceso logiką: verslo transakcijos įvykdymas elektroninėje erdvėje gali inicijuoti neatšaukiamus procesus, kurių vykdomi pakeitimai nebūtinai yra apriboti elektronine erdve. (Pvz., išsiunčiamas kroviny s užsakovui.). Atsiranda pavojus kilti situacijoms, kai atstačius duomenų bazės būseną, verslo dalyvio būseną nesutaps su buvusiaja prieš pradedant vykdyti verslo procesą ar proceso etapą. Tokiais atvejais bus pažeidžiamas būtinas korektiško kompensavimo požymis. Analogiškai pagrindiniam verslo procesui, kompensuojantys veiksmai nėra apribojami duomenų bazės atstatymu – tai gali būti naujos transakcijos ar fizinė veikla.

SAGA modelio trūkumas yra modelio bendrumas. Šiuo modeliu neapibrėžiamas duomenų bazių transakcijų vykdymas ir kompensacinių veiksmų generavimas. Be to, nėra įvertinami atvejai, kai verslo proceso etapas įvykdomas tik iš dalies.

Darbe [9] transakcijų kompensavimui siūloma panaudoti registrą (1 lentelė), kuriame būtų saugoma DB pakeitimų informacija. Pagal registre esančią informaciją prireikus gali būti automatiškai sugeneruojamos kompensuojančios duomenų bazės transakcijos.

**1 lentelė. Transakcijų registras**

register
id
business_process_id
state_before
modified_table
modification_type
parameters_before
parameters_after
restriction

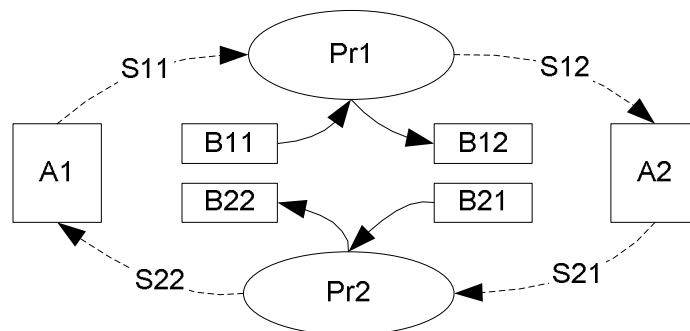
Dėl aukščiau išvardintų priežasčių, kai tiesioginis DB pakeitimų atstatymas gali sukelti neapibrėžtas ar kritines situacijas, automatinio kompensuojančių užklausų generavimo gali nepakakti. Korektiškumo aspektu SAGA modelis yra pranašesnis už straipsnyje [9] siūlomą automatinį užklausų generavimą, nes sistemos architektas gali įtraukti papildomas, su duomenų bazėse esančios informacijos atstatymu nesusietas, operacijas. Taikyti tik automatinį užklausų generavimą yra paranku jei kuriama sistema nėra sudėtinga, ir jos veiksmus galima trivaliai kompensuoti nepažeidžiant verslo logikos.

### 3. Komunikacinės veiksmų kilpos

Nors yra daug projektavimo įrankių – nuo paprastų schemų redaktorių iki galingų integruotų paketų, šiuo metu trūksta specializuotų priemonių B2B transakcijoms modeliuoti. Įprasta verslo procesus modeliuoti tradicinėmis priemonėmis (UML) [1,2,5], tačiau jų panaudojimas yra ribotas [1].

Darbe [8] pasiūlyta verslo transakcijas modeliuoti komunikacinių veiksmų kilpomis, kurios sudaro organizacijos modeliavimo EM (angl. Enterprise Modeling) metodo [4] pagrindą. EM metodo idėja yra ta, kad organizacijos modeliui specifikuojami ne tik veiklos veiksmai, bet ir pranešimų srautai.

2 pav. pateikta komunikacinė (binarinė) veiksmų kilpa sudaryta iš dviejų aktorių – kliento (A1) ir atlikėjo (A2), pranešimų srautų S11÷S22, ir dviejų veiksmų – kliento veiksmo (Pr1) ir atlikėjo veiksmo (Pr2). Dinaminiai situacijos pasikeitimai prieš ir po veiksmų pateikiami kaip perėjimai nuo būsenų B11, B21 į būsenas B12, B22 ir aibė pragmatinių bei semantinių priklausomybių tarp jų.



2 pav. Komunikacinė veiksmų kilpa

Tam tikros srities versle vykdomos transakcijos yra tarpusavyje susijusios. Kiekviena transakcija vienaip ar kitaip keičia verslo santykių būseną tarp bendradarbiaujančių partnerių. Verslo transakcijų metu bendradarbiaujančios šalys keičiasi verslo informacija (pranešimais), kuri vis papildo vykstantį verslo procesą ir veda jį į užbaigimo būseną, atitinkančią verslo partnerių siekiamus tikslus.

Bendru atveju verslo transakcijos veiksmas turi „prieš“ ir „po“ būseną. Binarinė kilpa susideda iš dviejų veiksmų, kurių kiekvienas turi „prieš“ ir „po“ būsenas (iš viso 4 būsenos). Pirmojo transakcijos veiksmo „po“ būseną gali sutapti su antrojo veiksmo „prieš“ būseną. Būsenomis galima išreikšti aktorių tikslus ir nustatyti, ar komunikacinė kilpa yra gyvybinga. Pagrindinį gyvybingumo kriterijų galima išreikšti taip: jei aktorius A2 galimybė neigiamai veikia aktorius A1 problemą, tuomet A2 galimybė teigiamai veikia A1 tikslą (patekti į būseną B22) ir A1 tikslas neigiamai veikia A2 problemą (A2 padeda A1 pasiekti tikslą – būseną S22, A1 sumažina arba panaikina A2 problemą).

Komunikacinėmis kilpomis galima modeliuoti situacijas, kai verslo transakcija įvykdoma tik iš dalies, t.y. srautas S22 yra tik dalis srauto, kurio sulaukti tikisi aktorius A1.

### 4. Duomenų srautai

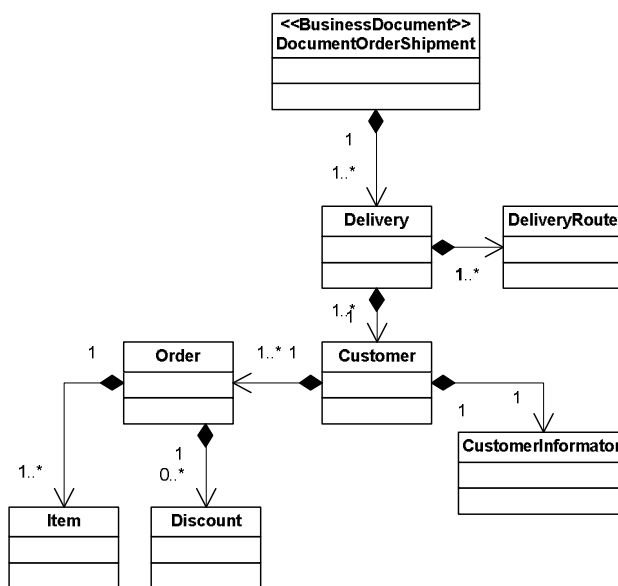
Komunikacinių veiksmų kilpose duomenų srautų parametrai šaltinyje [9] yra gaunami perduodamą verslo informaciją skaidant iki žemiausio lygmens. Šių parametru reikšmių vektoriai sudaro tarp verslo partnerių siunčiamus duomenų srautus. Tačiau, dekomponavus verslo informaciją iki žemiausio lygmens parametru rinkinių, gali būti prarandama objektų tarpusavio ryšiais nusakoma informacija.

Šaltiniuose [1,2,3] išskiriami šie verslo proceso analizės etapai:

1. Verslo lygmuo (business layer).
2. Specifikacijų lygmuo (specification layer)

Verslo lygmuo yra skirtas bendrai verslo transakcijų apžvalgai. Čia globalia klasių diagrama specifikuojama verslo procesų apdorojama informacija [1]. Pažymėtina, jog šiame analizės lygmenyje nėra pranešimo sąvokos [2].

Specifikacijų lygmenyje duomenų srautai yra aprašomi atskiromis klasių diagramomis [3]. Kiekviena iš šių klasių diagramų yra atskiras globalios klasių diagramos atvaizdas. Be to, pranešimai gali būti susieti su verslo taisyklių rinkiniu, aprašytu formalia arba neformalia kalba. Kiekvienas pranešimas yra aprašomas atskira klasių diagrama. Svarbus šios diagramos akcentas – pranešimo struktūra yra medžio formos (3 pav.). Medžio šaknis klasių diagramoje yra klasė, kurios stereotipas – <<BusinessDocument>>. Modeliuojant B2B transakcijas duomenų srautas yra vienareikšmiškai nurodomas medžio šaknimi. Likusios pranešimą sudarančios klasės yra agregacijos ryšiais susietos su šaknimi. Medyje gali būti ne daugiau nei viena, šakninė, <<BusinessDocument>> stereotipo klasė.



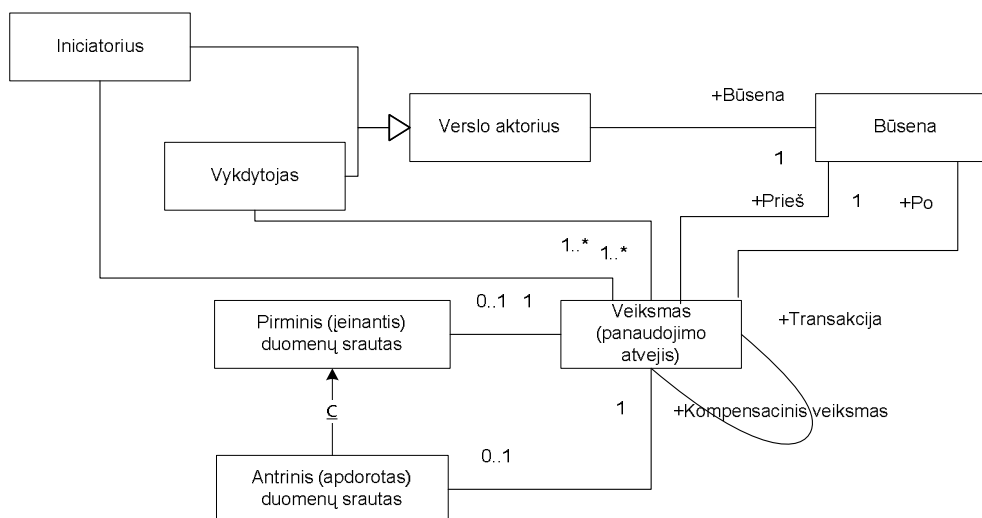
3 pav. Medžio struktūros duomenų srautas

Medžio tipo struktūra suteikia šiuos privalumus:

1. struktūra yra nesudėtingai aprašoma XML schemomis;
2. medžio struktūrą yra patogu realizuoti reliacinėje duomenų bazėje;
3. medžio šaknis vienareikšmiškai identifikuoja duomenų srauto (pranešimo) tipą.

## 5. Metamodelis

Ištyrus galimas B2B transakcijų modeliavimo metodikas ir įvertinus jų privalumus bei trūkumus, pasiūlytas ilgalaikių transakcijų metamodelis (4 pav.), kuriuo grindžiamas projektuojamas B2B transakcijų redaktorius.



4 pav. B2B transakcijų metamodelis

Metamodelio elementai:

1. *Verslo aktorius*. Verslo aktorius yra vienas iš dviejų ilgalaikės transakcijos partnerių. Galimi du subtipai: Iniciatorius ir Vykdytojas.
2. *Būsena*. Kiekvienas verslo partneris yra tam tikroje būsenoje. Vykdamas verslo procesus, verslo aktorius gali keisti savo būseną į kitas, kurios yra jam galimų būsenų aibėje.
3. *Veiksmas*. Verslo procesas vykdomas partneriams atliekant veiksmus. Atskiri veiksmai gali būti suprantami kaip panaudojimo atvejai bei turi „prieš“ ir „po“ būsenas. Kiekvienas veiksmas turi kompensacinį veiksmą.

4. *Duomenų srautai*. Duomenų srautai yra medžio pavidalo duomenų struktūros, vienareikšmiškai nusakomos atitinkamo medžio šaknimi. (žr. 4 skyrių).

Šis metamodelis vienija bei praplečia SAGA ir komunikacinių kilpų metodologijų privalumus. Sukūrus metamodeliu paremtą grafinį redaktorių, bus sudarytos galimybės projektuoti:

- verslo aktorius;
- aktorių veiksmus ir aktorių būsenų kaitą vykdant verslo procesą;
- aktorių tikslus, išreikiamus jų būsenomis;
- duomenų srautus, apibrėžti jų tarpusavio sąryšius ir išplėsti duomenų struktūras;
- kompensacinius veiksmus.

## 6. Išvados

1. Siekiant užtikrinti patikimą verslo informacinių sistemų funkcionavimą, būtina formaliai aprašyti verslo transakcijas. Projektuojant verslo valdymo sistemas, verslo transakcijoms specifikuoti tikslinga naudoti komunikacines veiksmų kilpas.
2. Projektuojant ilgalaikes B2B transakcijas, būtina specifikuoti kompensacinius veiksmus verslo proceso etapams. Šiam tikslui yra parankus SAGA modelis.
3. Verslo proceso duomenų srautus yra patogu projektuoti medžio struktūros klasių diagramomis.
4. Darbe pasiūlytas duomenų metamodelis kuriamam verslo B2B transakcijų redaktoriui. Šiuo metamodeliu suvienijamos bei praplečiamos SAGA ir komunikacinių kilpų metodologijos ir sudaromas pagrindas įvairiapusiškai modeliuoti ilgalaikius verslo procesus.

## Literatūros sąrašas

- [1] **Brimont P., Dubois E., Eshuis R., Gregoire B., Ramel S.** Animating ebXML Transactions with a Workflow Engine. *Interneto prieiga: <http://www.cs.univie.ac.at/upload//shared/mis/publications/IEEE-CEC06-final.pdf>*, 2006.
- [2] **Brimont P., Dubois E., Eshuis R., Gregoire B., Ramel S.** EFFICIENT: A Tool Set for Supporting the Modelling and Validation of ebXML Transactions. *Joint 9th European Software Engineering Conference (ESEC) & 11th SIGSOFT Symposium on the Foundations of Software Engineering (FSE-11)*, ACM Press. 2003, pp. 359-362.
- [3] **Dubois E., Incoul Ch., Schmitt M.** Supporting Business Experts In the Design Of B2B Transactions Through Interactive Process Simulation. *International Workshop on Enterprise and Networked Enterprises Interoperability (ENEI'2005)*, 2005.
- [4] **Gustas R.** Towards a Communication Based Approach for Enterprise Modelling Integration. *International Conference on Integration and Reuse*, 2000.
- [5] **Hofreiter B., Huemer Ch., Zapletal M.** Registering UMM Business Collaboration Models in an ebXML Registry. *Interneto prieiga: <http://www.cs.univie.ac.at/upload//shared/mis/publications/IEEE-CEC06-final.pdf>*, 2006.
- [6] **Young Ch.**, BizTalk Server 2006: The Compensation Model. *Interneto prieiga: <http://geekswithblogs.net/cyoung/articles/100424.aspx>*, 2006.
- [7] **Nemuraitė L.** Elektroninių verslo procesų modeliavimo metodų tendencijos. *Interneto prieiga: <http://www.leidykla.vu.lt/inetleid/inf-mok/21/str7.html>*, 2002.
- [8] **Nemuraitė L., Paradauskas B., Salelionis L.** Extended Communicative Action Loop for Integration of New Functional Requirements. *Information technology and control, Technologija*, 2002, No. 2(23), pp. 18 - 26.
- [9] **Naujokaitis G., Paradauskas B.** Verslo transakcijų atvaizdavimas duomenų bazės transakcijomis. *Informacinės technologijos '2007*, 2007.

## MODELLING OF LONG-RUNNING BUSINESS TRANSACTIONS

In this paper tools for modelling long-running B2B transactions and related database transactions are overviewed. Measures are estimated for defining actors and data flows of business process together with actions and states of business partners. Method for management and compensation of long-running transactions is chosen. Finally, metamodel of long-running transactions is proposed, which a projected CASE tool is based upon.