

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

VERSLO INFORMATIKOS KATEDRA

Lina Slaninaite

**Verslo procesų, aprašytų BPMN notacija,
transformavimas į PLA modelius**

Magistro darbas

Darbo vadovas

prof., habil. dr. H. Pranevičius

Kaunas, 2006

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

VERSLO INFORMATIKOS KATEDRA

Lina Slaninaitė

**Verslo procesų, aprašytų BPMN notacija,
transformavimas į PLA modelius**

Magistro darbas

Kalbos konsultantė

2006-05-15

Lietuvių k. katedros dėst.
I. Mickienė

Vadovas

prof., habil. dr. H. Pranevičius
2006-05-29

Recenzentas

2006-05-26

Atliko

doc. E. Karčiauskas

2006-05-29

IF M-0/1 gr. stud.
Lina Slaninaitė

Kaunas, 2006

TURINYS

1. ĮVADAS	4
2. VERSLO PROCESŲ (BUSINESS PROCESS) MODELIAVIMO KONCEPCIJA	6
2.1. Verslo procesų modeliavimo (Business Process Modeling) sąvokos.....	6
2.1.1. Pagrindinės verslo procesų sąvokos.....	6
2.1.2. Proceso dalyviai	8
2.1.3. Dalyvių vaidmenys	8
2.1.4. Resursai	9
2.1.5. Vidinė struktūra	10
2.1.6. Išorinė struktūra	10
2.2. Verslo procesų modeliavimas.....	10
2.3. Verslo procesų valdymas	12
2.4. Verslo procesų re-inžinerijos samprata.....	14
3. VERSLO PROCESŲ MODELIAVIMO KALBŲ APRAŠYMAS	17
3.1. BPMN standartas verslo procesams aprašyti	17
3.1.1. BPMN (<i>Business Process Modeling Notation</i>) apžvalga.....	17
3.1.2. BPMN grafiniai žymėjimai	19
3.1.2.1. Įvykiai (<i>Events</i>).....	20
3.1.2.2. Veiklos (<i>Activities</i>).....	21
3.1.2.3. Vartų elementas (<i>Gateway</i>)	22
3.1.2.4. Konteineriai (<i>Pools</i>) ir juostos (<i>Lanes</i>)	22
3.1.2.5. Grafiniai sujungimo objektai	23
3.2. Verslo procesų modeliavimas, naudojant UML diagramas	24
3.2.1. Verslo panaudojimo atvejų modelis.....	26
3.2.2. Verslo objektų modelis.....	29
4. VERSLO PROCESŲ, APRAŠYTŲ BPMN NOTACIJA, TRANSFORMAVIMO Į PLA MODELIOUS METODIKA	31
4.1. BPMN notacijos parinkimas modeliuojant verslo procesus	31
4.2. Agregatinės specifikacijos aprašymas	33
4.2.1. Formalių specifikacijų metodai	33
4.2.2. Formalios specifikacijos apibrėžimas, savybės	33
4.2.3. Verslo procesų formalizavimas, naudojant PLA modelį	34
4.3. Verslo procesų transformavimas į PLA modelius	36
4.3.1. Modelių transformavimas.....	36
4.3.2. BPMN ir agregato metamodeliai	38

4.3.3. Transformavimo metodika	42
4.3.3.1. Transformavimo apibendrinimas	63
5. IŠVADOS	65
7. SUMMARY	66
6. LITERATŪRA.....	67
8. PRIEDAI.....	69
8.1. Verslo proceso pinigų išėmimo iš bankomato agregatinė specifikacija	69

1. ĮVADAS

Darbo tikslas – sudaryti verslo procesų, aprašytų BPMN (*Business Process Modeling Notation*) notacija, transformavimo į PLA modelius metodiką, kuri leistų transformuoti verslo procesus, aprašytus BPMN modeliu, į PLA modelį.

Verslo procesų modeliavimas aktualus ne tik programinės įrangos kūrėjams, bet ir užsakovams. Aplinkoje, kurioje keičiasi tiek technologijos, tiek rinka, tiek patys dalyviai, verslo procesų modeliavimas, ne tik padeda išsaugoti organizacijai svarbias žinias ir palengvinti komunikaciją tarp verslo analitikų bei IT specialistų, bet ir sumažina riziką, padeda taupyti laiką, užtikrina tikslų užduočių įvykdymą.

Pastaruoju metu didelis dėmesys skiriamas verslo procesų modeliavimui, naudojant BPMN standartą. Kadangi įvairiose aplinkose skiriasi verslo procesų aprašymas, atsiranda poreikis standartizuoti skirtingus verslo procesus naudojant vieningą notacijos sistemą.

BPMN yra naujas standartas, skirtas modeliuoti verslo procesų srautus ir web servisų procesus, išleistas BPMI (*Business Process Management Initiative*) standartų organizacijos [1]. BPMN standarto pagrindinis tikslas – pateikti visiems verslo dalyviams lengvai suprantamą verslo procesų modeliavimo notaciją. Kitas svarbus tikslas – užtikrinti, kad vykdomosios verslo procesų kalbos XML pagrindu, pavyzdžiui, BPEL4WS (*Business Process Execution Language For Web Services*) ir BPML (*Business Process Modeling Language*), būtų lengvai pavaizduojama bendra notacija.

BPMN apima ne vieną modeliavimo būdą ir leidžia kurti tiesioginius verslo procesus. BPMN modelis gali būti trijų tipų:

- Privatūs (vidiniai) verslo procesai – vidiniai organizacijos procesai.
- Abstraktūs (vieši) procesai – procesai, kurie vaizduoja sąveikavimą tarp vidinių verslo procesų ir kitų procesų ar dalyvių.
- Bendradarbiavimo (globalūs) procesai – procesai, kurie vaizduoja sąveikavimą tarp dviejų ar daugiau verslo objektų.

BPMI veikla nukreipta verslo procesams gerinti. Ši organizacija suvienijo ankstesnius skirtingus procesų modeliavimo, vaizdavimo metodus, organizacijos taikomųjų programų integravimą, verslas-verslui (*business-to-business*) integravimą į bendrą standartą. Veiklos procesai modeliuojami daugelį metų. Tačiau tie metodai ir priemonės buvo tik iš dalies sėkmingi arba visiškai nesėkmingi, nes trūko standartų ir pilnų galimybių, apimančių verslo procesų modeliavimą, projektavimą ir vykdymą. Norint suprasti procesų architektūrą, modeliavimą, paskirstymą, reikalingi verslo modeliavimo ir verslo vykdomosios kalbos standartai.

BPMI standartų organizacija plėtoja tris standartus:

- BPMN – verslo procesų modeliavimo standartą.

- Verslo proceso modeliavimo kalbos standartą (BPML) – verslo vykdomųjų kalbų standartą.
- Verslo proceso užklausų kalbos standartą (BPQL) – kaip el. verslo procesų paskirstymo ir vykdymo vartotojo sąsajos standartą.

Svarbi BPMI standartų savybė yra ta, kad jie buvo sukurti remiantis matematiniais pagrindais – Pi-Calculus. Tai formalių skaičiavimų metodas, leidžiantis apibrėžti dinامينius ir mobilius procesus. Jo privalumai:

- Pi-Calculus leidžia tikrinti nuoseklumą, logiškumą.
- Pi-Calculus užtikrina aklaviečių ir klaidų suradimą.
- Pi-Calculus leidžia optimizuoti procesus.

Modeliuojant verslo procesus BPMN turi nemažai privalumų [2], lyginant su UML (*Unified Modeling Language*). Pirma, ji leidžia aprašyti procesus srautais, tai lengvai suprantama verslo analitikui, ir turi išsamią tipinių modelių elementų aibę. Antra, dėl matematinio pagrindo ji lengvai atvaizduojama verslo vykdomosiomis kalbomis.

Yra daug kalbama apie verslo procesų modeliavimą, naudojant BPMN standartą, tačiau mažai yra kalbama apie verslo procesų, aprašytų BPMN notacija, transformavimą į formalizuotą aprašą. Tad šio darbo pagrindinis tikslas ir yra išsiaiškinti, ar galima verslo procesus, aprašytus BPMN notacija, transformuoti į formalizuotą aprašą ir kokie BPMN modelio elementai yra transformuojami į kokius PLA modelio elementus.

Šiame darbe analizuojant verslo procesus, diagramos buvo braižomos naudojant Visio 2003 ir MagicDraw 10.5 paketus.

Šis magistro darbas susideda iš trijų dalių. Pirmojoje dalyje (antrajame skyriuje) yra surinktos ir susistemintos pagrindinės verslo procesų modeliavimo sąvokos, kurios leidžia kalbėti apie struktūrizuotą verslo procesą. Antrojoje dalyje (trečiajame skyriuje) yra nagrinėjamos verslo procesų modeliavimo kalbos (BPMN (*Business Process Modeling Notation*) ir UML (*Unified Modeling Language*)), kuriomis galima aprašyti verslo procesus. Trečiojoje dalyje (ketvirtajame skyriuje) yra pateikta metodika, kuri apibrėžia kaip galima transformuoti verslo procesus aprašytus BPMN modeliu, į PLA modelį. Transformavimo metodika pateikta, naudojantis pinigų išėmimo iš bankomato verslo procesu, kuriam buvo sudarytas BPMN modelis, po to šis modelis, naudojant modelių transformavimo teoriją, buvo transformuotas į PLA modelį. Transformavimo rezultatai buvo susisteminti ir buvo pateikti bendri reikalavimai, kurie apibrėžia BPMN modelio transformavimą į PLA modelį.

2. VERSLO PROCESŲ (BUSINESS PROCESS) MODELIAVIMO KONCEPCIJA

2.1. Verslo procesų modeliavimo (Business Process Modeling) sąvokos

Šio skyriaus tikslas yra surinkti ir susisteminti pagrindines verslo proceso sąvokas, kurios leis kalbėti apie struktūrizuotą verslo procesą. Sąvokų rinkinys nepriklauso nuo to, ar modeliavimo objektas (verslo procesas) jau egzistuoja tikrovėje (proceso analizė), ar ne (proceso projektavimas).

2.1.1. Pagrindinės verslo procesų sąvokos

Dauguma verslo procesų modeliavimo sąvokų yra kur nors jau apibrėžtos, pavyzdžiui BPR (*Business Process Reengineering* – verslo procesų re-inžinerijos) literatūroje [3]. Verslo procesų modeliavimas reikalauja sąvokų, kurios leistų apibūdinti verslo procesą nepriklausomai nuo to, dėl ko atsirado tokios informacijos poreikis.

Verslo procesai gali būti skirstomi į tokias rūšis:

- Organizacijos vidaus verslo procesai (produktų gamyba).
- Tarp organizacijų vykstantys verslo procesai. Tai yra vadinama verslas-verslui transakcijomis (*business to business*).
- Vadybiniai verslo procesai (*managerial*) (darbuotojų samdymo procesai).
- Operaciniai verslo procesai (*operational*) (serviso skambučiai, aptarnauti informacijos centre, banko kasininkų turima informacija reikalinga sąskaitos atidarymui ir t.t.).
- Į veiklą orientuoti verslo procesai: pokyčiai, kurie vyksta duomenų saugykloje.

Remiantis BPR literatūra, verslo procesas yra apibrėžiamas kaip rinkinys dalinai sutvarkytų veiklų, siekiant tikslo [3]. Tikslų pavyzdžiai gali būti tokie:

- Pinigų gavimas už parduotą prekę.
- Programinės įrangos kūrimas pagal specifikacijas.
- Paciento išrašymas iš ligoninės ir kt.

Greta pateiktų apibrėžimų, gali būti vartojamos ir tokios verslo proceso sąvokos [4]:

- Proceso **tikslas**.
- **Veikla**, dažnai vadinam **užduotimi**.
- **Laikas** – ašis, kuria remiasi dalinė tvarka. Laiką galime laikyti absoliučiu ar reliatyviu, diskrečiu ar tolydžiu, vidiniu ar išoriniu.

Tikslo sąvoka leidžia numanyti, ar daugelyje laiko intervalo momentų, kai procesas egzistuoja, proceso tikslas yra pasiektas, ar ne. Jei jis nėra pasiektas, reikia sužinoti kaip toli yra proceso tikslas.

Todėl yra reikalinga proceso **būsenos** sąvokos apibrėžimas. Būsena gali būti *galinė* (jei tikslas buvo pasiektas) arba *tarpinė* (kitu atveju).

Norint aptarti proceso būseną, nestebint viso proceso vystimosi realiame pasaulyje, yra reikalingas būdas, leidžiantis pavaizduoti proceso būseną popieriuje, kompiuteryje. Tai „veda“ link **būsenos atvaizdavimo** sąvokos.

Būsenos sąvoka gali padėti apibrėžti *veiklos* sąvoką. Veikla gali būti apibrėžiama kaip *veiksmas* (veiksmų rinkinys), skirtas tam tikru būdu pakeisti proceso būseną.

Veiklos apibrėžimas yra paremtas procesų būsenos **pokyčio** sąvoka. Toks pokytis gali įvykti, ne tik kaip veiklos pabaigos rezultatas, bet ir kitais, dažnai nenumatytais, atvejais. Pavyzdžiui, dalis programos kodo, parašyto programine įranga, gali būti prarasta sugedus diskui.

Laiko momentas, kai proceso *būsena* keičiasi, gali būti pavadintas **įvykiu** proceso gyvavimo cikle. Kiekviena baigta *veikla* yra *įvykio* rezultatas. Bet kaip buvo minėta anksčiau, įvykiai tai pat gali įvykti nenumatyta.

Visų įvykių seka susideda iš proceso **vidinio laiko** (niekas neįvyksta tarp įvykių). Proceso būsenų seka po kiekvieno įvykio formuoja proceso evoliucijos **istoriją**.

Pastaba. Du įvykiai negali įvykti tuo pačiu momentu, o yra atskirti vienas nuo kito laiko intervalais, kad ir kokie maži jie bebūtų. Ši prielaida neturi praktinių išvadų, bet ji supaprastina analizę.

Bet kuris verslas susideda iš daugelio procesų, vykstančių lygiagrečiai. Taigi dažnai esame suinteresuoti susieti kiekvieno proceso *vidinį laiką* su **išoriniu laiku**, kuris yra bendras visiems procesams. Tai gali būti įvykdyta, kai turimas kiekvieno įvykio įrašas, kuris užfiksuoja išorinį šio proceso laiką. Chronologinė įvykių seka, registruota proceso kadre, suformuoja **kroniką**, t.y. užrašo šio proceso istoriją.

Verslo proceso sąvoka, apie kurią buvo kalbėta anksčiau, remiasi konkrečiais procesais, kurie vyksta laike. Procesai, kurie turi panašų tikslą ir panašų elgsenos modelį (tos pačios rūšies veiklas), gali būti sujungti **proceso tipo** sąvoka.

Minėtosios sąvokos leidžia apibrėžti verslo procesą kaip dinaminę sistemą, kuri juda visų įmanomų procesų erdvėje, tol kol ji pasiekia galinę būseną (tikslą). Tokios sistemos sukelia **aktyvią** (*proactive*) ir **pasivytią** (*reactive*) elgseną. Elgsena yra *aktyvi*, kadangi kiekvienos veiklos vykdymas paprastai įvyksta ne kaip reakcija į signalą iš išorės, o dėl vidinės proceso evoliucijos logikos. Elgsena yra *pasyvi*, nes išoriniai įvykiai gali reaguoti į juos, pvz., žemės drebėjimo sugriautos sienos atstatymas.

Dėl aktyvios ir pasyvios elgsenos suliejimo, neįmanoma tiksliai apibrėžti proceso *tikslo*. Tikslas greičiau turėtų būti apibrėžtas kaip galinių būsenų rinkinys, kurios yra pasiekiamos ir kurios yra artimiausios turimai tarpinei būsenai.

2.1.2. Proceso dalyviai

Verslo proceso dalyviai gali būti suskirstyti į tris kategorijas: **artefaktus** (dirbtiniai objektai), žmogaus sąvoką ir organizacijos sąvoką.

Artefakto (dirbtinio objekto) sąvoka yra naudojama nusakyti fizinį ar abstraktų objektą, tokį kaip dokumentas, produktas, kompiuterio programa ir t.t. Tam tikro artefakto vaidmuo įvairiuose procesuose gali kisti nuo buvimo pasyviu objektu proceso metu, į aktyvų įrankį, naudojamą procese veiklų vykdymui. Artefaktas gali turėti sudėtinę struktūrą, pvz., artefaktas gali turėti kitus artefaktus, įtrauktus kaip jo sudedamoji dalis, arba jis gali turėti šiek tiek kitos rūšies ryšių su kitais artefaktais.

Žmogaus sąvoka apibūdina asmenį, dalyvaujantį procese. Asmens vaidmuo gali kisti nuo buvimo aktyviu dalyviu, vykdančiu veiklas, iki santykinai pasyviu susisiekimu partneriu ar netgi pasyviu objektu į kurį proceso veiklos yra nukreiptos, pvz., pacientas ligoninėje.

Dažnai žmonės dalyvauja verslo procese ne dėl savęs, bet dėl kai kurių **organizacijų**, tokių kaip kompanijos, politinės partijos, ministerijos, komandos. Žmonės, susiję su tam tikra organizacija ir dėl jos dalyvaujantys verslo procese, gali būti vadinami **nariais**. Nariai gali būti susiję su organizacija darbo ryšiais, naryste. Narystė gali būti pastovi arba laikina (pvz., darbo sutartis). Kiekvienas narys turi savo vaidmenį, susijusį su tam tikra verslo proceso rūšimi. Tas pats asmuo gali būti kelių organizacijų nariu. Dar daugiau, tas pats asmuo gali turėti skirtingus vaidmenis toje pačioje organizacijoje.

Organizacijos gali turėti ryšių tarp savęs, kurie tęsiasi daug ilgiau nei tam tikras verslo procesas. Šie ryšiai gali būti įvairių tipų, pavyzdžiui, nuosavybės ryšiai, bendradarbiavimo, pirkėjo, pardavėjo.

2.1.3. Dalyvių vaidmenys

Artefaktai, žmonės ir organizacijos tam tikruose verslo procesuose gali turėti įvairių vaidmenų. Patys svarbiausi vaidmenys yra apibrėžti žemiau.

Atsižvelgiant į tam tikrus procesus, organizaciją galima suskirstyti į dvi kategorijas, **proceso valdytoja** ir **išorinę organizaciją**. Valdytojo nariai tampa **personalu**, kuris užbaigia proceso sistemos veiklas. Jie yra atsakingi už planavimą ir veiklų vykdymą. Išorinių organizacijų nariai tampa **kontaktiniais asmenimis**, kurie dalyvauja įvairiose bendradarbiavimo veiklose.

Vidinis dalyvis, pavyzdžiui, personalas yra proceso varomoji jėga. Jie yra atsakingi už planavimą ir veiklų vykdymą. Kontaktiniai asmenys tarnauja tikrai kaip kolegų bendradarbiavimo veiklose, pvz., nusiunčia laišką asmeniui X, paskambina Y, ir t.t.

Vienas iš personalo narių gali būti paskirtas tam tikro proceso *vadovu*. Buvimas vadovu nereiškia, kad asmuo, kuris atlieka šį vaidmenį, turi išsamiai kontroliuoti procesą. Jis paprasčiausiai atsakingas už problemų, jei jos iškyla tam tikro proceso metu, sprendimą.

Proceso *subjektas* (tema) yra tai, į ką proceso veiklos yra nukreiptos. Subjektas gali būti artefaktas, žmogus (žmogiška būtybė), organizacija, ar visų proceso dalyvių kombinacija. Artefaktas kaip subjektas gali simbolizuoti produktą, kuris yra parduodamas, perkamas ar gaminamas. Organizacija gali tarnauti kaip proceso subjektas, galintis keisti jos struktūrą, kad paverstų organizacijos darbą efektyvesniu.

Proceso beneficiantas (naudos gavėjas) yra organizacija ar asmuo, dėl kurio interesų procesas yra vykdomas. Beneficiantas dažnai nurodomas kaip klientas. Beneficiantas gali būti išorinė organizacija (išorinis pirkėjas) arba proceso valdytojas (vidinis pirkėjas). Tam tikrame procese gali būti daugiau nei vienas beneficiantas. Pavyzdžiui, pirkimo procese pirminis beneficiantas yra asmuo ar organizacija, kuri gauna prekes/paslaugas. Kaip bebūtų, pardavėjas turi naudos gaudamas pinigus iš šių pirkinių. Beneficiantas gali būti proceso valdytojas, pvz., pirkimo proceso atveju, ar netgi subjektas, pvz., paciento buvimas ligoninėje.

Norint užbaigti įvairias proceso veiklas, gali prireikti įvairių rūšių įrankių, medžiagų ir t.t. Šios sąvokos yra sujungiamos į vieną sąvoką – *priemonės*.

2.1.4. Resursai

Norint vykdyti veiklas procese, reikia žmonių (darbuotojų) ir priemonių. Tada, kai tam tikros rūšies asmenys ar artefaktai (pvz., įrankis) yra ribojami vienokiu ar kitokiu būdu, yra kalbama apie *resursus* ar *resursų paskirstymą*. Labai dažnai (bet ne visada) apribojimais atsiranda, kai tas pats asmuo ar artefaktas yra naudojamas daugiau nei viename procese ar daugiau nei vienoje veikloje, vykstančioje tame pačiame procese.

Paskirstant resursus tarp kelių procesų ar kelių to paties proceso veiklų, gali būti reikalingi vienas ar keli *resursų valdytojai*. Resursų valdytojas gali sutapti su vadovu, bet tai nėra privaloma. Paprastai, resursų valdytojai aptarnauja daugiau nei vieną proceso dalį ir daugiau nei vieną proceso tipą. Antra vertus, procesų valdytojo buvimas nereiškia, kad jis susijęs su resursų paskirstymu kiekvienoje veikloje bei procese už kuriuos jis yra atsakingas. Paskirstymas daugeliu atvejų gali būti įvykdytas automatiškai. Resursų valdytojas gali būti reikalingas konflikto atveju, kad paskirtų prioritetus procesams ir veikloms.

Visi resursų poreikiai gali būti apytikriai suskirstyti į dvi grupes: *žmogiškieji resursai*, pvz., personalas, ir *nežmogiškieji resursai*, pvz., įrengimai, įmonės kabinetas, ir t.t. Žmogiškieji resursai yra svarbiausi verslo procese. Paprastai žmogiškųjų resursų skirstymas yra tvirtai susijęs su valdytojo organizacijos struktūra. Struktūra gali būti labai komplikuota. Norint išskirstyti resursus, galima

supaprastinti struktūrą, susidedančią iš tam tikro skaičiaus *resursų konteinerių* (*pools*). Kiekvienas konteineris apima keletą personalo narių ir turi vieną ar keletą resursų valdytojų, susijusių su juo. Kiekvienas resursų konteineris yra susietas su tam tikru veiklų skaičiumi.

2.1.5. Vidinė struktūra

Procesas gali turėti hierarchinio tipo vidinę struktūrą. Procesas gali būti sudarytas iš *subprocesų*. Subprocesas gali būti apibrėžtas, kaip rinkinys veiklų (įtraukiant ir tą atvejį kai rinkinyje yra tik viena veikla), norint pasiekti gerai apibrėžtą *subtikslą*.

Proceso struktūra gali būti sudėtinga, įtraukiant daug parametrų (kintamųjų) ir ryšių tarp jų. Subtikslas gali būti apibrėžtas, kaip proceso būsenos tam tikra dalis, kuri atitinka sąlygas.

Du subprocesai yra *nepriklausomi*, nes jie keičia proceso būseną, nepersikirsdami. Nepriklausomi subprocesai gali būti vykdomi *lygiagrečiai*.

2.1.6. Išorinė struktūra

Du skirtingi procesai gali turėti kažką bendro, pavyzdžiui, jie gali dalintis žmogiškaisiais resursais, pavyzdžiui, vienas ir tas pats asmuo vykdo veiklas. Kiti apima procesus su bendrais valdytojais, procesus su bendrais subjektais ir t.t.

Kai kurie ryšiai tarp skirtingų procesų yra susiję. Kiti ryšiai yra reikšmingesni, pvz., jie išreiškia situacijas, kai vienas procesas negali būti pradėtas ar baigtas be kito. Du procesai turintys reikšmingus ryšius yra vadinami *sujungtais*.

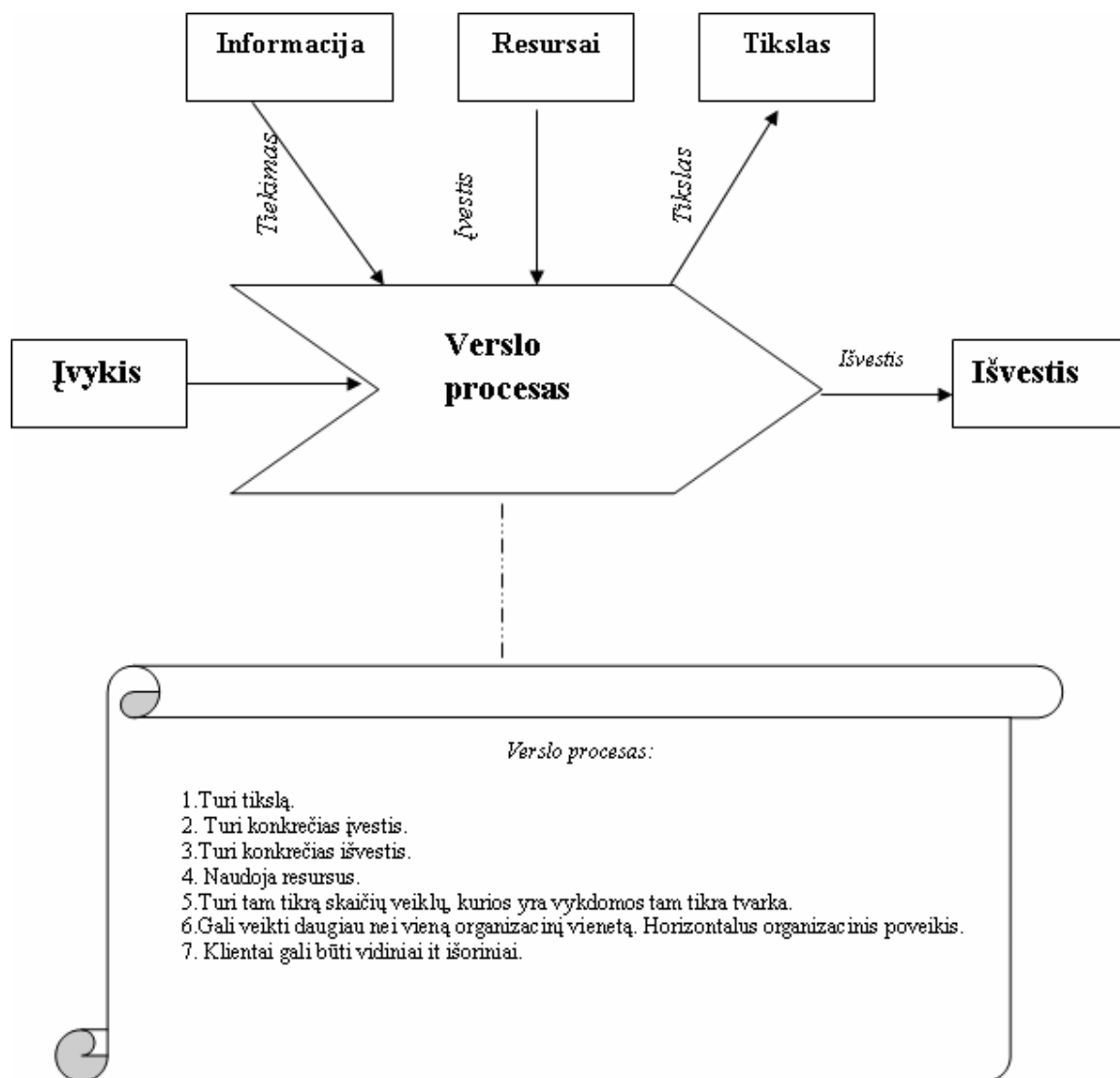
2.2. Verslo procesų modeliavimas

Verslo procesų modeliavimas reikalingas, nes:

- Modeliavimas yra būdas aprašyti tikrovę, supaprastinus ją iki valdomo informacijos kiekio.
- Formalizuotas būdas užfiksuoti, perduoti ir pasinaudoti žiniomis.
- Būdas sisteminti ir klasifikuoti surinktas žinias.
- Būdas kontroliuoti IT sistemos kūrimą tam tikrame detalumo lygmenyje.

Verslo procesų modeliavimas yra rinkinys arba grupė panašių veiklų ar verslo procesų, kurie teikia konkrečią paslaugą ar produktą [5]. Procesas yra tam tikras darbo veiklų išdėstymas, kuris turi pradžią ir pabaigą, aiškiai apibrėžtas įvestis ir išvestis: veiklos struktūrą. Verslo procesai vizualiai yra pateikiami diagramomis, kurias sudaro paprasta dėžė (veiklų vaizdavimui) su į ją įeinančiomis ir išeinančiomis rodyklėmis, prie kurių pateikti trumpi apibūdinimai – informacija, susijusi su veikla. 1

pav. pateiktas verslo procesų modeliavimo grafinis vaizdas, kuris nusako verslo procesų modeliavimo sudedamąsias dalis.



1 pav. Verslo procesų modeliavimo apibūdinimas

Informacija. Verslo procesai naudoja informaciją tam, kad atlikti ar užbaigti veiklas. Informacija, skirtingai nei resursai, nėra proceso išseikvojama – ji yra naudojama kaip pakeitimų proceso dalis. Informacija gali ateiti iš išorinių šaltinių, iš klientų, iš vidinių organizacinių elementų, ir netgi gali būti proceso produktu.

Tiekimo sąsaja iš informacijos objekto. Tiekimo sąsaja nurodo, kad informacija ar objektas, susijęs su procesu, nėra sunaudojami apdorojimo etape. Pavyzdžiui, dokumento šablonai gali būti vėl ir vėl naudojami, parūpinant naujas tam tikro stiliaus rūšis – šablonai nėra perdirbami arba sunaudojami veiklos metu.

Resursai. Resursas yra įvestis į verslo procesą, ir kitaip, nei informacija, yra sunaudojamas proceso metu.

Įvesties sąsaja iš resursų objekto. Įvesties sąsaja nurodo, kad prijungtas objektas ar resursas yra suvartojamas apdorojimo procedūrų metu. Pavyzdžiui, kai kliento užsakymai yra apdoroti, jie jau yra atlikti ir užbaigti, ir paprastai yra naudojami tik kartą per vienetinį resursą (užsakymą).

Tikslas. Verslo procesas turi tam tikrą aiškiai apibrėžtą tikslą. Jis yra organizacijos darbo priešastis, ir turi būti pilnai apibrėžtas šios organizacijos proceso pelno terminais, kad tenkintų verslo poreikius.

Tikslo sąsaja į tikslo objektą. Tikslo sąsaja nurodo prie verslo proceso prijungtą objektą, aprašo proceso tikslą. Tikslas yra verslo veiklos vykdymo rezultatas.

Išvestis. Verslo procesas paprastai pateikia verslui vieną ar daugiau vertingų išvesčių, dėl vidinio panaudojimo tam, kad patenkintų išorinius reikalavimus. Išvestis gali būti fizinis objektas (pranešimas ar sąskaita faktūra), pradinių resursų transformavimas į naujus išdėstymus (dienotvarkę ar sąrašą), ar apskritai verslo rezultatas toks, kaip kliento užsakymo įvykdymas.

Verslo procesų išvestis gali būti tiekama kitiems procesams, arba kaip užklauso rezultatas ar kaip impulsas pradėti naują veiklą.

Būsenos srauto sąsaja iš įvykio. Būsenos srauto sąsaja nurodo, kad kai kurie objektai pereina į verslo procesą. Jis užfiksuoja valdymo perdavimą kitam objektui ar procesui, su būsenos ar informacijos perdavimu iš veiklos į veiklą.

2.3. Verslo procesų valdymas

Verslo procesų valdymas (BPM (*Business Process Management*)) (veiklų rinkinys) yra būtinas naujam ar jau egzistuojančiam verslui, nes jis padeda organizacijoms optimizuoti dabartinius verslo procesus ir ateityje vyksiančius organizacinius ir operacinius pakeitimus. BPM yra būdas, maksimizuojantis organizacijos pajamas, kokybę ir pagerinantis vartotojų poreikių tenkinimą.

Verslo procesų valdymas yra atliekamas skirtingų įrankių pagalba, kurie padeda surenkant, modeliuojant, projektuojant, integruojant, išdėstant, testuojant ir prižiūrint kai kurias verslo veiklas. Kiekvienos kompanijos sėkmė ar nesėkmė priklauso nuo to, kaip ji sugeba valdyti ištisą savo procesų ciklą.

Verslo procesų valdymas reikalingas:

- Planuojant naują verslą ar darant pakeitimus versle.
- Siekiant išvengti dažnai pasitaikančių klaidų projekte.
- Aprašant verslo procesus visiems suprantama kalba, kuri padeda IT ir ne IT specialistams.
- Sudarant verslo ir duomenų modeliavimo diagramas ir surenkant verslo taisykles.

- Įvertinant duomenis.
- Pagerinant verslas-verslui transakcijas.

Verslo procesų valdymo privalumai:

- BPM veiklos, tokios kaip modeliavimas, automatizavimas, stebėjimas, analizavimas ir verslo procesų tobulinimas, padeda organizacijoms gauti pelną per kuo trumpesnę laiką.
- BPM verslo procesų modeliai vizualizuoja organizacijos veiklas ar verslas-verslui (B2B) transakcijas.
- BPM proceso srautų modeliai vizualizuoja organizacijose vykstančius procesų srautus, verslo-verslui transakcijas ir ryšius tarp procesų srautų.
- BPM duomenų srautų modeliai vizualizuoja organizacijos duomenų srautus ir verslo-verslui transakcijas.
- BPM suskaidytos diagramos, skirtos verslo procesų modeliavimui, proceso srautų diagramoms ir duomenų srautų diagramoms, detalai vizualizuoja procesus ir veiklas.
- Visos verslo proceso procedūros gali būti dokumentuotos bendrai prieinamoje saugykloje. Tai padeda specialistams ir ne specialistams suprasti skirtingus procesus, kurie vyksta kiekviename organizacijos skyriuje, jų našumą ir kiekvieno verslo proceso rezultatus.
- BPM padidina organizacijų tiekiamų produktų ar paslaugų kokybę ir kieki.
- BPM pranešimai gali būti labai naudingi organizacijos aukščiausiems vadovams ir visiems tiems, kam tie pranešimai gali būti reikalingi.
- BPM sumažina darbuotojų darbo valandas.
- BPM, lyginant su ne automatiniiais procesais, sumažina klaidų ir nesusipratimų skaičių.
- BPM padeda organizacijoms tvirtai laikytis valdžios instrukcijų.
- BPM yra B2B sandorių pagrindas.
- Suteikia konkurencinį pranašumą prieš konkurentus.
- Skirtingų scenarijų modeliavimo metodai gali būti panaudojami verslo procesams, tiriant pakeitimų rezultatus.

Verslo procesų valdymo (BPM) įrankiai yra naudojami kuriant taikomąją programą, kuri yra naudinga kuriant verslo proceso modelius, proceso srauto modelius, duomenų srauto modelius, taisykles, ir taip pat yra naudinga modeliuojant, optimizuojant, stebint ir prižiūrint įvairius procesus.

Be procesų koordinavimo, BPM įrankiai taip pat palengvina procesams taikomų taisyklių surinkimą, verslo procesų modeliavimą, procesų vykdymą nuosekliu būdu, verslo procesų stebėjimą, darbo rezultatų ataskaitų kūrimą, kurios padeda vadovams priimti geresnius sprendimus.

2.4. Verslo procesų re-inžinerijos samprata

Dažnai organizacijoms prireikia persvarstyti ir iš naujo peržiūrėti savo pasirinkimus, tikslus, užduotis ir t.t., todėl reikia patobulinti savo veiklą daugeliu aspektų. Tokia re-inžinerinė veikla yra vadinama verslo proceso re-inžinerija, taip pat dar žinoma kaip verslo proceso pertvarkymas arba verslo proceso patobulinimas [6].

Verslo proceso diagramų, proceso srauto diagramų (darbo srauto diagramų) ir duomenų srauto diagramų analizavimas gali labai pagelbėti verslo proceso re-inžinerijoje, todėl, kad šios diagramos yra labai efektyvios vizualizuojant organizacijos veiklas, procesus ir duomenų srautus.

Verslo procesų (biznio procesų) re-inžinerija (BPR) – viena iš priemonių, kurios pagalba kompanijos gali efektyviai ir inovatyviai pertvarkyti verslo procesus siekiant: sumažinti darbus ir jų atlikimo trukmę, tobulinti produktų ir paslaugų kokybę, supaprastinti kompanijos valdymo struktūrą, greitai reaguoti į vartotojų poreikius bei konkurentų veiksmus [6].

Šiandieną, būtų sunku įsivaizduoti pelningai ir sėkmingai veikiančią kompaniją, kurioje nenaudojamos informacinės technologijos (IT), nes tai tapo neatsiejama bet kokio verslo dalimi. Dažniausiai įgyvendinami BPR (*Business Process Reengineering*) projektai, kuriuose yra naudojamos ir integruojamos informacinės technologijos. Viena vertus tai yra efektyviausi projektai, kuriuos sėkmingai įdiegus pasiekiamas maksimalus rezultatyvumas, lyginant su BPR ne IT projektais, tačiau kita vertus tai yra rizikingiausi projektai.

Biznio transformacija – koncepcija, apimanti daug įvairių strategijų, kurias organizacijos naudoja norėdama tobulinti biznio (verslo) procesus. Geriausiai išvystytos ir dažniausiai taikomos strategijos yra šios:

- Verslo procesų re-inžinerija.
- Organizacinis vystymas, mokymas.
- Absoliutus kokybės valdymas.
- IT integracija ir naudojimas.

BPR ir IT integracija išskiriamos kaip dvi skirtingos strategijos, tačiau realiai tai yra glaudžiai susijusios ir lygiagrečiai realizuojamos strategijos. 1990 m., kai BPR tik pradėta naudoti organizacijose, išties IT ir BPR buvo dvi skirtingos strategijos, tačiau dabar jos kaip informacinės technologijos, yra plačiai naudojamos įvairiose verslo procesuose ir versle, išskirti šias dvi strategijas nėra tikslo.

BPR pagrindinis dėmesys buvo skiriamas IT panaudojimui ir integracijai, radikaliai keičiant organizacinius procesus. Toliau plėtojant BPR buvo suvokta, norint skirti visą dėmesį ir pastangas IT integracijai į organizaciją, iš esmės yra neteisingas požiūris, įtakojantis BPR projektų žlugimą.

Evoliucionuojant BPR daugiau dėmesio yra skiriama procesams, organizacijos strategijai, kultūrai ir struktūrai, visa tai susiejus gaunamas “top-down” organizacijos verslo scenarijus.

Vienas pagrindinių BPR tikslų yra greitai ir tiksliai reorganizuoti procesus organizacijoje. Reorganizuojant procesus siekiama, kad organizacija:

- Greitai reaguotų į vartotojo poreikius.
- Minimizuotų ciklų skaičių visuose verslo procesuose.
- Gerintų kokybę.
- Mažintų riziką.
- Siektų lankstumo.
- Siektų inovatyvumo.
- Siektų rezultatyvumo.
- Minimizuotų žmogiškųjų išteklių naudojimą.
- Optimizuotų administracines veiklas.
- Siektų vadovavimo tobulinimo.
- Siektų bendrų verslo kaštų sumažinimo.

Siekiant įgyvendinti šiuos tikslus yra naudojamos įvairios BPR metodologijos ir jų kombinacijos. Dabartinė IT rinka BPR tikslams įgyvendinti siūlo didelį pasirinkimą įvairių priemonių.

Metodologija tai yra tikslingai transformuota teorija į praktiką, kuri panaudojama realaus pasaulio situacijose (Preece ir Peppard, 1996). BPR metodologija yra nuoseklių technikų ir nuorodų rinkinys, kurio pagalba perprojektuojami ir reorganizuojami verslo procesai organizacijoje (Valaris ir Glykas, 1999). Metodologijos naudojimas yra svarbus dėl daugelio priežasčių:

- Metodologija yra ankstesnė patirtis, žinios ir idėjos. Sutaupomas laikas kaupiant žinias ir informaciją, išvengiama “dviračio išradinėjimo” sindromo.
- Metodologija duoda tam tikrą organizuotumo lygį, supaprastina planavimą ir kontrolę. Metodologijos inicijavimas leidžia organizacijai aiškiau matyti bendrą verslo “vaizdą”, geriau įvertinti galimas problemas ir lūkesčius, kokybiškai planuoti ir organizuoti BPR. Kai yra įvertintos visos galimos grėsmės ir lūkesčiai, kai aiškiai apibrėžta veikla, organizacija su daug mažesne rizika gali BPR projektus pritaikyti konkrečiai savo veiklai. Kad ir kokia gera būtų metodologija ar BPR priemonės, tai turi būti adaptuota prie organizacijos poreikių ir lūkesčių.
- Metodologija leidžia geriau suprasti kiekvienam BPR veikėjui už ką jis/ji atsakinga ir kokią funkciją jis/ji turi vykdyti. Tokiu būdu yra sutaupomi žmogiškieji resursai, nes išvengiama nereikalingų veikėjų, kurie neaišku ką daro ir už ką atsako.
- Metodologija leidžia aiškiai apibrėžti ir identifikuoti verslo standartus ir reikalingus reikalavimus BPR veikėjams.

- Metodologijos pagalba visas procesas gali būti valdomas ir tobulinamas, globaliai kaip vienas vienetas, arba atskiruose vienetuose.

Kiekviena metodologija turi trūkumų, kaip ir privalumų. Dažniausiai yra atsisakoma esamų metodologijų, nes jos visą BPR procesą “spaudžia” į taisyklių rinkinius, kurie konkrečiai modeliuoja situacijas ir procesus. Manoma, kad tai varžo verslo procesų kūrybiškumą, inovatyvumą ir natūralų evoliucionavimą. Atsižvelgiant į tai yra atsisakoma sėkmingesnio kelio, kurio pagalba būtų siekiama užsibrėžto tikslo ir nekartojamos tos pačios dažniausiai pasitaikančios klaidos. Naudojant metodologijas būtina atsižvelgti į organizacijos poreikius ir savitumą, asimiliuojant tai kas yra reikalinga, modeliuoti savitas situacijas remiantis kitų patirtimi.

Daugybė įvairių modelių ir BPR metodologijų yra siūloma šiuolaikinėje rinkoje, dauguma jų pateikia tokius pačius sprendimus ir veiksmų kelius (Butler, 1994). Prieštaraujant Butler galima teigti, kad tarp didelės įvairovės produktų galima rasti, kas reikalinga konkrečios organizacijos verslo procesams modeliuoti. Šiandieną prieinamų priemonių ir įvairių technikų sintezė, pateikia inovacinius metodus ir metodologijas.

Dažniausiai praktikoje naudojami 3 arba 4 fazių modeliai. Coopers & Lybrand, konsultacinė firma Didžiojoje Britanijoje, kuri konsultavo 40 organizacijų, apibendrino ir pateikė tokį keturių fazių modelį:

- Esamo proceso apibrėžimas.
- Įvertinimas ir identifikavimas esamo proceso trūkumų ir privalumų.
- Testavimas alternatyvių modelių, parinkimas optimaliai tinkančio modelio esamam procesui.
- Proceso realizavimas.

Konsultacinė firma Texas Instruments, JAV, pateikė kiek kitokį modelį, nei kolegos iš senojo kontinento:

- Inicijavimas ir pasiruošimas.
- Įsisavinimas ir diagnozė.
- Perprojektavimas.
- Įgyvendinimas.

Literatūroje yra daug įvairių metodologijos modelių, kuriuose pagrindinis dėmesys skiriamas planavimui, kontrolei ir įgyvendinimui. Reussmann (1994) pateikia trijų fazių modelį:

- Problemos apibrėžimas.
- Proceso analizė, diagnozė ir perprojektavimas.
- Įgyvendinimas.

Pateikta modelio struktūra apima visas reikalingiausias fazes, atsisakant papildomų fazių ir procesų, kurie yra inicijuojami BPR.

3. VERSLO PROCESŲ MODELIAVIMO KALBŲ APRAŠYMAS

3.1. BPMN standartas verslo procesams aprašyti

3.1.1. BPMN (*Business Process Modeling Notation*) apžvalga

BPMN (*Business Process Modeling Notation*) yra vienas iš kelių standartų, kuris yra skirtas verslo procesams modeliuoti [1]. Veiklos (verslo) procesai modeliuojami daugelį metų, ir ypač suaktyvėjo jų modeliavimas per paskutinius metus. Tačiau metodai ir priemonės buvo tik iš dalies sėkmingi arba visiškai nesėkmingi, nes trūko standartų ir pilnų galimybių, apimančių verslo procesų modeliavimą, projektavimą ir vykdymą. Norint suprasti procesų architektūrą, modeliavimą, paskirstymą, reikalingi verslo modeliavimo ir verslo vykdomosios kalbos standartai.

BPMN yra naujas standartas skirtas modeliuoti verslo procesų srautų ir web servisų procesus, išleistas BPMI (*Business Process Management Initiative*) standartų organizacijos, jos veikla yra nukreipta verslo procesams gerinti. Standartų organizacijos pagrindinis tikslas – pateikti visiems verslo dalyviams: verslo analitikams, technikos kūrėjams, verslo žmonėms lengvai suprantamą verslo proceso modeliavimo notaciją. Kitas ne ką mažiau svarbus tikslas – užtikrinti, kad vykdomosios verslo procesų kalbos XML pagrindu, pavyzdžiui, BPEL4WS (*Business Process Execution Language For Web Services*) ir BPML (*Business Process Modeling Language*), būtų lengvai atvaizduojamos bendra notacija. Ši organizacija suvienijo ankstesnius skirtingus procesų modeliavimo, vaizdavimo metodus, organizacijos taikomųjų programų integravimą, verslas-verslui (*business-to-business*) integravimą į bendrą standartą.

Norint suprasti procesų architektūrą, modeliavimą, paskirstymą, reikalingi verslo modeliavimo ir verslo vykdomosios kalbos standartai. Todėl BPMI standartų organizacija plėtoja tris standartus:

- BPMN – verslo procesų modeliavimo standartą;
- Verslo proceso modeliavimo kalbą (BPML) – verslo vykdomųjų kalbų standartą;
- Verslo proceso užklausų kalbą (BPQL) – kaip el. verslo procesų paskirstymo ir vykdymo vartotojo sąsajos standartą.

Svarbi BPMI standartų savybė yra ta, kad jie buvo sukurti remiantis matematiniu pagrindu – Pi-Calculus. Tai formalių skaičiavimų metodas, leidžiantis apibrėžti dinامينius ir mobilius procesus. Jo privalumai:

- Pi-Calculus leidžia tikrinti nuoseklumą, logiškumą.
- Pi-Calculus užtikrina aklaviečių ir klaidų suradimą.
- Pi-Calculus leidžia optimizuoti procesus.

BPMN notacijos vienas iš privalumų, kad ji lengvai suprantama visų verslo naudotojų: verslo analitikų, technikos kūrėjų ir kt.

BPMN turi nemažai privalumų [2], lyginant su UML (*Unified Modeling Language*):

- Pirma, ji leidžia aprašyti procesus srautais, tai lengvai suprantama verslo analitikui, ir turi išsamią tipinių modelių elementų aibę.
- Antra, dėl matematinio pagrindo ji lengvai atvaizduojama verslo vykdomosiomis kalbomis.

Esminiai BPMN privalumai:

- Nepriklausoma metodologija sudėtingų verslo procesų atvaizdavimo atžvilgiu.
- Lengvai išreiškiama vykdomosiomis kalbomis (BPML, BPEL4WS).
- Prieinama ir vartotina verslo visuomenės aplinkoje.
- Naudinga vaizduojant sudėtingus BPML procesus.
- Naudinga vaizduojant sąveikas tarp organizacijų B2B (*business-to-business*) situacijose.
- Palaiko tik tas modeliavimo sąvokas, kurios susijusios su verslo procesais.

BPMN apima ne vieną modeliavimo būdą ir leidžia kurti tiesioginius verslo procesus. BPMN modelis gali būti trijų tipų:

- Privatūs (vidiniai) verslo procesai – vidiniai organizacijos procesai.
- Abstraktūs (vieši) procesai – procesai, kurie vaizduoja sąveikavimą tarp vidinių verslo procesų ir kitų procesų ar dalyvių.
- Bendradarbiavimo (globalūs) procesai – procesai, kurie vaizduoja sąveikavimą tarp dviejų ar daugiau verslo objektų.

Visiems verslo procesams atvaizduoti BPMN notacija specifikuoja keturias pagrindines kategorijas objektų:

- Srauto objektus (*Flow Objects*).
- Sujungimo objektus (*Connecting Objects*).
- Sritis (*Swimlane*).
- Žmogaus darbo produktus (artefaktus) (*Artifacts*).

Srauto objektai yra pagrindiniai elementai, kurie apibrėžia verslo procesus. Srauto objektai yra trijų rūšių:

- Įvykiai (*Events*).
- Veiklos (*Activities*).
- Vartai (*Gateways*).

Sujungimo objektai jungia tarpusavyje srauto objektus, arba kitą kokią informaciją. Sujungimo objektai skirstomi taip:

- Sekos srautas (*Sequence flow*) - paprastas perdavimas.
- Pranešimų srautas (*Message flow*) – perdavimas skirtas pranešimų įvykiams.

- Asociacija (*Association*) – naudojama asocijuoti duomenis, tekstą ir kt. su srauto objektais.

Sritys – reiškia dalyvius, t.y. sritys turi vardus, kurie reiškia, kas ką daro.

Žmogaus darbo produktai (artefaktai) naudojami pateikti papildomą informaciją apie procesus.

Yra trys standartizuoti žmogaus darbo produktai, bet modeliuotojai gali modeliuoti ir daugiau, tiek kiek jiems reikia. Standartizuoti žmogaus darbo produktai yra:

- Duomenų objektas (*Data Object*).
- Grupė (*Group*).
- Anotacija (*Annotation*).

Verslo procesų modeliavimo notacija (BPMN) sukuria standartizuotą „tiltą“ tarp proceso analizės ir proceso įgyvendinimo. Standartizuojant procesus BPMN galima išspręsti verslo procesų komunikavimą.

3.1.2. BPMN grafiniai žymėjimai

BPMN susideda iš vienos diagramos, vadinamos BPD (verslo procesų diagrama) (*Business Process Diagram*). Ši diagrama buvo sukurta, siekiant du dalykus daryti gerai. Pirmą, ji yra lengvai naudojama ir suprantama. Ją galima naudoti, norint greitai ir lengvai modeliuoti verslo procesus, kurie lengvai yra suprantami ne-techniniams vartotojams. Antra, ji yra aiški, modeliuojant labai sudėtingus verslo procesus ir gali būti lengvai reiškiamas verslo vykdomosiomis kalbomis.

Norint modeliuoti proceso srautą, paprastai pirmiausia yra modeliuojami įvykiai, sukiantys procesą. Tada analizuojami toliau vykstantys procesai ir galiausiai proceso srauto rezultatai. Verslo srauto sprendimai ir šakojimai modeliuojami naudojant vartų (*gateway*) simbolių.

Be to, galima aprašyti vidinius (*sub-processes*) procesus, kurie modeliuojami kitoje verslo proceso diagramoje, kuri turi nuorodą (*hyperlink*) proceso simboliuje. Jei procesas neturi vidinių procesų, tai jis vertinamas kaip veiksmas – žemiausias proceso lygis. Proceso simboliuje ženklas „+“ pažymi, kad procesas yra detalizuotas. Jei jis neturi šio ženklo, tai reiškia užduotį (*task*).

Leidžiantis į gilesnę verslo procesų analizę, galima vaizdžiai atskirti, kas ką atlieka proceso metu ir kokie įvykiai įvyksta. Įvykiai ir procesai yra sukeliama į atskirus konteinerius (*Pools*), pagal kuriuos aiškiai matosi, kas už kokią veiklą ar procesą yra atsakingi.




Norint dar tiksliau detalizuoti veiklas pagal jų atlikėjus, galima tą patį konteinerį suskirstyti į juostas (*Lanes*). Konteineris paprastai rodo organizaciją, o linijos išskiria organizacijos padalinius (taip pat šie žymėjimai gali vaizduoti ir funkcijas, taikomąsias programas, sistemas ir kt.).

Modeliuojant sudėtingesnius verslo procesus, pavyzdžiui, verslas-verslui, reikia aprašyti sudėtingesnius įvykius (pranešimų, laiko, verslo taisyklių ir klaidų sąlygas). BPMN notacija leidžia tiksliai pateikti kylančių įvykių tipus ir pavaizduoti juos grafinais simboliais.

3.1.2.1. Įvykiai (*Events*)

Modeliuojant verslo procesą, modeliuojami įvykiai, kurie vyksta versle, ir parodoma kaip jie veikia procesų srautus. Įvykis arba pradeda proceso srautą, arba įvyksta srauto metu, arba užbaigia jį. BPMN notacijoje yra išskirta trijų pagrindinių tipų įvykiai: pradžios (*Start*), tarpiniai (*Intermediate*) ir pabaigos (*End*), jų žymėjimai pateikti lentelėje Nr. 1.

Lentelė Nr. 1 BPMN notacijos įvykių pagrindiniai tipai

Pradžios įvykis (<i>Start Event</i>)		Tarpinis įvykis (<i>Intermediate Event</i>)		Pabaigos įvykis (<i>End Event</i>)	
	Pradeda proceso srautą		Atsitinka proceso srauto metu		Pabaigia proceso srautą

Pradžios ir dauguma tarpinių įvykių turi „sukelėjus“ (*Triggers*), kurie nurodo įvykio priežastį. Minėti įvykiai gali būti sukeliama daugeliu būdų. Tarpiniai įvykiai įvyksta tarp pradžios ir pabaigos įvykių. Tai įvykis, kuris įvyksta po to, kai procesas prasideda. Tarpiniai įvykiai daro įtaką proceso srautui, tačiau nepradeda ir neužbaigia proceso. Pabaigos įvykiai apibrėžia rezultatą, kuris yra sekos srauto (*Sequence Flow*) pabaigos rezultatas.

Visi įvykių tipai yra mažo apskritimo formos, tik skirtingų stilių linijų, iš kurių atskiriama kokio tipo tai įvykis. Apskritimo vidurys yra skirtas „sukelėjo“ ar „rezultato“ ikonai. Sudėtingesni įvykiai (žr. 2 pav.), t. y. įvykiai su ikonomis (pvz.: pranešimo (*Message*), laiko žymeklio (*Timer*), klaidos (*Error*), atšaukimo (*Cancel*), kompensacijos (*Compensation*), taisyklės (*Rule*), nuorodos (*Link*), daugialypis (*Multiple*), nutraukimo (*Terminate*)), yra naudojami modeliuojant sudėtingus procesų srautus, tokius kaip verslas-verslui (B2B) web servigus.



2 pav. Sudėtingesnių įvykių tipai BPMN notacijoje

3.1.2.2. Veiklos (Activities)

Veikla yra darbas atliekamas verslo proceso viduje. Yra trijų tipų veiklos: procesas (*Process*), sub-procesas (*Sub-Process*), kuris dar skirstomas į sutrauktą sub-procesą (*Collapsed Sub-Process*) ir išplėstą sub-procesą (*Expanded Sub-Process*), ir užduotis (*Task*). Kiekviena jų yra grafiškai pavaizduota suapvalintu stačiakampio simboliu (žr. lentelę Nr. 2). Tačiau procesas nėra konkretus grafinis objektas. BPMN notacijoje procesas yra srauto objektų (*Flow Object*) diagrama, kurią sudaro kitų grafinių objektų rinkinys.

Lentelė Nr. 2 BPMN notacijos pagrindiniai veiklų (Activities) tipai

Užduotis (<i>Task</i>)		Sutrauktas sub-procesas (<i>Collapsed Sub-Process</i>)	Išplėstas sub-procesas (<i>Expanded Sub-Process</i>)
	Nedaloma veikla. Atliekama tada, kai yra vykdoma		
		Sudėtinė veikla, kurios detalės nematomos	Sudėtinė veikla, kurios detalės matomos

Sub-procesas yra sudėtinė veikla, apibrėžiama kaip kitų veiklų srautas. Proceso sraute sub-procesas yra grafinis objektas, tačiau norint parodyti kitą (priklausomą ar nepriklausomą) procesą, jis gali būti ir „atidaromas“ (paspaudus „+“ ženklą). Sub-proceso skiriamasis ženklas yra „+“ suapvalinto stačiakampio viduryje. Sub-procesai yra vaizduojami dvejopai: „sutraukti“ (*collapsed*), t. y. tokie, kurių detalės yra nematomos; „išplėsti“ (*expanded*), kurių detalės matomos. BPMN notacija specifikuoja penkių tipų sub-procesų žymeklius. Sutraukto sub-proceso žymeklis („+“ ženklas) gali

būti kombinuojamas su kitais keturiais žymekliais (ciklo (*Loop*), lygiagretumo (*Parallel*), kompensavimo (*Compensate*) ir specialiu (*Ad-Hoc*)).







Užduotis yra nedaloma veikla, kuri įtraukiama į procesą. Tiesioginis vartotojas ir/arba programa atlieka užduotį tada, kai ji yra vykdoma. BPMN notacijoje yra trijų tipų užduotys (ciklo (*Loop*), kartotinumų (*Multiple Instance*), kompensavimo (*Compensation*)).

3.1.2.3. Vartų elementas (*Gateway*)

Verslo proceso srauto sprendimai, suliejimai, išsišakojimai ir sujungimai modeliuojami naudojant vartų (*Gateway*) simbolį. Terminas „vartai“ sufleruoja, kad šiame srauto taške yra leidžiamas arba neleidžiamas praėjimas. Taigi, vyksta sprendimas – atsakymas į klausimą. Klausimas turi apibrėžtą rinkinį alternatyvių atsakymų, t. y. vartai – išėjimų. Svarbi savybė yra ta, kad vartai turi stereotipus. Pavyzdžiui, vartų, žymimo rombu, viduje pavaizduotas pliusas reiškia, kad visi iš jo išeinantys sekos srautai vyksta tuo pačiu metu.

Vartai yra skirtingų tipų (žr. lentelę Nr. 3), kurių kiekvienas apibrėžia kiek bus galimų išėjimų. BPD diagramoje vartai yra rombo formos, kurio viduryje gali būti ikona, nusakanti vartų tipą. Vartai kontroliuoja išsišakojimą ir/arba susiliejančią sekos srautą (*Sequence Flow*).

Lentelė Nr. 3 Vartų (*Gateway*) tipai BPMN notacijoje



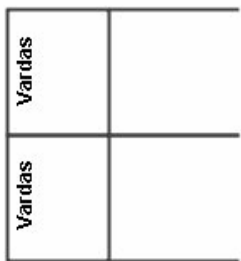

Vienintelio sprendimo/suliejimo (<i>Exclusive Decision/Merge</i>)		Daugiau nei vieno sprendimo/suliejimo (<i>Inclusive Decision/Merge</i>)		Sudėtinio sprendimo/suliejimo (<i>Complex Decision/Merge</i>)	Lygiagretaus išsišakojimo/sujungimo (<i>Parallel Fork/Join</i>)	
	arba 		Arba (OR)			Ir (AND)
Duomenimis paremtas (<i>Data-Based</i>)						
	Įvykiu paremtas (<i>Event-Based</i>)					

3.1.2.4. Konteineriai (*Pools*) ir juostos (*Lanes*)

Konteineris (*Pool*) simbolizuoja proceso dalyvį. Dalyvis gali būti tam tikra verslo esybė (organizacija) ar rolė (pirkėjas, pardavėjas, gamintojas, tiekėjas ir t. t.). Konteinerį galima padalyti į juostas (*Lanes*), kurios tęsiasi per visą konteinerio ilgį. Kiekviena juosta vaizduoja atsakingumą už tam tikrą dalį visų veiklų. Patalpinti procesai į konteinerius ir juostas, tiksliai apibrėžia, kas ką daro: įvykiams apibrėžia, kur jie įvyksta, o vartams – kur priimami sprendimai ar kas juos priima.

Konteineris yra stačiakampio formos sritis ir gali būti išdėstyta horizontaliai arba vertikaliai (žr. lentelę Nr. 4).

Lentelė Nr. 4 Konteinerių (Pools) ir juostų (Lanes) elementai BPMN notacijoje

Horizontalus konteineris (<i>Horizontal Pool</i>)		Vertikalus konteineris (<i>Vertical Pool</i>)	
	Konteineris		Konteineris
	Konteineris padalintas į juostas		Konteineris padalintas į juostas

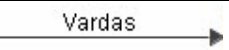


3.1.2.5. Grafiniai sujungimo objektai

Sekos srautas (*Sequence Flow*) parodo, kokia tvarka atliekamos veiklos procese. Juos vaizduoja strėle su užpildyta rodyklyte (žr. lentelę Nr. 5). Sekos srautas gali būti dar sąlyginis (*Conditional Sequence Flow*), pagal nutylėjimą (*Default Sequence Flow*).

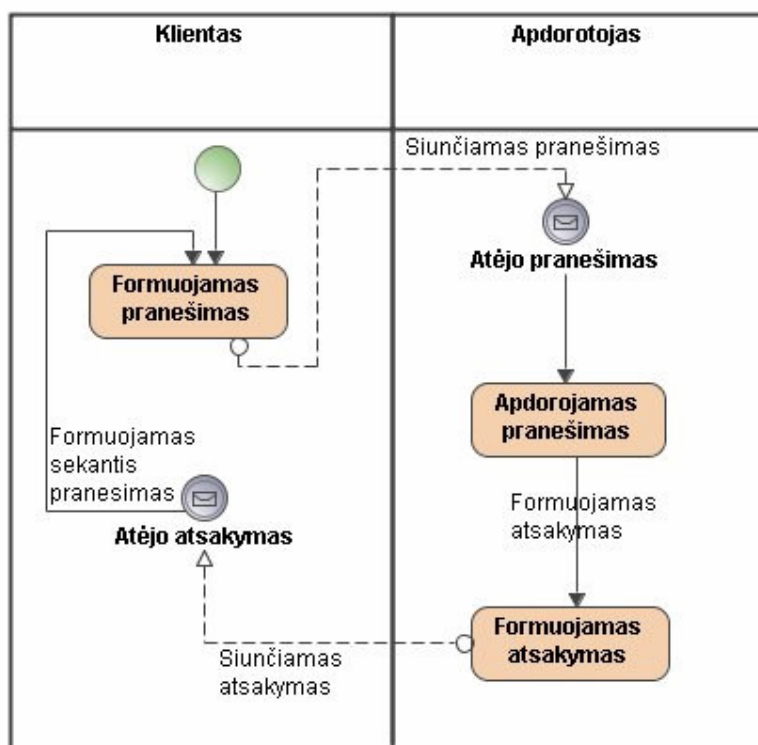
Punktų linijos su rodyklėmis rodo pranešimo srautą (*Message Flow*) tarp dviejų esybių. Pranešimų srautas gali būti naudojamas tik tarp įvykių, procesų ir sprendimų, esančių skirtinguose konteineriuose.

Asociacija (*Association*) yra naudojama artefaktams jungti su srauto objektais. Tekstiniai ir grafiniai ne srauto objektai gali būti sujungti su srauto objektais ir pačiu srautu. Asociacija gali būti ir kryptinė (*Directional*). Kryptinė asociacija naudojama parodyti ar duomenų objektas (*Data Object*) yra įvestis į veiklą ar išvestis iš jos.

Lentelė Nr. 5 Sujungimo elementai

Sekos srautas (<i>Sequence Flow</i>)		Pranešimų srautas (<i>Message Flow</i>)		Asociacija (<i>Association</i>)	
	Parodo kokia tvarka yra atliekamos veiklos procese		Parodo pranešimų srautą tarp esybių		Artefaktus jungia su srauto objektais

Kadangi buvo aptarti visi pagrindiniai verslo proceso diagramos elementai, tai galima pabandyti aprašyti BPMN notacijos žymėjimais koki nedidelį verslo procesą. 3 pav. pateiktas paprastas e-paslaugos verslo procesas.



3 pav. e-paslaugos verslo procesas, aprašytas BPMN notacija

Šiame verslo procese klientas formuoja pranešimą, suformuotą pranešimą siunčia apdorotojui, apdorotojas gavęs pranešimą jį apdoroja ir formuoja atsakymą. Apdorotojas suformavęs atsakymą, siunčia tą atsakymą klientui, klientas gavęs atsakymą formuoja sekantį pranešimą.

Diagramoje yra konteineris (*Pool*) suskirstytas į juostas (*Lanes*), kurių pavadinimai „Klientas“ ir „Apdorotojas“. „Klientas“ vykdo užduotį (*Task*) „Formuojamas pranešimas“, o „Apdorotojas“ vykdo užduotis (*Tasks*) „Apdorojamas pranešimas“ ir „Formuojamas atsakymas“, taip pat yra tarpiniai pranešimo įvykiai (*Intermediate Message Events*) „Atėjo pranešimas“ ir „Atėjo atsakymas“. Pranešimų srauto (*Message Flow*) elementas „Siunčiamas pranešimas“ jungia užduotį „Formuojamas pranešimas“ su tarpiniu pranešimo įvykiu „Atėjo pranešimas“. Pranešimų srauto elementas „Siunčiamas atsakymas“ jungia užduotį „Formuojamas atsakymas“ su tarpiniu pranešimo įvykiu „Atėjo atsakymas“.

3.2. Verslo procesų modeliavimas, naudojant UML diagramas

UML (*Unified Modeling Language*) – standartinė vizualaus modeliavimo notacija (žymėjimas), naudojamas atliekant PĮ analizę ir projektavimą, gali būti efektyviai panaudotas ir kuriant vaizdų verslo modelį [10]. UML modeliavimo kalba yra gana paplitusi, lengvai išmokstama ir patogi realizavimui, specifikavimui, dokumentavimui. UML modeliavimo kalbai būdinga diagramų įvairovė, todėl ši modeliavimo kalba labai lanksti ir patogi įvairiam projektavimui. UML modeliavimo kalboje diagramos skirstomos į tris kategorijas: statines, dinamines ir organizavimo, valdymo [11]. Dokumentuoti verslo procesams verslo analitikas gali naudoti tuos pačius žymėjimus ir priemones, kuriuos PĮ architektai ir projektuotojai naudoja dokumentuojant PĮ sistemas. Kalbant ta pačia kalba, dvi grupės gali žymiai geriau suprasti viena kitą, labiau užtikrinant, kad PĮ sistemos tenkins verslo poreikius. UML leidžia specifiuoti, vizualizuoti ir dokumentuoti programinės įrangos sistemų modelius – jų struktūrą ir projektus. Verslo modeliavimą galima apibrėžti, kaip verslo organizacijos elementų ir sąryšių tarp jų abstrakciją.

UML autoriai siekė tokių tikslų: Galėti sistemas modeliuoti visapusiškai, pradedant idėja ir baigiant realizacija.

- Išspręsti modelio problemą (detalizacijos lygius).
- Sukurti kalbą, kurią vartotų ne tik žmogus, bet ir kompiuteris.

UML apibrėžia:

- Elementus, kurie sudaro semantikos modelį (semantika).
- Notaciją, skirtą vizualiam modelio elementų pateikimui (sintaksė).
- Naudojimo taisyklės (pragmatika).
- UML apibrėžti kalbos koncepcijos išplėtimo ir specializavimo mechanizmai.

UML neapibrėžia ir nenurodo:

- Programavimo kalbos – UML yra semantinis modelis, kuris lengvai pernešamas į bet kokį objektiškai orientuotą modelį, bet nepriklauso nuo konkrečios realizavimo kalbos.
- Instrumentus – UML nekelia reikalavimų CASE priemonėms, padarytoms UML pagrindu ir neapibrėžia jų naudojimo. Bet instrumentas, palaikantis UML, turi tiksliai atitikti kalbos semantiką.
- Kūrimo procesas – kūrimo proceso apibrėžimas nebuvo autorių tikslas, UML specialiai buvo padaryta nepriklausoma nuo kūrimo proceso.

UML turi turtingą diagramų rinkinį. Iš esmės tai yra elementų grupės grafinis vaizdas. Diagramų tikslas - vaizduoti skirtingus sistemos modelio aspektus, todėl diagrama yra sistemos modelio projekcija ir pasižymi tam tikru abstrakcijos laipsniu. Keletas diagramų, kaip būsenų srautų ir veiklos diagramos, nėra kilę iš objektiškai orientuotos paradigmos, ir sukelia potencialias problemas prijungiant šias prie kitų sistemos modelio diagramų (Berkem 1998).

Nors verslo ir verslo procesų modeliavimas yra minimas skirtingose UML specifikacijos dalyse, tačiau pati dokumentacija (OMG 2000) stokoja pagrįstos argumentacijos, kaip atskiros diagramos gali būti pritaikytos verslo modeliavimo kontekste.

Žemiau yra pateikiama, ką kiekviena diagrama naudojantis galima aprašyti:

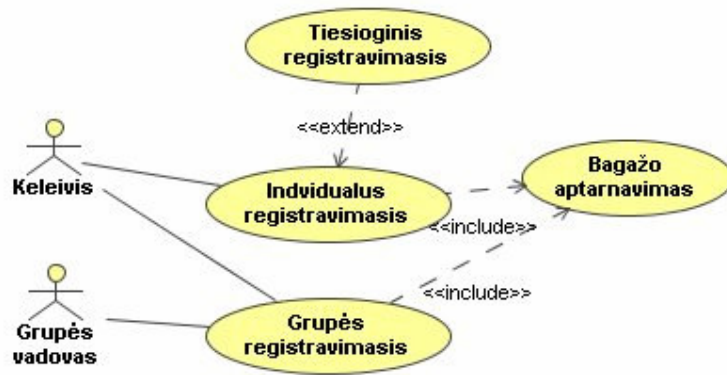
- Panaudojimo atvejų diagrama - vaizduojamas sistemos funkcionalumas matomas išoriniams aktoriams.
- Klasių diagrama - vaizduoja nekintamą sistemos dalį klasėmis ir sąryšiais tarp jų.
- Objektų diagrama - vaizduoja atskiras klasių modelio dalis.
- Bendradarbiavimo diagrama - atvaizduoja objektų sąveikas siunčiant pranešimus, išskiriant sąveikaujančius aktorius ir jų ryšius.
- Sekų diagrama - vaizduoja objektų sąveikas išskiriant sąveikavimo seką.
- Būsenų diagramos - vaizduoja tam tikros klasės objektų būsenas ir jų perėjimus.
- Veiklos diagrama - vaizduoja kiekvieno apibrėžto proceso veiklų srautus.
- Komponentų diagrama – vaizduoja programinės įrangos komponentų priklausomybes.
- Išdėstymo diagrama - vaizduoja laikinių procesų elementus.

3.2.1. Verslo panaudojimo atvejų modelis

Veiklos procesai abstrakčiame lygyje aprašomi kaip UML panaudojimo atvejai. Kaip atrodo geras verslo modelis? Pirmiausia, jis yra sudarytas iš dviejų pagrindinių dalių: *verslo panaudojimo atvejų modelio* ir *verslo objektų modelio*; abu šiuos modelius galima sukurti naudojant UML modeliavimo kalbą [10].

Verslo panaudojimo atvejų modelis aprašo ir pavaizduoja verslo procesus, ir jų sąveikas su išore. Šie procesai pažymimi, kaip seka veiksmų, kurie yra reikšmingi verslo aktoriams. Verslo procesams atstovauja verslo panaudojimo atvejai, o išoriniams dalyviams – verslo aktoriai.

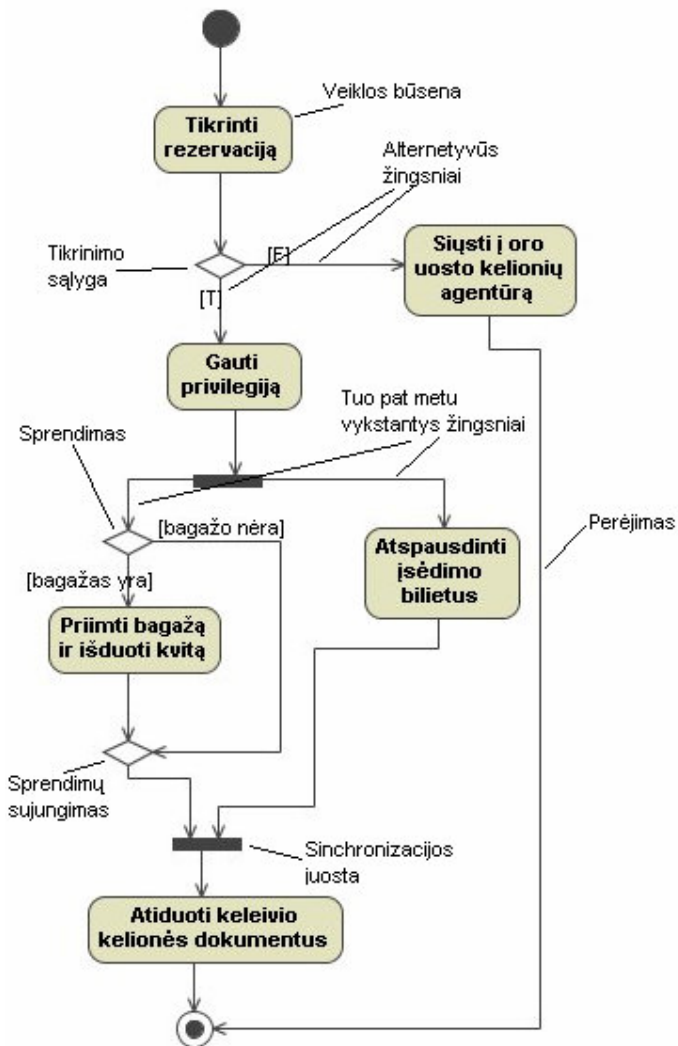
Iš kitos pusės žiūrint, galima teigti, kad verslo panaudojimo atvejų modelis aprašo smulkias, detalias darbo sekas. Pagrindiniai panaudojimo atvejai 4 pav. yra „Individualus registravimasis“ ir „Grupės registravimasis“. Diagramoje taip pat parodyti „verslo aktoriai“: žmogeliukais pažymėti „Keleivis“ ir „Grupės vadovas“. Kad pilnai suprasti verslo tikslus, reikia gerai žinoti, su kuo verslas sąveikauja; kas jam kelia reikalavimus, arba kas yra suinteresuoti produkcija, gamybos apimtimi, gamybiniu pajėgumu. Įvairių tipų „sąveikautojai“ yra pateikiami kaip verslo aktoriai. Dažniausia tai būna užsakovai, klientai, bet gali būti tiekėjai, partneriai, potencialūs pirkėjai, vidaus valdžia, ar iš dalies verslo kolegos ir kt.



4 pav. Panaudojimo atvejų diagrama keliems verslo procesams

Panaudojimo atvejįs detalai susijęs su verslu yra dokumentuojamas verslo panaudojimo atvejų specifikacijoje. Joje yra teksto taip kaip ir vienoje ar daugiau UML veiklos diagramų ir galimas atvejįs sistemos panaudojimo atvejų diagramose. Verslo panaudojimo atvejų specifikacijose paprastai būna šie užrašai: pavadinimas, trumpas aprašymas, vykdomi tikslai, nauda, darbų seka, ypatingi reikalavimai, praplėtimo ypatybės, tarpusavio ryšiai, veiklos diagramos, panaudojimo atvejų diagramos.

Esminis dalykas yra darbų seka/veiksmų seka, kuri nusako ką verslas atlieka, kad suteiktų reikšmę verslo aktoriui, o ne kaip verslas spendžia problemas. Visi, kuriuos liečia verslas, turi suprasti aprašymą. Darbų sekų struktūra aprašoma grafiškai, UML veiklos diagramų pagalba. Vaizdingą darbų sekos struktūros pateikimą atitinka, aprašymas teksto forma pateiktas verslo panaudojimo atvejų specifikacijoje. 5 pav. pateikiamas veiklos diagramos pavyzdys.



5 pav. UML veiklos diagrama, aprašanti darbų sekos struktūrą

Šios diagramos komponentai yra:

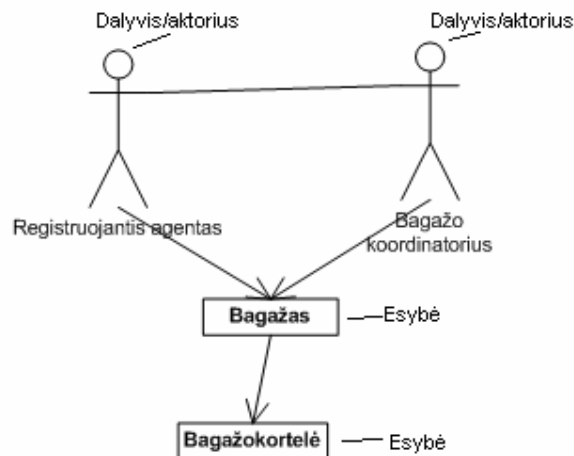
- Veiklos būsenos, kurios vaizduoja veiklos vykdymą (atlikimą), arba žingsnį (poelgį) veiksmų sekoje.
- Perėjimai parodo, kuri veiklos būseną seka paskui prieš tai buvusią. Toks perėjimo tipas vadinamas baigtiniu perėjimu, nes jis nereikalauja detalios įvykio įvykimo priežasties (paaiškinimo); perėjimas sukeliamas veiklos, kurią vaizduoja veiklos būseną, baigimu.
- Sprendimai, kuriems yra nustatytas tikrinimo sąlygų rinkinys. Tikrinimo sąlygos kontroliuoja, kuris perėjimas įvyks iš alternatyvių perėjimų rinkinio, kai tik veikla bus įvykdyta. Taip pat galima naudoti sprendimų simbolį. Sprendimai ir tikrinimo sąlygos leidžia parodyti alternatyvius žingsnius verslo panaudojimo atvejų veiksmų sekoje.
- Synchronizacijos juosta (*bar*), kuri parodo lygiagrečiai vykstančius subsrautus. Jie leidžia parodyti tuo pačiu metu vykstančius žingsnius (*threads*).

Šioje diagramoje neparodyta, kas atlieka nustatytą veiklą, arba ką veikla naudoja ir gamina; visa tai yra verslo objektų modelio dalis.

Režiuuojant, pirma verslo modelio dalis yra verslo panaudojimo atvejų modelis. Jis susideda iš vienos arba daugiau panaudojimo atvejų diagramų, kuriose yra vienas arba daugiau verslo panaudojimo atvejų. Verslo panaudojimo atvejai yra dokumentuojami specifikacijose, kurios sudarytos iš tekstinės dalies (svarbiausia: darbų sekos aprašymas) ir grafinės (veiklos diagramos). Verslo panaudojimo atvejų modelis leidžia sudaryti didelį verslo aktorių perspektyvų vaizdą.

3.2.2. Verslo objektų modelis

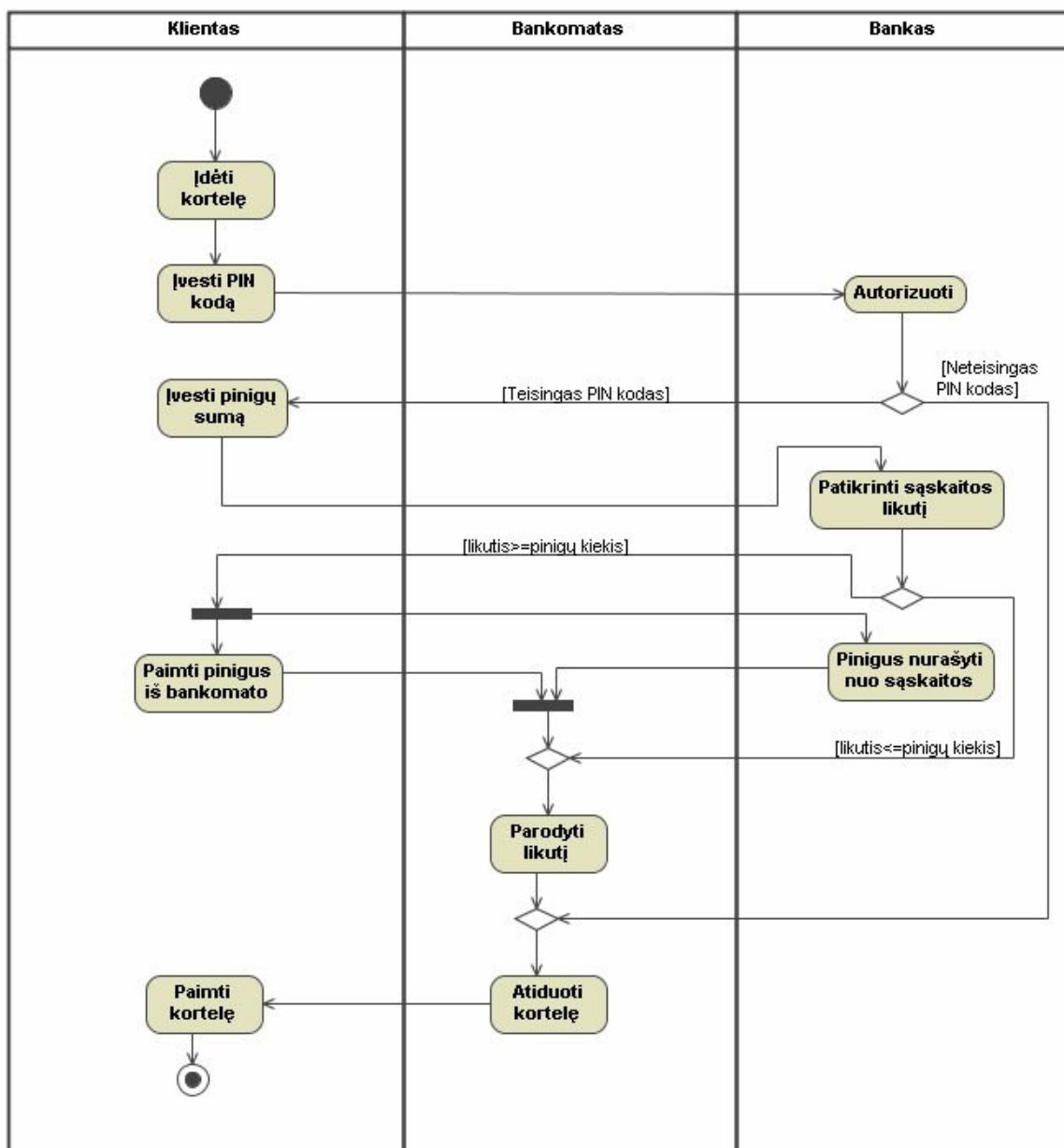
Antra dalis verslo modelio yra verslo objektų modelis. Verslo panaudojimo atvejų modelis nusako, ką verslo procesas atliks. Verslo objekto modelis nusako, kaip tai bus atlikta [10]. Šis modelis kaip abstrakcija, rodanti, kaip verslo dalyviai ir esybės turėtų būti susieti ir kokia tvarka turėtų bendradarbiauti, kad verslas vyktų. 6 pav. pateikta verslo objektų modelio dalis. Tai verslo klasių diagrama, parodanti verslo darbuotojus ir verslo esybes.



6 pav. Verslo klasių diagrama, parodanti ryšius tarp darbuotojų ir verslo esybių

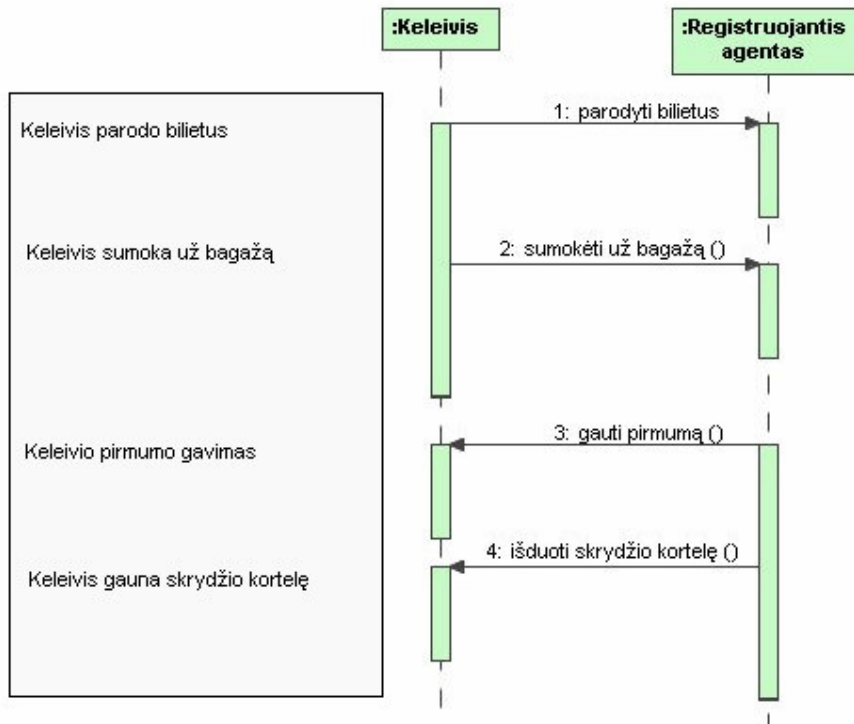
Ši diagrama parodo, kaip darbuotojai susiję vienas su kitu ir su esybėmis susijusiomis su verslu.

Kitas diagramos tipas, naudojamas verslo objektams modeliuoti, yra veiklos diagrama pateikta 7 pav. Šiuo atveju diagramoje naudojamos juostos arba stulpeliai, kurie parodo, kuris verslo darbuotojas atlieka kiekvieną veiklą. Tokiu būdu detalčiau parodomas verslo panaudojimo atvejų vykdymas.



7 pav. Veiklos diagrama rodo, kas kokias veiklas atlieka

Trečias diagramos tipas naudojamas verslo objektų modeliavimui yra verslo sekų diagrama. Sekų diagrama vaizdžiai aprašo sąveikas tarp verslo darbuotojų ir verslo aktorių, ir taip pat parodo kokia prieiga prie verslo esybių, kurios buvo apibrėžtos apibūdinant verslo panaudojimo atvejus. Tokia sekų diagrama kaip pavaizduota 8 pav. trumpai apibūdina ką daro bendradarbiaujantys verslo darbuotojai, kaip jie komunikuoja siųsdami vienas kitam pranešimus, ir kaip valdomos susiję verslo esybės. Sekų diagrama taip pat gali parodyti verslo aktorių bendradarbiavimą su verslo sistema.



8 pav. Verslo sekų diagrama rodo verslo aktorių bendradarbiavimą su verslo darbuotojais

Visos šios diagramos kartu sudaro verslo objektų modelį. Šis modelis pateikia detalią informaciją, kaip verslo procesas yra vykdomas.

4. VERSLO PROCESŲ, APRAŠYTŲ BPMN NOTACIJA, TRANSFORMAVIMO Į PLA MODELIOUS METODIKA

4.1. BPMN notacijos parinkimas modeliuojant verslo procesus

Kadangi darbo tikslas yra transformuoti verslo procesus, aprašytus neformalia modeliavimo kalba į formalų metodą, tai pirmiausia, reikėjo išanalizuoti nors keletą verslo procesų modeliavimo kalbų ir nustatyti, kuri gi verslo procesų modeliavimo kalba yra geresnė, norint aprašyti verslo procesus. Išanalizavus verslo procesų modeliavimo kalbas (UML ir BPMN) buvo nustatyti šių modeliavimo kalbų privalumai ir trūkumai, kurių apibendrinimai yra žemiau.

BPMN yra naujas verslo procesų modeliavimo standartas, skirtas modeliuoti verslo procesų srautus ir web servisus. Jis suteikia galimybę suprasti vidinius verslo procesus grafiškai juos vaizduojant ir suteikia galimybę šiuos procesus standartizuoti. BPMN procesų modeliavimo pagrindas yra metodologija, kuri suteikia galimybę žmonėms iš skirtingų įmonių suprasti vieni kitų verslo procesus, sujungti verslo procesus, ar juos suskaidyti į atskirus. Daugelis verslo analitikų neneigia, kad UML notacija nėra pilnai pritaikyta verslo procesams modeliuoti. Taip yra ne dėl to, kad netinka

simbolių formos ar linijos, o todėl, kad bet kokių atveju UML metodologija buvo orientuota programinės įrangos kūrėjams, o ne procesų valdymui.

BPMN turi nemažai privalumų, naudingų verslo procesų modeliavimui, lyginant su UML (*Unified Modeling Language*). Pirma, ji pateikia proceso srauto modeliavimo metodą, kuris palengvina verslo analitikų modeliavimo procesą. Antra, matematinio pagrindo dėka, ji lengvai vaizduojama verslo vykdomosiomis kalbomis, kai tuo tarpu UML tokios galimybės nepalaiko.

Esminiai BPMN privalumai:

- Nepriklausoma metodologija sudėtingų verslo procesų atvaizdavimo atžvilgiu.
- Lengvai išreiškiama vykdomosiomis kalbomis (BPML, BPEL4WS).
- Prieinama ir vartotina verslo visuomenės aplinkoje.
- Naudinga vaizduojant sudėtingus BPML procesus.
- Naudinga vaizduojant sąveikas tarp organizacijų B2B (*business-to-business*) situacijose.
- Palaiko tik tas modeliavimo sąvokas, kurios susijusios su verslo procesais.

UML ir BPMN notacijos daugeliui modelių užtikrina panašius sprendimus, nusako, kokios artimos yra UML ir BPMN. Abi notacijos dalijasi daugeliu panašių grafinių formų, tiems patiems tikslams atvaizduoti (pvz., užapvalinti stačiakampiai - veiklų vaizdavimui, rombai – sprendimams ir t.t.). Vienas iš skirtumų galėtų būti skirtinga abiejų diagramų terminologija. Pavyzdžiui, veiklos diagrama turi pradinį tašką, o verslo proceso diagrama turi pradžios įvykį. Duomenų srautų vaizdavimas skiriasi abiejose notacijose. UML veiklos diagramoje duomenys griežtai susieti su veiklomis, kai tuo tarpu BPMN notacijoje jie tiesiogiai neveikia proceso srauto ir tarpusavyje nesusiejami. Abu standartai buvo kuriami, siekiant išspręsti bendrą procesinių verslo procesų modeliavimo problemą. BPMN visos pastangos nukreiptos į verslo atstovams tinkamą verslo procesų modeliavimo notaciją. UML notacija, iš anksčiau žinoma, kaip geriau taikytina programinės įrangos kūrėjams. Ir pastangos ją naudoti verslo procesams modeliuoti, labiausiai atsispindi veiklos diagramos galimybėse. Nors UML 2.0 versijoje veiklos diagrama ryškiai atnaujinta verslo atstovams, modeliuojantiems verslo procesus. Yra didelė tikimybė, kad ateityje šios notacijos bus apjungtos, tiksliau UML bus išplėsta BPMN stereotipais.

Svarbiausia, verslo dalyviams turėti lengvai suprantamą verslo proceso modeliavimo notaciją. Taip pat ne mažiau svarbu, kad XML kalbos, tokios kaip BPEL4WS (*Business Process Execution Language For Web Services*) ir BPML (*Business Process Modeling Language*), sukurtos verslo procesų vykdymui, būtų lengvai vaizduojamos bendra notacija.

Kadangi BPMN notacija geriau tinka, norint aprašyti verslo procesus, negu UML, nes BPMN notacija ir buvo sukurta tam tikslui, kad aprašinėti verslo procesus, tai buvo pasirinkta BPMN notacija verslo procesų aprašymui.

4.2. Agregatinės specifikacijos aprašymas

4.2.1. Formalių specifikacijų metodai

Formalių specifikacijų metodai skiriasi pagal tai, kokioje paradigmoje specifikacija iš tikrųjų yra [14]:

Istorija grįsta specifikacija. Čia sistema specifikuojama charakterizuojant maksimalią aibę priimtinių istorijų (elgesių). Savybės yra specifikuojamos pagal laikinus loginius sprendinius apie sistemos objektus. Tokie sprendiniai priverčia operatorių kreiptis į praeities, esamas ir ateities būsenas. Sprendiniai yra interpretuojami laiko rėmų struktūrose.

Būsenomis pagrįsta specifikacija. Čia vietoje charakterizuojant priimtiną istorijas yra charakterizuojamos priimtinos sistemos būsenos. Savybės yra specifikuojamos invariantais iš momentinių sistemos objektų būsenų ir iš momentinių prieš tai einančių ir po to einančių sistemos operacijų sprendinių. Prieš tai einantys sprendiniai paima silpniausią reikalingą sąlygą įėjimo būsenoje tai operacijai atlikti, po to einantys sprendiniai – stipriausią sąlygą išėjimo būsenoje jei ta operacija buvo atlikta. Šia paradigma remiasi tokios kalbos kaip Z, VDM ar B. Orientuoti į objektus šio metodo pataisymai taip pat buvo pasiūlyti.

Perėjimais pagrįsta specifikacija. Čia yra specifikuojami sistemos perėjimai iš būsenos į būseną. Savybės yra specifikuojamos kaip aibė perėjimo funkcijų būsenų automate. Perėjimo funkcija kiekvienam sistemos objektui duoda įėjimo būseną ir trigerių įvykį bei atsakančią išėjimo būseną. Trigerio įvykio atsiradimas yra pakankama sąlyga perėjimui prasidėti.

Funkcinė specifikacija. Pagrindinis principas yra specifikuoti sistemą kaip matematinių funkcijų kolekciją. Funkcijos gali būti grupuojamos pagal jų objektų tipus arba pagal abstrakčius duomenų tipus. Kitu atveju, funkcijos gali būti grupuojamos į logines teorijas. Tokios teorijos turi tokią informaciją kaip tipų apibrėžimus, kintamųjų apibrėžimus, aksiomų apibrėžimus.

Kokį specifikuojimo būdą pasirinkti priklauso nuo aibės kriterijų, kurie realiai yra visapusiškai priklausomi ir netgi konfliktuojantys. Realiai pasirinkimas daugiau priklauso nuo specifikuotojų prioritetų specifikuojamai konkrečiai sistemai.

4.2.2. Formalios specifikacijos apibrėžimas, savybės

Apskritai, formali specifikacija – tai sistemos savybių rinkinio išraiška kažkokoje formalioje kalboje, kažkokiame abstrakcijos lygyje [14]. Tikslus apibrėžimas galimas tik tada, kai tiksliai žinome: kas slepiasi po žodžiu „sistema“, kokios savybės yra įdomios, koks abstrakcijos lygis bus naudojamas ir galiausiai, kokia formali kalba bus naudojama.

Žodis „formalus“ dažnai yra maišomas su žodžiu „tikslus“. Aišku, kad pastarasis paveldi pirmąjį, bet tik ne atvirkščiai. Specifikacija yra formali, jei kalba, kuria ji parašyta yra sudaryta laikantis trijų taisyklių:

- Aiškios ir griežtos sintaksės taisyklės.
- Taisyklės aprašančios semantiką.
- Taisyklės, kuriomis galima išvelgti naudingą informaciją specifikacijoje.

Norint parašyti teisingą specifikaciją, reikia įvertinti šiuos aspektus:

- Specifikacija turi būti adekvati – ji turi labai tiksliai nusakyti problemą.
- Specifikacija turi būti nuosekli – jei paimsime visas specifikuotas savybes į visumą, ta turi būti teisinga.
- Specifikacija turi būti nedviprasmiška – negali turėti dviprasmybių suvokiant kuri nors teiginį kaip tiesą.
- Specifikacija turi būti pilna – žemesnio lygio savybių rinkinys, turi būti pakankamas, kad būtų galima apibūdinti aukštesnio lygio teiginį.
- Specifikacija turi būti minimali – negali turėti perteklinės informacijos, ar savybių kurios nesusijusios su problema.

4.2.3. Verslo procesų formalizavimas, naudojant PLA modelį

Atkarpomis tiesinių agregatų (PLA) formalizavimas naudojamas, kuriant dinامينius verslo procesų modelius [15]. Šio formalaus metodo ypatumai:

- Galima paruošti analizuojamos sistemos formalius aprašymus turint tik vieną reikšmę.
- Modeliavimo ypatybės gali būti analizuojamos naudojant matematinės įrangos techniką.
- Formalus aprašymų metodas yra teorinis pagrindas kuriant programinės įrangos priemones formalių specifikacijų kompiuteriniam analizavimui (tikrinimui, kontrolei, modeliavimui).

Dauguma egzistuojančių semantinių modelių, kalbų ir logikų, skirtų aprašymui ir samprotavimams apie laiko sistemas, netiesiogiai peržiūri kintančios, momentinių diskrečių veiksmų ir tolydžių fazių, sekos vykdymą. Formalus aprašymas, kiekvieną taip aprašytą sistemą gali susieti su perėjimo sistema ar automatu, sudarytais iš būsenų rinkinio, pradinių būsenų rinkinio, diskrečių veiksmų rinkinio, diskrečių žingsnių rinkinio ($s' \xrightarrow{a} s$ parodo, kad iš būsenos s' sistema gali akimirksniu pereiti į būseną s per diskretų įvykį a'), laiko žingsnių rinkinio ($s' \xrightarrow{d} s$ parodo, kad iš būsenos s' sistema gali pereiti į būseną s per laiko tarpą d , kurio metu neįvyksta diskrečių veiksmų). Šios perėjimo sistemos teikia labai abstraktų originalios sistemos elgesio vaizdą, kuriame daug

aspektų, tokių kaip lygiagrečių komponentų skaičius, bendravimas tarp jų, sistemos plėtojimosi būdas besitęsiančių fazių metu ir t. t., nėra pateikiami.

Atkarpomis tiesiniai agregatai (PLA) yra automatinio modelio atskiras atvejis. Šiuo metodu sistemos funkcionavimas formalizuojamas taip [14]:

1. Įėjimų signalų aibė X – čia kiekvienam agregato įėjimui yra priskiriama konkreti perduodamo signalo duomenų struktūra.
2. Išėjimų signalų aibė Y – čia kiekvienam agregato išėjimui yra priskiriama konkreti perduodamo signalo duomenų struktūra.
3. Išorinių įvykių aibė E' – įvykiai, įvykę agregato išorėje, bet veikiantys agregato būseną.
4. Vidinių įvykių aibė E'' – įvykiai, įvykę agregato viduje ir keičiantys agregato būseną.
5. Valdymo seka $e_i'' \mapsto \{\xi_j^i\}, j = \overline{1, \infty}$.
6. Diskrečioji būsenos dedamoji $v(t_m)$ – aibė agregato elementų būsenų reikšmių, galinčių įgyti diskretines reikšmes.
7. Tolydžioji būsenos dedamoji $z_v(t_m)$ – aibė agregato elementų, galinčių įgyti tolydines reikšmes.
8. Pradinė būseną $v(t_0), z_v(t_0)$ – diskretinių ir tolydinių būsenos vektoriaus komponentų pradinės reikšmės.
9. Perėjimo ir išėjimo operatoriai: $H(e_i), G(e_i)$ – algoritmai (sakinių sekos), pagal kuriuos keičiama agregato būseną ir išduodami signalai į agregato išorę.

Agregato būseną gali pasikeisti tik dviem atvejais: kai įėjimo signalas x_i patenka į agregatą ar kai tolydusis komponentas įgyja nustatytą reikšmę.

Agregato funkcionavimas nagrinėjamas diskrečiais laiko momentais, kurie priklauso aibei $T = \{t_0, t_1, \dots, t_m, \dots\}$. Tais laiko momentais gali įvykti vienas ar keli įvykiai, kurie sukelia agregato būsenos pasikeitimą. Agregato įvykių aibę $E = E' \cup E''$ sudaro įvykiai $E' = \{e_1', e_2', \dots, e_N'\}$, kurie įvyksta dėl įėjimo signalų atėjimo, ir įvykiai $E'' = \{e_1'', e_2'', \dots, e_N''\}$, kurie fiksuoja operacijų pabaigą. Kiekvienam vidiniam įvykiui priskiriama valdymo seka: $e_i'' \mapsto \{\xi_j^i\}, j = \overline{1, \infty}$.

Tolydžiosios dedamosios koordinatės $z_v(t_m)$ apibrėžia laiko momentus, kada aggregate galimi įvykiai, čia $z_v(t_m) = \{w(e_1'', t_m), w(e_2'', t_m), \dots, w(e_f'', t_m)\}$. Be to, visada $w(e_i'', t_m) \geq t_m$, kur $w(e_i'', t_m)$ - operacijos pabaigos momentas. Agregato būseną $z(t_m)$ gali pasikeisti tik diskrečiais laiko momentais $t_m, m = 1, 2, 3, \dots$, likdama pastovi laiko tarpuose $[t_m, t_{m+1}), m = 0, 1, 2, \dots$, čia t_0 - pradinis sistemos funkcionavimo momentas. Kai sistemos būseną yra žinoma $z(t_m), m = 0, 1, 2, \dots$, įėjimo signalo

atvykimas į agregatą arba lygybė $t_{m+1} = \min\{w(e_i'', t_m)\}, 1 \leq i \leq f$ nustato laiko momentą t_{m+1} , kai įvyksta kitas įvykis. Sekančio įvykio e_{m+1} rūšis yra nustatoma įėjimo signalo, jei įėjimo signalas patenka į agregatą laiko momentu t_{m+1} , arba yra nustatomas valdymo koordinatės, turinčios minimalią reikšmę laiko momentu t_m , t. y. koordinatė $w(e_i'', t_m)$ tampa minimalia, $e_{m+1} \in E''$.

Operatorius H apibrėžia naują agregato būseną: $z(t_{m+1}) = H[z(t_m, e_i)], e_i \in E' \cup E''$.

Operatorius G apibrėžia išėjimo signalus: $y = G[z(t_m, e_i)], e_i \in E' \cup E'', y \in Y$.

Apžvelgiant agregatų sistemą, susidedančią iš K agregatų, įskaitant ir išorės agregatus.

Agregato galimų įvykių aibę sudaro vidinių įvykių poaibio sąjunga, t. y.: $E \in \bigcup_{k=1}^K E_k''$, kur E_k'' - k-jo agregato įvykių aibė.

Sekantis laiko momentas t_{m+1} , kai įvyksta aibės E įvykis yra apibrėžiamas:

$$t_{m+1} = \min_k \min_r \{w(e_r'', t_m)\}, 1 \leq k \leq K, e_r'' \in E_k''.$$

Agregato sistemos charakteristikos gali būti nustatytos analizuojant seką:

$$(z(t_0), e_1, z(t_1)), (z(t_2), e_2, z(t_2)), (z(t_2), e_3, z(t_3)), \dots, (z(t_m), e_{m+1}, z(t_{m+1}))) \dots$$

kur $z(t_m)$ - yra agregato sistemos būsena po sistemos sudarymo ir $e_i \in E$.

Diskrečiais laiko momentais t_m , kai sistema juos pakeičia, susikuria nelygybių seka:

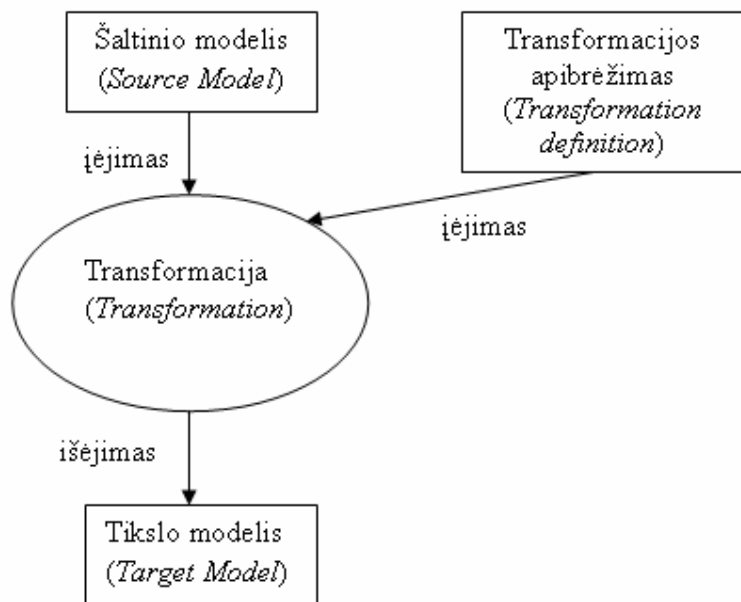
$$t_0 \leq t_1 \leq t_2 \leq \dots \leq t_m \leq t_{m+1} \leq \dots$$

Dažnai tokios charakteristikos kaip: laukimo eilėje laikas, laiko tarpas, kurį pranešimas išbūna sistemoje, ilgis ir t. t. yra naudojami apskaičiuojant agregatų sistemos našumą.

4.3. Verslo procesų transformavimas į PLA modelius

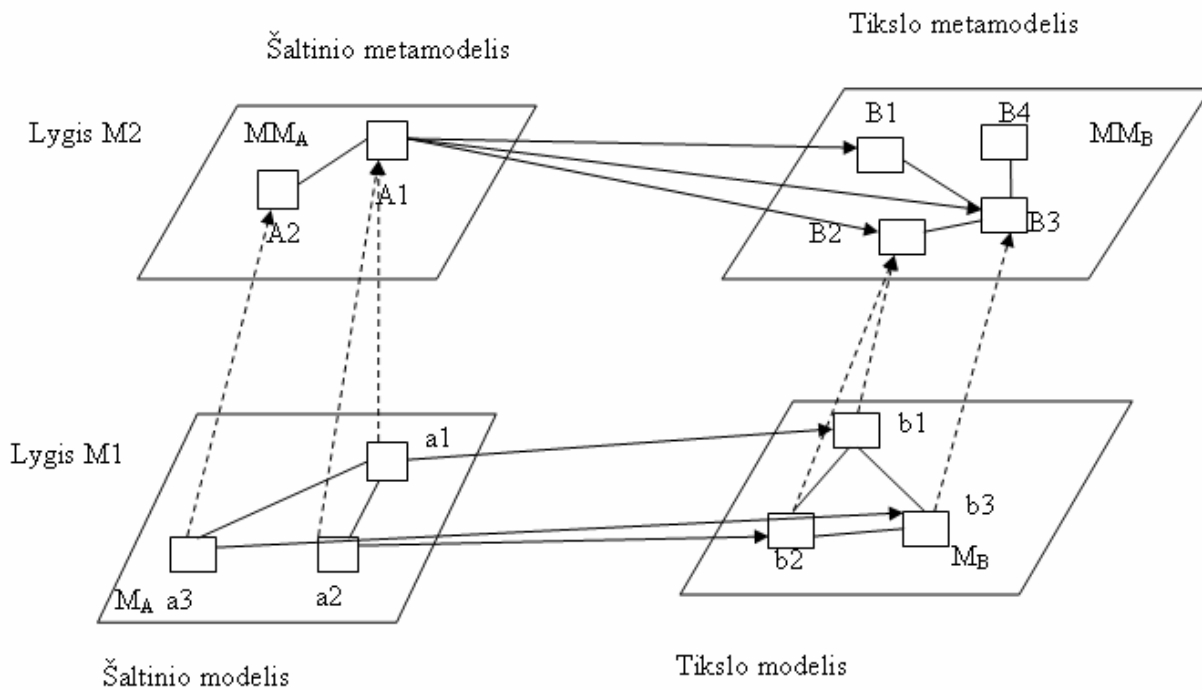
4.3.1. Modelių transformavimas

Modelių transformacija yra procesas, konvertuojantis tikslo (*target*) modelį į šaltinio (*source*) modelį, remiantis transformavimo apibrėžimu, kuris išreiškiamas transformavimo kalba [17] (žr. 9 pav.).



9 pav. Modelių transformavimas

Transformuojant šaltinio modelį į tikslo modelį, reikia susieti šaltinio modelio elementus su tikslo modelio elementais. Be to, vienas elementas šaltinio modelyje gali būti transformuotas į keletą alternatyvių elementų tikslo modelyje. Vienas iš būdų, kaip yra transformuojamas šaltinio modelis į tikslo modelį, naudojant šių modelių metamodelius, yra pateiktas 10 pav.



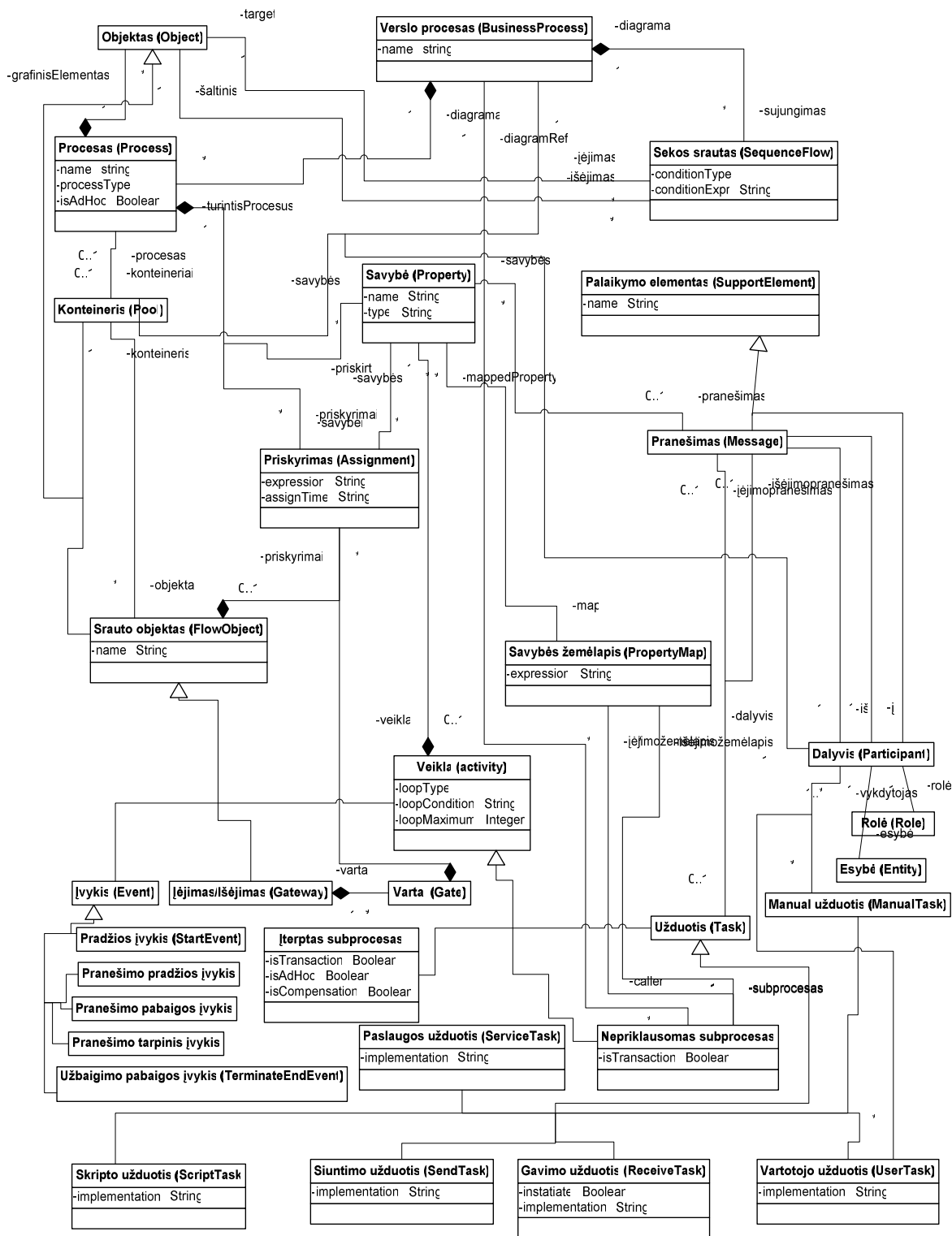
10 pav. Šaltinio modelio transformacija pagrįsta metamodelio transformacija

Šaltinio ir tikslo modeliai yra išsidėstę M1 lygyje, o šaltinio ir tikslo metamodeliai išsidėstę M2 lygyje. Šaltinio modelio M_A elementai yra atvaizduojami į šaltinio metamodelio MM_A elementus. M_A turi tris elementus a1, a2, a3. M_A yra transformuojamas į modelį, kuris yra atvaizduojamas į tikslo metamodelį MM_B . Du metamodeliai yra naudojami, norint nustatyti galimas transformavimo taisykles. MM_A turi du elementus A1, A2. MM_B turi keturis elementus B1, B2, B3, B4.

Šaltinio metamodelio kiekvienas elementas gali atitikti keletą tikslo metamodelio elementų. Elementas A1 šaltinio metamodelyje gali atitikti tris elementus B1, B2, B3 tikslo metamodelyje.

4.3.2. BPMN ir agregato metamodeliai

Kadangi darbo tikslas yra transformuoti verslo procesą, aprašytą BPMN notacija (BPMN modeliu), į PLA modelį, tai atliekant modelių transformavimą, bus reikalingi BPMN metamodelis [16] ir PLA metamodelis [14], nes reikės nustatyti, kokie BPMN metamodelio elementai gali būti transformuojami į kokius agregato metamodelio elementus. BPMN metamodelio fragmentas ir agregato metamodelis yra pateikti atitinkamai 11 ir 12 paveikslėliuose.



11 pav. BPMN metamodelis

Žemiau yra pateikti BPMN metamodelio elementų ryšių paaiškinimai.

Verslo procesas (*BusinessProcess*) savyje turi daug procesų (*Process*), o tas pats procesas gali būti tik vienas viename verslo procese. Taip pat verslo procesas savyje turi daug sekos srautų (*SequenceFlow*), o tas pats sekos srautas gali būti tik vienas tame pačiame verslo procese. Verslo procesas (*BusinessProcess*) siejasi ryšiu (1 su 1..*) su konteineriu (*Pool*), t.y. verslo procesas gali turėti vieną arba daug konteinerių ir tas ryšys vadinamas konteineriais (*pools*). Be to tas pats konteineris gali būti tik vienas tame pačiame verslo procese. Verslo procesas siejasi ryšiu (1 su *) su nepriklausomu subprocesu (*IndependentSubprocess*), t. y. nepriklausomų subprocesų verslo procese gali būti daug, o tas pats nepriklausomas subprocesas gali būti tik vienas tame pačiame verslo procese.

Objektas (*Object*) siejasi dviem ryšiais (1 su *) su sekų srautu (*SequenceFlow*) ir yra vadinami įėjimu (*in*) ir išėjimu (*out*), t.y. objektas turi daug įėjimų ir daug išėjimų. O sekos srautas objektui yra tikslas (*target*) ir šaltinis (*source*).

Procesas savyje turi daug objektų, o objektas taip pat siejasi ryšiu su procesu. Procesas taip pat siejasi ryšiu (0..1) su konteineriu, o konteineris taip pat siejasi ryšiu (0..1) su procesu. Procesas savyje turi daug priskyrimų (*Assignment*), o priskyrimas taip pat siejasi ryšiu su procesu. Procesas savyje turi ir daug savybių (*Property*), o savybė siejasi ryšiu (0..1) su procesu.

Konteineris gali turėti daug srauto objektų (*FlowObject*), o tas pats srauto objektas konteineryje gali būti tik vienas. Konteineris siejasi ryšiu (1..* su 1) su dalyviu. Objektas paveldi konteinerį ir srauto objektą.

Srauto objektas paveldi įvykio elementą, vartų (*Gateway*) elementą, veiklos (*Activity*) elementą. Įvykis (*Event*) paveldi pradžios įvykį (*StartEvent*), pranešimo pradžios įvykį (*MessageStartEvent*), pranešimo pabaigos įvykį (*MessageEndEvent*), pranešimo tarpinį įvykį (*MessageIntermedEvent*), užbaigimo pabaigos įvykį (*TerminateEndEvent*).

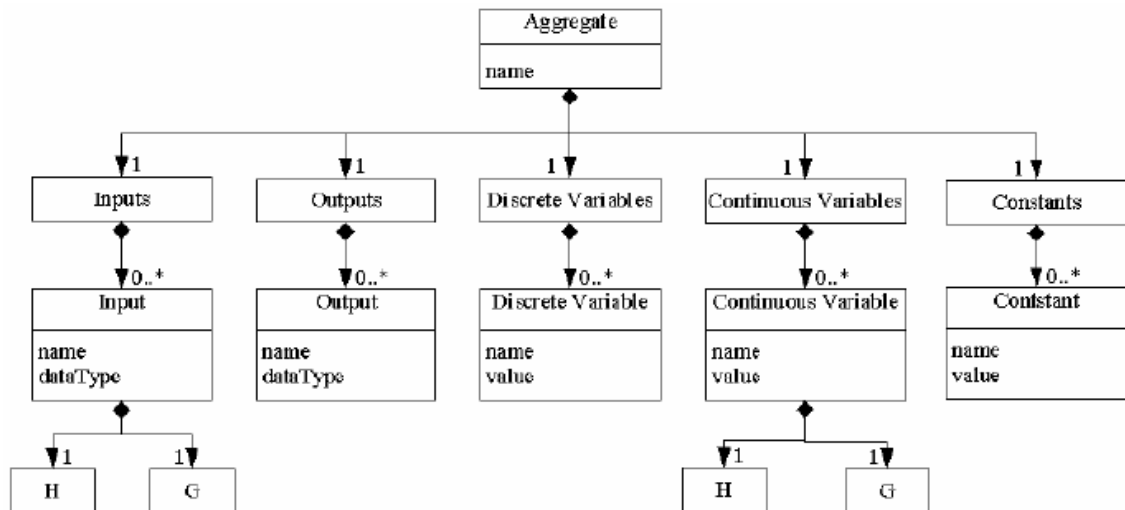
Veikla savyje turi daug savybių, o savybė siejasi ryšiu su veikla. Veikla paveldi įterptinį subprocesą, užduotį, nepriklausomą subprocesą.

Savybė siejasi ryšiu (* su 1) su priskyrimu (*Assignment*). Taip pat savybė siejasi ryšiu (* su 0..1) su pranešimu (*Message*) ir taip pat siejasi ryšiu su nepriklausomu subprocesu (*IndependentSubprocess*).

Pranešimas siejasi dviem ryšiais su užduotimi (*Task*), kur vienas ryšys yra įėjimas, o kitas išėjimas. Pranešimas taip pat siejasi ryšiais su dalyviu (*Participant*), kur vienas ryšys yra į, o kitas iš.

Support elementas (*SupportElement*) paveldi pranešimą ir dalyvį. Dalyvis siejasi ryšiu su role (*Role*), su esybe (*Entity*), su *manual* užduotimi (*ManualTask*) ir su vartotojo užduotimi (*UserTask*).

Užduotis (*Task*) paveldi paslaugos užduotį (*ServiceTask*), skripto užduotį (*ScriptTask*), siuntimo užduotį (*SendTask*), gavimo užduotį (*ReceiveTask*), vartotojo užduotį (*UserTask*).

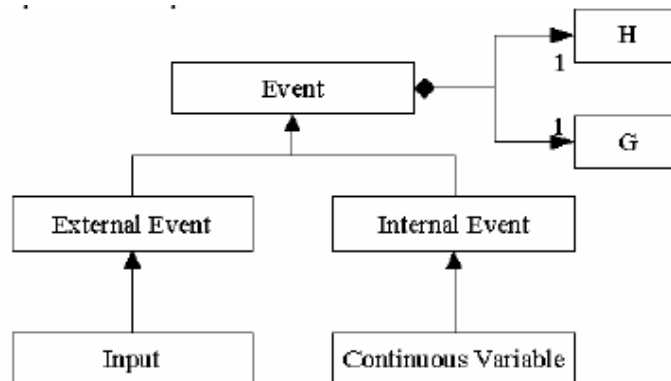


12 pav. Agregato metamodelis

Agregatas susideda iš:

1. Įėjimų aibės, kurioje gali būti neribotas kiekis įėjimų. Įėjimą aprašo: pavadinimas ir tipas, kuris rodo, kokio tipo užklauskos gali patekti į agregatą per įėjimą.
2. Išėjimo aibės, kurioje gali būti neribotas kiekis išėjimų. Išėjimą aprašo: pavadinimas ir tipas, kuris rodo, kokio tipo užklauskos yra siunčiamos iš agregato per išėjimą.
3. Diskrečiųjų kintamųjų aibės, kuriai gali priklausyti neribotas kiekis diskrečiųjų kintamųjų. Diskretūs kintamieji naudojami aprašyti agregato būsenos dedamąsias.
4. Tolydžiuųjų kintamųjų aibės, kurioms gali priklausyti neribotas kiekis tolydžiuųjų kintamųjų. Tolydūs kintamieji yra naudojami aprašyti laiko momentus, kuriais įvyksta vidiniai agregato įvykiai pasibaigus operacijoms.

Agregato įėjimų signalai iššaukia išorinius įvykius. Vidiniai ir išoriniai įvykiai gali keisti agregato vidinę būseną – šiuos pokyčius aprašo H (perėjimo) operatorius; ir gali keisti signalus – šiuos pokyčius aprašo G (išėjimo) operatorius.



13 pav. H ir G operatoriai

4.3.3. Transformavimo metodika

Remiantis modelių transformavimo teorija [17], šiuo atveju, reikia verslo procesą, aprašytą BPMN notacija (BPMN modelių) transformuoti į PLA modelį. Pagal 10 pav., šaltinio modelis yra BPMN modelis, kurį turime, o tikslo modelis yra PLA modelis, kurį reikia gauti, naudojant transformaciją. Atliekant transformaciją, pirmiausia, reikia nustatyti kokius BPMN metamodelio elementai, turės būti transformuojami į kokius agregato metamodelio elementus (žr. lentelę Nr. 6).

Lentelė Nr. 6 BPMN metamodelio elementų transformavimas į agregato metamodelio elementus

BPMN metamodelio elementas	Agregato metamodelio elementas	Transformavimo paskirtis
Konteineris (<i>Pool</i>)	Agregatas (<i>Aggregate</i>)	BPMN metamodelio elementas konteineris transformuojamas į agregatą agregato metamodelyje
Įeinantis srautas (Pranešimo srautas (<i>Message Flow</i>), Sekos srautas (<i>Sequence Flow</i>), Krypties asociacija (<i>Directional Association</i>))	Įėjimo signalas (<i>input</i>)	BPMN metamodelio elementas įeinantis srautas transformuojamas į įėjimo signalą agregato metamodelyje
Išeinantis srautas (Pranešimo srautas (<i>Message Flow</i>), Sekos srautas (<i>Sequence Flow</i>), Krypties asociacija (<i>Directional Association</i>))	Išėjimo signalas (<i>output</i>)	BPMN metamodelio elementas išeinantis srautas transformuojamas į išėjimo signalą agregato metamodelyje
Įvykis (<i>Start/Intermediate Event</i>)	1. Išorinis įvykis (<i>external event</i>). 2. Vidinis įvykis (<i>internal event</i>).	BPMN metamodelyje įvykis transformuojamas į išorinį įvykį arba gali būti transformuojamas ir į vidinį įvykį agregato metamodelyje
Užduotis (<i>Task</i>)	1. Vidinis įvykis 2. Išorinio įvykio perėjimo operatorius. 3. Vidinio įvykio perėjimo operatorius.	Užduotis BPMN metamodelyje gali būti transformuojama į vidinį įvykį, išorinio įvykio perėjimo operatorių ir į vidinio įvykio perėjimo operatorių agregato metamodelyje
Įvykis (<i>Start/Intermediate Event</i>)	Valdančios sekos elementas	Vidinių įvykių skaičius nusako valdančią seką
Užduotis (<i>Task</i>); Įvykis (<i>Start/Intermediate Event</i>)	Tolydžioji agregato būsenos dedamoji	Tolydžioji agregato būsenos dedamoji sudaroma pagal vidinių įvykių skaičių. Kiekvienam vidiniam įvykiui apibrėžiama tolydinė komponentė. Todėl kiekvienai užduočiai arba kiekvienam įvykiui yra apibrėžiama tolydinė komponentė, kuri yra transformuojama į agregato

		tolydinę komponentę.
Neišskirta	Diskrečioji agregato būseną	BPMN modelyje nėra išskirta diskrečioji būseną. Jos kintamųjų prasmė nepateikta jokių komentarų
Pradžios įvykis (<i>Start Event</i>)	Pradinė būseną	Nustatoma agregato pradinė būseną

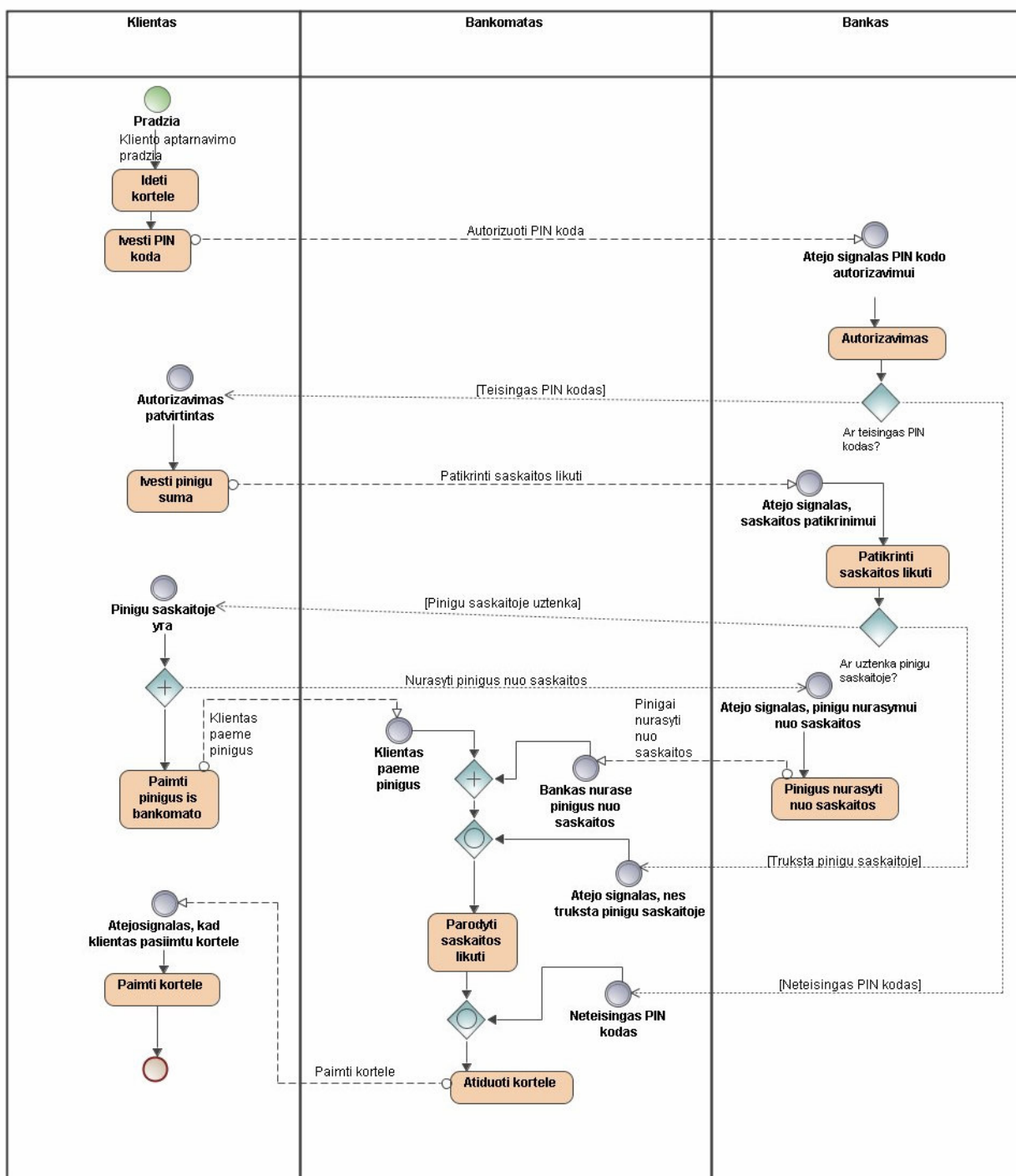
Konteineris transformuojamas į agregatą. Konteinerio vardas transformuojamas į agregato pavadinimą. Įeinantis pranešimų srautas į konteinerį yra transformuojamas į įėjimo signalą į agregatą. Išeinantis pranešimų srautas iš konteinerio transformuojamas į išėjimo signalą iš agregato. Pradžios/Tarpinis įvykis (*Start/Intermediate Event*) transformuojamas į išorinį įvykį arba vidinį įvykį. Išorinis įvykis yra tas į kurį iš kito agregato ateina įėjimo srautas, o vidinis – tas iš kurio į kitą agregatą išeina išėjimo srautas. Užduotis (*Task*) gali būti transformuojama į vidinį įvykį arba į išorinio įvykio arba vidinio įvykio perėjimo operatorius. Vidinių įvykių skaičius nusako valdančios sekos elementą. BPMN metamodelyje nėra išskirta diskrečioji būseną, kuri yra agregato metamodelyje. Jos kintamųjų prasmė nepateikta jokių komentarų, todėl ji nebus gauta transformuojant. Agregato metamodelyje esantys tolydieji kintamieji yra sudaromi iš BPMN metamodelio vidinių įvykių skaičiaus. Kiekvienam vidiniam įvykiui transformuojant yra apibrėžiama tolydinė komponentė. BPMN metamodelyje esantis pradžios įvykis yra transformuojamas į agregato pradinės būsenos dedamąją.

Siekiant atlikti modelių transformavimą (verslo proceso, aprašyto BPMN notacija, transformavimo į PLA modelį), yra pasirenkamas pavyzdys t. y. pinigų išėmimo iš bankomato verslo procesas.

Yra sudaromas pinigų išėmimo iš bankomato verslo proceso konceptualusis modelis.

Pinigų išėmimo iš bankomato konceptualusis modelis: Klientas norėdamas nusiimti pinigų iš savo banko sąskaitos bankomate, pirmiausia į bankomatą įdeda kortelę, tada turi įvesti kortelės PIN kodą. Kai PIN kodas yra įvestas tada bankas tikrina, ar teisingas PIN kodas, jei neteisingas PIN kodas tada yra pranešama, kad neteisingas PIN kodas ir atiduodama kortelė klientui. Jei PIN kodas teisingas, tada klientas turi įvesti pinigų sumą, kurią nori nusiimti iš savo sąskaitos banke. Įvedus pinigų sumą, bankas tikrina sąskaitos likutį, jeigu pinigų sąskaitoje neužtenka, yra parodomas sąskaitos likutis ir atiduodama kortelė, jeigu pinigų užtenka, yra išduodami pinigai iš bankomato ir taip pat yra nurašomi tie pinigai nuo sąskaitos, klientas pasiima pinigus iš bankomato, yra parodomas likutis ir atiduodama kortelė klientui.

Pagal konceptualųjį modelį sudaroma verslo proceso diagrama (BPMN modelis) (žr. 14 pav.), naudojant BPMN notacijos žymėjimus.



14 pav. Verslo proceso pinigų išėmimo iš bankomato aprašymas, naudojant BPMN notaciją

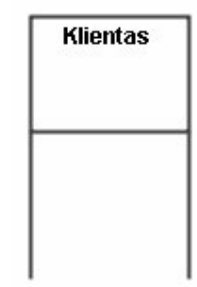


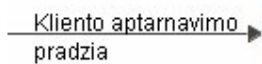


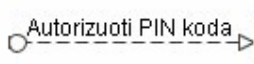
Verslo proceso pinigų išėmimo iš bankomato BPD paaiškinimai yra pateikti žemiau. Diagramoje yra konteineris (*Pool*), kuris suskirstytas į tris juostas (*Lanes*), kurių pavadinimai yra „Klientas“, „Bankomatas“ ir „Bankas“. Pradžios įvykis (*Start Event*) „Pradžia“ nurodo, kur yra verslo proceso pradžia. Iš pradžios įvykio sekos srautas (*Sequence Flow*) „Kliento aptarnavimo pradžia“, nurodo sekančią užduotį (*Task*), kuri turi būti vykdoma, t.y. kurios pavadinimas yra „Įdėti kortelę“;

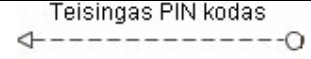


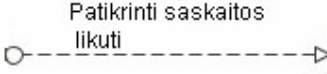
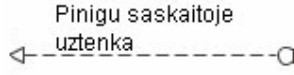

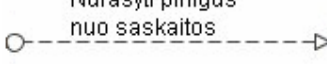

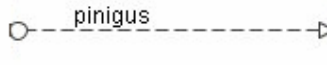
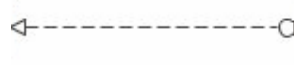


sekanti užduotis turės būti vykdoma „Įvesti PIN kodą“. Iš šios užduoties išeina pranešimų srautas (*Message Flow*) „Autorizuoti PIN kodą“ į tarpinį įvykį (*Intermediate Event*) „Atėjo signalas PIN kodo autorizavimui“, kuris yra konteineryje (*Pool*) „Bankas“, kadangi PIN kodas turės būti autorizuojamas, todėl sekanti užduotis bus vykdoma „Autorizavimas“. Toliau bus tikrinama sąlyga „Ar teisingas PIN kodas?“, jei PIN kodas teisingas tada signalas nueis į įvykį „Autorizavimas patvirtintas“, kuris yra konteineryje „Klientas“, o jei PIN kodas neteisingas, tada signalas nueis į įvykį „Neteisingas PIN kodas“, kuris yra konteineryje „Bankomatas“ ir t.t., taip analizuojant verslo proceso konceptualųjį modelį yra sudaromas BPMN modelis.



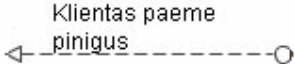

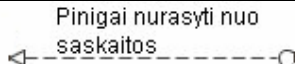
Transformuojant verslo procesą, aprašytą BPMN modeliu, reikia nustatyti kokius BPMN modelio elementai turės būti transformuojami į kokius PLA modelio elementus, t. y. kokius BPMN modelio elementai atitinka kokius PLA modelio elementus, kad būtų lengviau po to rašyti transformavimo taisykles.


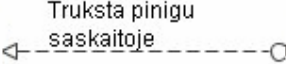


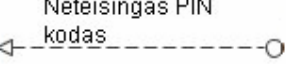


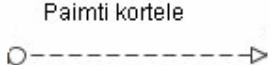
Šiam pavyzdžiui (pinigų išėmimo iš bankomato verslo procesui) yra nustatomi BPMN modelio ir PLA modelio elementų atitikimai (žr. lentelę Nr. 7), tam kad rašant transformaciją būtų aišku kaip rašyti transformavimo taisykles.

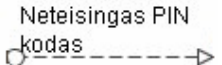
Lentelė Nr. 7 BPMN modelio ir PLA modelio elementų atitikimai



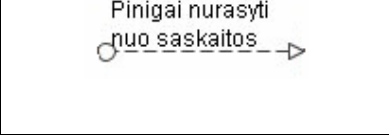
BPMN modelio elementai		PLA modelio elementai		
<i>Žymėjimas</i>	<i>Pavadinimas</i>	<i>Pavadinimas</i>	<i>Žymėjimas</i>	<i>Reikšmė</i>
	Konteineris (<i>Pool</i>)	Agregatas		Klientas
	Pradžios įvykis (<i>Start Event</i>)	Išorinis įvykis	e_1'	Pradžia
	Sekos srautas (<i>Sequence Flow</i>)	Įėjimo signalas	x_1	Kliento aptarnavimo pradžia
	Užduotis (<i>Task</i>)	Vidinis įvykis	e_1''	Įdėti kortelę
	Užduotis (<i>Task</i>)	Vidinis įvykis	e_2''	Įvesti PIN kodą
	Pranešimo srautas (<i>Message Flow</i>)	Išėjimo signalas	y_1	Autorizuoti PIN kodą

	Pranešimo srautas (<i>Message Flow</i>)	Įėjimo signalas	x_2	Teisingas PIN kodas
	Tarpinis įvykis (<i>Intermediate Event</i>)	Išorinis įvykis	e_2'	Autorizavimas patvirtintas
	Užduotis (<i>Task</i>)	Vidinis įvykis	e_3''	Įvesti pinigų sumą
	Pranešimo srautas (<i>Message Flow</i>)	Išėjimo signalas	y_2	Patikrinti sąskaitos likutį
	Pranešimo srautas (<i>Message Flow</i>)	Įėjimo signalas	x_3	Pinigų sąskaitoje užtenka
	Tarpinis įvykis (<i>Intermediate Event</i>)	Išorinis įvykis	e_3'	Pinigų sąskaitoje yra
	Pranešimo srautas (<i>Message Flow</i>)	Išėjimo signalas	y_3	Nurašyti pinigus nuo sąskaitos
	Užduotis (<i>Task</i>)	Vidinis įvykis	e_4''	Paimti pinigus iš bankomato
	Pranešimo srautas (<i>Message Flow</i>)	Išėjimo signalas	y_4	Klientas paėmė pinigus
	Pranešimo srautas (<i>Message Flow</i>)	Įėjimo signalas	x_4	Paimti kortelę
	Tarpinis įvykis (<i>Intermediate Event</i>)	Išorinis įvykis	e_4'	Atėjo signalas, kad klientas paimtų kortelę
	Užduotis (<i>Task</i>)	Vidinis įvykis	e_5''	Paimti kortelę
Kiekvienam vidinių įvykių aibėje esančiam įvykiui yra apibrėžiama valdanti seka, t. y. kiekvienai užduočiai yra apibrėžiama valdančios sekos komponentė		Valdanti seka	$e_1'' \rightarrow \{\xi_{1i}\}, i = 0, \infty$ $e_2'' \rightarrow \{\xi_{2i}\}, i = 0, \infty$ $e_3'' \rightarrow \{\xi_{3i}\}, i = 0, \infty$	

			$e_4'' \rightarrow \{\xi_{4i}\}, i = 0, \infty$ $e_5'' \rightarrow \{\xi_{5i}\}, i = 0, \infty$ ξ_1 - laiko tarpas iki kortelės įdėjimo; ξ_2 - laiko tarpas iki PIN kodo įvedimo; ξ_3 - laiko tarpas iki pinigų sumos įvedimo; ξ_4 - laiko tarpas iki pinigų paėmimo iš bankomato; ξ_5 - laiko tarpas iki kortelės paėmimo
Tolydžioji agregato būsenos dedamoji sudaroma pagal vidinių įvykių skaičių. Kiekvienam vidiniam įvykiui apibrėžiama tolydinė komponentė, t.y. kiekvienai užduočiai yra apibrėžiama tolydinė komponentė	Tolydžioji agregato būsenos dedamoji	$w(e''_1, t_m)$ – įdėjimo kortelės laiko momentas; $w(e''_2, t_m)$ – PIN kodo įvedimo laiko momentas; $w(e''_3, t_m)$ – pinigų įvedimo laiko momentas; $w(e''_4, t_m)$ – paėmimo pinigų laiko momentas; $w(e''_5, t_m)$ – kortelės paėmimo laiko momentas	
BPMN modelyje nėra išskirta diskrečioji būseną. Jos kintamųjų prasmė nepateikta jokių komentarų	Diskrečioji būseną	$v(t_m) = \emptyset$	
	Konteineris (Pool)	Agregatas	 Bankomatas
 Klientas paėmė pinigus	Pranešimo srautas (Message Flow)	Įėjimo signalas	x_1 Klientas paėmė pinigus
 Klientas paėmė pinigus	Tarpinis įvykis (Intermediate Event)	Išorinis įvykis	e_1' Klientas paėmė pinigus
 Pinigai nurašyti nuo sąskaitos	Pranešimo srautas (Message Flow)	Įėjimo signalas	x_2 Pinigai nurašyti nuo sąskaitos

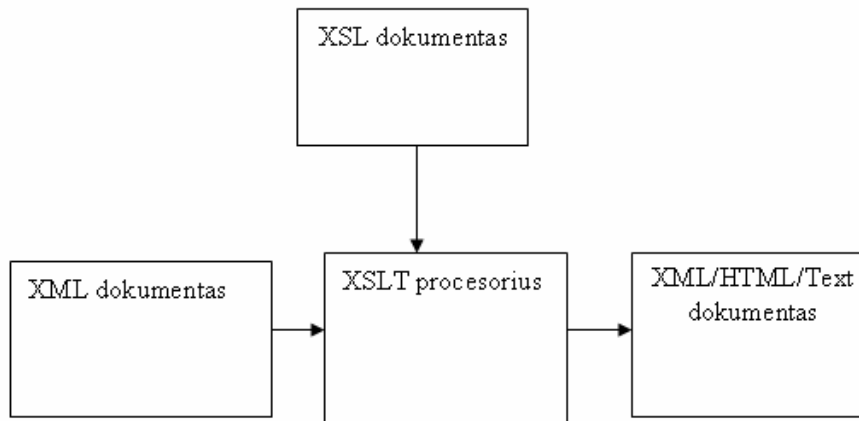
	Tarpinis įvykis (<i>Intermediate Event</i>)	Išorinis įvykis	e_2'	Bankas nurašė pinigus nuo sąskaitos
	Pranešimo srautas (<i>Message Flow</i>)	Įėjimo signalas	x_3	Trūksta pinigų sąskaitoje
	Tarpinis įvykis (<i>Intermediate Event</i>)	Išorinis įvykis	e_3'	Atėjo signalas, kad trūksta pinigų sąskaitoje
	Užduotis (<i>Task</i>)	Vidinis įvykis	e_1''	Parodyti sąskaitos likutį
	Pranešimo srautas (<i>Message Flow</i>)	Įėjimo signalas	x_4	Neteisingas PIN kodas
	Tarpinis įvykis (<i>Intermediate Event</i>)	Išorinis įvykis	e_4'	Atėjo signalas, kad trūksta pinigų sąskaitoje
	Užduotis (<i>Task</i>)	Vidinis įvykis	e_2''	Atiduoti kortelę
	Pranešimo srautas (<i>Message Flow</i>)	Išėjimo signalas	y_1	Paimti kortelę
Kiekvienam vidinių įvykių aibėje esančiam įvykiui yra apibrėžiama valdanti seka, t.y. kiekvienai užduočiai yra apibrėžiama valdančios sekos komponentė		Valdanti seka	$e_1'' \rightarrow \{\xi_{1i}\}, i = 0, \infty$ $e_2'' \rightarrow \{\xi_{2i}\}, i = 0, \infty$ ξ_1 - laiko tarpas iki sąskaito likučio parodymo; ξ_2 - laiko tarpas iki kortelės atidavimo	
Tolydžioji agregato būsenos dedamoji sudaroma pagal vidinių įvykių skaičių. Kiekvienam vidiniam įvykiui apibrėžiama tolydinė komponentė, t.y. kiekvienai užduočiai yra apibrėžiama tolydinė komponentė		Tolydžioji agregato būsenos dedamoji	$w(e_1'', t_m)$ – sąskaitos likučio parodymo laiko momentas; $w(e_2'', t_m)$ – kortelės atidavimo laiko momentas	
BPMN modelyje nėra išskirta diskrečioji būseną. Jos kintamųjų prasmė nepateikta jokių komentarų		Diskrečioji būseną	$v(t_m) = \emptyset$	

	Konteineris (<i>Pool</i>)	Agregatas		Bankas
	Pranešimo srautas (<i>Message Flow</i>)	Įėjimo signalas	x_1	Autorizuoti PIN kodą
	Tarpinis įvykis (<i>Intermediate Event</i>)	Išorinis įvykis	e_1'	Atėjo signalas PIN kodo autorizavimui
	Užduotis (<i>Task</i>)	Vidinis įvykis	e_1''	Autorizavimas
	Pranešimo srautas (<i>Message Flow</i>)	Išėjimo signalas	y_1	Teisingas PIN kodas
	Pranešimo srautas (<i>Message Flow</i>)	Išėjimo signalas	y_2	Neteisingas PIN kodas
	Pranešimo srautas (<i>Message Flow</i>)	Įėjimo signalas	x_2	Patikrinti sąskaitos likutį
	Tarpinis įvykis (<i>Intermediate Event</i>)	Išorinis įvykis	e_2'	Atėjo signalas sąskaitos patikrinimui
	Užduotis (<i>Task</i>)	Vidinis įvykis	e_2''	Patikrinti sąskaitos likutį
	Pranešimo srautas (<i>Message Flow</i>)	Išėjimo signalas	y_3	Pinigų sąskaitoje užtenka
	Pranešimo srautas (<i>Message Flow</i>)	Išėjimo signalas	y_4	Trūksta pinigų sąskaitoje
	Pranešimo srautas (<i>Message Flow</i>)	Įėjimo signalas	x_3	Nurašyti pinigus nuo sąskaitos

	Tarpinis įvykis (<i>Intermediate Event</i>)	Išorinis įvykis	e_3'	Atėjo signalas, pinigų nurašymui nuo sąskaitos
	Užduotis (<i>Task</i>)	Vidinis įvykis	e_3''	Pinigus nurašyti nuo sąskaitos
	Pranešimo srautas (<i>Message Flow</i>)	Išėjimo signalas	y_5	Pinigai nurašyti nuo sąskaitos
Kiekvienam vidinių įvykių aibėje esančiam įvykiui yra apibrėžiama valdanti seka, t.y. kiekvienai užduočiai yra apibrėžiama valdančios sekos komponentė		Valdanti seka	$e_1'' \rightarrow \{\xi_{1i}\}, i = 0, \infty$ $e_2'' \rightarrow \{\xi_{2i}\}, i = 0, \infty$ $e_3'' \rightarrow \{\xi_{3i}\}, i = 0, \infty$ ξ_1 - laiko tarpas iki autorizavimo; ξ_2 - laiko tarpas iki sąskaitos likučio patikrinimo; ξ_3 - laiko tarpas iki pinigų nurašymo nuo sąskaitos	
Tolydžioji agregato būsenos dedamoji sudaroma pagal vidinių įvykių skaičių. Kiekvienam vidiniam įvykiui apibrėžiama tolydinė komponentė, t.y. kiekvienai užduočiai yra apibrėžiama tolydinė komponentė		Tolydžioji agregato būsenos dedamoji	$w(e''_1, t_m)$ – autorizavimo laiko momentas; $w(e''_2, t_m)$ – sąskaitos patikrinimo laiko momentas; $w(e''_3, t_m)$ – pinigų nurašymo laiko momentas	
BPMN modelyje nėra išskirta diskrečioji būseną. Jos kintamųjų prasmė nepateikta jokių komentarų		Diskrečioji būseną	$v(t_m) = \emptyset$	

Kadangi konkrečiam pavyzdžiui, yra nustatyti BPMN modelio ir PLA modelio elementų atitikimai, tai belieka aprašyti transformavimą.

Tiek BPMN modelį, tiek PLA modelį galima aprašyti XML formatu, tai galima taikyti tokią transformavimo kalbą, kuri vieną XML dokumentą transformuotų į kitą XML dokumentą. Tokia transformavimo kalba yra XSLT (*Extensible Stylesheet Language Transformations*), kuri vieną XML dokumentą transformuoja į kitą XML dokumentą. XSLT yra sukurta naudojimui kaip XSL (*Extensible Stylesheet Language*) kalbos dalis. 15 pav. parodytas bendras XSLT naudojimo atvejis.



15 pav. Bendras XSLT naudojimo atvejis

XSLT procesoriui yra paduodamas XML dokumentas kuris bus transformuojamas ir XSL dokumentas, kuris nurodo kaip transformuotas dokumentas atrodys. Rezultate gaunamas naujas dokumentas (dažniausiai XML, HTML ar tekstinio tipo dokumentas). XSLT transformacijos taip pat gali veikti nepriklausomai nuo XSL.

Sekantis žingsnis yra toks, pinigų išėmimo iš bankomato verslo proceso BPMN modelis yra aprašomas XML formatu.

Transformuojant BPMN modelį XML formatu į agregatinį modelį XML formatu yra sudaroma XSLT transformacijos aprašas. Žemiau yra pateikta kiekvieno BPMN modelio elemento transformavimo aprašymas į agregatinio modelio elementus.

BPMN metamodelio elementas *Pool* (Konteineris) turi tokius elementus:

- name (pavadinimas)

Žemiau esančioje lentelėje parodyta į kokius agregato metamodelio elementus šie elementai yra transformuojami.

Lentelė Nr. 8 Elemento *Pool* transformacija

BPMN metamodelio elementas	PLA metamodelio elementas	Transformavimo paskirtis
Pool	Aggregate	Elementas Pool (konteineris) transformuojamas į elementą Aggregate (Agregatas)
Pool.name	Aggregate.name	Konteinerio vardas transformuojamas į agregato vardą

Žemiau esančioje lentelėje parodyta elemento *Pool* (Konteineris) XSLT transformacijos aprašas. Kadangi nagrinėjame pavyzdyje, t.y. pinigų išėmimo iš bankomato BPMN modelyje (žr. 14 pav.), yra trys konteineriai: „Klientas“, „Bankomatas“, „Bankas“, tai jie atitinkamai yra transformuojami į tris agregatus, kurių pavadinimai: „Klientas“, „Bankomatas“, „Bankas“.

Lentelė Nr. 9 Elemento Pool XSLT transformacijos aprašas

BPMN metamodelio elemento aprašas XML formatu	XSLT transformacijos tekstinis aprašymas	PLA metamodelio elemento aprašas XML formatu	Formalizuotas aprašas
<pre><Pool_1 name="Klientas"> </Pool_1></pre>	<pre><xsl:for-each select="Pool_1"> <Aggregate> <xsl:for-each select="@name"> <xsl:attribute name="name"> <xsl:value-of select="."/> </xsl:attribute> </xsl:for-each> </Aggregate> </xsl:for-each></pre>	<pre><Aggregate name="Klientas"> </Aggregate></pre>	Agregatas Klientas
<pre><Pool_2 name="Bankomatas"> </Pool_2></pre>	<pre><xsl:for-each select="Pool_2"> <Aggregate> <xsl:for-each select="@name"> <xsl:attribute name="name"> <xsl:value-of select="."/> </xsl:attribute> </xsl:for-each> </Aggregate> </xsl:for-each></pre>	<pre><Aggregate name="Bankomatas"> </Aggregate></pre>	Agregatas Bankomatas
<pre><Pool_3 name="Bankas"> </Pool_3></pre>	<pre><xsl:for-each select="Pool_3"> <Aggregate> <xsl:for-each select="@name"> <xsl:attribute name="name"> <xsl:value-of select="."/> </xsl:attribute> </xsl:for-each> </Aggregate> </xsl:for-each></pre>	<pre><Aggregate name="Bankas"> </Aggregate></pre>	Agregatas Bankas

XML sintaksės paaiškinimas: XML standarte norima struktūra išgaunama naudojantis žymėmis (*tags*). Kiekviena žymė prasideda simboliu „<“ ir baigiasi simboliu „>“. Jų viduje užrašomas žymės vardas ir papildoma informacija, pavyzdžiui, atributai, vardų sritys ir kt. Yra pradžios ir pabaigos žymės. Pabaigos žymėje prieš jos vardą rašomas simbolis „/“. Pradžios ir pabaigos žymių apgautas turinys vadinamas XML elementu. Elemento viduje gali būti kiti elementai. XML elementas gali turėti atributus. Elementas negali turėti dviejų atributų vienodais vardais. Kai kurie atributai (pvz. `xsl:attribute`) yra rezervuoti, t. y. nurodo elemento kalbą, šiuo atveju XSL kalbą.

Taigi `<Pool_1 name="Klientas"></Pool_1>` yra XML elementas `Pool`, kurio atributas yra `name` ir po lygybės užrašytas atributo vardas tarp dvigubų kabučių, aprašantis BPMN modelio elementą konteineris, kurio vardas yra „Klientas“. XSLT apraše sakiny: `<xsl:for-each select="Pool_1">` reiškia, kad yra išrenkamas elementas `Pool`, `<Aggregate>` reiškia, kad išrinktas elementas `Pool` bus transformuotas į elementą `Aggregate`, `<xsl:for-each select="@name">` reiškia, kad yra išrenkamas elemento `Pool` atributo vardas „Klientas“, `<xsl:attribute`

name="name"> reiškia, kad išrinktas atributas bus name, kurio vardas buvo išrinktas „Klientas“. Taigi BPMN modelio elementas konteineris, kurio vardas „Klientas“ <Pool_1 name="Klientas"></Pool_1> yra transformuotas į agregato elementą agregatas <Aggregate name="Klientas"> </Aggregate>, kurio vardas „Klientas“. BPMN modelio elementas konteineris, kurio vardas „Bankomatas“ <Pool_2 name="Bankomatas"></Pool_2> yra transformuotas į agregato elementą agregatas <Aggregate name="Bankomatas"> </Aggregate>, kurio vardas „Bankomatas“. BPMN modelio elementas konteineris, kurio vardas „Bankas“ <Pool_3 name="Bankas"></Pool_3> yra transformuotas į agregato elementą agregatas <Aggregate name="Bankas"> </Aggregate>, kurio vardas „Bankas“.

Elementų <Pool_1 name="Klientas"></Pool_1>, <Pool_2 name="Bankomatas"></Pool_2>, <Pool_3 name="Bankas"></Pool_3> viduje bus rašomi kiti elementai, t.y. Pool elemento viduje tarp pradžios ir pabaigos žymių bus rašomi kiti elementai.

BPMN metamodelio elementas *Event* (Įvykis) turi tokius elementus:

- name (pavadinimas)

Žemiau esančioje lentelėje parodyta į kokius agregato metamodelio elementus šie elementai yra transformuojami.

Lentelė Nr. 10 Elemento Event transformacija

BPMN metamodelio elementas	PLA metamodelio elementas	Transformavimo paskirtis
event	external_event	Elementas event (Įvykis) yra transformuojamas į elementą external_event (Išorinį įvykį)
event.name	external_event.name	Įvykio pavadinimas transformuojamas į išorinio įvykio pavadinimą

Žemiau esančioje lentelėje parodyta elemento *Event* (Įvykis) XSLT transformacijos aprašas.

Lentelė Nr. 11 Elemento Event XSLT transformacijos aprašas

BPMN metamodelio elemento aprašas XML formatu	XSLT transformacijos tekstinis aprašas	PLA metamodelio elemento aprašas XML formatu	Formalizuotas aprašas
<event name="Autorizavimas patvirtintas"></event>	<xsl:for-each select="event"> <external_event> <xsl:for-each select="@name"> <xsl:attribute name="name"> <xsl:value-of select="."/> </xsl:attribute> </xsl:for-each> </external_event>	<external_event name="Autorizavimas patvirtintas"></external_event>	Išorinis įvykis e ₂ ' – autorizavimas patvirtintas

	</xsl:for-each>		
--	-----------------	--	--

BPMN metamodelio elementas Įvykis (*event*) transformuojamas į agregato metamodelio elementą išorinį įvykį (*external event*).

XML aprašyme `<event name="Autorizavimas patvirtintas"> </event>` yra XML elementas `event`, kurio atributas yra `name` ir po lygybės užrašytas atributo vardas tarp dvigubų kabučių „Autorizavimas patvirtintas“, aprašantis BPMN modelio elementą įvykis, kurio vardas yra „Autorizavimas patvirtintas“. Šis BPMN modelio elementas įvykis yra transformuojamas į agregato modelio elementą išorinis įvykis e_2' – autorizavimas patvirtintas.

Paimkime dar vieną įvykio transformavimo pavyzdį. Transformavimo aprašymas pateiktas lentelėje Nr. 12.

Lentelė Nr. 12 Elemento Event XSLT transformacijos aprašas

BPMN metamodelio elemento aprašas XML formatu	XSLT transformacijos tekstinis aprašas	PLA metamodelio elemento aprašas XML formatu	Formalizuotas aprašas
<pre><event_4 name="Atejo signalas, kad klientas paimtu kortele"/></pre>	<pre><xsl:for-each select="event_4"> <external_event> <xsl:for-each select="@name"> <xsl:attribute name="name"> <xsl:value-of select="."/> </xsl:attribute> </xsl:for-each> </external_event> </xsl:for-each></pre>	<pre><external_event name="Atejo signalas, kad klientas paimtu kortele"> </external_event></pre>	Išorinis įvykis e_4' – atėjo signalas, kad klientas paimtų kortelę

BPMN metamodelio elementas Įvykis (*event*) transformuojamas į agregato metamodelio elementą išorinį įvykį (*external event*).

XML aprašyme `<event_4 name="Atejo signalas, kad klientas paimtu kortele"/>` yra XML elementas `event`, kurio atributas yra `name` ir po lygybės užrašytas atributo vardas tarp dvigubų kabučių „Atėjo signalas, kad klientas paimtų kortelę“, aprašantis BPMN modelio elementą įvykis, kurio vardas yra „Atėjo signalas, kad klientas paimtų kortelę“. Šis BPMN modelio elementas įvykis yra transformuojamas į agregato modelio elementą išorinis įvykis e_4' – atėjo signalas, kad klientas paimtų kortelę, šis išorinis įvykis yra agregate „Klientas“, agregato vardo nurodyti nereikia, nes visi elementai, kurie yra susiję su agregatu, yra rašomi XML elemento `<Aggregate>` `</Aggregate>` viduje tarp pradžios ir pabaigos žymių. Ir pagal tokias pačias taisykles, BPMN modelio visi įvykių elementai yra transformuojami į agregato modelio išorinius įvykius.

BPMN metamodelio elementas *Task* (Užduotis) turi šiuos elementus:

- name (pavadinimas)

Žemiau esančioje lentelėje parodyta į kokius agregato metamodelio elementus šie elementai yra transformuojami.

Lentelė Nr. 13 Elemento Task transformacija

BPMN metamodelio elementas	PLA metamodelio elementas	Transformavimo paskirtis
task	internal_event	Elementas task (užduotis) yra transformuojamas į elementą internal_event (vidinis įvykis)
task.name	internal_event.name	Užduoties vardas yra transformuojamas į vidinio įvykio vardą

Žemiau esančioje lentelėje parodyta elemento *Task* (Užduotis) XSLT transformacijos aprašas.

Lentelė Nr. 14 Elemento Task XSLT transformacijos aprašas

BPMN metamodelio elemento aprašas XML formatu	XSLT transformacijos tekstinis aprašas	PLA metamodelio elemento aprašas XML formatu	Formalizuotas aprašas
<pre><task name="Ivesti PIN koda"> </task></pre>	<pre><xsl:for-each select="task"> <internal_event> <xsl:for-each select="@name"> <xsl:attribute name="name"> <xsl:value-of select="."/ > </xsl:attribute> </xsl:for-each> </internal_event> </xsl:for-each></pre>	<pre><internal_event name="Ivesti PIN koda"> </internal_event></pre>	Vidinis įvykis e ₂ – įvesti PIN kodą

XML aprašyme `<task name="Ivesti PIN koda"> </task>` yra XML elementas *task*, kurio atributas yra *name* ir po lygybės užrašytas atributo vardas „Ivesti PIN koda“ tarp dvigubų kabučių. Šis XML elementas aprašo BPMN modelio elementą užduotis (*Task*), kurio vardas yra „Ivesti PIN koda“. Šis BPMN modelio elementas užduotis yra transformuojamas į agregato modelio elementą vidinis įvykis e₂ – įvesti PIN kodą.

Sekančioje lentelėje pateikta užduoties ir pranešimų srauto transformavimo aprašymas. BPMN modelyje pranešimų srauto elementu (*Message Flow*) yra nurodomas srautas tarp dviejų esybių.

Lentelė Nr. 15 Elementų Task ir Message Flow transformavimo aprašas

BPMN metamodelio elemento aprašas XML formatu	XSLT transformacijos tekstinis aprašas	PLA metamodelio elemento aprašas XML formatu	Formalizuotas aprašas
<pre><task_2 name="Ivesti PIN</pre>	<pre><xsl:for-each select="task_2"></pre>	<pre><internal_eve nt</pre>	Vidinis įvykis e ₂ – įvesti PIN kodą.

<pre>koda"> <message_flow_y1 name="Autorizuoti PIN koda" participant_to="Ban kas"/> </task_2></pre>	<pre><internal_event> <xsl:for-each select="@name"> <xsl:attribute name="name"> <xsl:value-of select="."/ /> </xsl:attribute> </xsl:for-each> <xsl:for-each select="message_flow_y1"> <output> <xsl:for-each select="@name"> <xsl:attribute name="reiksme"> <xsl:value-of select="."/ /> </xsl:attribute> </xsl:for-each> <xsl:attribute name="name"> <xsl:text>y1</xsl:text> </xsl:attribute> <xsl:for-each select="@participant_to"> <xsl:attribute name="iagregata"> <xsl:value-of select="."/ /> </xsl:attribute> </xsl:for-each> </output> </xsl:for-each> </internal_event> </xsl:for-each></pre>	<pre>name="Ivesti PIN koda"> <output reiksme="Auto rizuoti PIN koda" name="y1" iagregata="Ba nkas"/> </internal_ev ent></pre>	<p>Išėjimo signalas y₁ – autorizuoti PIN kodą</p>
--	---	--	--

XML apraše `<task_2 name="Ivesti PIN koda"> </task_2>` yra XML elementas `task`, kurio atributas `name` ir po lygybės tarp dvigubų kabučių užrašytas atributo pavadinimas „Ivesti PIN koda“, yra XML aprašymas BPMN modelio elemento užduotis, kurio vardas yra „Ivesti PIN koda“. Šis BPMN modelio elementas yra transformuojamas į agregato elementą vidinis įvykis e_2 – įvesti PIN kodą. O XML apraše `<message_flow_y1 name="Autorizuoti PIN koda" participant_to="Bankas"/>` yra XML elementas `message_flow_y1`, kurio atributai `name` ir `participant_to`, kurie reiškia `name` – elemento vardą ir atributą – `participant_to`, aprašo BPMN modelio elementą pranešimo srautą (*Message Flow*), kurio vardas yra „Autorizuoti PIN kodą“ ir kad šis pranešimų srautas keliauja į konteinerį „Bankas“. BPMN modelio elementas, kurio XML aprašymas yra `<message_flow_y1 name="Autorizuoti PIN koda" participant_to="Bankas"/>` yra transformuojamas į agregato modelio elementą išėjimo signalą y_1 – autorizuoti PIN kodą.

BPMN modelio elementas *Message Flow* (Pranešimo srautas) turi šiuos elementus:

- `name` (pavadinimas)
- `participant_to`

- participant_is

Žemiau esančioje lentelėje parodyta į kokius agregato metamodelio elementus šie elementai yra transformuojami.

Lentelė Nr. 16 Elemento Message Flow transformacija

BPMN metamodelio elementas	PLA metamodelio elementas	Transformavimo paskirtis
message_flow	input	Elementas message_flow (pranešimų srautas) yra transformuojamas į elementą input (įėjimas)
message_flow	output	Elementas message_flow (pranešimų srautas) yra transformuojamas į elementą output (išėjimas)
message_flow.name	input.name	Pranešimų srauto pavadinimas transformuojamas į įėjimo pavadinimą
message_flow.name	output.name	Pranešimų srauto pavadinimas transformuojamas į išėjimo pavadinimą
message_flow.participant_to	input.iagregata	Konteinerio pavadinimas į kurį pranešimų srautas įeina transformuojamas į agregato pavadinimą, į kurį signalas įeina
message_flow.participant_is	output.isagregato	Konteinerio pavadinimas iš kurio pranešimų srautas išeina transformuojamas į agregato pavadinimą, iš kurio signalas išeina

Žemiau esančioje lentelėje parodyta elemento *Message Flow* (Pranešimo srautas) XSLT transformacijos aprašas. Pranešimo srauto elementas BPMN modelyje transformuojamas į įėjimo signalą agregato modelyje.

Lentelė Nr. 17 Elemento Message Flow XSLT transformacijos aprašas

BPMN metamodelio elemento aprašas XML formatu	XSLT transformacijos tekstinis aprašas	PLA metamodelio elemento aprašas XML formatu	Formalizuotas aprašas
<pre><message_flow name="Paimti kortele" participant_is ="Bankomatas"/ ></pre>	<pre><xsl:for-each select="message_flow"> <input> <xsl:for-each select="@name"> <xsl:attribute name="reiksme"> <xsl:value-of select="."/ > </xsl:attribute> </xsl:for-each> <xsl:attribute name="name"> <xsl:text>x4</xsl:text> </xsl:attribute> <xsl:for-each select="@participant_is"> <xsl:attribute name="isagregato"></pre>	<pre><input reiksme="Paimti kortele" name="x4" isagregato="Bankom atas"/></pre>	Įėjimo signalas x4 – paimti kortele

	<pre><xsl:value-of select="."/> </xsl:attribute> </xsl:for-each> </input> </xsl:for-each></pre>		
--	---	--	--

XML aprašymas `<message_flow name="Paimti kortele" participant_is="Bankomatas"/>`, reiškia XML elementą `message_flow`, kuris turi atributus `name` ir `participant_is`, kurių pavadinimai atitinkamai yra „Paimti kortelę“ ir „Bankomatas“, šis XML aprašymas aprašo BPMN modelio elementą pranešimų srautą (*Message Flow*), kurio vardas yra „Paimti kortelę“ ir šis pranešimų srauto elementas ateina iš konteinerio „Bankomatas“. BPMN modelio elementas pranešimų srautas „Paimti kortelę“ yra transformuojamas į įėjimo signalą x_4 – paimti kortelę.

Pranešimo srauto elementas (*Message Flow*), kuris aprašo išėjimo srautą iš elemento BPMN modelyje gali būti transformuojamas į išėjimo signalą agregato modelyje (žr. lentelę Nr. 18).

Lentelė Nr. 18 Elemento Message Flow XSLT transformacijos aprašas

BPMN metamodelio elemento aprašas XML formatu	XSLT transformacijos tekstinis aprašas	PLA metamodelio elemento aprašas XML formatu	Formalizuotas aprašas
<pre><message_flow name="Autorizuoti PIN koda " participant_to="Bankas"/></pre>	<pre><xsl:for-each select="message_flow"> <output> <xsl:for-each select="@name"> <xsl:attribute name="reiksme"> <xsl:value-of select="."/> </xsl:attribute> </xsl:for-each> <xsl:attribute name="name"> <xsl:text>y1</xsl:text> </xsl:attribute> <xsl:for-each select="@participant_to"> <xsl:attribute name="iagregata"> <xsl:value-of select="."/> </xsl:attribute> </xsl:for-each> </output> </xsl:for-each></pre>	<pre><output reiksme=" Autorizuoti PIN koda " name="y1" iagregata="Ban kas"/></pre>	Išėjimo signalas y_1 – Autorizuoti PIN kodą

XML aprašymas `<message_flow name="Autorizuoti PIN koda" participant_to="Bankas"/>`, reiškia XML elementą `message_flow`, kuris turi atributus `name` ir `participant_to`, kurių pavadinimai atitinkamai yra „Autorizuoti PIN koda“ ir „Bankas“, šis XML aprašymas aprašo BPMN modelio elementą pranešimų srautą (*Message Flow*) „Paimti kortelę“ ir šis pranešimų srauto elementas nueina į konteinerį, kurio vardas yra „Bankas“. BPMN modelio elementas

pranešimų srautas „Autorizuoti PIN koda“ yra transformuojamas į išėjimo signalą y_1 – autorizuoti PIN kodą.

Kadangi BPMN metamodelyje nėra išskirta diskrečioji būseną, kuri galėtų aprašyti, tarkim, konteinerio diskrečiąją būsenos dedamąją, ir jos kintamųjų reikšmė nėra pateikta jokių komentarų, o agregato metamodelyje yra išskirta diskrečioji agregato būsenos dedamoji, tai reikėtų papildyti tekstinį XSLT transformacijos aprašymą. XSLT transformacijos aprašymą papildome sakiniiais, kurie aprašytų agregato diskrečiąją būsenos dedamąją (žr. lentelę Nr. 19)

Lentelė Nr. 19 Diskrečiosios būsenos dedamosios XSLT aprašymas

XSLT transformacijos aprašas	PLA metamodelio elemento aprašas	Formalizuotas aprašymas
<pre><DiscreteVariable> <xsl:attribute name="v(tm)"> <xsl:text>0</xsl:text> </xsl:attribute> </DiscreteVariable></pre>	<pre><DiscreteVariable vtm="0"/></pre>	Diskrečioji būsenos dedamoji $v(t_m) = \emptyset$

XSLT aprašyme sakiny `<DiscreteVariable> </DiscreteVariable>`, reiškia, kad transformuotame apraše bus XML elementas `<DiscreteVariable> </DiscreteVariable>` ir turės atributą `vtm` ir to atributo pavadinimas „0“. Atlikus transformaciją bus gautas toks XML elementas `<DiscreteVariable vtm="0"/>`, kuris aprašo agregato diskrečiąją būsenos dedamąją ir diskrečiosios būsenos aibė yra tuščia $v(t_m) = \emptyset$, kadangi BPMN modelyje nėra išskirta diskrečioji būsenos dedamoji. Ir diskrečioji agregato būseną yra nurodoma taip pat visuose agregatuose.

Kadangi agregato metamodelyje yra apibrėžta tolydžioji agregato būsenos dedamoji, tai ir BPMN metamodelyje reikia surasti kas gi galėtų atitikti diskrečiąją būsenos dedamąją. Taigi, tolydžioji agregato būsenos dedamoji yra sudaroma pagal BPMN modelio užduočių skaičių, kiekvienai BPMN modelio užduočiai yra apibrėžiama tolydinė komponentė. Nes agregato tolydžioji būsenos dedamoji sudaroma pagal vidinių įvykių skaičių, kiekvienam vidiniam įvykiui apibrėžiant tolydinę komponentę. Kadangi BPMN modelyje užduotis yra transformuojama į agregato modelio vidinį įvykį, tai BPMN modelyje kiekvienai užduočiai yra apibrėžiama tolydinė komponentė. Tolydžiosios būsenos transformavimas pateiktas lentelėje Nr. 20.

Lentelė Nr. 20 Tolydžiosios agregato būsenos dedamosios transformavimo aprašymas

BPMN metamodelio elemento aprašas XML formatu	XSLT transformacijos tekstinis aprašas	PLA metamodelio elemento aprašas XML formatu	Formalizuotas aprašas
<pre><TolydziosiosBusenos> <TolydziojiBusena_1 name="w(e1)" reiksme="Saskaitos likucio parodymo laiko momentas"/> <TolydziojiBusena_2</pre>	<pre><xsl:for-each select="TolydziosiosBusenos"> <ContinuousVariables> <xsl:for-each select="TolydziojiBusena_1"> <ContinuousVariable></pre>	<pre><ContinuousVariables> <ContinuousVariable name="w(e1)" reiksme="Saskaitos likucio parodymo laiko momentas"/> <ContinuousVariable</pre>	$w(e^1, t_m)$ – saskaitos likučio parodymo laiko momentas; $w(e^2, t_m)$ – kortelės

<pre>name="w(e2)" reiksme="Korteles atidavimo laiko momentas"/> </TolydziosiosBusenos></pre>	<pre><xsl:for-each select="@name"> <xsl:attribute name="name"> <xsl:value-of select="."/> </xsl:attribute> </xsl:for-each> <xsl:for-each select="@reiksme"> <xsl:attribute name="reiksme"> <xsl:value-of select="."/> </xsl:attribute> </xsl:for-each> </ContinuousVariable> </xsl:for-each> <xsl:for-each select="TolydziojiBusena_ 2"> <ContinuousVariable> <xsl:for-each select="@name"> <xsl:attribute name="name"> <xsl:value-of select="."/> </xsl:attribute> </xsl:for-each> <xsl:for-each select="@reiksme"> <xsl:attribute name="reiksme"> <xsl:value-of select="."/> </xsl:attribute> </xsl:for-each> </ContinuousVariable> </xsl:for-each> </ContinuousVariables> </xsl:for-each></pre>	<pre>e name="w(e2)" reiksme="korteles atidavimo laiko momentas"/> </ContinuousVariables></pre>	<pre>atidavimo laiko momentas</pre>
--	--	---	-------------------------------------

XML aprašyme `<TolydziojiBusena_1 name="w(e1)" reiksme="Saskaitos likucio parodymo laiko momentas"/>` aprašo XML elementą `TolydziojiBusena_1`, kuris turi atributus `name „w(e1)“` ir `reiksme „Saskaitos likucio parodymo laiko momentas“`. Šis XML elementas aprašo BPMN modelio konteinerio tolydžiąją būseną. Taip pat šis elementas yra transformuojamas į agregato modelio elementą, tolydžioji agregato būsenos dedamoji, kurio XML aprašas yra `<ContinuousVariable name="w(e1)" reiksme="Saskaitos likucio parodymo laiko momentas"/>`, o formalizuotas aprašas yra $w(e_1, t_m)$ – sąskaitos likučio parodymo laiko momentas. Kitas XML aprašyme elementas `<TolydziojiBusena_2 name="w(e2)" reiksme="Korteles atidavimo laiko momentas"/>`, aprašo BPMN modelio tolydžiąją konteinerio būseną ir yra transformuojamas į PLA modelio elementą `<ContinuousVariable name="w(e2)" reiksme="korteles atidavimo laiko momentas"/>`, kuris aprašo agregato tolydžiąją būsenos

dedamąją, formalizuotas aprašas - $w(e''_2, t_m)$ – kortelės atidavimo laiko momentas. Lentelėje Nr. 20 yra parodyta konteinerio „Bankomatas“ tolydžiųjų būsenų transformavimas į agregato tolydžiųjų būsenų dedamąsias, kadangi tolydžiosios agregato būsenos nustatomos pagal vidinių įvykių skaičių, kiekvienam vidiniam įvykiui apibrėžiant tolydines komponentes, todėl yra dvi tolydinės komponentės, nes agregate „Bankomatas“ yra du vidiniai įvykiai. Pagal tokias pat transformavimo taisykles yra transformuojamos ir kitų agregatų „Klientas ir „Bankas“ tolydžiosios būsenų dedamosios, kiekvienam vidiniam įvykiui apibrėžiant tolydinę komponentę.

Pagal tokias pat taisykles yra transformuojama ir valdanti seka. PLA modelyje kiekvienam vidinių įvykių aibėje esančiam įvykiui yra apibrėžiama valdanti seka. Lentelėje Nr. 21 yra parodyta konteinerio „Bankomato“ valdančios sekos transformavimas į agregato „Bankomatas“ valdančią seką.

Lentelė Nr. 21 Valdančios sekos transformavimo aprašymas

BPMN metamodelio elemento aprašas XML formatu	XSLT transformacijos tekstinis aprašas	PLA metamodelio elemento aprašas XML formatu	Formalizuotas aprašas
<pre><ValdantiSeka> <ValdantiSeka_1 name="ksi_1" reiksme="Laiko tarpas iki saskaitos likucio parodymo"/> <ValdantiSeka_2 name="ksi" reiksme="Laiko tarpas iki korteles atidavimo"/> </ValdantiSeka></pre>	<pre><xsl:for-each select="ValdantiSeka"> <ValdantiSeka> <xsl:for-each select="ValdantiSeka_1"> <ValdanciosSekosDedamoji> <xsl:for-each select="@name"> <xsl:attribute name="name"> <xsl:value-of select="."/> </xsl:attribute> </xsl:for-each> <xsl:for-each select="@reiksme"> <xsl:attribute name="reiksme"> <xsl:value-of select="."/> </xsl:attribute> </xsl:for-each> </Valdančios SekosDedamoji> </xsl:for-each> <xsl:for-each select="ValdantiSeka_2"> < Valdančios SekosDedamoji > <xsl:for-each select="@name"> <xsl:attribute name="name"> <xsl:value-of select="."/> </xsl:attribute> </xsl:for-each> <xsl:for-each select="@reiksme"></pre>	<pre><ValdantiSeka> <ValdanciosSekosDed amoji name="ksi_1" reiksme="Laiko tarpas iki saskaitos likucio parodymo"/> <ValdanciosSekosDed amoji name="ksi_2" reiksme="Laiko tarpas iki korteles atidavimo"/> </ValdantiSeka></pre>	$e_1'' \rightarrow \{\xi_{1i}\}, i = 0, \infty$ $e_2'' \rightarrow \{\xi_{2i}\}, i = 0, \infty$ ξ_1 - laiko tarpas iki sąskaito likučio parodymo; ξ_2 - laiko tarpas iki kortelės atidavimo.

	<pre> <xsl:attribute name="reiksme"> <xsl:value-of select="."/> </xsl:attribute> </xsl:for-each> </Valdančios SekosDedamoji > </xsl:for-each> </ValdantiSeka> </xsl:for-each> </pre>		
--	--	--	--

XML elemento aprašymas `<ValdantiSeka_1 name="ksi_1" reiksme="Laiko tarpas iki saskaitos likucio parodymo"/>`, reiškia konteinerio „Bankomatas“ valdančios sekos dedamąją, kuri apibrėžiama „Laiko tarpas iki saskaitos likucio parodymo“, šis elementas yra transformuojamas į agregatinio modelio XML elementą `<ValdanciosSekosDedamoji name="ksi_1" reiksme="Laiko tarpas iki saskaitos likucio parodymo"/>`, kuris reiškia agregato valdančios sekos dedamąją ξ_1 - laiko tarpas iki sąskaitos likučio parodymo. Kitas XML elementas `ValdantiSeka_2 name="ksi" reiksme="Laiko tarpas iki korteleles atidavimo"/>`, kuris aprašo konteinerio valdančios sekos dedamąją, kuris yra transformuojamas į agregato valdančios sekos dedamąją, kurios XML aprašas yra `ValdanciosSekosDedamoji name="ksi_2" reiksme="Laiko tarpas iki korteleles atidavimo"/>`, kurio formalizuotas aprašas yra $e_2 \rightarrow \{\xi_{2i}\}, i = 0, \infty$, ξ_2 - laiko tarpas iki kortelės atidavimo. Pagal tokias pačias taisykles yra transformuojamos ir kitų konteinerių valdančios sekos į agregato valdančios sekos dedamąsias.

BPMN modelyje kiekvienai užduočiai yra apibrėžiami išėjimo operatoriai, nes agregato modelyje kiekvienam vidiniam įvykiui yra apibrėžiamas išėjimo operatorius, kuris keičia išėjimo signalus.

Lentelė Nr. 22 Išėjimo operatorių transformavimo aprašymas

BPMN metamodelio elemento aprašas XML formatu	XSLT transformacijos tekstinis aprašas	PLA metamodelio elemento aprašas XML formatu	Formalizuotas aprašas
<pre> <Isejimai> <Gevent_1 name="0"/> <Gevent_2 name="y1"/> </Isejimai> </pre>	<pre> <xsl:for-each select="Isejimai"> <IsejimoOperatoriai> <xsl:for-each select="Gevent_1"> <IsejimoOperatorius> <xsl:for-each select="@name"> <xsl:attribute name="name"> <xsl:value-of select="."/> </xsl:attribute> </xsl:for-each> <xsl:for-each </IsejimoOperatorius> </xsl:for-each> <xsl:for-each </pre>	<pre> <IsejimoOperatoriai> <IsejimoOperatorius name="0"/> <IsejimoOperatorius name="y1"/> </Isejimai> </pre>	<pre> G(e₁''): y = 0 G(e₂''): y = y₁ </pre>

	<pre> select="Gevent_2"> < IsejimoOperatorius> <xsl:for-each select="@name"> <xsl:attribute name="name"> <xsl:value-of select="."/ /> </xsl:attribute> </xsl:for-each> </IsejimoOperatorius> </xsl:for-each> </IsejimoOperatoriai> </xsl:for-each> </pre>		
--	--	--	--

XML elementas `<Gevent_1 name="0"/>` aprašo BPMN modelio išėjimą, „0“ reikšmė reiškia, kad vidinis įvykis nekeičia išėjimo operatoriaus. XML elementas `<Gevent_2 name="y1"/>` aprašo BPMN modelio išėjimo operatorių, kur vidinis įvykis keičia išėjimo operatorių išduodamas signalą y_2 . Šie elementai yra transformuojami į agregato išėjimo operatorius, kurie XML aprašomi taip: `<IsejimoOperatorius name="0"/>` `<IsejimoOperatorius name="y1"/>`, kurie aprašo agregato išėjimo operatorius, kurių formalizuotas aprašas yra: $G(e1'')$: $y = 0$; $G(e2'')$: $y = y_1$. Pagal tokias pačias taisykles yra transformuojami ir kiti BPMN modelio išėjimo operatoriai į agregato išėjimo operatorius.

Pagal tokias pačias taisykles yra transformuojami ir BPMN modelio perėjimo operatoriai, kurie yra transformuojami į agregato modelio perėjimo operatorius. Kiekvienam vidiniam ir išoriniam įvykiui yra nurodomi perėjimo operatoriai.

Pagal tokias pačias taisykles yra transformuojami ir kiti verslo proceso pinigų išėmimo iš bankomato BPMN modelio elementai, t.y. įėjimo signalai, išėjimo signalai, išoriniai įvykiai, vidiniai įvykiai, diskrečiosios agregato būsenos dedamosios, valdymo sekos dedamosios, tolydžiosios agregato būsenos dedamosios, pradinė būsenos dedamoji, perėjimo ir išėjimo operatoriai.

Atlikus XSLT transformaciją yra gaunamas PLA modelio aprašas XML formatu. Iš PLA modelio aprašymo XML formatu yra sudaromas agregatinės specifikacijos aprašas formalizuotu aprašu (verslo proceso pinigų išėmimo iš bankomato agregatinės specifikacijos aprašas pateiktas prieduose).

4.3.3.1. Transformavimo apibendrinimas

Transformuojant BPMN modelius į agregatinę specifikaciją, norint gauti išpildomą specifikaciją reikia papildomos informacijos apie kintamųjų reikšmes ir jų kitimą, nes BPMN modelyje nėra galimybės tokios informacijos pateikimui.

Apibendrinant verslo proceso, aprašyto BPMN modeliu, pateiktą transformavimo metodiką, galima daryti tokias išvadas, kurios pateiktos žemiau. Pateikta transformavimo metodika, leidžia verslo

procesus aprašytus, BPMN notacija, transformuoti į PLA modelį. Transformavimo metodiką galima suskirstyti į tokius žingsnius:

- Aprašyti, kokie BPMN metamodelio elementai transformuojami į kokius agregato metamodelio elementus.
- Sudaryti BPMN modelį.
- Nustatyti atitikimus tarp BPMN modelio elementų ir PLA modelio elementų, kuriais naudojantis, vėliau bus rašomos transformavimo taisyklės.
- Sudaryti BPMN modelio aprašą XML formatu.
- Sudaryti XSLT transformacijos aprašymą.
- Gautą PLA modelio aprašymą XML formatu, formalizuoti.

Apibendrinat verslo proceso, aprašyto BPMN modeliu, transformavimą, konteineris yra transformuojamas į agregatą. Konteinerio vardas yra transformuojamas į agregato pavadinimą. Įeinantis pranešimų srautas (*Message Flow*) į konteinerį BPMN modelyje yra transformuojamas į įėjimo signalą į agregatą agregatiniame modelyje. Išeinantis pranešimų srautas (*Message Flow*) iš konteinerio yra transformuojamas į išėjimo signalą iš agregato agregatiniame modelyje. Pradžios/Tarpinis įvykis (Start/Intermediate Event) transformuojamas į išorinį arba vidinį įvykį. Išorinis įvykis yra tas į kurį iš kito konteinerio ateina pranešimų srautas, o vidinis – tas iš kurio į kitą konteinerį išeina pranešimo srautas. Vidinių įvykių skaičius nusako valdančią seką. Kiekvienam vidinių įvykių aibėje esančiam įvykiui yra apibrėžiama valdanti seka. Užduotis BPMN modelyje yra transformuojama į vidinį įvykį agregatiniame modelyje. BPMN modelyje neišskirta diskrečioji būseną. Jos kintamųjų prasmė nepateikta jokių komentarų, tai atliekant transformavimą yra nurodoma, kad diskrečioji agregato būsenos dedamoji yra tuščia aibė. Tolydžioji agregato būsenos dedamoji yra sudaroma pagal vidinių įvykių skaičių. Kiekvienam vidiniam įvykiui yra apibrėžiama tolydinė komponentė. Vidiniai įvykiai taip pat gali keisti signalus, šiuos pokyčius aprašo išėjimo (G) operatorius. Pagal tokias pačias taisykles yra transformuojami ir BPMN modelio perėjimo operatoriai į agregato perėjimo operatorius, kiekvienam vidiniam ir išoriniam įvykiui yra apibrėžiami perėjimo operatoriai (H).

Šio darbo naujumas yra toks, kad verslo procesų, aprašytų BPMN notacija, transformavimo metodika į formalizuotą aprašymą nėra standartizuota, nes BPMN standartas yra pakankamai naujas.

5. IŠVADOS

1. Išanalizavus verslo procesų modeliavimo sąvokas, koncepciją, notacijas (UML, BPMN), kurios dažniausiai naudojamos, norint aprašyti verslo procesus, nustatyta, kad BPMN notacija yra tinkamesnė nei UML, dėl BPMN notacijos privalumų, lyginant su UML:
 - Pirma, ji leidžia aprašyti procesus srautais, tai lengvai suprantama verslo analitikui, ir turi išsamią tipinių modelių elementų aibę.
 - Antra, lengvai suprantama modeliuojant ir sudėtingus verslo procesus.
2. Sudaryta verslo procesų, aprašytų BPMN modeliu, į PLA modelius transformavimo metodika, kuri leidžia verslo procesus, aprašytus BPMN notacija, transformuoti į PLA modelius.
3. Metodika išbandyta, sudarant verslo proceso pinigų išėmimo iš bankomato BPMN modelį, kuris buvo transformuotas į PLA modelį.
4. Transformuojant BPMN modelius į agregatinę specifikaciją, norint gauti išpildomą specifikaciją reikia papildomos informacijos apie kintamųjų reikšmes ir jų kitimą, tačiau šią informaciją BPMN modelyje galima pateikti komentarų pavidalu.

7. SUMMARY

Transformation business process, described by BPMN, to PLA models

The notations of business processes can be used to design IT systems. If we want to reduce the expenses and the amount of mistakes of the bigger projects we need to create additional models, which can help us to find out whether the business processes are well understandable and correct. The question is how to reduce the expenses for creation of business process models. The purpose of this thesis is to answer to this question: how to transform BPMN models to PLA method.

The basic concepts were gathered and systemized, that will allow us to talk about business processes in a structured way. Also was made analysis of business process modeling languages (BPMN (Business Process Modeling Notation) and UML (Unified Modeling Language)). Also was described piece-linear aggregate (PLA) method, which is used for creation of dynamical models of business process.

Was made BPMN model of ATM (automatic teller machine) business process, which was transformed to PLA model. The results of transformation were systemized and were made general requirements how to transform BPMN models to piece-linear aggregate.

6. LITERATŪRA

- [1] *Business Process Modeling Notation (BPMN)*. Version 1.0. May 3, 2004. Prieiga per internetą <http://www.bpmn.org/Documents/BPMN%20V1-0%20May%203%202004.pdf>
- [2] Owen M., Raj J. *BPMN and Business Process Management. Introduction to the New Business Process Modeling Standard*.
- [3] *Business Process Modelling and Standardisation*. Prieiga per internetą http://www.cimosade.com/Standards/BPM_and_Standardisation.pdf.
- [4] *Business Process Modeling – Concepts*. Prieiga per internetą <http://www.ibissoft.se/events/pbpm/Concepts.htm>
- [5] Rosemarie G. *Development Meets Business Process Modeling*. Prieiga per internetą http://www.developer.com/design/article.php/10925_3492721_2.
- [6] Davenport T. *Process innovation: re-engineering work through informatikon technology*. Harvard Business School Press, Boston, MA, JAV.
- [7] Violino B. *Getting better all the time companies turn to business-performance management tools to stay on track to meet their goals*. Optimizme Magazine, No 22.
- [8] Nemuraitė L., Simanaitytė K. *Veiklos procesų modeliavimas naudojant verslo procesų modeliavimo notaciją*. Informacinė visuomenė ir universitetinės studijos '2004: konferencijos pranešimų medžiaga. Kaunas, 2004 balandžio 15 d.
- [9] Simanaitytė K. *Elektroninio verslo procesų modeliavimo metodų tobulinimas*. Magistrinis darbas. Prieiga per internetą <http://submit.library.lt/ETD-afiles?KTU/etd-LABT20040529-170817-70192/unrestricted/magistrinis.pdf>
- [10] Heumann J. *Introduction to Business Modeling Using the Unified Modeling Language (UML)*. Rational Software. Prieiga per internetą <http://www-128.ibm.com/developerworks/rational/library/content/RationalEdge/mar01/IntroductiontoBusinessModelingUsingUMLMarch01.pdf>
- [11] *UML 2.0 Superstructure Specification*. OMG Adopted Specification, 2003. Prieiga per internetą <http://www.omg.org/technology/documents/formal/uml.htm>
- [12] Hagel J., Seely Brown J. *Cut Loose From Old Business Process*. Optimizme Magazine, No 22.
- [13] Lamsweerde, V., *A Formal specification: a Roadmap*. The Future of Software Engineering – 2000: tarptautinės konferencijos pranešimo medžiaga [Ireland, 2000] p. ACM Press, 147-159, USA, 2000.
- [14] **Pranevicius H.** Aggregate approach for specification, validation, simulation and implementation of computer network protocols. Lecture notes in Computer Science No502, Springer–Verlag, pp.433–477. 1991.

- [15] **Pranevicius, H.** Formalization and simulation of business process. Modelling and simulation of business systems. International conference. Vilnius, Lithuania. 2003.
- [16] Kalnins A., Vitolins V. *Use of UML and Transformation for Workflow Process Definitions.* University of Latvia.
- [17] Kurtev I., Adaptability of Model Transformations. Faculty of Electrical Engineering, Mathematics & Computer Science.
- [18] Dufresne T., Martin J. *Process Modeling for E-Business.*

8. PRIEDAI

8.1. Verslo proceso pinigų išėmimo iš bankomato agregatinė specifikacija

Agregato Klientas aprašymas

1. Įėjimo signalų aibė:

$$X = \{x_1, x_2, x_3, x_4\}$$

x_1 – kliento aptarnavimo pradžia;

x_2 – teisingas PIN kodas;

x_3 – pinigų sąskaitos patikrinimas (pinigų sąskaitoje užtenka);

x_4 – paimti kortelę.

2. Išėjimo signalų aibė:

$$Y = \{y_1, y_2, y_3, y_4\}$$

y_1 – autorizuoti PIN kodą;

y_2 – patikrinti sąskaitos likutį;

y_3 – nurašyti pinigus nuo sąskaitos;

y_4 – klientas paėmė pinigus.

3. Išorinių įvykių aibė:

$$E' = \{e_1', e_2', e_3', e_4'\}$$

e_1' – atėjo signalas x_1 (pradžia);

e_2' – atėjo signalas x_2 (autorizavimas patvirtintas);

e_3' – atėjo signalas x_3 (pinigų sąskaitoje yra);

e_4' – atėjo signalas x_4 (atėjo signalas, kad klientas paimtų kortelę).

4. Vidinių įvykių aibė:

$$E'' = \{e_1'', e_2'', e_3'', e_4'', e_5''\}$$

e_1'' – įdėti kortelę;

e_2'' – įvesti PIN kodą;

e_3'' – įvesti pinigų sumą;

e_4'' – paimti pinigus iš bankomato

e_5'' – paimti kortelę.

5. Valdymo sekos:

$$e_1'' \rightarrow \{\xi_{1i}\}, i = 0, \infty$$

$$e_2'' \rightarrow \{\xi_{2i}\}, i = 0, \infty$$

$$e_3'' \rightarrow \{\xi_{3i}\}, i = 0, \infty$$

$$e_4'' \rightarrow \{\xi_{4i}\}, i = 0, \infty$$

$$e_5'' \rightarrow \{\xi_{5i}\}, i = 0, \infty$$

ξ_1 - laiko tarpas iki kortelės įdėjimo;

ξ_2 - laiko tarpas iki PIN kodo įvedimo;

ξ_3 - laiko tarpas iki pinigų sumos įvedimo;

ξ_4 - laiko tarpas iki pinigų paėmimo iš bankomato;

ξ_5 - laiko tarpas iki kortelės paėmimo.

6. Diskrečioji būsenos dedamoji:

$$v(t_m) = \emptyset$$

BPMN modelyje nėra išskirta diskrečioji būsena. Jos kintamųjų prasmė nepateikta jokių komentarų.

7. Tolydžioji būsenos dedamoji:

$$Z_v(t_m) = \{w(e_1'', t_m), w(e_2'', t_m), w(e_3'', t_m), w(e_4'', t_m), w(e_5'', t_m)\}$$

$w(e''_1, t_m)$ – įdėjimo kortelės laiko momentas;
 $w(e''_2, t_m)$ – PIN kodo įvedimo laiko momentas;
 $w(e''_3, t_m)$ – pinigų įvedimo laiko momentas;
 $w(e''_4, t_m)$ – paėmimo pinigų laiko momentas;
 $w(e''_5, t_m)$ – kortelės paėmimo laiko momentas.

8. Pradinė būseną:

$k(t_0) = 0, Z_v(t_m) = \{\infty, \infty, \dots, \infty\}$

9. Perėjimo ir išėjimo operatoriai:

$H(e_1')$:

$w(e_1'', t_{m+1}) = t_m + \xi_{1m}$

$H(e_1'')$:

$w(e_1'', t_{m+1}) = t_m + \xi_{2m}$

$H(e_2')$:

$w(e_2'', t_{m+1}) = t_m + \xi_{2m}$

$H(e_2'')$:

$w(e_2'', t_{m+1}) = t_m + \xi_{3m}$

$H(e_3')$:

$w(e_3'', t_{m+1}) = t_m + \xi_{3m}$

$H(e_3'')$:

$w(e_3'', t_{m+1}) = t_m + \xi_{4m}$

$H(e_4')$:

$w(e_4'', t_{m+1}) = t_m + \xi_{4m}$

$H(e_4'')$:

$w(e_4'', t_{m+1}) = t_m + \xi_{5m}$

$H(e_5'')$:

$w(e_5'', t_{m+1}) = \infty$

$G(e_1'')$: $y = 0$

$G(e_2'')$: $y = y_1$

$G(e_3'')$: $y = y_2$

$G(e_4'')$: $y = y_4$

$G(e_5'')$: $y = 0$

Agregato Bankomatas aprašymas

1. Įėjimo signalų aibė:

$X = \{x_1, x_2, x_3, x_4\}$

x_1 – klientas paėmė pinigų;

x_2 – pinigai nurašyti nuo sąskaitos;

x_3 – trūksta pinigų sąskaitoje;

x_4 – neteisingas PIN kodas.

2. Išėjimo signalų aibė:

$Y = \{y_1\}$

y_1 – paimti kortelę.

3. Išorinių įvykių aibė:

$E' = \{e_1', e_2', e_3', e_4'\}$

e_1' – atėjo signalas x_1 (klientas paėmė pinigų);

e_2' – atėjo signalas x_2 (bankas nurašė pinigų nuo sąskaitos);

e_3' – atėjo signalas x_3 (atėjo signalas, kad trūksta pinigų sąskaitoje);

e_4' – atėjo signalas x_4 (atėjo signalas, kad neteisingas PIN kodas).

4. Vidinių įvykių aibė:

$E'' = \{e_1'', e_2''\}$

e_1'' – parodyti sąskaitos likutį;

e_2'' – atiduoti kortelę.

5. Valdymo sekos:

$$e_1'' \rightarrow \{\xi_{1i}\}, i = 0, \infty$$

$$e_2'' \rightarrow \{\xi_{2i}\}, i = 0, \infty$$

ξ_1 - laiko tarpas iki sąskaito likučio parodymo;

ξ_2 - laiko tarpas iki kortelės atidavimo.

6. Diskrečioji būsenos dedamoji:

$$v(t_m) = \emptyset$$

BPMN modelyje nėra išskirta diskrečioji būsena. Jos kintamųjų prasmė nepateikta jokių komentarų.

7. Tolydžioji būsenos dedamoji:

$$Z_v(t_m) = \{w(e_1'', t_m), w(e_2'', t_m)\}$$

$w(e_1'', t_m)$ – sąskaitos likučio parodymo laiko momentas;

$w(e_2'', t_m)$ – kortelės atidavimo laiko momentas

8. Pradinė būsena:

$$k(t_0) = 0, Z_v(t_m) = \{\infty, \infty\}$$

9. Perėjimo ir išėjimo operatoriai:

$H(e_1')$:

$$w(e_1'', t_{m+1}) = t_m + \xi_{1m}$$

$H(e_1'')$:

$$w(e_1'', t_{m+1}) = t_m + \xi_{2m}$$

$H(e_2')$:

$$w(e_1'', t_{m+1}) = t_m + \xi_{1m}$$

$H(e_2'')$:

$$w(e_2'', t_{m+1}) = \infty$$

$H(e_3')$:

$$w(e_1'', t_{m+1}) = t_m + \xi_{1m}$$

$H(e_4')$:

$$w(e_2'', t_{m+1}) = \infty$$

$G(e_1'')$: $y = 0$

$G(e_2'')$: $y = y_1$

Agregato Bankas aprašymas

1. Įėjimo signalų aibė:

$$X = \{x_1, x_2, x_3\}$$

x_1 – autorizuoti PIN kodą;

x_2 – patikrinti sąskaitos likutį;

x_3 – nurašyti pinigus nuo sąskaitos.

2. Išėjimo signalų aibė:

$$Y = \{y_1, y_2, y_3, y_4, y_5\}$$

y_1 – teisingas PIN kodas;

y_2 – neteisingas PIN kodas;

y_3 – pinigų sąskaitoje užtenka;

y_4 – trūksta pinigų sąskaitoje;

y_5 – pinigai nurašyti nuo sąskaitos.

3. Išorinių įvykių aibė:

$$E' = \{e_1', e_2', e_3'\}$$

e_1' – atėjo signalas x_1 (atėjo signalas PIN kodo autorizavimui);

e_2' – atėjo signalas x_2 (atėjo signalas sąskaitos patikrinimui);
 e_3' – atėjo signalas x_3 (atėjo signalas pinigų nurašymui nuo sąskaitos).

4. Vidinių įvykių aibė:

$$E'' = \{e_1'', e_2'', e_3''\}$$

e_1'' – autorizavimas;

e_2'' – patikrinti sąskaitos likutį;

e_3'' – pinigus nurašyti nuo sąskaitos.

5. Valdymo sekos:

$$e_1'' \rightarrow \{\xi_{1i}\}, i = 0, \infty$$

$$e_2'' \rightarrow \{\xi_{2i}\}, i = 0, \infty$$

$$e_3'' \rightarrow \{\xi_{3i}\}, i = 0, \infty$$

ξ_1 - laiko tarpas iki autorizavimo;

ξ_2 - laiko tarpas iki sąskaitos likučio patikrinimo;

ξ_3 - laiko tarpas iki pinigų nurašymo nuo sąskaitos.

6. Diskrečioji būsenos dedamoji:

$$v(t_m) = \emptyset$$

BPMN modelyje nėra išskirta diskrečioji būsena. Jos kintamųjų prasmė nepateikta jokių komentarų.

7. Tolydžioji būsenos dedamoji:

$$Z_v(t_m) = \{w(e_1'', t_m), w(e_2'', t_m), w(e_3'', t_m)\}$$

$w(e_1'', t_m)$ – autorizavimo laiko momentas;

$w(e_2'', t_m)$ – sąskaitos patikrinimo laiko momentas;

$w(e_3'', t_m)$ – pinigų nurašymo laiko momentas.

8. Pradinė būsena:

$$k(t_0) = 0, Z_v(t_m) = \{\infty, \infty, \infty\}$$

9. Perėjimo ir išėjimo operatoriai:

$H(e_1')$:

$$w(e_1'', t_{m+1}) = t_m + \xi_{1m}$$

$H(e_1'')$:

$$w(e_1'', t_{m+1}) = t_m + \xi_{2m}$$

$H(e_2')$:

$$w(e_2'', t_{m+1}) = t_m + \xi_{2m}$$

$H(e_2'')$:

$$w(e_2'', t_{m+1}) = t_m + \xi_{3m}$$

$H(e_3')$:

$$w(e_3'', t_{m+1}) = t_m + \xi_{3m}$$

$H(e_3'')$:

$$w(e_3'', t_{m+1}) = \infty$$

$G(e_1'')$: $y = y_1$ arba $y = y_2$

$G(e_2'')$: $y = y_3$ arba $y = y_4$

$G(e_3'')$: $y = y_5$