

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond  
Tarkvarateaduse instituut

Ron Ojava 153563IABM

**SIMULATSIOONI ABIL KONTROLLITUD  
BPMN STANDARDI PROTSESSIMUSTRID  
ÕPPENÄIDETENA**

Magistritöö

Juhendaja: Tarmo Veskioja  
teadur

Tallinn 2017

## **Autorideklaratsioon**

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Ron Ojava

08.05.2017

## **Annotatsioon**

Käesoleva töö eesmärk on anda ülevaade peamistest BPMNi antimustritest ning nendele vastanduvate mustrite põhjal konstrueerida õppenäited. Lisaks uuritakse, kas tarkvaraarenduses kasutatav testjuhitud meetodika võiks sobida ka tööprotsesside modelleerimise keskkonda.

Olulisemad töös käsitletavad probleemid on seotud üldiste tööprotsesside mustrite transformeerimisega CASE-vahendi disainile sobivaks. Lisanduvad üldised protsesside analüüsid, korduvate ja katkestavate elementide käitumise uurimine simulatsioonimootoris ning testjuhitud meetodika põhine lähenemine mudelite koostamisel.

Töö tulemusena valmivad BPMN 2.0 standardile vastavad mustripõhiselt konstrueeritud õppenäited, mida saab kasutada Tallinna Tehnikaülikooli õppeaines „Äriprotsesside modelleerimine ja automatiseerimine“. Õppenäited on simuleeritavad ja võimaldavad analüüsida protsesside ressursikulu.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 48 leheküljel, 6 peatükki, 55 joonist, 1 tabelit.

# **Abstract**

## **Simulation tested BPMN Process Patterns as Learning Examples**

The aim of this thesis is to give an overview of the main antipatterns used in business process modeling while using Business Process Modeling Notation (BPMN) and to construct models that could be used as learning materials. In addition, the suitability of test-driven modeling technique will be assessed.

The main problems dealt with are transforming general workflow patterns to be compatible with the CASE tool, analysing processes and creating process models, examining the behaviour of multi-instance tasks and interrupting events and modeling in accordance with test-driven principles.

The main results of the work are pattern-based process models which are in accordance with BPMN 2.0 standard and can be used as learning materials in the course „Business Process Modeling and Automation“. The models are also compatible with the simulation engine of the CASE tool used in the course.

The thesis is in Estonian and contains 48 pages of text, 6 chapters, 55 figures, 1 table.

## Lühendite ja mõistete sõnastik

Antimuster [1]	Halb lahendus probleemile.
BPMN [3]	<i>Business Process Model and Notation</i> , märgisüsteem protsesside modelleerimiseks.
BPEL [2]	<i>Business Process Execution Language</i> , keel millega saab protsesse realiseerida.
Moodul [20]	Keerulisema struktuuriga seadme, hoone, eseme vms. terviklik standardne üksus või osa.
Mudel [20]	Originaalobjektiga kindlas vastavuses olev tehisobjekt, ka skeem, seoste matemaatiline kirjeldus vms.
Muster [1]	Hea lahendus probleemile.
Protsess [20]	Teatud tulemuseni jõudev asjade v. nähtuste muutumine, teisenemine, liikumine, areng.
Simulatsioon [9]	Töövahend mudeli käitumise hindamiseks üle pikema ajavahemiku ning erinevate tingimuste juures, et vähendada ebaõnnestumise riski ja suurendada tingimustele vastavust, elimineerida tupikuid ja ennetada ressursivõimekuse ülehindamist ning optimeerida protsessi.
Testjuhitud meetoodika	Tarkvaraarenduse <i>Test-driven</i> meetoodika põhimõtetel põhinev tehnika, kus mudelid koostatakse simulatsioonimootoris toimivatest moodulitest.
UML [7]	Unified Modeling Language, modelleerimiskeel, millega saab protsesse kirjeldada.
XPDL [3]	<i>XML Process Definition Language</i> , keel millega saab protsesse realiseerida.

## Sisukord

1 Sissejuhatus .....	11
1.1 Äriprotsesside modelleerimine .....	12
1.2 BPMN.....	13
1.3 Mustrid äriprotsesside modelleerimises .....	14
1.4 Simulatsioonid modelleerimises.....	17
2 BPMNi mustrid ja antimustrid .....	19
2.1 Kaudne lahknemine ( <i>Implicit Parallel Split</i> ) .....	20
2.2 Kaudne mittevälstav lahknemine ( <i>Implicit Inclusive Split</i> ) .....	21
2.3 Mitmekordsed kaudsed lüüsid ( <i>Multiple Implicit Gateways</i> ) .....	22
2.4 Kaudsed ühenduslüüsid ( <i>Implicit Join Gateways</i> ) .....	23
2.5 Kombineeritud ühendus ja lahknemislüüsid ( <i>Combined Join-Split Gateway</i> ).....	24
2.6 Mitmekordne vastuvõttev sündmus ( <i>Multiple Catching Event</i> ).....	25
2.7 Mitmekordne saattev sündmus ( <i>Multiple Throw Event</i> ) .....	26
2.8 Mitmekordne paralleelne sündmus ( <i>Multiple Parallel Event</i> ).....	27
2.9 Puuduv algsündmus ( <i>No Start Event</i> ).....	28
2.10 Puuduv lõppsündmus ( <i>No End Event</i> ).....	29
2.11 Mitmekordsed algsündmused ( <i>Multiple Start Events</i> ) .....	30
2.12 Paralleelsetesse moodulitesse sisenevad vood ( <i>Flows into Parallel Blocks</i> ).....	31
2.13 Paralleelsetest moodulitest väljuvad vood ( <i>Flows out of Parallel Blocks</i> ).....	32
2.14 Tsüklid ( <i>Loops</i> ) .....	33
2.15 Sõnumite ja saatvate tegevuste kombineerimine ( <i>Mixing Message Events and Tasks</i> ).....	34
2.16 Tegutseja rajal süsteemi tegevuse kujutamine ( <i>Service Task in User Lane</i> ).....	35
2.17 Siduvate sündmuste kasutamine ( <i>Link events</i> ) .....	36
2.18 Ekraanivoogude kujutamine ( <i>Screen Flows</i> ).....	36
2.19 Puuduv/tühi sündmus ( <i>None Event</i> ) .....	37
3 Mudelite koostamise nõuded .....	37
3.1 Testjuhitud tarkvaraarenduse meetoodika.....	37

3.1.1	Metoodika kohaldamine modelleerimisele.....	38
3.2	Nõuded mudelitele.....	38
4	Õppenäidete mudelite kirjeldused .....	40
4.1	Telekommunikatsiooniettevõtte erakliendi äriprotsess .....	41
4.1.1	Ühise basseini mudel.....	42
4.1.2	Eraldi basseinidega mudel.....	46
4.2	Õppeaine läbimise protsess .....	48
4.3	Klienditeeninduse protsess kohvikus .....	51
4.4	Abstraktne katkeva vooga protsess .....	53
5	Uuringu tulemuste analüüs .....	55
5.1	Mustrite rakendamise võimalikkus.....	55
5.2	Testjuhitud modelleerimise metoodika efektiivsus .....	56
6	Kokkuvõte .....	58
	Summary.....	59
	Kasutatud kirjandus .....	60
	Lisa 1 – Õppenäidete mudelid.....	63

## Jooniste loetelu

Joonis 1. BPMNi elemendid [3] .....	14
Joonis 2. Bizagi Modeleri simulatsioonimootori viga.....	18
Joonis 3. Kaudse lahknemise antimuster [8] .....	20
Joonis 4. Korrekne lahknemise muster [8][13][14][15] .....	20
Joonis 5. Kaudse mittevälitava lahknemise antimuster [8].....	21
Joonis 6. Mittevälitava lahknemise muster [8][13][14][15] .....	21
Joonis 7. Mitmekordsete kaudsete lüüside antimuster [8].....	22
Joonis 8. Mitmekordsete lüüside muster [8].....	22
Joonis 9. Kaudse ühenduslüüsi antimuster [8] .....	23
Joonis 10. Kaudse ühenduslüüsi muster [8][13][14][15] .....	23
Joonis 11. Kombineeritud ühendus- ja lahknemislüüsi antimuster [8] .....	24
Joonis 12. Kombineeritud ühendus- ja lahknemislüüsi muster [8][13][14][15] .....	24
Joonis 13. Mitmekordse vastuvõtva sündmuse antimuster [8].....	25
Joonis 14. Mitmekordse vastuvõtva sündmuse muster [8][14][15] .....	25
Joonis 15. Mitmekordse väljastava sündmuse antimuster [8] .....	26
Joonis 16. Mitmekordse väljastava sündmuse muster [8] .....	26
Joonis 17. Mitmekordse paralleelse sündmuse antimuster [8] .....	27
Joonis 18. Mitmekordse paralleelse sündmuse muster [8][14] .....	27
Joonis 19. Puuduva algsündmuse antimuster [8].....	28
Joonis 20. Protsessi alguse muster [8][13][14][15] .....	28
Joonis 21. Puuduva lõppsündmuse antimuster [8] .....	29
Joonis 22. Korrekne protsessi lõppsündmuse muster [8][13][14][15] .....	29
Joonis 23. Mitmekordsete algsündmuste antimuster [8] .....	30
Joonis 24. Mitmekordsete algsündmuste muster [8][13][14][15] .....	30
Joonis 25. Paralleelsetesse moodulitesse sisenevate voogude antimuster [8] .....	31
Joonis 26. Paralleelsetesse moodulitesse sisenevate voogude muster [8][13][14][15] ..	31
Joonis 27. Paralleelsetest moodulitest väljuvate voogude antimuster [8] .....	32
Joonis 28. Paralleelsetest moodulitest väljuvate voogude muster [8][13][14][15] .....	32
Joonis 29. Tsüklite antimuster [8] .....	33



Joonis 30. Tsükli muster alamprotsessina [8][13][14][15].....	33
Joonis 31. Dubleeriva sõnumivahetuse antimuster [8].....	34
Joonis 32. Dubleeriva sõnumivahetuse muster [8].....	34
Joonis 33. Integreeritud ujumisradade antimuster [8] .....	35
Joonis 34. Ujumisradade muster [8].....	35
Joonis 35. Siduvad sündmused kui antimustrid [8].....	36
Joonis 36. Ekraanivoog kui protsess [8].....	36
Joonis 37. Puuduv sündmus [8].....	37
Joonis 38. Kontseptuaalne telekommunikatsiooniettevõtte erakliendi äriprotsess .....	41
Joonis 39. Reusable-tüüpi alamprotsessidega simulatsioon.....	42
Joonis 40. Embedded-tüüpi alamprotsessi elementide simulatsioon.....	43
Joonis 41. Reusable-tüüpi alamprotsess .....	44
Joonis 42. Embedded-tüüpi alamprotsess.....	44
Joonis 43. Seadme hõivatuse hindamine .....	45
Joonis 44. Seadme kasutamise alamprotsess .....	45
Joonis 45. Seadme tagastamise alamprotsess .....	45
Joonis 46. Mitme basseini mustri elementide liigsus .....	47
Joonis 47. Liigsuse vähendamine mitme basseiniga mudelis.....	47
Joonis 48. Kahe katkestava sündmusega alamprotsess põhiprotsessi tasemel.....	49
Joonis 49. Kahe katkestava sündmusega alamprotsessi detailne vaade .....	49
Joonis 50. Kahe katkestava sündmusega alamprotsessi sisaldava mudeli simulatsioon	49
Joonis 51. Õppeaine läbimise protsessi simulatsioonitulemused .....	50
Joonis 52. Mitmekordsete algsündmuste muster .....	51
Joonis 53. Mitmekordsete algsündmuste antimuster .....	52
Joonis 54. Mitme algsündmusega kohviku klienditeeninduse põhiprotsess .....	53
Joonis 55. Katkeva vooga protsess .....	54

## **Tabelite loetelu**

Tabel 1. Modelleerimise meetodikate ligikaudne ajakulu.....	57
---	----

# 1 Sissejuhatus

Tänapäeva äriprotsesside agiilsus on tekitanud olukorra, kus nende modelleerijad peavad pidevalt muutma kaardistatud protsesside mudeleid, et saada võimalikult informatiivne tulem. Kuna protsessivoogu on võimalik ka simuleerida, siis mudelite täpsusest sõltuvad otseselt hinnangud ressursside kuludele. Seetõttu on oluline, et loodud mudelid töötaksid ka simulatsiooni oludes ning annaksid nendega töötavatele inimestele rohkem teavet kui pelgalt visuaalse ülevaate äriprotsessidest.

Selleks, et luua mudeleid, mis toimiksid ka simulatsioonides, on vaja järgida vastava modelleerimiskeele reegleid. Kuid tihti peale ei piisa ainuüksi reeglite järgimisest, et mudeli simulatsioon toimiks. Põhjuseid võib olla mitmeid, kuid enamasti on selleks modelleerimistarkvara iseärasused. Seetõttu tasub modelleerimisel lisaks reeglitele jälgida ka mustreid. Muster on idee, mis on olnud kasulik mingis ühes kontekstis ja on arvatavasti kasulik teistes kontekstides [1]. Kui muster kirjeldab halba lahendust probleemile, siis nimetatakse seda antimustriks [1].

Käesoleva lõputöö eesmärk on uurida ja kirjeldada äriprotsesside modelleerimise notatsiooni *Business Process Model and Notation* (edaspidi BPMN) mustreid ning neid kasutades täiendada ja luua õppenäiteid, mille käitumist oleks võimalik simuleerida Bizagi Modeleri tarkvara abil. Lisaks püüab autor välja selgitada, kas eespool mainitud modelleerimiskeelt ja tarkvara kasutades oleks võimalik efektiivselt rakendada testjuhitud („test driven“) meetodikat, kus äriprotsessi mudel luuakse moodulitest, mille simulatsioon toimib.

Huvitatud poolteks oleksid eelkõige Tallinna Tehnikaülikooli äriinfotehnoloogia eriala tudengid, kes saaksid loodud materjali kasutada õppeaines „Äriprotsesside modelleerimine ja automatiseerimine“, ning samuti selle aine õppejõud.

Töö jaguneb kolmeks osaks: esimeses osas uuritakse teoreetilist tausta ning tutvustatakse levinumaid mustreid ja antimustreid. Teises osas kirjeldatakse modelleerimisel kasutatavat meetodikat ning rakendatakse teoreetilisi teadmisi praktikas, et täiendada õppenäiteid ja leida mustreid mudelitele, mille simulatsioon ei toimi. Rakendamise käigus

uuritakse lisaks mustritele, kas mudeli jagamine alamosadeks ja modelleerimine protsessi alamosade kaupa on efektiivsem kui tervikliku protsessi kohene modelleerimine. Viimases osas esitatakse autori järeldused ja soovitused, mis põhinevad eksperimendi tulemustel.

## **1.1 Äriprotsesside modelleerimine**

Kuidas teha toiminguid kiiremini ja efektiivsemalt? See on küsimus, mis on evolutsiooni tagant tõuganud juba inimkonna ajaloo algusest saadik. Produktiivse tegevuse tulemi maksimaliseerimine on alati olnud inseneeria esmane eesmärk. Sellist eesmärki täidab ka protsesside modelleerimine.

Äriprotsesside modelleerimine on läbimõeldud tegevus, et parandada ettevõtte toimimise efektiivsust. See termin on aktuaalseks saanud ennekõike infotehnoloogia ja telekommunikatsiooni tehnoloogia ülikiire arenguga alates 1990. aastatest. Tänu interneti arengule ja veebi leiutamisele tekkis organisatsioonidel võimalus jõuda enda klientidele lähemale ning pakkuda neile oluliselt mugavamaid teenuseid. Samal ajal kerkisid esile ka protsesside automatiseerimise võimalused.

Infotehnoloogia arengule ei ole piiri ette tulnud ning võimalusi, kuidas kasvatada ettevõtte käivet ja kasumit, tekib aina juurde. Tänapäeva ettevõtted on tugeva surve all, et mitte jääda ümbritseva keskkonna tehnoloogilises arengus maha ja et kasutada kõiki võimalusi, mida see areng pakub. Seetõttu on pidevas muutumises ka äriprotsessid. [2]

Modelleerimisega tegelevad enamasti spetsialistid kolmest erinevast grupist: ärianalüütikud, süsteemiarhitektid ja arendajad. Nende kolme valdkonna ekspertide töö tulemusena sünnib süsteem, mis võimaldab muuta protsessi efektiivsemaks. Ärianalüütiku kaardistatud protsesside töövood ja nende täiustused on aluseks sellise süsteemi kontseptuaalse mudeli väljatöötamisele. Süsteemiarhitekt muudab kontseptuaalse mudeli arhitektuuriliseks mudeliks ning arendajad teostavad selle ehituse ja implementatsiooni. [2]

Samas ei ole eespool mainitud spetsialistid ainukesed, kes osalevad protsesside modelleerimises või kes vajavad täielikku arusaama nende toimimisest. Lisaks IT spetsialistidele vajavad seda ka ettevõtte omanikud, juhid ja teised töötajad ehk

niinimetatud äripool. Seetõttu on välja töötatud mitmed modelleerimiskeeled ja modelleerimise notatsioonid, millest enim levinud on BPMN. [3]

## 1.2 BPMN

BPMN loodi 2004. aastal Business Process Management Initiative'i poolt. Peamine eesmärk oli luua üheselt mõistetav märgisüsteem, mida mõistaksid nii ärianalüütikud, arendajad kui ka ettevõtte töötajad, kellel ei ole IT-alaseid spetsiifilisi teadmisi. Suure huvi tõttu adopteeriti BPMN standardina 2006. aastal. Hetkel kehtib versioon BPMN 2.0. [4]

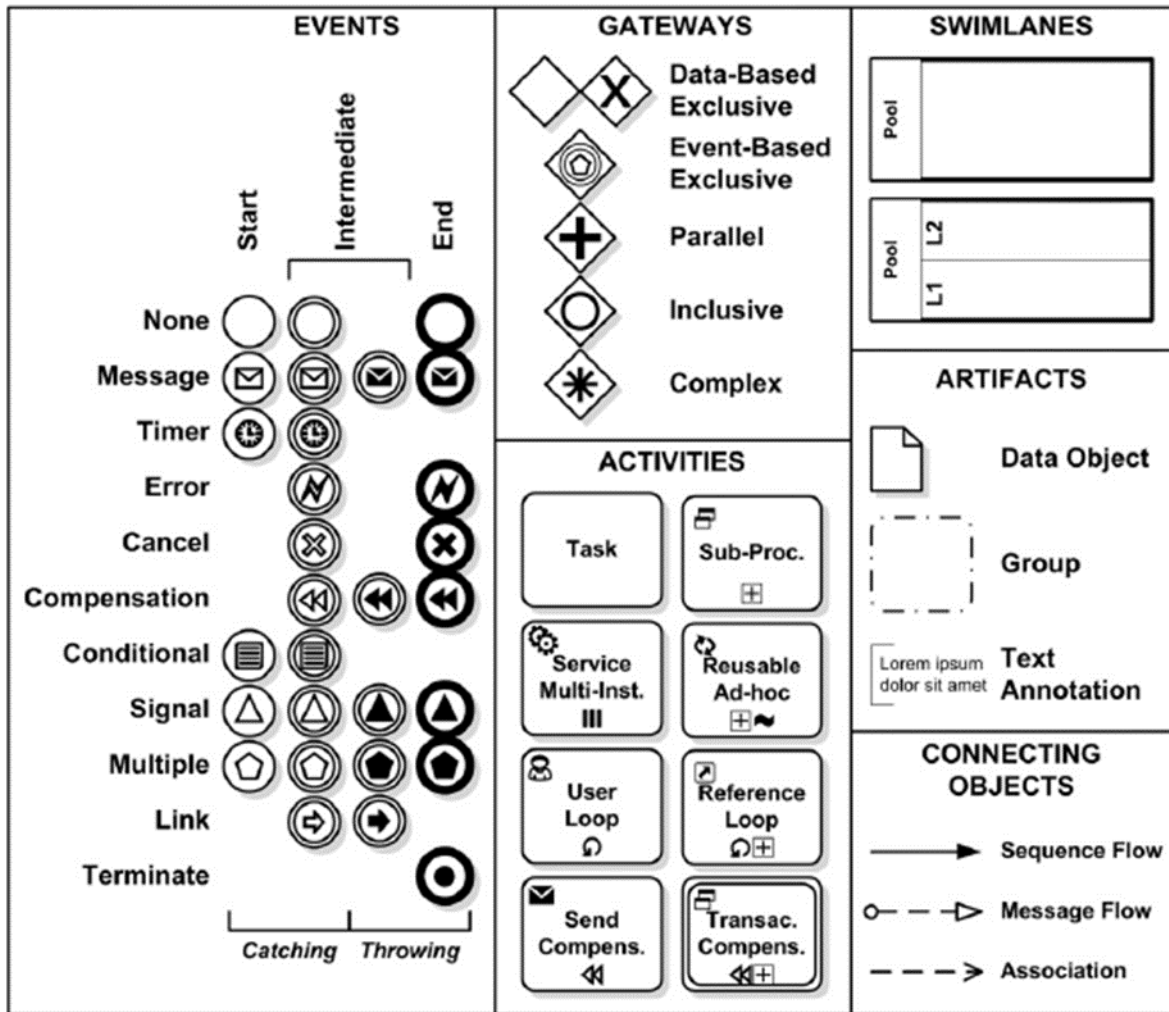
BPMNi tugevused seisnevad selle sobivuses erinevatele pooltele. Ärivaldkonna eksperdid saavad tuvastada protsessi graafilise kujutise õigsuse ning selles leiduvad kitsaskohad, nagu pudelikaelad, katkestavad tingimused ja tsüklid. Analüütikutel on võimalik koguda andmeid ressursside kasutuse kohta, et tuvastada, kas protsessi on võimalik optimeerida või mitte, ning arendajad huvituvad võimalustest tõlkida diagramm masinloetavasse keelde jagamis- või käivituseesmärkidel. Sellisteks keelteks on enamasti XPDL või BPEL [3].

Olemas on mitmeid modelleerimise notatsioone, mis pakuvad protsesside kujutamiseks erinevaid graafilisi sümboleid, kuid mille põhielemendid (tegevused, vood) on samad. Nende notatsioonide intuiitivsus on tihtipeale küsitav ja oleneb kasutajate tarkvaralisest töökeskkonnast. Ka BPMNi puhul on uuringutes välja toodud, et elementidest arusaamine ning nende abil mudelite koostamine võib olla liialt keeruline isegi kogunud ärianalüütikutele [4], kuid protsesside kirjeldamise aspektist vaadatuna on arvatavasti tegemist parima modelleerimise notatsiooniga käesoleval ajal. [3] [4] [5]

BPMNi elemendid jagunevad viide kategooriasse: tegevusvoo elemendid, ühenduselemendid, ujumisrajad, artefaktid ja infoelemendid. Tegevusvoo elemendid esindavad kõikvõimalikke tegevusi, mis võivad aset leida protsessi käigus ja mis määravad selle käitumise. Tegevusvoo elementide alla kuuluvad sündmused (Events), tegevused (Activities) ja lüüsid (Gateways). Ühenduselementide tüübid jagunevad kolmeks: jadavoo (Sequence flow), sõnumivoo (Message flow) ja assotsiatsiooni (Association) elemendid. Ujumisrajad (Swimlanes), kuhu kuulub ka basseini element (Pool), annavad võimaluse grupeerida protsessi käigus kasutatavaid teisi elemente ja

defineerida tegutsejaid. Artefaktid (Artifacts) ehk tehised illustreerivad protsessi kohta käivat lisainformatsiooni (nt dokumendiringlus), mis ei mõjuta üldist protsessivoo toimimist. Infoelementide (Data) grupp tekkis BPMN 2.0 standardi kasutuselevõtuga ning sinna alla kuuluvad tehiste grupist eemaldatud andmekollektsioonide ja sõnumite elemendid. [3]

BPMNi põhielementide grupid on kirjeldatud joonisel 1.



Joonis 1. BPMNi elemendid [3]

### 1.3 Mustrid äriprotsesside modelleerimises

Kuigi sissejuhatuses esitatud mustri definitsioon sobib kasutamiseks igas valdkonnas, võiks siinkohal seda veidi teistmoodi sõnastada. Mustrid on mudelid, mis on piisavalt üldistavad, kergesti modifitseeritavad ja imiteerimise väärilised [6]. Nad kujutavad

ametlikke lahendusi spetsiifilistele probleemidele, säilitades samal ajal suure abstraktsuse, mis on vajalik adapteerumiseks erinevate probleemide korral [5].

Koos Unified Modeling Language'ga (edaspidi UML) on BPMN peamiseks notatsiooniks, millega protsesse modelleeritakse. BPMNi eelised peituvad tema semantilises rikkuses. Kui UMLi tegevusdiagrammil on umbes 20 elementi, siis BPMNil on neid ligikaudu 100. Selline võimaluste rohkus tekitab aga probleeme, sest CASE-vahenditel puuduvad efektiivsed mudelite valideerimise funktsioonid, mistõttu ei ole raske konstrueerida puudulikke või valesid mudeleid. [7]

Tüüpiline äriprotsessi modelleerimise projekt algab ärianalüütikust, kes koos lõppkasutajatega ning valdkonna ekspertidega töötab välja esialgse BPMNi mudeli [8]. Algne mudel on tihtipeale äriprotsessi „as is“ ehk olemasoleva ja reaalselt toimiva protsessi imitatsioon. Tavaliselt jäädakse selle mudeli juures piiratud arvu lihtsakoeliste elementide juurde, mis on kirjeldusele vastaval algtasemel. Tegemist on ärianalüüsi esimese etapiga ja sellisel tasemel ei saa mustreid rakendada.

Järgnevalt asub analüütik protsessi optimeerima, mille tulemusena sünnib „to be“-mudel. Siin tuleks eristada kahte erinevat modelleerimise taset. Esimene on protsessi loogiline disain, mis on BPMNi standardi keskne. Sellel tasemel saab rakendada standardile vastavaid tööprotsesside mustreid, sest modelleerijal on olemas teadmised protsessi toimimisest. Samas peab analüütik arvestama protsessi füüsilise disaini tasemega, mis sõltub modelleerimisrakendusena kasutatavast CASE-vahendist. Loogilise disaini taseme mustrid võivad, aga ei pruugi olla ülekantavad füüsilise disaini tasemele. Rakenduse arhitektuurilised omadused võivad seada piiranguid äriprotsessi detailanalüüsi mudelile. Näiteks ei sobi osa BPMNi mustrid kasutamiseks Bizagi Modeleri rakendusega.

Kui lõppkasutajad on seisukohal, et nad mõistavad mudelit, ning analüütik on enda arvates piisavalt detailselt protsessi kirjeldanud, antakse mudel arendajatele täideviimiseks. Pärast teostust verifitseerib äripool tulemit, et kinnitada nõuetele vastavust. Selles etapis tulevad esile sisemised vastuolud äripoole ja arendajate oskustes. Tänu laiematele teadmistele BPMNi notatsioonist võib arendav pool muuta mudelit oluliselt keerulisemaks. Olukorda raskendab tõsiasi, et BPMN lubab tihtipeale mitut eri viisi, kuidas sama tulemusega diagrammi koostada, ning sisaldab ka mitmeid „otseteid“ ehk elemente, mis peidavad olulisi detaile. Tulemuseks on enamasti suurenenud ajakulu,

halvematel juhtudel ka protsesside valesti tõlgendamine ja läbikukkunud projekt. Kasutades modelleerimise häid tavasid, väheneb selliste olukordade esinemissagedus märgatavalt. Langetades valiku diagrammi mõistetavuse kasuks olukordades, kus tuleb erinevaid modelleerimistehnikaid rakendada, suureneb ka arusaamine kliendi ja arendaja vahel ning väheneb üksteisest möödarääkimise oht. Peamine viis, kuidas seda teha, on vältida BPMNi antimustreid. [8]

Järgnevalt on välja toodud peamised põhjused, miks mustreid kasutatakse.

- Töö lihtsustamine. Mustrid tagavad kohese kasuteguri, mis on tingitud minimaliseeritud disainist ja selle integratsioonist. Lisaks tagab nende kasutamine tihtipeale pikaajalise järjepidevuse protsessi disainis. [6]
- Modelleerimise heade tavade propageerimine. Disaine kutsutakse mustriteks, kui nad kujutavad protsessi käitumise eelistatud viise, mistõttu nad on väärt imiteerimist. Seetõttu on mustrid alati „to be“-mudelite kandidaadid. [6]
- Protsesside analüüsimine. Mustrid aitavad näha seoseid protsesside vahel, kus neid esmapilgul ei näi olevat. Näiteks on võimalik leida seoseid auto- ja arvutitööstuse vahel, kus toodete kokkupanekuprotsessi etapid on sisuliselt samad. Niiviisi on võimalik pakkuda välja lahendusi ettevõtete omanikele, kuidas tootmishooneid mitmekülgsemalt rakendada. [6]
- Ebaefektiivsuse tuvastamine. Võttes aluseks paki kohaletoimetamise mustri, saab välja tuua, et kuller saadab paki asukohast samal ajal teavet nii kliendile kui ka ettevõttele, kust klient kauba ostis. Mõlemale poolele edastust on vaja, et kauplus saaks kliendi päringutele vastata. Samas leidub kontrastina selliseid „as is“-mudeleid, kus teavitust läheb ainult müüjale, kes siis edastab selle kliendile. Võrreldes sellist mudelit mustriga, on kerge näha protsessi ebaefektiivsust, mida saaks tõhustatud „to be“-mudelis vältida. [6]
- Protsessist üleliigsete etappide eemaldamine. Näiteks võib tulla ette, et nii müüja kui ka kuller saavad kliendile informatsiooni kauba liikumise kohta. Selline toiming on aga üleliigne ega oma lõpptarbija jaoks suuremat väärtust. [6]
- Erinevate süsteemide sidususe tugevdamine. Ühise mustri kasutamine protsessi disainimisel lubab eri valdkondade protsessidel kasutada sama süsteemi. Sama süsteemi



kasutamine informatsiooni vahetamiseks alandab oluliselt uute lahenduste kasutuselevõtu ja ülalhoiu kulusid. [6]

- Modulaarsuse ja läbipaistvuse suurendamine. Mida rohkem disainitakse mustreid, mis oleksid võimalikult suurel määral taaskasutatavad, seda suuremat kasu toob nende järgimine. Mustrite üldistamine loob väga head väljavaated äriprotsesside modulaarse arhitektuuri arenguks, vähendades samal ajal uute lahenduste modelleerimise, kasutuselevõtu ja hilisema ülalhoiu kulusid. Näiteks muutub lihtsamaks ühe teenuse asendamine teisega, et saavutada eesmärgid kvaliteedis või eelarves. Samuti kergeneb sisemiste funktsionaalsuste ülevõtmine väliste teenuste poolt. [6]

#### **1.4 Simulatsioonid modelleerimises**

Kõik populaarsemad BPMNi modelleerimisvahendid pakuvad võimalust kontrollida kasutaja disainitud mudeli valiidsust ehk standardile vastavust [3]. Esmapilgul ei pruugi selline nüanss tunduda olulisena, kuid selle mõju mudeli simulatsioonile ei saa alahinnata. Seda lihtsal põhjusel: kui valideerimine ei õnnestu, siis üldjuhul ei toimi ka simulatsioon. Valideerimise korral kontrollitakse elementide ühendusi ja asukohta. Samas ei anna valideerimise õnnestumine mitte mingisugust garantiid, et mudel toimib simulaatoris. Probleem seisneb eespool mainitud tarkvara efektiivsuses. Protsessi loogikat ei kontrollita, sest iga protsess võib olla potentsiaalselt ebastandardne ning olla õige lahendus [9].

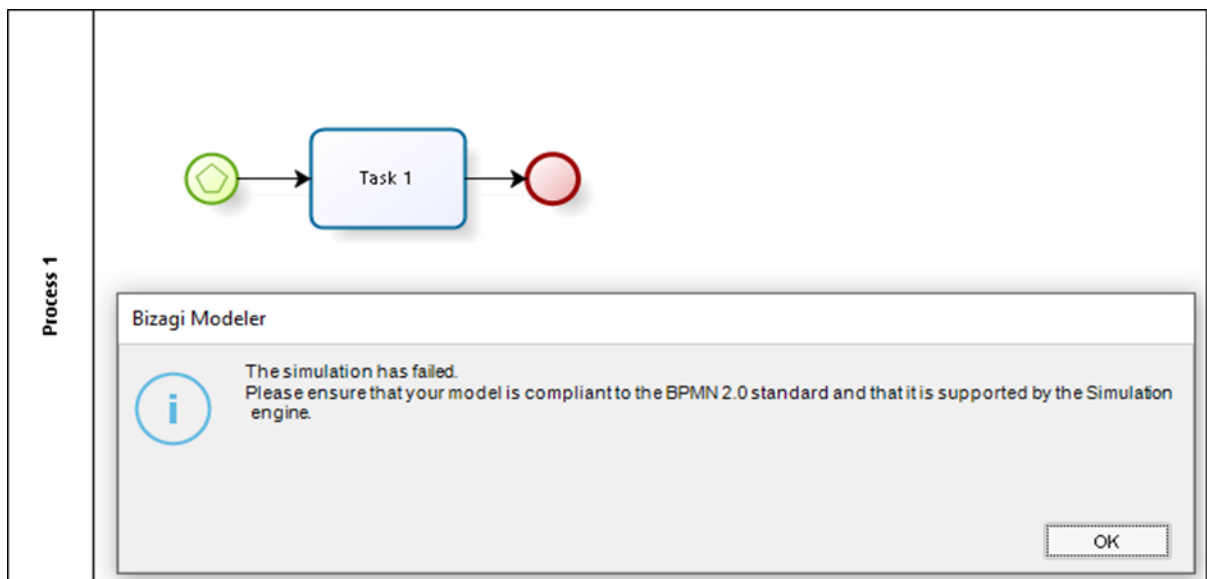
Itaalias asuvas Insubria Ülikoolis korraldati uuring, kus küsitluse käigus keskenduti kolmele aspektile: BPMNi kasutamine üldistes kontekstides, kasutajate teadmised BPMNi elementidest ning ülevaate andmine enim levinud ja eelistatud BPMNi modelleerimisvahenditest. Uuringu käigus küsitleti 30 analüütikut. Kuigi uuringu peamised teemad ei ole käesoleva töö jaoks esmatähtsad, tuli selle käigus välja mitmeid olulisi fakte. [3]

Tulemuste kohaselt peab 56% vastanuist ülioluliseks, et mudelid oleksid alati valiidsed. 20% seostas valideeritud mudeleid selgema arusaamisega ja parema loetavusega. Uuringust tuli veel välja, et 60% vastanutest kasutab tarkvara, millel on võimekus ka simulatsioone jooksutada. Peamisteks põhjusteks, miks simulatsioone kasutati, olid

pudelikaelade, tupikute ja erandite leidmine protsessides. Vähem mainiti põhjustena aja- ja ressursikulu hindamist ning lõputute tsüklite avastamist. [3]

CASE-vahendite populaarsuse väljaselgitamiseks kõrvutati küsitluse valimi tulemused LinkedIn'i keskkonnas korraldatud uuringuga, kus küsiti enam kui 40 vastaja käest, mis on nende eelistatud BPMNi modelleerimise vahend. Eesmärk oli eristada analüütikute ja teiste erialade esindajate eelistusi. Bizagi Modeler osutus populaarseimaks nii analüütikute (13,21%) kui ka tavakasutajate (17,07%) seas. Kombineeritud tulemus (30,28%) edestas oluliselt organisatsioonides väga levinud CASE-vahendi Enterprise Architecti tulemust (9,76%). [3]

Simulatsioonide rakendamisel tuleb tunda füüsilise disainitaseme ehk tarkvaratootjate iseärasusi. Alati ei piisa mudeli valideerimisest, et tagada simulatsiooni toimimine. Järgnevalt tuuakse joonisel 2 välja selline juhtum.



Joonis 2. Bizagi Modeleri simulatsioonimootori viga

Kujutatud on abstraktset protsessi, mis võib aset leida mitme erineva algündmuse tõttu. Mudel vastab BPMN 2.0 standardile, kuid ei käivitu simulatsioonimootoris. Põhjuseks on siinkohal Bizagi Modeleri ehituslikud anomaaliad, millele täpse seletuse oskavad anda vaid tarkvara loonud isikud. Arvatavasti taandub see siiski selliste elemendi seadistusvõimaluste puudumisele, mis vastaksid sümboli eesmärgile. Selliseid näiteid leidub veel ning üks võimalus, kuidas neid vältida, on mustrite kohaldamine tarkvarale vastavaks.

## 2 BPMNi mustrid ja antimustrid

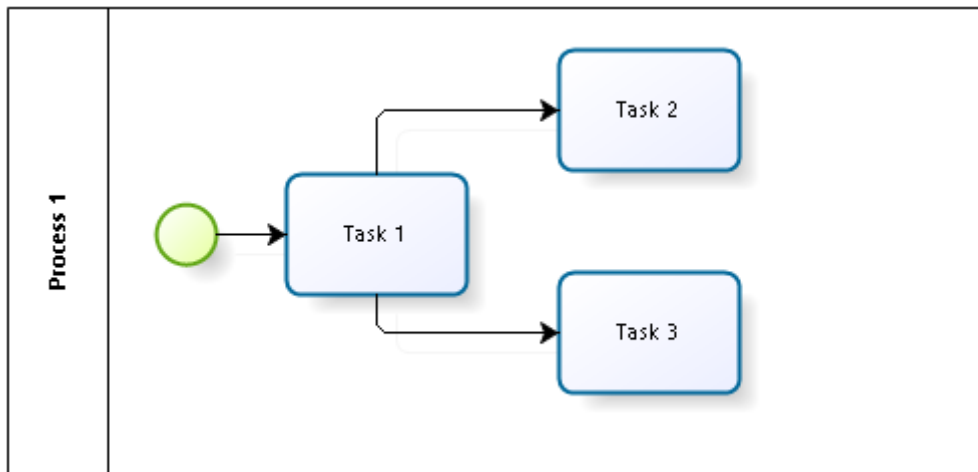
Äriprotsesside modelleerimise mustreid ei ole üheselt standardina kirjeldatud. Juhul, kui neid on vaja kasutada, peab enamasti otsima BPMNi modelleerimistarkvara tootjate kodulehekülgedelt. Seega peab arvesse võtma fakti, et enamasti on need mustrid kohaldatud ka vastavalt tarkvarale ehk tegemist on füüsilise disaini mustritega. See nüanss ei muuda neid valeks, kuid kindlasti mitte ka standardiks.

Antimustrite leidmisega võib probleeme tekkida veidikene rohkem, sest neid tootjad üldiselt ei kirjelda. Leidub küll mitmeid blogijaid ja teisi valdkonna eksperte, kes aeg-ajalt kirjeldavad antimustreid ja alternatiivseid viise, kuidas neid vältida, ning pakuvad ülevaateid ja lahendusi erinevatele situatsioonidele [10][11]. Neid lahendusi ei saa aga võtta teadustöö aluseks, sest tõenduskäik on tihtipeale puudulik või puudub üldse. Seetõttu peab uurima loogilise disainitaseme tööprotsesside mustreid, mis ei ole otseselt seotud ühegi modelleerimiskeelega, kuid on nendesse ülekantavad. Töövooprotsesside mustrite ja antimustrite kirjeldamisega tegelevad Eindhoven'i Tehnikaülikool ja Queensland'i Tehnikaülikool ühise projekti raames aastast 1999 [12].

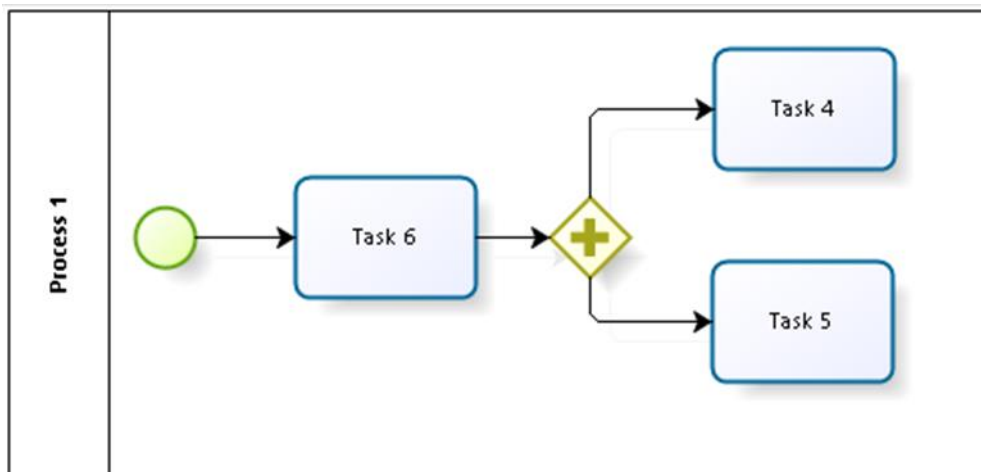
Antimustrite kirjeldamisel keskendutakse peamiselt ummikseisude (Deadlocks) ja teiste selliste praktikate vältimisele, mis võiksid takistada protsessi täielikku toimimist. Antimustrite hulka loetakse ka mustrid, mis halvendavad oluliselt mudeli mõistetavust, kuid ei pruugi olla otseselt BPMNi reeglitega vastuolus. [8]

Kuna käesolevas lõputöös keskendutakse eelkõige põhiliste antimustrite vältimisele ja nendele vastavate korrektsete mustrite abil õppenäidete koostamisele, siis ei kirjeldata järgnevalt ainult BPMNi mustreid, vaid ka antimustreid.

## 2.1 Kaudne lahknemine (*Implicit Parallel Split*)



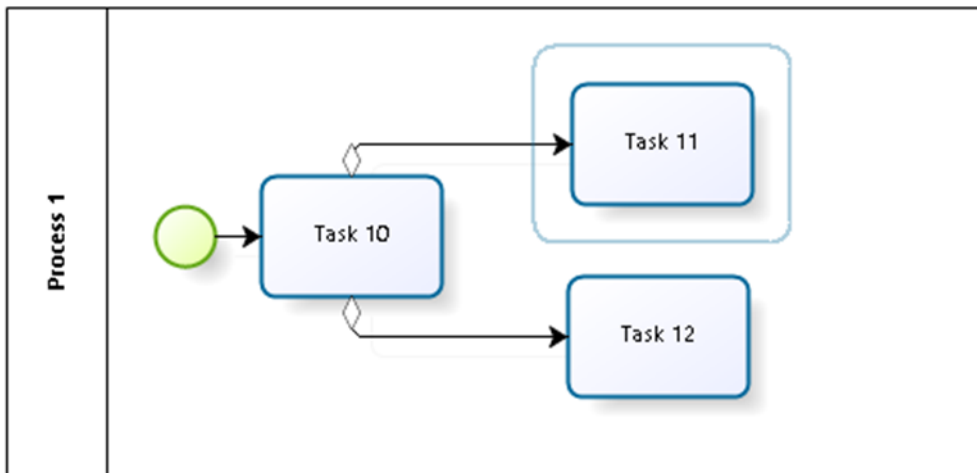
Joonis 3. Kaudse lahknemise antimuster [8]



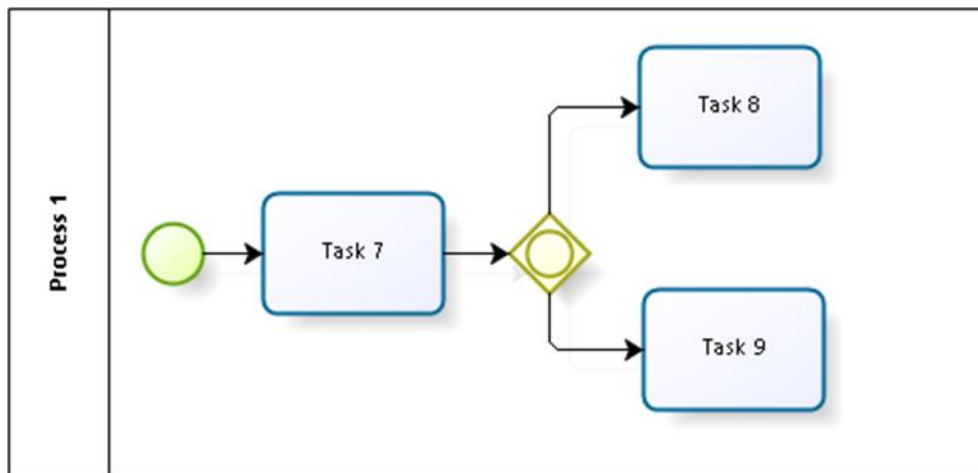
Joonis 4. Korrekne lahknemise muster [8][13][14][15]

Mitu erinevat järgnevusvoogu, mis väljuvad tegevusest või sündmusest, käituvad samamoodi nagu paralleelne lüüs. Erinevad modelleerimistarkvarad võivad tõlgendada antimustrit üksteisest erinevalt ning seetõttu võib mudel töötada ebakorrektelt. Paralleelset lüüsi kasutades see probleem kaob ja samuti suureneb mudeli loetavus. [8]

## 2.2 Kaudne mittevälstav lahknemine (*Implicit Inclusive Split*)



Joonis 5. Kaudse mittevälstava lahknemise antimuster [8]

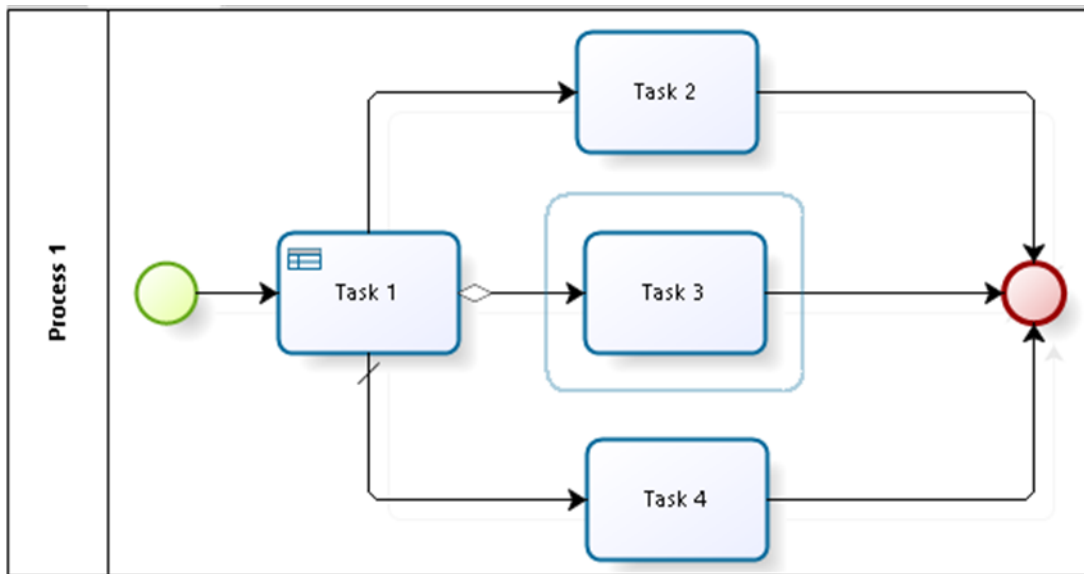


Joonis 6. Mittevälstava lahknemise muster [8][13][14][15]

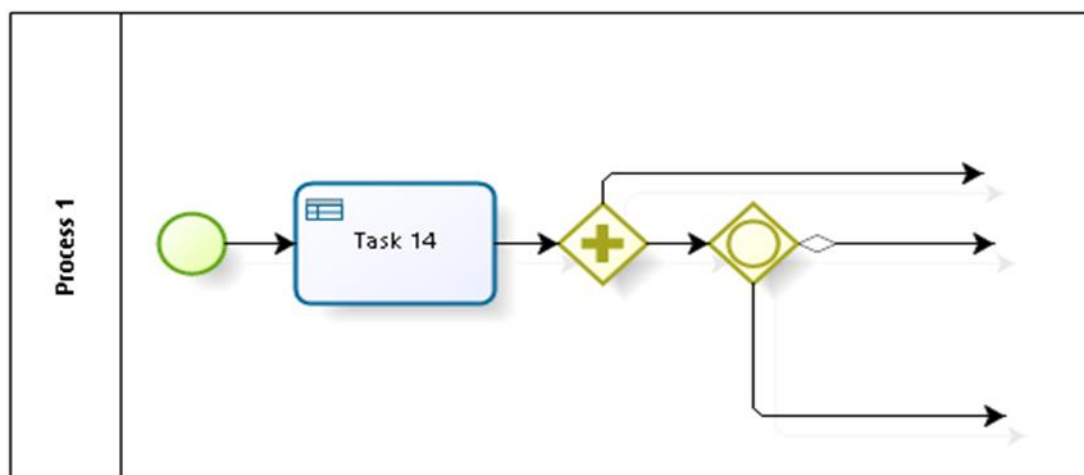
Juhul kui tegevusest või sündmusest suunduvad välja tingimuslikud järgnevusvood, tekib mudelis kaudne mittevälstav lahknemine. Esimene põhjus, miks sellist antimustrit peaks vältima, on võimalus, et suurema mudeli korral jääb kasutajal järgnevusvoogude tingimuslikkus märkamata. Kuigi antimustri puhul on järgnevusvoo elemendi välimus väga sarnane mustri välimusega, siis nende käitumine erineb oluliselt. Teine põhjus seisneb faktis, et osa modelleerimisvahendid ei luba sellist antimustrit kasutada. [8]

Bizagi Modeler käivitab antimustri simulatsioonimootoris, kuid parema loetavuse ja universaalsuse nimel peaks kasutama korrektset mustrit.

### 2.3 Mitmekordsed kaused lüüsid (*Multiple Implicit Gateways*)



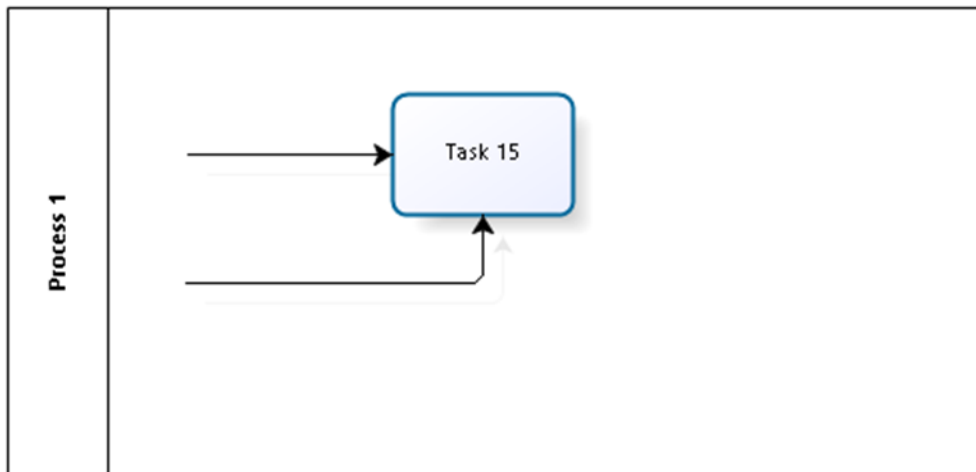
Joonis 7. Mitmekordsete kaudsete lüüside antimuster [8]



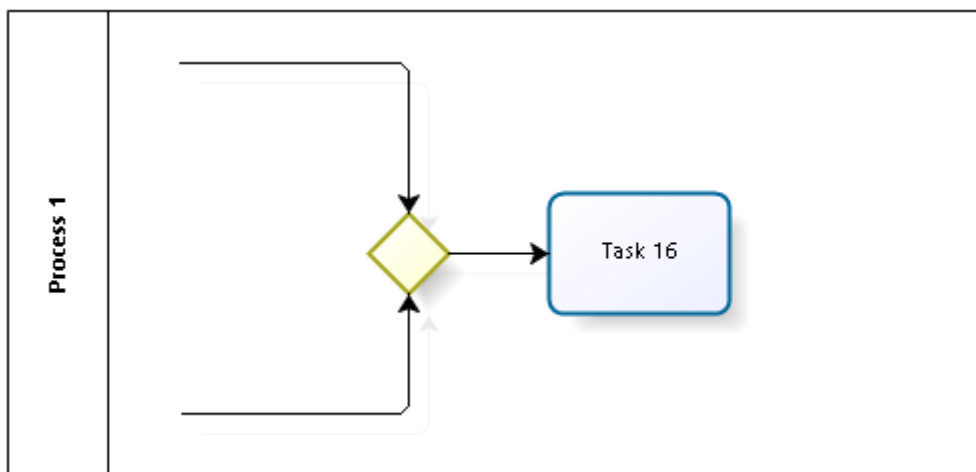
Joonis 8. Mitmekordsete lüüside muster [8]

Antimustri puhul on kujutatud ärireegli tegevust, kust väljuvad tingimuseta, tingimuslik ja vaikimisi valitud järgnevusvoog. Selline käitumine on sama, kui paralleelsele lüüsile järgneks mitteväljavõetav lüüs. Siinsel juhul ei ole antimuster paljudele kasutajatele kergelt loetav ega intuiitivne. Lisaks on hargnevuste tõenäosusi kergem määrata, kasutades korrektset mustrit. [8]

## 2.4 Kaudsed ühenduslüüsid (*Implicit Join Gateways*)



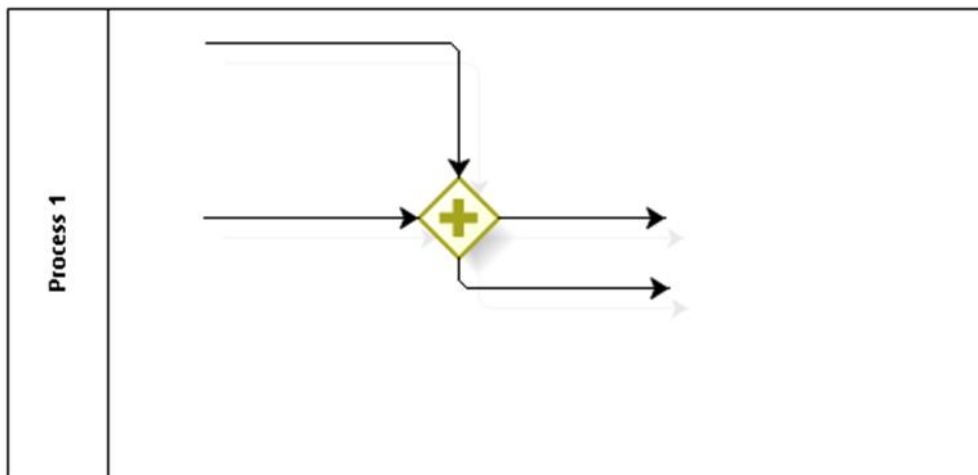
Joonis 9. Kaudse ühenduslüüsi antimuster [8]



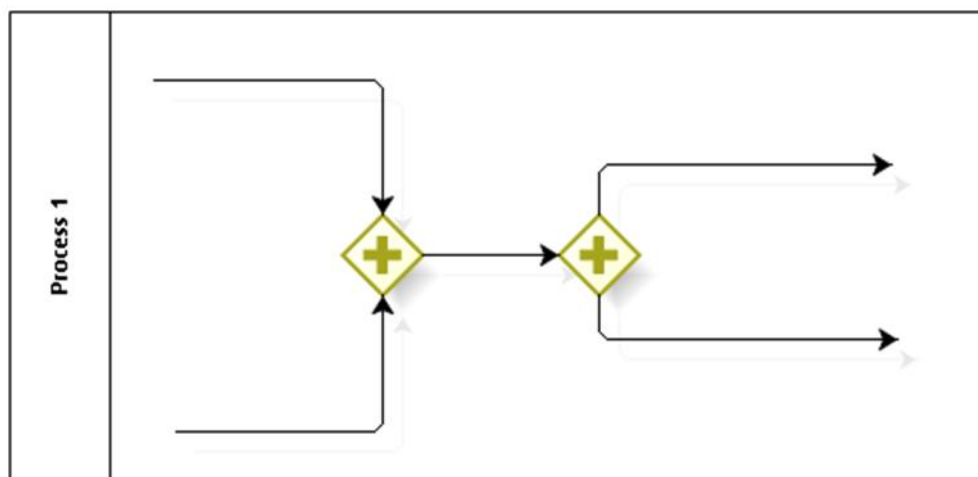
Joonis 10. Kaudse ühenduslüüsi muster [8][13][14][15]

Tegemist on kaudse lahknemise antimustri vastandiga. Kui mitut väljuvat voogu käsitatakse paralleelsete voogudena, siis mitut saabuvat voogu käsitatakse kui üksteist välistavaid vooge. BPMNi standard kirjeldab sellist mustrit kui „kontrollimatut voogu”. Tegevus käivitatakse iga kord, kui kumbki voog jõuab elemendini. Juhul kui tegemist ei ole tsükliga, siis enamasti eeldavad kasutajad, et tegevus käivitub protsessis ühe korra. Seetõttu peaks kasutama välistavat lüüsi enne voo tegevuseni/sündmuseni jõudmist. [8]

## 2.5 Kombineeritud ühendus ja lahknemislüüsid (*Combined Join-Split Gateway*)



Joonis 11. Kombineeritud ühendus- ja lahknemislüüsi antimuster [8]



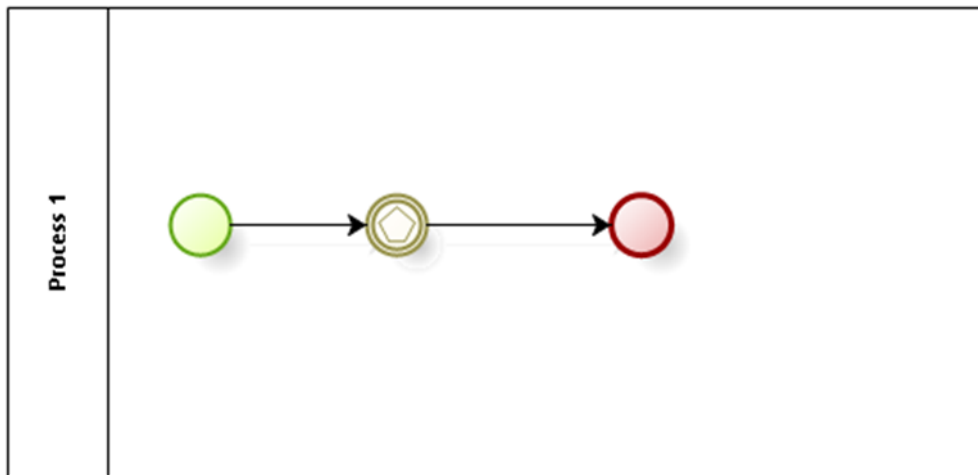
Joonis 12. Kombineeritud ühendus- ja lahknemislüüsi muster [8][13][14][15]

BPMN lubab ühte lüüsi mitut sisenevat ja väljuvat voogu. Siiski tuleks mudeli parema loetavuse ja kergema modifitseerimise nimel kasutada voogude ühendamiseks ja hargnemiseks kahte järjestikust paralleelset lüüsi. [8]

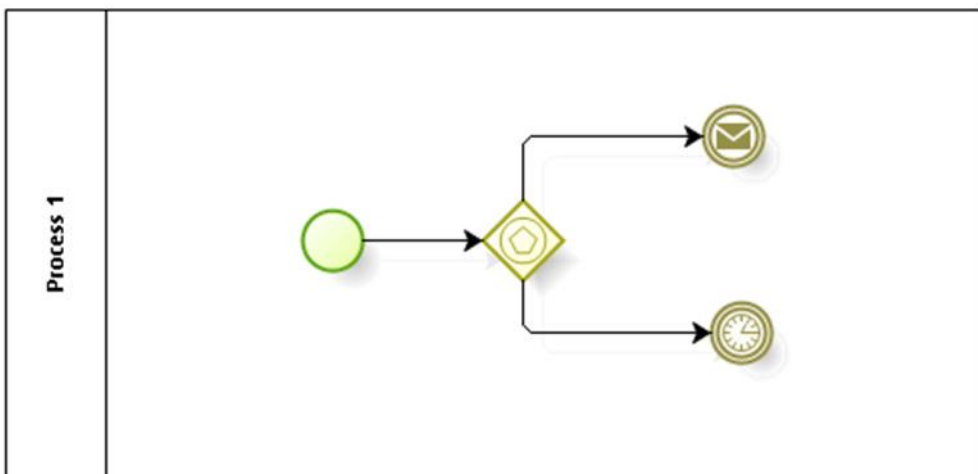
Seda antimustrit ei luba Bizagi Modeler kasutada.



## 2.6 Mitmekordne vastuvõttev sündmus (*Multiple Catching Event*)



Joonis 13. Mitmekordse vastuvõtva sündmuse antimuster [8]

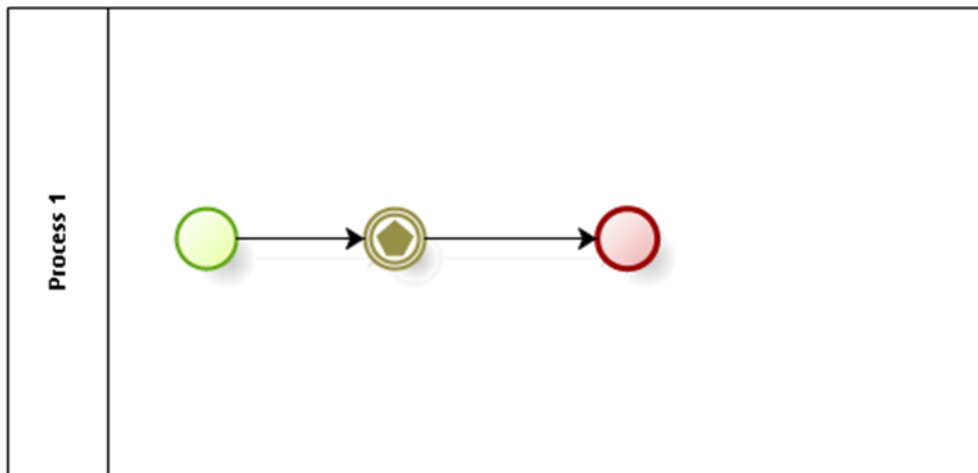


Joonis 14. Mitmekordse vastuvõtva sündmuse muster [8][14][15]

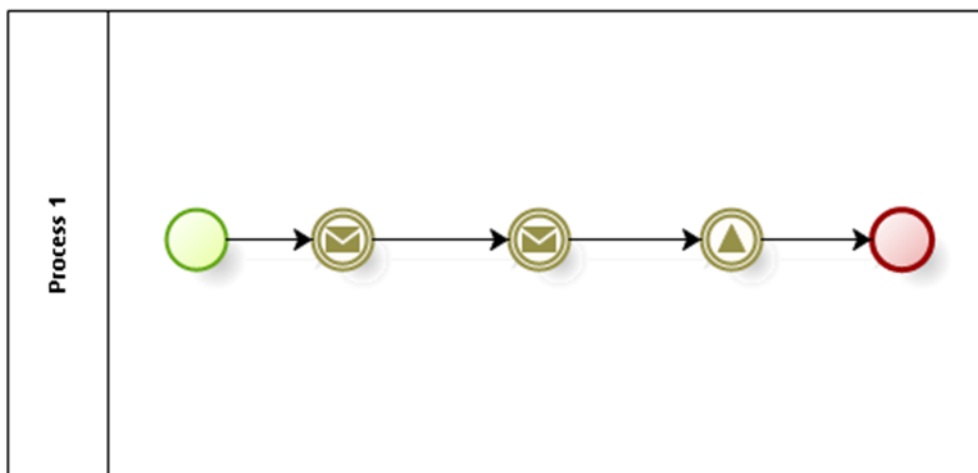
Juhul kui konfigureeritakse mitu saabuvat sündmust ühte vastuvõtvasse sündmusesse, siis käivitab esimesena toimunud sündmus vastuvõtva sündmuse. Selline käitumine on analoogne kahe järgnevusvoo saabumisega välistavasse lüüsi. Antimuster võib seega varjata lugeja jaoks olulist informatsiooni. Kuna sündmusepõhine lüüs lisati BPMNi täpselt selle eesmärgiga, siis peaks seda ka kasutama. See paljastab sündmused, mis protsessis toimuvad. [8]

Bizagi Modeler valideerib antimustri, kuid mudeli simuleerimine ei õnnestu.

## 2.7 Mitmekordne saatev sündmus (*Multiple Throw Event*)



Joonis 15. Mitmekordse väljastava sündmuse antimuster [8]

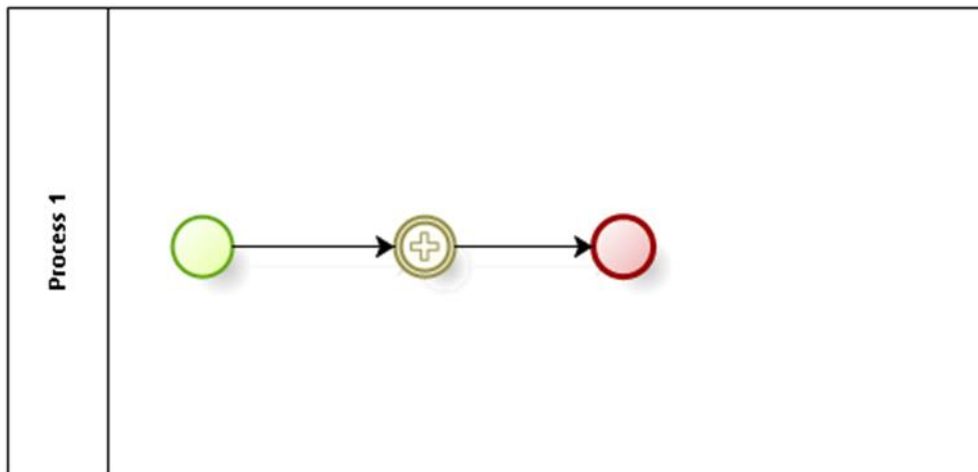


Joonis 16. Mitmekordse väljastava sündmuse muster [8]

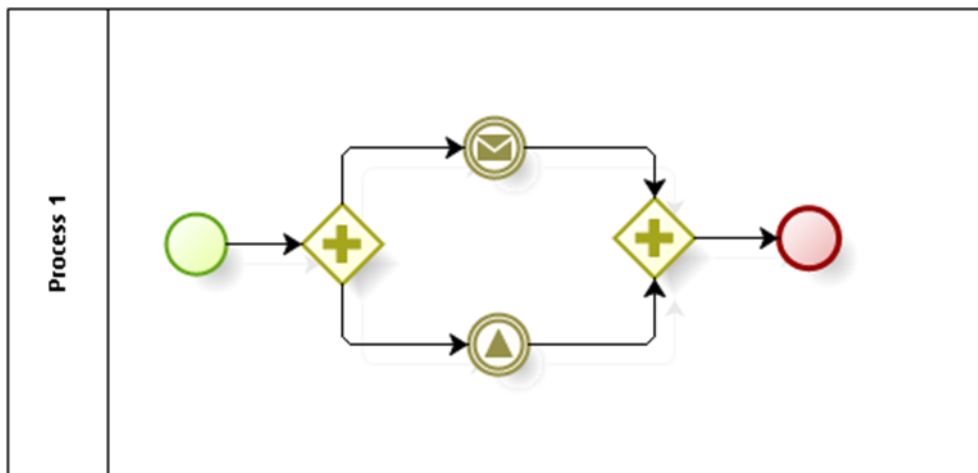
Nagu ka eelmise antimustri puhul, seisneb probleem peamiselt kasutaja eest olulise info varjamises. Erinevalt mitmekordsest püüdvast sündmusest käivitatakse kõik peidetud sündmused ning nende järjekord ei ole defineeritud. Kui protsessis tekib viga, ei saa olla kindel, milline sündmus selle põhjustas. Need elemendid on mõeldud ruumi säästmiseks suurtes mudelites, kuid samas peidavad olulist informatsiooni ega sobitu simulatsioonimootoriga. [8]

Bizagi Modeler valideerib antimustri, kuid mudeli simuleerimine ei õnnestu.

## 2.8 Mitmekordne paralleelne sündmus (*Multiple Parallel Event*)



Joonis 17. Mitmekordse paralleelse sündmuse antimuster [8]

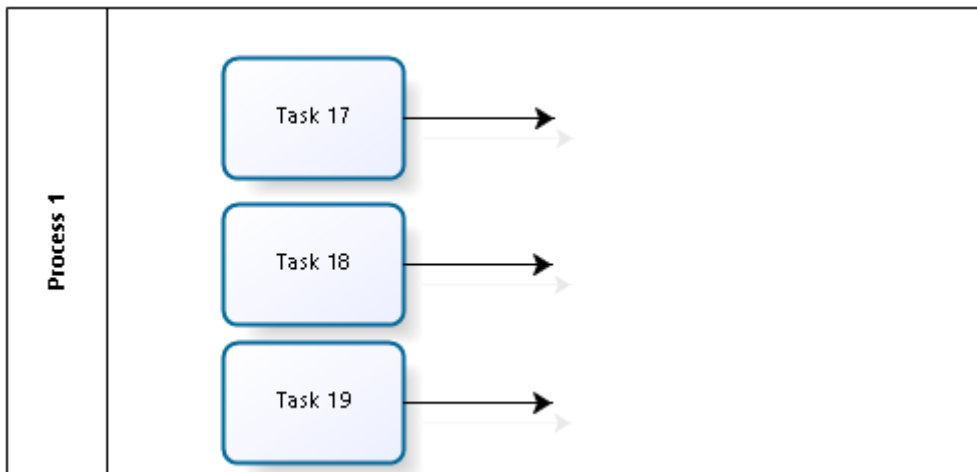


Joonis 18. Mitmekordse paralleelse sündmuse muster [8][14]

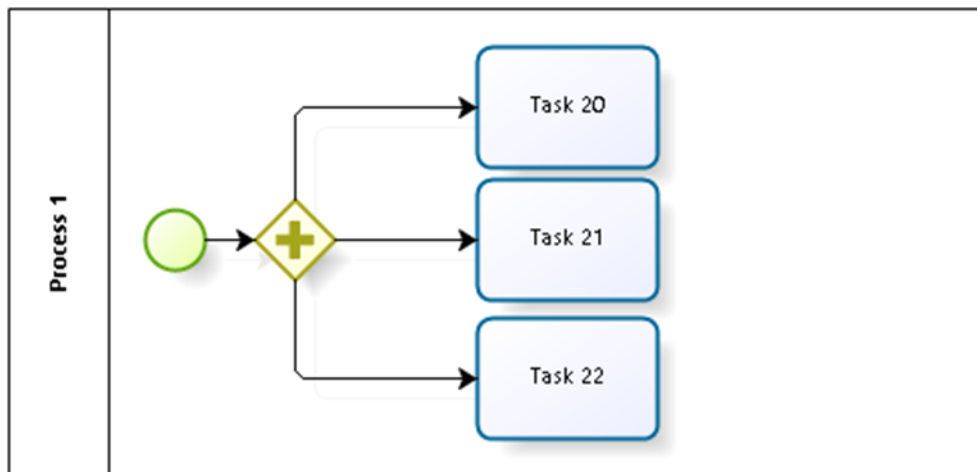
Antimustri puhul on tegemist samade probleemidega nagu eelmiste mitmikielementide puhul. Kuna tegemist on sisuliselt paralleelse mooduliga protsessis, tasub see informatiivsuse nimel alati pikemalt modelleerida. [8]

Bizagi Modeler valideerib antimustri, kuid mudeli simuleerimine ei õnnestu.

## 2.9 Puuduv algündmus (*No Start Event*)



Joonis 19. Puuduva algündmuse antimuster [8]

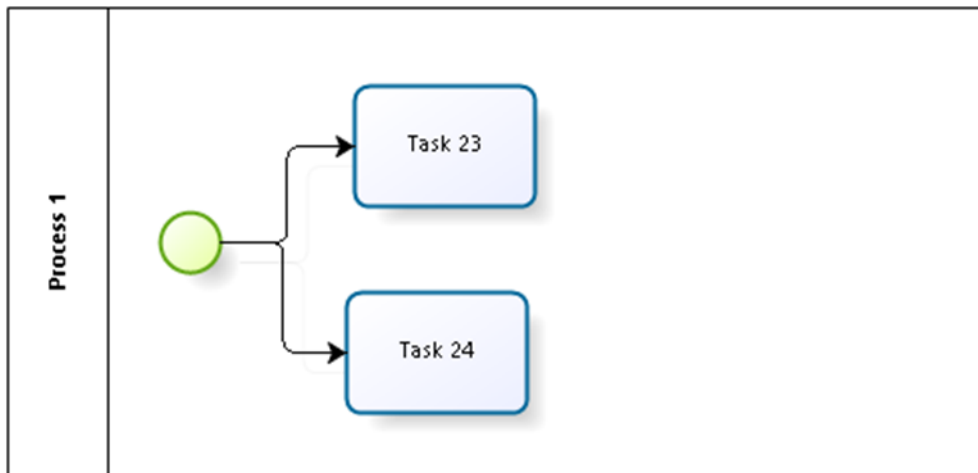


Joonis 20. Protsessi alguse muster [8][13][14][15]

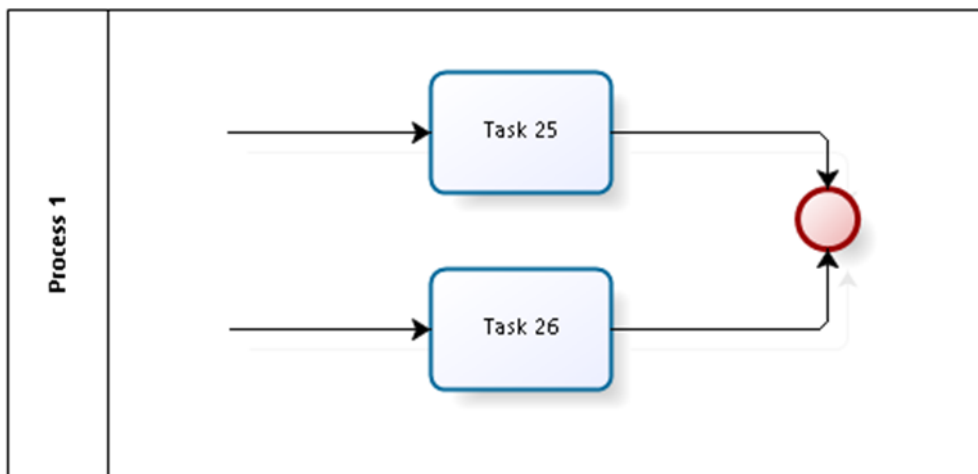
BPMNi standard lubab defineerida protsessi ka ilma algündmuse elementi kasutamata. Tegemist on üpriski halva modelleerimistehnikaga, sest mudelis käsitletakse kõiki leiduvaid tegevusi ja sündmusi võrdsetena ning käivitatakse need paralleelselt. Kui tegevusi on rohkem kui üks, võib mudeli lugeja sattuda segadusse ega pruugi aru saada järgnevate voogude toimimisest. [8]

Osa programmid ei luba sellist praktikat, kuid Bizagi Modeler ei ole üks nendest ja antimuster toimib selle puhul ka simulatsioonimootoris.

## 2.10 Puuduv lõppsündmus (*No End Event*)



Joonis 21. Puuduva lõppsündmuse antimuster [8]

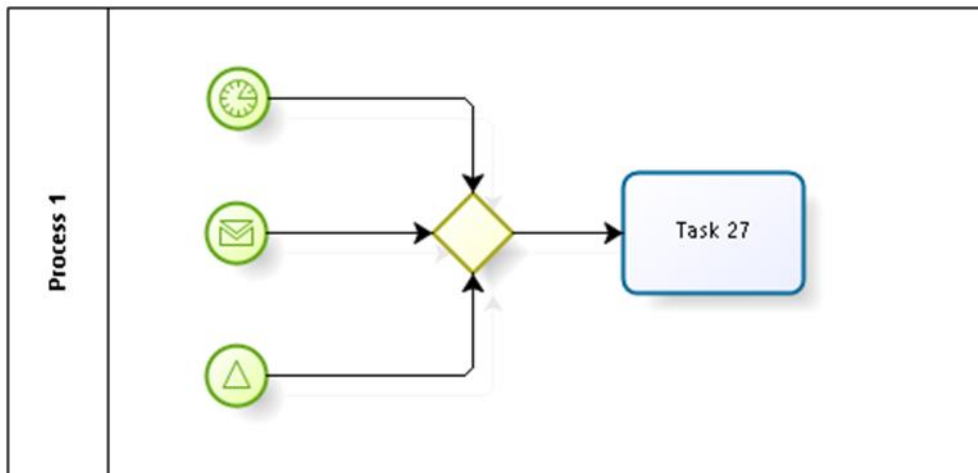


Joonis 22. Korrektn protsessi lõppsündmuse muster [8][13][14][15]

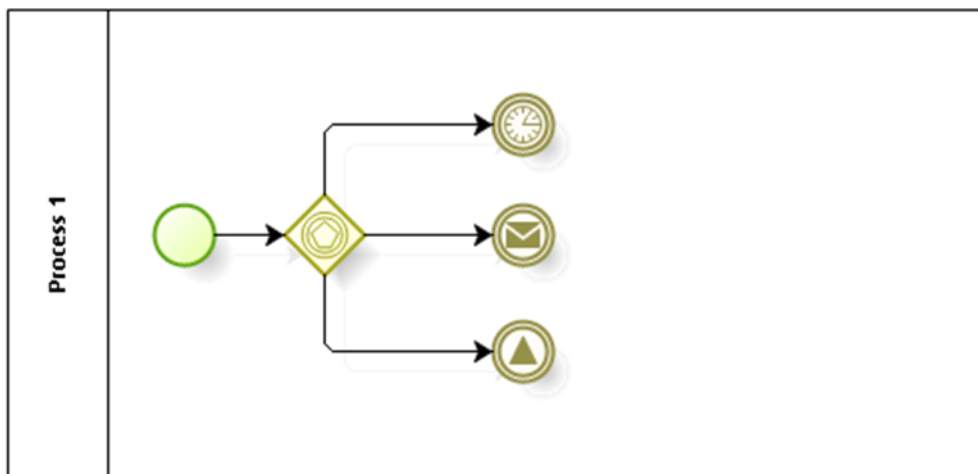
Probleemid selle antimustri puhul on analoogsed eelmisega. Juhul kui mudelis on mitu erinevat tegevust/sündmust, millele ei järgne lõppsündmust, muutub mudel raskesti loetavaks. Ühe erijuhuna võib selliseid antimustreid kasutada alamprotsessides, kus on ühe alg- ja lõppsündmusega tegevus. [8]

Bizagi Modeler valideerib antimustri ning mudel toimib ka simulatsioonimootoris.

## 2.11 Mitmekordsed algündmused (*Multiple Start Events*)



Joonis 23. Mitmekordsete algündmuste antimuster [8]

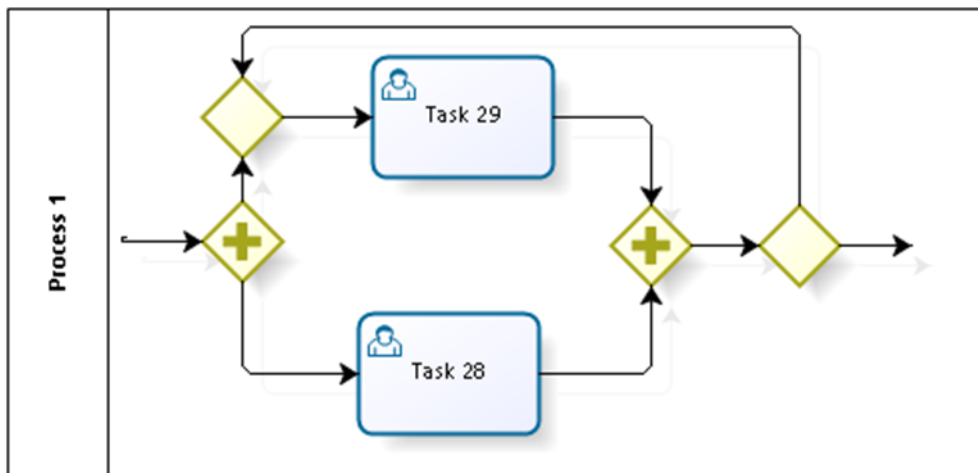


Joonis 24. Mitmekordsete algündmuste muster [8][13][14][15]

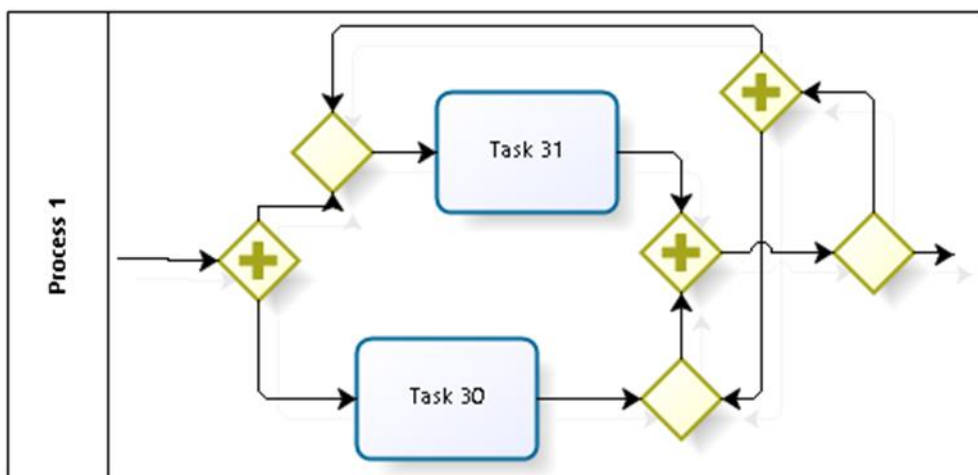
Antimustri probleem seisneb selles, et ükskõik milline algündmuse esinemine kutsub esile ka terve protsessi käivitumise. Kui järgnevalt käivitub mõni teine algündmus, siis alustab see täiesti uue protsessi juhtumi, mis ei oma loogilist järgnevust ega seost esimesena käivitatud töövooga. Tüüpiline näide oleks protsess, mis algaks tänu sissetulevale sõnumile, millel võivad olla eri saatjad. Olenemata saatjast käivitub protsess ning lisanduvate sõnumite saabumine ei muudaks töövoogu. [8]

Antimuster valideerub Bizagi Modeleris ja käivitub ka simulatsioonimootoris.

## 2.12 Paralleelsetesse moodulitesse sisenevad vood (*Flows into Parallel Blocks*)



Joonis 25. Paralleelsetesse moodulitesse sisenevate voogude antimuster [8]

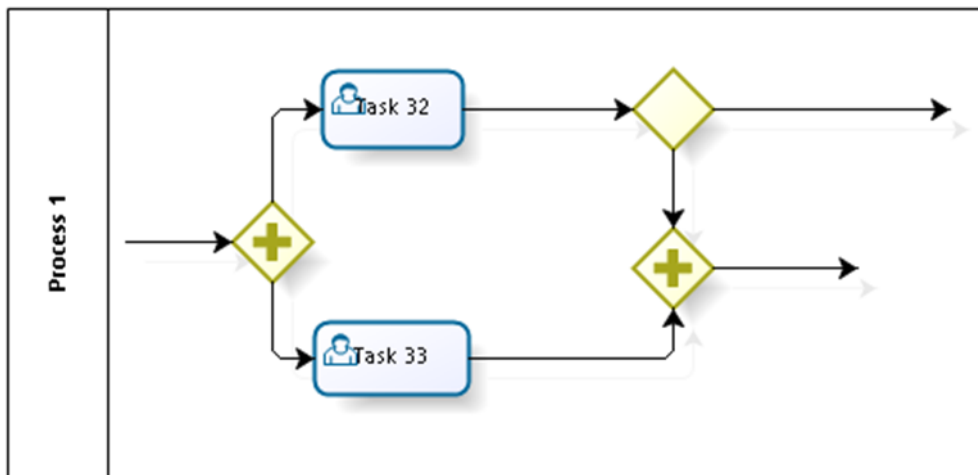


Joonis 26. Paralleelsetesse moodulitesse sisenevate voogude muster [8][13][14][15]

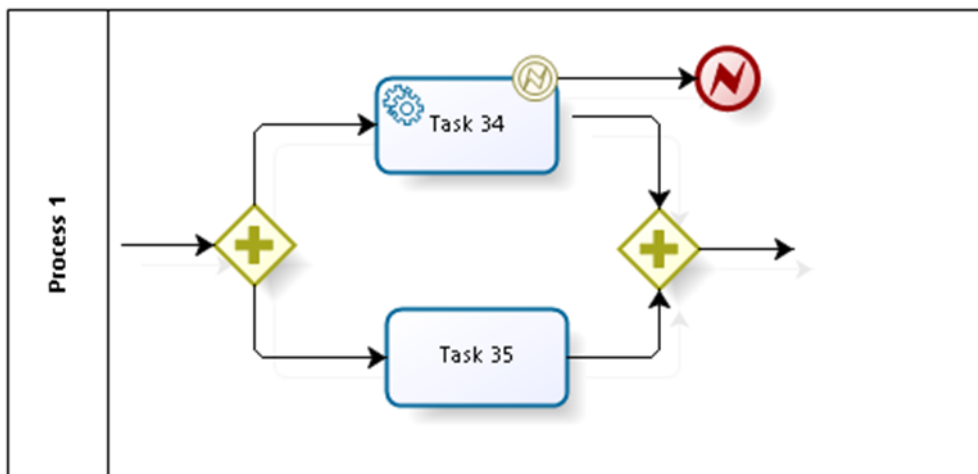
Tegemist on juhtumiga, kus antimuster tundub esmapilgul oluliselt loetavam ja kergemini mõistetavam kui muster. Paraku tekib oht, et pärast mooduli läbimist suundub voog välistavast lüüsisist tagasi moodulisse, tekitades niiviisi ummikseisu. Olukord tekib, sest voogude arv ei ole võrdne, ning laheneb, kui lisada kaks lüüsi. Niiviisi tekib järgnevusvool võimalus suunduda edasi alternatiivse tee kaudu ning kaob ummikseisu oht. [8]

Antimuster valideerub Bizagi Modeleris ja käivitub ka simulatsioonimootoris.

## 2.13 Paralleelsetest moodulitest väljuvad vood (*Flows out of Parallel Blocks*)



Joonis 27. Paralleelsetest moodulitest väljuvate voogude antimuster [8]



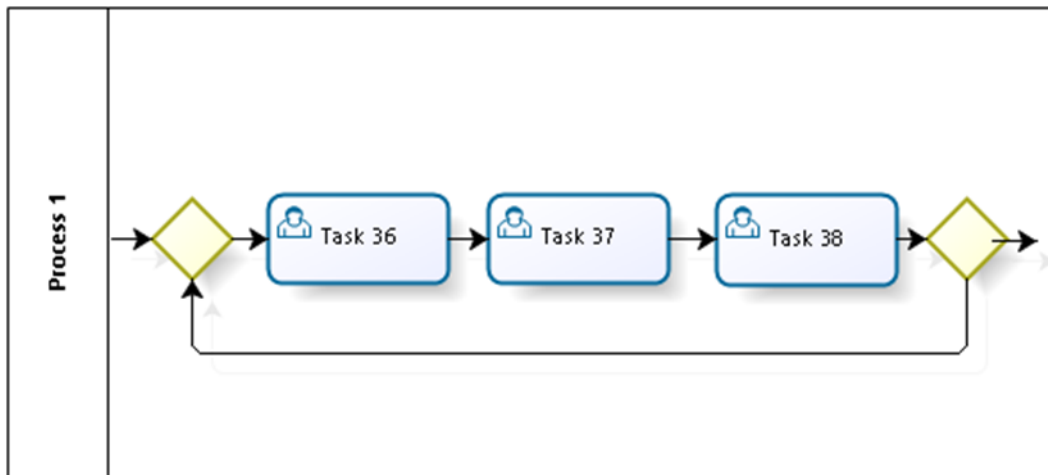
Joonis 28. Paralleelsetest moodulitest väljuvate voogude muster [8][13][14][15]

Kuna paralleelsete moodulite ühenduslüüsidesse peavad sisenema mõlemad protsessivood, siis võib antimustri korral tekkida oht, et tingimus ei ole rahuldatud ja tekib ummikseis. Mustri korral ei toimi küll see põhitõde, et paralleelsesse moodulisse peaks tulema ja väljuma üksainus voog, kuid vea korral katkestatakse korrektselt terve protsess ega teki surnud punkti. [8]

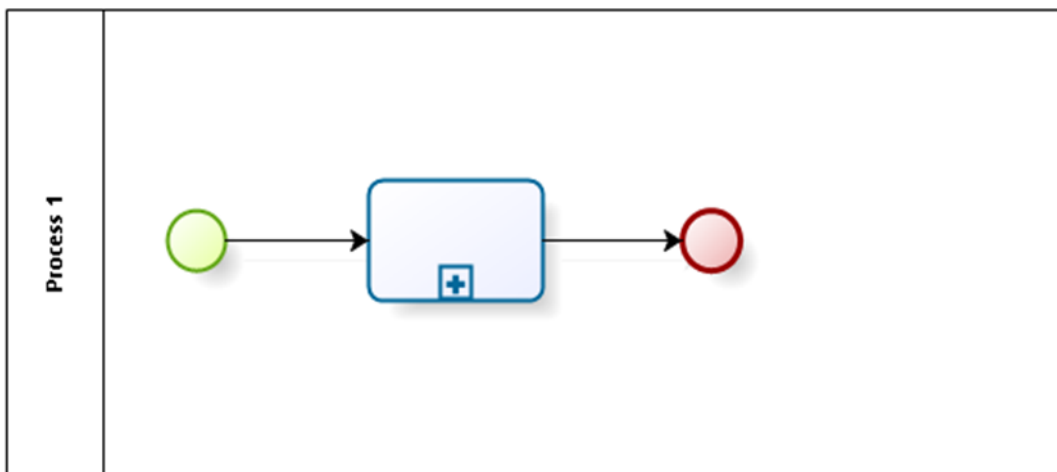
Bizagi Modeler valideerib antimustri ja käivitab selle ka simulatsioonimootoris.



## 2.14 Tsüklid (*Loops*)



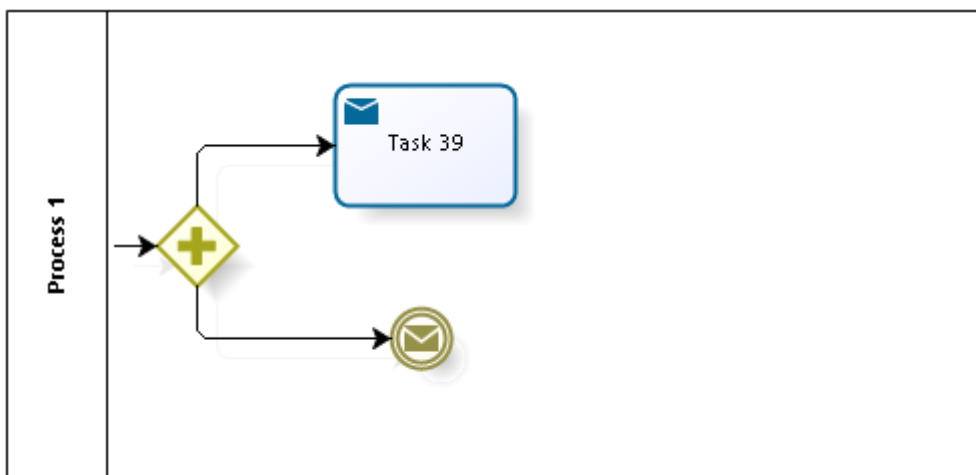
Joonis 29. Tsüklite antimuster [8]



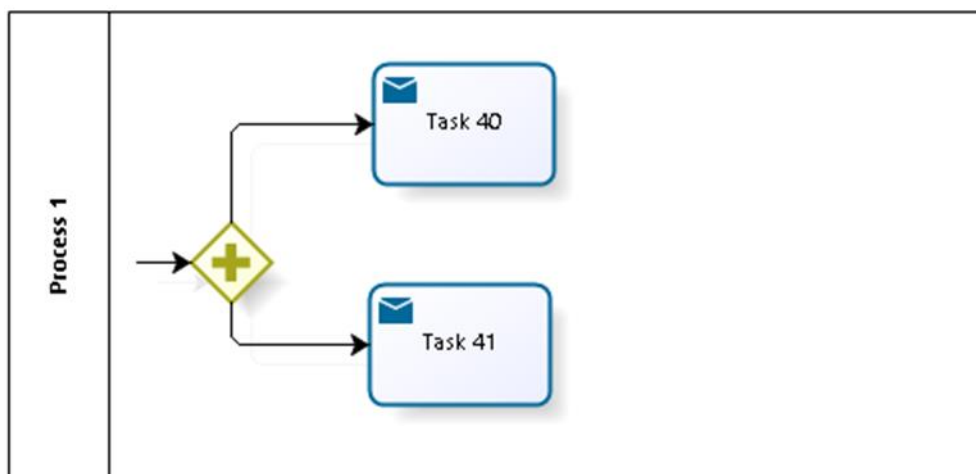
Joonis 30. Tsükli muster alamprotsessina [8][13][14][15]

Tsüklid on tihtipeale lahutamatud alamosad protsessimudelitest. Kahjuks kannavad nad endas ka riski tekitada lõpmatult kestvaid protsesse. Kuigi seda suudetakse üldiselt vältida, on siiski targem kirjeldada ühe sisend- ja väljundvooga tsükleid alamprotsessidena. Tsüklid muudavad mudelid raskemini loetavamaks ning alamprotsessi muster aitab seda vältida. [8]

## 2.15 Sõnumite ja saatvate tegevuste kombineerimine (*Mixing Message Events and Tasks*)



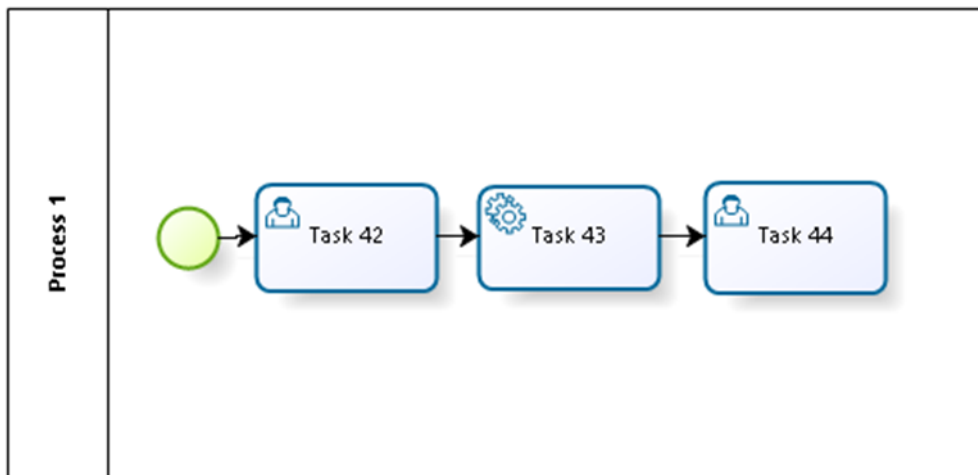
Joonis 31. Dubleeriva sõnumivahetuse antimuster [8]



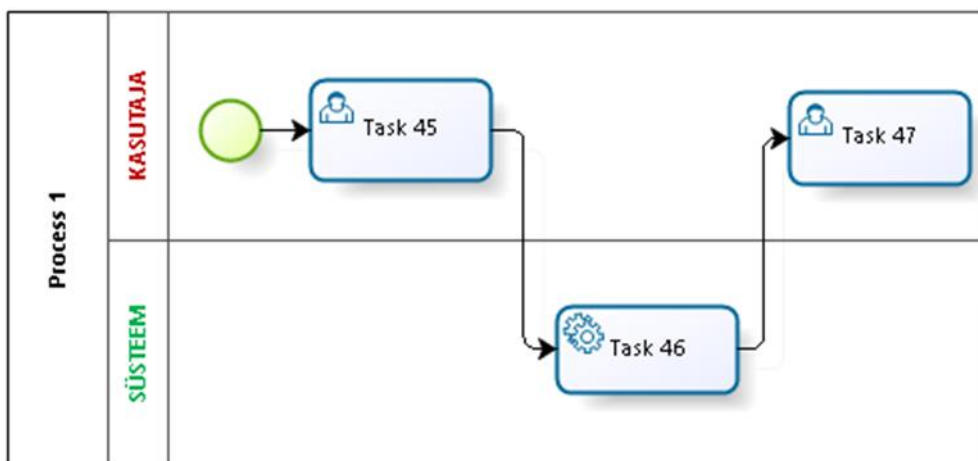
Joonis 32. Dubleeriva sõnumivahetuse muster [8]

Joonistel kujutatud elementide käitumine on sisuliselt sama. Seetõttu on segadusseajav, kui neid paralleelses moodulis koos kasutada. Kuna tegevuse elemendil on suurem kasutatavuse valdkond ning sellega saab ka veaolukordi kasutada, on soovitatav eelistada tegevuse elementi. [8]

## 2.16 Tegutseja rajal süsteemi tegevuse kujutamine (*Service Task in User Lane*)



Joonis 33. Integreeritud ujumisradade antimuster [8]

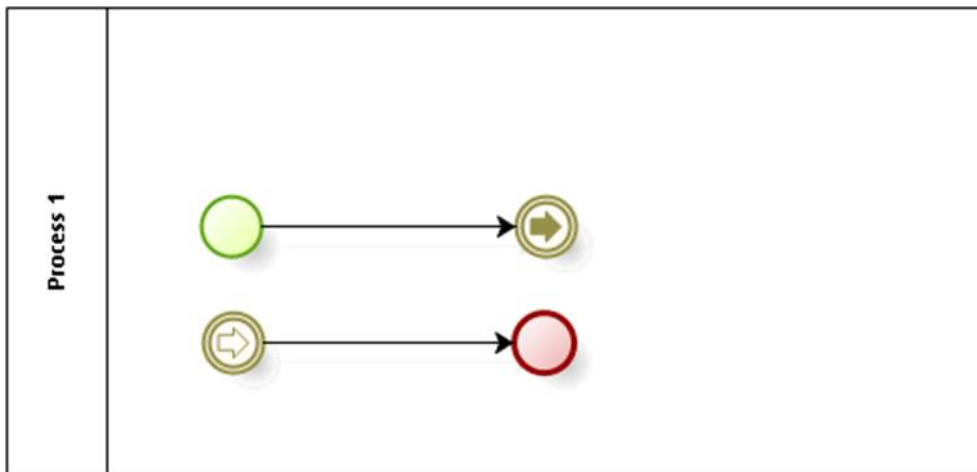


Joonis 34. Ujumisradade muster [8]

On halb tava näidata automatiseeritud tegevusi tegutseja rajal. Selline praktika võib mudeli kasutajat segadusse ajada ega anna edasi piisavalt infot juhul, kui süsteeme, mida tegutseja kasutab, on mitu [8]. Samas on väidetud, et süsteemi tegevuste elementide jaoks ei pea eraldi rada kasutama ning nende paigutamine suvalisele kohale skeemil ei oma mitte mingisugust mõju mudeli loetavusele [16].

Bizagi Modeler käsitab automatiseeritud tegevuste näitamist antimustrina [9].

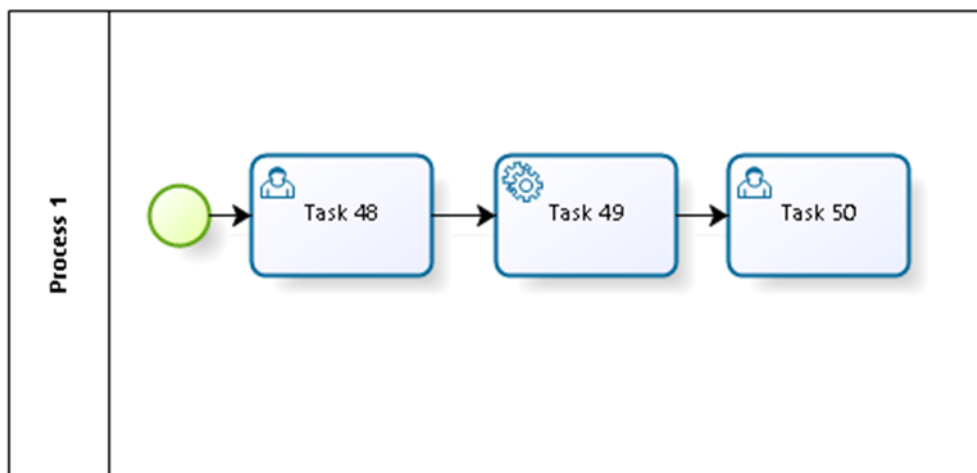
## 2.17 Siduvate sündmuste kasutamine (*Link events*)



Joonis 35. Siduvad sündmused kui antimustrid [8]

Siduv sündmus käitub järgnevusvoo elemendiga täiesti analoogselt. Seetõttu on selle kasutamine protsessi modelleerimisel aktsepteeritav ainult väga suurte diagrammide puhul, kus vooelemendi katkestus ei ole välditav. [8]

## 2.18 Ekraanivoogude kujutamine (*Screen Flows*)

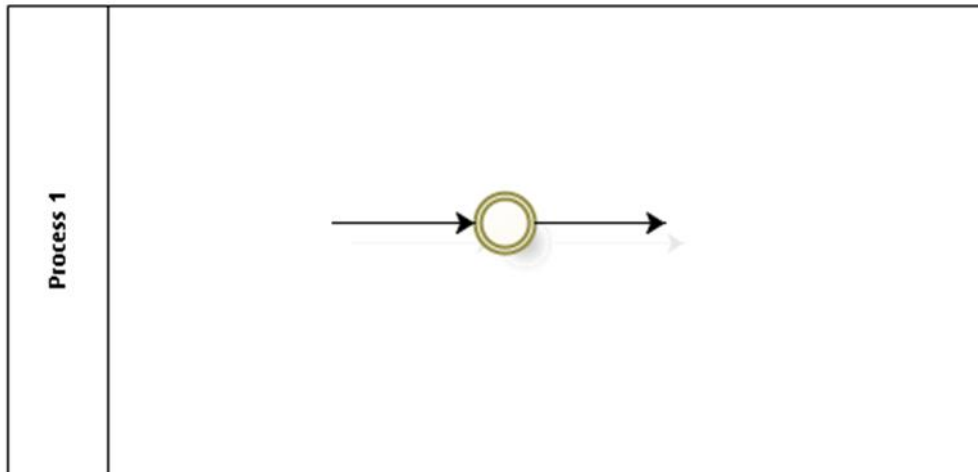


Joonis 36. Ekraanivoog kui protsess [8]

BPMN loodi eesmärgiga kujutada äriprotsessi toimimist ning tegevuselemendid peaksid kujutama mõõdetavaid tööprotsessi osi. Infosüsteemi kasutajaliidese toimingute tegevus on küll mõõdetav töö osa, kuid see on kujutatav ühe tegevusena. Lisaks toob selline ühe

tegevuse jaotamine mitmeks kaasa statistikat, mis võib simulatsioonis mõjutada tegelikke andmeid. [8]

## 2.19 Puuduv/tühi sündmus (*None Event*)



Joonis 37. Puuduv sündmus [8]

BPMN lubab defineerida sündmusi, millel pole definitsiooni. Selline sündmus ei lisa protsessi mingit informatiivsust ega väärtust. Osa modelleerijad kasutavad seda mustrit mingisuguse sündmuse esiletõstmiseks. Siiski saab selle praktika abil muuta simuleeritavaks protsesse, kus järgnevusvoog katkeks, ning see on üks põhjuseid, miks seda antimustrit kasutatakse. [8]

## 3 Mudelite koostamise nõuded

### 3.1 Testjuhitud tarkvaraarenduse meetodika

Sellel tarkvaraarenduse meetodikal on kaks lihtsat põhimõtet:

- Ära kirjuta uut koodirida, kui sul ei ole olemas sellele vastavat läbikukkunud automaattesti.
- Eemalda duplikatsioonid.

Protsess algab testi kirjutamisega, mis esialgu ei pruugi olla isegi kompileeruv. Seejärel keskendutakse koodi kirjutamisele nii, et see rahuldaks testi tingimusi. Viimaks teostatakse refaktooring ehk eemaldatakse koodist duplikatsioonid ja üleliigsed osad. Niiviisi toimub arendus funktsionaalsusest lähtuvalt, mistõttu keskendutakse ainult olulisele ehk sellele, mida programm tegema peab. [17]

### **3.1.1 Metoodika kohaldamine modelleerimisele**

Selle metoodika põhimõtteid ei ole võimalik täpselt sama sõnastusega üle kanda. Loomulikult on programmeerimine ja modelleerimine praktikas kaks täiesti erinevat tegevust, kuid kontseptuaalsel tasemel ei ole nende erinevus nii suur. Äriprotsessi puhul ei ole lõpptulemus enamasti defineeritav parameetritena, sest need tekivad simulatsiooni tulemusel, mistõttu puudub võimalus neid parameetreid testida. Selle asemel defineeritakse testina toimiv simulatsioon.

Protsessi moodulipõhine modelleerimine ei ole iseenesest uus tehnika. Tegemist on peaaegu mustripõhise modelleerimise sünonüümiga ning selle kohta tehtud uuringuid on kirjeldatud ka mitmes kasutatud allikas [2][5][7][14][18]. Iga muster saab olla moodul, kuid iga moodul ei saa olla muster. Autorile teadaolevalt ei ole aga rakendatud nende kahe metoodika kombineeritud versiooni.

## **3.2 Nõuded mudelitele**

Äriprotsesside valim on koostatud käesoleva magistr töö juhendaja – kes on ühtlasi ka kõnealuse õppeaine õppejõud – soovitusi järgides. Valituks osutusid järgmised protsessid:

- Telekommunikatsiooniettevõtte erakliendi äriprotsess.
- Tudengi õppeaine läbimise protsess. [19]
- Klienditeeninduse protsess kohvikus. [19]

Loetelu kaks viimast protsessi on ka praegu õppenäidetena kasutuses, kuid õppejõu hinnangul saaks neid mustripõhiselt arendada ning simulatsioonimootorile sobivaks disainida. Lisaks uuritakse abstraktse näitena protsessi, kus näiliselt katkev protsessivoog põhjustab simulatsiooni ebaõnnestumise.

Peamised protsessidele esitatavad nõuded, mis moodustavad ka protsesside ühisosa, on järgmised:

- Protsessid sisaldavad lisaks põhiprotsessile ka alamprotsessi.
- Alamprotsessides toimuvad katkestavad sündmused.
- Protsessidel on mitmeid hargnemisi.

Mudelite koostamisel on lähtunud järgmistest kriteeriumitest:

- **Modelleerimine peab olema mustripõhine ehk antimustrite rakendamine peab olema tugevalt põhjendatud.**
- **Modelleerimisel kasutatakse võimalusel testjuhitud metoodikat.**
- **Mudel peab töötama simulatsioonimootoris.**

## 4 Õppenäidete mudelite kirjeldused

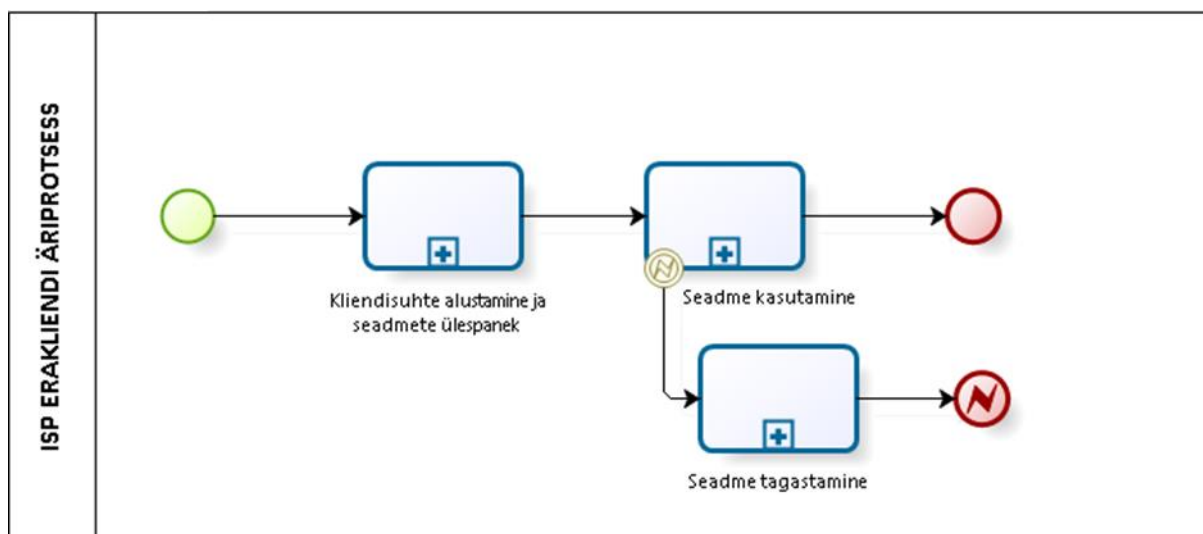
Järgnevalt on esitatud väljavalitud äriprotsesside modelleerimise tulemusena loodud mudelite kirjeldused. Siin toodud protsessid ei pruugi olla modelleeritud ja arvatavasti ei ole ka analoogsed reaalses elus toimuvate protsessidega, sest töö autoril puudub nendest täpne tehniline ülevaade. See tõsiasi ei mõjutanud töö eesmärke ega meetodikat.

Esmalt konstrueeriti kõik mudelid järgides testjuhitud modelleerimismetodikat, kus mudelid koostatakse esmalt simulatsioonimootoris kontrollitud moodulitest. Seejärel loodi mudelitest analoogne koopia järgides loogilist protsessivoogu ehk elemendid lisati mudelitesse vastavalt nende järjestusele protsessis. Hinnangu andmiseks analüüsiti ligikaudset ajakulu.

Täielikud mudelid on esitatud lisa 1. Elementidele määratud ressursikulu ning enamikke simulatsioonide tulemusi ei esitatud kahel põhjusel. Esiteks ei olnud eesmärgiks analüüsida protsesse, vaid viia mustritepõhised mudelid sellise kujuni, kus on võimalik neid simulatsioonimootoris käivitada. Teiseks ei annaks samade andmetega simulatsioonide taaskäivitamine analoogseid tulemusi, kuna sündmuste puhul on kasutatud juhuslikkust tekitavaid tõenäosusjaotuseid. Sellest lähtuvalt pole andmete esitamine mõistlik.



## 4.1 Telekommunikatsiooniettevõtte erakliendi äriprotsess



Joonis 38. Kontseptuaalne telekommunikatsiooniettevõtte erakliendi äriprotsess

Protsess kirjeldab, kuidas algab kliendisuhe telekommunikatsiooni teenusepakkujaga, kuidas toimub suhtlus, kui leping on sõlmitud, ning mil viisil on võimalik lepingu lõppedes seadmed tagastada. Tegemist on levinud protsessiga, mis ei ole väga unikaalne ja sobib õppenäiteks.

See protsess osutus valituks mitme põhjuse tõttu. Esiteks saab seda protsessi modelleerida kolmel erineval viisil. Tegutsejad on võimalik tuua välja eraldi basseinides või näidata neid erinevate rollidena ühises basseinis. Lisaks on võimalik basseinide kasutamisest loobuda. Kuigi teoorias ei tohiks muutuda protsessi käitumine, ei kinnita praktika seda fakti.

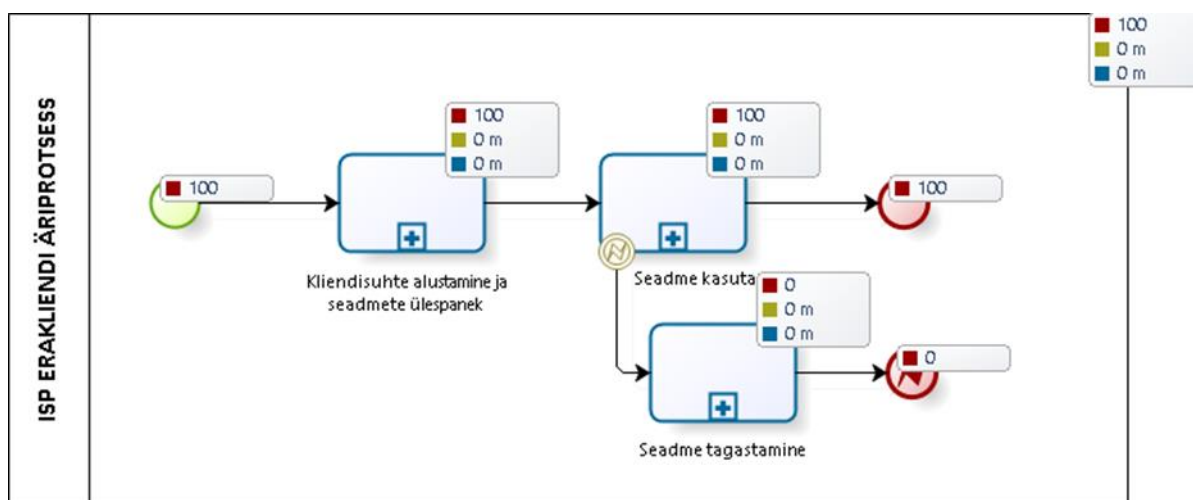
Teiseks oli tegemist sobiva protsessiga selle keerukuse tõttu. Kui põhiprotsessi kirjeldus tundub väga lihtne, siis lähemal vaatlusel selgub, et see sisaldab endas kolme alamprotsessi, millest üks aktiveerub ainult veasituatsioonis. Tegutsejaid oli olenevalt modelleerimisviisist 3–4.

Lisaks eespool mainitud põhjustele valiti see protsess õppetöö käigus tekkinud küsimuse tõttu. Probleem seisnes protsessi käigus kasutatavate seadmete hõivatuse leidmises. Seda karakteristikut uuriti ressursside simuleerimise teel ning eesmärk oli hinnata, kas saadud tulemused võiksid olla adekvaatsed.

### 4.1.1 Ühise basseini mudel

Täielik mudel on esitatud lisas 1.

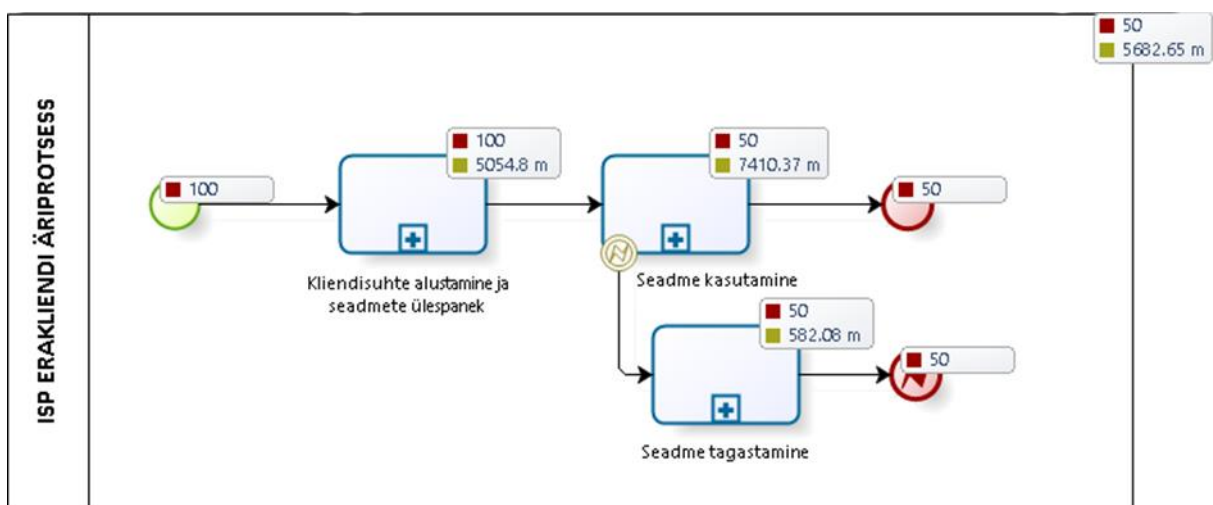
Mudel koosneb kolmest alamprotsessist: „Kliendisuhte alustamine ja seadmete ülespanek“, „Seadme kasutamine“ ja „Seadme tagastamine“. Muustrite rakendamisel tekkis probleeme peamiselt alamprotsesside ühildamatusest simulatsioonimootoriga. Alamprotsesse saab liigitada kaheks: taaskasutatav (*Reusable*) ja ühekordne ehk protsessisisene (*Embedded*). Taaskasutatavate alamprotsesside puhul ei simuleerita selles leiduvaid elemente ning ainukene viis, kuidas simulatsioon toimima hakkab, on manuaalselt aja konfigureerimine alamprotsessi elemendile. Järgnevalt on joonisel 39 kujutatud mudeli simulatsioon, kus on modelleerimisel kasutatud *Reusable*-tüüpi alamprotsessi elemente.



Joonis 39. Reusable-tüüpi alamprotsessidega simulatsioon

Simulatsioonimootor käivitab järjestikused alamprotsessid, kuid kuna nende elementide sees peituvaid protsesse ei käivitata, ei saa tekkida veaolukorda. Seepärast ei käivitu viimane alamprotsess „Seadme tagastamine“ ning tulemuseks on täiesti kasutatud simulatsiooniandmed. Alati on võimalus käivitada simulatsioon kahes esimeses alamprotsessis ning seejärel vastavalt veaolukordade tekkele ka kolmandas alamprotsessis. Saadud tulemused annaksid alamprotsessi simulatsiooniandmed, kuid terviklikku protsessi ei ole võimalik nende puhul analüüsida, sest tulemusi lihtsalt kokku liita ei saa.

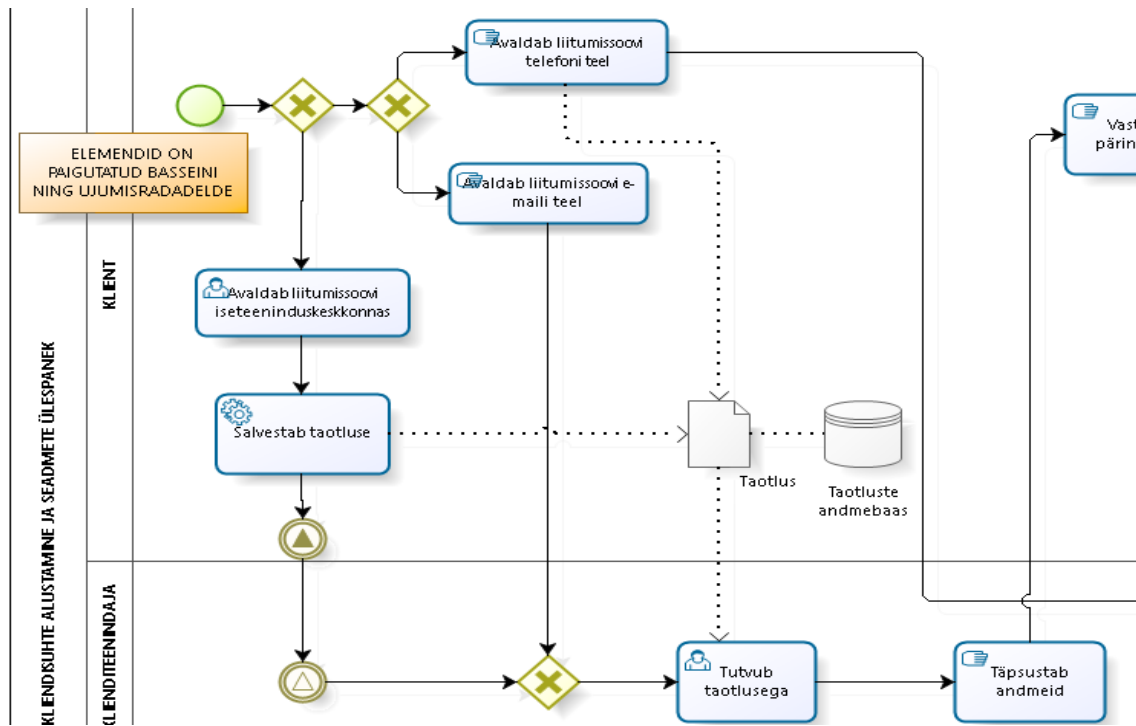
Sellise ühildamatuse tõttu on kasutaja valiku ees: kas loobuda diagrammide visuaalsest täpsusest või kasutada ühekordset alamprotsessi. Viimane ei luba kasutada basseine ega ujumisradu, mistõttu kaob ülevaade tegutsejatest. Basseinid ja ujumisrajad käituvad piltlikult öeldes kommentaaridena, mistõttu on lahendus sellisele probleemile üllatavalt lihtne. Kasutada tuleks protsessisiseseid (*Embedded*) alamprotsessi elemente ning protsessis osalejate tegevused tuleks grupeerida. Selleks tuleb kasutada elementi *Group*. Põhiprotsessi tasemel muutuvad elemendid visuaalselt vähesel määral, muudatus on tehniline. *Embedded*-tüüpi alamprotsessi elemente sisaldav mudel on kujutatud joonisel 40.



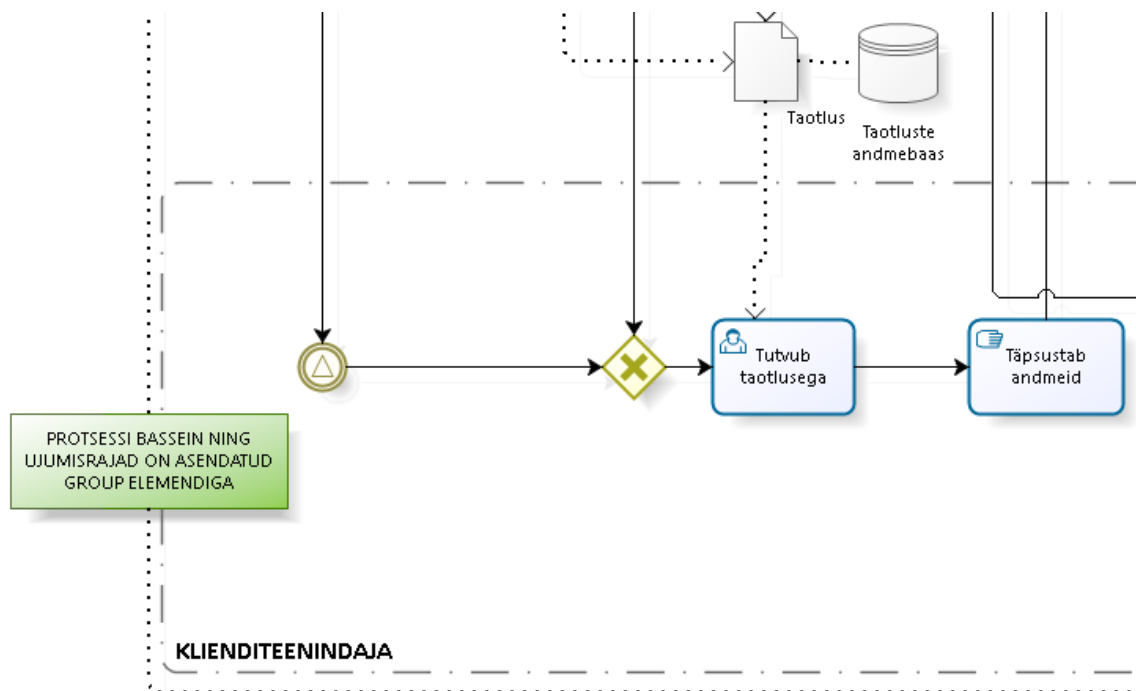
Joonis 40. Embedded-tüüpi alamprotsessi elementide simulatsioon

Jooniselt on näha, et protsessi simulatsioon töötab veatult ning järgnevusvoog jõuab ka veaolukorras kolmandasse alamprotsessi.

Alamprotsesside tasemel näevad diagrammid välja teistsugused, kuid simulatsiooni tulemust see ei mõjuta, sest ressursid on määratud kindlaks tegevuste juures. Järgnevalt on joonistel 41 ja 42 välja toodud *Reusable*- ning *Embedded*-tüüpi alamprotsessi erinevused.

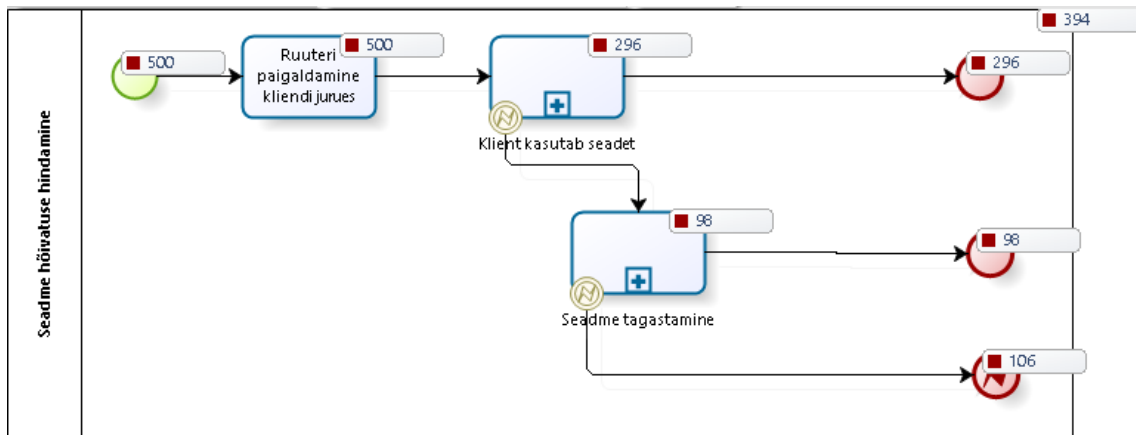


Joonis 41. Reusable-tüüpi alamprotsess

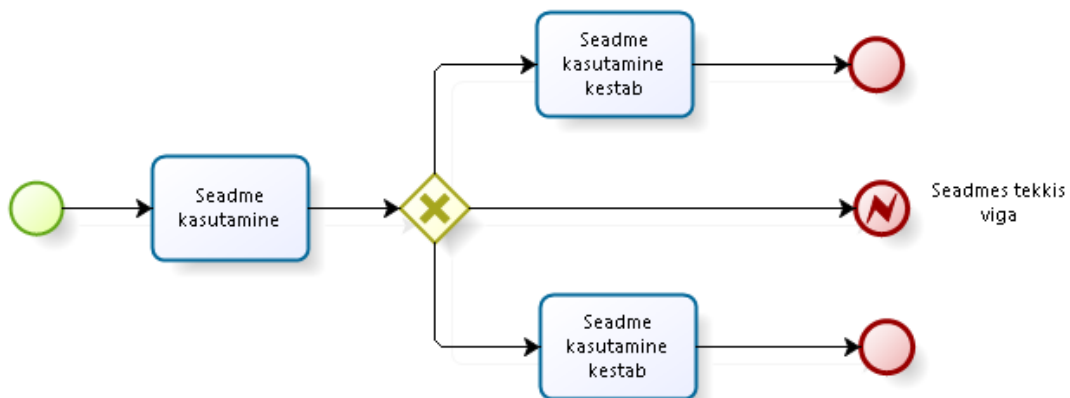


Joonis 42. Embedded-tüüpi alamprotsess

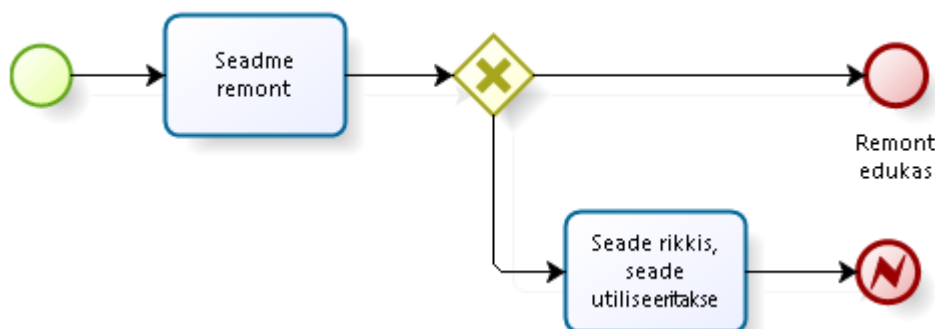
Seadme hõivatuse hindamiseks kasutati lihtsustatud mudelit, mis on kirjeldatud joonistel 43-45.



Joonis 43. Seadme hõivatuse hindamine



Joonis 44. Seadme kasutamise alamprotsess



Joonis 45. Seadme tagastamise alamprotsess

Seadme hõivatuse hindamisel tuleb teha vahet protsessi käigus oleva ressursi kogusel ja selle poolt tehtava töö mahul. Praegusel juhul mõistetakse hõivatuse all kui suur on seadmete kogus erinevate protsessi stsenaariumite korral, et ettevõtte saaks hinnata teenuse pakkumiseks vajaminevat kogust. Kõikidele tegevustele on määratud tegutsejaks „ruuter“ ehk siis seade, mille hõivatust soovitakse hinnata. Jooniselt 43 on näha, et

protsessi käivitatakse 500 korda, millest 296 jõuab soovitud lõppsündmuseni, kus klient jätkab seadme kasutamist. Järgnevas alamprotsessis õnnestub 98 juhul seade parandada ning kuigi seda ei kasutata enam protsessis edasi on lõpp ettevõtte jaoks positiivne, sest selle saab ringlusesse tagasi suunata. 106 juhul ei ole aga seadme remont edukas ning see suunatakse utiliseerimisele. Mustri puhul on oluline, et katkestav veasündmus töötaks ning kõik tõenäosusjaotused oleksid võimalikult täpsed. Võttes arvesse fakti, et telekommunikatsiooniettevõtetal on väga täpsed andmed seadmete rikete kohta, võib öelda, et seadme hõivatuse arvutamine sellisel kujul annab adekvaatsed andmed.

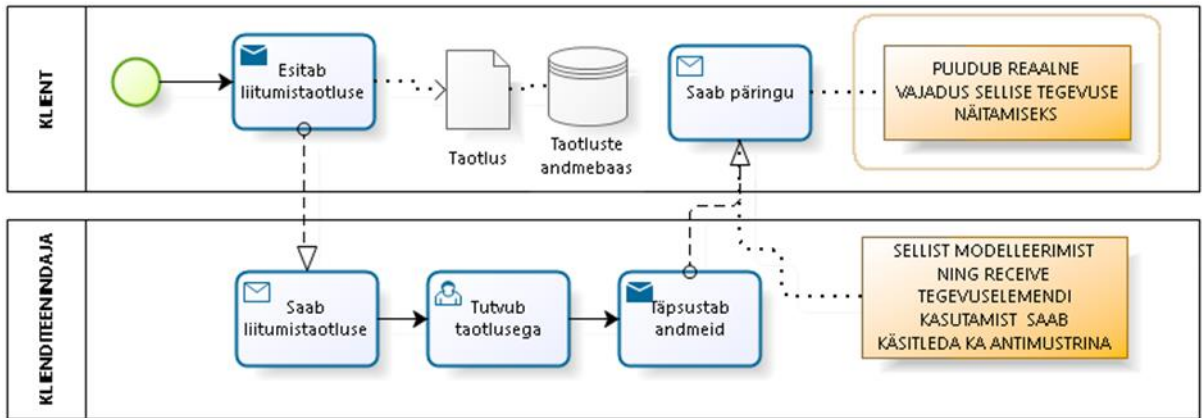
#### **4.1.2 Eraldi basseinidega mudel**

Täielik mudel on esitatud lisas 1.

Selle näite puhul seati peamiseks eesmärgiks uurida, kas eraldi basseinide kasutamine mõjutab simulatsiooniandmeid. Esialgse eesmärgi – kontrollida mustri optimaalsust sellele protsessile – muutis mitteoluliseks eelmises mudelis kasutatud muster. Järgnevalt kirjeldatav mudel ei sobiks nii suure äriprotsessi kaardistamiseks, sest selle lugemine ja töötlemine on laialivalgumise tõttu oluliselt raskendatud.

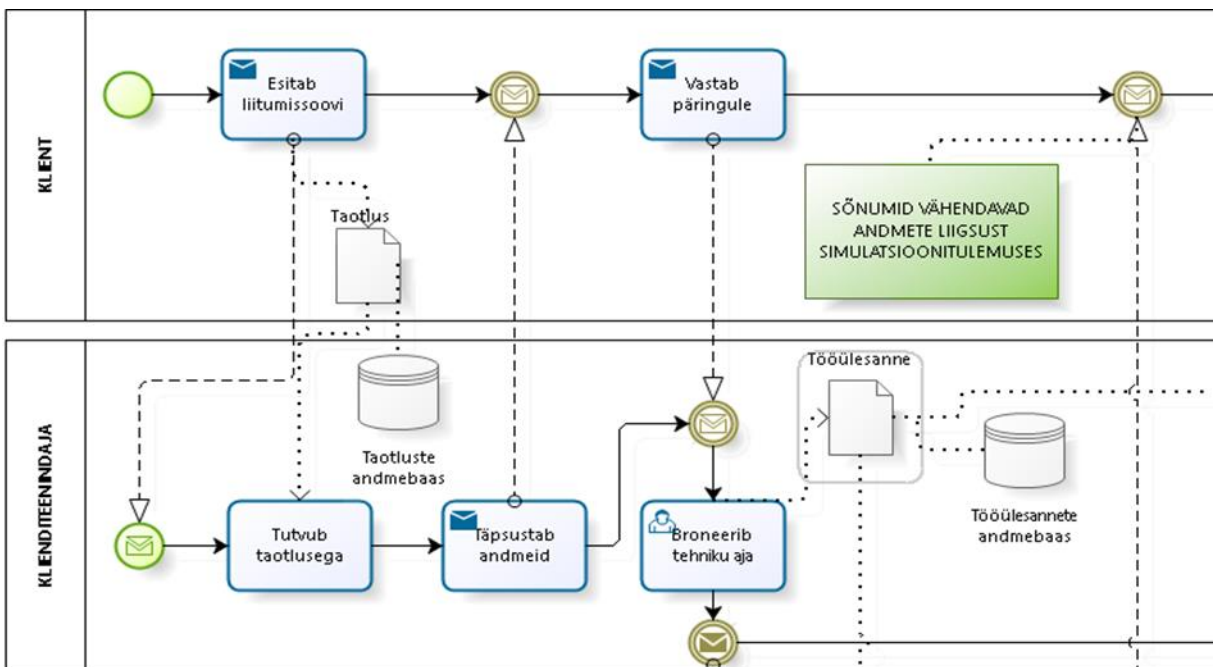
Kuna simulatsioonimootor ei suuda käivitada *Reusable*-tüüpi alamprotsesse, ei saaks kasutada eraldi basseinidega modelleerimise mustrit, sest *Embedded*-tüüpi elemendid ei võimalda basseine kasutada. Seetõttu ei olnud ka mõtet protsessi täielikult modelleerida ning simulatsioonide võrdlemiseks kasutati „Kliendisuhete alustamine ja seadmete ülespaneku“ alamprotsessi.

Modelleerimisel kerkisid esile probleemid, mis tulenesid BPMNi basseinidevahelise suhtluse spetsiifikast. Nimelt on selline suhtlus võimalik, kasutades kas *Send-/Receive*-sündmuse või samaväärse tegevuse elemente. Puudub võimalus kasutada järgnevusvoogu, mistõttu sisaldab mudel arvestatavas koguses elemente, mille ainus eesmärk on tagada vastavus BPMNi standardile. Ilma nendeta mudel ei valideeruks ning simulatsioon ei töötaks. Samas on oluline hoida selliste elementide hulk võimalikult minimaalsena, et vähendada infomüra ja kunstlike andmete tekkimist simulatsioonitulemusel. Järgnevalt on kirjeldatud elementide liigsust joonisel 46.



Joonis 46. Mitme basseini mustri elementide liigsus

Simulatsioonimootor eeldab, et ka *Receive*-tegevustel on määratud tegutseja ning ajakulu. Kui tegutseja tuleb määrata, siis ajakuluks saab panna 0. Tulemuses kuvatakse, et tegevus on läbitud, kuid ressursikulu sellega ei ole kaasnenud. Seega on tegemist liigsete andmetega, mis ei aita kaasa protsessi analüüsile. Liigsuse vähendamiseks saab alles jätta *Send*-tegevused ning asendada *Receive*-tegevused samaväärsete sündmustega. Sündmustele ei saa määrata ressursse, mistõttu väheneb infomüra. Joonisel 47 on kujutatud *Receive*-tegevuste asendamine sündmustega.



Joonis 47. Liigsuse vähendamine mitme basseiniga mudelis

Elementide rohkest saaks veelgi vähendada, muutes tegevuste tüüpi. Näiteks saaks muuta klienditeenindaja tegevuse „Broneerib tehniku aja“ *Send*-tegevuseks. Selline valik

tõstataks aga küsimuse, kas mudeli lihtsuse nimel tuleks loobuda detailsusest. Autori hinnangul vähendab selline tehnika mudelis kujutatud protsessi mõistmist, sest kaob arusaam, millisel viisil tegevusi läbi viiakse.

Alamprotsessi simulatsiooni tulemused seda mustrit kasutades ei muutu. Seadme hõivatust saab simuleerida samal moel nagu ühe basseini mudelis. Samas ei tasu mitme basseini mustrit kasutada mahukate protsesside modelleerimiseks, sest see ei võimalda põhiprotsessi kompaktsena konstrueerida.

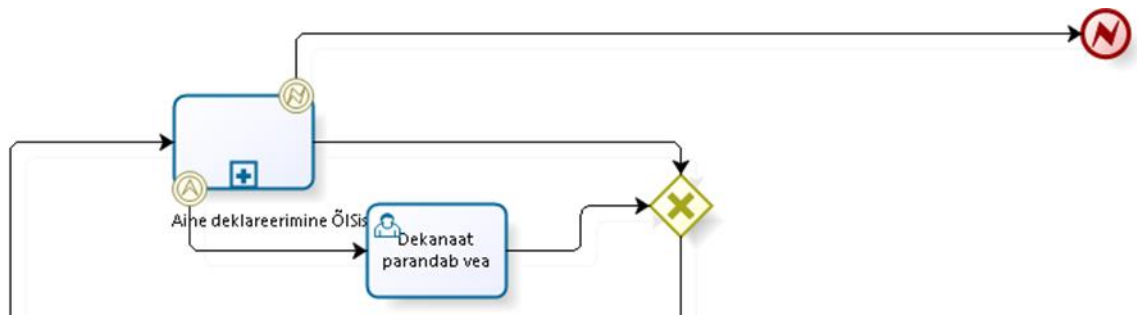
## **4.2 Õppeaine läbimise protsess**

Täielik mudel on esitatud lisas 1.

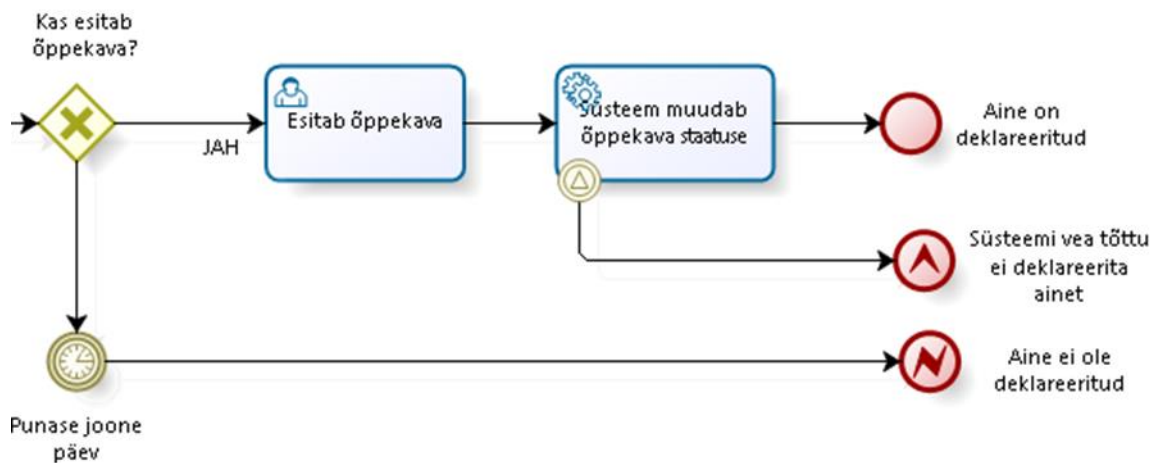
Selle protsessi puhul tekitab raskusi põhiprotsessi keerukus ning eri tüüpi alamprotsesside elementide kasutamine. Alamprotsessidena modelleeriti õppeaine deklareerimine, loengud, praktikumid, kontrolltöö ja koondhinde määramine. Esmapilgul võib selline lähenemine tunduda liialt detailsena, kuid põhjus seisneb ajamahukate tegevuste väljatoomise vajalikkuses. Näiteks esineb „Koondhinde määramise“ alamprotsessis tegevus „Parandab töö“, mis on õppejõu ressursikulu seisukohast üsna mahukas.

Tegemist on mudeliga, kus on võimalik rakendada paralleelsetest moodulitest väljuva voo mustrit ja seda lausa mitmel korral. Esimene probleem tekkis alamprotsessi „Aine deklareerimine ÕISIS“ käivitamisel. Protsessis saab tekkida kaks vea olukorda. Esimene neist tekiks siis, kui tudeng unustab punase joone päevaks esitada õppekava. Teise juhtumi korral peaks ÕISIS tekkima viga ning õppekava staatust ei muudeta, mistõttu aine jääb deklareerimata. Järgnevalt on joonistel 48–50 kirjeldatud, kuidas olid need kaks olukorda modelleeritud esialgses versioonis.

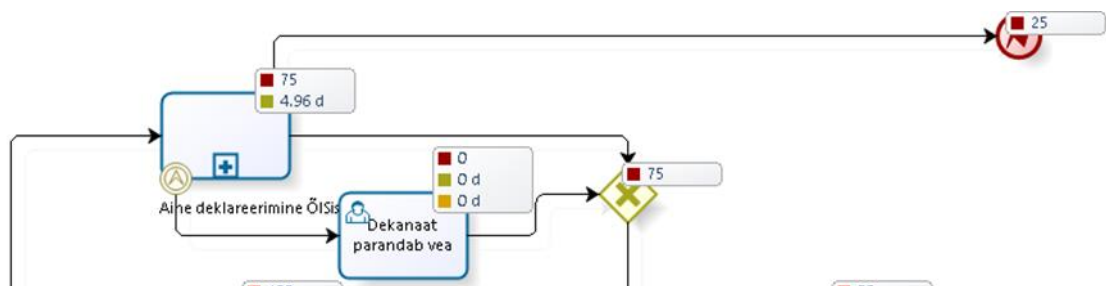




Joonis 48. Kahe katkestava sündmusega alamprotsess põhiprotsessi tasemel



Joonis 49. Kahe katkestava sündmusega alamprotsessi detailne vaade



Joonis 50. Kahe katkestava sündmusega alamprotsessi sisaldava mudeli simulatsioon

Jooniselt 50 on näha, et simulatsioon toimib, kuid katkestavat eskaleerivat sündmust ei käivitata mitte ühtegi korda. Kui alamprotsessis asendada veasündmus eskalatsiooniga ning võtta süsteemi tegevuse küljest sündmus ära, siis toimib simulatsioon korrektselt. Seega ei saa viga seineda sündmuse tüübis, vaid simulatsioonimootori suutmatuses tekitada järgnevusvoole erinevaid radasid. Kuna tegemist on küllaltki ebatõenäolise juhtumiga, otsustati modelleerimisel loobuda eskaleerivast sündmusest ning alles jäeti tudengi vea tõttu tekkinud olukord.

Samas alamprotsessis tekkis ka olukord, kus autori hinnangul oleks õigustatud süsteemi kui tegutseja esiletoomine. Süsteemi tegevuste vähesuse tõttu seda siiski ei tehtud. Süsteemi tegevuse modelleerimine eraldi basseinis/rajal oli eespool esitatud muustrina. Bizagi Modeler ise defineerib sellist modelleerimist antimustrina ning ka mitmed analüütikud kalduvad pigem sellist lähenemist kasutama. Arvamuste lahknemise ning kindlate reeglite puudumise tõttu soovitab autor süsteemi kui protsessis osaleja tegevuste eraldi modelleerimist siis, kui selle poolt tehtavaid tegevusi on rohkem kui kolm või kui tema olulisus protsessis on nii suur, et teda saab käsitleda tegutsejana. Aine deklareerimise puhul oleks autori hinnangul selline lahendus põhjendatud, sest kogu tegevus toimub süsteemis.

Loengute ja praktikumide alamprotsesside elemendid olid *Loop*- ehk tsükli tüüpi. Enne simulatsiooni käivitamist esines kartus, et seda tüüpi elemendid ei pruugi käituda simulatsiooni korral korrektselt ning tekkivad andmed võivad olla eksitavad. Mõlemad alamprotsessid olid määratud korduma 16 korda ehk sama palju, kui on semestris õppetöö nädalaid. Joonisel 51 on kirjeldatud simulatsiooni tulemused.

Name	Type	Instances completed	Instances started	Min. time	Max. time	Avg. time
KURSUSE PÕHIPROTSESS	Process	47	100	28d 23h 22m 58s	31d 51m 36s	30d 9h 34m 58s

Joonis 51. Õppeaine läbimise protsessi simulatsioonitulemused

Protsessi käivitati 100 korda ning lõpetati 47 korral. Aeg, mis on kulunud protsessi läbimiseks, on üsna aktsepteeritavas vahemikus. Selles ajavahemikus sisalduvad tegevused, mis viiakse läbi kõikide tegutsejate poolt, ning seetõttu on ka selle maht niivõrd suur. Näiteks kulub tudengil semestri jooksul praktikumi tundides käimisele üks päev. Kui sinna juurde veel lisada kodus tehtava töö maht ning õppejõu tegevused, tuleb minimaalne aeg 28 päeva üsna kergelt kokku. Aegade erinevus tekib järgnevusvoo radade ehk erinevate stsenaariumite tõttu.

Lisaks tsükli tüüpi alamprotsessile tekitas küsimusi ka mitmekordse (*Multi Instance*) tegevuse käitumine. Ametlikult ei toeta Bizagi Modeleri simulatsioonimootor seda tüüpi

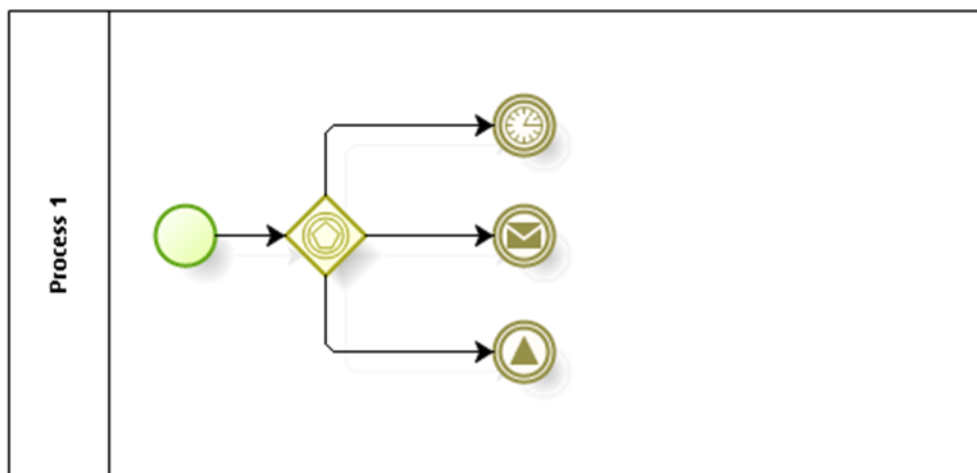
tegevust, kuid uurimise eesmärgil prooviti kontrolltöö alamprotsessis sellise elemendi käitumist simuleerida. Tegevuseks oli „Viib läbi järeltöö“, mida käivitatakse maksimaalselt kaks korda. Probleeme sellise elemendi kasutamisega ei tekkinud ning anomaalseid kõrvalekaldeid simulatsiooniandmetes ei täheldatud. Siiski on kindlam kasutada tsükli tüüpi alamprotsessi, mida simulatsioonimootor toetab. Nende elementide käitumine simulatsioonis on analoogne.

### 4.3 Klienditeeninduse protsess kohvikus

Täielik mudel on esitatud lisas 1.

Näite modelleerimisel keskenduti peamiselt mitme algündmusega mustri leidmisele. Mudel on praegu kasutusel õppenäitena, kuid selle uurimisel õppeaines keskendutakse ressursikulu analüüsile. Käesolevas lõputöös keskendutakse protsessi algusele, kus võib toimuda mitu erinevat stsenaariumit korraga.

Joonisel 52 on alapeatükis 2.11 välja toodud muster.



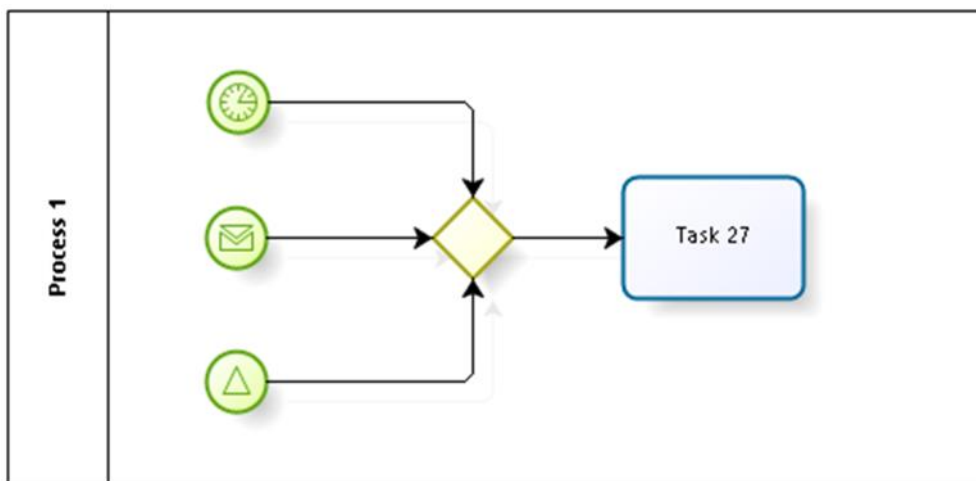
Joonis 52. Mitmekordsete algündmuse muster

Kahjuks ei sobi selline muster situatsioonis, kus kohviku uksest võib sisse jalutada üks inimene või kolm inimest. Olukorra pöörab ekstreemseks see, kui saabub hoopis turistigrupp 40 inimesega. Põhjus, miks muster ei sobi, seisneb sündmuste toimumise tõenäosuse moonutamises ja elementide sobimatuses. Nimelt kui kasutada mustrit, siis määratakse välistavas sündmusepõhises (*Event-Based Exclusive*) lüüsis protsessivoo tekkonnale tõenäosus. Kui modelleerija ei reguleeri tõenäosuste sätteid ise, on selleks

võrdselt jagatud tõenäosus. Mustri näite puhul oleks selleks  $100 : 3 = 33,33$ . Sündmuse objektile on võimalik lisada vastav tõenäosusjaotus, kuid lõplik tõenäosus oleks jaotuste summa, mis annab vale tulemuse.

Teine probleem seisneb kasutatava lüüsi tüübis. „Välistav“ ei ole omadus, mida saaks kasutada situatsioonis, kus sündmused saavad juhtuda korraga. Samuti ei saa kasutada paralleelset lüüsi, sest alati ei pruugi kõik sündmused toimuda ning simulatsioonimootoris tekiks viga. Seega tekitaks mustri kasutamine kaks probleemi: protsessi moonutatud tõenäosus ja vale toimimine simulatsioonimootoris.

Olukorda aitab lahendada antimuster. Joonisel 53 on toodud alapeatükis 2.11 esitatud antimuster.



Joonis 53. Mitmekordsete algündmuste antimuster

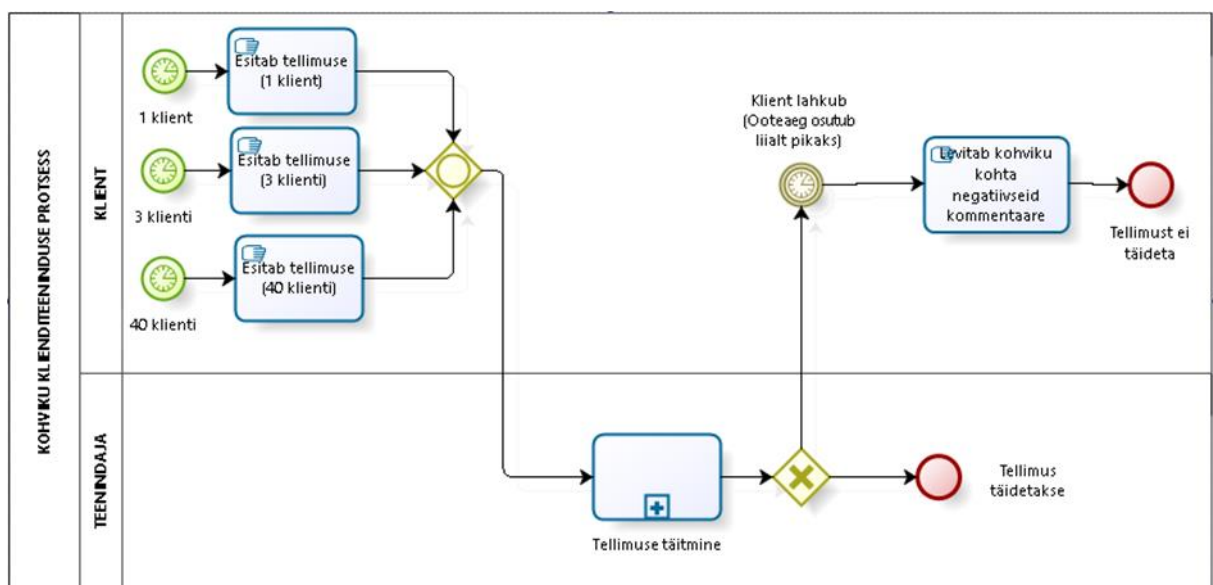
Esiteks sobib elementide järjekord protsessi loogikaga. Sündmused toimuvad protsessi alguses ja nende toimumise tõenäosusjaotus ei sõltu lüüsisist. Kindlasti tuleb sündmuste elementidele määrata tüübid või vähemalt kahele neist. Vastasel juhul annab mudeli simulatsioon veateate. Samas ei ole antimuster täielikult sobiv. Vale lüüsi tüüp esineb ka siin. Kuna tegemist on järjekordselt välistava lüüsiga, tuleb see asendada sisaldava (*Inclusive*) lüüsiga. See lüüs ei välista mitte ühtegi haru ning samas ei eelda nende täitmist.

Järgmine probleem tekib sündmustele järgneva tegevuse elemendiga. Kolme erineva sündmuse puhul on tegevuse ressursikulu erinev. Antimustri puhul seda erinevust ei teki.

Tegevuse element tuleb paigutada igasse harusse, mis eelneb harusid koondavale lüüsile. Sel viisil toimub ressursikulu arvutamine iga sündmuse puhul eraldiseisvalt.

Tellimuse täitmise osas ei pea tegevuse elemente juurde tekitama, sest ressursikulu suureneb hüppeliselt tellimuse esitamise osas. Täitmine toimub ikka endisel moel, kuid simulatsiooni paremaks toimimiseks on alamprotsessi elemendi küljest eraldatud katkestav taimeri sündmus.

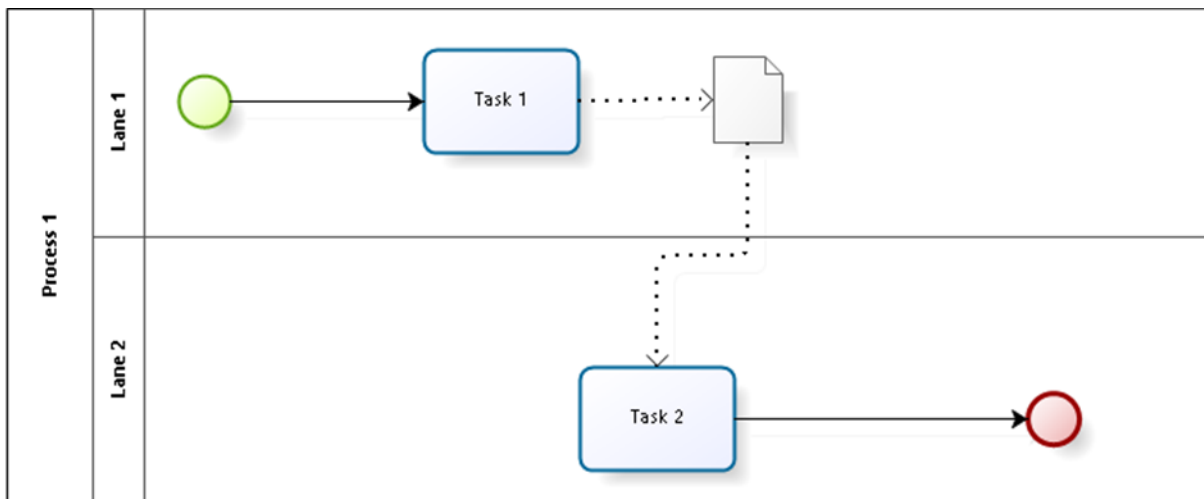
Järgnevalt on joonisel 54 kujutatud kohviku klienditeeninduse protsess kasutatud mustriga.



Joonis 54. Mitme algündmusega kohviku klienditeeninduse põhiprotsess

#### 4.4 Abstraktne katkeva vooga protsess

Järgnevalt on joonisel 55 kujutatud uurimisobjektina protsessi, kus elemendid ei ole ühendatud vooelementidega.



Joonis 55. Katkeva vooga protsess

Probleem seisneb sellise mudeli ühildamatuses simulatsioonimootoriga. Mudel ise valideerub BPMNi reeglite suhtes. Näitena võiks tuua posti liikumise ettevõttes sekretäride vahel. *Lane 1* ja *Lane 2* oleksid erinevad sekretärid ning protsess algab esimese poolt tehtava tegevusega, mille tulemusel lisandub protsessi tehis. See tehis on sisendiks teise sekretäri tegevusele, kuid simuleeritavat ühendavat elementi kahe tegevuse vahel ei ole. Sekretärid ei suhtle omavahel, post jäetakse kindlasse kohta, kust selle saab igal ajal edasi toimetada. Kuidas muuta selline protsess simuleeritavaks?

Tegelikult esines probleemi püstitamisel viga. Sellist olukorda, kus vooelement puudub, ei saa loogiliselt tekkida. Näiliselt võib ühendus tegevuste vahel puududa, kuid realses olukorras jätkub protsess mingisuguse sündmusega. Selliseks sündmuseks võib olla näiteks teise tegutseja otsus hakata järgnevat toimingut teostama. Vahel kasutatakse tühja sündmuse antimustrit. Optimaalne lahendus oleks lisada kahe tegevuse vahele *Lane 2* rajale taimer (*Timer*) tüüpi sündmus. Sündmust saaks käivitada tsükliliselt või tingimuslikult ning lisaks saab sellele määrata tõenäosusliku käitumise. Tegevuselementide otsene ühendamise ei ole vale ning võimalus määrata tõenäosuslik käitumine säilib, kuid alati ei pruugi järgneda tehisele tegevus, vaid näiteks lüüs, ja sellisel juhul on vaja kasutada sündmust. Selle näite puhul saab öelda, et sekretär 2 tuleb mingi aeg kindlasti vaatama, kas posti on. Seda saab vaadelda kui sündmust, mis kindlasti toimub, ehk sellele on võimalik määrata ajaline piirang.

## 5 Uuringu tulemuste analüüs

### 5.1 Mustrite rakendamise võimalikkus

Mustrite rakendamine ei osutunud Bizagi Modeleris lihtsaks. Peamine põhjus seisneb tarkvara disainis, mis seab simuleeritavatele mudelitele häirivad piirangud. Kui protsessi loogilise disaini tasemel ei teki mustrite rakendamisega probleeme, siis füüsilise disaini tasemel pidi muutma kõikide äriprotsesside mudeleid.

Telekommunikatsiooni ettevõtte äriprotsessi mudeli esimeses versioonis muutus alamprotsesside elementide tüübi sobimatuse tõttu kõikide alamprotsesside loogiline disain. Kuigi see ei mõjutanud mudeli käitumist simulatsioonimootoris, tekitab selline ühildamatus segadust tegutsejate esiletoomisega ja raskendab mudeli loetavust. Mudeli simulatsioonis toimus seadmete hõivatuse simuleerimine. Teise versiooni puhul sai kiirelt selgeks, et mitme basseini muster ei ole otstarbekas suuremahuliste protsesside kirjeldamiseks.

Tudengi õppeaine läbimise protsessi simuleerimisel ei leitud lahendust anomaalsetele protsessivoogudele, mis tekkisid alamprotsessiga kaasnevate mitmekordsete katkestavate sündmuste mõjul. Probleemi puhul osutus määravaks mitte sündmuste arv, vaid nende asukohad. Tegevuse elemendi külge modelleeritud sündmus suunab protsessivoo läbi kõrvalise haru, kuid peamine marsruut jääb läbimata. Simuleeritava mudeli nimel loobuti tegevusega seotud veaolukorrast.

Kahjuks ei õnnestunud eristada, kas simulatsioonimootor kohtleb sisaldavat (*Inclusive*) lüüsi korrektselt. Andmete uurimisel jäi pigem mulje, et element käitub analoogselt välistava (*Exclusive*) lüüsiga. Samas suudeti kohviku näites tuvastada sellise elemendi erisus paralleelsest (*Parallel*) lüüsisist. Põhiprotsessi tasemel sai kinnitust teoreetilises osas tutvustatud paralleelsetest moodulitest väljuvate voogude mustri töökindlus. Simulatsioon toimus veatult ning tulemuses ebaadekvaatseid andmeid ei täheldatud. Mitmekordsete (*Multi Instance*) ja tsükliliste (*Loop*) tegevuste elementide simuleerimisel vigu ei esinenud, kuid ametliku toe puudumise tõttu tasub esimest varianti vältida.

Kohviku klienditeeninduse protsessi näites õnnestus näidata, miks sobib antimuster probleemi lahendamiseks. Samuti õnnestus simulatsioonimootoris tuvastada sisaldava (*Inclusive*) lüüsi käitumise erisus paralleelse (*Parallel*) lüüsisist.

Katkeva protsessivooga mudeli puhul õnnestus leida teoreetiline põhjendus, miks sellist mustrit ei saa tekkida. Tehised võivad olla ja enamasti ongi tegutsejate tegevuste sisendiks, kuid näiline vooelemendi puudumine ei tähenda selle reaalsel puudumist.

## **5.2 Testjuhitud modelleerimise metoodika efektiivsus**

Kahjuks ei osutunud testjuhitud modelleerimise metoodika efektiivseks. Põhjus seisneb erinevates disaini etappides, kus selliseid mooduleid on võimalik kasutada. Nagu eespool välja toodi, ei saa esialgses ärianalüüsi etapis mustreid kasutada. Erandid võivad tekkida olukorras, kus mudeli koostajal on märkimisväärseid teadmisi äriprotsessi valdkonnast ja spetsiifilisest organisatsioonist.

Sellise metoodika kasutamine eeldab teadmisi protsessi kohta, mida esialgu ei pruugi modelleerijal olla – need on teadmised, mis käivad erinevate protsessivoode harude kohta või hõlmavad tegutsejate omavahelist suhtlust. Sellist informatsiooni omamata ei ole võimalik ennustada, milline protsess välja nägema peaks, mistõttu on moodulite loomine äärmiselt raske.

Kui sellised teadmised on olemas, siis peab mooduli konstrueerimisel füüsilisel tasemel arvestama CASE-vahendi piirangutega. Seega ei pruugi BPMNi suhtes kõige korrektsem mudel töötada. Sellise ajakuluga ei pea arvestama vilunud tarkvara kasutajad, kuid algajatele võib see probleeme valmistada.

Esile tuli, et kauem läks aega selliste mudelite koostamisega, mis tuli luua „as is“-staadiumis. „To be“-mudelitega, mida kasutati protsessides, kus oli eksisteeriv õppenäide olemas, läks veidi kiiremini. Huvitava asjaoluna selgus, et eraldi basseinidega mudelite konstrueerimise ajad olid metoodikate erinevuse korral üpris analoogsed. Autori enda hinnangul aitas eraldiseisev bassein fokusseerida tähelepanu mustrite sobivusele, sest protsessis osalejate tegevused tulid paremini esile. Ajakulu erisused on esitatud tabelis 1.



Tabel 1. Modelleerimise metoodikate ligikaudne ajakulu

<b>Protsess</b>	<b>Ajakulu puudumisel</b>	<b>Ajakulu testjuhitud metoodika korral</b>
Telekommunikatsiooniettevõtte erakliendi protsess: basseinideta mudel	4	7
Telekommunikatsiooniettevõtte erakliendi protsess: eraldi basseinidega mudel	6	6
Õppeaine läbimise protsess	4	6
Klienditeeninduse protsess kohvikus	4	2

BPMNi kõige olulisem omadus mõistetavuse kõrval on tema paindlikkus. Paindlikkus on kahjuks ka mustrite raskendatud rakendamisevõimaluste põhjuseks. Standardi põhjal on defineeritud elemendid ja põhireeglid kuidas neid kasutama peab. Neid kasutades ei teki protsessi loogilise disaini korral mustritega suuri probleeme, kuid need reeglid ei kehti kahjuks füüsilise disaini tasemel. Standardit enda kasuks tõlgendav CASE-vahend sunnib tihtipeale simuleeritava mudeli konstrueerimisel muutma mustreid tundmatuseni või kasutama elemente, mis loogiliselt ei sobiks protsessi disainiga.

## 6 Kokkuvõte

Lõputöö eesmärkideks olid BPMNi peamiste antimustrite kirjeldamine ning neile vastavate mustrite abil simuleeritavate õppenäidete loomine ja täiendamine. Lisaks püüti anda hinnang testjuhitud metoodika sobivusele äriprotsesside modelleerimisel. Sellise metoodika puhul luuakse mustrite abil moodulid, mida kontrollitakse enne ühendamist simulatsioonimootoris.

Töö peamiste tulemustena valmisid Bizagi Modeleri CASE-vahendis simuleeritavad õppenäited, mille konstrueerimisele läheneti võimalikult mustripõhiselt. Samuti kujunes välja testjuhitud metoodika sobivuse esialgne hinnang.

Äriprotsesside puhul on mustripõhine lähenemine üpriski raskendatud. Protsessi unikaalsus võib põhjustada mustri üldise sobimatuse loogilise disaini tasemel, kuid ei pruugi seda teha. CASE-vahendi piirangud sunnivad aga mustrit enamasti tugevalt kohandama, mistõttu võib öelda, et füüsilise disaini tasemel tekib täiesti uus muster. Seetõttu on lihtsam kaardistada protsess „as is“-etapis mustreid kasutamata. Eespool nimetatud põhjuste tõttu on ka testjuhitud metoodika oluliselt raskendatud ning mõjutab negatiivselt „as is“-mudelite koostamist.

Töö edasiarendustena näeb autor testjuhitud metoodika rakendamise põhjalikumat uurimist „to be“-etapi mudelite modelleerimisel, kus ilmnesisid mõned sobivuse tunnused.

## Summary

The main goal of the thesis was to describe the main antipatterns of BPMN and construct simulation-capable business process models which could be used as learning examples, in accordance with correct patterns. Additionally, an attempt was made to give an evaluation of the suitability of test-driven modelling. Test-driven modelling technique combines the use of patterns to create modules and validation of the CASE-tool's simulation engine in order to construct a business process model.

The main results of the work were simulation-capable pattern based models that can be used as learning examples. Also, an initial evaluation of test-driven modelling technique was given.

While modelling business processes, a pattern-based approach can be rather difficult to use. On a logical design level, use of patterns can be defined by the uniqueness of the process. Furthermore, on the physical design level the limitations of the CASE-tool enforce the user to change the patterns so much, that they cannot be recognized. It is easier to create „As Is“ models without using patterns. Because of those reasons, test-driven technique is hard to apply and has a negative effect on creating „As Is“ models.

Future work could include a thorough research on constructing „To Be“ models while using test-driven technique because of the promising signs it showed in those situations.

## Kasutatud kirjandus

1. Eessaar, E. (2012) Mustrid. Andmete modelleerimise mustrid. TTÜ õppeaine „Andmebaasid I“ (IDU0220) teema 8 konspekt. [WWW] [http://maurus.ttu.ee/ained/IDU0220\\_2012/doc/4/Teema\\_IDU0220\\_8\\_2012.pdf](http://maurus.ttu.ee/ained/IDU0220_2012/doc/4/Teema_IDU0220_8_2012.pdf) (02.05.2017)
2. Brahe S., Bordbar B. (2007) A Pattern-Based Approach to Business Process Modeling and Implementation in Web Services. [WWW] <http://www.cs.bham.ac.uk/~bxb/Papres/steen06.pdf> (02.05.2017)
3. Chinosi, M., Trombetta, A. (2011) BPMN: An introduction to the standard [WWW] <https://profesores.ing.unab.cl/~gbadillo/archivos/cursos/business-process-management/Study%20Materials/Papers/Intro%20to%20the%20Standard%20bpmn.pdf> (02.05.2017)
4. Jošt, G., Huber, J., Heričko, M., Polančič, G. (2016) An empirical investigation of intuitive understandability of process diagrams. – *Computer Standards & Interfaces*. 48, 90–111. [Online] ScienceDirect (02.05.2017)
5. Monteiro, P., Monteiro, M.P. (2012) Documenting patterns with business process models. [WWW] [http://ctp.di.fct.unl.pt/~mpm/pubs/Monteiro\(PLoP-2012\).pdf](http://ctp.di.fct.unl.pt/~mpm/pubs/Monteiro(PLoP-2012).pdf) (02.05.2017)
6. Glushko, Robert J., McGrath, Tim. (2008) Document Engineering: Analyzing and Designing Documents for Business Informatics and Web Services. London: The MIT Press.
7. Correia, A., Brito e Abreu, F. (2012) Adding Preciseness to BPMN Models. – *Procedia Technology*. 5, 407–417. [Online] ScienceDirect (02.05.2017)
8. BPM Anti-Patterns – Improving the practice of BPM. *BPMN Anti-Patterns*. [WWW] <http://www.bpmantipatterns.com/> (02.05.2017)
9. Bizagi Modeler User Guide – A Business Process Modeling Tool [WWW] <http://help.bizagi.com/process-modeler/en/> (02.05.2017)

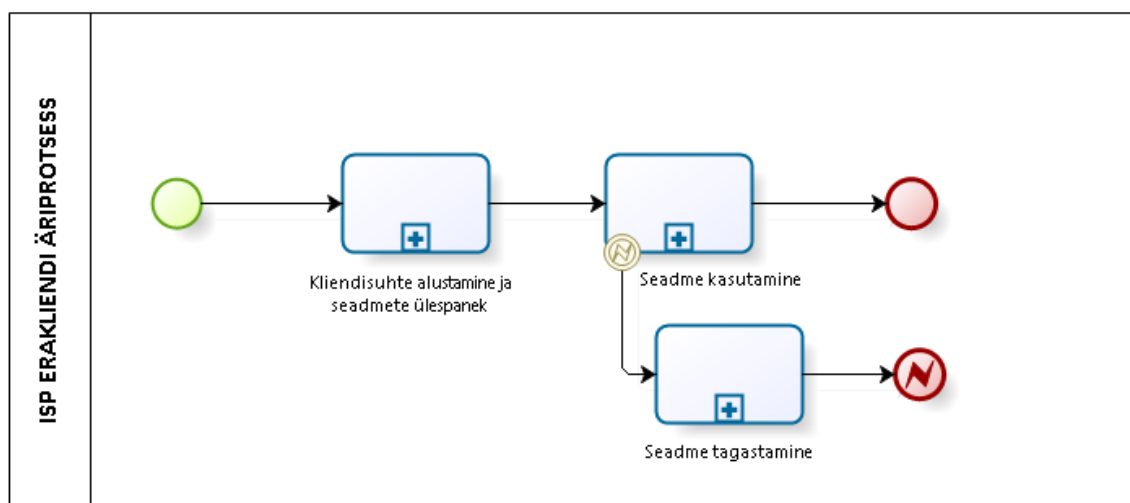
10. Good e-Learning e-õppekeskkonna kodulehekülg. [WWW]  
<http://blog.goodelearning.com/category/bpmn/> (03.05.2017)
11. Anatoli Beljatsuki äriprotsesside modelleerimise blogi. *Process Is The Main Thing*  
 [WWW] <http://mainthing.ru/> (03.05.2017)
12. Projekti Workflow Patterns kodulehekülg. [WWW] <http://www.workflowpatterns.com/>  
 (03.02.2017)
13. Business Process Modeling Initiative. (2003) Business Process Pattern Examples.  
 [WWW] <http://www.workflowpatterns.com/vendors/documentation/BPMN-pat.pdf>  
 (03.05.2017)
14. Van der Aalst, W.M.P, Dumas, M., ter Hofstede, A.H.M, Russell, N. (2005) Pattern-based Analysis of BPMN - An extensive evaluation of the Control-flow, the Data and the Resource Perspectives. [WWW]  
<http://www.workflowpatterns.com/documentation/documents/BPM-06-17.pdf>  
 (03.05.2017)
15. Wohed, P., van der Aalst, W.M.P, Dumas, M., ter Hofstede, A.H.M, Russell, N. (2006) On the Suitability of BPMN for Business Process Modelling. [WWW]  
<http://www.workflowpatterns.com/documentation/documents/BPMN-eval-BPM06.pdf>  
 (03.05.2017)
16. Beljatsuk, A. (2011) Limited Usability of BPMN Lanes – *Process Is The Main Thing*.  
 [WWW] <http://mainthing.ru/item/470/> (03.05.2017)
17. Beck, K. (2002) Test-Driven Development By Example. [WWW]  
[http://www.eecs.yorku.ca/course\\_archive/2003-04/W/3311/sectionM/case\\_studies/money/KentBeck\\_TDD\\_byexample.pdf](http://www.eecs.yorku.ca/course_archive/2003-04/W/3311/sectionM/case_studies/money/KentBeck_TDD_byexample.pdf)  
 (03.05.2017)
18. Yamasathien, S., Vatanawood, W. (2014) An Approach to Construct Formal Model of Business Process Model from BPMN Workflow Patterns – *Proceedings of the 4<sup>th</sup> International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*, Bangkok, 2014, 211–215. [Online] IEEE Xplore Digital Library  
 (03.05.2017)

19. TTÜ õppeaine „Äriprotsesside modelleerimine ja automatiseerimine“ (IDU0111) koduleht. Praktikumide materjalid. [WWW] <http://193.40.244.77/idu0330/idu0111praktikum/> (03.05.2017)
20. EKSS. „Eesti keele seletav sõnaraamat“ [WWW] <http://www.eki.ee/dict/ekss> (03.05.2017)
21. Object Management Group. (2014) Documents Associated With BPMN Version 2.0.2. [WWW] <http://www.omg.org/spec/BPMN/2.0.2/> (03.05.2017)

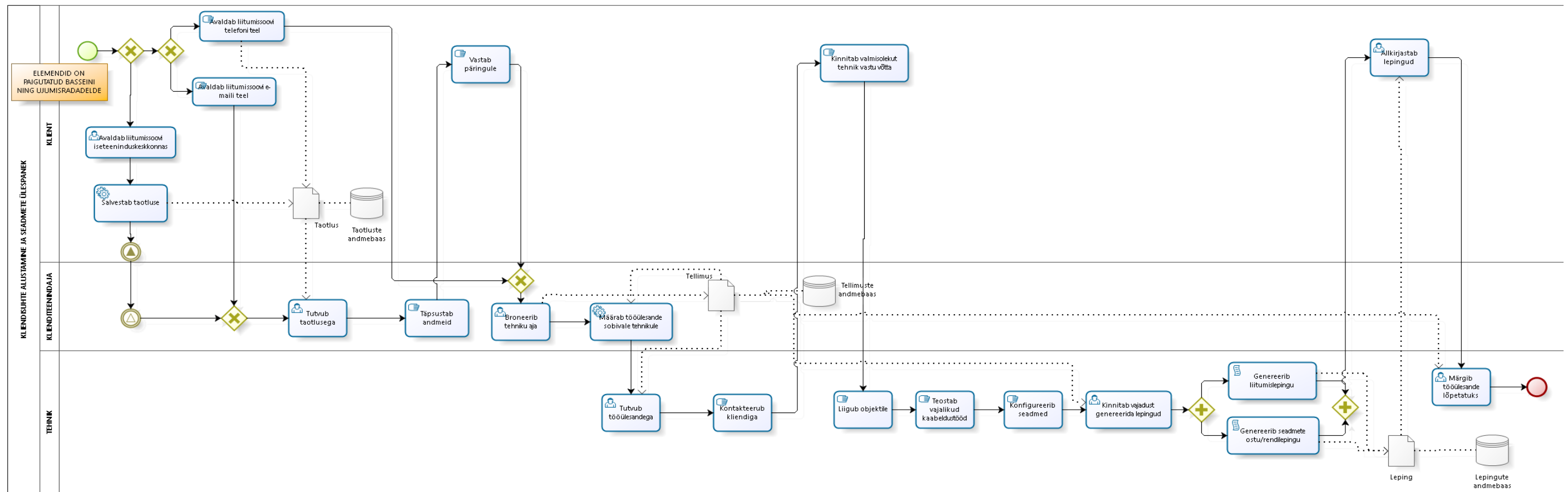
## Lisa 1 – Õppenäidete mudelid

Mudelid asuvad füüsilisel kujul failidena tööle lisatud CD plaadil.

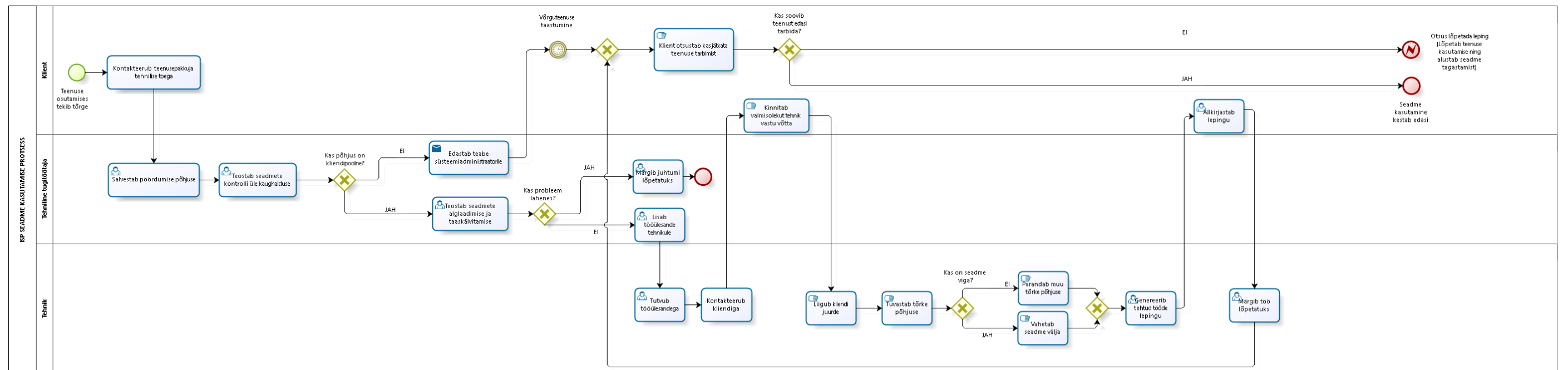
Alapeatükis 4.1.1 kirjeldatud “Ühise basseini mudel”. Tegemist on põhiprotsessi tasemega. Mudelis on kasutatud *Reusable* tüüpi alamprotsessi elemente.



Alapeatükis 4.1.1 kirjeldatud “Ühise basseini mudel”. “Kliendisuhete alustamine ja seadmete ülespanemine” *Reusable* tüüpi alamprotsess.

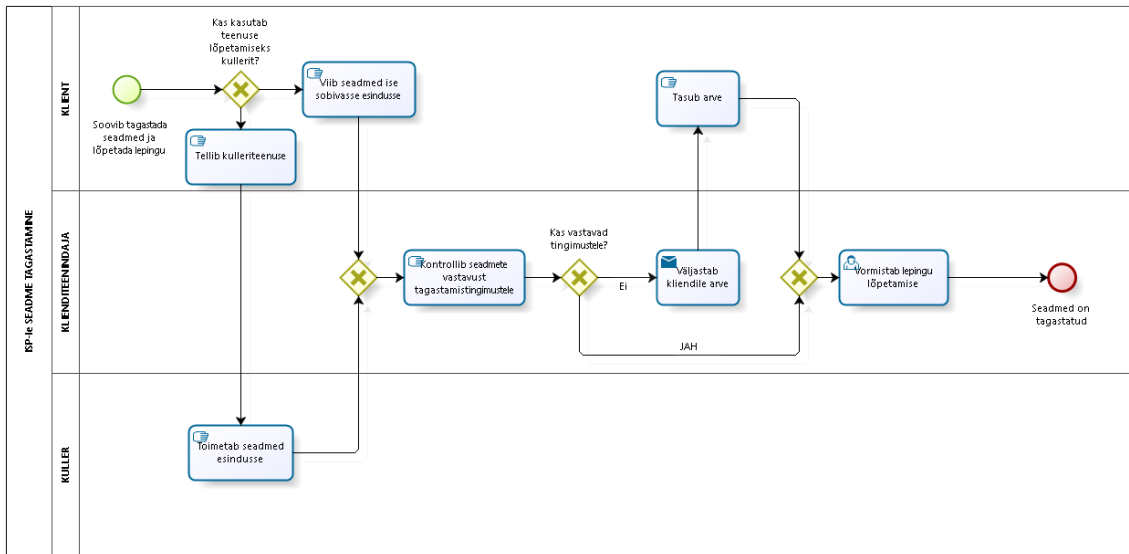


“Seadme kasutamine” *Reusable* tüüpi alamprotsess.

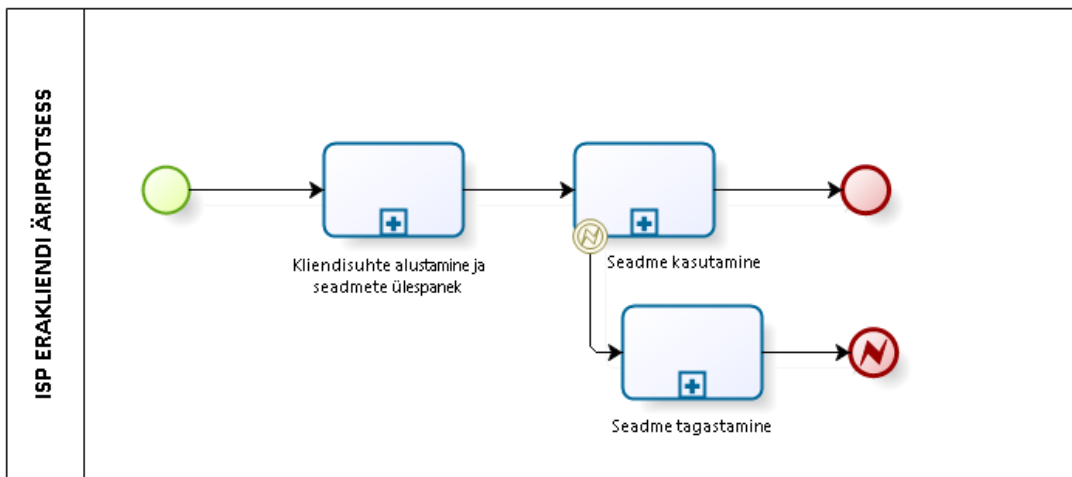




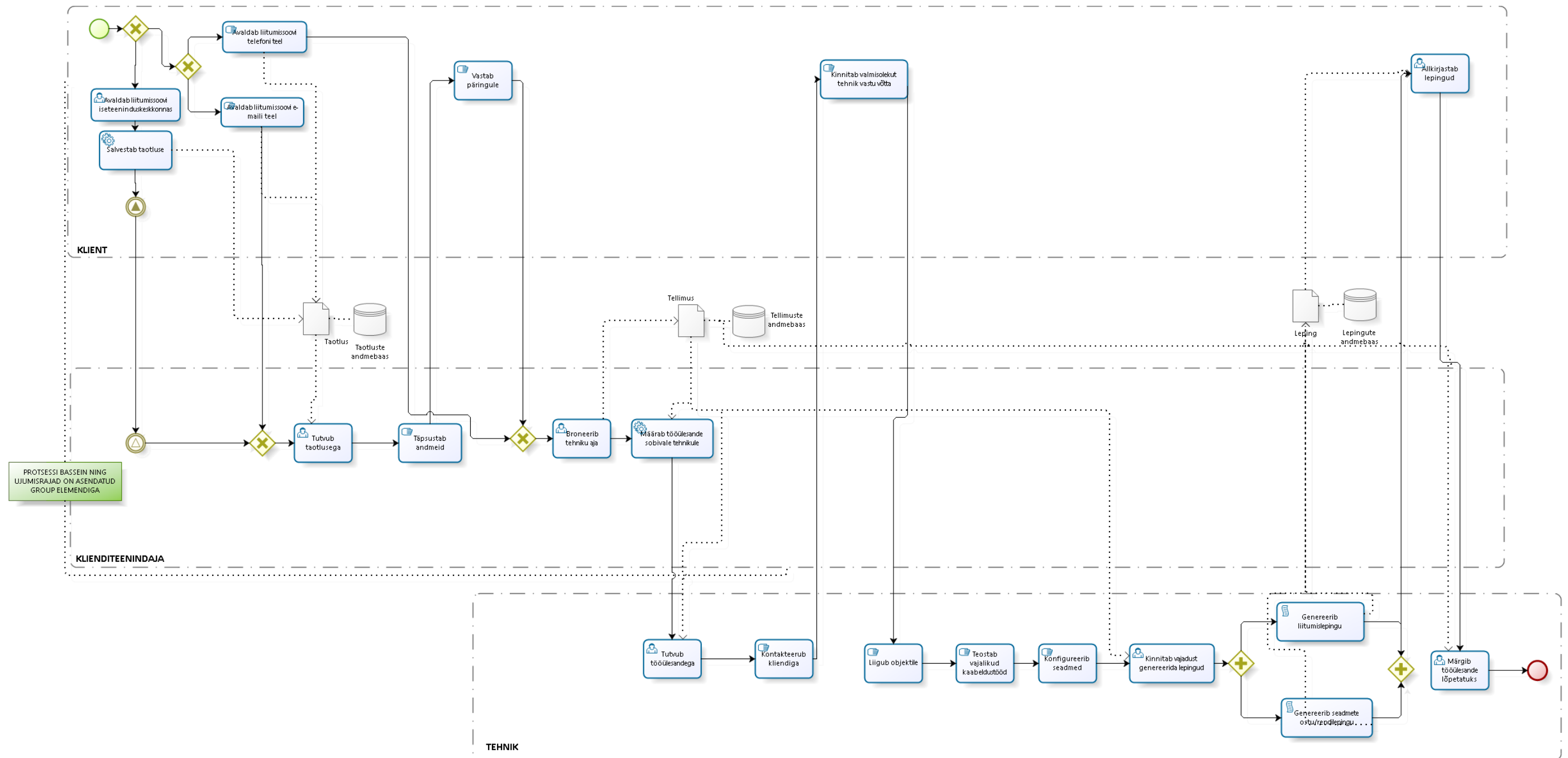
“Seadme tagastamine” *Reusable* tüüpi alamprotsess.



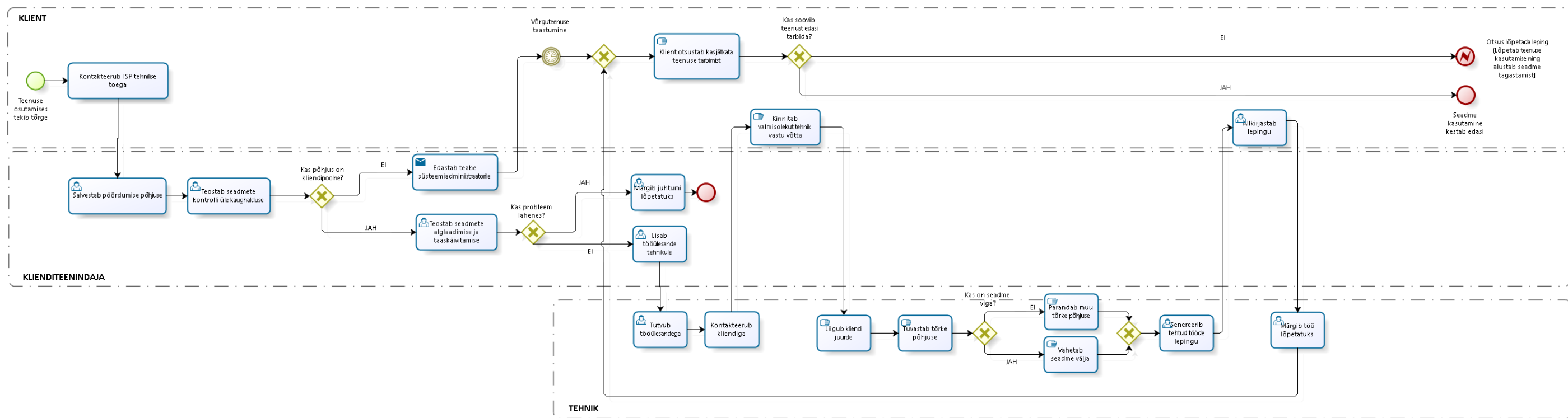
Alapeatükis 4.1.1 kirjeldatud “Ühise basseini mudel”. Tegemist on põhiprotsessi tasemega. Mudelis on kasutatud *Embedded* tüüpi alamprotsessi elemete.



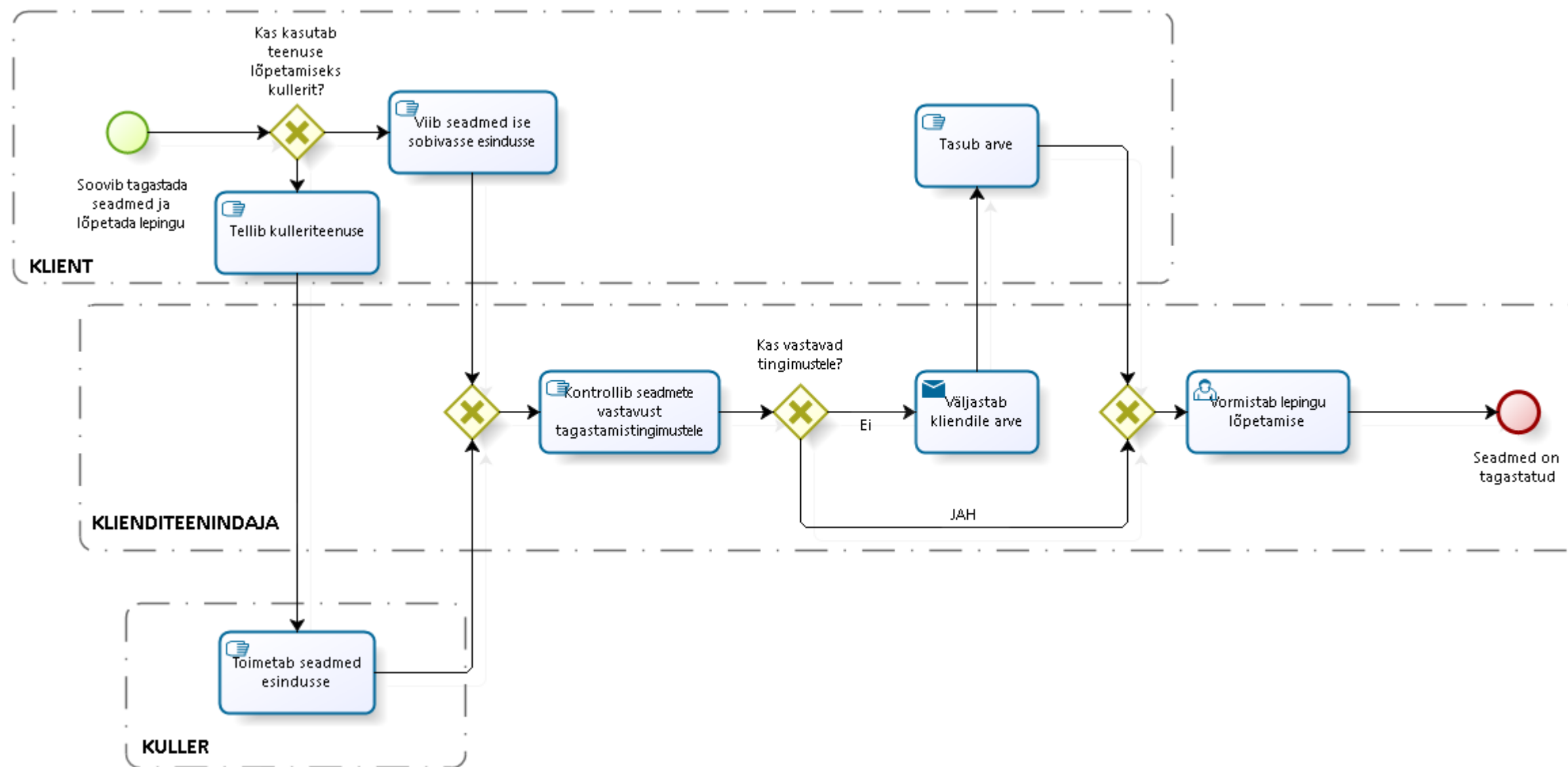
Alapeatükis 4.1.1 kirjeldatud “Ühise basseini mudel”. “Kliendisuhete alustamine ja seadmete ülespanemine” *Embedded* tüüpi alamprotsess.



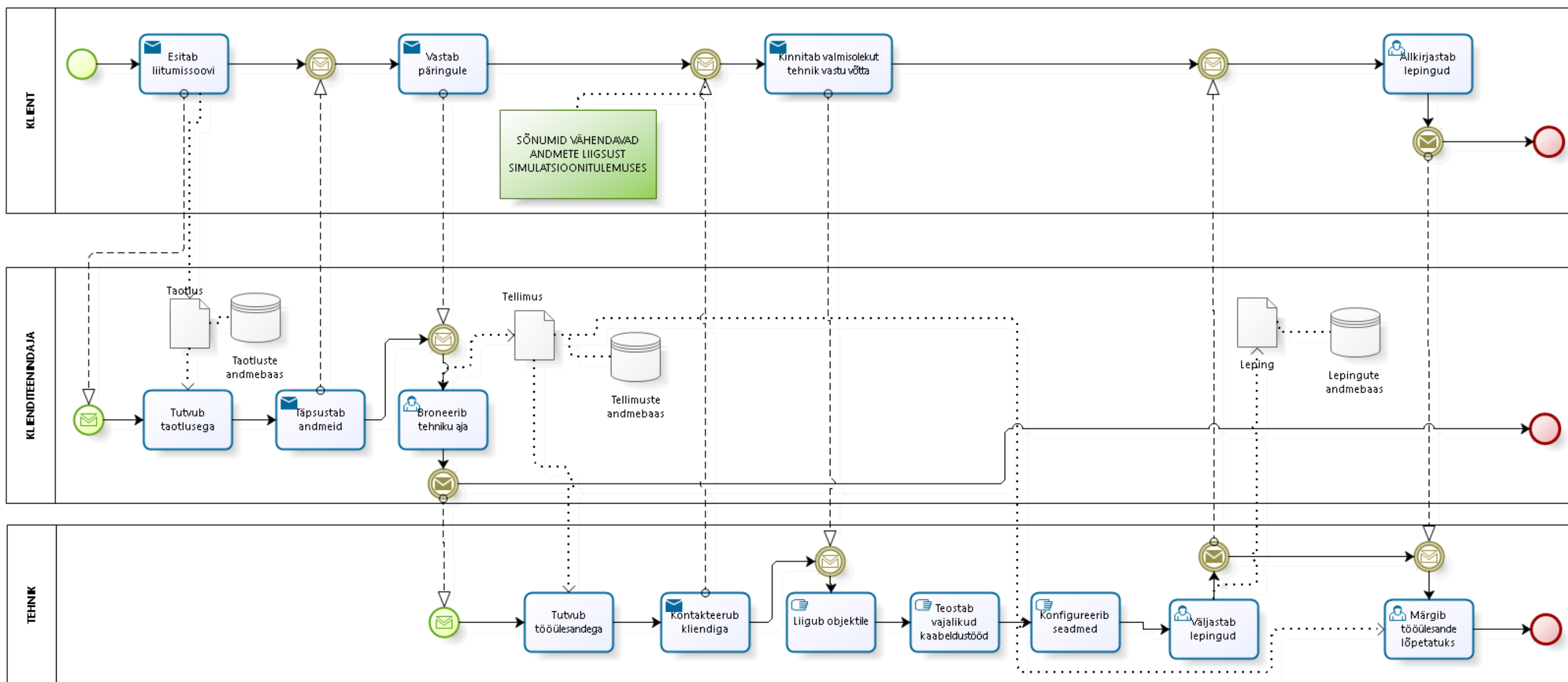
Seadme kasutamine” Embedded tüüpi alamprotsess.



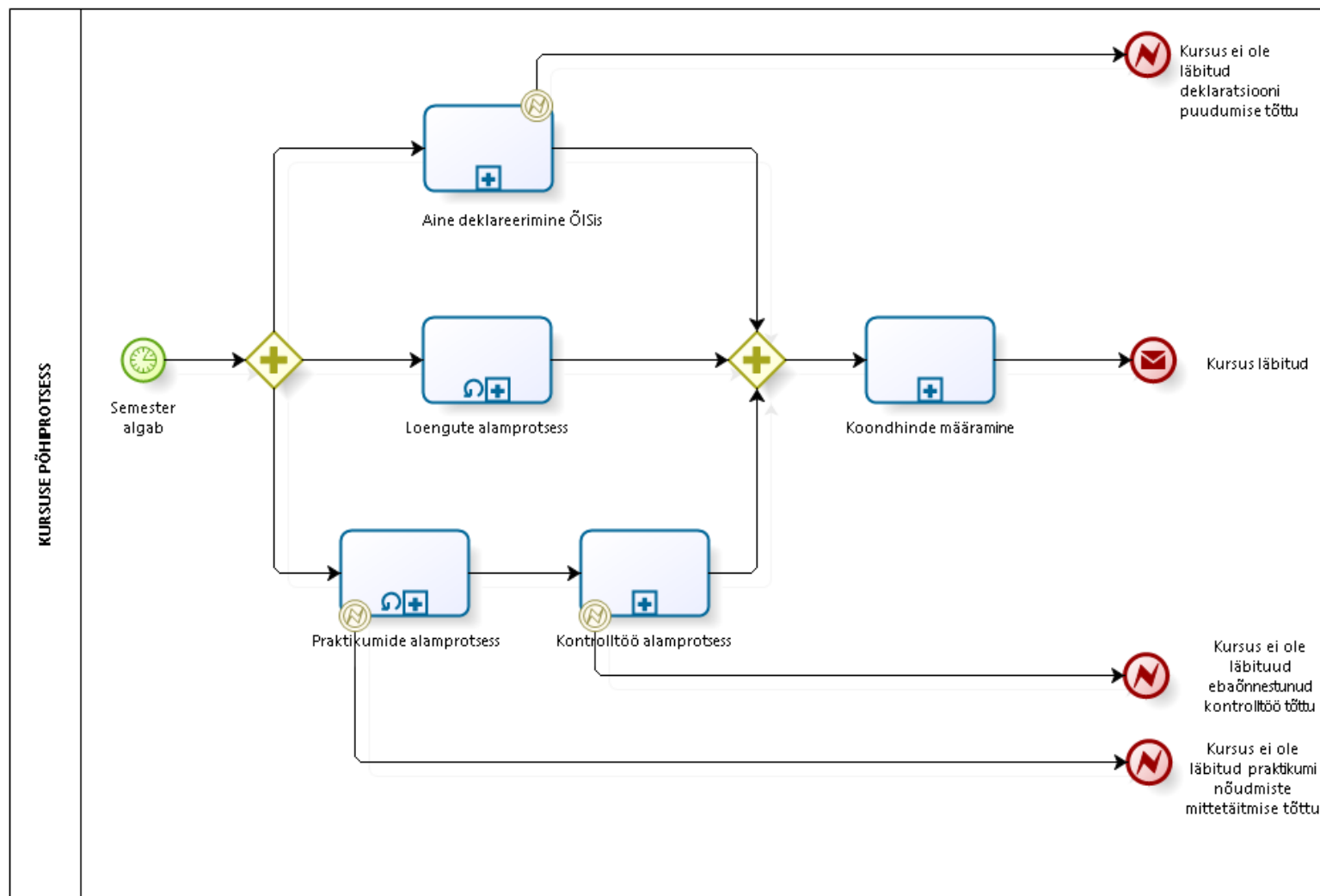
“Seadme tagastamine” *Embedded* tüüpi alamprotsess.



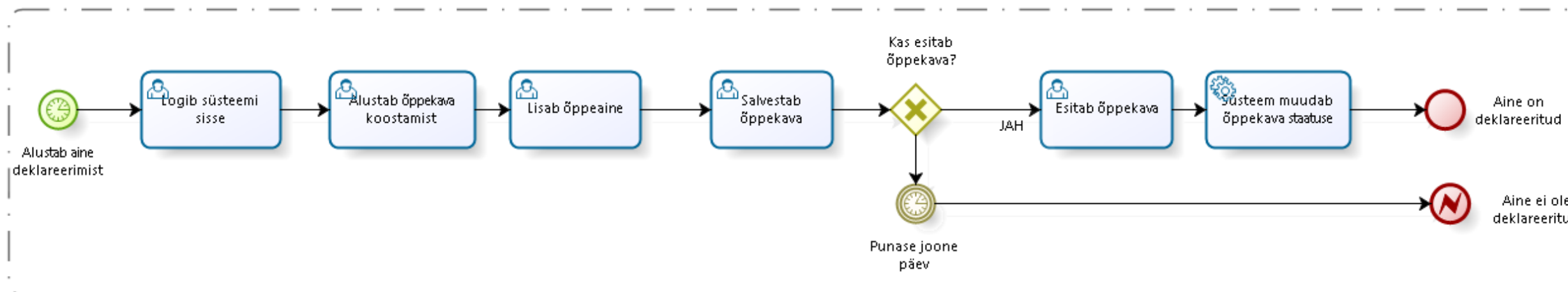
Alapeatükis 4.1.2 kirjeldatud "Eraldi basseinidega mudel". "Kliendisuhete alustamine ja seadmete ülespanemine" alamprotsess.



Alapeatükis 4.2 kirjeldatud „Õppeaine läbimise“ protsess.

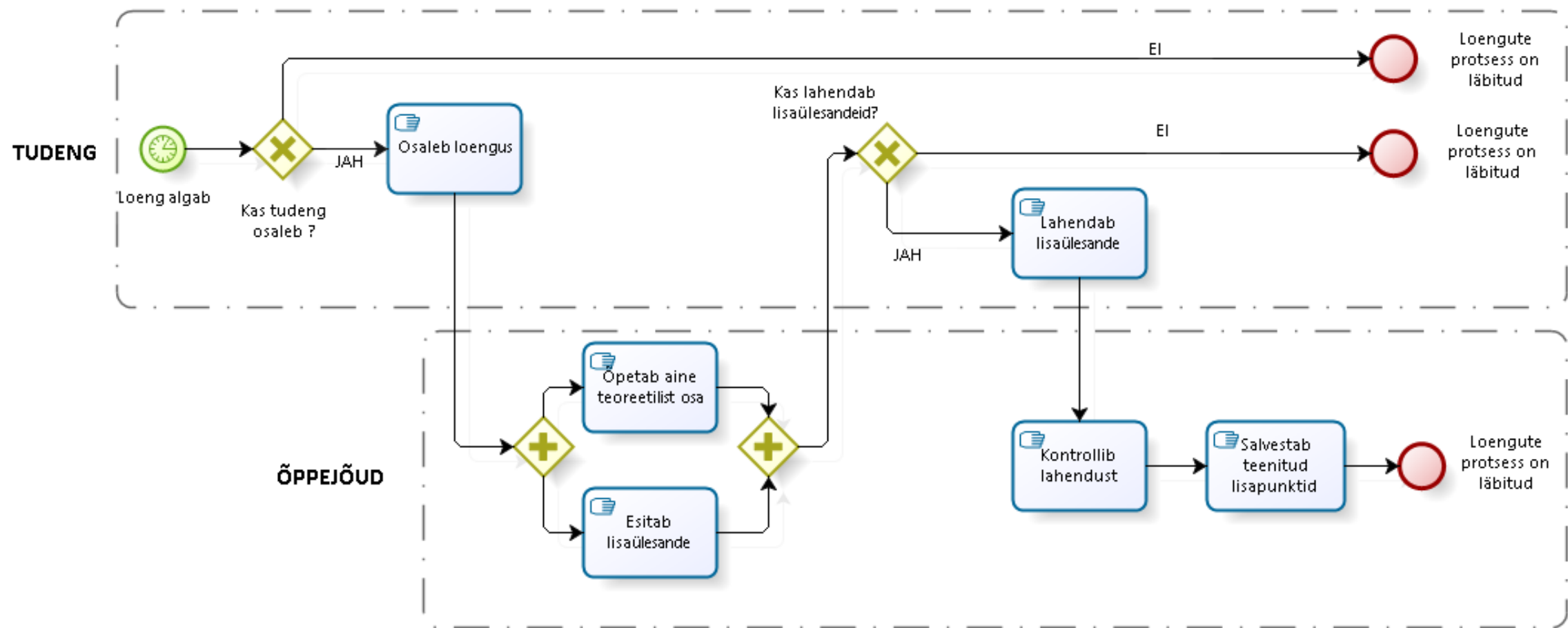


„Aine deklareerimine ÖISis“ alamprotsess.

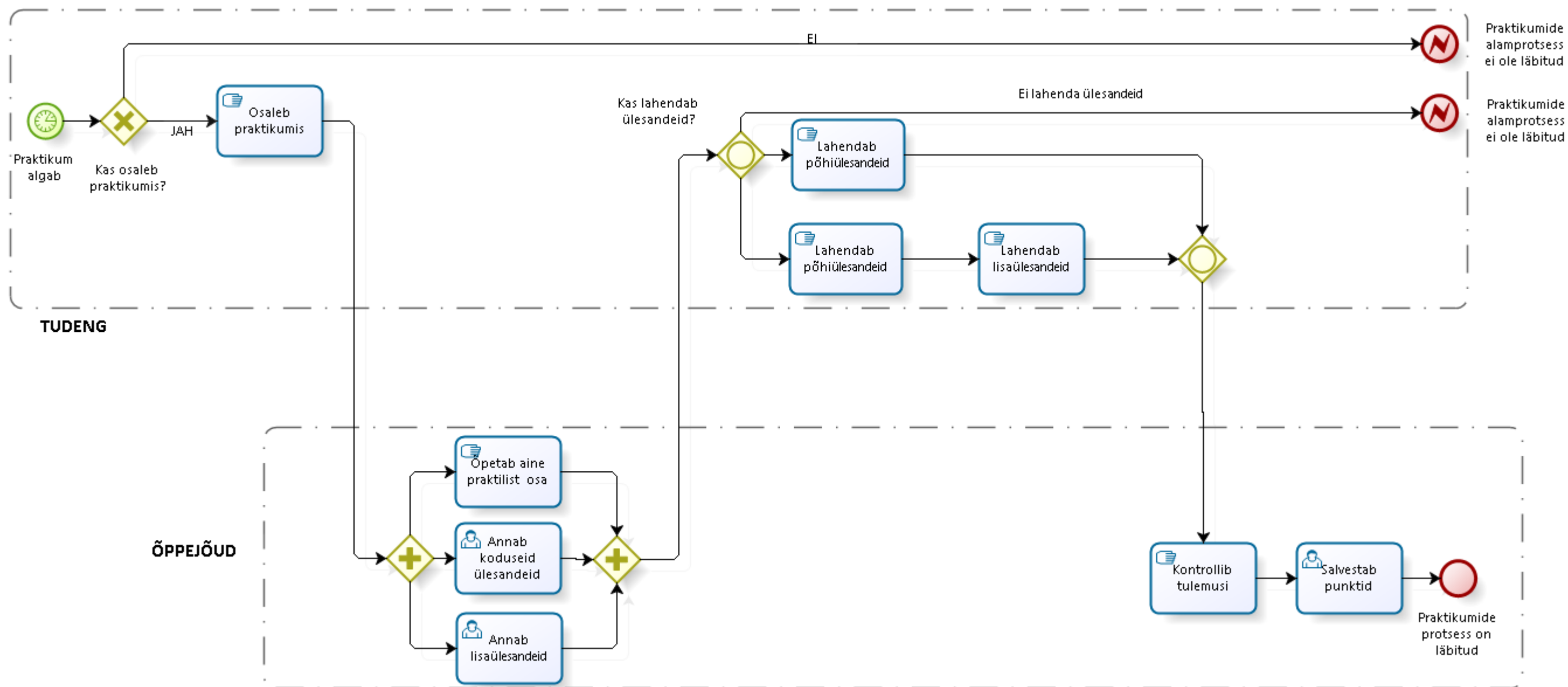


TUDENG

„Loengud“ alamprotsess.

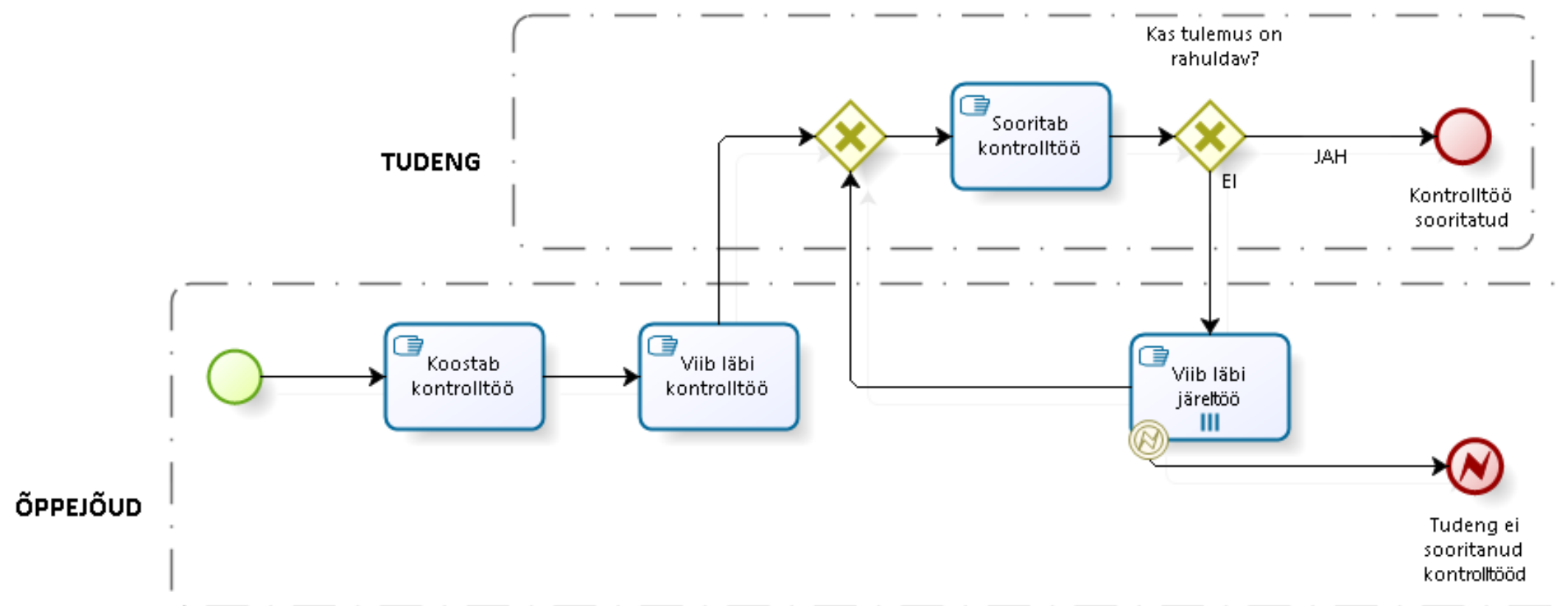


„Praktikumid“ alamprotsess.

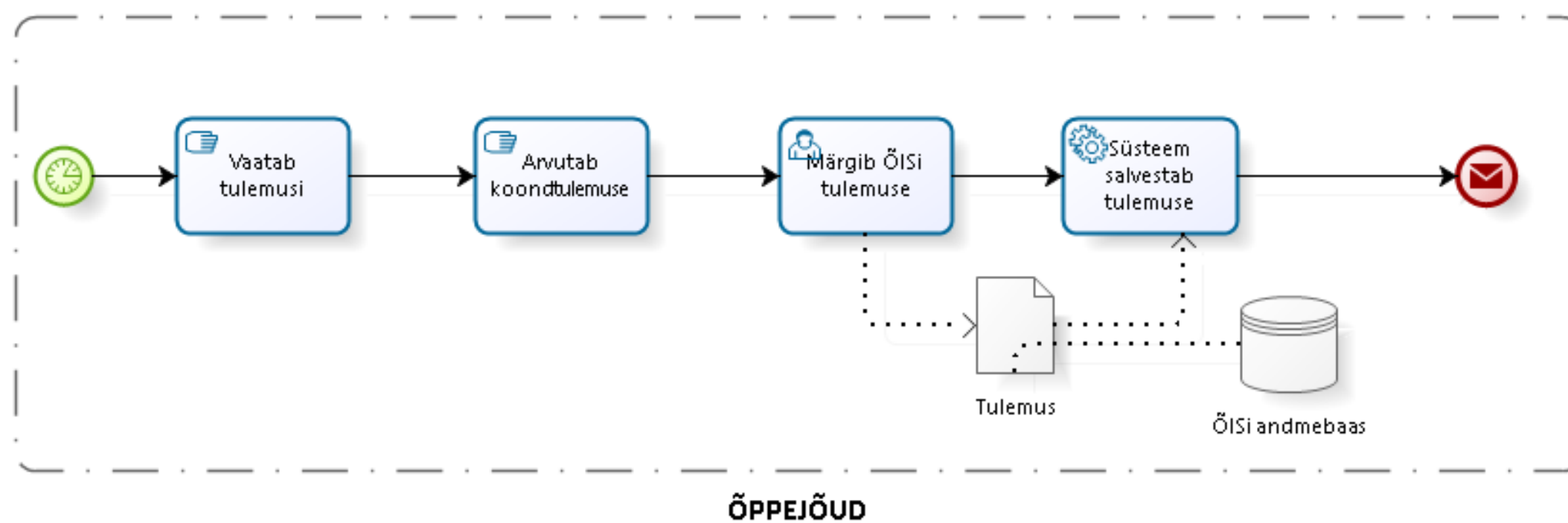




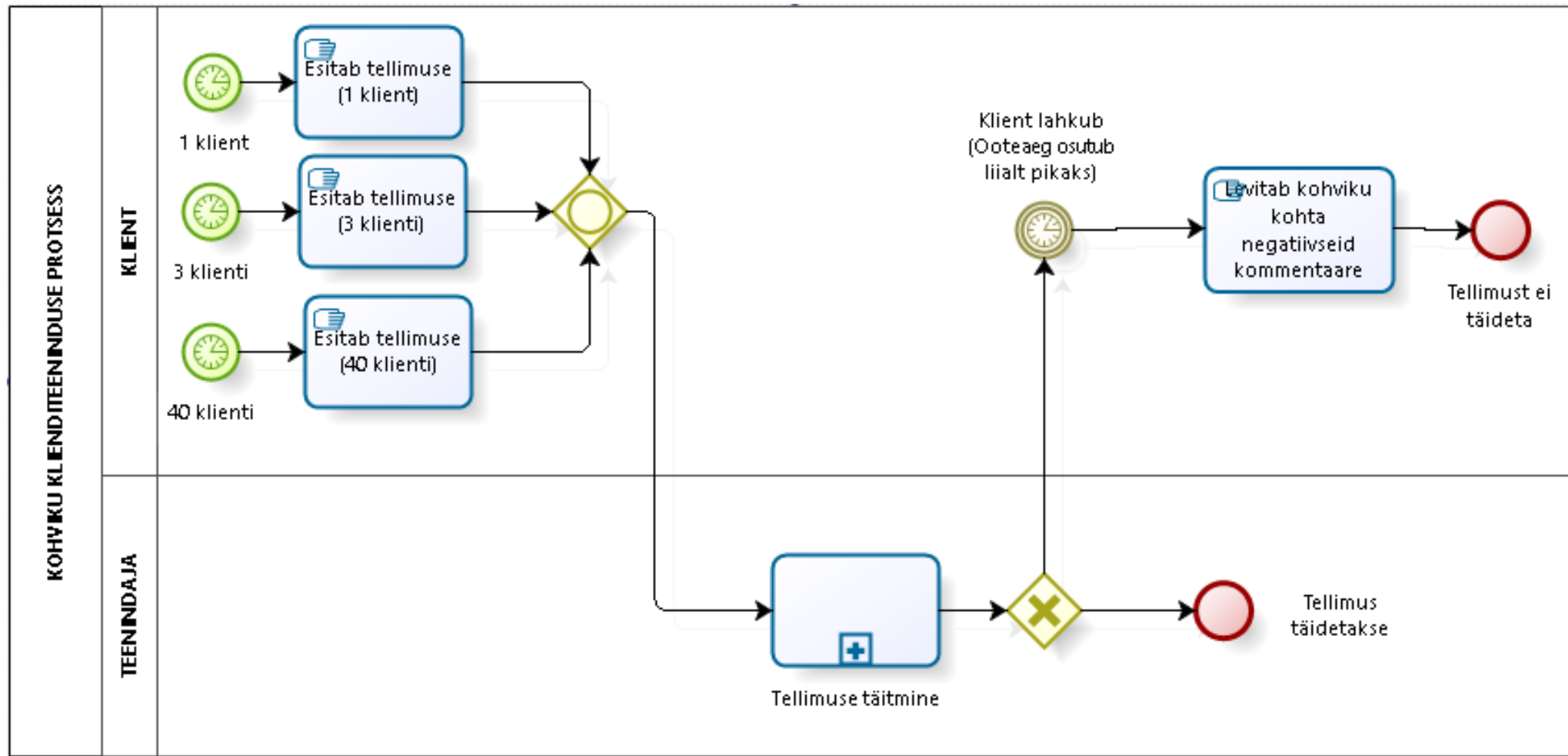
„Kontrolltöö“ alamprotsess.



„Koonddinde määramine“ alamprotsess.



Alapeatükis 4.3 kirjeldatud „Kohviku klienditeeninduse“ protsess.



„Tellimuse täitmine“ alamprotsess.

