

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond  
Tarkvarateaduse instituut

Eve Sirel 172783IABM

**PROTSESSIKAEVE ALGORITMIDE  
EKSPERIMENTEERIMINE JA MUDELITE  
VÕRDLUS EUROOPA PÕLLUMAJANDUSE  
TAGATISFONDI ANDMETE PÕHJAL**

Magistritöö

Juhendaja: Innar Liiv  
Ph.D.

Tallinn 2018

## **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: Eve Sirel

06.05.2018

## Annotatsioon

Käesoleva magistritöö teemaks on protsessikaeve algoritmide eksperimenteerimine ja mudelite võrdlus Euroopa Põllumajanduse Tagatisfondi andmete põhjal. Töö peamiseks eesmärgiks on teostada võrdlus erinevate protsessimudelit genereerivate algoritmide vahel, arvestades valitud protsessikaeve tarkvarasid. Täiendavaks eesmärgiks on protsessikaeve projekti läbimine ning selle tulemusena vaadeldava äriprotsessi logide põhjal mõjutegurite leidmine soovimatute lõpptulemuste realiseerumisel. Töös kasutatakse Euroopa Põllumajanduse Tagatisfondi otsetoetuste taotlemisprotsessi logisid Saksamaal 2015.–2017. aasta kohta. Protsessikaeve tarkvaradest on analüüsiks valitud ProM, Disco ja Celonis.

Protsessimudeleid genereerivate algoritmide võrdlus teostatakse läbi praktilise modelleerimise ning testimise, arvestades valitud kvaliteedinõudeid. Mudeleid koostatakse kõigi kolme tarkvaraga, kasutades kokku nelja erinevat algoritmi ( $\alpha$ -algoritm, heuristiline, induktiivne ja hajus algoritm). Protsessi analüüsi osas kasutatakse erinevate stsenaariumite mõju olulisuse hindamisel hüpoteeside statistilist testimist.

Töö tulemusena koostatakse protsessimudelite loomise algoritmide võrdlev analüüs, arvestades kasutatud tarkvarasid, ning võimalik lähenemine protsessikaeve projektile koos protsessi analüüsi ja tähelepanekutega lõppsündmuse mõjutavate tegevuste osas.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 49 leheküljel, 7 peatükki, 26 joonist, 10 tabelit.

## **Abstract**

### **Process Mining European Agricultural Guarantee Fund Data: Experiments and Model Comparison**

The subject of the master's thesis is to experiment various process mining algorithms and comparing models using data from the European Agricultural Guarantee Fund. The main purpose of the thesis is to compare different process model generating algorithms considering certain process mining software. Additionally, the purpose is to participate in a process mining challenge and discovering factors from a dataset of business process logs resulting in negative end results. European Agricultural Guarantee Fund dataset concerning the application process logs in Germany over the period of 2015 – 2017 is exploited in the thesis. The process mining software explored in the thesis is ProM, Disco and Celonis.

Process model generating algorithms are compared with practical modelling and testing taking into consideration quality requirements. The models are constructed in all of the three software, using four different algorithms in total ( $\alpha$ -algorithm, heuristic, inductive and fuzzy algorithm). The statistical significance of different scenarios is estimated by statistical hypothesis tests in the process analysis section.

The thesis results in a comparative analysis of process model generating algorithms, taking into account the software exploited and a possible approach towards the process mining challenge with process analysis and observations regarding factors affecting end results.

The thesis is in Estonian and contains 49 pages of text, 7 chapters, 26 figures, 10 tables.

## Lühendite ja mõistete sõnastik

BPI	<i>Business Process Intelligence</i> Valdkond, mis rakendab äriteavet äriprotsessides.
BPIC	<i>Business Process Intelligence Challenge</i> Eindhoven University of Technology poolt korraldatav väljakutse BPI valdkonna populariseerimiseks.
BPM	<i>Business Process Management</i> Äriprotsesside juhtimine.
BPMN	<i>Business Process Modelling Notation</i> Märgisüsteem protsesside modelleerimiseks.
EAGF	<i>European Agricultural Guarantee Fund</i> Euroopa Põllumajanduse Tagatisfond
EL	Euroopa Liit
MXML	<i>Mining eXtensible Markup Language</i> Sündmuste logifailide standardvorm enne XES-i.
XES	<i>eXtensible Event Stream</i> Standardiseeritud logide failiformaat.
XML	<i>Extended Markup Language</i> Standardne ja üldotstarbeline märgistuskeel, mille eesmärgiks on struktureeritud info jagamine infosüsteemide vahel.

## Sisukord

1 Sissejuhatus .....	10
1.1 Eesmärk .....	10
1.2 Ülevaade tööst .....	11
2 Üldine taust.....	13
2.1 Protsessikaeve.....	13
2.2 Protsessikaeve tüübid .....	15
2.3 Sündmuste logid .....	18
2.4 Protsessikaeve tarkvarad.....	19
3 Protsessikaeve projekti ülesehitus ja algfaas .....	21
3.1 Protsessikaeve projekti etapid .....	21
3.2 Analüüsitavad andmed .....	22
3.3 Protsessi stabiilsuse kontroll.....	24
3.4 Logide analüüs.....	25
4 Protsessimodelite koostamine.....	28
4.1 Metoodika.....	28
4.2 Celonise ja Disco poolt genereeritud protsessimodelid.....	29
4.3 Protsessimudel ProM-s kasutades $\alpha$ -algoritmi .....	32
4.4 Protsessimudel ProM-s kasutades heuristilist algoritmi .....	33
4.5 Protsessimudel ProM-s kasutades induktiivset algoritmi .....	35
4.6 Protsessimudel ProM-s kasutades hajusat algoritmi .....	37
4.7 Algoritmide võrdlev analüüs .....	38
5 Protsessi analüüs.....	41
5.1 Ajakulu ja jõudluse analüüs.....	41
5.2 Ressursi kasutamise analüüs.....	43
5.3 Erinevad lõpptulemuste stsenaariumid .....	45
5.4 Lõpptulemust mõjutavad tegurid .....	48
5.5 Tagasiside protsessile .....	51
6 Tulemuste analüüs ja järeldused.....	53
6.1 Protsessikaeve projekti ülesehitus .....	53

6.2	Protsessimudeleid genereerivate algoritmide võrdlus .....	54
6.3	Lõpptulemust mõjutavate tegurite leidmine .....	55
6.4	Protsessikaeve projekti jaoks sobivaim tarkvara .....	56
7	Kokkuvõte .....	59
	Kasutatud kirjandus .....	60
	Lisa 1 – Protsessiga seotud dokumentitüüpide loend .....	62
	Lisa 2 – Logifaili atribuutide loend .....	63
	Lisa 3 – Alandmete ja filtreeritud logide graafilised esitused .....	65
	Lisa 4 – Peamiste sündmuste ja dokumentide esinemise sagedused.....	67
	Lisa 5 – Taotlussummade jaotus läbi aastate.....	69
	Lisa 6 – Disco poolt genereeritud protsessimudel.....	70
	Lisa 7 – Kõige suurema ajakuluga protsessiosad Disco järgi .....	73

## Jooniste loetelu

<b>Joonis 1.</b> Protsessikaeve seos andmete ja protsesside analüüsi valdkondade vahel. ....	14
<b>Joonis 2.</b> Andmete ja mudeli kasutamine erinevates BPM-i elutsükli osades. ....	14
<b>Joonis 3.</b> Protsessikaeve seos reaalse elu ja sündmuste logidega. ....	15
<b>Joonis 4.</b> Kolm viisi sündmuste logide ja protsessimudelite seostamiseks. ....	17
<b>Joonis 5.</b> XES metamudel. ....	19
<b>Joonis 6.</b> <i>Concept Drift</i> tulemus sündmuste paarile <i>begin payment-abort payment</i> . ....	24
<b>Joonis 7.</b> <i>Concept Drift</i> tulemus sündmuste paarile <i>decision-revoke decision</i> . ....	25
<b>Joonis 8.</b> Protsessimudel Celonisega. ....	30
<b>Joonis 9.</b> Kohandatud Celonise protsessimudel. ....	30
<b>Joonis 10.</b> $\alpha$ -algoritmiga genereeritud uuritava protsessi Petri võrk. ....	33
<b>Joonis 11.</b> Protsessimudeli Petri võrk kasutades heuristilist algoritmi. ....	34
<b>Joonis 12.</b> Protsessipuu kasutades induktiivset algoritmi. ....	35
<b>Joonis 13.</b> Protsessimudel kasutades induktiivset algoritmi. ....	36
<b>Joonis 14.</b> Sündmuste vahelised seosed hajusa algoritmi põhjal. ....	38
<b>Joonis 15.</b> Suure ajakuluga protsessiosad Celonise animatsioonis. ....	42
<b>Joonis 16.</b> Taotlusprotsessiga seotud algatajad arvestades töödeldavat mahtu. ....	44
<b>Joonis 17.</b> Osapoole DP-Z koormus uuritaval ajaperioodil. ....	44
<b>Joonis 18.</b> Seosed erinevate lõpptulemuste vahel. ....	46
<b>Joonis 19.</b> Sündmuste toimumine vastavalt ajatemplile. ....	65
<b>Joonis 20.</b> Filtreeritud logis sündmuste toimumine vastavalt ajatemplile. ....	65
<b>Joonis 21.</b> Logide järjestus taotlusprotsessi pikkust arvestades. ....	66
<b>Joonis 22.</b> Filtreeritud logide järjestus taotlusprotsessi pikkust arvestades. ....	66
<b>Joonis 23.</b> Taotlussummad aastatel 2015-2017. ....	69
<b>Joonis 24.</b> Kuni 100 000€ taotlussummade jaotus 2015-2017. ....	69
<b>Joonis 25.</b> Discoga genereeritud protsessimudel. ....	72
<b>Joonis 26.</b> Kõige suurema ajakuluga protsessiosad Disco järgi. ....	73



## Tabelite loetelu

<b>Tabel 1.</b> Protsessikaeve tarkvarade üldine võrdlus. ....	20
<b>Tabel 2.</b> Logifaili karakteristikute võrdlus enne ja pärast logide filtreerimist.....	26
<b>Tabel 3.</b> Läbitud sündmused Disco protsessimudeli põhjal. ....	31
<b>Tabel 4.</b> Suurima ajakuluga sündmused. ....	43
<b>Tabel 5.</b> Õigeaegselt alustatud ja viivituses maksete hulk aastate kaupa. ....	45
<b>Tabel 6.</b> Lõpptulemuste järgi grupeeritud taotlused aastate kaupa. ....	47
<b>Tabel 7.</b> Erinevate tunnuste statistilised testid. ....	49
<b>Tabel 8.</b> BPIC 2018 protsessis kasutatavad dokumentide tüübid selgitustega. ....	62
<b>Tabel 9.</b> BPIC 2018 logifailis kasutatud atribuudid selgitustega.....	63
<b>Tabel 10.</b> Olulisemates alamprotsessides sündmuste ja dokumentide sagedused. ....	67

# 1 Sissejuhatus

Organisatsiooni juhtimisel pööratakse järjest enam tähelepanu protsessipõhisele juhtimisele. See aga eeldab oskusi ning teadmisi protsesside mudeldamise ja analüüsimise kohta. Protsesside analüüsimisel võib olla mitmeid erinevaid eesmärke, näiteks kõrvalekallete ja mittevastavuste leidmine, protsessi või protsessiosa realisatsiooni mõjutavate tegurite uurimine, ressursside kaardistamine, selgem ülevaade protsessi käitumisest jms. Sageli on parimaks lähenemiseks analüüs läbi sündmuste logide ehk protsessikaeve.

Protsessikaeves on olulisel kohal protsessimudeli ja reaalsuse vahel võimalikult tugeva seose loomine ning sellest tingituna on väga oluline orienteeruda erinevates meetodikates ning tehnilistes lahendustes, kuna erinevad uurimiseesmärgid vajavad erinevaid lähenemisi. Omades ülevaadet protsessikaeve meetodikatest, on võimalik planeerida tööd vajalike algandmetega – sündmuste logidega.

## 1.1 Eesmärk

Käesoleva magistritööl on mitu eesmärki, mis moodustavad terviku. Peamiseks eesmärgiks on teostada võrdlus erinevate protsessimudelit genereerivate algoritmide vahel, arvestades valitud protsessikaeve tarkvarasid. Töö eesmärgiks ei ole genereerida kõiki võimalikke tüüpi protsessimudeleid, vaid ainult valitud uurimisteema analüüsiks sobivat mudelit. Samuti ei analüüsita kõikvõimalikke algoritme, vaid ainult neid, mida pakuvad kasutatavad tarkvarad. Konkreetsem uurimisteema on valitud selliselt, et see oleks pigem sagedasti esinev ning seetõttu on töö tulemused vaadeldavad ka laiemas kontekstis.

Teiseks, täiendavaks eesmärgiks on protsessikaeve projekti läbimine ning selle tulemusena vaadeldava äriprotsessi logide põhjal mõjutegurite leidmine soovimatute lõpptulemuste realiseerumisel. Töös kasutatakse Euroopa Põllumajanduse Tagatisfondi (EAGF) põllumajandustootjate otsetoetuste taotlemisprotsessi logisid Saksamaal 2015.–

2017. aasta kohta. EAGF loodi Euroopa Liidu (EL) poolt ühise põllumajanduspoliitika rahastamise eesmärgil 2005. aastal. [1]

Teema valiku üheks põhjuseks on autori praegune töö finantssektoris protsessijuhtimise valdkonnas, kuid protsessikaeve tehnikaid kasutamata. Protsessikaeve kasutamine loob täiendavaid võimalusi infosüsteemidesse talletatud logide põhjal saada teavet selle kohta, kuidas protsess tegelikult käitub ning kuidas läbi tagasisidestamise protsesse parendada.

Eesmärgile jõudmiseks püstitati järgmised uurimisküsimused:

- Millistest etappidest ja analüüsides ehitada üles protsessikaeve projekt?
- Millise algoritmiga genereeritud mudel annab protsessi edasiseks analüüsiks sobivaima tulemuse?
- Milline on tagasiside äriprotsessi omanikule antud uuritava probleemi osas?
- Milline kasutatavatest tarkvaradest toetab antud protsessikaeve projekti läbimist?

Antud töö erineb varasematest töödest selle poolest, et see ei keskendu puhtalt tarkvarade või algoritmide võrdlemisele, vaid töötab nende tarkvaradega, kasutades reaalseid logisid ning teostab selle põhjal võrdluse. Mudelite võrdluses tuginetakse valitud kvaliteedikriteeriumitele. Hinnang mudeli, tarkvara ja projekti etappide sobivuse kohta antakse läbitud protsessikaeve projekti vaatest.

## 1.2 Ülevaade tööst

Magistritöö esimeses, teoreetilises osas antakse üldine ülevaade protsessikaevest ning erinevatest protsessikaeve tehnikatest. Kuna sündmuste logid omavad olulist rolli protsessikaeves ning nende standardiseerimine aitab selle valdkonna efektiivsemale kasutamisele kaasa, siis on tutvustatud ka logide XES standardit [2]. Teooria osa lõpetuseks tutvustatakse lühidalt kasutatavaid protsessikaeve tarkvarasid.

Töö praktiline osa on jaotatud kolmeks peatükiks.

- **Protsessikaeve projekti ülesehitus ja algfaas.** Kõigepealt on vaatluse all protsessikaeve projekti etapid ja nende sisu. Seejärel viiakse läbi protsessimudeli koostamisele eelnevad tegevused, milleks on tutvumine alusandmete ja uuritava probleemiga, logide inspekteerimine ning protsessi stabiilsuse analüüs.

- **Protsessimudelite koostamine.** Teises ja peamises osas koostatakse ning testitakse protsessimudeleid, tuginedes erinevatele algoritmidele ja protsessikaeve tarkvaradele. Testimisel võetakse aluseks meetodikat tutvustavas alapeatükis välja valitud kvaliteedikriteeriume. Protsessimudelite koostamise juures tuuakse välja ka erinevate algoritmide tööpõhimõtte. Peatüki lõpus teostatakse kasutatud algoritmide võrdlev analüüs ning valitakse arvestades mudelite kvaliteet välja mudel või mudelid, mida kasutatakse protsessi analüüsis.
- **Protsessi analüüs.** Viimases praktilist osa hõlmavas peatükis analüüsitakse mudeldatud protsessi eesmärgiga tuvastada erinevaid tegureid, mis kutsuvad esile soovimatud lõpptulemused. Nende stsenaariumite analüüsimisel kasutatakse tulemuste olulisuse hindamisel hüpoteeside statistilist testimist.

Käesoleva töö lõpetuseks võetakse kokku saadud tulemused ning hinnatakse valitud mudelite, projekti etappide ja tarkvara sobivust konkreetsest protsessikaeve projektist lähtuvalt.

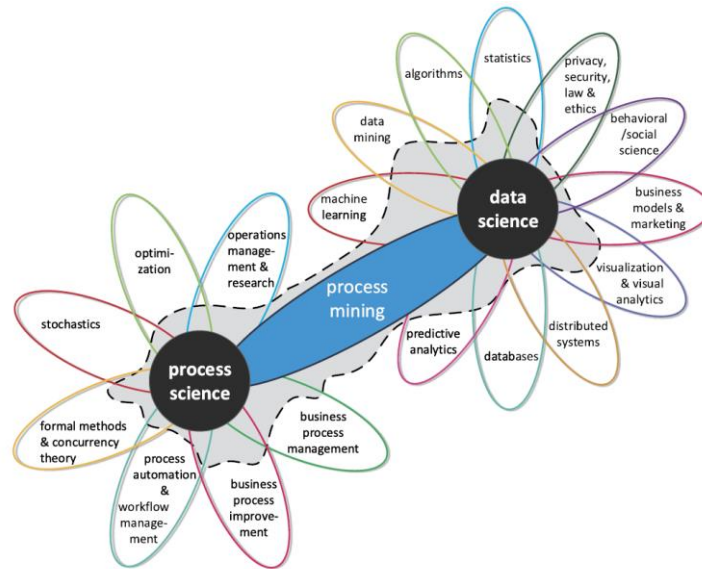
## 2 Üldine taust

Selle peatüki eesmärk on anda käesoleva töö jaoks vajalik ülevaade protsessikaeve põhiollemusest, klassifitseerimisest ning kasutatavatest tarkvaradest. Protsessikaeve aluseks on sündmuste logid, seega tehakse lühike tutvustus ka logikirjete struktuuri ning logifaili formaadi XES (*eXtensible Event Stream*) kohta.

### 2.1 Protsessikaeve

Seoses protsesside osatähtsuse suurenemisega on muutunud oluliseks, kuidas organisatsioon korraldab oma töid, haldab ressursse, milline on teostatavuse järjekord või kuidas hallata seda kõike rollipõhiselt. Võib juhtuda, et organisatsioonis on väga selgelt kirjeldatud töökorraldus ning eksisteerib seda toetav infosüsteem. Kuid isegi sellisel juhul võib tegelik tööviis märkimisväärselt erineda ettenähtust. Teistel juhtudel tööprotsesside kirjeldused puuduvad või on ajakohastamata ja puudulikud. Infot toimuvate protsesside kohta on võimalik koguda erinevatest infosüsteemidest, kusjuures eelduseks ei ole spetsiaalse töövoogu (*workflow*) infosüsteemi olemasolu. Protsessikaeve mõistet kasutataksegi olukorras, kus reaalse info põhjal (ehk kaevandamisel) saadakse infot protsessi kirjeldamiseks ja analüüsimiseks. [3]

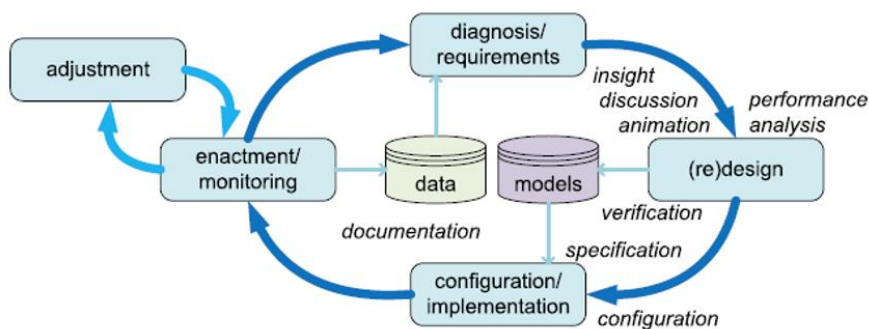
W. van der Aalst paigutab oma raamatus *Process Mining* [4] protsessikaeve ühelt poolt masinõppe ja ennustava analüüsi ning teiselt poolt protsesside modelleerimise ja analüüsi vahele. Seetõttu võibki protsessikaevet vaadelda kui ühendavat lüli andmete töötlemisega seotud valdkondade (*data science*) ja protsesse analüüsivate valdkondade vahel (*process science*). Protsessikaeve peamine idee on avastada, monitoorida ja täiustada tegelikke protsesse (st mitte eeldatavaid), kogudes infot sündmuste logidest.



**Joonis 1.** Protsessikaeve seos andmete ja protsesside analüüsi valdkondade vahel.

Protsessimodelite vajalikkus on erinevatele osapooltele, kes on mingi protsessiga seotud, juba üsna selge ja ei vaja sageli täiendavat selgitamist. Olgu selleks siis äri poolelt vajadus saada protsessist visuaalne ülevaade ning tagada sellega osapoolte vahel parem kommunikatsioon või IT poolelt verifitseerimise ja spetsifikatsiooni abivahend ning võimalus teostada läbi simuleerimise jõudlusteste. Protsessimodeli koostamist kui asja iseenesest ei tohi siiski protsessikaevetega segamini ajada.

Järgneval joonisel on näha, millist rolli mängivad BPM-i elutsükli protsessimodelid ja andmed. Protsessimodel on olulisel kohal ümberdisainimise, konfigureerimise ja rakendusetapis kui andmed on määrava tähtsusega õigusloome, monitoorimise ja nõuete etapis. [4]



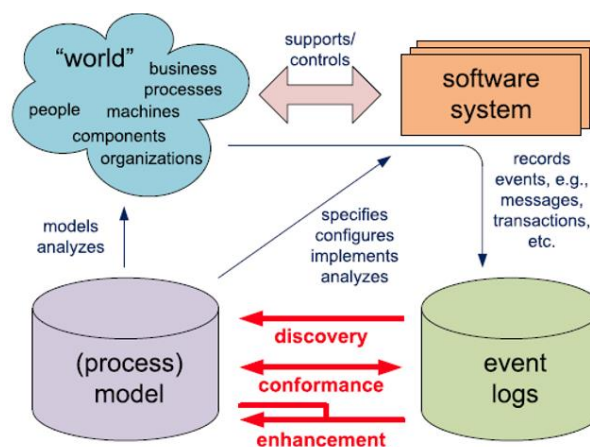
**Joonis 2.** Andmete ja mudeli kasutamine erinevates BPM-i elutsükli osades.

## 2.2 Protsessikaeve tüübid

Eristatakse kolme erinevat vaadet protsessikaevele: protsessi, organisatsiooniline ja juhtumi vaade. Protsessi vaade keskendub juhtimiskeskonnale, st tegevuste järjestamisele. Selle saavutamiseks on eesmärgiks leida kõigi võimalike teede piisavalt täpne kirjeldus, mida väljendatakse näiteks Petri võrguna või BPMN-s. Organisatsiooniline ülevaade keskendub sündmuste algatajale, st millised osapooled osalevad tegevuste läbiviimisel ja kuidas nad nendega seotud on. Eesmärgiks on organisatsiooni struktureerimine, inimeste liigitamine rollide ja organisatsiooniliste üksuste kaupa või konkreetse osapoole seoste näitamine (st sotsiaalsete võrgustike loomine). Juhtumi vaade keskendub juhtumi omadustele. Juhtumeid võib iseloomustada protsessi läbimise kaudu, juhtumi algatajate läbi või vastavate andmeelementide väärtustega. [3]

Ortogonaalne kolmevaateline lähenemine (protsess, organisatsioon ja juhtum) võib protsessi kaevandamisel tuua välja erinevaid loogilisi ja/või jõudluse probleeme. Kui fookuses on protsessi vaade, siis keskendutakse protsessimudeli loogilisele struktuurile ja/või jõudlusküsimustele. Organisatsioonilise vaate puhul võib rõhk olla rollidel või sotsiaalsel võrgustikul jne. Kui eesmärgiks ei ole piirata end kindla vaatega, siis võib osutada kasulikuks erinevate lähenemiste kasutamine. [3]

Joonis 3 illustreerib, kuidas protsessikaeve loob seose tegelike protsesside, nende andmete ning teiselt poolt protsessimudelite vahel. Samuti on joonisel märgitud ära peamised protsessikaeve tehnikate tüübid. [4]



Joonis 3. Protsessikaeve seos reaalse elu ja sündmuste logidega.

Eristatakse kolme peamist tüüpi protsessikaeve tehnikaid: avastav (*discovery*), vastavuse loomine (*conformance*) ja täiustamine (*enhancement*). [4]

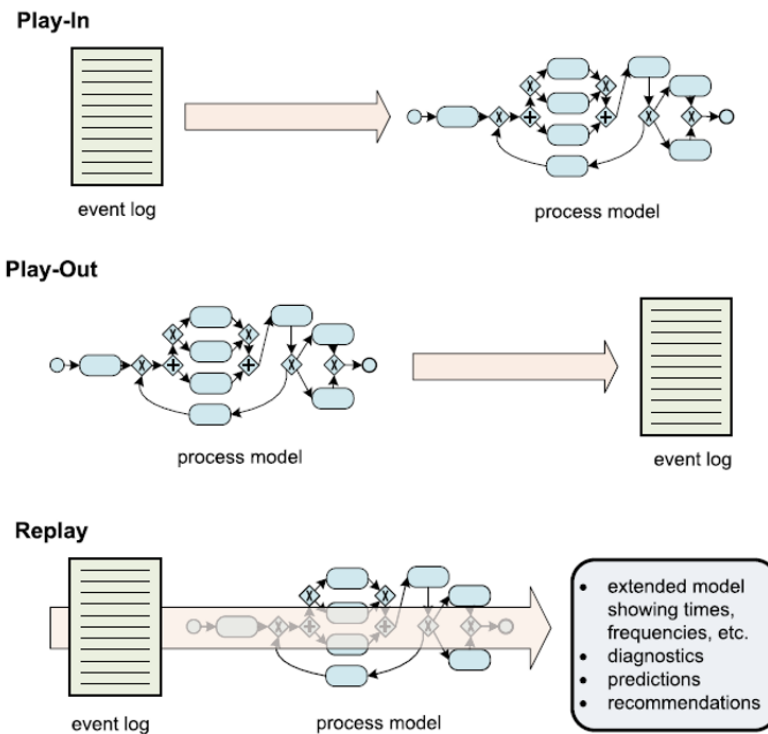
Avastava protsessikaeve tehnika puhul võetakse sündmuste logid ja luuakse mudel eelneva teabeta. Peamine näide on selleks  $\alpha$ -algoritm, kus piisava hulga logide olemasolu korral suudetakse genereerida Petri võrk täiendavat infot omamata. Kui sündmuste logi sisaldab ka teavet ressursside kohta, siis on avastamistehnika abil võimalik genereerida ka nt sotsiaalseid võrgustikke.

Teise võimaliku protsessikaeve tüübi puhul võrreldakse olemasolevat protsessimudelit sama protsessi sündmuste logiga. Vastavusse seadmisega saab kontrollida, kas reaalsus vastab salvestatud logidele ja vastupidi. Lisaks kontrollile, kas ärireegleid ka tegelikkuses rakendatakse, saab tuvastada ka riskiolukordi – näiteks üks töötaja saab teostada üksi liiga pika protsessiosa, võimalikud pettusekohad protsessis või kõrvalekalded logides.

Kolmandat tüüpi tehnika idee on laiendada või täiustada olemasolevat protsessimudelit, kasutades infot protsessi ja logide kohta. Kuna vastavuse kontrollimise käigus mõõdetakse mudeli ja reaalsuse vastavust, siis siin on peamiseks eesmärgiks mudeli muutmine või laiendamine. Üks võimalik muutmise eesmärk on mudeli parandamine, st et mudel kajastaks reaalselt protsessi paremini ja täpsemini. Mudeli laiendamine on täiendavate andmete lisamine. Nimelt, kui koostada äriprotsessile protsessijoonis, siis sageli kujutatakse lihtsalt juhtimisvoogu (*control-flow*), st tegevuste järjestusi. See on aga suhteliselt piiratud vaade äriprotsessile. Enamus modelleerimiskeeli võimaldavad lisada täiendavaid andmekihte: organisatsiooniline vaade, ressursid, andmete vaade, aja perspektiiv jne. Nende andmekihtide lisamiseks saabki infot sündmuste logisid analüüsides.

Protsessikaeve võtmekohaks on tugeva seose loomine protsessimudeli ja reaalsuse vahel, mida kajastavad sündmuste logid. W. van der Aalst kasutab oma raamatus logide ja protsessimudeli erinevate seoste kajastamisel mõisteid *Play-In*, *Play-Out* ja *Replay*. [4]





**Joonis 4.** Kolm viisi sündmuste logide ja protsessimudelite seostamiseks.

*Play-Out* kujutab endast protsessimudelite klassikalist kasutamist. Seda saab kasutada nii äriprotsesside analüüsimiseks kui ka jõustamiseks. Samuti liigitub simulatsioonide kasutamine sama olukorra alla.

*Play-In* on täpselt vastupidine *Play-Out* olukorrale – aluseks võetakse reaalne käitumismuster ning sellest genereeritakse mudel. Siia alla kuuluvad just avastava protsessikaeve tehnikad. Ka enamik andmekaeve tehnikatest järgivad *Play-In* seost, st näidete alusel kujundatakse mudel, kuid andmekaeve puhul ei ole tegemist protsessimudelite, vaid muudel kujudel olevate mudelitega, nagu otsustuspuud, assotsiatsioonireeglid jms. Ka käesolevas töös kasutatakse just avastavat protsessikaevet ning *Play-In* seost.

*Replay* kasutab sisendina nii protsessimudelit kui sündmuste logi. Tekkinud sündmuste logi „mängitakse“ uuesti läbi protsessimudelis. Selline tegevus võib toimuda erinevatel eesmärkidel: mudeli ja logide vastavuse määramiseks, mudeli laiendamiseks aja ja sageduse info, ennustavate mudelite loomiseks jms.

## 2.3 Sündmuste logid

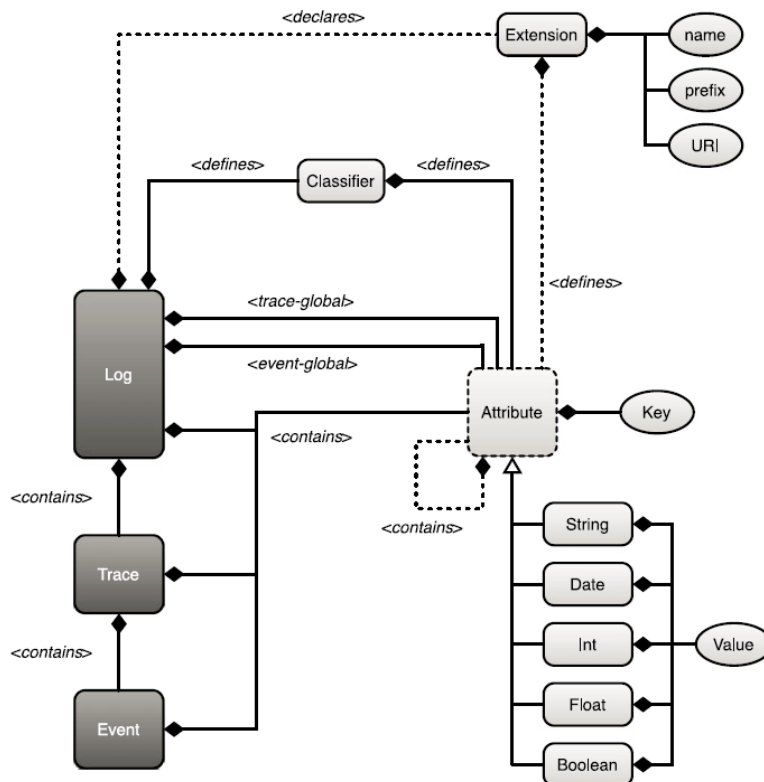
Protsessikaeve oluliseks osaks on sündmuste logid. Informatsiooni sündmuste ja nende toimumise kohta jääb erinevatesse infosüsteemidesse maha hulganisti, kuid erinevad süsteemid võivad seda teavet väga erineval kujul talletada.

XES on sündmuste logide XML-põhine standard [2], mis vahetas 2010. aastal välja Mxml (*Mining eXtensible Markup Language*). Selle standardi kehtestamise eesmärk on anda ühtne vorming sündmuste logiandmete vahetamiseks erinevate rakenduste vahel. Peamine vajadus standardiseerimise järele on protsessikaeve valdkond, kus nende sündmuste logide põhjal protsesse analüüsitakse. Kuid XES standardit saab kasutada ka üldises andmekaevanduses, tekstitöötluses ja statistilises analüüsis. [5]

XES standardi kujundamisel on järgitud järgmisi karakteristikuid [5], [6]:

- **Lihtsus.** XES standardi logisid peaks olema lihtne genereerida ja analüüsida ning need peaksid olema võrdselt hästi loetavad ka inimesele.
- **Paindlikkus.** XES standard peaks suutma salvestada sündmuste logisid mistahes rakenduses.
- **Laiendatavus.** Tulevikus peaks saama standardit kergelt ja läbipaistvalt laiendada, säilitades võimekuse ühilduvusele. Samamoodi peab olema võimalik laiendada erinõuete standardit, nt konkreetsete valdkondade või rakenduste jaoks.
- **Ekspressiivsus.** Kogu logide andmestik on vaatamata standardsele formaadile kasutaja poolt kergesti loetav.

Protsessis saab toimunud protsessisammudele ehk sündmustele omistada infot selle kohta, mis tegevus toimus, kuna see sündmus toimus, kelle poolt teostati, kuidas, kus jne. Sündmustes osalev objekt ja sündmused läbi protsessi moodustavad logijälje (*trace*). Üks logijälg sisaldab objekti ning sündmuste atribuute. Objekt omab alati ühte identifitseerivat atribuuti (nimi, ID vms), kuid sageli veel ka täiendavat infot. Sündmused, mis on selle objektiga seotud, omavad alati tegevuse (*action*) ja ajatempli (*timestamp*) atribuute. Kõik muud atribuudid (kus, kes, millal algas või lõppes jne) on täiendavad atribuudid. [7] Joonisel 5 on kujutatud XES standardi metamudel, mis näitab kuidas erinevad logi komponendid ja atribuudid moodustavad terviku. [4]



Joonis 5. XES metamudel.

Lisaks identifitseerivatele atribuutidele (logijälje puhul juhtumiga seotud ning sündmuste puhul tegevusega seotud väärtus) ja sündmuste ajajälgedele lisatakse logisse sageli ka järgmisi täiendavaid atribuute:

- elutsükliga seotud atribuut sündmuste jaoks (nt lõpetatud, katkestatud jne),
- organisatsiooniga seotud atribuut (roll, grupp, ressurss jne)
- täiendavad ajaga seotud atribuudid (sündmuse algusaeg, salvestamise aeg jne) jne.

Korrektne sündmuste logide salvestamine, kogumine ning kättesaadavus analüüsi tarbeks on olulise tähtsusega protsessikaeve jaoks. Lisaks protsessikaevele on kvaliteetsetel logidel ka täiendav väärtus, näiteks auditeerimine. Kvaliteedi üks peamisi mõõdikuid on siin mittemanipuleeritavus. [4]

## 2.4 Protsessikaeve tarkvarad

Protsessikaeve tarkvarad on võimelised looma logidest protsessimudeli või andmevoo-keemi. Samuti pakuvad sellele valdkonnale spetsialiseerunud tarkvarad erinevaid lahendusi protsessikaeve analüüside teostamiseks, protsesside visualiseerimiseks jms. Käesolevas töös kasutatakse protsessimudelite koostamisel, logide ja protsessi analüüsis

kolme tarkvara, milleks on avatud koodiga ProM [8] ning valmistarkvarad Disco [9] ja Celonis [10]. Antud tarkvarad on valitud lähtuvalt kättesaadavusest akadeemiliseks tööks. Tarkvarade peamised parameetrid on välja toodud võrdlevas tabelis 1 [11].

**Tabel 1.** Protsessikaeve tarkvarade üldine võrdlus.

<b>Informatsioon</b>	<b>ProM</b>	<b>Disco</b>	<b>Celonis</b>
Organisatsioon	The Process Mining Group, Eindhoven Technical University	Fluxicon	Celonis
Meeskond	W. van der Aalst ja tema meeskond	Dr. A. Rozinat, Dr. C.W. Gunther	Bastioin, Nominacha, M. Klenk, A. Rinke
Riik	Holland	Holland	Saksamaa
Litsents	Avatud koodiga	Kommerts	Kommerts
Versioon	Prom 6.7	<i>Enterprise Edition</i>	<i>Enterprise Edition</i>
Platvormi toetus	<i>Desktop Version</i>	<i>Desktop Version</i>	<i>Software as service versoin</i>
Protsessimudelite väljundid	BPMN, Work-Flow, Petri võrk, Heuristiline mudell jne.	Hajus mudel	Hajus mudel, graafikud
Andmete filtreerimine	Jah	Jah	Jah
Avastav protsessikaeve	Jah	Jah	Jah
Vastavuse kontroll	Jah	Ei	Ei
Sotsiaalvõrgustik	Jah	Ei	Ei
Otsustusreeglid	Jah	Ei	Ei
Protsessi visualiseerimine	Jah	Jah	Jah
Tootlikkuse raport	Jah	Jah	Jah
Imporditav failitüüp	csv, mxml, xes	mxml, xes, fxl, dsc	xes, csv, xls

Kui suuresti on kõigi nende tarkvarade eesmärk sama, siis oma ülesehituselt erineb ProM teistest nimetatutest. Kui Disco ja Celonis on valmis pakendatud, siis ProM on avatud, mis tähendab seda, et sinna saavad kasutajad ise nii endale kui teistele kasutamiseks analüüsipakette koostada.

## 3 Protsessikaeve projekti ülesehitus ja algfaas

Käesolevas peatükis on kesksel kohal protsessikaeve projekti võimaliku ülesehituse kaardistamine ning protsessimudeli koostamisele eelnevate tegevuste ja analüüside teostamine.

### 3.1 Protsessikaeve projekti etapid

Kuus sigmat (*Six Sigma*) on meetodite ja vahendite kogum protsessi täiustamiseks, mis kasutab statistilist analüüsi probleemide kindlaksmääramisel ja parendustegevuste elluviimisel. Tüüpiline kuue sigma projekt järgib järgmist nn DMAIC tegevuskava [4]:

- *define* – määratle probleem ja määra eesmärk;
- *measure* – hinda võtmeindikaatoreid ja kogu andmeid;
- *analyze* – analüüsi andmeid põhjuste ja tagajärgede seoste uurimiseks ja kontrollimiseks;
- *improve* – paranda praegust protsessi tehtud analüüsi põhjal;
- *control* – kontrolli protsessi kõrvalekallete minimeerimiseks.

Ettevõtetes rakendatavad kuue sigma tehnikad võivad olla üles ehitatud ainult andmete vaatest, mis ei ole aga piisav. Protsessikaeve tehnikate kasutamine võimaldab anda sellele metoodikale täiendavat sisu. [4]

Käesolevas töös läbiviidava protsessikaeve projekti aluseks võetakse loetletud kuue sigma projekti etapid. Erinevaid etappe sisustatakse konkreetsete protsessikaevaga seotud tegevustega. Need tegevused on kaardistatud erinevate soovitude [12], [13], [14] põhjal kirjandusest ning läbi eksperimenteerimise kujundatakse protsessikaeve projekti struktuur, mis oli antud töö jaoks toimiv. Saadud raamistik tuuakse välja tulemuste ja järelduste peatükis.

Kuue sigma projekti etappidest vajab lahti kirjutamist eeskätt kolmas, analüüsi etapp. Teised nimetatud etapid on oma sisu poolest pigem universaalsed ja üheselt mõistetavad.

Analüüside osas on soovitatud järgmisi tegevusi:

- protsessi ajas muutumise analüüs;
- logide inspekteerimine ja eelanalüüs (sh vajadusel filtreerimine ja klasterdamine);
- protsessimudeli koostamine;
- ajakulu ja jõudluse analüüs (nn pudelikaelad);
- ressursi analüüs (organisatsioonilisest vaatest).

Kui sissejuhatus tõi välja käesoleva töö eesmärgi ja ülesandepüstituse, siis järgnevas alapeatükis tutvustatakse põhjalikumalt logide päritolu, reaalselt sisu ning sellisel määral ka uuritavat probleemi, mis on vajalik sobiva protsessimudeli tüübi määramiseks. Mudel, millega jätkatakse protsessi analüüsi, selgitatakse välja protsessimudelite koostamise ja testimise järgselt vastavas võrdlevas analüüsis. Andmetega tutvumisele järgneb logide inspekteerimine, filtreerimine ja esmane analüüs. Protsessimudeli koostamine, testimine ning protsessi analüüs on teostatud järgnevates peatükkides.

### **3.2 Analüüsitavad andmed**

Töös on logideks kasutatud *Business Process Intelligence Challenge 2018* (BPIC) andmeid [15]. Tegemist on kaheksandat aastat toimuva väljakutsega, kus pakutakse osalejatele reaalses toimunud sündmuste logisid ning neid andmeid analüüsides tuleb leida vastuseid ja lahendusi konkreetsetele äriprotsesse puudutavatele küsimustele. Lisaks sellele, et logid on vabalt kättesaadavad, on heaks eeliseks selle allika kasutamisel teadmine, et logid on pärit reaalsest äriprotsessist, andmehulk on piisavalt suur põhjalikemateks analüüsideks ning logifailis olevad kirjed on piiritletud mingi kindla protsessiga. Käesolevas töös kasutatakse küll BPIC 2018 alusandmeid, kuid mitte konkreetset väljapakutud uurimistemasid, kuna see oleks väljunud antud töö raamidest.

EL kulutab suure osa oma eelarvest ühisele põllumajanduspoliitikale. Nende kulutuste hulka kuuluvad ka põllumajanduse otsetoetused. Ülejäänud põllumajandusvaldkonna eelarve kulutatakse turuga seotud kuludeks ja maaelu arendamiseks. Nende vahendite jaotamisega seotud protsessid kuuluvad EL-i ja siseriiklike õigusaktide reguleerimisalasse. Liikmesriigid peavad käitama integreeritud haldus- ja kontrollisüsteemi, mis hõlmab IT-süsteeme toetuste jagamise protsesside toetamiseks. [15]

BPIC 2018. aasta väljakutse puhul ongi alusandmeteks EL-i otsetoetuste taotluste menetluste logid Saksamaa farmeritele EAGF-st 2015.–2017. aasta jooksul. Andmed pärinevad Saksamaa IT-ettevõttelt *data experts*, kelle üheks tegevuseks on muuhulgas spetsiifiliste tarkvaralahenduste loomine. Tegemist on ettevõttega, kelle loodud lahendused töötavad väga erinevates valdkondades – tervishoiust tööstuse ja avaliku halduseni. Nad on välja töötanud põllumajandusministeeriumile ja selle allüksustele lahenduse põllumajandustoetuste haldamiseks, alustades taotlemisest ja kui kõik läheb hästi, lõpetades makse lubamisega. [16]

Antud taotlusprotsessis registreeritakse mitmeid erinevaid dokumenditüüpe, mis on mingite alamprotsessidega seotud. Dokumentide loomise, redigeerimise ja kinnitamise info logides annab infot toimunud sündmuste järjestuse kohta. Dokumenditüüpide loetelu ning seos konkreetse alamprotsessiga on ära toodud antud töö lisas [Lisa 1]. Tabelis on märgitud kolme erinevat tüüpi maatükkidega seotud dokumente, mis on järjestikustel aastatel asendatud. Edasistes analüüsides, kus on võimalik ja mõistlik, on antud dokumentide nimed omavahel võrdsustatud ning kasutatud nimetus *Geo parcel document*.

Lähtuvalt logide XES formaadist on teada, et taotlusel ja sündmustel on kirjeldatud erinevad atribuudid. Vastavate atribuutide loetelu on välja toodud töö lisas [Lisa 2]. Kui sama isik esitab taotluse erinevatel aastatel, siis käsitletakse neid erinevate taotlustena.

Tavaliselt avatakse taotlusvoor aprillis-mais ning aasta lõpuks peaksid toetused olema välja makstud. Siiski on igal aastal juhtumeid, kus seda ei saavutata. Soovimatuteks tulemusteks on makse hiline mine ja menetluse taasavamine. Makse on õigeaegne, kui aasta lõpuks saavutatakse makseprotsessi algatamine selliselt, et sellele ei järgne enam makse katkestamist. Teine ebasoovitav tulemus on taotluse uuesti läbivaatamine (*reopened*), mille võib algatada mingil põhjusel menetlev allüksus (alamprotsess *Change*) või vaidlustada taotluse esitanud isik (alamprotsess *Objection*). Taotluse uuesti läbi vaatamine võib kaasa tuua ka täiendavaid makseid. Protsessi analüüsi osas püütaksegi leida, millised tegurid mõjutavad kirjeldatud soovimatute lõpptulemuste teket.

Protsessimudeli koostamisel on oluline valida algoritm, mis tagaks seatud uurimiseesmärgi täitmise. Algoritmi valimisel tuleb arvestada, kas tegemist on reaalse logidega (st võib esineda palju müra), kas soovitakse keskenduda juhtimisvoole, ressurssidele ja

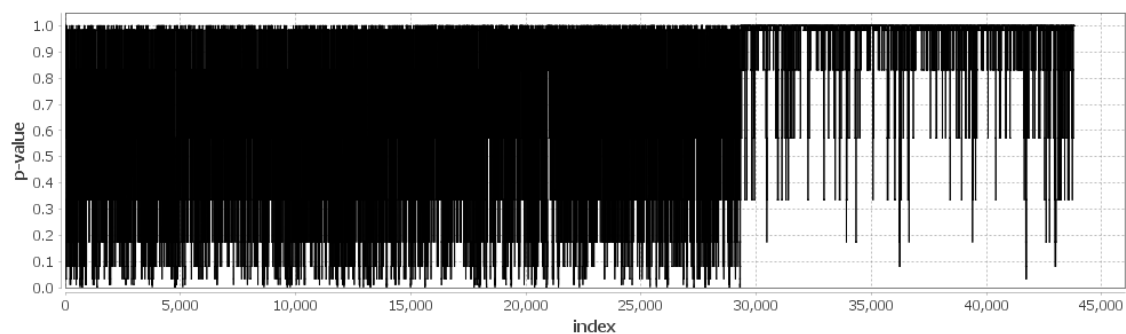
jõudlusele, sündmuste algatajatele (organisatoorne lähenemine) jne. Käesolevas töös keskendutaksegi juhtimisvoole ja protsessimudeli koostamisel on eesmärgiks saada mudel, mis toetaks huvipakkuvate kitsaskohtade uurimist. Muude vaadete loomise eesmärgiks protsessi analüüsifaasis on leida võimalikke mõjutegureid mitteõigeaegsete maksete ja taasavatud taotluste tekkimiseks.

### 3.3 Protsessi stabiilsuse kontroll

Paratamatult peavad protsessid ajas muutuma, et kohaneda muutuvate oludega. Muudatuste algatajateks võivad olla muutused õigusruumis, ärinõuetes või -loogikas, ümbritsevas keskkonnas vms. Kuigi paindlikkuse teema on BPM valdkonnas hästi uuritud, eeldavad tänapäevased protsessikaeve meetodid tasakaalus protsessi. Teisisõnu, kui sündmuste logide põhjal protsessimudelit koostatakse ja analüüsitakse, siis eeldusega, et sõltumata sellest, et läbitakse samu sündmusi, protsess ise ei ole muutunud. Selleks, et logide põhjal saadud tulemused ei oleks moonutatud, on oluline uurida sellise täiendava dünaamika ehk kõrvalekallete (*Concept drift*) võimalikku esinemist. [17], [18]

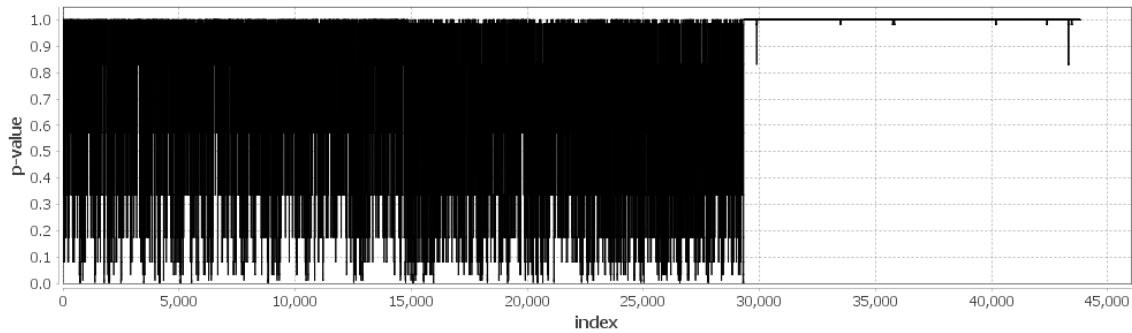
Kõrvalekallete analüüsis on peamiseks väljakutseks tuvastada, kas protsessimuutus on üldse toimunud. Kui muutus on aset leidnud, siis teise sammuna tuleb kindlaks teha ajavahemikud, millal see muutus toimus. Järgnevalt tuleks juba analüüsida muudatust ennast ning millised protsessiosad said sellest mõjutatud. Muudatuse olemuse avastamine on keerukas probleem, mis hõlmab endast nii muudatuse perspektiivi tuvastamist (andmed, ressursid jne) kui ka muutust ise. [17]

Analüüs on teostatud ProM-i paketiga *Concept Drift*. Analüüsides logifaili, leidis kaks sündmust, mis andsid tulemuseks kõrvalekalde: *revoke decision* ja *abort payment*.



Joonis 6. *Concept Drift* tulemus sündmuste paarile *begin payment-abort payment*.





**Joonis 7.** *Concept Drift* tulemus sündmuste paarile *decision-revoke decision*.

Mõlemal juhul on tegemist taotluse suhtes ebasoovitava tulemusega (makse või otsuse tühistamine) ja kuigi logifailide avaldamise ajaks oleks pidanud ka 2017. aasta taotluste läbivaatamise protsess olema lõppenud, siis ilmselt on tegemist sündmustega, mis lisanduvad mittelõpetatud taotluste menetlemisega 2018. aastal. Kokkuvõttes, kui viimase aasta üle aja minevad taotlused kõrvale jätta, siis protsessis muutusi ei tuvastatud, ehk uuritav protsess on stabiilne ning antud logisid võib täies ulatuses kasutada protsessimudeli koostamisel.

### 3.4 Logide analüüs

Pärast andmete importi ning veendumist, et protsess ei ole ajas muutunud, tuleks inspekteerida logide andmeid. Uute andmetega töötamisel kaardistatakse, milline on logifaili struktuur, sh millised on selles esinevad atribuudid, millised võimalikud sündmused protsessis esinevad jne. Ka käesoleval juhul selline ülevaatus sooritati. Mõned üldisemad tulemused on esitatud tabelis 2 koos filtreeritud logide andmega.

Logifaili filtreerimine on oluline ProM-ga töötamisel. Paljude ProM-i protsessikaeve pakettide puhul on eeltingimuseks, et protsessi algus ja lõpp oleksid fikseeritud ühe kindla sündmusega. Vastava paljususe ilmnemisel tuleb logisid filtreerida. Antud algandmete juures näitas ProM protsessi mõlemas otsas üle ühe võimaliku sündmuse, seega edasiste analüüside jaoks ProM-ga tuleb logid filtreerida. Celonise ja Discoga töötamisel sellist täiendavat eeltööd ei ole vaja teha. Küll on aga mõlemal tarkvaral olemas eraldi vaade logide inspekteerimiseks ja filtreerimiseks.

ProM-s on logide filtreerimisel kasutatud paketti *Filter Log using Simple Heuristics* eesmärgiga fikseerida sündmuste hulgast välja üks taotlusprotsessi alustav ja lõpetav

sündmus ning ainult enim kasutatud vahepealsed sündmused, lugedes sellega kõik teised sündmused ebaolulisteks. [12], [13]

Ülekaalukalt (88%) on taotlusprotsessi alguseks sündmus *mail income* ning lõpp-sündmuseks *finish payment* (79%). Vahepealseteks sündmusteks, mida hiljem mudelites kajastatakse, on valitud samad sündmused, mis on Celonise mudelil vaikimisi.

**Tabel 2.** Logifaili karakteristikute võrdlus enne ja pärast logide filtreerimist.

Karakteristik	Algandmed	Filtreeritud logid
Taotluste arv	43 809	30 941
Sündmuste arv	2 514 266	1 544 952
Sündmuste klassid	41	13
Seotud isikud	165	163
Minimaalne sündmuste arv	24	22
Maksimaalne sündmuste arv	2 973	898
Keskmine sündmuste arv	57	50

Teiseks logide eelanalüüsi paketi on ProM-s *Log to Model Explorer*, mis sisaldab endas klasterdamise, filtreerimise ning ümbernimetamise funktsionaalsust. Antud andmestiku puhul ei andnud selle paketiga töötamine täiendavat efekti ning mudelite sisendiks jäi eelnevalt filtreeritud logifail.

Pärast filtreerimist on hea võrrelda logisid ka graafiliselt. Visualiseerimiseks on ProM-s pakett *Project Log on DottedChart* [14]. Algandmete graafikul on näha sündmuste hajumist – nii erinevat tüüpi sündmuste esinemisel kui ka taotlusprotsessi venimisel üle aja (milleks on taotluse esitamise aasta lõpp). Hajumine toimub küll vähesel määral, kuid kokkuvõttes on arvestav hulk taotlusi, mis ületavad tähtaja kordades. Mõlema logide faili jaoks on genereeritud kaks erinevat vaadet, kujutades graafiliselt seoseid taotluste ja ajatempli ning taotlusprotsessi kestuse vahel. Kõik nimetatud graafikud on esitatud käesoleva töö lisa [Lisa 3].

Sündmuste ja dokumentide sagedusi uurides selgus, et kõige sagedasem sündmus on *calculate*, mida läbitakse kokku 466 141 korda (18,54%). Erinevaid dokumente lisatakse iga taotluse kohta umbes kaks, kusjuures pooled neist on taotluse maksekorraldusega seotud. Makseid algatatakse 1,75 korda rohkem, kui neid lõpetatakse. See näitab, et maksete katkestamiste osakaal on üsna suur. Samas üsna suurele hulgale taotlustest (99,8%) tehakse makse, mis moodustab 99,7% sellest summast, mida sooviti.

Olulisemate alamprotsessidega seotud sündmuste esinemise sagedus koos seotud dokumentidega on välja toodud töö lisas [Lisa 4].

Uuritav andmestik hõlmab kolme järjestikuse aasta taotluste infot. Kuna taotleja on markeeritud läbi logide sama identifikaatoriga, siis on võimalik välja filtreerida korduvtaotlejad ning analüüsida nende taotluste tulemuste muutumist ajas. Järjestikustel aastatel taotlusi esitanud isikuid on 96%. Kolme aasta taotlussummasid kõrvutades ilmneb, et üle 99% taotlustest jäävad alla 170 000€ ning alla 1% (umbes 100 taotlust aastas) taotlevad summat vahemikust 575 000€-610 000€. Graafikud taotluste jagunemise kohta taotlussummade järgi läbi aastate on toodud välja töö lisas [Lisa 5].

Logide inspekteerimisel ilmnas, et leidis neli taotlust, mille esimese sündmuse ajatempli kuupäev oli aasta võrra varasem ning ei olnud kooskõlas ka väljal *year* oleva infoga. Ilmselt on tegemist manuaalse sisestamise veaga. Teisi selliseid anomaalseid olukordi ei tuvastatud. Kuna protsessimudelit ei ole veel koostatud ning analüüsitud, siis praeguses logide inspekteerimise faasis on tegemist esialgse analüüsiga, mis annab analüüsi teostajale laiemat vaadet andmestikust. Edasiste analüüside käigus tekkivate erinevate küsimuste, probleemide ja hüpoteeside korral tuleb kindlasti tulla tagasi logide tasandile.

## 4 Protsessimudelite koostamine

Selles peatükis keskendutakse protsessimudelite loomisele ja testimisele, kasutades valitud modelleerimise algoritme, protsessikaeve tarkvarasid ning kvaliteedikriteeriume. Järgnevates alapeatükkides tutvustatakse erinevate algoritmide üldist tööpõhimõtet, koostatakse ja testitakse protsessimudeleid ning teostatakse võrdlev analüüs. Võrdlevas analüüsis leitakse piisava kvaliteediga protsessimudel või mudelid, mis võetakse aluseks protsessi analüüsi faasis. Valitud mudeli või mudelite sobivust antud protsessikaeve projekti suhtes hinnatakse käesoleva töö tulemuste analüüsi ja järelduste peatükis.

### 4.1 Metoodika

Algoritmide testimiseks on erinevaid metoodikaid. Üheks võimaluseks on testida, kuidas tulevad algoritmid toime erineva keerukusega mudelite loomisega. Sellist lähenemist tutvustasid P. Weber jt oma artiklis *A Framework for Comparing Process Mining Algorithms* [19]. Nad kasutasid oma uurimuses erineva keerukusega mudeleid, millest genereeriti läbi simulatsiooni logifail ning seejärel rakendasid kaevandamisel erinevaid algoritme.

Käesolevas töös kasutatakse reaalseid logisid, mis sisaldavad müra jm võimalikke kõrvalekaldeid, ning analüüsi alustades ei ole teada, milline on tegelik protsessimudel. Teisisõnu on proovitud käesolevas töös tekitada võimalikult realistlik olukord, kus läbi eksperimenteerimise leitakse erinevad protsessimudelid, mida testida.

Omaette väljakutseks on mudeli usaldusväärsuse testimise kriteeriumite valimine. Tuleb arvestada olukorraga, et erinevad algoritmid annavad erinevaid väljundeid. Osad protsessimudelid annavad tulemuseks Petri võrgu, protsessipuu või BPMN mudeli, millele saab kehtestada teistsugused kriteeriumid kui võimalikke järgnevasi väljendavale protsessigraafile.

Mudeli ja logide võrdluseks on kirjanduses kasutatud järgmisi kvaliteedimõõdikuid [20]:

- **Sobivus** (*fitness*). Mudelil on täiuslik sobivus, kui iga logikirje suudab mudeli perfektselt läbida. Sageli väljendatakse sobivust suhtarvuna puhtalt mudeli läbinud logide arv jagatuna logide koguarvuga (0..1).
- **Täpsus** (*precision*). Mudel on täpne, kui ei eksisteeri ülemäärast sobivust. Teisisõnu, mudel on ebatäpne, kui lubab selgelt rohkem käitumist, mis erineb sündmuste logis olevast.
- **Üldistamine** (*generalization*). Üldjuhul ei tohiks protsessimudel piirata võimalikku käitumist ainult konkreetsetes logides esinevaga, st mudel peab olema protsessi jaoks piisava üldistuse astmega.
- **Lihtsus** (*simplicity*). Lihtsaim protsessimudel, mis järgib sündmuste logi.

Petri võrgu usaldusväärsuse (*soundness*) kontrollimiseks saab sobivuse ja täpsuse kriteeriume täpsustada järgmiselt [21]:

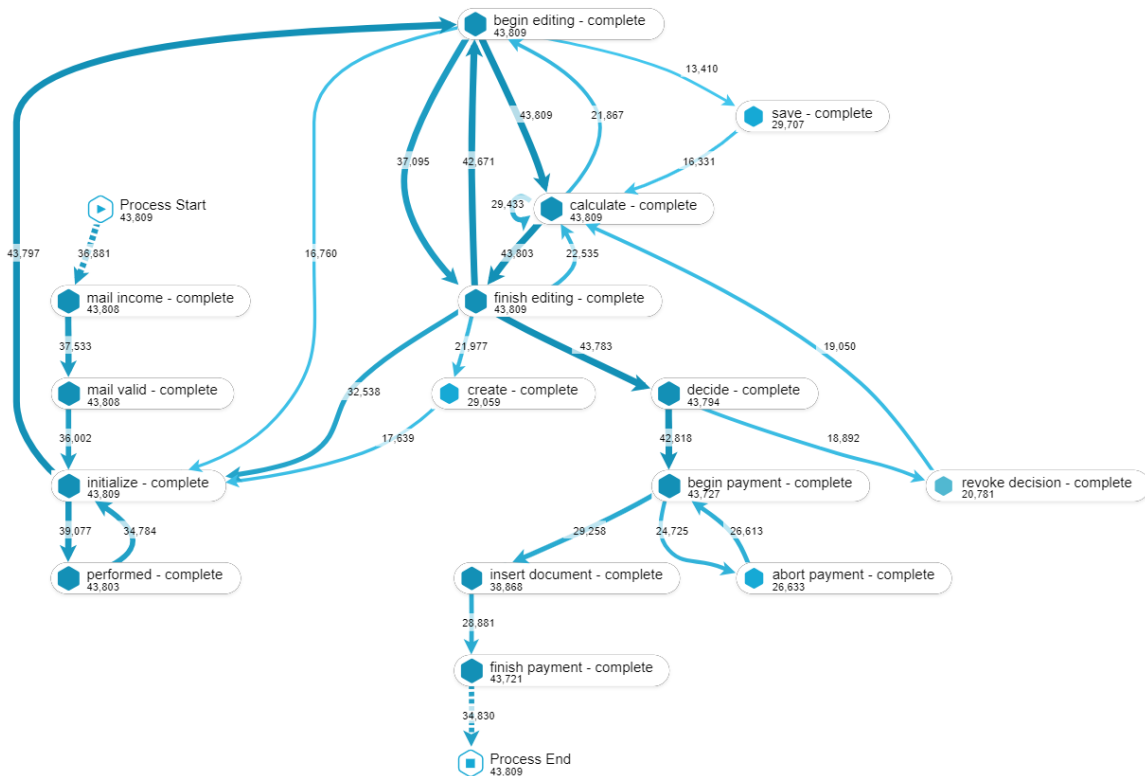
- **Mudeli korrektse läbimise test**. Kontrollitakse, kas mudelit on võimalik korrektselt läbida. Testahelaks valitakse sagedaseim *happy end* sündmuste järgnevus.
- **Mitteloopivate transaktsioonide test**. Kontrollitakse, et ei esineks sündmusi, millest ei saa liikuda lõppsündmuseni.

Teiste mitterangete protsessijooniste puhul hinnatakse täpsust tarkvaras kuvatava detailsusastmega (kui suur osakaal sündmustest ja seostest on kuvatud). Muude kriteeriumite osas tuleb anda mudelile subjektiivne hinnang.

## 4.2 Celonise ja Disco poolt genereeritud protsessimudelid

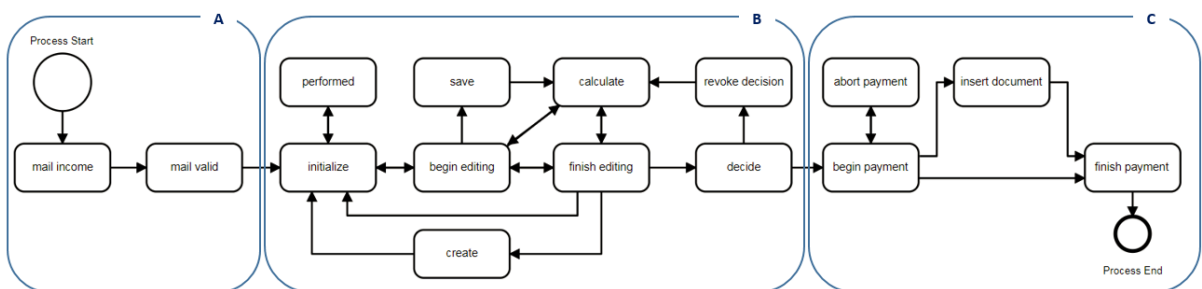
Sündmuste võimaliku järjestuse kohta saab kiire ülevaate Celonise või Disco tarkvara kasutades. Vaikimisi genereeritakse mudel, mis koosneb vähemalt 80% esinenud sündmustest. Nii kuvatavate sündmuste kui seoste hulka saab manuaalselt muuta. Mõlemad tarkvarad kasutavad mudeli genereerimisel hajusat algoritmi, mida tutvustatakse lähemalt käesoleva töö alapeatükis 4.6. Selles alapeatükis on täiendav fookus protsessist üldisema ettekujutuse saamisel ning esmase protsessimudeli koostamisel. Mudelite ja algoritmide võrdlevas analüüsis vaadeldakse ka selle punkti mudeleid konkreetse algoritmi vaatest.

Joonisel 8 kujutatud Celonise protsessimudelil on sündmuste ja seoste juurde lisatud taotluste arv (antud joonisel kordusi ei ole arvestatud). Nende näitajate asemel on võimalik kuvada ka absoluutset sagedust või kulunud aega (min, max, keskmine).



**Joonis 8.** Protsessimudel Celonisega.

Mudel sisaldab mitmeid hargnemisi ning tsükleid. Alusandmeid tutvustavas osas ning logide analüüsis on välja toodud, et alamprotsessidega on seotud erinevad dokumentide tüübid (vt ka Lisa1). Celonise jooniselt need erinevad alamprotsessid välja ei tule. Edasiste analüüside tarbeks on antud mudelit visuaalselt kohandatud (vt joonis 9).



**Joonis 9.** Kohandatud Celonise protsessimudel.

Arvestades sündmuste olulisust, on protsessisammude pealiin joondatud horisontaalselt. Ainult viimase sammu puhul (*begin payment* → *insert document* → *finish payment*) on

dokumentide lisamine sagedasem, kui selle sammu vahelejätmine. Samuti on edasise analüüsi huvides grupeeritud protsess kolmeks protsessiosaks A, B ja C.

Analüüsidis Disco poolt genereeritud protsessimudelit, mis on oma mõõtmete tõttu lisatud käesoleva töö lisasse [Lisa 6], selgus, et protsessiosa B läbitakse tõesti seoses erinevate dokumentidega korduvalt. Disco protsessimudeli põhjal läbitud sündmused ja seotud dokumendid on välja toodud järgmises tabelis.

**Tabel 3.** Läbitud sündmused Disco protsessimudeli põhjal.

<b>Protsessiosa (tsükkel)</b>	<b>Sündmuse lühend</b>	<b>Sündmus</b>	<b>Seotud dokumendi tüüp</b>
A	MI MV	<i>mail income</i> <i>mail valid</i>	<i>Payment Application</i>
B(1)	In BE FE Cr	<i>initialize</i> <i>begin editing</i> <i>finish editing</i> <i>create</i>	<i>Geo Parcel Document</i>
B(2)	In BE FE	<i>initialize</i> <i>begin editing</i> <i>finish editing</i>	<i>Control Summary</i>
B(3)	In Pf	<i>initialize</i> <i>performed</i>	<i>Reference Aligment</i>
B(4)	In BE Ca FE De RD	<i>initialize</i> <i>begin editing</i> <i>calculate</i> <i>finish editing</i> <i>decide</i> <i>revoke decision</i>	<i>Payment Application</i>
C	BP AP ID FP	<i>begin payment</i> <i>abort payment</i> <i>insert document</i> <i>finish payment</i>	<i>Payment Application</i>

Kõrvutades antud tabelis olevaid dokumentide andmeid lisas 1 toodud andmetega, on näha, et dokumendi tüüp *Entitlement Application*, mis BPIC kirjelduses on seotud alamprotsessiga *Main*, ei ole tabelis 3 esindatud. Suurendades Disco protsessijoonisel detailsusastet, selgub, et sündmused, kus nimetatud dokumenti on käsitletud, asuvad tsükelite B(3) ja B(4) vahel ning läbitakse tabelis toodud sündmusi (*In-BE-Ca-FE-De*). Kui pöörduda tagasi logide tasandile, siis selgub, et antud dokumente esitatakse vaidlustuse korral ning võrreldes taotluse menetlemisega alles järgmisel kalendriaastal.

### Mudeli usaldusväarsuse testimine.

- Kuna nii Disco kui Celonise mudelil on kasutatud vaikimisi seadistusi, siis saab öelda, et sobivus on 0,8.
- Metoodika osas välja toodud täpsuse kriteeriume ei saa selliselt siin kasutada, kuna nende mudelite puhul on tegemist nn mitterangete mudelitega, kus ei ole fikseeritud sündmustevaheliste seoste iseloom, vaid lihtsalt leiduvad seosed.
- Lihtsuse osas saab öelda, et tegemist on keskmise keerukusastmega mudelitega. Mudelistest parema ülevaate saamiseks tuli mudelid viia lihtsustatud kujule.

Edasises analüüsis võib joonisel 9 toodud protsessimudeli võtta nn baasmudeliks, mille põhjal saab genereerida levinuima sündmuste ahela, arvestades, et protsessiosa B läbitakse erinevate dokumentide loomise, muutmise ja kontrollimise tõttu korduvalt. Samuti on vaja meeles pidada, et erinevad dokumendid läbivaid seda tsüklit erinevalt, kuid on teada, et pärast erinevate dokumentide lisamist ja muutmist ning enne taotluse üle otsustamist, on üldjuhul sündmus *finish editing*.

### 4.3 Protsessimudel ProM-s kasutades $\alpha$ -algoritmi

Protsessikaeve tehnikatest saab sagedast kajastust  $\alpha$ -algoritm, mis oli üks esimesi logidest protsesside genereerimise algoritme suutes adekvaatselt tegeleda sündmuste kokkulangevustega. Algoritm justkui skaneerib sündmuste logi konkreetse mustri jaoks, otsides sündmuste otseseid ja võimalikke järgnevusi ja tuvastades hargnemisi. [4]

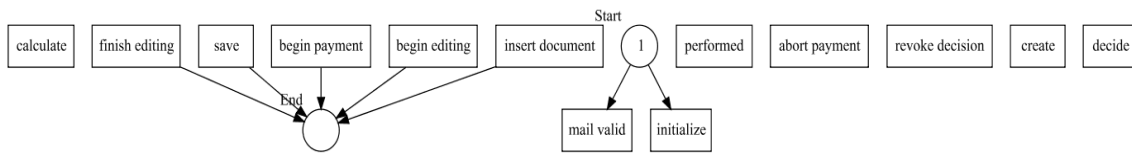
Sündmuste omavahelised seosed võivad olla järgmised:

- **Otsene järgnevus:**  $x > y$  parajasti siis, kui leidub selline sündmus  $x$ , mis järgneb otseselt sündmusele  $y$ .
- **Põhjuslikkus:**  $x \rightarrow y$  parajasti siis, kui  $x > y$  ja  $y \neq x$ .
- **Paralleelsus:**  $x \parallel y$  parajasti siis, kui  $x > y$  ja  $y > x$ .
- **Valik:**  $x \# y$  parajasti siis, kui  $x \neq y$  ja  $y \neq x$ . [22]

$\alpha$ -algoritmi ei peeta siiski väga praktiliseks protsessikaeve tehnikaks, kuna algoritmi toimimist takistavad müra, harvaesinevad ja mittetäielikud käitumised. Need takistused on aga reaalse tegevuse puhul pigem tavapärased. [23]



$\alpha$ -algoritmiga mudeli genereerimiseks on ProM-s pakett *Mine for a Petri Net using Alpha-Algorithm*. Nimetatud pakett annab väljundiks joonisel 10 kujutatud Petri võrgu.



**Joonis 10.**  $\alpha$ -algoritmiga genereeritud uuritava protsessi Petri võrk.

Jooniselt on näha, et  $\alpha$ -algoritm ei suuda erinevad sündmused siduda üheks terviklikuks protsessimudeliks. Üheks juba mainitud põhjuseks on see, et algoritm kaasab mudeli loomisse kogu info, sh müra, ning tulemuseks on sageli mittetoimiv protsessimudel. Teiseks probleemiks on erinevad kordused ehk tsüklid, mida algoritm ei suuda tuvastada. Varasemast analüüsist on teada, et erinevaid kordusi esineb aga selles protsessis hulganisti. Kolmandaks puuduseks on algoritmi suutmatust tuvastada mitte otseselt järgnevate sündmuste vahelisi seoseid. Teisisõnu, kui mingi sündmus mõjutab mitu sammu hiljem toimuvat sündmust, siis algoritm seda seost ei näe ning ei oska ka mudelis kajastada. [24]

Mudeli usaldusväärsuse testimine ei ole antud tulemuse puhul mõttekas, kuna ei suuda läbida ühtegi kriteeriumi.

#### 4.4 Protssimudel ProM-s kasutades heuristilist algoritmi

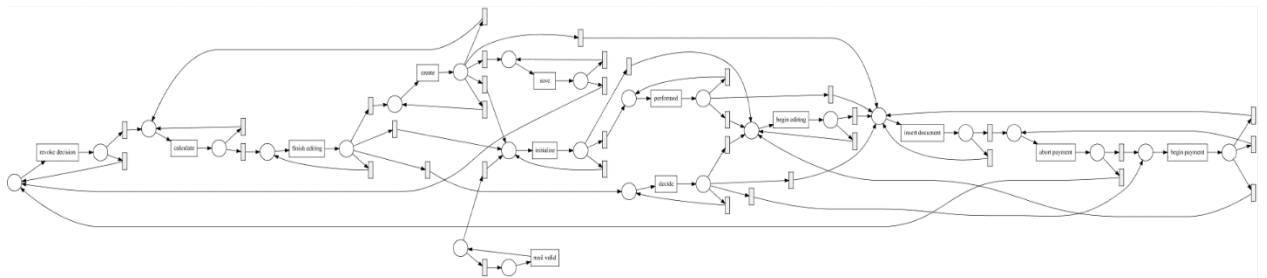
Heuristiline algoritmi näol on tegemist  $\alpha$ -algoritmi täiustatud versiooniga. Nii nagu ka  $\alpha$ -algoritmi puhul, genereeritakse kõigepealt otseste järgnevuste ( $x > y$ ) maatriks, kuid arvestades sagedusi. Sageduste arvestamine võimaldab välja filtreerida ebavajaliku müra ja harvaesinevad olukorrad. Järgmiseks genereeritakse omavaheliste seoste maatriks, mis näitab, kui tugevas seoses antud järjestuspaaris sündmused on. Seose tugevust arvutatakse valemiga:

$$|x \rightarrow y| = \frac{|x > y| - |y > x|}{|x > y| + |y > x| + 1},$$

kus järgnevused on summeeritud üle logifaili. Saadud seose tugevuse näitaja jääb -1 ja 1 vahele ning mida lähemal on saadud näitaja väärtusele 1, seda tugevama seosega on tegemist. On ilmne, et tugevad seosed tekivad juhul, kui sündmuste paari otsest järgnevust

ühte pidi esineb sageli ning teist pidi mitte kunagi või väga harva. Saadud maatriksite põhjal saab genereerida Petri võrgu või mõne muu analoogse mudeli. Heuristiline algoritm ei taga siiski usaldusväärse mudeli loomist, seega tuleb saadud mudelit kindlasti testida. [3], [25]

ProM-s saab heuristilist algoritmi kasutada mudeli loomisel erinevate pakettidega. Käesolevas töös on kasutatud paketti *Mine for a Heuristics Net using Heuristics Miner*. Algoritmi käivitades võimaldab pakett kasutajal anda täiendav sisend selle kohta, kui tugevas seoses peab olema mudel ja logid. Tulemus on konverteeritud Petri võrguks. [26]



**Joonis 11.** Protsessimudeli Petri võrk kasutades heuristilist algoritmi.

**Mudeli korrektse läbimise test.** Kõige tõenäolisemate kordustega arvestades on genereeritud sündmuste ahel, mida saab kontrollida heuristilise algoritmi põhjal saadud Petri võrgu vastu. Kontrollimisel peab arvestama, et algoritm ei ole kivanud protsessijoonisele filtreeritud logides fikseeritud algus- ja lõppsündmust (*mail income* ja *finish payment*). Seega võib mõtteliselt lugeda selle mudeli alguseks *mail valid* ja *begin payment*.

Valitud sündmuste jada: *mail income* → *mail valid* → *initialize* → *begin editing* → *finish editing* → *create* → *initialize* → *performed* → *initialize* → *begin editing* → *save* → *calculate* → *finish editing* → *decide* → *revoke decision* → *calculate* → *finish editing* → *decide* → *begin payment* → *abort payment* → *begin payment* → *insert document* → *finish payment*.

Antud ahelat ei ole võimalik sellel mudelil läbida. Probleemseteks osutusid järgmised sündmuste seosed: *begin editing* → *finish editing*, *performed* → *initialize*, *begin editing* → *save* → *calculate*.

**Mittelõppevate transaktsioonide test.** Selliseid sündmusi, millest ei saa liikuda lõppsündmuseni, ei tuvastatud.

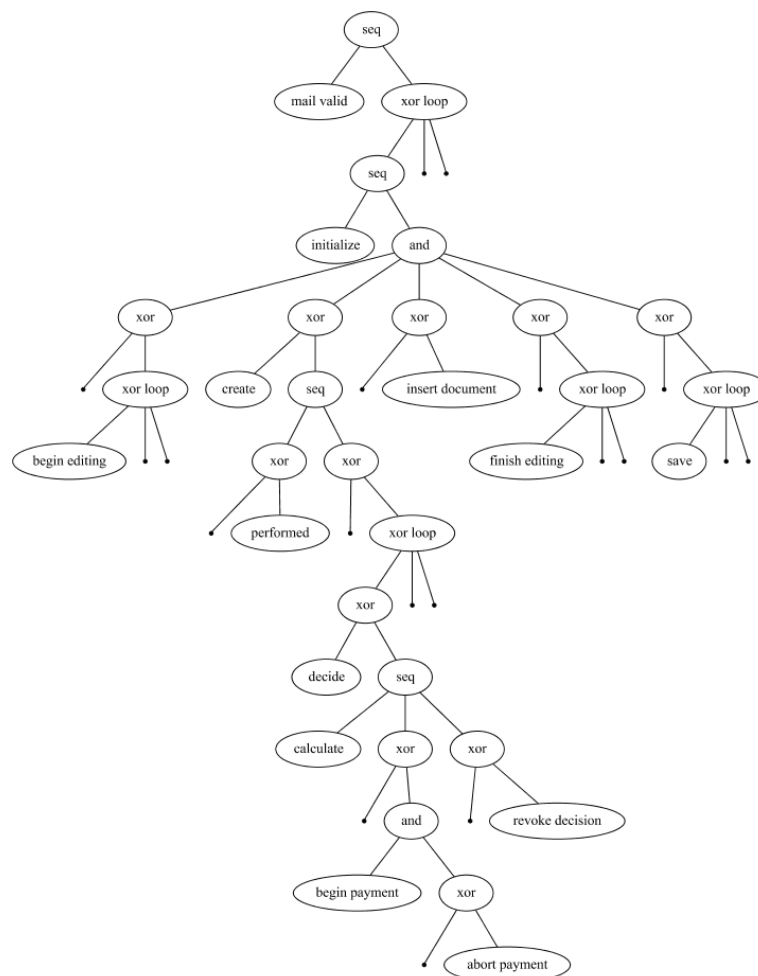
Mudel on visuaalselt keerukas ja ei võimalda levinumaid sündmusi nende järjestuses taasesitada. Seetõttu võib väita, et tegemist ei ole usaldusväärse mudeliga.

#### 4.5 Protsessimudel ProM-s kasutades induktiivset algoritmi

Järgmisena on analüüsimiseks kasutatud induktiivset algoritmi, mis on näide nendest vähestest algoritmidest, mis tagavad tulemuseks usaldusväärse protsessimudeli. Selle algoritmi tööpõhimõte koosneb järgmistest sammudest:

- leiab sündmuste logidest kõige suurema lahknevuse
- määrab operaatori, mis neid alamlogisid seob
- jätkab eraldi mõlema alamlogiga. [27], [28]

ProM-s on olemas induktiivse algoritmi kasutamiseks mitu paketti. *Mine process tree with Inductive Miner* annab väljundiks protsessipuu, mida saab ka soovi korral visualiseerida BPMN-s või Petri võrguna.



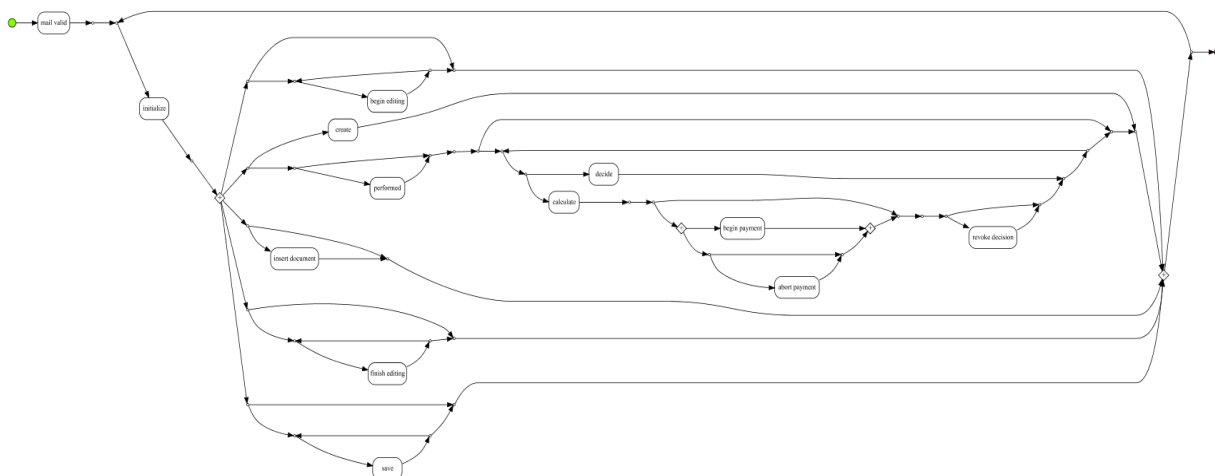
Joonis 12. Protsessipuu kasutades induktiivset algoritmi.

Protsessipuu on hierarhiline graaf, mis koosneb sõlmedest ja okstest (valikutest). Sõlmede puhul kasutatakse järgmisi tähistusi:

- **sequence** ehk järjestus näitab, millises järjestuses tuleb kõik valikud realiseerida;
- **or** – tähistab olukorda, kus tuleb realiseerida vähemalt üks valik;
- **xor** – tähistab olukorda, kus tuleb realiseerida üks valik;
- **xor loop** – tähistab olukorda, kus tuleb realiseerida esimene valik;
- **concurrent** – kõik valikud tuleb realiseerida, tegevused võivad ajaliselt kattuda;
- **interleaved** – kõik valikud tuleb realiseerida, kuid tegevused ei või ajaliselt kattuda;
- **and** – kõik valikud tuleb realiseerida, kuid ei ole täpsustatud ajalise kattuvuse nõuet.

Joonisel 12 kujutatud protsessipuu vastavust baasmudelile kontrollitakse edaspidi koos sama algoritmi põhjal genereeritud mudeliga.

Teiseks võimaluseks on kasutada paketti *Mine with Inductive visual Miner* [28], mis kujutab eespool saadud protsessipuu protsessijoonisena ning võimaldab seda mudelit ka animeerida. Nii nagu protsessipuu, nii saab ka protsessimudelit esitada Petri võrguna. Siinkohal on protsessimudel esitatud paketi autori Sander J.J. Leemans'i poolt välja töötatud kujul, mis sarnaneb üsnagi Petri võrgule või BPMN-le. [29] Joonisel 13 kasutatud operaatoritest ringikujuline sõlmpunkt tähistab välistavat valikut ning ruudu sees olev „+“ tähistab olukorda, kus kõik valikud tuleb realiseerida, kuid ajalise kokkulangevuse osas piirangud puuduvad. Roheline ja punane ring tähistavad vastavalt protsessi algust ja lõppu.



**Joonis 13.** Protsessimudel kasutades induktiivset algoritmi.

Kuigi induktiivse algoritmiga koostatud protsessimudel ei ole käesolevas töös esitatud Petri võrguna, vastab mudel testimiseks vajalikele nõuetele (on fikseeritud sündmustevaheliste seoste iseloom).

**Mudeli korrektse läbimise test.** Testimisel kasutatakse sama sündmuste jada, mis alapunktis 4.4. : *mail income* → *mail valid* → *initialize* → *begin editing* → *finish editing* → *create* → *initialize* → *performed* → *initialize* → *begin editing* → *save* → *calculate* → *finish editing* → *decide* → *revoke decision* → *calculate* → *finish editing* → *decide* → *begin payment* → *abort payment* → *begin payment* → *insert document* → *finish payment*.

Nii protsessipuu (joonis 12) kui -mudeli (joonis 13) puhul on tegemist sama mudeli erinevate esitusviisidega ning mõlema esituse puhul on antud testahel korrektselt läbitav.

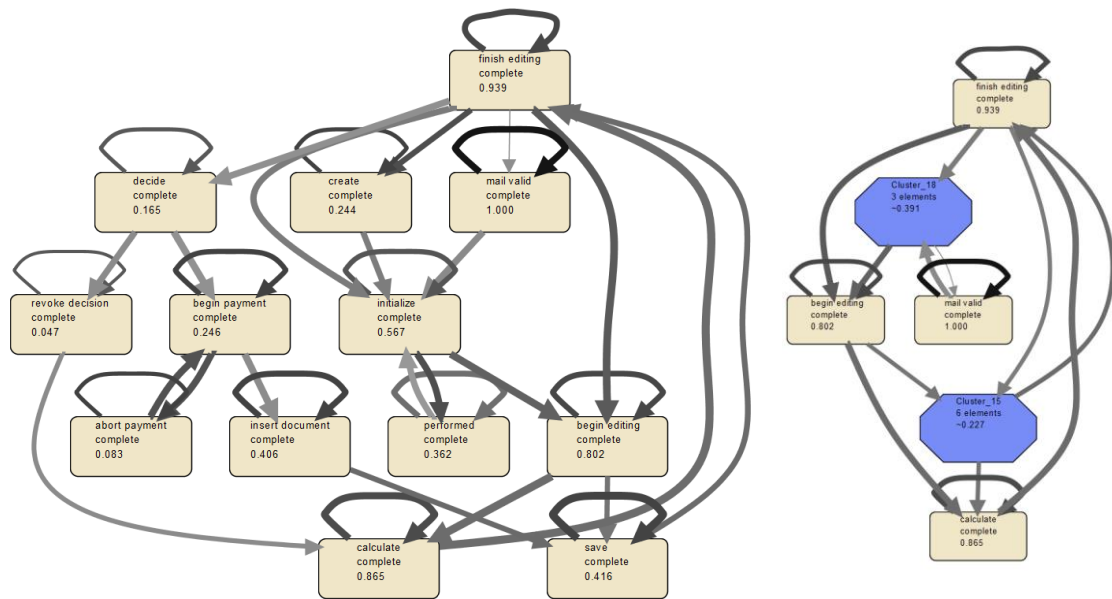
**Mittelõppevate transaktsioonide test.** Selliseid sündmusi, millest ei saa liikuda lõppsündmuseni, ei tuvastatud.

Mudel on mõõduka keerukusega ning seda saab lugeda usaldusväärseks.

## 4.6 Protsessimudel ProM-s kasutades hajusat algoritmi

Viimaseks algoritmiks, mida ProM-s protsessimudeli loomiseks kasutatakse (*Mine for a Fuzzy Model*), on hajus algoritm. Erinevalt eelmistest mudelitest, kus väljunditeks on Petri võrk ja selle alternatiivid, annab hajus algoritm väljundiks protsessi graafi. Peamiseks puuduseks hajusal algoritmil on see, et ta ei suuda eristada valikuid ( $x \# y$ ) ja paralleelsusi ( $x \parallel y$ ). Hajus algoritm tuleb hästi toime müraga ja suudab detailsusastet muutes näidata graafil soovitud hulka seoseid, kuid ei näita seoste iseloomu. Analoogiliselt induktiivse algoritmi paketiga ProM-s, saab ka siin kasutada protsessi animeerimist, mille abil on võimalik jälgida logides kajastatud valikuid, kuid keerulisema mudeli puhul ei aita see alati valikute iseloomu avada. [30]

Joonisel 14 on välja toodud kaks näidet hajusa algoritmi kasutamisel saadud protsessi graafidest. Vasakpoolsel joonisel on kuvatud kõik sündmused, parempoolsel on vähemolulised klasterdatud.



Joonis 14. Sündmuste vahelised seosed hajusa algoritmi põhjal.

Antud graafi puhul on vastavus logidega ehk sobivus ProM järgi 77,45%, st selline hulk logisid on võimalik edukalt taasesitada antud graafil. Analoogiliselt Disco ja Celonise mudeliga ei saa ka siin täpsust hinnata. Lihtsuse osas on mudel pigem keeruline, ei ole eristatavad protsessi algus- ja lõppsündmus ning raske on aru saada, kuidas nn keskmine protsessiahel võiks toimida. Selle põhjal võib öelda, et protsess on pigem ebasaldusväärne.

#### 4.7 Algoritmide võrdlev analüüs

Antud peatüki keskmis oli erinevate vahenditega antud logifailist protsessimudeli genereerimine. Erinevad tarkvarad ja lähenemised annavad tulemusi erinevateks eesmärkideks, nende paralleelne kasutamine täiendab sõltumata eesmärgist vaadet protsessile. Arvestades käesoleva töö uurimiseesmärki, siis on sobivaks protsessimudeli jaoks juhtimisvoost lähtuv protsessivaade, mille kvaliteet ja kasutajamugavus on piisav mõistmaks sündmuste vahelisi järgnevusi ja seoseid. Arvestades protsessi ja sellest tulenevalt ka logide iseloomu, kus tervikprotsess läbib erinevaid alamprotsesse ning nendega on omakorda seotud erinevad dokumentide liigid, oleks soovituslik, kui protsessimudel võimaldaks ka selle info kasutamist.

Erinevate tarkvarade ja algoritmidega eksperimenteerimine näitas ilmekalt protsessikaave tehnikate erisusi. Järgnevalt on tehtud kokkuvõtte tulemustest.

- **$\alpha$ -algoritm (ProM)**. Algoritmi käigus otsitakse järjestikuste sündmuste vahelisi seoseid, nagu otsene järgnevus, paralleelsus, valik, põhjuslikkus. Suudab luua protsessimudeli (Petri võrgu) nn puhtast logist, st logidest, kus ei esine müra, kordusi jms. Antud logifaili juures ei suutnud usaldusväärset mudelit genereerida.
- **Heuristiline algoritm (ProM)**. Kasutades sündmuste esinemise sageduse infot, suudab välja filtreerida müra, kuid ei suuda toime tulla rohkem kui otsesest järgnevusest tulenevate põhjuslikkusega. Antud algoritmi kasutamine ei taga üldjuhul usaldusväärse mudeli saamist, nii ka antud andmestiku puhul.
- **Induktiivne algoritm (ProM)**. Suudab tänu oma lähenemisele tagada tulemuseks protsessimudeli. Algoritmi põhiidee on logide eraldamine, saades maksimaalselt suure erinevusega alamhulgad. Sama tegevust korratakse kuni saavutatakse sarnane hulk, mida ei ole vaja enam jagada. Mudel esitatakse sageli Petri võrguna, kuid samaväärselt ka protsessipuuna või BPMN mudelina. Mudel säilitas küll mõningase keerukuse, kuid läbi sobivuse ja täpsuse testi – seega võib mudelit lugeda usaldusväärseks.
- **Hajus algoritm (ProM)**. Annab tulemuseks protsessigraafi, arvestades soovitud täpsust logide ja graafi vahel. Näitab kõiki esinenud seoseid sündmuste vahel, kuid ei suuda täpsustada seoste iseloomu (järgnevus, ja/või, ajaline kattuvus/mittekattuvus jne). Algoritm ei suutnud kasutatavat protsessijoonist genereerida, mis muudab mudeli vähe usaldusväärseks (kuigi seoste hulk graafil oli antud sobivuse suhtes korrektne).
- **Hajus algoritm (Celonis ja Disco)**. Celonise ja Disco mudelid baseeruvad samuti hajusal algoritmil ning seetõttu esinevad neil ka mõningad hajusa algoritmi puudused nagu mittefikseeritud info seoste iseloomu kohta. Samas osa puudusi on tarkvaralahendused suutnud kõrvaldada ning erinevalt ProM-i hajusa algoritmi mudelist, suudetakse siin anda protsessijoonisega edasi selgem arusaamine protsessi iseloomust. Celonise mudelil puudus dokumentidega seotud tasand (st ei olnud arvestatud sündmuse logides *Activity* väljaga), kuid oli protsessi vaatest paremini hoomatav. Disco mudel andis hea ülevaate sellest, kuidas tegelikult erinevad protsessiosa B kordused toimusid, mis on väga oluline just edasise uurimisküsimuse suhtes.

Kokkuvõttes võib öelda, et antud protsessikaeve projekti reaalsete logidega töötamisel suutsid kasutatava mudeli genereerida ProM-i induktiivne algoritm ning Disco ja Celonis, mis baseeruvad hajusal algoritmil. Milline neist mudelitest oli kogu projekti vaatest sobivaim, selgub analüüsis pärast projekti läbimist tulemuste kokkuvõttes peatükis.

Mõnes muus olukorras tuleks konkreetse algoritmi valikul lähtuda täpsemalt mudeli loomise eesmärgist ning logide kvaliteedist ja struktuurist. Mida keerulisem on mudeldatav protsess või mida ebakvaliteetsemad on logiandmed, seda suurem on kvaliteedi erinevus mudelite vahel. [19] Lõpliku otsuse tegemisel tuleb arvestada ka konkreetse tarkvara poolt pakutavaid lisavõimalusi ning kasutaja vajadusi.



## 5 Protsessi analüüs

Selles peatükis tehtavate analüüside eesmärgiks on saada teavet protsessi ja protsessis osalevate sündmuste omavaheliste seoste kohta. Analüüside valikul on lähtutud soovist leida lahendus ebasoovitavate lahendite avastamiseks ning leida võimalusel kohad protsessis, mille muutmine vähendaks tulevikus selliste olukordade teket. Praeguseni teostatud tegevused, nagu logide inspekteerimine ja protsessimudeli koostamine, on andnud juba mõningase ettekujutuse protsessi olemusest. Enne seoste otsimist erinevate stsenaariumite ja lõpptulemuste vahel, vaadeldakse protsessi jõudlust ning võimalikku seost täiendava ajakulu ning seotud ressursside vahel. Analüüside aluseks on eelmises peatükis välja valitud Disco ja Celonise ja ProM-i induktiivse algoritmiga saadud mudel.

### 5.1 Ajakulu ja jõudluse analüüs

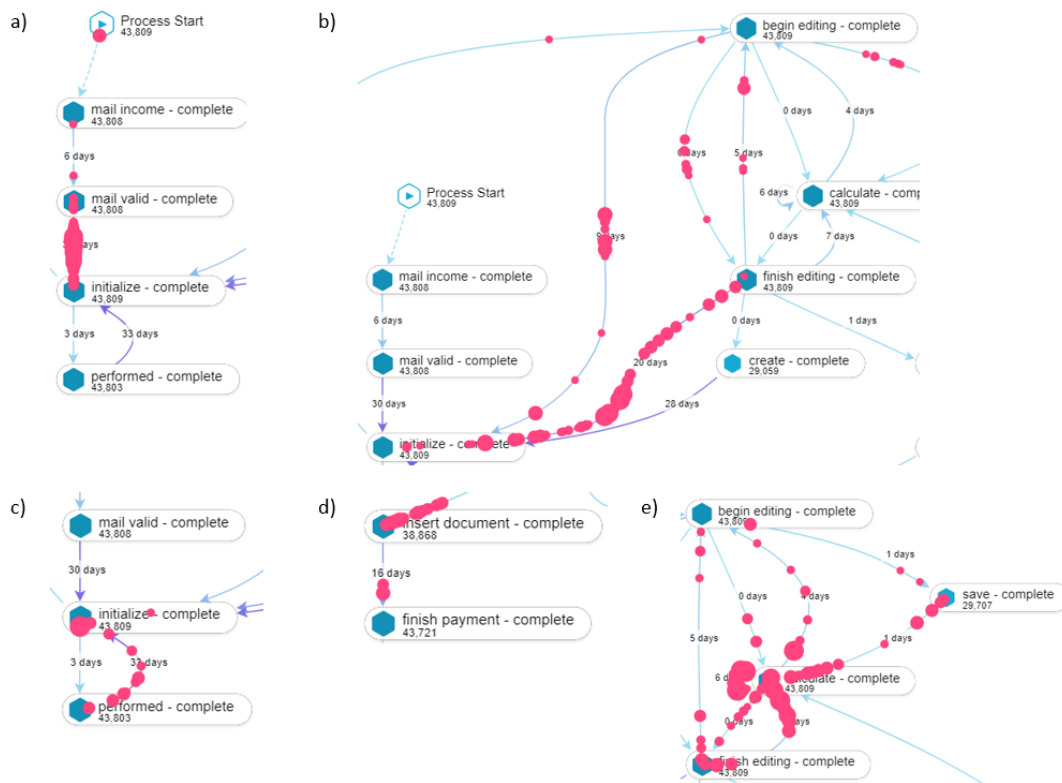
Ebaefektiivse ajakasutuse avastamine on väärtuslikuks sisendiks protsessi ümberkorraldamiseks, mille läbi on võimalik tõsta protsessi tootlikkust. Jõudlusprobleemiga tegelemine on üks sagedasemaid eesmärke juhtimisvoo keskses protsessikaave analüüsis, kuid käesolevas töös pigem täiendav vaade näitlikustamiseks, kuidas sellisest vaatenurgast pärit teave võib olla sisendiks protsessi sisulises analüüsis.

Erinevad protsessisammud kulutavad erinevatel põhjustel erineva hulga aega. Selleks, et hinnata, kas kulunud aeg on piisav või mitte, on vaja teada, milliselt on seda ajakulu planeeritud. See teave sisaldub üldjuhul kirjeldatud protsessi- või ärinõuetes, samuti võib olla ka tegemist regulatiivsete normidega (näiteks kehtestatud maksimaalne aeg mingile tegevusele) jne. Seega, kui ajakulu analüüs tooks välja, et maksete otsustusetapi järel on protsess liiga kaua ootel, siis võib see olla tingitud sellest, et makselahendus käivitataksegi sõltumata muudest protsessisammudest alles mingil kindlal kuupäeval. Samuti ei tähenda kõige ajamahukamate sündmuste väljatoomine midagi, kui need on tegelikult planeeritud ajagraafikus ja pudelikaela tekitab hoopis sündmus, mis ühepäevase kestuse asemel toimub keskmisel kolm päeva.

Need olid üldistatult põhjused, miks ajakulu ja jõudluse analüüsi sisuliselt sellisel tasemel, nagu seda reaalses olukorras peaks tehtama, siin ei teostata. Küll aga tuuakse näitlikustamiseks välja kõige aeganõudvamad sündmused ja viisid nende tuvastamiseks.

Antud töös kasutatavad tarkvarad pakuvad erinevat ajakuluga seotud statistikat, filtreerimist ning täiendava võimalusena protsessi visualiseerimiseks ka animeerimist. Eespool tutvustatud ProM-i lahendustest sisaldab animeerimise võimekust induktiivse ja hajusa algoritmiga modelleerimine. Nii Celonis kui Disco võimaldavad protsessijoonise analüüsivaates saada infot kulunud aja ja tootlikkuse (*performance*) kohta. Samuti on nendes tarkvarades lahendatud samas vaates terve protsessi või protsessiosa animeerimine.

Protsessi animeerimist võibki vaadelda kui esimest analüüsisammu, et tuvastada suurima ajakulu punktid ehk pudelikaelad. Kuid nagu eespool selgitatud, siis täiendavat ajakulu võivad tekitada ka suhteliselt väikese ajakuluga sündmused, mis kestavad nõuetes kehtestatud ajaga võrreldes kauem. Suuremate mudelite korral on parema tulemuse saamiseks mõistlik enne animeerimist uuritav protsessiosa välja filtreerida.



**Joonis 15.** Suure ajakuluga protsessiosad Celonise animatsioonis.

Pudelikaelade tuvastamiseks kõrvutati Celonise animatsioon ja Disco tootlikkuse näitajad. Celonise ajakuluga protsessijoonis ja animatsioon tõid välja mitmeid kitsaskohti. Kuigi Celonise joonis ei näita sündmustega seotud dokumente, siis animatsioonis on näha, et erinevaid protsessiosi läbitakse korduvalt, kuid ei ole teada, milliste dokumentidega seoses seda tehakse. Disco joonis, mis on lisatud käesoleva töö lissasse [Lisa 7], kus on sündmused dokumendi tasemel eraldatud, on näha selgelt, milliste dokumentidega seoses need pudelikaelad tekivad.

Järgnevasse tabelisse on koondatud kõige suurema ajakuluga sündmused koos taotluste hulga ja kulunud ajaga. On selgelt näha, et kõige aeganõudvamaks sündmuseks on *initialize* ning seda sõltumata dokumendist, mida parasjagu käideldakse.

**Tabel 4.** Suurima ajakuluga sündmused.

Sündmus	Seotud dokument	Taotluste hulk	Keskmine aeg
<i>initialize</i>	<i>Geo parcel document</i>	29 059	27,6 p
<i>initialize</i>	<i>Reference alignment</i>	43 802	54,9 p
<i>initialize</i>	<i>Payment application</i>	43 809	73,1 p
<i>finish payment</i>	<i>Payment application</i>	43 721	16,1 p

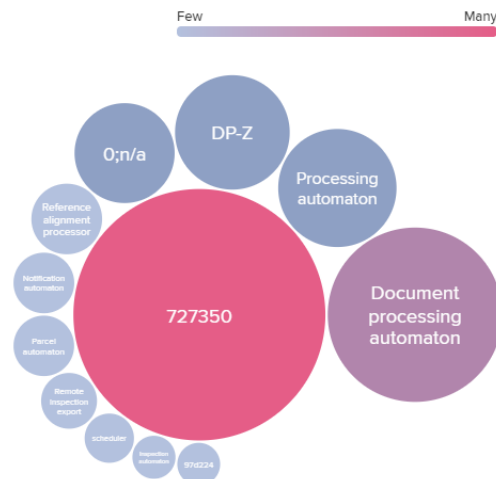
Järgmise sammuna on uurimise all kasutatud ressursid ning seoste otsimine leitud ajamahukate sündmustega. Ka siin on tegemist vajaliku taustainfo puudumisel üldisema uurimisega.

## 5.2 Ressursi kasutamise analüüs

Taotlusprotsess on peaaegu terves ulatuses automaatne, kuid siiski leidub logides jälgi ka manuaalsest sekkumisest. Üheks olukorraks on taotlused, mis on valitud manuaalselt kontrolli (*selected\_manually*). Teiseks on dokumentide manuaalsed muutmised, millele järgneb alati sündmus *Save*. Vaadates allüksuste tööd, siis selgub, et kõik allüksused menetlevad nende juurde saadetud taotlused lõpuni ära ning taotlusprotsessi vältel üksuste vahelist liikumist ei toimu.

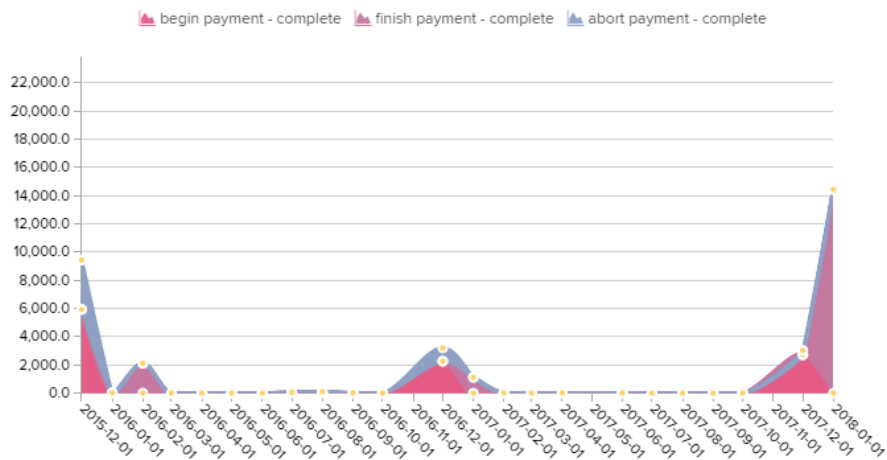
Klassikalises organisatsioonist lähtuv protsessikaeve keskendub sündmusega seotud tegevuse algatajale. Peamiselt on fookuses, kes ja mis ulatuses tegevusi läbi viib, kuidas toimib rollipõhine töökorraldus, kas tegelik tehtud tegevuste hulk on vastavuses

reeglitega jne. Antud juhul puudub täiendav info osapoolte ehk algatajate kohta, millised on neile kehtestatud reeglid (kohustused, õigused ja ligipääsud) ning nõuded tööviisile. Seega saab olemasolevat informatsiooni läbi töötades ning erinevaid kasutatavate tarkvarade tehnilisi lahendusi kasutades vaadata, kas hakkavad silma huvipakkuvad seosed või mustrid. Kõik kolm käesolevas töös kasutusel olevat protsessikaave tarkvara pakuvad erinevaid lahendusi ressursi kulu ja jaotuse uurimiseks. Sotsiaalvõrgustikke saab genereerida ProM-ga, teised tarkvarad pakuvad lihtsamaid graafilisi lahendusi.



**Joonis 16.** Taotlusprotsessiga seotud algatajad arvestades töödeldavat mahtu.

Joonisel 16 on kujutatud kõige sagedamini esindatud algatajad nii, nagu need logides kajastuvad (Celonis). Analüüsi teostajal ei ole kirjeldusi, mis kodeeritud algatajate taga on. Vaadates lähemalt, millised on algatajate peamised tegevused läbi aja, siis on osade algatajate funktsioon kergemini tuletatav. Näiteks, algatajad DP-Z ja DP-R tegelevad ilmselt maksetega seotud sündmustega. Joonis 17 näitab algataja DP-Z ressursikasutust.



**Joonis 17.** Osapool DP-Z koormus uuritaval ajaperioodil.

Kui lisaks maksete teostamisega tegelevatele algatajatele kaheksa algatajat on erinevad automaatprotsessid ning see väljendub ka algataja nimes, siis ülejäänud on kodeeritud ning nende puhul on keerulisem aru saada millega tegu. Kodeeritud algatajates pea pooled on tegevad ainult ühe allüksuse juures, siis ülejäänud on seotud erinevate allüksustega. Ka erinevate tegevuste ja dokumentide filtreerimine ei andnud täiendavat infot selle kohta, mida erinevad algatajad teevad või ei tee.

Kui praktikas oleks saanud selles analüüsis osas täiendada eelmises punktis leitud jõudluse infot ning otsida organisatsioonilisi kitsaskohti, siis antud andmestiku juures jäi infot väheks ning täiendavat teavet protsessi kohta siit ei tulnud.

### 5.3 Erinevad lõpptulemuste stsenaariumid

Üldjuhul kategoriseeritakse protsessi lõpptulemused positiivseteks ja negatiivseteks. Lisaks sellele, et oodatud tulemus realiseeruks, on oluline jälgida, et see toimuks ka etteantud aja jooksul. Sellest lähtuvalt on defineeritud võimalikud huvipakkuvad lõpptulemused.

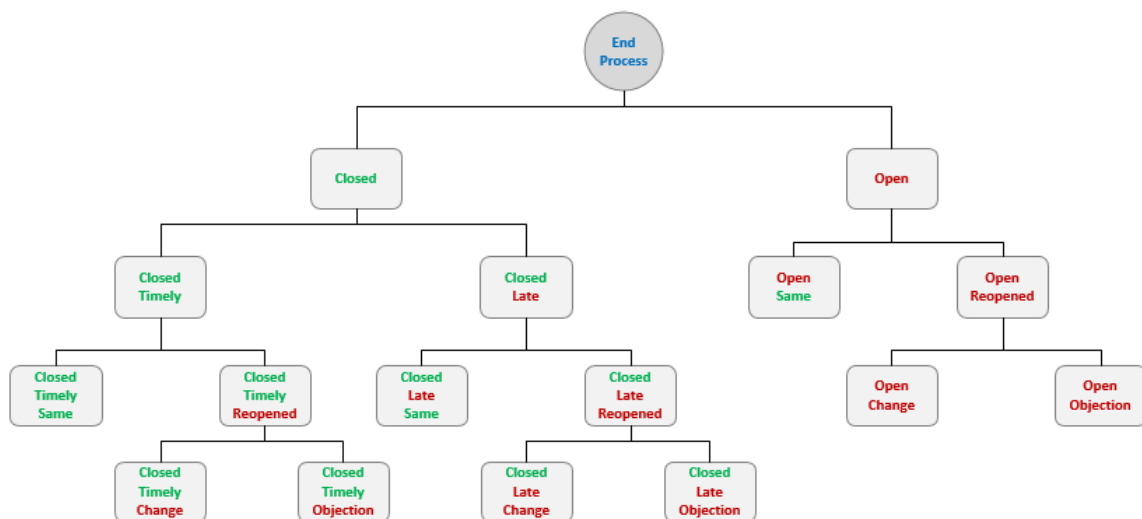
Jooksva aasta taotluste väljamakse korraldused peaksid eelduslikult saama algatatud sama aasta lõpuks. Siiski on igal aastal juhtumeid, kus seda ei saavutata. Soovimatuteks tulemusteks on makse hilinemine ja maksetaotluse menetluse uuesti avamine. Makse on õigeaegne kui aasta lõpuks saavutatakse toiming *begin payment*, millele ei järgne enam sündmust *abort payment* [15]. Logisid lähemalt uurides selgub, et taotlus võib korduvalt läbida sündmusi *begin payment* ja *abort payment*, enne kui liigub edasi sündmusesse *finish payment*. Kõige viimase maksealgatuse tuvastamisel tuleb arvestada, et see sündmus võib olla välja kutsutud seoses erinevate alamprotsessidega (*Application, Change, Objection*).

**Tabel 5.** Õigeaegselt alustatud ja viivituses maksete hulk aastate kaupa.

Aasta	Taotluste arv	Viimane makse algatati õigeaegselt	Makse algatamine viibis	Makset ei ole algatatud
2015	14 750	14 261	486	3
2016	14 552	14 338	209	5
2017	14 507	14 433	0	74
<b>Kokku</b>	<b>43 809</b>	<b>43 032</b>	<b>695</b>	<b>82</b>

Tabelist on selgelt näha, et viivitustega maksete alustamine on tugevalt langustrendis. Teine ebasooviv tav tulemus on taotluse uuesti läbivaatamine (*reopened*), mille võib mingil põhjusel algatada menetlev allüksus (alamprotsess *Change*) või vaidlustada taotluse esitanud isik (alamprotsess *Objection*). Taotluse uuesti läbi vaatamine võib kaasa tuua ka täiendavaid makseid. Täiendav makse kajastub positiivse summana väljal  $payment\_actual\{x\} > 0, x > 0$ . Kui logides on maksesumma 0, siis ei ole tegemist reaalse maksega, vaid olukorraga, kus makse on tagasi lükatud, kuid taotlus kajastub makseprotsessis. Negatiivse summa korral peab taotleja määratud summa tagasi maksuma. Samas võib olla tegemist olukorraga, kus vaidlustus või muutmise on algatatud, kuid täiendava makseni ei ole veel jõutud. Kui on esitatud täiendav nõue, siis kajastub see andmeväljal  $amount\_applied\{x\}, x > 0$ .

Erinevate lõpptulemuste klassifitseerimisel tuleb kõigepealt defineerida taotluse staatused avatud (*open*) ja suletud (*closed*). Kõik taotlused, kus leidub algatatud alamprotsessiga (*Application, Change, Objection*) seotud makse lõpetamise sündmuse *finish payment*, saab lugeda analüüsi alustamise seisuga suletuteks. Pärast makse lõpetamist võib olla logis veel kandeid, see ei tähenda, et taotlus oleks avatud staatuses. Tegemist võib olla näiteks dokumenteerimisega seotud tegevustega. Kui taotluse elutsükli jooksul ei ole taotlust vaidlustatud ega muudetud, kuid taotlus ei ole jõudnud ka makseni, siis on taotlus avatud. Samuti avatud juhtudel, kui taotlus on taasavatud ning ei ole jõutud makseni.



**Joonis 18.** Seosed erinevate lõpptulemuste vahel.

Tegelikult jagunevad erinevad ebasoovitavad lõpptulemused omakorda erinevate tingimuste kaupa (erinevad inspekterimised, maksenõuded taotlejale, makse pole veel algatatud jne), kuid antud töö raames neid detailsemalt ei analüüsita.

**Tabel 6.** Lõpptulemuste järgi grupeeritud taotlused aastate kaupa.

Lõpptulemus	Lühend	Aasta	Taotluste arv	Taotluste osakaal
Taotlus suletud Õigeaegne makse Sama taotlus	<i>Closed_timely_same</i>	2015	11 634	78,87%
		2016	12 681	87,14%
		2017	14 349	98,91%
Taotlus suletud Õigeaegne makse Taotlus muudetud	<i>Closed_timely_ch</i>	2015	1 105	7,49%
		2016	396	2,72%
		2017	0	0%
Taotlus suletud Õigeaegne makse Taotlus vaidlustatud	<i>Closed_timely_obj</i>	2015	0	0%
		2016	0	0%
		2017	0	0%
Taotlus suletud Hilinenud makse Sama taotlus	<i>Closed_late_same</i>	2015	35	0,24%
		2016	50	0,34%
		2017	0	0%
Taotlus suletud Hilinenud makse Taotlus muudetud	<i>Closed_late_ch</i>	2015	9	0,06%
		2016	0	0%
		2017	0	0%
Taotlus suletud Hilinenud makse Taotlus vaidlustatud	<i>Closed_late_obj</i>	2015	401	2,72%
		2016	142	0,98%
		2017	0	0%
Taotlus avatud Sama taotlus	<i>Open_same</i>	2015	0	0%
		2016	5	0,03%
		2017	77	0,53%
Taotlus avatud Taotlus muudetud	<i>Open_ch</i>	2015	1 360	9,22%
		2016	1 195	8,21%
		2017	43	0,30%
Taotlus avatud Taotlus vaidlustatud	<i>Open_obj</i>	2015	206	1,40%
		2016	83	0,57%
		2017	38	0,26%

See tabel ilmestab selgelt olukorda, kus 2017. aasta tulemused ei ole varasemate aastatega osades jaotustes võrreldavad, kuna paljude taotlused ei ole lõpplahenduseni jõudnud. Teisisõnu, mingi hulk *closed* staatuses taotlused lähevad tegelikult tagasi *open* staatusesse, kuna taotluse esitajad ei ole veel jõudnud vaidlustusi esitada. Sellest tingituna teostatakse lõpptulemuste mõjutegurite analüüsi ainult 2015. ja 2016. aasta andmete põhjal. Lisaks selgub tabelist, et vaidlustatud taotlused ei jõua õigeaegsesse maksegraafikusse. Seega edaspidi seda lõpptulemust ei vaadelda.

## 5.4 Lõpptulemust mõjutavad tegurid

Erinevate lõpptulemuste realiseerumine võib sõltuda väga erinevatest teguritest. Lisaks sellele, et taotluse uuesti läbi vaatamine ilmselt tekitab olukorra, kus makse viibib, on oluline uurida, miks taotlused uuesti läbi vaatamisele lähevad. Arvestades taasavatud ja mitteõigeaegsete maksetega taotluste osakaalusid, on selge, et negatiivsest lõpptulemusest on problemaatilisemad just uuesti menetletud taotlused.

Vaidlustused esitatakse taotlustele alles järgmisel kalendriaastal, seega ei ole peale langetatud otsuse rohkem otseseid tegureid, mis sellise olukorra tekitavad ning tuleb hakata otsima kaudseid. Samuti tuleb otsida põhjuseid, mis tingivad makse viibimise ja taotluse taasavamise muutmise põhjusel. Üks võimalus selleks on erinevate hüpoteeside (stsenariumite) genereerimine ja nende testimine.

Hüpoteeside genereerimisel on üheks võimalikuks meetodiks sõeluda välja alateemade kaupa potentsiaalsed mõjutegurid ning neist huvipakkumamad läbi testida. Selliseid mõjutegureid võib otsida järgmistest hulkadest:

- võimalikud sündmused, mille läbimine või mitteläbimine võiks mõjutada lõpptulemust;
- ajakulu mingile kindlale sündmusele või protsessiosale (liiga palju või liiga vähe);
- varasemate aastate tulemused;
- statistilised erisused alusandmetes;
- jne.

Erinevate stsenaariumite mõju olulisuse hindamiseks kasutatakse hüpoteeside statistilist testimist. Kui tulemused on juhuslikud, siis ei saa eeldada, et tulevikus sama valikuga grupis tuleks sama lõpptulemus. Testimisel kasutatakse kahepoolset z-testi, mille eelduseks on suuremad sõltumatud andmehulgad. [31] Testimiseks on vaja fikseerida valim, nullhüpotees ja olulisusnivoo. Antud testides on olulisusnivoo 0,05 (5%), st kui z-testis  $p < 0,05$ , siis võib nullhüpoteesi ümber lükata ning väita, et tulemused on gruppide kaupa erinevad. Teisisõnu, genereeritud stsenaarium avaldab sellisel juhul tõenäoliselt valitud lõpptulemusele mõju. Statistilised testimised on teostatud RStudio-s [32]. Nullhüpoteeside selgitused ning tulemused on sõnastatud pärast kokkuvõtlikku tabelit. Kõikide testimiste valimites on ainult 2015. ja 2016. aasta andmed. Kui hüpotees sisaldab



järjestikuste aastate korduvtaotlusi, siis esinevad valimis ainult nende isikute taotlused, mis esinesid mõlemal aastal.

**Tabel 7.** Erinevate tunnuste statistilised testid.

ID	Lõpptulemus	Testitavad grupid	Taotlusi	p
T1	Makse viibib	Alusandmete täiendav kontroll Täiendavat kontrolli ei tehtud	83 554	8.44e-09
T2	Makse viibib	Dokumentide menetlemine > 20p Dokumentide menetlemine < 20p	333 304	< 2.2e-16
T3	Makse viibib	Maatükkide arv > 34 Maatükkide arv < 34	240 397	< 2.2e-16
T4	Makse viibib	Young Farmer Application = Yes Young Farmer Application = No	59 578	0.1158
T5	Makse viibib	Eelneval aastal makse viibis Eelneval aastal makse õigeaegne	35 143	< 2.2e-16
T6	Taotlus vaidlustatakse	Esmane makse < 70% soovitus Esmane makse > 70% soovitus	113 719	< 2.2e-16
T7	Taotlus vaidlustatakse	Eelneval aastal vaidlustus Eelneval aastal ei vaidlustanud	52 156	< 2.2e-16
T8	Taotlust muudetakse	Taotlus täiendaval kontrollimisel Taotlust täiendavalt ei kontrollitud	252 3 813	0.01267
T9	Taotlust muudetakse	Eelneval aastal muudeti Eelneval aastal ei muudetud	927 578	< 2.2e-16

**Makse viibib.** Selle lõpptulemuse juures on valimisse võetud ainult suletud taotluste hulk (*Closed\_late* ja *Closed\_timely*). Avatud staatuses taotlused võivad olla küll makse õigeaegselt alustanud, kuid ei ole alust arvata, et taotlused ei lähe uuesti menetlemisse ja see tooks kaasa täiendavaid makseid.

- **T1.** Nullhüpotees: makse hilinemine ei sõltu sellest, kas teostati taotluse aluseks oleva maa andmete üle täiendavat kontrolli. Test näitab, et nullhüpoteesi võib ümber lükata ning nende gruppide vahel on statistiliselt oluline erinevus. Kui taotlus on valitud kontrolli (juhuslikult, manuaalselt, riskipõhiselt või eelneva aasta kontrollimise tulemusel), siis on 1,95 korda suurem tõenäosus, et makse viibib kui kontrollimata taotluste puhul.
- **T2.** Nullhüpotees: makse hilinemine ei sõltu sellest, kas taotluse aluseks olevate dokumentide (*Geo parcel document*) menetlemine võtab aega rohkem või vähem kui keskmiselt (20 päeva). Testimine näitab, et selle hüpoteesi võib ümber lükata.

Koguni 2,6 korda on tõenäosus makse hilinemisele, kui vastavaid dokumente menetletakse liiga kaua.

- **T3.** Nullhüpotees: makse hilinemine ei sõltu taotleja maatükkide arvust (arvestades keskmist 34 maatükki). Test näitab, et selle hüpoteesi saab ümber lükata ning suurem tõenäosus makse hilinemiseks (3,37 korda suurema tõenäosusega) on just neil, kellel on keskmisest enam maatükke. See tulemus on osaliselt seotud ka eelmise testiga ning annab aluse arvata, et taotluse maaga seotud alusdokumentide menetlemine on seotud maatükkide arvuga (vajalikke katastritoiminguid tehakse ilmselt tükipõhiselt).
- **T4.** Nullhüpotees: makse hilinemine ei sõltu sellest, kas esitatakse taotluse tüüp *Young Farmer Application*. Alternatiivse hüpoteesi püstitamisel on eeldatud, et noore farmeri toetust esitavad pigem väiksema kogemusega taotlejad ning menetlusaeg võib sellest tulenevalt pikeneda. Test näitas, et nullhüpotees jääb kehtima, seega ei ole statistiliselt nendel taotlusegruppidel olulist vahet.
- **T5.** Nullhüpotees: makse hilinemine ei sõltu sellest, kas eelmisel aastal samal taotlejal makse viibis või mitte. Test lükkab nullhüpoteesi ümber ja saab väita, et kui eelneval aastal makse viibis, siis tõenäoliselt viibib see ka järgmisel aastal. Seos eelneva aasta tulemustega on arvestatav, koguni 7,64 korda on tõenäosus suurem makse hilinemiseks kui see juhtus taotlejal ka eelneval aastal.

**Taotlus vaidlustatakse.** Valimisse on arvestatud kõik 2015. ja 2016. aasta taotlused, mis on vaidlustatud.

- **T6.** Nullhüpotees: vaidlustuste hulk ei sõltu sellest, kas esmane saadud summa on alla 70% või üle. Saadud tulemus lubab selle hüpoteesi ümber lükata. Selgub, et kui saadud taotlussumma jääb alla 70%, siis on 9,4 korda suurem tõenäosus, et taotlus vaidlustatakse.
- **T7.** Nullhüpotees: vaidlustuse esitamine ei sõltu sellest, kas eelneval aastal taotleja esitas vaidlustuse. Testimine näitas, et selle nullhüpoteesi saab ümber lükata. Analoogselt seosega, et eelneval aastal hilinenud makse on tugevas seoses jooksva aastal tekkiva samalaadse olukorraga, nii ka sellel juhul. Koguni 7,98 korda suurem tõenäosus on vaidlustuse uuesti esitamisel, kui eelmisel aastal on vaidlustus esitatud.

**Taotlus läheb muutmisele.** Valimis on taotlused, mis on läinud muutmisele, kuid ei ole vaidlustatud.

- **T8.** Nullhüpotees: taotluse muutmisotsus ei sõltu sellest, kas taotlus on suunatud täiendavasse kontrolli. Tõepoolest, hüpoteesi testimine näitab, et 5% olulisusnivoo puhul on väike erinevus gruppide vahel olemas. Kuid kui muuta see 1%-ni, siis ei oleks statistiliselt olulist erinevust taotluse muutmisprotsessi algatamisel, kas taotlus on läbinud kontrolli või mitte.
- **T9.** Nullhüpotees: taotluse muutmisse suunamine ei sõltu sellest, kas taotlust eelneval aastal muudeti. Ka siin võib nullhüpoteesi ümber lükata, koguni 7,87 korda on suurem tõenäosus, et taotlus läheb muutmisele, kui taotleja eelneva aasta taotlust muudeti.

Hüpoteeside testimine tõi välja, et üks arvestatav indikaator, et taotlus kas taasavatakse või makse ei ole õigeaegne, on kui vastav ebasobiv tulemus on eelneval aastal realiseerunud. Samuti on riskifaktoriteks taotluse alusandmete rohkus või nende pikem menetlemisaeg.

Ideaalses olukorras soovib taotlusprotsessi haldaja saada hoiatava signaali soovimatu lõpptulemuse võimalikust realiseerimisest enne kui tehakse otsus taotluse rahuldamise kohta (*Payment application+application+decide*) [15]. Kui kitsendada analüüsi avastamise aja osas, siis tuleks täiendavalt sõeluda ja testida selliseid stsenaariume, mis saavad vajaliku info enne sündmust *decide*.

## 5.5 Tagasiside protsessile

Protsessi analüüsi osas teostati protsessile ajakulu, jõudluse ja ressursi analüüs, et saada sisendit võimalike lõpptulemust mõjutavate tegurite osas. Nimetatud analüüsidel oli käesolevas töös pigem illustratiivne roll, sisendit hüpoteeside genereerimiseks saadi vähe.

Mõjutegurite analüüsi osas saab viidata võimalikele probleemkohtadele, kuid kuna tegemist on selliste algandmetega, mille puhul protsessi omanik ka ise veel ei tea kõigi võimalike seoste hulka [15], siis puudub valideerimisvõimalus.

Tähelepanekud protsessi kohta ettepanekute vormis taotlusprotsessi haldajale:

- Arvestades tugevat seost järgnevate aastate tulemuste vahel, lisada logi struktuuri täiendav riskikategooria, mis sisaldaks infot eelneva aasta negatiivse stsenaariumi realiseerumisest.
- Mitu võtmetähtsusega sündmust algatatakse aasta viimastel kuudel. Kuna maksete tühistamise osakaal on arvestatav ning soov on aasta lõpuks tagada maksimaalselt suur hulk selliseid taotlusi, mis enam katkestamisse ei lähe, siis võiks mingite taotluste puhul teostada kas eelkontrolle või algatada otsustamis- ja makseprotsesse varem.
- Üle vaadata ning teostada protsessile regulaarset seiret maaga seotud dokumentide menetlusprotsessi osas, mis on üks võtmeprotsessidest. Isegi juhul kui protsessi omaniku hinnangul on protsess ja sellega seotud tegevused (kontrollid jms) juba maksimaalselt efektiivsed.
- Kõrvutada protsessikaevetarkvarades ajakulu andmed oodatavate ajakulu andmetega ning anda hinnang protsessi jõudlusele. Markeerida sündmused ja protsessiosad, mis algatatakse esimesel võimalusel ning need mis algatatakse kindlal ajavahemikul. Selline täiendava info lisamine võimaldab teostada sisukamat ajakulu analüüsi.
- Arvestades teavet sündmuste algatajate ja tekkinud pudelikaelade kohta, viia läbi täiendav analüüs selgitamiseks välja, kas ressursikasutus on efektiivne. Üle vaadata algatajate info kajastamine logides: hetkel sama algataja taga suurem hulk erinevaid tegevusi, mis ei võimalda saada piisavalt detailset vaadet ressursianalüüsiks.

## **6 Tulemuste analüüs ja järeldused**

Käesolevas töös läbiti protsessikaave projekt eesmärgiga näitlikustada ühte võimalikku viisi sellise projekti teostamiseks, viidi läbi protsessimodelite test tegemaks kindlaks, millisel algoritmil põhinev mudel on antud andmete juures kvaliteetseim ning antud projektile sobivaim, leiti ebasobivaid lõpptulemusi mõjutavad tegurid ning hinnati tehtud analüüside valguses kasutatud protsessikaave tarkvarasid.

### **6.1 Protsessikaave projekti ülesehitus**

Protsessikaave projekti üldisel ülesehitusel lähtuti kuue sigma projekti etappidest, mille analüüsi osasse otsiti kirjandusest erinevaid protsessikaave analüüse, et läbi eksperimenteerimise teha kindlaks, millised analüüsid selles näidisprojektis toetasid projekti eesmärki ja mis mitte.

Ettevalmistuse faasis on soovitav lisaks andmete, sisu ja struktuuri, üle vaatamisele kontrollida ka protsessi stabiilsust. Tegemist on sellise varjatud dünaamikaga, mis ei avaldu teisi analüüse tehes, kuid kontrollitava deformatsiooni olemasolu korral võib saadud tulemusi oluliselt moonutada. Kõrvalekallete analüüsi eelduseks on piisava hulga logiandmete olemasolu. Analüüsi teostamine andis olulist infot protsessi kohta ning jääb kindlasti soovitatavate analüüside nimekirja.

Logide eelanalüüsi faasis katsetati lisaks filtreerimisele ka klasterdamist, kuid mis antud projektis ei leidnud kasutust. Mõne muu eesmärgiga protsessikaave projekti puhul võib selline eeltöötlus olla väga kasulik. Logide filtreerimine on vajalik tegevus erinevates projekti osades ning kuulub kindlasti oluliste etappide hulka.

Protsessimodeli koostamisest tuleb detailsemalt juttu järgmises alapeatükis. See on etapp, mis peaks kindlasti sisalduma igas protsessikaave projektis.

Protsessi analüüs näitas, et on mõistlik kasutada võimalusel erinevaid protsessimodeli vaateid (lisaks juhtimisvoo vaatele ka näiteks organisatsioonilist), kuna suurema andmehulga puhul võib teistest vaadetest ilmnedu täiendavat infot. Üheks selliseks

näiteks oli töös ajakulu analüüs, mis ei olnud otseselt ülesande püstituses, otsimaks protsessi kitsaskohti ning läbi selle võimalikke mõjutegureid lõpplahendusele.

Ebasoovitavaid lõpptulemusi mõjutavate tegurite leidmiseks koguti kokku eelnevate analüüside põhjal koostatud erinevad hüpoteesid ning teostati statistiliste testidega olulisuse kontroll. Need analüüsid ja testimised on lisatud analüüsi osa viimaseks etapiks: täiendavad analüüsid põhjuste ja tagajärgede seoste uurimiseks.

Kokkuvõttes saab välja tuua läbi viidud protsessikaeve projekti etapid, mis ostusid eesmärgi saavutamist toetavateks:

- *define* – projekti eesmärgi määramine;
- *measure* – oluliste mõõdikute fikseerimine ning logiandmete kogumine (tehtud protsessi omaniku poolt);
- *analyze* – andmete analüüsimine põhjuste ja tagajärgede seoste uurimiseks ja kontrollimiseks;
  - protsessi ajas muutumise analüüs;
  - logide inspekteerimine ja eelanalüüs (sh logide filtreerimine);
  - protsessimudeli koostamine ja kvaliteedi testimine;
  - ajakulu ja jõudluse analüüs;
  - ressursi analüüs (organisatsioonilisest vaatest);
  - täiendavad analüüsid põhjuste ja tagajärgede seoste uurimiseks;
- *improve* – praeguse protsessi parandamine tehtud analüüsi põhjal (ettepanekud protsessi tagasiside alapeatükis);
- *control* – protsessi kontrollimine kõrvalekallete minimeerimiseks (saab teostada uute logide tekkimisel).

## 6.2 Protsessimudeleid genereerivate algoritmide võrdlus

Selles alapeatükis keskendutakse kokkuvõtvalt mudelite võrdlusele. Põhjalikum selgitus on toodud käesoleva töö punktis 4.7. Mudelite esmasel hindamisel lähtuti mudeli kvaliteedist. Kvaliteedihinnangud oli kõrgeimad induktiivse algoritmiga mudelil (ProM), sellele järgnesid hajusal algoritmil põhinevad Disco ja Celonise mudelid ning seejärel heuristiline ProM-i mudel, ProM-i hajusal algoritmil põhinev protsessigraaf ning kõige lõpuks ProM-i  $\alpha$ -algoritmiga genereeritud mudel. Viimane algoritm andis ainukesena mudeli, mida ei olnud võimalik üldse kasutada ning seda ei vaatle ka seetõttu järgnevas

analüüsis. Kvaliteedihinnangud on seotud logide ja protsessi keerukusega, seega väljendavad need hinnangud mudelite kvaliteeti antud andmete kontekstis. Protsessi analüüsi jaoks valiti välja neist kolm esimest.

ProM-i heuristiline ja induktiivne mudel töötasid ainult selliselt filtreeritud logide peal, kus ei esinenud protsessi algus- ja lõppsündmuse osas hargnemist. Filtreerimise käigus tuli ära valida ka nende sündmuste hulk, mida soovitakse mudelil kajastada. Soovides muuta mudeli täpsust, tuleks genereerida teine filtreeritud logifail ning selle põhjal mudeldada. Selline eeltöö teeb mudelite genereerimise veidi kohmakaks. Samas omistab ainult induktiivne algoritm sündmuste vahelistele seostele loogilised operaatorid.

Protsessimudelist ülevaate saamiseks sobivad Disco ja Celonis väga hästi. Miinuseks on ainult ühe algoritmi põhjal ühe mudeli saamine. Üksikute sündmuste lisamise või eemaldamise võimalus pigem ohustab mudeli kvaliteeti. ProM plussiks on lisaks erinevate algoritmide kasutamise võimalusele ka erinevate väljunditüüpide olemasolu.

Selleks, et saada teada, milline neist mudelitest oli antud protsessikaave projekti jaoks sobivaim, siis tuleks kõrgeima kvaliteediga mudeleid omavahel võrrelda. Ajakulu, jõudluse ja ressursside analüüsis kasutati Disco ja Celonise mudeleid võrdselt. Nende mudelite eeliseks oli mugav visuaalne lahendus. ProM-i mudel leidis vähem kasutust, kuna ressursi info ei olnud piisava detailsusega ning vajadust detailsema info saamiseks puudus. Lõpptulemusi mõjutavate hüpoteeside kaardistamisel lähtuti kõige rohkem Disco mudelist, kuna see sisaldas seotud dokumentide infot.

Kokkuvõtteks võib öelda, et antud andmestiku arvestades, olid kõige kvaliteetsemad mudelid ProM-i induktiivsel algoritmil põhinev ning Disco ja Celonise hajusal algoritmil põhinevad mudelid. Käesoleva töö protsessikaave projekti suhtes osutus sobivaimaks (leidis kõige rohkem kasutust) Disco mudel. See aga ei tähenda, et kogu projekti tarkvara valikul oleks olnud see sobivaim.

### **6.3 Lõpptulemust mõjutavate tegurite leidmine**

Lõpptulemust mõjutavate tegurite leidmisel kasutati läbi projekti saadud teavet ning selle põhjal sõeluti välja erinevad hüpoteesid. Kuna andmestik oli kolme järjestikust aastat hõlmav, siis oli võimalik ära kasutada korduvtaotluste käitumine. Samas tuli arvestada, et viimase aasta andmeid ei saa analüüsides kasutada, kuna eelmiste aastate põhjal saab

väita, et paljud neist on tegelikult veel avatud olekus. Hüpoteeside olulisuse hindamiseks kasutati statistilisi teste (kahepoolsed z-testid).

Selle analüüsi suurimaks väljakutseks oli lõpptulemuste klassifitseerimine ja reaalselt logide markeerimine. Kuna tegemist ei olnud nn puhta baasiga vaid müra sisaldava reaalsete logidega, siis erandolukordi tuli rohkem ette kui mis tahes näidisbaasiga oleks tekkinud.

Tulemused esitati soovitude näol, mida analüüsi läbiviija (ehk töö kirjutaja) ise andmestikuga või protsessiga ette võtaks kui ta töötaks antud taotlusprotsessi eest vastutavas organisatsioonis. Ettepanekud sisaldasid soovitusi täiendada logide andmestruktuuri täiendavate atribuutide näol (sündmuste eristamine algatushetkest lähtuvalt, organisatsiooniliste osapoolte kohta täiendava info lisamine, eelneva aasta lõpptulemuse kajastamine eraldi riskikategooria väljal), vaadata üle võtmeprotsesside läbiviimine vältimaks ebaefektiivsust (maksete algatamine ainult aasta lõpus, maaga seotud dokumentide menetlemise protsess) ning kordusanalüüside tegemine omades täiendavat infot ärinõuetest (ajakulu, jõudluse ja ressursi analüüsid).

Edaspidiselt ei oleks selline hüpoteesidel põhinev sõelumismetoodika jätkusuutlik vaid mõistlik oleks kasutada täiendavalt mõnda prognoosivat analüüsitehnikat (*predictive analysis*) ja protsessikaeve tarkvarade poolt pakutavaid analüüsipakette. Kindlasti tuleks aga jätkata protsessi korrapärase logiandmetel põhineva monitoorimisega.

## 6.4 Protsessikaeve projekti jaoks sobivaim tarkvara

Käesolevas töös kasutati läbivalt kolme protsessikaeve tarkvara. Kui Disco ja Celonis on rohkem orienteeritud visuaalselt pakendatud tarkvaralahendusele, siis ProM on rohkem panustanud sisulise poole pidevale arendamisele. Olgugi, et mudeli osas osutus antud protsessikaeve projekti osas sobivaimaks Disco mudel, siis siin võrdleme tarkvarasid kogu projekti ulatuses.

Kasutatud tarkvarade võrdlus teemade kaupa:

- **Andmete import.** Kõik kolm tarkvara on võimelised töötama XES failidega. Disco puhul ei saa suuremahulisteks analüüsideks kasutada prooviversiooni, käesoleva töö puhul kasutati akadeemilist ligipääsu. Celonise puhul ei olnud prooviversioon antud andmemahu (u 2 GB) suurus takistuseks. ProM pakub



erinevaid impordivariante, st saab valida kui suurt osa andmestikust peegeldatakse ja kui suurt osa loetakse sisse. Antud projekti osas toimisid kõik kolm tarkvara võrdselt hästi.

- **Protsessi stabiilsuse kontrollimine.** Sellist analüüsi pakkus ainult ProM.
- **Logide analüüs.** Disco võimaldas logiandmeid vaadata kohe pärast importi. Vastav vaade sisaldas ka erinevate filtrite kasutamise võimalust. Celonise puhul tuli vastava väljavõtte jaoks eraldi vaade genereerida. Nii Disco kui Celonise puhul sai filtreerimist rakendada ka otse mudelivaatest. ProM otseselt tabeli kujul logikirjete vaatamist ei paku, kuid pakub mitmekülgsete vaadetega logide inspekteerimise osa. Samuti on võimalik valida erinevaid logide filtreerimise, klasterdamise, asendamise jms pakette. Antud projektis kasutati logide filtreerimisel ProM-i paketti ettevalmistava osana enne mudeldamist, Disco ja Celonise puhul protsessi analüüsi osas hüpoteeside genereerimisel. Kokkuvõttes võrdne hinnang logide analüüsi osas.
- **Protsessimudeli koostamine.** Protsessimudeli koostamise analüüs on esitatud alapeatükis 6.2. Tarkvarade osas on hinnang analoogne, mis logide analüüsiskis. Kui on vaja kiiret ja visuaalset vaadet, siis sobivad Disco ja Celonis hästi. Samas kui on soov kasutada rohkem sisulisi lahendusi, siis on selleks ProM väga hea. Käesoleva projekti raames leidis Disco mudel kõige rohkem kasutust.
- **Ajakulu ja jõudluse analüüs.** Ajakulu ja jõudluse analüüsiks pakkusid kõik tarkvarad erinevaid lahendusi alates animatsioonist kuni konkreetsete filtriteni. Kuna antud töös ei olnud see konkreetne analüüs relevantne, siis olid Celonise ja Disco animatsioon ning keskmised ajakulu andmed piisavad.
- **Ressursi analüüs.** Sotsiaalvõrgustikke võimaldab modelleerida kolmest tarkvarast ainult ProM. Antud juhul, ebapiisavate andmete tõttu, ei andnud see konkreetne rakendus informatiivset tulemust. Celonis pakub läbi graafikute esmase ülevaate seotud osapooltest ning ressursikulust ning antud ülesande iseloomu arvestades oli see seekord täiesti piisav.
- **Üldine kasutajamugavus.** Disco ja Celonis on kindlasti teemakaugematele kasutajatele mugavamad. Disco eeliseks on hea lisamaterjali olemasolu, Celonise plussiks veebipõhine rakendus. Kui on aga soovi veidigi tõsisemalt valdkonnaga tegeleda, siis tuleks kindlasti ProM-i ja tema poolt pakutavatesse lahendustesse

süveneda. Ka siin on pakettide autorid välja andnud hulgaliselt selgitavad materjale ning teadusartikleid.

Kõigile kolmele tarkvarale on olemas oma kindel sihtgrupp. Enne kindla tarkvara välja valimist on soovitav teha proovitöö kõiki kolme tarkvara kasutades. Vähegi pikemaajaliste projektide puhul võiks kasutada paralleelselt kahte tarkvara, milleks oleksid ProM ja kas Disco või Celonis (vastavalt võimalustele ja eelistustele). Käesoleva protsessikaeve projekti osas said kõige rohkem kasutust ProM ja Celonis.

## 7 Kokkuvõte

Käesolevas töös järgiti kahte eesmärki: teostada võrdlus erinevate protsessimudelit genereerivate algoritmide vahel, arvestades valitud protsessikaeve tarkvarasid, ning protsessikaeve projekti läbimine ja selle tulemusena vaadeldava äriprotsessi logide põhjal mõjutegurite leidmine soovimatute lõpptulemuste realiseerumisel.

Mõlema seatud eesmärgi elluviimine tugines teineteisele: algoritmide testimine viidi läbi protsessikaeve projekti ühe osana, samas võrdluses leitud protsessimudel oli vajalik projekti uurimiseesmärgi saavutamiseks. Töö käigus analüüsiti läbi logide struktuur ja erinevad mustrid, koostati ja testiti erinevaid mudeleid, analüüsiti protsessi nii protsessi terviku kui juhtumi vaatest ning testiti võimalikke mõjustsenaariume.

Käesoleva magistritöö peamised tulemused on:

- erinevate protsessimudelite koostamise algoritmide võrdlus, arvestades konkreetse protsessikaeve tarkvara pakutud võimalusi;
- protsessikaeve projekti etappide kaardistus ning näitlik projekti läbimine, kasutades reaalseid andmeid ning tuues välja reaalseid tekkida võivaid kitsaskohti;
- tagasiside protsessi parendamiseks ja täiendavate analüüside teostamiseks, mis võimaldaks edaspidi varem tuvastada ebasobivaid lõpptulemusi;
- protsessikaeve tarkvarade võrdlus lähtuvalt konkreetsest läbitud projektist.

Magistritöös püstitatud eesmärgid said täidetud ning töö autor plaanib kindlasti ka töös tutvustatud protsessikaeve tehnikaid oma igapäevases töös kasutama hakata.

## Kasutatud kirjandus

- [1] European Agricultural Guarantee Fund. Wikipedia. [WWW]  
[https://en.wikipedia.org/wiki/European\\_Agricultural\\_Guarantee\\_Fund](https://en.wikipedia.org/wiki/European_Agricultural_Guarantee_Fund)  
(03.05.2018)
- [2] IEEE 1849-2016 XES Standard. [WWW]  
<http://www.xes-standard.org/> (03.05.2018)
- [3] A.J.M.M. Weijters, W.M.P. van der Aalst, A.K. Alves de Medeiros. Process Mining with the HeuristicsMiner Algorithm. 2006.
- [4] W. M. P. van der Aalst. Process Mining: Data Science in Action. Springer, 2016.
- [5] C.W. Günther, E. Verbeek. XES. Standard Definition. 2014 [WWW]  
[http://www.xes-standard.org/\\_media/xes/xesstandarddefinition-2.0.pdf](http://www.xes-standard.org/_media/xes/xesstandarddefinition-2.0.pdf)  
(03.05.2018)
- [6] An Introduction to the XES Standard. Fluxicon weblog. [WWW]  
<https://fluxicon.com/blog/2010/09/intro-to-xes/> (03.05.2018)
- [7] ProM Video course: Event logs. [WWW]  
<https://www.futurelearn.com/courses/process-mining/0/steps/15621>  
(03.05.2018)
- [8] ProM 6.7 [WWW]  
<http://www.promtools.org/doku.php> (03.05.2018)
- [9] Disco [WWW]  
<https://fluxicon.com/disco> (03.05.2018)
- [10] Celonis [WWW]  
<https://www.celonis.com/> (03.05.2018)
- [11] A. Devi , T. Kumudavalli, M.V. Sudhamani. An Informative and Comparative Study of Process Mining Tools. International Journal of Scientific & Engineering Research Volume 8, Issue 5. 2017.
- [12] Inspecting and Cleaning an Event Log. [WWW]  
<http://www.promtools.org/doku.php?id=tutorial:preprocessing> (03.05.2018)
- [13] H.M.W. Verbeek, R. P. J.C. Bose. ProM 6 Tutorial. 2010.
- [14] Introduction to Process Mining with ProM. [WWW]  
<https://www.futurelearn.com/courses/process-mining> (03.05.2018)
- [15] 14th International Workshop on Business Process Intelligence 2018 [WWW]  
<http://www.win.tue.nl/bpi/doku.php?id=2018:challenge> (03.05.2018)

- [16] data experts [WWW]  
<https://www.data-experts.de/> (03.05.2018)
- [17] R.P. J. C. Bose, W. M.P. van der Aalst, I. Zliobaite, M. Pechenizkiy. Handling Concept Drift in Process Mining. 2011.
- [18] E. Povalyaeva, I. Khamitov, A. Fomenko. BPIC 2017: Density Analysis of the Interaction With Clients. 2017.
- [19] P. Weber, B. Bordbar, P. Tino, B. Majeed. A Framework for Comparing Process Mining algorithms. 2010.
- [20] W.M.P. van der Aalst, A. Adriansyah, B. van Dongen. Replaying History on Process Models for Conformance Checking and Performance Analysis. 2011.
- [21] Workflow Soundness. [WWW]  
[http://mlwiki.org/index.php/Workflow\\_Soundness](http://mlwiki.org/index.php/Workflow_Soundness) (03.05.2018)
- [22] W. M. P. van der Aalst. Discovering Petri Nets. [WWW]  
[http://www.processmining.org/\\_media/presentations/rozenberg\\_mining.pdf](http://www.processmining.org/_media/presentations/rozenberg_mining.pdf)  
(03.05.2018)
- [23] ProM Tips — Which Mining Algorithm Should You Use? [WWW]  
<https://fluxicon.com/blog/2010/10/prom-tips-mining-algorithm/> (03.05.2018)
- [24] ProM Video course: Alpha miner in ProM. [WWW]  
<https://www.futurelearn.com/courses/process-mining/0/steps/15637>  
(03.05.2018)
- [25] ProM Video course: Heuristics Miner. [WWW]  
<https://www.futurelearn.com/courses/process-mining/0/steps/15639>  
(03.05.2018)
- [26] ProM Video course: Heuristics miner in ProM. [WWW]  
<https://www.futurelearn.com/courses/process-mining/0/steps/15640>  
(03.05.2018)
- [27] ProM Video course: Inductive miner in ProM. [WWW]  
<https://www.futurelearn.com/courses/process-mining/0/steps/15643>  
(03.05.2018)
- [28] S.J.J. Leemans, D. Fahland, W.M.P. van der Aalst. Process and Deviation Exploration with Inductive visual Miner. 2014.
- [29] S.J.J. Leemans. Inductive visual Miner. Manual. 2017.
- [30] ProM Video course: Fuzzy miner in ProM. [WWW]  
<https://www.futurelearn.com/courses/process-mining/0/steps/15645>  
(03.05.2018)
- [31] Z Test: Definition & Two Proportion Z-Test. [WWW]  
<http://www.statisticshowto.com/z-test/> (03.05.2018)
- [32] RStudio. [WWW]  
<https://www.rstudio.com/> (03.05.2018)

## Lisa 1 – Protsessiga seotud dokumentitüüpide loend

**Tabel 8.** BPIC 2018 protsessis kasutatavad dokumentide tüübid selgitustega.

Dokumendi tüüp	Alamprotsess	Selgitus
Payment application	Main Application Objection Change	Taotlusedokument maksete tegemiseks, tavaliselt iga aasta
Entitlement application	Main Objection Change	Otsetoetuste taotlemise dokument, tavaliselt luuakse uue rahastamisperioodi alguses
Inspection	On-Site Remote	Dokument, mis sisaldab kohapealsete või kaugkontrollide tulemusi
Control summary	Main	Dokument, mis sisaldab erinevate kontrollide kokkuvõtlikke tulemusi
Reference alignment	Main	Dokument, mis sisaldab maatükkide kirjeldamise tulemusi arvestades katastri jms võrdlusinfot
Department control parcels (enne 2017)*	Main	Dokument, mis sisaldab ühe taotleja maatükkide kehtivuse kontrolli tulemusi
Parcel Document (enne 2016)*	Main	Dokument, mis sisaldab kõiki maatükke, mille jaoks toetusi taotletakse
Geo Parcel Document (asendas Parcel document alates 2016 ja Department control parcels alates 2017)*	Main Declared Reported	Dokument, mis sisaldab kõiki maatükke, mille jaoks toetusi taotletakse.

\* Kui dokumendid on agregeeritud, siis kasutatakse nimetust *Geo parcel document*

## Lisa 2 – Logifaili atribuutide loend

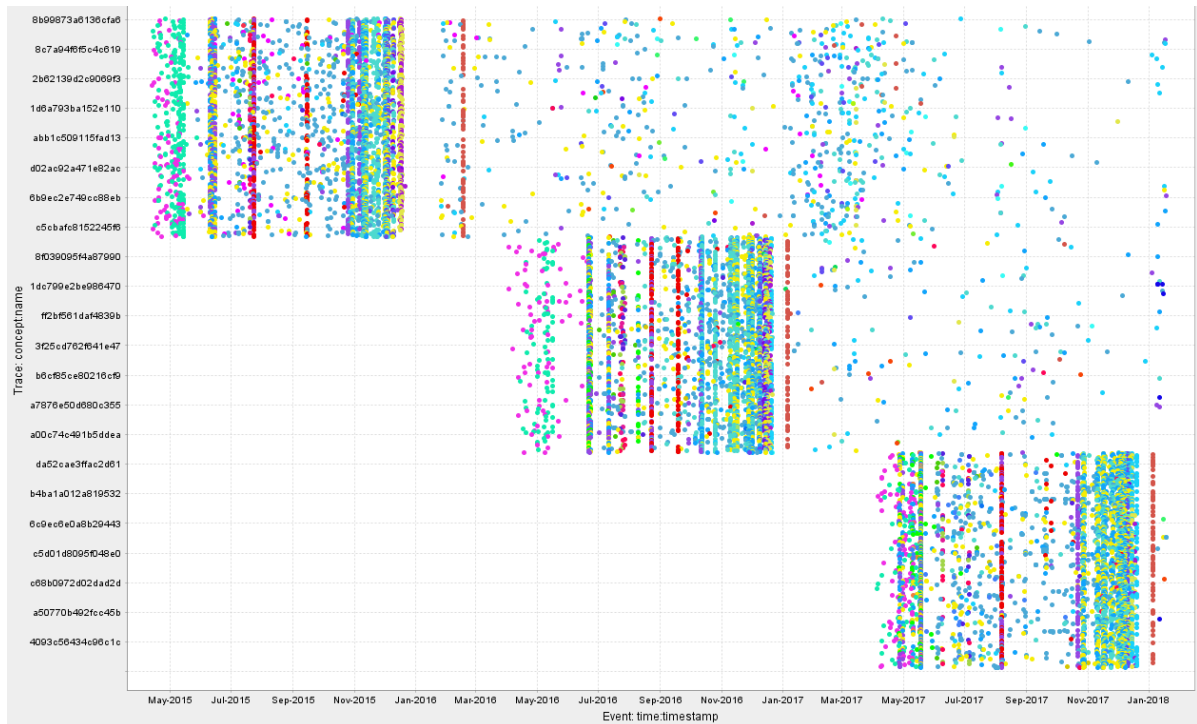
Tabel 9. BPIC 2018 logifailis kasutatud atribuudid selgitustega.

Atribuudi tüüp	Atribuut	Tüüp	Selgitus
Taotlus	program-id	literal	Rahastamisprogrammi sisemine tunnus
Taotlus	concept:name (and application)	literal	Taotluse unikaalne ID
Taotlus	identity:id	UUID	Globaalne taotluse ID (UUID)
Taotlus	Department	literal	Kohaliku osakonna ID
Taotlus	application	literal	Taotleja ID, sama läbi aastate
Taotlus	year	literal	Käesolev aasta
Taotlus	number_parcels	discrete	Maatükkide arv
Taotlus	area	continuous	Maatükkide kogupindala
Taotlus	basic_payment	boolean	Põhitoetuse taotlus
Taotlus	greening	boolean	Rohestamise toetuse taotlus
Taotlus	redistribution	boolean	Ühtse pindalatoetuse taotlus
Taotlus	small farmer	boolean	Väiketootjate taotlus
Taotlus	young farmer	boolean	Toetus alustavatele noortele põllumajandustootjatele
Taotlus	applicant	literal	Taotleja umbisikustatud tunnus
Tuletatud	penalty_{xxx}	boolean	Näitab trahvi põhjusel {xxx}.
Tuletatud	amount_applied{x}	continuous	Summa (EUR), mida taotletakse taotluses. $x=0$ praegune makseprotsess. Kui juhtum vajab osakonna muudatust või taotleja vaidlustust, suurendatakse seda arvu iga makse kohta 1 võrra.
Tuletatud	payment_actual{x}	continuous	Tegelikult saadud summa (EUR). $x$ -i tähendus siin ja edaspidi sama, mis eespool kirjeldatud.
Tuletatud	penalty_amount{x}	continuous	Osakonna kohaldatav karistus, näiteks maatükkide suuruse üledeklareerimise tõttu. Olemas ainult siis, kui <i>penalty_application</i> on tõene.
Tuletatud	risk_factor	continuous	Mittekohustuslik, manuaalselt lisatud riskihindamise tegur.
Tuletatud	cross_compliance	continuous	Karistusmäär, kui taotlus ei vasta nõuetele
Tuletatud	selected_random	boolean	Kas taotlus on valitud kontrolli juhuslikult?
Tuletatud	selected_risk	boolean	Kas taotlus on valitud kontrolli riskihindamise põhjal?

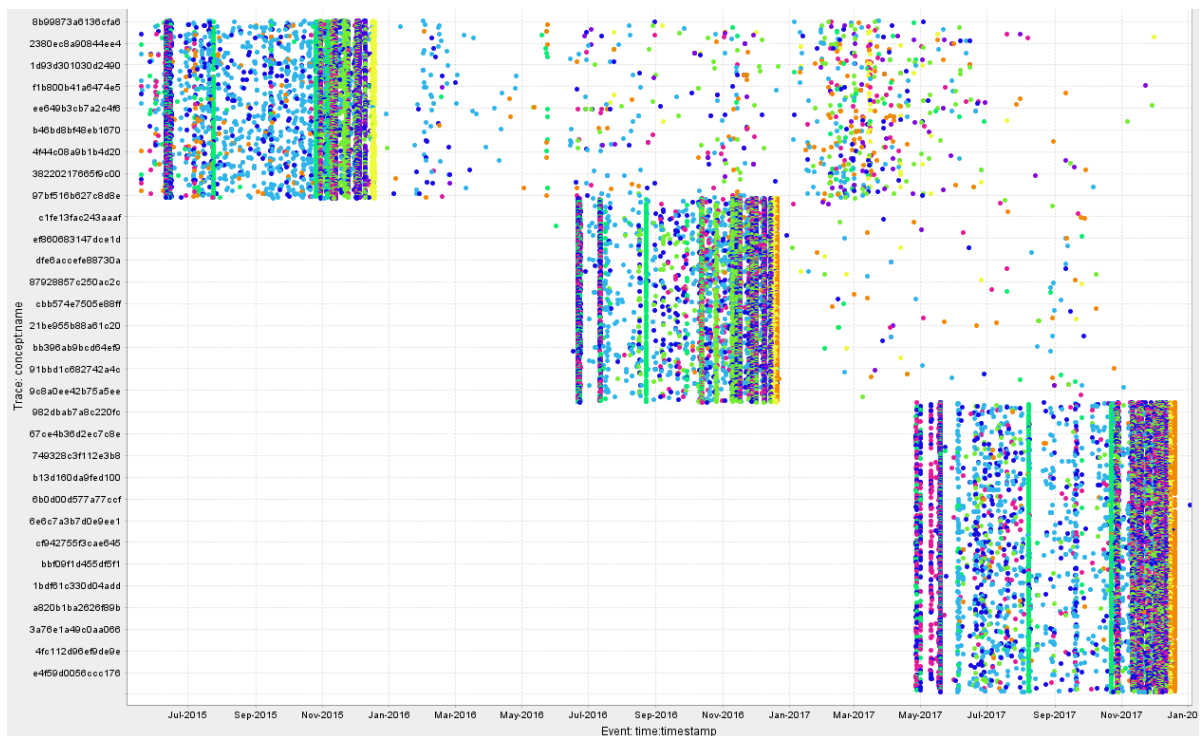
Atribuudi tüüp	Atribuut	Tüüp	Selgitus
Tuletatud	selected_manually	boolean	Kas taotlus on valitud kontrolli mauaalselt?
Tuletatud	rejected	boolean	Terve taotlus on tagasi lükatud
Sündmus	success	boolean	Näitab sündmuse edukat sooritust
Sündmus	concept:name (and activity)	literal	Tegevus, mida antud sündmuse käigus teostati
Sündmus	docid	literal	Dokumendi sisemine ID
Sündmus	doctype	literal	Dokumendi tüüp
Sündmus	eventid	literal	Sündmuse sisemine ID (võib olla tuletatud sündmuse korral 0)
Sündmus	lifecycle:transition	literal	Väärtus <i>complete</i> kõigi sündmuste jaoks.
Sündmus	note	literal	Sündmusega seotud täiendav vaba tekstiväli. <i>none</i> kui väli ei ole täidetav.
Sündmus	org:resource	literal	Näitab sündmuse eest vastutavat ressursi
Sündmus	subprocess	literal	Alamprotsess, millesse sündmus kuulub. Iga dokument on jagatud ühe või mitme alamprotsessi juurde.
Sündmus	time:timestamp	timestamp	Sündmuse toimumise aeg (ajatempel). Identsete ajatemplitega sündmuste järjestamine ei ole võimalik. Mõned ajatemplid on sisestatud käsitsi ja võivad sisaldada kirjavigu.
Sündmus	docid_uuid	UUID	Globaalne unikaalne dokumendi ID. On üksühene seos docid ja docid_uuid vahel.
Sündmus	identity:id	UUID	Iga sündmuse globaalne ID. Asendab atribuudi eventid, kui see pole unikaalne (nt null). Sündmustel on unikaalne id-attribuut kõigis failides.



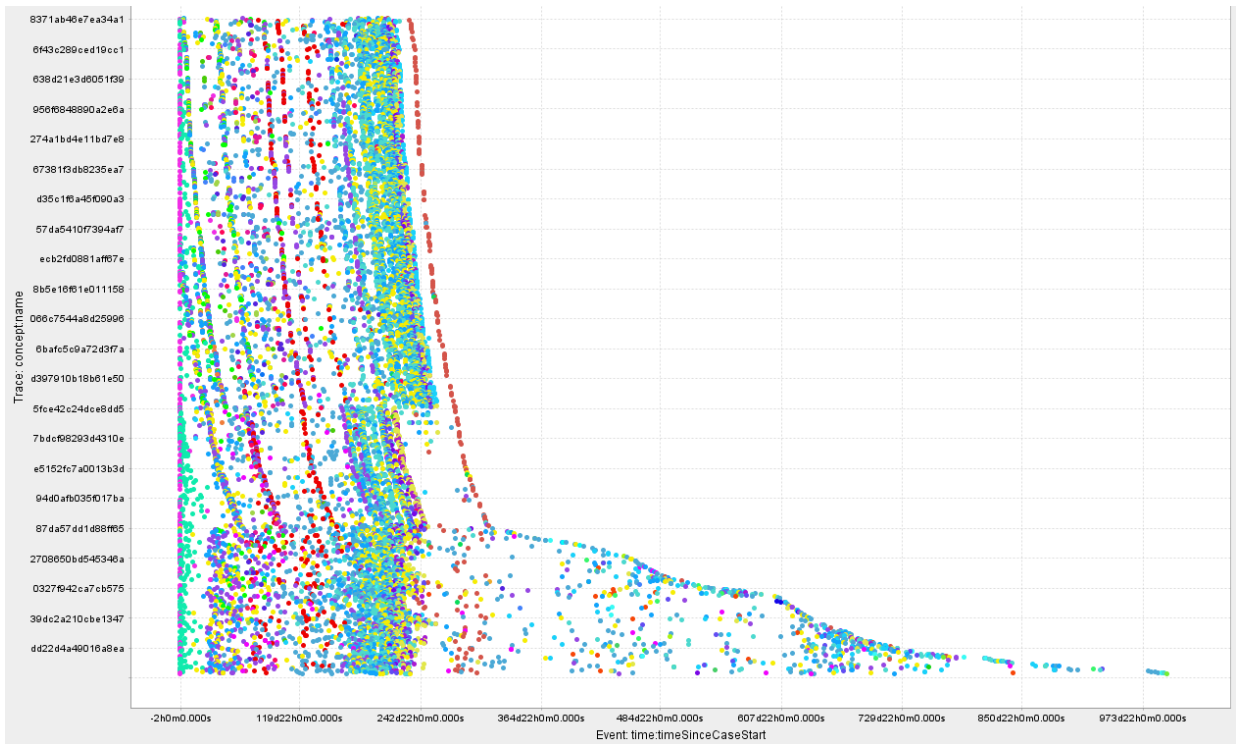
## Lisa 3 – Algandmete ja filtreeritud logide graafilised esitused



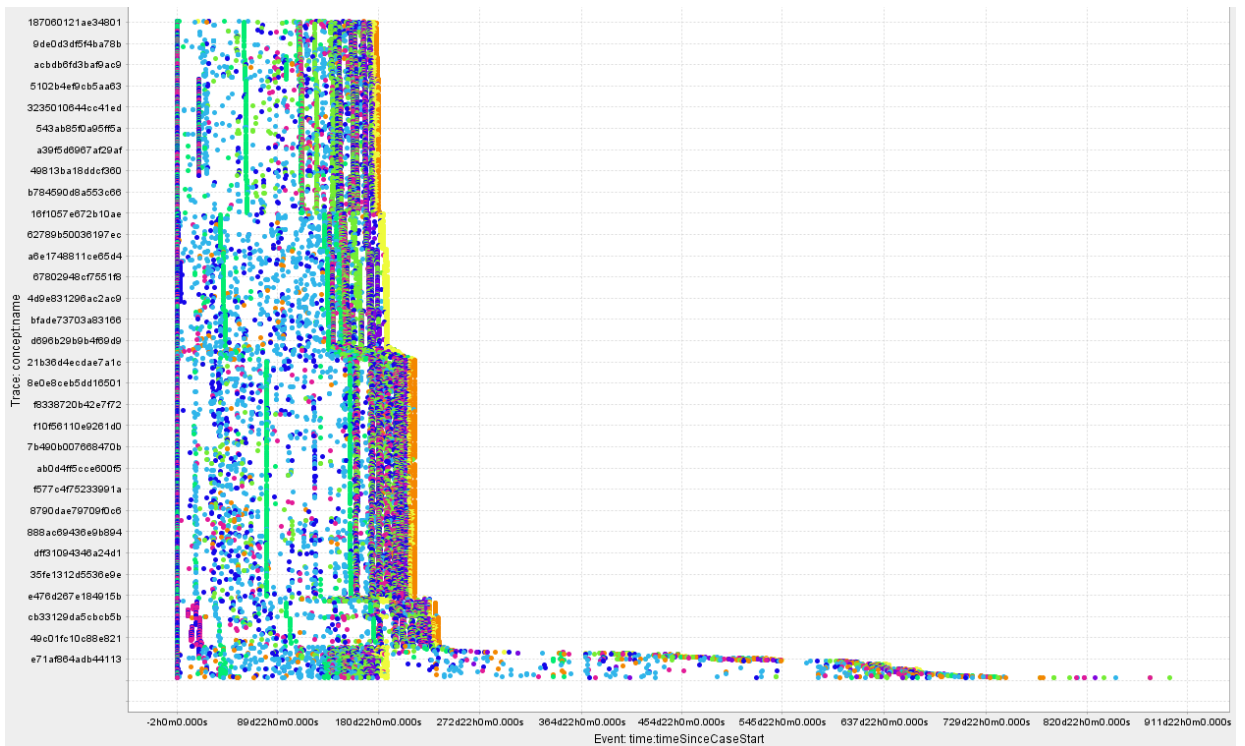
Joonis 19. Sündmuste toimumine vastavalt ajatemplile.



Joonis 20. Filtreeritud logis sündmuste toimumine vastavalt ajatemplile.



Joonis 21. Logide järjestus taotlusprotsessi pikkust arvestades.



Joonis 22. Filtreeritud logide järjestus taotlusprotsessi pikkust arvestades.

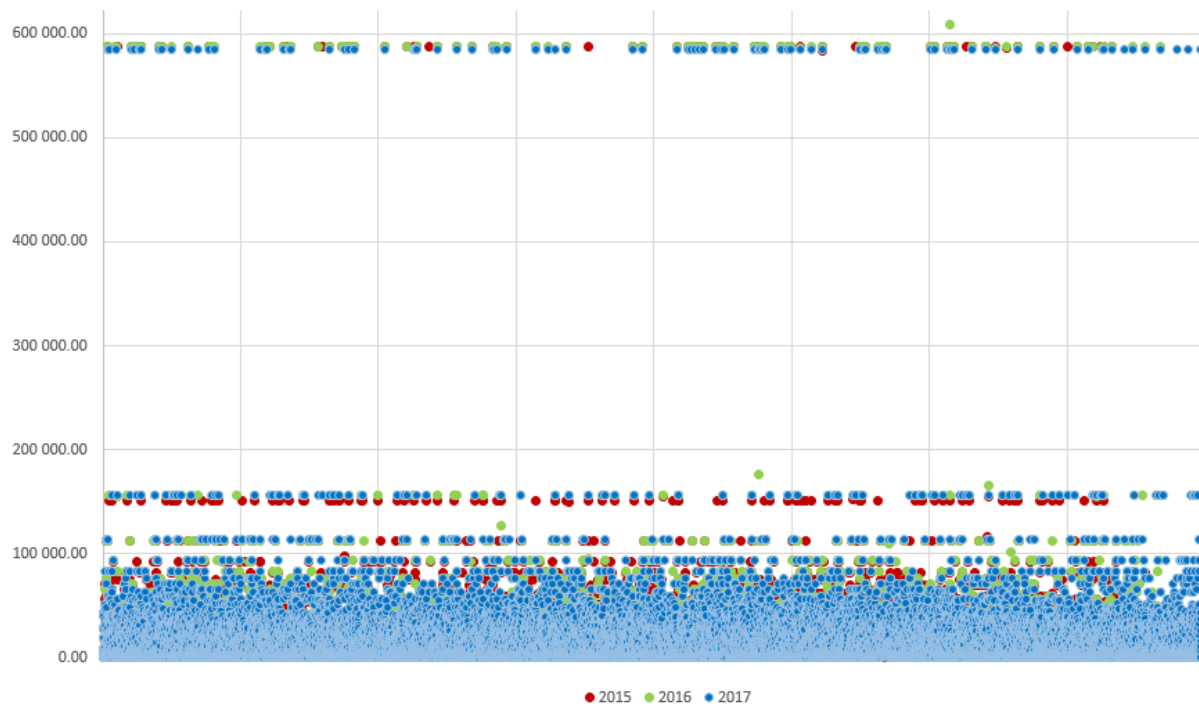
## Lisa 4 – Peamiste sündmuste ja dokumentide esinemise sagedused

Tabel 10. Olulisemates alamprotsessides sündmuste ja dokumentide sagedused.

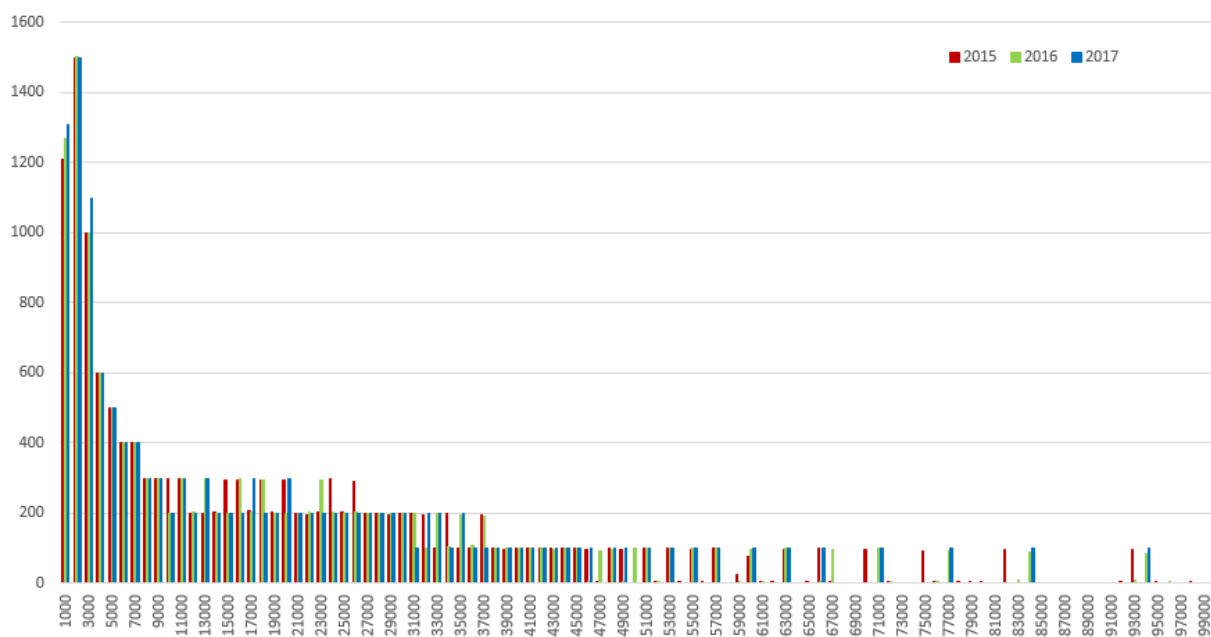
Sündmus	Sagedus	Osakaal	Dokument	Alamprotsess	Sagedus	Osakaal
calculate	466 141	18.54%	Payment Application	Application	256 611	10.21%
			Geo parcel document	Reported	94 884	3.77%
			Entitlement application	Main	71 977	2.86%
			Geo parcel document	Declared	25 770	1.03%
			Payment Application	Change	7 267	0.29%
			Entitlement application	Change	5 509	0.22%
			Payment Application	Objection	2 484	0.10%
			Entitlement application	Objection	1 639	0.07%
finish editing	456 013	18.14%	Payment Application	Application	138 337	5.50%
			Geo parcel document	Reported	95 300	3.79%
			Geo parcel document	Declared	59 843	2.38%
			Entitlement application	Main	58 751	2.34%
			Geo parcel document	Main	46 145	1.84%
			Control summary	Main	43 808	1.74%
			Payment Application	Change	2 127	0.09%
			Entitlement application	Change	1 984	0.08%
			Payment Application	Objection	922	0.04%
Entitlement application	Objection	659	0.03%			
begin editing	397 133	15.80%	Payment Application	Application	117 495	4.67%
			Geo parcel document	Reported	81 441	3.24%
			Geo parcel document	Declared	59 037	2.35%
			Entitlement application	Main	58 427	2.32%
			Control summary	Main	43 808	1.74%
			Geo parcel document	Main	24 597	0.98%
			Payment Application	Change	4 880	0.19%
			Entitlement application	Change	3 515	0.14%
			Payment Application	Objection	738	0.03%
			Entitlement application	Objection	534	0.02%
save	288 902	11.49%	Geo parcel document	Main	85 465	3.40%
			Reference alignment	Main	37 039	1.47%
			Control summary	Main	29 057	1.16%
			Entitlement application	Main	7 847	0.31%
			Payment Application	Main	5 237	0.21%
initialize	205 082	8.16%	Control summary	Main	43 808	1.74%
			Payment Application	Application	43 809	1.74%
			Reference alignment	Main	43 802	1.74%

			Geo parcel document	Main	43 809	1.74%
			Entitlement application	Main	15 260	0.61%
			Payment Application	Change	4 541	0.18%
			Entitlement application	Change	3 156	0.13%
			Payment Application	Objection	850	0.03%
			Entitlement application	Objection	562	0.02%
decide	87 696	3.49%	Payment Application	Application	65 437	2.60%
			Entitlement application	Main	16 099	0.64%
			Payment Application	Change	2 292	0.09%
			Entitlement application	Change	2 113	0.08%
			Payment Application	Objection	1 048	0.04%
			Entitlement application	Objection	707	0.03%
insert document	83 260	3.31%	Payment Application	Application	43 881	1.75%
			Geo parcel document	Main	15 043	0.60%
			Payment Application	Main	7 325	0.29%
			Entitlement application	Change	6 887	0.27%
			Reference alignment	Main	3 798	0.15%
			Payment Application	Change	2 027	0.08%
			Payment Application	Objection	1 143	0.05%
			Entitlement application	Main	975	0.04%
			Entitlement application	Objection	937	0.04%
			Geo parcel document	Reported	865	0.03%
			Geo parcel document	Declared	141	0.01%
			Control summary	Main	6	0.00%
begin payment	80 200	3.19%	Payment Application	Application	76 657	3.05%
			Payment Application	Change	2 633	0.11%
			Payment Application	Objection	910	0.04%
performed	76 731	3.05%	Reference alignment	Main	43 802	1.74%
			Geo parcel document	Main	29 297	1.17%
mail valid	74 326	2.96%	Payment Application	Application	43 808	1.74%
			Entitlement application	Main	30 518	1.21%
finish payment	45 907	1.83%	Payment Application	Application	43 721	1.74%
			Payment Application	Change	1 643	0.07%
			Payment Application	Objection	543	0.02%
mail income	43 808	1.74%	Payment Application	Application	43 808	1.74%
create	41 563	1.65%	Geo parcel document	Declared	41 563	1.65%
abort payment	34 293	1.36%	Payment Application	Application	32 936	1.31%
			Payment Application	Change	990	0.04%
			Payment Application	Objection	367	0.02%
revoke decision	22 004	0.88%	Payment Application	Application	21 363	0.85%
			Entitlement application	Main	284	0.01%
			Entitlement application	Change	128	0.01%
			Entitlement application	Objection	125	0.01%
			Payment Application	Change	50	0.00%
			Payment Application	Objection	54	0.00%

## Lisa 5 – Taotlussummade jaotus läbi aastate

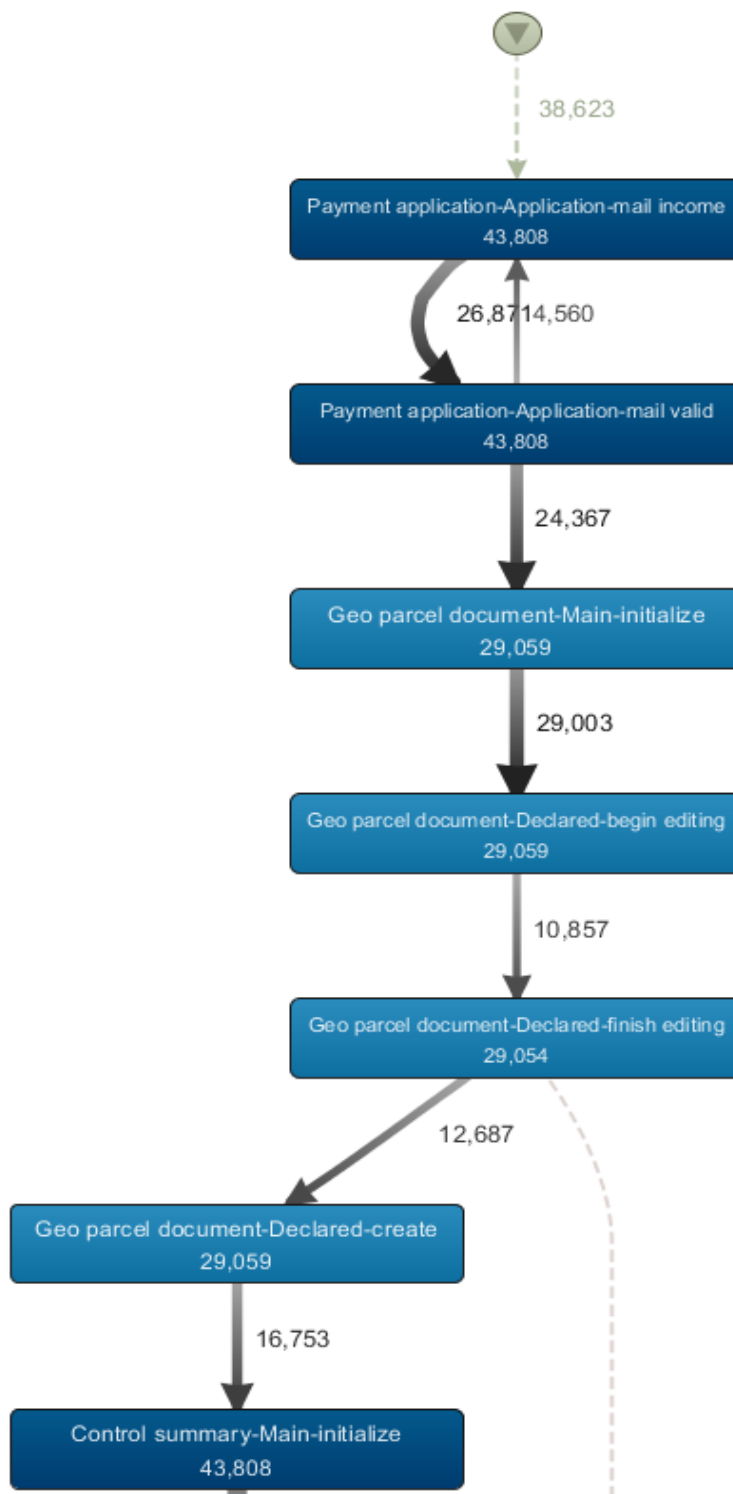


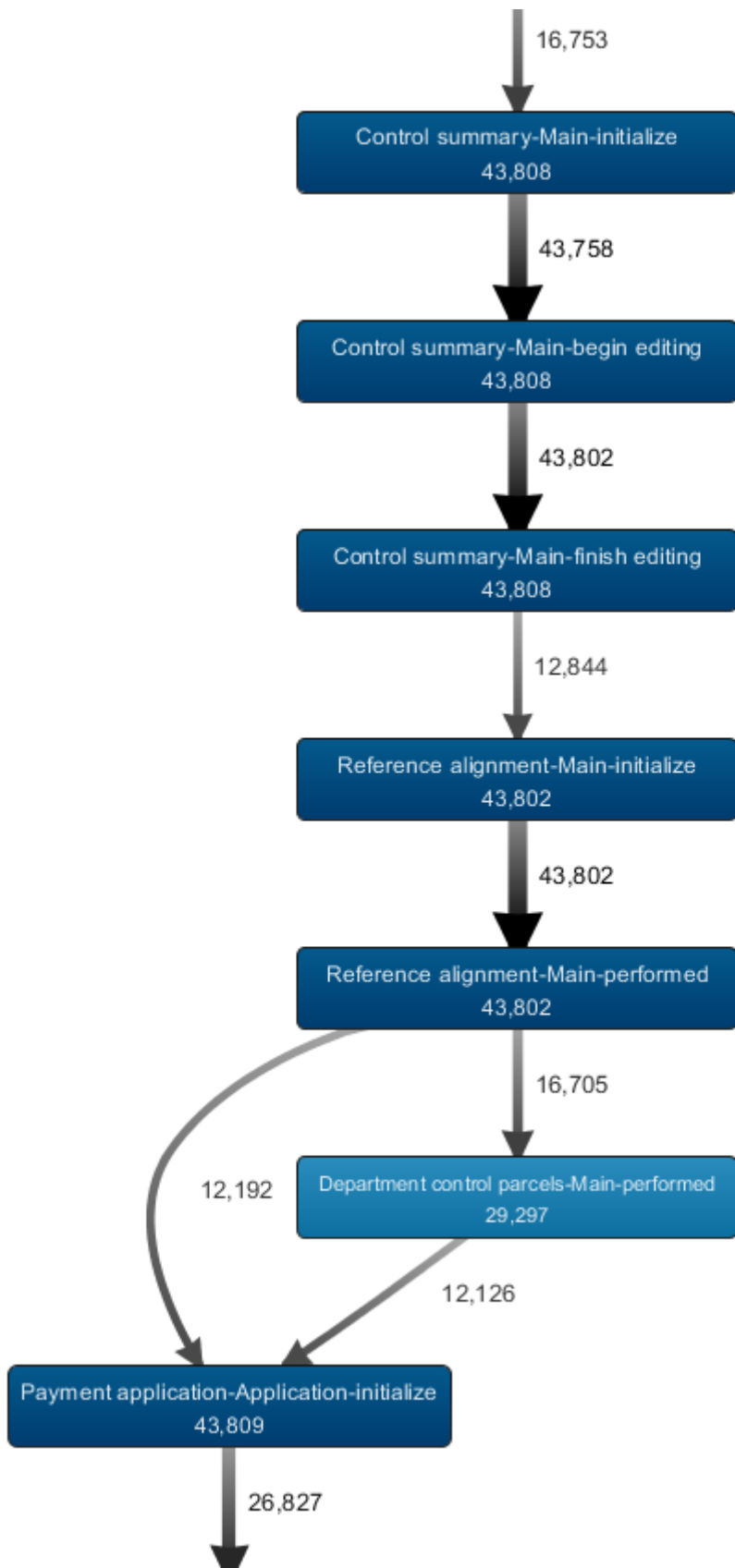
Joonis 23. Taotlussummad aastatel 2015-2017.



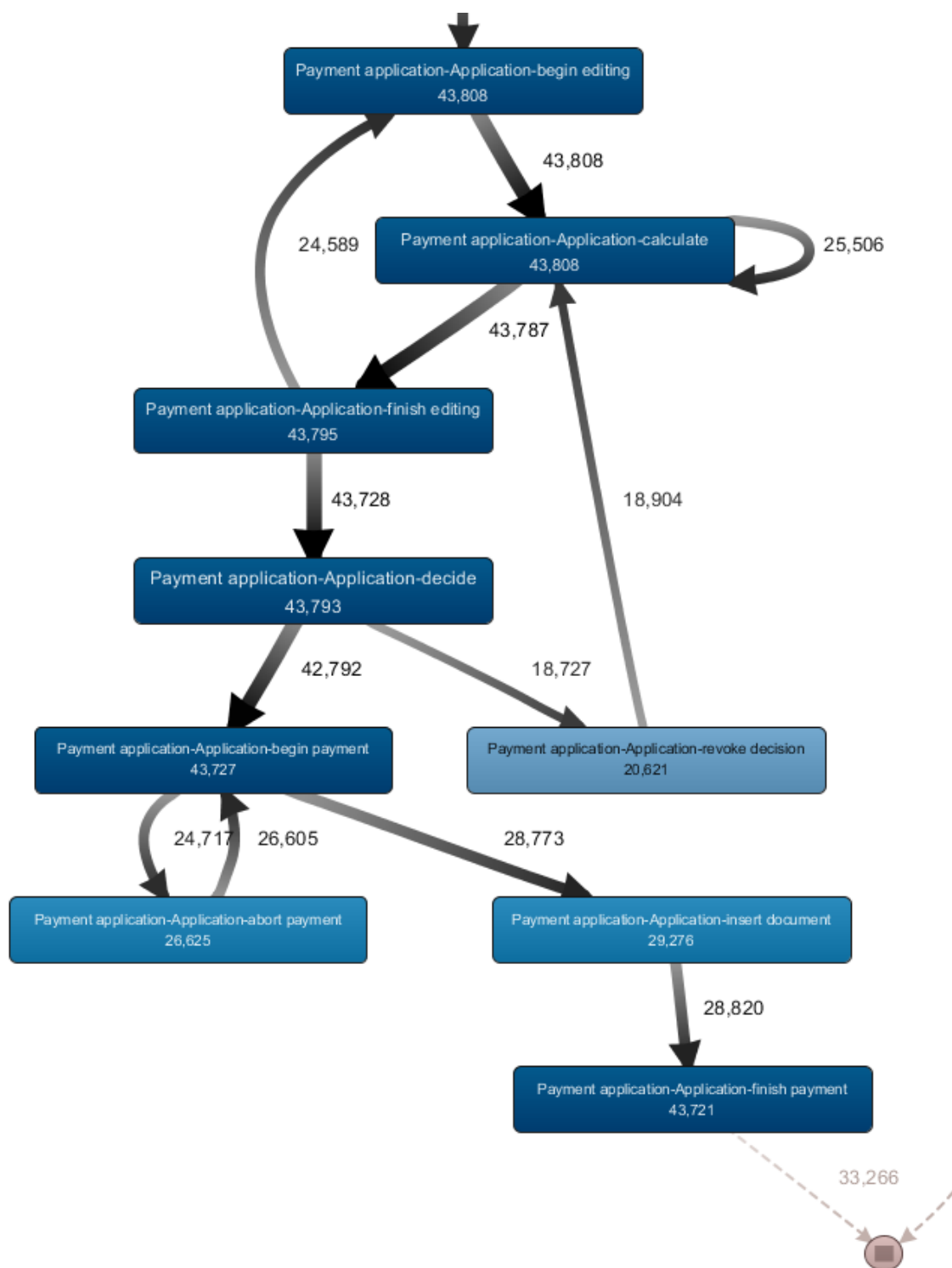
Joonis 24. Kuni 100 000€ taotlussummade jaotus 2015-2017.

## Lisa 6 – Disco poolt genereeritud protsessimudel





366

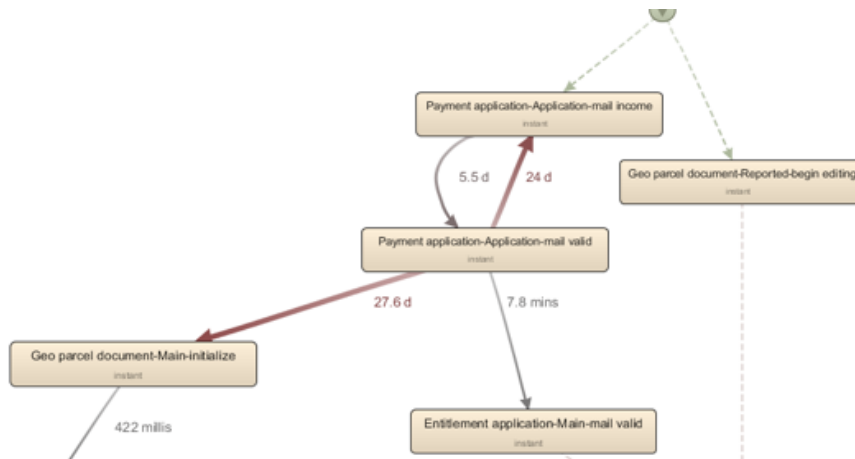


Joonis 25. Discoga genereeritud protsessimudel.

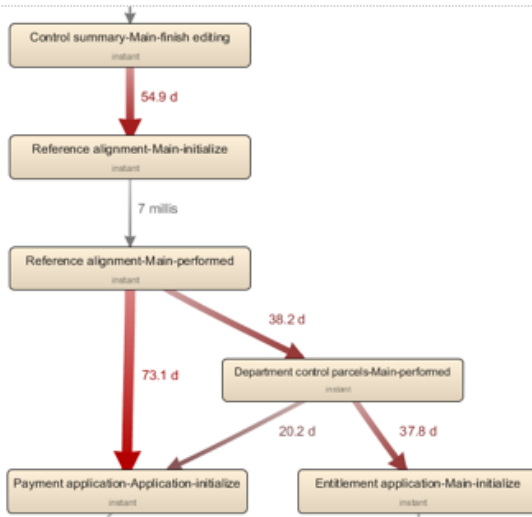


## Lisa 7 – Kõige suurema ajakuluga protsessiosad Disco järgi

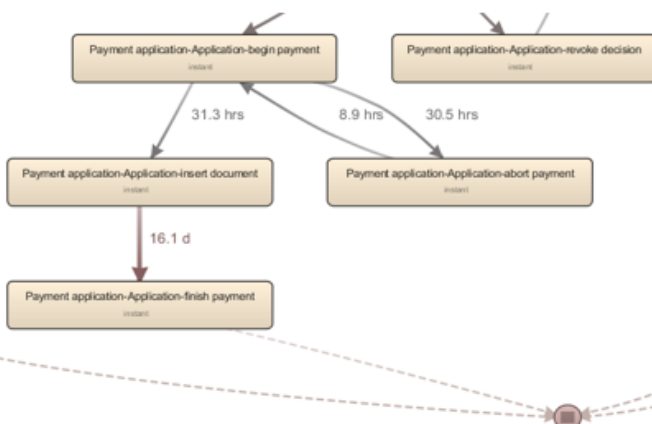
a)



b)



c)



Joonis 26. Kõige suurema ajakuluga protsessiosad Disco järgi.