UNIVERSITY OF TARTU
Institute of Computer Science
Software Engineering Curriculum

Allar Soo

Automated Process Discovery: A Literature Review and a Comparative Evaluation With Domain Experts

Master's Thesis (30 ECTS)

Supervisor: Fabrizio Maria Maggi, PhD
Supervisor: Fredrik Payman Milani, PhD
Supervisor: Andrea Marrella, PhD
Supervisor: Massimo Mecella, Prof

Tartu 2017

Automaatne äriprotsesside avastamine: kirjanduse ülevaade ning võrdlev hindamine koostöös domeeniekspertidega

Lühikokkuvõte: Protsesside kaevandamise meetodid võimaldavad analüütikul kasutada logides talletatud protsesside täitmis informatsiooni, et saada teadmisi talletatud protsesside tegeliku sooritamise kohta. Üks enim uuritud protsesside kaevandamise toiminguid on automatiseeritud protsesside avastamine. Sündmuste logi võetakse sisendina automatiseeritud protsesside avastamise meetodi poolt ning väljundina toodetakse äriprotsessi mudel, mis kujutab juhtumite logis kirjeldatud ülesannete vahelist kontrollvoogu. Viimase kahe kümnendi jooksul on väljapakutud mitmeid automatiseeritud protsessi avastamise meetodeid, kasutades toodetavate mudelite juures silmatorkavalt erinevaid kompromisse mastaabiga kohanemise, täpsuse ning keerukuse vahel. Siiani on automatiseeritud protsesside avastamise meetodid hinnatud mitteüldistaval (*ad-hoc*) viisil, kus erinevad autorid kasutavad erinevaid andmestike, eksperimentide seadistusi, hindamismeetmeid ning alustõdesid, mis viivad tihti võrdlematute tulemusteni ning mõnikord suletud andmestike kasutamise tõttu ka mittetaastoodetavate tulemusteni. Eelpool nimetatud mõistes sooritatakse antud magistritöö raames süstemaatiline kirjanduse ülevaade automatiseeritud protsesside avastamise meetoditest ning ka süstemaatiline hindav võrdlus olemasolevate automatiseeritud protsesside avastamise meetodite kohta koostöös domeeniekspertidega kasutades reaalset sündmuste logi rahvusvahelisest tarkvara ettevõttest ning nelja kvaliteedi näitajat. Kirjanduse ülevaate ning hindamise tulemused tõstavad esile puudujääk ning seni uurimata kompromisse valdkonnas nelja äriprotsessi mudeli kvaliteedi näitaja kontekstis. Antud magistritöö tulemused võimaldavad teaduritel parandada puudujäägid automatiseeritud protsesside avastamise meetodites ning samuti vastatakse küsimusele protsesside avastamise tehnikate kasutatavuse kohta tööstuses.

Võtmesõnad: äriprotsesside kaevandamine, äriprotsesside avastamine, kirjanduse ülevaade, ekspertidega hindamine

CERCS: P170 - Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Automated process discovery: A literature review and a comparative evaluation with domain experts

Abstract:

Process mining methods allow analysts to use logs of historical executions of business processes in order to gain knowledge about the actual performance of these processes. One of the most widely studied process mining operations is automated process discovery. An event log is taken as input by an automated process discovery method and produces a business process model as output that captures the control-flow relations between tasks that are described by the event log. Several automated process discovery methods have been proposed in the past two decades, striking different tradeoffs between scalability, accuracy and complexity of the resulting models. So far, automated process discovery methods have been evaluated in an ad hoc manner, with different authors employing different datasets, experimental setups, evaluation measures and baselines, often leading to incomparable conclusions and sometimes unreproducible results due to the use of non-publicly available datasets. In this setting, this thesis provides a systematic review of automated process discovery methods and a systematic comparative evaluation of existing implementations of these methods with domain experts by using a real-life event log extracted from a international software engineering company and four quality metrics. The review and evaluation results highlight gaps and unexplored tradeoffs in the field in the context of four business process model quality metrics. The results of this master thesis allows researchers to improve the lacks in the automated process discovery methods and also answers question about the usability of process discovery techniques in industry.

Keywords: process mining, process discovery, literature review, domain experts evaluation

CERCS: P170 - Computer science Computer science, numerical analysis, systems, control

Contents

1 Introduction

Today's information systems tend to maintain detailed trails of supported business processes, including records of key process execution events, such as case creation or execution of a task within a case that is ongoing. Process mining techniques allow to extract details about the as-is (the actual performance of a process) from collections of such event records. These event records are also known as event logs [vdAWM04]. So an event log is a set of traces, where each trace itself is a sequence of events related to a given case (for example handling of an incident is a case).

Process mining is a bridge between data mining and traditional model-driven Business Process Managemen [vdAAdM+11]. Process mining is composed of three main branches: *i*) process discovery, *ii*) conformance checking, and *iii*) process enhancement.

Out of these three main branches, automated process discovery is the most widely studied process mining operation. An event log is taken as input, and a business process model is produced as output, where usually the control-flow relations between tasks that are observed in or implied by the event log are captured. To be useful, the automatically discovered models must accurately reflect the behaviour from the log. So the model needs to be fit (should be able to parse all the traces in the log), general (should be able to parse traces that are not from the log, but could happen during process reflected from the log), precise (should not allow traces that don't belong to the process reflected from the log), and simple (expressed via model complexity metrics). Thus the methods need to balance between these four metrics (fitness, generalization, precision and simplicity).

The problem of automated discovery of process models from event logs has been intensively researched in the past two decades. Despite a rich set of proposals, state-of-the-art automated process discovery methods suffer from two recurrent deficiencies when applied to real-life logs [WBVB12]: (i)they produce large and spaghetti-like models; and (ii) they produce models that either poorly fit the event log (low fitness) or grossly over-generalize it (low precision or low generalization). Moreover there is no concrete overview of the work done. The latest review was done in year 2011 by De Weerdt et al. in [WBVB12]. Striking a trade-off between these quality dimensions in a robust manner has proved to be a difficult problem and with this thesis we want to find out, if the situation in the field is improved and are the produced outputs usable for the business people.

Hence, this thesis aims to fill this gap by: *i*) providing a systematic review of automated process discovery methods; and *ii*) a comparative evaluation of three

implementations of representative methods, using an real-life event log from an international software engineering company, four quality metrics covering all four dimensions mentioned before (fitness, precision, generalization and complexity), and feedback from the domain experts (users and owners of the process).

In addition to real-life logs, there also synthetic and artificial event logs. So, three kinds of logs are possible: **(i) real-life logs** containing behaviour recorded from the real-life process; **(ii) synthetic logs**, which contain behaviour produced automatically from a (real-life) process model; **(iii) artificial logs**, which contain behaviour automatically extracted from a non-real-life process model, or by manually creating events.

The main research question of this thesis is **"Are automated business process discovery methods acceptable in the industry and which automated business process discovery method is the best in domain experts opinion?"**. To answer this questions, at first an overview of the topic is needed and starting from this overview an user evaluation with domain experts is needed for assessing their opinion. Hence, the outcomes of this research are a classified inventory of automated process discovery methods (a taxonomy) and an analysis of automated process discovery methods in industry.

The rest of the thesis is structured as follows. Section 2 presents the systematic literature review methodology. Section 3 classifies the approaches identified in the review, while Section 4 presents the discussion of research questions. Section 5 introduces and discusses the evaluation and it's results. Section 6 refers to the previous reviews and comparative studies in the field. Finally, Section 7 concludes the paper and outlines future work directions.

2 Methodology for the Literature Review

To understand what has been done in the field of process discovery, a systematic literature review (SLR) was performed. An SLR consists of identifying, evaluating and interpreting research done in a specific domain. To perform such an SLR, a protocol was created and followed. It was inspired from scientific, rigorous and replicable approach as specified by Kitchenham in [Kit04].

Based on that, first research questions were specified (2.1), then research string was created (2.2) and data sources were selected (2.3). After that the author specified inclusion and exclusion criteria (2.4), methods for articles quality assessment (2.5), study selection (2.6), data extraction (2.7) and data analysis (2.8).

2.1 Research Questions Formulation

The goal of this SLR is to identify and analyse studies related to the (business) process discovery. SLR in this paper focus on the approaches that produce model from data (e.g. event log). To serve this purpose, the following research questions were created:

- **RQ1**: Which are the existing approaches that deal with process discovery?
- **RQ2**: Which kinds of process model (i.e., imperative, declarative or hybrid) are discovered by the existing approaches?
- **RQ3**: Which process constraints (e.g. loop, XOR, parallel) are inferred by the models generated?
- **RQ4**: What tools exist to perform process discovery?
- **RQ5**: How the existing approaches have been evaluated?
- **RQ6**: In which domains have existing process discovery approaches been applied?

The RQ1 is used to identify only the articles in the business process discovery domain. RQ2 is meant for identifying the output of an approach. With RQ3 we assessed the expressivity of an approach. With RQ4 we find the tools where the approaches have integrated into. With RQ5 we look at the details about evaluation and with RQ6 details about the domain of application (e.g. a real-life hospital) of an approach are captured. This information is further used to select approaches for testing.

2.2 Search String

In order to perform search over data sources (2.3), a search string must be created. Since the field of business process discovery is broad, the first search string created was **"process discovery"**. This one covers majority of the domain. Since process could be a synonym for workflow or discovery could be be a synonym for learning, three additional search strings were created. So the following query words were used:

- **"process discovery"**
- **"workflow discovery"**
- **"process learning"**
- **"workflow learning"**

This ends up in four different queries over a database.

2.3 Data Source Selection

The defined query words are applied to relevant databases to find studies that are related to the business process discovery. Following seven on-line libraries were used:

- Scopus
- Web of Science
- IEEE Xplore
- ACM Digital Library
- SpringerLink
- ScienceDirect
- Google Scholar

These sources include the information about the most relevant journals, conferences and workshops where the business process discovery community publishes its research.

In Scopus, Web of Science, IEEE Xplore, ACM, SpringerLink and ScienceDirect the search was done by using the default search provided by the search engine. Afterwards Google Scholar was used with search over full text of article to ensure that potentially relevant studies weren't excluded.

2.4 Inclusion and Exclusion Criteria

To select relevant studies for the SLR, inclusion and exclusion criteria must be identified. The **inclusion criterion** is the following:

- The paper is about process discovery, describes a process discovery technique and is newer than the latest paper in [WBVB12]

The following **exclusion criteria** were created:

- The article describes how to use a method, but the paper proposes no improvements to the method
- The article is about conformance checking or process enhancement
- The paper is a non-peer reviewed publication
- The study is not presented in English
- The study is not accessible for free through the standard university library proxy service of the University of Tartu
- If a study is an improvement of previous studies, then the previous ones are discarded.
- The approach does not provide any implementation (if the claimed implementation is not freely available, the proposal is still considered as implemented)

If any of the above exclusion criteria is matched, the study is dropped.

2.5 Quality Assesment

Since one of the goals of this paper is to assess the quality of the implemented approaches dealing with process discovery, the following metrics were created:

- **Log type**
- **Number of logs used**
- **Log sizes**
- **Type of experiments**
- **Type of validation**
- **Comparison to the state-of-the art**

For each of these possible metric, the values are LOW/MEDIUM/HIGH. For log type LOW means the usage of only artificial logs (logs, which contain behaviour automatically extracted from a non-real-life process model, or by manually creating events), MEDIUM means the usage of artificial and synthetic (logs, which contain behaviour produced automatically from a (real-life) process model) logs, and HIGH means the usage of real-life logs. For synthetic or artificial the number of logs is LOW, when it less or equal to 13, MEDIUM if it is between 14 to 26, and HIGH, if it is more than 26. For real-life logs MEDIUM is scored if up to 2 logs are used, and HIGH is scored, if more than 2 logs are used. For logs generated from procedural models size value is LOW when there are up to 825 traces, MEDIUM when 826 to 1650 traces, and HIGH when more than 1650 traces. For log generated from declarative models LOW is scored when there is up to 1056 traces, MEDIUM is scored when there is 1057 to 2112 traces, and HIGH is scored when there is more than 2112 traces. Real-life log size is evaluated LOW when there is up to 4712 traces, MEDIUM when there is 4813 to 9424 traces, and HIGH when more than 9424 traces.

These ranges were obtained by plotting the distribution of the number of logs used and the number of traces in the papers separately and then selecting the thresholds. So all the values that would be over a selected subjective maximum would also be considered as HIGH. Thus, for artificial, synthetic or real-life log size, and for artificial or synthetic number of logs used, the maximum value that wouldn't mess up scales, was considered as 100%. So the range for LOW is from 0 to 33% of maximum, MEDIUM is as 33 to 66% and HIGH as above of 66%. For procedural artificial and synthetic log size the sub-maximal value is 2500, for real-life log size the sub-maximal values is 14279 and for declarative artificial and synthetic log size sub-maximal value is 3200. For artificial and synthetic number of logs used, the sub-maximal value is 40.

Type of experiments value is LOW when tests are made only with noise-free log, only with complete log or all the logs used are of the same size. MEDIUM is scored when tests are done with noisy log, incomplete log, log sizes are varying or real-life log is used. HIGH is scored when three aspects are fulfilled (e.g. tests done with noise-free, noisy and varying size of logs).

Type of validation value is LOW when only static overview of features is present or a illustrative example on log with one or two traces is present, MEDIUM when tests are done by using data as input and the output is analysed, HIGH when output is put in the context of the state-of-the-art.

Comparison to the state-of-the-art value is LOW when it is not done, MEDIUM when only static comparison of features is present, HIGH when compared through testing.

For a paper to be present in testing, it must score at least two HIGH ratings from triple of (log type, number of logs used, log type), one HIGH from tuple (type of experiments, type of validation) and one HIGH from any of unused. Moreover, only procedural approaches with freely available implementation integrable to ProM are considered for testing.

Paper is included into SLR results, if it has 10 or more citations (for papers from 2016 it is two) or scores two HIGH from triple of (log type, number of logs used, log type), one HIGH from tuple (type of experiments, type of validation) and one HIGH from any of unused.

2.6 Study Selection

The SLR in this paper is carried out by applying query words specified in the Section 2.2 over the seven libraries from Section 2.3. With first search in Scopus using string "process discovery" and no timespan restriction, we discovered that the last SLR on process discovery was done at 2011 by De Weerdt et al. [WBVB12] With this, first query restriction was added, article must be published 2011 or later.

Table 1 describes the findings with our queries. Following describes the exception situations and their solutions. For string "process discovery" in Google Scholar only papers published within last year were considered (with option Search:Abstracts), it was done so to find only the newest papers. For string "process learning" SpringerLink, ScienceDirect and Google Scholar gave more than 1000 results, thus the query was refined. For ScienceDirect it was modified to ("process learning" AND "process mining"). Also string "business process learning" was used. For SpringeLink it was modified to ("process learning" AND "process mining"). Also string "business process learning" was used. For Google Scholar, string ("process learning" AND "process mining") was used and also search term "business process learning" was used for search and results were sorted by date.

Our queries ended up with 2165 results to apply inclusion criterion on. To prune this list, an iterative approach was taken. With first run, based on the title, abstract, keywords, conclusion and a brief look on the content, the article was either marked down as candidate paper for next iteration, or discarded. If the first look didn't give enough information for the decision, article was also marked down. Also duplicate papers were dropped. Thus, the first iteration resulted in

Search string/ Data source	process discovery	workflow discovery	process learning	workflow learning
Scopus	331	10	222	21
Web of Science	175	4	97	7
IEEE Xplore	68	4	35	2
ACM	21	1	47	1
SpringerLink	335	29	47	17
ScienceDirect	110	13	29	12
Google Scholar	79	182	126	140
Total	**1,119**	**243**	**603**	**200**

Table 1: Number of studies matching the inclusion criteria (as of October 2016).

Table 2: Summary of papers left for each string

stage	**process discovery**	**workflow discovery**	**process learning**	**workflow learning**	**Sum**
Second	196	46	67	22	331
Third	101	13	4	7	125
Fourth	54	3	1	2	60

331 papers.

In the second iteration, papers are marked as yes or no. It is done by taking an in depth look at the articles, especially at the implementation details. If the approach is not implemented, connection with business process discovery domain is weak or there is no novelty, the article is marked as no. If the approach is implemented, but there is not enough information to find out, if the implementation is available, or it is a commercial tool with no trial period, the article is still marked with yes. All the articles that are marked as yes will be moved to third iteration, where the quality assessment of the articles will be done. So all together 125 papers.

In the third iteration the quality domains as mentioned in Section 2.5 were applied. This ended up in 60 papers. Figure 1 illustrates the study selection process and Table 2 summaries the number of studies remaining after each iteration for each query word. Papers included in this study are up to second week of October 2016.

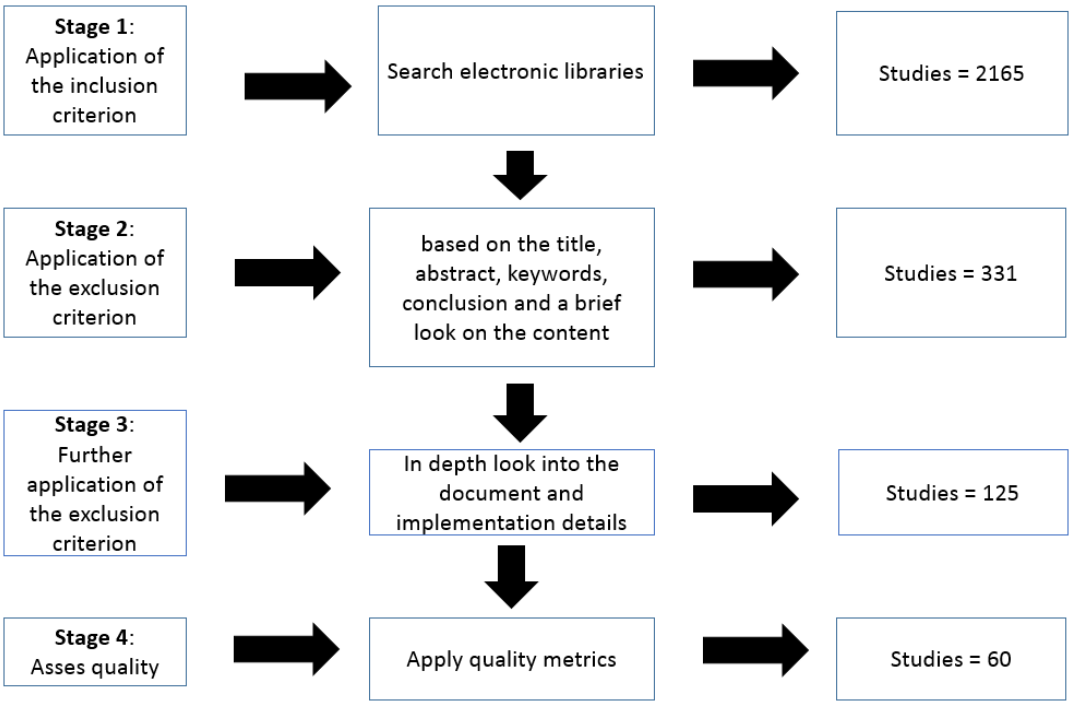

Figure 1: Selection process

2.7 Data Extraction Strategy

For all the papers identified during iteration 2 from Section 2.6 the relevant information in order to answer the research questions defined in Section 2.1 was extracted. The following information was taken into consideration:

1. General information - Paper name, Authors, Year published, Citations
2. Family of algorithm
3. Tool, where approach is implemented
4. Model output from the approach
5. Used database to find the paper
6. Implementation details - Language implemented in, name of the approach and availability for free
7. Validation details - Number of logs used, log type, type of validation, type of experiments, number of experiments, comparison to the state-of-the-art,

log size, event types, number of activities, longest traces, smallest traces, average trace.

2.8 Data Analysis

Further data analysis was carried out to answer the research questions defined in Section 2.1. In detail, the data is synthesized by answering the information extraction categories specified in Section 2.7. The **RQ1** will be answered with categories 1, 2, 3, 4. **RQ2** will be answered by using category 4. Categories 2, 4, 6 answer the **RQ3**. For answering **RQ4**, category 3 gives the answer. **RQ5** will be answered with category 7.

Afterwards, we categorized the approaches using the following taxonomy:

1. Frequency based heuristics
2. Genetic based heuristics
3. Theory of regions based
4. Probabilistic
5. Others

The following chapter summaries the approaches.

3 Results of the Literature Review

The results obtained from the SLR are presented in this Section. Figure 2 shows how the primary studies are distributed over the years. It can be seen that from year 2013 on a growth in number of papers published in the field of process discovery has happened. This indicates that the field has gained the interest of academics and is becoming more and more important. Sudden increase in number of research papers also indicates the maturity of the field. Low number of studies included form years 2011 and 2012 can be explained by fact that most of the approaches proposed by then have already been updated or improved by their corresponding authors or researchers in the field. The number of papers selected from year 2016 is smaller than from 2015 due to the fact, that this study queried data sources at October 2016, as specified in Section 2.6. This distribution also shows that process discovery is a "hot" theme in the field of process mining.

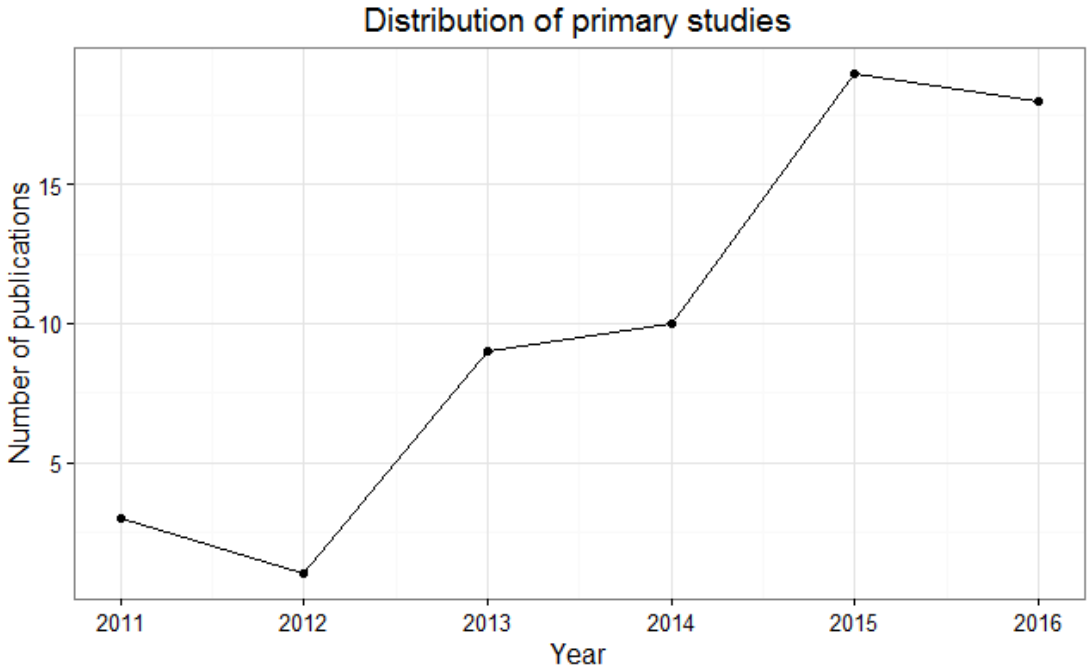

Figure 2: Distribution of primary studies by publication year

These 60 papers were further classified into five classes by asking for each paper: "What are the underlying methods used for implementation?" By answering this question for each primary paper, it was noticed that an approach can be frequency based, genetic based, theory of regions based, probabilistic based or others (i.e. didn't clearly fit into the four before). In the Figure 3 the distribution of papers

into the taxonomy is given. From this figure can be seen that most of the papers fall into the frequency based heuristics category. Table 3 includes all the primary research papers.

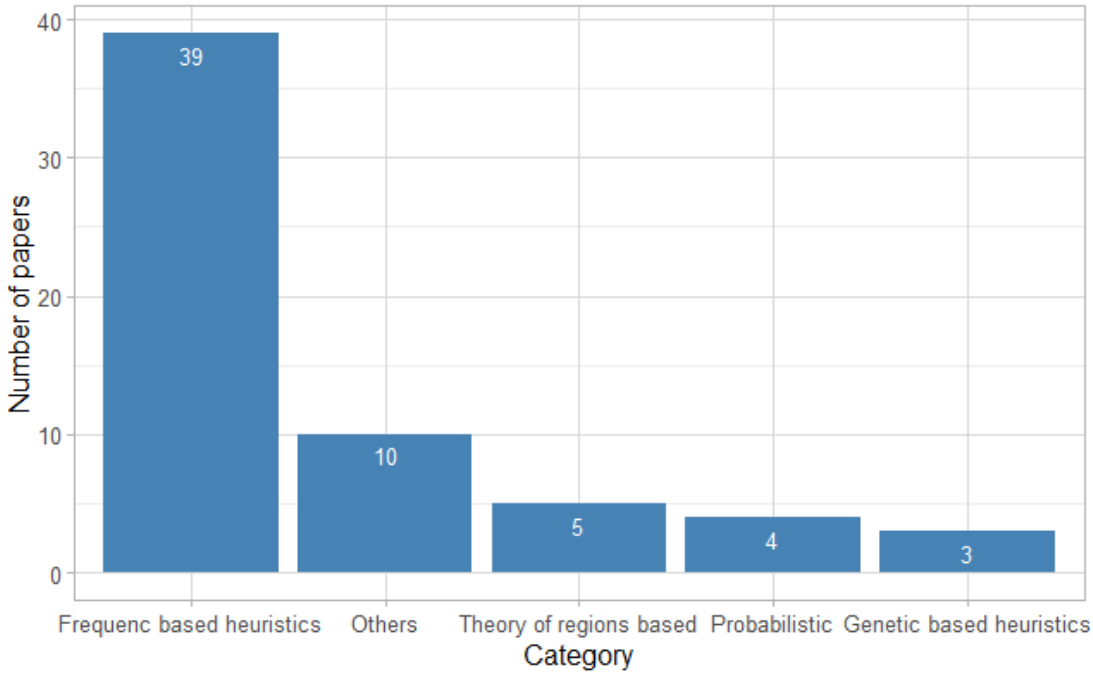

Figure 3: Distribution of primary studies to the taxonomy classes

In the next five subsections (3.1, 3.2, 3.3, 3.4 and 3.5) the main approaches of process discovery identified by this study are described and discussed under the groups listed before. For better readability they are further grouped by the process model produced.

3.1 Frequency based heuristics

In this section, the approaches that use frequency based heuristics to derive a model, are described. These approaches use the number of occurrences of an activity, event or trace in the log to find dependencies. If the number of dependencies is under a certain threshold, it is ignored.

Study identifier			
P1 Bernardi et al. [BCFM16]	P16 Augusto et al[ACD+16]	P31 Leemans and van der Aalst [LvdA14]	P46 Arbab-Zavar et al. [ACN14]
P2 Leemans et al. [LFvdA16]	P17 Breuker et al. [BMDB16]	P32 Popova et al. [PFD15]	P47 Ferilli [Fer14]
P3 Evermann [Eve16]	P18 Ladiges et al. [LHFL15]	P33 Vazquez-Barreiros et al. [VML15]	P48 van der Aalst [VdA13b]
P4 Di Ciccio [DMM16]	P19 Molka et al. [MRD+15]	P34 Leemans et al. [LFvdA14b]	P49 Maggi et al. [MDGM13]
P5 Conforti et al. [CDGR16]	P20 Van Zelst et al. [vZvDvdA15]	P35 Song et al. [SJYM16]	P50 Maggi et al. [MBvdA13]
P6 Leemans et al. [LFvdA15b]	P21 Leemans et al. [LFvdA15a]	P36 Bleser et al. [BDB+15]	P51 Leemans et al. [LFvdA13]
P7 Yahya et al. [YSB+16]	P22 van der Aalst et al. [vdAKRV15]	P37 Redlich et al. [?]	P52 de Leoni and van der Aalst [dLvdA13]
P8 de Leoni et al. [dLStHvdA16]	P23 Ponce-de-LeÃşn et al. [PCvB15]	P38 De Cnudde et al. [CCP14]	P53 Verbeek et al. [VvdA12]
P9 van Eck et al. [vESvdA16]	P24 Pizarro and SepÃžlveda [PS14]	P39 Buijs et al. [BvDvdA14]	P54 Di Ciccio and Mecella [DM13b]
P10 Maggi et al. [MMDM16]	P25 Di Ciccio et al. [DMMM15]	P40 Leemans et al. [LFvdA14a]	P55 Di Ciccio and Mecella [DM13a]
P11 Mokhov et al. [MCB16]	P26 Verbeek and van der Aalst [VvdA14]	P41 Leemans et al.[LFvdA13]	P56 van der Aalst [vdA13a]
P12 Srinivasan et al. [SBVA15]	P27 Liesaputra et al. [LYC15]	P42 Abe and Kudo [AK14]	P57 Huang and Kumar [HK12]
P13 SchÃűnig et al. [SRC+16]	P28 Greco et al. [GGLP15]	P43 Conforti et al. [CDGR14]	P58 Ribeiro and Weijters [RW11]
P14 Kala et al. [KMDF16]	P29 Guo et al. [GWW+15]	P44 Vasilecas et al. [VSL14]	P59 Motahari-Nezhad et al. [NSCB11]
P15 Tax et al. [TSHvdA16]	P30 Folinio et al. [FGP15]	P45 Maggi et al. [MSR14]	P60 Li et al. [LBvdA10]

Table 3: Primary research papers

3.1.1 Fuzzy maps

A two phase pattern-based approach was implemented by Li et al. [LBvdA10] as ProM plug-in. It uses Fuzzy maps as modelling language for process models and is therefore a procedural approach. Their approach can handle sequence, exclusive choice, parallel and loop constraints, it can't handle OR-choice. They named their approach as Fuzzy Map Miner. In their two phase approach, the log is at first preprocessed by using analyst domain knowledge or using the combination of analyst knowledge and the common execution patterns found, and then, in a second stage, mined by using modified Fuzzy Miner[GvdA07], that can deal with abstract activities.

3.1.2 State machines

Next two approaches use state machines as process modelling language. The usage of state machines means, that in general, parallel activities can't be shown explictly. They can't also describe OR-constructs, but they can catch sequence, XOR and loops. The first approach in this group is implemented by Motahari et al. [NSCB11]. It is called Process Spaceship and is implemented as Eclipse plug-in. Finite state machines are used here to describe a mined log, thus it is a procedural technique. This approach uses event correlation for mining a log. This tool can be used in automated or in semi-automated mode. In the automated mode, the tool discovers a model by on a click. In the semi-automated mode, the user supervises the discovery and can select candidate attributes and conditions. The limitations for this approach is not catching concurrency, not tackling imperfect logs and not handling long traces.

A second approach that uses state machines as a way to present a process model, was implemented by Ladiges et al.[LHFL15] In this technique a subset of Petri nets

is used, Machine State Petri nets. Due the usage of Petri nets, it is procedural approach and also catches parallel activities. This approach bases on causality and also takes into account timing and behaviour of a signal. This reduces complexity when compared just considering causality. This approach is designed to be used in the context of industrial machinery (e.g. conveyor belts). It tackles with anomaly detection.

3.1.3 Causal nets

Ribeiro and Weijters[RW11] use the idea of event cubes to produce a causal net. It has been implemented as ProM plug-in. So multidimensional process models are created. In this approach the event log is at first indexed, based on those indexes, a event cube is built and mining is done on the top of the cube. Mining is done by relying on the Flexible Heuristics Miner multidimensional version. It is an approach for doing multidimensional analysis. Due the nature of the causal nets, it is a procedural approach.

Another procedural approach for mining causal nets was created by Greco et al. [GGLP15] as ProM plug-in. It is called CNMining. The idea behind it lies in precedence constraints (information from the log in combination with prior knowledge as constraints over the topology of model) based graph manipulation. At first the dependence graph is created without forbidden edges. Then, not forbidden edges are added to the graph incrementally until underlying positive path and edge constraints are satisfied. The selection of edges is done by using causal scores. Then binding are added to ensure that all the traces are covered and a causal net is returned. If no path constraints exists, the dependency graphs is created without removing the negated edge constraints, only edges with low casual score are ignored. To find forbidden behaviour, the causal predecessors and causal successors are used. Finally causal net is returned as in case where the path constraints exist. The input for this approach is taken as MXML or XES event log.

3.1.4 Declare

Di Coccio and Mecella have created MINERful++ algorithm as standalone and ProM plug-in [DM13b]. This algorithm was later used in their work about MailOfMine [DM13a] as underlying engine for producing declarative process models. The used modelling language was Declare. Both approaches can deal with sequence, exclusive choice, parallel and loop constructs. MailOfMine is a tool for creating process models out of e-mails. It can work IMAP folder and also with .eml files. After the fetchers have done their job, an even log is created and mined with MINERful++. Their implementation is not catching data-flow. MINERful++ itself bases

on creating at first a knowledge base of execution traces temporal statistics and then computing the statistical supports. So all the constraints in the model are checked for the validity.

Another approach for discovering Declare models was implemented by Maggi et al. [MBvdA13]. It is called Declare Miner and uses domain-knowledge as references maps (model repair) or as activity clusters for discovering. It can use as input both event log or event log and a process model. In addition to the use of the domain-knowledge, pruning techniques are also used to reduce the number of discovered constraints by removing redundant ones. Weaker constraints are removed if they are implied by stronger ones, removing transitive closures or by using reduction rules. Declare Miner has been further improved by Maggi et al. [MDGM13] for discovering data-aware models, an extension to the Declare, called data-aware Declare. The addition bases on constraint activation. The idea itself is straightforward, only the constraints that pass the minimum fulfilment ratio metric, are considered. Another addition to the Declare Miner was done by Kala et al. [KMDF16]. It added the combination of sequence analysis (analysis of events position in a trace) and of the usage of the Apriori algorithm to the Declare Miner. The target is yet again on discovering only the frequent constraints, i.e constraints that pass given threshold. This addition is a two step process. At first sets of frequent activities are generated from the log by using the Apriori algorithm and in the second stage at first candidate constraints are generated by using previously discovered frequent activities and are then later pruned by only keeping those that pass the compliance threshold. Their addition is implemented as a separate JAR-file and could be further integrated into ProM. Declare Miner itself is available as ProM plug-in and it can discover sequence, exclusive choice, parallel and loop constructs.

Yet another approach for discovering Declare models is an addition to the MINERful by Di Ciccio et al. [DMMM15] It uses set of predefined constraints and an automata product-monoid to produce a consistent model. Through the automata product-monoid the conflicts in a model a resolved and model with guaranteed consistency is produced. Also the redundancies are removed from the model. This way minimal set of consistent constraints are produced. MINERful can discover sequence, parallel, exclusive choice and loop constructs and is implemented as ProM plug-in and also a standalone JAR.

MINERful-vacuityCheck, implemented by Maggi et al. [MMDM16] is also an approach for discovering the Declare constraints. This tool is available as a standalone JAR and needs the core of the MINERful in order to be used. It uses an

extended automata for semantically characterizing activation and relevance of the temporal constraints. Thus only relevant constraints are extracted.

Di Ciccio et al. [DMM16] created an approach to discover Target-Branched Declare models and named it TB-MINERful. It has prototypical standalone implementation. Due the usage of extension of the Declare language, this tool can show the exclusive choice explicitly in the process model, i.e the target of a constraint can be a set of activities. TB-MINERful itself is a three stage approach. At first a knowledge base is built from the event log. In the second stage the support and confidence of the constraints are queried from the knowledge base and in the third stage, the constraints are pruned by using the branching factor, minimum support and minimum confidence metrics. Thus again, only relevant and interesting constraints are extracted.

Also the Non-Atomic Declare Miner can be used for producing a Declare process model. It is implemented by Bernardi et al. [BCFM16] as a ProM plug-in and bases on the WEKA framework. It uses discriminative rule mining to produce a model. As the name suggest, this approach can deal with non-atomic activities and non -instantaneous activities, i.e. activities that have different events in their life-cycles. In order to mine a model, this approach needs a log and a life-cycle transaction model as input. After the input is received, a five step process is initiated. At first life-cycle consistency is checked, in the second stage the boundary states are found, in the third stage the inter-life-cycle relation are discovered, in the fourth stage, the intra-life-cycle relation are discovered and in the final stage, intra-life-cycle relation selection is done. Through these stages, the constraints are verified and only the most meaningful ones are selected for the model.

The SQLMiner implemented by Schöning et al. [SRC+16] can be used for discovering constraints of the various declarative process modelling languages form the relational event log databases. This standalone implementation uses the SQL syntax to specify a query. Overfitting issues are overcome by using higher confidence thresholds. Due the high customization of the queries, not only control-flow perspectives can be discovered, but also others, like organizational. Although, this approach is lacking the discovery of data-related constraints. So it is more a supervised approach.

3.1.5 Process trees

There exists whole family of approaches, the Inductive Miner family, to derive a process tree from a log by using frequency based heuristics. They all have been implemented as ProM plug-ins and are divide-and-congure based. It started with

classical Inductive Miner implemented by Leemans et al. [LFvdA13]. It grantees to find sound, i.e there are no deadlocks, and fitting models. It works by computing log splits, namely a split is computed directly based on the ordering of activities in a log. A directly-follows graph is split into n set of the nodes by the operators that indicate the orders of the behaviour. There exists cuts for sequence, exclusive choice, parallel and loop. In order to produce language-rediscoverable models, the underlying log must be directly-follows complete to the underlying model and model underlying the log must be representable as process tree without duplicate activities and silent activities, and all the loops are without same start and end activities (e.g. no self loops). In order to produce a model with guaranteed soundness and fitness to the log, the log is not required to be directly-follows complete.

To tackle the infrequent behaviour in the log, the Inductive Miner - infrequent was created by Leemans et al. [LFvdA13]. It can also handle large event logs. As in regular Inductive Miner, the soundness (at least 80%) of the model is once again guaranteed. The idea of tackling the infrequent behaviour is by using of the frequency-based (traces and events) filters. These filters will be applied only when the Inductive Miner returns a flower model. If needed, the filters will applied on the operator and cut selection (heuristic-style filtering and eventually-follows graph), on base cases and on log splitting. Also, this filtering is done locally. This approach has some issues when dealing with filtering parallel behaviour, especially when the log is incomplete.

To deal with incomplete logs, the Inductive Miner - incompletness was created by Leemans et al. [LFvdA14a]. The activity partition problem in the Inductive Miner is now replaced by the optimisation problem. This problem is solved by three steps. At first the graph of the log and its transitive closure are computed. In the second step, the cuts that have highest probability, are chosen by relying on the SMT-solver [dMB08]. In the final step, sublogs are created based on the cuts and on each sublog the base case is recursively found. Process tree is created by the hierarchy of recorded operators. Noise is not handled by this approach.

Inductive visual Miner is process exploration tool and was created by Leemans et al. [LFvdA14b] as ProM plug-in to provide better visualization. It is directly-follows based and can also handle parallelism. It could be used as process discovery tool, because it is basically a variation of the Inductive Miner - infrequent. Process trees are used internally here and final model is a Petri net with BPMN parallel gateways. This could be transformed to Petri net or BPMN model. For improved visualization three filters can be used, for frequent paths, for frequent activities and for specific activities. The focus of this approach is on visualising deviations

of the fitness. Once again the soundness and fitness are guaranteed.

To deal with big event logs, the Inductive Miner - directly-follows based (IMD) framework was created by Leemans et al. [LFvdA15a, LFvdA16] as ProM plug-in. It is a directly-follows based approach as name suggest. Inductive Miner is modified to work on directly-follows graphs and rather not on the logs. This IMD has three variations, base, one to deal with incompleteness and one to deal with infrequency. The IMD frameworks takes the directly follows graph as an input and produces a model. The base, infrequent and incomplete approach follow similar procedure as in Inductive Miner, Inductive Miner - infrequent and in Inductive Miner - incomplete respectively. Due the low usage of the RAM (for handling five billion events, 2GB of RAM was used) showed that it can handle the big logs and could be even acceptable in the streaming environments. Once again, as in IM, the soundness and rediscoverability are guaranteed. There are also limitations of IMD. At first, the process must be block-structured and directly-follows complete in order to guarantee rediscoverability. Some incompletenesses may not be handled correctly. In infrequent behaviour some cuts may be incorrect as in Inductive Miner - infrequent. Also the balance between fitness, precision, generalisation and simplicity might be off in some cases.

Another derivation of the Inductive Miner is Inductive Miner - life cycle (IMLC), which was implemented by Leemans et al. [LFvdA15b] as ProM plug-in. This approach was created to correctly distinguish concurrency and interleaving. At first, to work with logs, they need to be pre-processed, meaning that all the events that contain other life cycle annotations than start or end, are ignored. It is also assumed, that logs contain some non-atomic activities. In order to leverage the atomicity of process trees, the collapsed process trees (link between start and complete is kept) are used. To deal with life cycle data, the IM cut detection is changed to deal with start and end activities and also with collapsed activities. Fall troughs is changed to construct a model by counting the number of times when activity is concurrent with itself. For detecting interleaving three stages are used. At first candidate footprints are found from the directly-follows graph. In second step log split by the candidate footprints and in third step the real footprints are found recursively and false positives are replaced with exclusive choice. Trough this, sound and consistent models are produced. The main problems of using collapsed process trees is representational bias (not being able to show restrictions on start and competition), non-convertible to most formalism due the usage of unbounded concurrency and not being able to achieve traditional perfect fitness in some cases.

3.1.6 Hybrid models

An approach, that stands between declarative and procedural process models domains, was created by Maggi et al. [MSR14]. It is named HybridMiner and have been implemented as ProM plug-in. A Declare model and a Petri net is combined together. This results in a hierarchical process model, where the top level model depends on the structuredness of the log. To create a model, at first a log is split into two sets, one containing structured events and one containing unstructured events. After this, the log is again analysed for structured and unstructured sequences and is split into two sub-logs accordingly. As third step, the procedural sub-processes are mined (one being in the structured sub-log). As fourth step, the declarative sub-processes are mined. If the similarity metric among the traces is lower or equal to 50%, the top level process is found with declarative miner. If not, then with procedural miner. As final step, the resulting models are combined together. To mine unstructured processes, Declare Miner is used and for structured processes, the Heuristic Miner is used.

3.1.7 Directed Acyclic Graphs

To provide support and a based for creating Bayesian belief networks, a procudural approach was created by Vasilecas et al. [VSL14]. In their work, a Directed Acyclic Graph (DAG) is extracted from the log. As the name suggest, a graph without loops is created. To create a graph, the log is iterated through and if a loop is detected, then when it is a incorrect loop (model level, doesn't actually exist in log), associated nodes are made independent of each other. When it is really a log level loop, then dummy nodes are added. So a DAG is created that shows the control-flow of process without loops.

3.1.8 BPMN

BPMN Miner is tool for creating hierarchical BPMN models. It was first created by Conforti et al. [CDGR14] in 2014 and later improved by Conforti et al. [CDGR16] in 2015 to detect and filter noise. The BPMN Miner is available as ProM or Apromore plug-in and also as standalone. The noise handling is done by using approximate dependency discovery techniques and a set of filters. To produce a model, the set of events of a event type is seen as relational table, likelihood of events that share same primary key will be long to the same process and that process-subprocess relations are indicated by foreign keys between event types are also used. At first a flat-model is mined with Heuristic Miner, ILP, InductiveMiner, Fodina Heuristic Miner or α-algorithm by user choice. Then this flat-model is re factored by using a set of heuristics to be hierarchical. BPMN Miner works best with logs that contain records of a single business process.

Another approach to discover BPMN models, Structured Miner, implemented by Augusto et al. [ACD+16] as ProM plug-in and as standalone tool, is quite similar to BPMN Miner. It follows "discover and structure" approach. So at first discovery and cleaning are done and then structuring is performed. More specifically, XES or MXML log is taken as input, mined with Heuristic Miner of Fodina Miner for a baseline model applied with heuristics ensuring that model contains single start and end event, enusring that split and join gateways are from the same type for every bond and replacing the quasi-bonds injection/ejection with XOR-gateways, and then structured as maximally as possible by using a technique for maximally block-structuring an acyclic model and a technique for block-structuring flowcharts. It is noteworthy that soundness of models is always guaranteed.

Another approach that could be used for mining BPMN models, is Dynamic Constructs Competition Miner(DCCM) by Redlich et al. [RMG+14]. It is procedural and bases on the "divide and conquer" principle. It works with streams of events and can discover sequence, parallel, exclusive choice and loops constructs from the streams. Since ageging techniques are used, the approach can capture and deal with dynamic changes in a process without heavy penalties on scalability. To enable scalable dynamic process discovery, the original Constructs Competition Miner, the base for DCCM, was modified. First, the base algorithm was split into two parts, one dealing with current footprints and one for creating a model from footprint. Secondly, instead of calculating the footprint in relation of all occurring traces, the footprint is calculated event-wise. Subset footprint calculation was changed to automatic derivation from parent set footprint. Also a measure was added to favour the last control-flow state. This approach can deal with noise and not-supported behaviour. It can also detect changes in a steam almost instantly.

3.1.9 Activity diagrams

Process models can also be discovered by using monitoring frameworks. One of these is a procedural process skeletonization based approach by Abe and Kudo [AK14], which produces workflow models. It can discover sequence, exclusive choice, parallel and loop constructs. Metrics and process instances are extracted from the log with single pass by using monitoring context and linking them at runtime. For this purpose correlation key definitions inclusion relationships are used and the lifecycle of monitoring context instance parents and children are handled independently. If the requirements of the analysis are changed, the log is not reanalysed. Only the values on analysis axis must be changed.

3.1.10 Heuristics nets

In order to provide higher validity and completeness for the Heuristic Miner, it was updated by De Cnudde et al. [CCP14] and underlying artifact was named Updated Heuristics Miner. First, new dependency measures are defined for loops of length of one and loops of length of two for detecting noise in them. For the loop of one, the strength of the loop is checked against the best connection the activity in the loop has. It is represented as an interval [0,1]. For loop of two, the strength of a loop dependent on the best connection an activity in the loop has, is checked. Extra dependency relations are modified to based on relative to best threshold and on the positive observation threshold. All-tasks-connected heuristic is extend to include loops. Correct interpretation of the AND- and XOR-relations is added when transforming from dependency graphs to causal nets and also the nature of the relations is captured correctly and right gateway is added. The condition, in which loop of two relation was not accepted, when two activities x or y were present in an loop of one and occur more than once in the loop of two, is dropped. Trough this fitness value near to 1.0 is achieved, noisy short loops removed and the overfitting of the data is avoided. It is claimed, that it could be used as ProM plug-in by simply marking a checkbox "Use updated Heuristics Miner" in the user interface of Heuristics Miner.

Heuristics nets can be also mined with approach by Liesaputra et al. [LYC15]. The name of the approach is Maximal Pattern Mining (MPM) and it is implemented as ProM plug-in. The idea is to mine a optimal set of patterns, that can cover all the traces in the whole log by using only event types. Other information is ignored. The soundness of returned models is guaranteed. Noise is dealt with by using user specified threshold for frequency of trace or event. Other complex constructs are also handled, although duplicate events are sometimes shown in the model. It is a approach for finding control-flow models.

Proximity miner created by Yahya et al. [YSB+16] provides another way to discover heuristics nets from the logs. It is an ILP-based approach and implemented as ProM plug-in. For discovery, the event relation in the log and users knowledge specified as constraints, are leveraged. Additionally, if the user is not familiar with domain, she can specify threshold for behaviour to be included into a log. Only behaviour that is satisfied by constraints, is extracted. Proximity scores are used for finding extra behaviour and to provide soundness. The soundness of a model is guaranteed iff the domain constraints are used. The model is simplified by merging event types according to activity. The approach has some issues, when discovering complex loops (illogical behaviour is added to the model). Another limitations are "subjective models" and the time it takes to complete the discovery.

3.1.11 Partial Order Graphs

Episode Miner is a plug-in for ProM by Leemans and van der Aalst[LvdA14]. Due it's nature, it can express sequence, parallel and loop constructs. The discovery is oriented on finding frequently occurring episodes(partially ordered events) from the log. Episodes candidates are generated by using Apriori algorithm and if the frequency value is equal or greater than threshold value, it is frequent. To be more efficient, pruning is at first to skip infrequent activities and activities with low temporal locality. Trough this noise is handled. Sound model is not guaranteed and the models are also not end-to-end.

3.1.12 Conceptual clustering models

A two-phase clustering-based (logical decision rules to dicover clusters and then extraction of behavioral fitness by merging as many clusters as possible without reducing the fitness) approach was created by Foliono et al. [FGP15] as standalone plug-in and named as MVPM-mine. The discovered models are multi-variant process models. The input for this approach is an event log. The approach is meant for dealing with lowly-structured processes. It can discover sequence, parallel, loop and exclusive choice constructs. A function from Flexible Heuristics Miner is used to deal with noise for workflow schemas.

3.1.13 Petri nets

A plug-in for discovering Petri nets with non-free choice constructs and invisible tasks was created by Guo et al. [GWW+15] as ProM plug-in and named as $\alpha\$$. The input for this approach is an event log. At first invisible tasks are detected by using improved mendacious dependencies, secondly, the reachable dependencies are supplemented in the context of discovered invisible task. Thirdly, by using implicit dependencies, the non-free choice constructs are discovered. Next, the invisible tasks are adjusted (by combining or splitting) to ensure the soundness of the model. Finally, a model is constructed. This approach can discover invisible tasks inside non-free choice constructs.

3.1.14 Conditional Partial Order Graphs

Mokhov et al. [MCB16] created an approach Workcraft plug-in for extracting Conditional Partial Order Graph (CPOG). It is also implemented as command line standalone tool named PGMINER. Sequence, parallel and exclusive choice constructs can be extracted from the logs. Control- and data-flows can be extracted. CPOGs can be mined by treating each trace as a totally ordered sequence of events or by exploiting the concurrency between the events. With both, all the traces

are covered from the log. Event attributes are used for adding data labels to the conditions. The quadratic explosion of the representation is avoided by using the algebra of Parametrised Graphs. The input is an event log. If log is imported directly, each trace is treated as a total order of events. If imported indirectly via PGMINER, the log undergoes the concurrency extraction. Second option allows handling of bigger logs.

3.1.15 Multiple outputs

Multiple process models, namely social networks and heuristics nets, can be discovered by an approach created by Pizarro and Sepúlveda [PS14]. It is implemented as ProM plug-in and named as OLAPDiscovery. It provides multi-perspective interactive process exploration on control-flow, organizational or time dimensions. Heuristics nets for covering control-flow are visualized by using Heuristics Miner and social networks for organizational view by using Social miner. Thus noise is handled. For discovery, OLAP Cube with three dimensions (variant, resource and time), is used. The intial model, when the discovery process is done, is heuristics model. Then the user can specify by clicking on tabs whether she wants to see the organizational model, comparison view, to reset the analysis, to save the analysis or to select the dimension (organizational, variants, time, version). So it is a tool for providing user a better productivity compared to the usage of chain-of-plug-ins.

Evermann[Eve16] implemented α-miner and Flexible Heuristics miner on the top on cloud-computing platform (Amazon Elastic Map-Reduce and Amazon S3 and Amazon EC2 clusters) with Map-Reduce framework. The idea of this work was to show, how the current process discovery algorithms could benefit from the Map-Reduce framework to deal with distributed event logs. The log-based ordering relations are redefined as map()-shuffle() and reduce() jobs. For FHM, the dependency measures are also redefined. They are emitted as occurrence counts from reducers, since combiners cannot work with commutative behaviour. Dependency graphs is defined as reducer job. The splits and joins are discovered again in a reducer job with the aid of dependency graph. The discovered models are still Petri nets and heuristics nets. The evaluation of the approach showed drastic (instead of full day, the data was process with 8 minutes for α and 17 minutes for FHM, the baseline was run of Map-Reduce on single cloud node) improvement in performance time.

3.2 Genetic based heuristics

In this section, the approaches that benefit from the evolutionary computing, are described. The basic idea is to find best fitting model from candidate set. The

process is ran, until a model is found or the maximum number of iterations reached.

3.2.1 BPMN

Evolutionary algorithms can be used to mine hiearchical business domain-specific models, e.g. BPMN models. The example of this is Diversity Guided Evolutionary Miner (DGEM) by Molka et al. [MRD+15]. This approach can deal with sequence, exclusive choice, parallel and loop constructs. In order to reduce complexity of models and offer a structuredness, the complex constructs are resolved in sequential shape. Early coverage to the local optima is avoided due the usage of diversity-adapted fitness function. Through this, also the noise is handled. Once a model, that best describes the underlying log, is found, the process stops.

3.2.2 Process trees

Another representative of genetic process discovery is the Evolutionary Tree Miner by Buijs et al. [BvDvdA14]. It is implemented as ProM plug-in, called ETM. It is a procedural approach that can discover sequence, loop, parallel, exclusive choice and inclusive choice, which not common in the field. Precision, generalization and simplicity are considered, when acceptable replay fitness is achieved. However, the ETM can be optimized towards any of the aforementioned four. Also, the sound process model is guaranteed. Due the usage of process trees as internal repesentation, the search space is reduced compared to Genetic Miner [vdAdMW05]. At first a population of random process trees is created and the overall fitness is calculated. Then the trees are changed (replacing entire tree, switch sub-trees between two trees, adding a node, removing a node and changing a node), if no stop criteria is matched. It is done, until a certain stop criteria is achieved. Then the fittest model is returned. The process trees can be later converted into a BPMN model. If no candidate is found in 1000 generations, the process is stopped.

3.2.3 Heuristic nets

Genetic approaches can be used to derive a Heuristics net from the log. ProDiGen, a standalone miner by Vazguez et al. [VML15] represents this idea. It is basically a modified Genetic Miner[vdAdMW05] and meant for control-flow discovery. New operators for crossover (selection based on the errors of the mined model) and mutation (relies on the log's causal dependencies) are specified, and also a new hierarchical fitness function that considers also completeness, precision (log and model based) and simplicity (model based), is used. Discovery is done in three-stages, at first log is pre-processed for removing the noise. Then, a model is mined, by using genetic approach and finally, the model is post-processed by removing infrequent and redundant arcs. Initial population is created by using the

results of Heuristics Miner [WvDADM06]. Population individuals are evaluated with the fitness function and if the best match is found or the maximum number of reinitializations is reached, the process is stopped. The approach is available as a web-based front-end.

3.3 Theory of regions based

In this section, region-based approaches are described. Theory of regions can be seen as state-based or language-based extraction of model. In state based approach, at first a transition system is created and then a model is extracted from that. In language-based, the activities in a trace are seen as the letters, the traces are seen as words, and the log itself as a language. The extracted model is then set of words allowed in a language [VML15].

3.3.1 Petri nets

Verbeek and van der Aalst [VvdA12] created passages-based ILP Miner for mining Petri nets. It has been implemented as ProM plug-in and should be freely available. This procedural approach can deal with sequence, exclusive choice, parallel and loop constraints in the log. As the name suggest, log is at first split into pairs of sets of event classes logs. The splitting is done by using directly-follows relations found with the Heuristics Miner. These logs are then mined with ILP Miner for Petri nets and in the third step, the mined Petri nets are combined together. This reduces runtime when compared to the standard ILP miner. This approach does not perform well with the logs with many event classes (already on the 720 classes the execution time was around 183 minutes).

The ILP Miner is also modified for using decomposition. This approach is called *Discover with ILP using Decomposition* and is implemented in ProM's DivideAndConquer package by Verbeek and van der Aalst [VvdA14]. The important difference of the regular ILP is not using the causal dependencies to search for places. Clusters of activities are created by discovering a causal activity matrix, then extracting a causal dependency graph by only considering the "true" causal dependencies and then clusters are extracted from the graph. This leads to the decomposition of event log. Then discovery is done on each sublog and the resulting models are merged together to create the final Petri net. Better runtime with penalty of less accurate models is achieved than in regular ILP miner. It is noteworthy, that DivideAndConquer package is general and could be used also with other discovery approaches.

Another ILP based approach is Hybrid ILP Miner implemented by van Zelst et

al [vZvDvdA15]. It is also a ProM plug-in. Hybrid regions are used for discovery. Hybrid means that single variable based region and dual variable based region is united into single hybrid variable-based language region with favouring minimal regions. This allows discovery of non-pure Petri nets with balanced usage of variables, i.e. complex constructs are found. Causal relations within an event log are used for finding multiple places. Using one or two variables for an activity allows gains in the performance time.

Petri nets can be mined also with Supervised Polyhedra, an standalone approach by Ponce de Leon et al [PCvB15]. It bases on numerical abstract domains and Satisfiability Modulo Theories. Also negative information (traces that cannot be executed) can be used for discovery. This information can be derived automatically from the log or be provided by domain experts. The underlying approach [CC14] is extended with an extra simplification step before Petri net transformation from polyhedron by reducing the coefficients of each inequality and adding negative information as points not to be enclosed by the polyhedron. The discovered Petri nets are pure, i.e. they have arbitrary arc weights and tokens. If the approach is used without negative information, it is rather a tool for model simplification and generalization.

3.3.2 BPMN

BPMN models can be mined by using a ProM plug-in in LocalizedLogs package. It is created by van der Aalst et al. [vdAKRV15]. Additional to BPMN model, a Petri net can also be produced. Due its nature, it is a procedural approach. In this approach sequence, exclusive choice, parallel and loops constructs are dealt. Localized means that for each event non-empty sets of regions are assigned. This means that sublogs are used to represent regions. This decomposes discovery and speeds up analysis. Events are considered to be unrelated, if they are not in same sublog. Thus only local completeness is needed. So unrelated models are created from sublogs and by putting all the models together, a final model is created. The discovered BPMN models are hierarchical.

3.4 Probabilistic

In this section probabilistic approaches are described.

3.4.1 Hidden Markov Models

A procedural approach that takes images of workflow as input and based on the input constructs a model was created by Arbab et al. [ACN14].We reference it

as *hierarchical approach*. The task of creating a model is split into activity and task recognition. To create a model, five stages are passed, at first detection and tracking, secondly activity classification, creation of area-based activity HMMs and boosted activity classification as third step, task extraction as preliminary step and finally task label assignment. So, at first Fourier transform is used to analyse images, then a K-Nearest Neighbours classifier classifies based on binary tree structure and as final steps, HMMs are used for more elaborate analysis. This approach performs best, when used in industrial environments with well defined tasks physically performed by humans.

Another approach, that can extract HMMs was created by Bleser et al. [BDB+15] as a procedural standalone. It bases on the European project COGNITO[gorecky2011cognito]. The input is received as image stream describing a process. Data from the on-body sensor network is used for user monitoring, then for workspace characterization and monitoring. In third step, the workflow recovery and monitoring is done and models are extracted. Finally, based on the learnt model, suggestion can be sent for an inexperienced user's head-mounted display. For workflow recovery, at first relational graph structures are created from sensor info, then from theses graphs, bag-of-relations histogram over sliding window is created. Finally the HMM states are associated with the atomic events of the workflow to create a model. Thus it is supervised approach. Due the usage of sensor networks, the approach can also learn and recognize multiple shape-based objects in real-time.

3.4.2 Petri nets

An EM algorithm based approach was created by Breuker et al[BMDB16]. It was named RegPFA and is implemented as standalone approach. It is meant more for predicting future behaviour with comprehensible models. The input is an event log and output a predictive model visualized as Petri nets or probabilistic automata (RegPFA notation model). User-defined threshold can be used to prune the model. Any behaviour the is expressible with state machines having designated start and end states can be expressed with RegPFA, with exception of cancellation patterns. Noise and incompleteness from event log is also handled. Through the usage of Bayesian regularization the overfitting of models is avoided. Hence the name, RegPFA (Regulated Probabilistic Fine Automata).

3.4.3 Logical Guarded Transition Systems

Logical Guarded Transition Systems, a generalisation of Petri nets, can be mined with approach by Srinivasan et al [SBVA15]. Their approach bases on the logical programming and probabilistic programming and comes handy in environments

with transition noise for extracting a biological system transition model. Noise is modelled as probabilistic transition system. The probabilistic automaton is created to distinguish and ignore the noise, and to describe the system model.

3.5 Others

In this section, approaches that didn't fit in any of the categories mentioned above, are described.

3.5.1 Activity diagrams

First of these is the approach created by Huang and Kuma [HK12]. They have named their approach as HK and it is procedural standalone, which is not freely available. This HK is based on the Best-First tree search and produces block-structured process models. It is claimed to handle noise. It can discover sequence, exclusive choice, parallel activities and loops from event log. Based on specification, their approach cannot discover advanced patterns, for example multi-choice. The discovered models allow similar behaviour to the restricted Petri nets. Mismerge scores are used to find the best process model.

3.5.2 Multiple outputs

This subsection describes approaches that don't produce one main model. First of these is ProCube by van der Aalst[vdA13a]. It is implemented in ProM and bases on OLAP data cubes. Therefore, multiple dimensions can be used for process mining. This approach can produce Fuzzy maps, Heuristic nets, social networks and dotted charts. Due the produced models, it is procedural approach. This implementation should be freely available as ProM plug-in.

Another approach to discover multiple process models from log, is Log On Map Replayer, which was created by de Leoni et al. [dLStHvdA16] as ProM plug-in. In their approach, the timeline maps, Cartesian graph-based maps and Petri nets are created. The main idea is on learning process states from the log and then visualizing them. The states are used for visually replaying the behaviour in event log using visual analytics. So it is like a film on how the process was executed. The input for this approach is an event log and a set of maps(can be automatically created, when using Automatic Map Generator plug-in), where activity instances positions are defined. In order to replay a process, it must be first generated from the logs, i.e no one-the-fly discovery. Due the usage of different models, the process can be analysed in multiple perspectives. Thus this tools provides means for analyst for deciding, where to focus it's attention.

3.5.3 Petri nets

A procedural approach to disocver Petri nets with data flow was implemented by de Leoni and van der Aalst [dLvdA13] in 2013. It is available as ProM plug-in and called as Data-flow Discovery. It can deal with sequence, parallel, exclusive choice and loop constructs. The idea behind this approach lies in alignments between event log and control-flow. Due the alignments, this approach can deal with deviating behaviour and complex control-flow constructs. Adding data and guards to the Petri net is viewed as classification task, which can be solved by using decision trees. The C4.5 algorithm is used in this approach for adding the guards. This Data-flow Discovery takes a Petri net without data, an event logs and a set of alignments as input. Therefore it lies somewhere between process discovery and model enhancement domains.

3.5.4 WoMan formalism

An First-Order Logic (FOL) based declarative standalone workflow learning approach was implemented by Ferilli[Fer14]. It is called WoMan. WoMan uses Inductive Logic Programming to derive a model. Due the special predicates used, parallelism is explicitly expressed and hence no statistical consideration are done. To find FOL descriptions from the log the case description under construction, the set of activities that are still running and the set of activities that are terminated structures are created and maintained. From FOL descriptions models are learned incrementally. At each iteration, least general generalization is sought. Pre- and post-conditions are automatically refined during the learning process. When the x XOR x patterns are present, the WoMan formalism model cannot be converted into Petri net. A Petri net could always be converted to WoMan model. Noise is handled trough updating of the weights of aforementioned sets.

3.5.5 Directed graphs

Song et al. [SJYM16] implemented an discovery algorithm to the ProM-D tool. It is a activity dependences (control dependences and data dependences) based, which can discover sequence, parallel, exclusive choice, inclusive choice and loop constructs from the log. The graphs could be further converted into Petri nets. Due the usage of activity dependences, incomplete logs are also dealt without heavy penalties in the model. It is achieved by at first preprocessing the log for removal of noise and irrelevant data with existing filtering tools in ProM. Then the dependence analysis is done and in third step, the model is extracted from dynamic dependence graph. If the interest is on data-driven processes, the dynamic dependence graphs can extracted separately. To get correct models with this approach, a dependence complete event log is a must. Rediscovering the original process is not guaranteed.

3.5.6 Guard-Stage-Milestone models

Popova et al. [PFD15] implemented an approach into ProM's ArtifactModeling package. At first artifact-centric logs are created by finding artifact structures and then creating a sublog for each artifact type. Then a Petri net is discovered from them by using existing discovery approaches (the choice is left free for the user, with exception that discovered net must be sound and free-choice) and finally translated into Guard-Stage-Milestone (GSM) model without compound stages. Major drawbacks are dealing with noise and incompleteness.

3.5.7 Process trees

In their work, Tax et al. [TSHvdA16] combined ideas from frequency based heuristics and genetic based heuristics to mine local process models. The internal representation used are process trees. Their approach is somewhere on the border of the process discovery and episode/sequential mining and has a procedural standalone implementation. With this method frequent behavioural fragments (focus is on frequently occurring patterns) captured as non-start-to-end-models are discovered from input log. Trees are visualized as Petri nets. Model discovery composes of four steps, at first initial set of candidates is created, then evaluation of candidate set, thirdly selection of candidate local process models (if max iterations is reached or selection set is empty, process is stopped) and fourthly, if no stop at third step, expansion of selection set for new candidates. Thresholds per dimension can be set. To enhance the speed of process tree generation, the initial pruning based on monotonicity is performed. Trough pruning, the noise is handled. Trough alignment-based evaluation, the incompleteness is handled. Sound models are guaranteed because of the usage of process trees.

3.5.8 State machines

With van Eck et al. [vESvdA16] approach, Composite State Machines (CSMs) can be discovered. The tool has been implemented as ProM plug-in and is called CSM Miner. CSMs can express sequence, parallel, exclusive choice and loops. The focus is on discovering states. Time for the log is not used for discovery in this approach, it is only used for adding statics at visualization stage. So a transition system is created from log, from which the CSMs are extracted. CSMs are further simplified, by removing redundant arcs, abstracting states and aggregating two given states into one. Support, confidence and lift are used to quantify the behavioural relations and can be used for further simplifying the model. Thus models can be interactively explored. The input is still an event log.

4 Discussion of RQs

In this section, the RQs defined in the Section 2.1 are discussed. Table 4 gives an compact overview of the primary studies in the context of primary studies and previously mentioned taxonomy. Following abbreviations are used in the table:

- H_{freq} - Frequency based heuristics
- H_{gen} - Genetic based heuristics
- T_{reg} - Theory of regions based
- **Prob.** - Probabilistic
- **Oth.** - Other
- **Model lang.** - Model language
- **Imple.** - Implementation
- **Eval.** - Evaluation

4.1 RQ1

To answer **"Which are the existing approaches that deal with process discovery?"**, we must take look on the Table 4. From this table, we can see that there exists 55 distinguishable approaches for the process discovery. For three approaches, S11, S16 and S38, exists more than one primary study. For S11, the relevant studies are [MBvdA13] [MDGM13] and [KMDF16], for S16 [CDGR14] and [CDGR16], and for S38 [LFvdA15a] and [LFvdA16]. This explains why the number of approaches is smaller than the number of primary studies.

The identified approaches were further classified as in Section 3. Out of these 55 approaches, 34 (62%) use frequency based heuristics, three (6%) use genetic based heuristics and five (9%) use theory of regions to produce a process model. There also exists four (7%) approaches, that use theory of probability to produce a model. Remaining nine (16%) approaches use combination of aforementioned underlying techniques. 18 approaches (S4, S12, S13, S15, S17, S21-24, S29, S32, S36, S40-41, S44, S46, S53 and S54) have claimed implementation in their paper, but these are not freely available to download. Remaining 37 approaches have freely available implementation. Approaches S10, S18-20, S38, S44 and S52 produce process trees, that is 13% of approaches. These process tree could be further converted for example into Petri net or BPMN model. Approaches S8-9, S30, S33, S36, S39

	Name	Authors	Year	Taxonomy					Model Type	Model lang.	Process constraints						Imple.	Eval.			Free?
				H_{freq}	H_{gen}	T_{reg}	Prob.	Oth.			AND	XOR	OR	Loop	DeclareCons	WoManCons		Rl	Synth	Arti	
S1	Fuzzy Map Miner	Li et al.[LLvdA10]	2011	✓					Proc	Fuzzy Map	✓	✓		✓			ProM	✓			✓
S2	Process Spaceship	Motahari et al.[NSCB11]	2011	✓					Proc	FSM		✓					Eclipse	✓	✓		✓
S3	Event Cube	Ribeiro and Weijters[RW11]	2011	✓					Proc	C-Nets	✓	✓	✓	✓			ProM			✓	✓
S4	HK	Huang and Kumar[HK12]	2012					✓	Proc	Petri net*	✓	✓		✓			Standalone	✓	✓		
S5	ProCube	van der Aalst[vdA13a]	2013					✓	Proc	Multi	✓	✓		✓			ProM				✓
S6	MaiXOMine	Di Ciccio, Mecella[DM13a]	2013	✓					Dec	Declare					✓		Standalone	✓	✓		✓
S7	MINERful++	Di Ciccio, Mecella[DM13b]	2013	✓					Dec	Declare					✓		ProM, standalone	✓	✓		✓
S8	passages-based ILP Miner	Verbeek et al.[VvdA12]	2013			✓			Proc	Petri net	✓	✓		✓			ProM	✓		✓	✓
S9	Data-flow Discovery	De Leoni, van der Aalst[dLvdA13]	2013					✓	Proc	Petri net	✓	✓		✓			ProM	✓	✓		✓
S10	Inductive Miner	Leemans et al.[LLvdA13]	2013	✓					Proc	Process trees, Petri net	✓	✓		✓			ProM			✓	✓
S11	Declare Miner	Maggi et al.[MBvdA13, MDGM13]; Kala et al. [KMDF16]	2013 2016	✓					Dec	Declare					✓		ProM	✓	✓		✓
S12	WoMan	Ferilli[Fer14]	2014					✓	Dec	WoMan						✓	Standalone	✓		✓	
S13	hierarchical approach	Arbab et al.[ACN14]	2014				✓		Proc	Task map(HMM)	✓	✓		✓			Standalone	✓			
S14	HybridMiner	Maggi et al.[MSR14]	2014	✓					Hybrid	Declare + PN	✓	✓		✓	✓		ProM	✓			✓
S15	DAG	Vasilecas et al.[VSL14]	2014	✓					Proc	DAG	✓	✓					Standalone	✓	✓		
S16	BPMNMiner	Conforti et al.[CDGR14, CDGR16]	2014; 2016	✓					Proc	BPMN	✓	✓		✓			ProM,Apromore	✓	✓		✓
S17	Runtime process skeletonization	Abe, Kudo[AK14]	2014	✓					Proc	Fuzzy map**	✓	✓		✓			Standalone	✓			
S18	Inductive Miner - infrequent	Leemans et al.[LLvdA13]	2014	✓					Proc	Process trees, PN	✓	✓		✓			ProM	✓			✓
S19	Inductive Miner - Incompleteness	Leemans et al.[LFvdA14a]	2014	✓					Proc	Process trees, PN	✓	✓		✓			ProM			✓	✓
S20	ETM	Buijs[BvDvdA14]	2014		✓				Proc	Process trees	✓	✓	✓	✓			ProM	✓		✓	✓
S21	Updated Heuristics Miner	De Cnudde et al.[CCP14]	2014	✓					Proc	Heuristics net	✓	✓		✓			ProM	✓		✓	
S22	Dynamic Constructs Competition Miner	Redlich et al.[RMG+14]	2015	✓					Proc	BPMN	✓	✓		✓			Standalone		✓		
S23	BoR	Bleser et al.[BDB+15]	2015				✓		Proc	HMM	✓	✓		✓			Standalone	✓			
S24	ProM-D	Song et al.[SJYM16]	2015					✓	Proc	Directed graph	✓	✓	✓	✓			Standalone	✓	✓		
S25	Inductive visual Miner	Leemans et al.[LFvdA14b]	2015	✓					Proc	PN+BPMN parallel GW	✓	✓		✓			ProM	✓			✓
S26	ProDiGen	Vazquez et al.[VML15]	2015		✓				Proc	Heuristic net	✓	✓		✓			Standalone		✓		✓
S27	ArtifactModeling	Popova et al.[PFD15]	2015					✓	Proc	GSM	✓	✓		✓			ProM			✓	✓
S28	Episode Miner	Leemans, van der Aalst[LvdA14]	2015	✓					Proc	Partial Order Graph	✓			✓			ProM	✓		✓	✓
S29	MVPM Mine	Folino et al.[FGP15]	2015	✓					Proc	MVPM	✓	✓		✓			Standalone	✓			
S30	o§	Guo et al.[GWW+15]	2015	✓					Proc	Petri net	✓	✓		✓			ProM	✓		✓	✓
S31	CNMining	Greco et al.[GGLP15]	2015	✓					Proc	Causal net	✓	✓		✓			ProM	✓	✓		✓
S32	Maximal Pattern Mining	Liesaputra et al.[LYC15]	2015	✓					Proc	Heuristic net	✓	✓		✓			ProM	✓	✓		
S33	Discover with ILP using Decomposition	Verbeek, van der Aalst[VvdA14]	2015			✓			Proc	Petri net	✓	✓		✓			ProM	✓			✓
S34	MINERful	Di Ciccio et al.[DMMM15]	2015	✓					Dec	Declare					✓		ProM, standalone	✓			✓
S35	OLAP Discovery	Pizarro, Sepúlveda[PS14]	2015	✓					Multi	Social network, heuristic net	✓	✓		✓			ProM	✓			✓
S36	Supervised Polyhedra	Ponce de Leon et al.[PCvB15]	2015			✓			Proc	Petri net	✓	✓		✓			Standalone	✓			
S37	LocalizedLogs	van der Aalst et al.[vdAKRV15]	2015			✓			Proc	PN, BPMN	✓	✓		✓			ProM	✓	✓		✓
S38	IMD framework	Leemans et al.[LFvdA15a, LFvdA16]	2015, 2016	✓					Proc	process tree, PN	✓	✓		✓			ProM	✓		✓	✓
S39	HybridILPMiner	Van Zelst et al.[vZvDvdA15]	2015			✓			Proc	PN	✓	✓		✓			ProM			✓	✓
S40	DGEM	Molka et al.[MRD+15]	2015		✓				Proc	BPMN based	✓	✓		✓			Standalone	✓		✓	
S41	Ladiges	Ladiges et al.[LHFL15]	2016	✓					Proc	Machine State Petri Nets	✓	✓		✓			Standalone	✓			
S42	RegPFA	Breuker et al.[BMDB16]	2016				✓		Proc	PN, RegPFA	✓	✓		✓			Standalone	✓	✓		✓
S43	Structured Miner	Augusto et al.[ACD+16]	2016	✓					Proc	BPMN	✓	✓		✓			ProM, Apromore	✓	✓	✓	✓
S44	Local	Tax et al.[TSHvdA16]	2016					✓	Proc	process trees	✓	✓		✓			Standalone	✓			
S45	SQLMiner	Schönig et al.[SRC+16]	2016	✓					Dec	n/a***					✓		Standalone	✓			✓
S46	TM	Srinivasan et al.[SBVA15]	2016				✓		Proc	LGTS	✓	✓		✓			Standalone		✓		
S47	PGminer	Mokhov et al.[MCB16]	2016	✓					Proc	CPOG	✓	✓					Workcraft	✓		✓	✓
S48	MINERful-vacuityCheck	Maggi et ail.[MMDM16]	2016	✓					Dec	Declare					✓		Standalone	✓			✓
S49	CSMMiner	van Eck et al.[vESvdA16]	2016					✓	Proc	CSM	✓	✓		✓			ProM	✓			✓
S50	Log OnMap Replayer	de Leoni et al.[dLStHvdA16]	2016					✓	Proc	multi	✓	✓		✓			ProM	✓			✓
S51	Proximity Miner	Yahya et al.[YSB+16]	2016	✓					Proc	Heuristic net	✓	✓		✓			ProM	✓			✓
S52	IMLC	Leemans et al.[LFvdA15b]	2016	✓					Proc	process tree, PN	✓	✓		✓			ProM			✓	✓
S53	TB-MINERful	Di Ciccio et al.[DMM16]	2016	✓					Dec	TB-Declare					✓		Standalone		✓		
S54	MapReduce	Evermann[Eve16]	2016	✓					Proc	PN, HN	✓	✓		✓			Standalone		✓		
S55	Non-Atomic Declare Miner	Bernardi et al.[BCFM16]	2016	✓					Dec	Declare					✓		ProM	✓	✓		✓

*** Finds Declare constraints, but doesn't visualize them.

** Mines models similar to the Fuzzy maps.

* Block-structure models, similar to restricted Petri net with invisible tasks

Table 4: Overview of primary studies

and S42 produce Petri nets, that is 13% of approaches. Approaches S16, S22, S37 and S43 produce BPMN models, that is 7% of approaches. Approaches S21, S26, S32 and S51 produce Heuristic nets, that is 7% of approaches. Approaches S3 and S31 produce Causal nets, that is 4% of approaches. Approaches S6-7, S11, S34, S48, S53 and S55 produce Declare models, that is 13% of approaches. Approaches S13 and S23 produce Hidden Markov models, that is 4% of approaches. Approach S14 produces Hybrid model, which consists of Declare map and a Petri net. That is 2% of approaches. Approaches S2, S41 and S49 produce state machines, that is 6% of approaches. Approach S1 produces Fuzzy maps, that is 2% of approaches. Approach S12 produces WoMan formalism model, that is 2% of approaches. Approach S15 produces directed acyclic graph, that is 2% of approaches. Approach S24 produces directed graph, that is 2% of approaches. Approach S25 produces Petri net, that has BPMN parallel gateways, that is 2% of approaches. Approach S27 produces GSM model, that is 2% of approaches. Approaches S28 and S48 produce partial order graphs, that is 4% of approaches. Approach S29 produce multi variant process model, that is 2% of approaches. Approach S40 produces BPMN based models, that is 2% of approaches. Approach S46 produces logic guarded transition systems, that is 2% of approaches. Approaches S5, S35 and S50 produce multiple process models, that is 6% of approaches. Out of these 55 approaches 6 were selected for testing for our article [ACD+17]. In addition for these six, Heuristic Miner was added to the testing as the winner of latest evluation done by de Weerdt etal[WBVB12]. So, in our follow up paper [ACD+17], the Inductive Miner - infrequent, Evolutionary Tree Miner, α\$, CN Mining, Hybrid ILP Miner and Structured Miner were selected.

4.2 RQ2

This section will answer the question: **"Which kinds of process model (i.e., imperative, declarative or hybrid) are discovered by the existing approaches?"** Figure 4 shows the distribution of model types. Nine of the approaches, (S6-S7, S11-S12, S34, S45, S48, S53 and S55) are declarative. These approaches produce model from Declare family, with exceptions of approaches S12, which produces WoMan formalism model. There is one approach, S14, that is hybrid, which means, it uses declarative and procedural notation on one model. It combines together Declare and Petri net. There are two approaches, S5 and S35, that could produce a separate procedural model or social network. In addition, S5 can produce also dotted charts. Remaining 43 approaches are procedural. So, most of the primary techniques, 78%, are procedural.

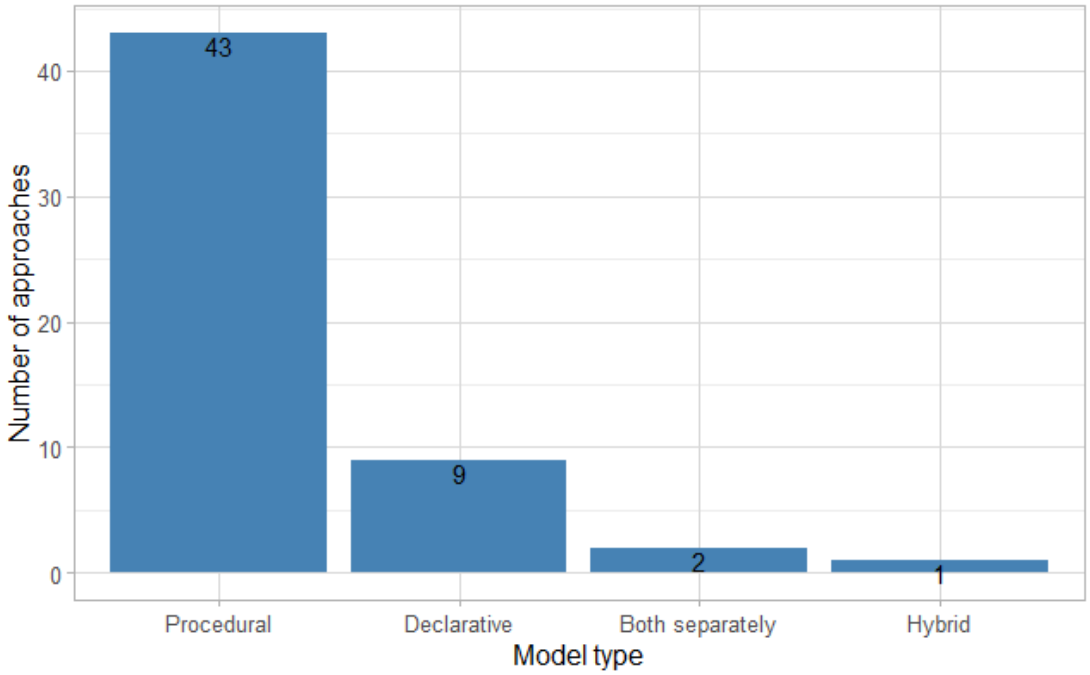

Figure 4: Distribution of model types

4.3 RQ3

This section answers the question: **"Which process constraints (e.g. loop, XOR, parallel) are inferred by the models generated?"** Although not explicitly shown in the Table 4, all the approaches can discover sequences. It can be seen from the Table 4 that only three approaches, S3, S20 and S24, can discover all constraints (concurrency, exclusive OR, inclusive OR and loops). Approach S28 can discover concurrency and loops, but not choice. This can be explained by its models, namely it produce partial order graphs and choice cannot be expressed in that formalism. Approach S2 can discover exclusive choice and loops, but cannot discover parallelism. This is due the modelling notation, finite state machines. Approaches S15 and S47 can discover parallel and exclusive choice constructs from the log, but no loops. In study S15, it is due the usage of directed acyclic graphs and in the study S47 due the usage of conditional partial order graphs. In other words, all the approaches can discover sequence, one approach cannot discover parallelism, one cannot discover exclusive choice, only three can discover inclusive choice and two cannot discover loops. Figure 5 gives an overview on how approaches discover constructs. It is also common, 71% of studies (39 approaches), to discover parallelism, XOR choice and loops.

From Figure 6 we can see that, 44 approaches can find parallelism, 44 ap-

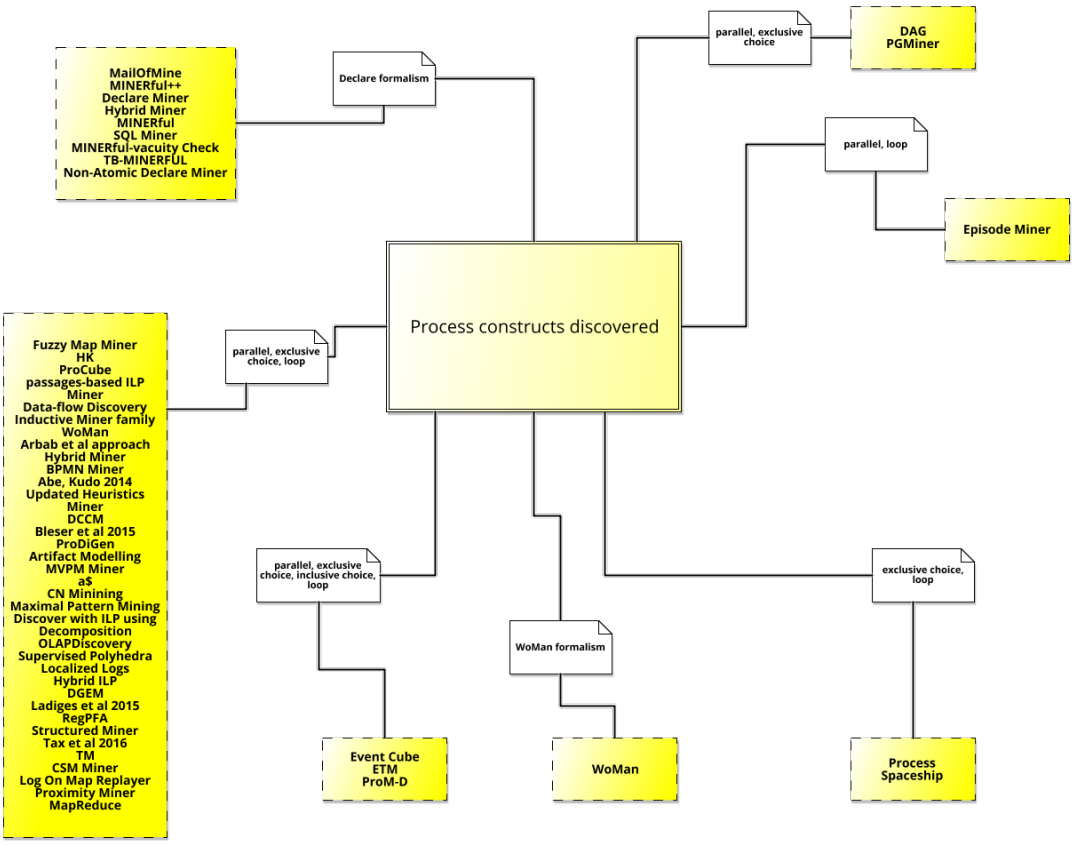

Figure 5: Discoverable process constructs

proaches can deal with XOR, three approaches can deal with inclusive-OR and 43 approaches can discover loops in the "classical" process constructs ways. In addition, 1 approach uses WoMan formalism and nine approaches use Declare formalism. Note that Inductive Miner family means all the variants of Inductive Miner from the Table 4.

One of the issues in the PD field could be the lack of expressing inclusive choice explicitly. To leverage this issue, the combination of process trees and genetic mining seems to be promising.

4.4 RQ4

This section answers the question: **"What tools exist to perform process discovery?"** From the Table 4 we can conclude that ProM is the most used tool

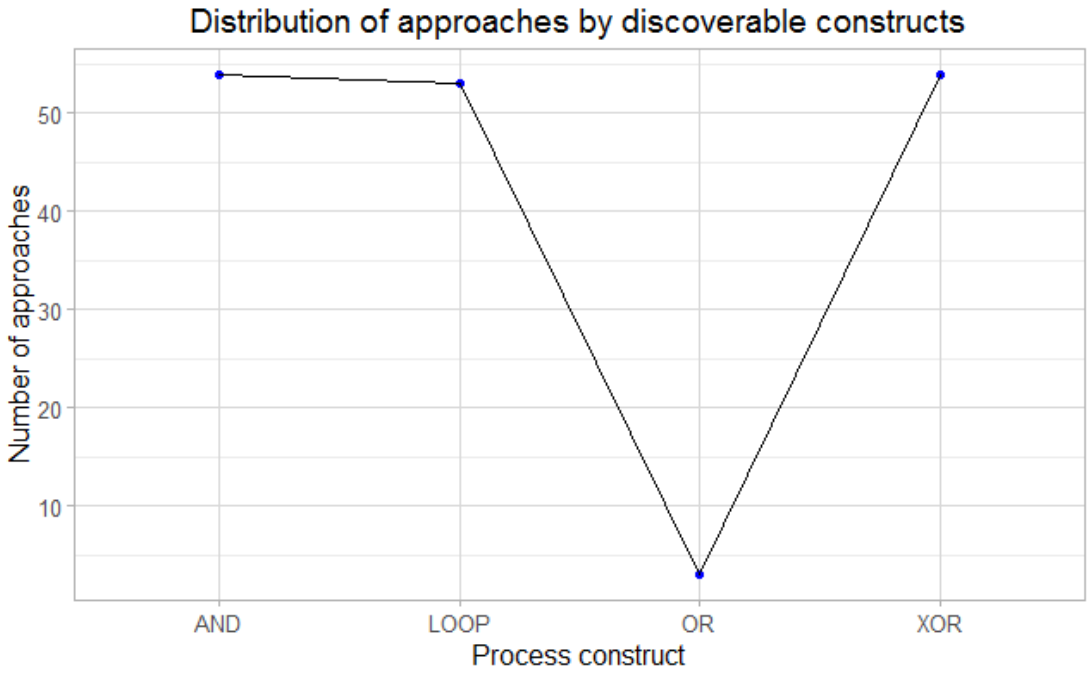

Figure 6: Distribution of constructs

for the process discovery. We can also see that approaches could be implemented as plug-ins for Eclipse, Apromore and Workcraft, or as standalone tool. So altogether five possibilities. To get the full picture of the possible tools, we must look deeper than the Table 4. So, other frameworks (in addition to the ones named), that could be used for process discovery, are Celonis Discovery [Cel17], Perceptive Process Mining citeperspective, Fuzzy Miner [GvdA07], SPMF [FGG+14], PMLAB-suite citecarmona2014pmlab, BPMNAnalysis [Bay11], SYNOPS [LHEZ12], Declarative Process Model Learner [CLM+09, CLRS10, BRL10]. This list bases on the analysis of primary articles. From Figure 7 we can see that 31 approaches have a plug-in for ProM. 24 approaches are available as standalone. Out of these 23, two, approaches S7 and S34 have also implementation as ProM plug-in. One approach, S2, is implemented as plug-in for Eclipse. One approach, S16, has been implemented on top of Apromore platform, but also has implementation as ProM plug-in and as standalone. S47 has been implemented as plug-in for Workcraft platform, but it also has a standalone implementation.

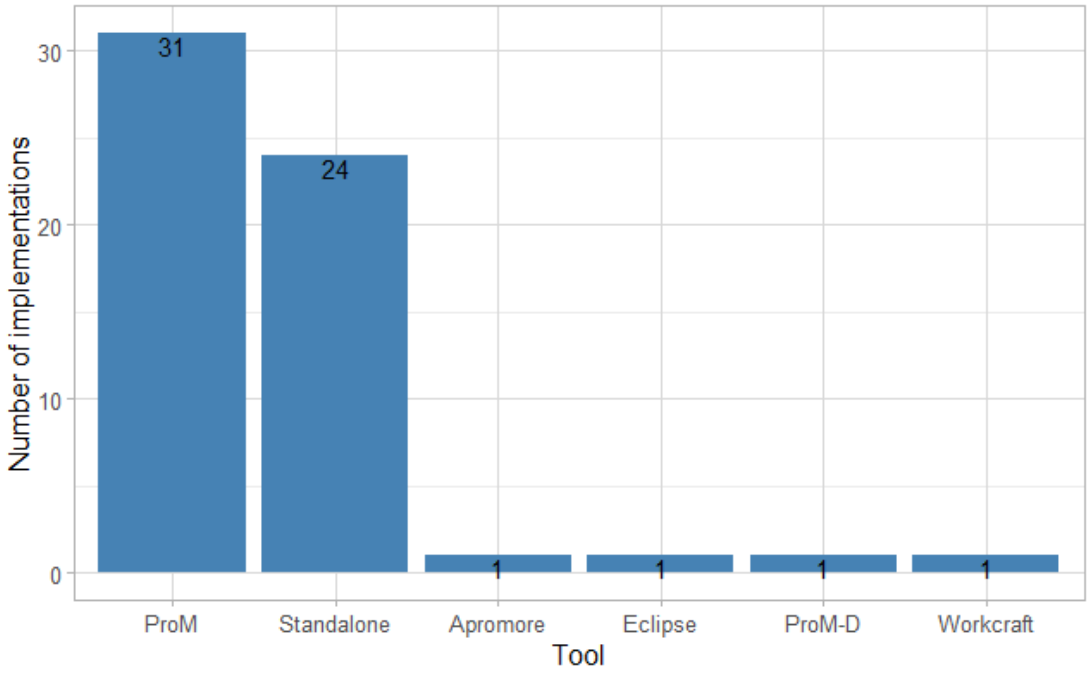

Figure 7: Implementation numbers

4.5 RQ5

These primary approaches could be tested in the context of real-life, synthetic and artificial logs. They could also be tested in combinations out these three. To answers the question: **"How the existing approaches have been evaluated?"**, we can say based on the Table 4 that primary approaches are mostly (44 out of 55, 80%) tested in the context of real-life environment. Approaches S2, S4, S6, S7, S9, S11, S15, S16, S24, S31, S32, S37, S42 and S55 are tested in real-life and synthetic context. Approaches S8, S12, S20-21, S28, S30, S38, S40 and S47 are tested in the real-life and artificial context. Approach S43 has been tested in context of real-life, synthetic and artificial logs. Approaches S3, S10, S19, S27, S39 and S52 have only been tested with artificial log. Approaches S22, S26, S46, S53 and S54 are tested with only synthetic log. The corresponding numbers are illustrated in the Figure 8.

Real-life logs usually contain full spectrum of behaviour and reflect how the process is really undergoing. To create a synthetic log, the process model is replayed number of times and the actions are logged. Artificial logs are useful for creating special cases to test the capability of an approach. Using only one type of log for testing creates the risk that some important behaviour is missed. Thus using more than one context for testing indicates higher quality of the evaluation.

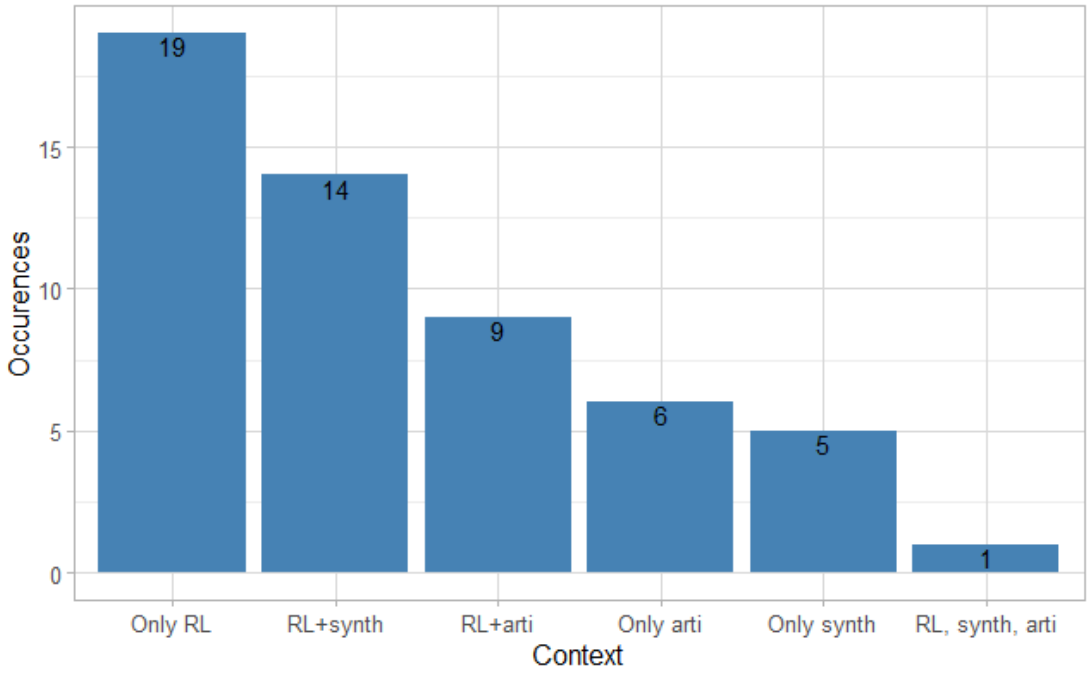

Figure 8: Occurrences in test contexts

The performance of the approaches have also been tested with comparison to the others. Only approaches S2, S3, S5, S6, S9, S10, S17, S22, S27, S46, S47, S50 and S54 are not been tested in the relation to the state-of-the-art.

4.6 RQ6

With previous section we answered the question about how the testing is done. In this section we are going to answer the question relating to the domains used, namely **"In which domains have existing process discovery approaches been applied?"**. Table 5 gives an overview of the real-life domains used for evaluation in the literature. Some of primary approaches from the Table 4 have been tested with Business Process Intelligence Challenge (BPIC) logs. In these challenges, real-life logs are used. BPIC takes place annually from year 2011. Following associations exist between BPIC logs and approaches :

- **BPIC11** - this log is from a Dutch academic hospital describing patients diagnosis and treatment in Gynaecology department. Approaches S6, S7, S11, S18, S32, S38 and S45.

- **BPIC12** - this log is from a Dutch Financial Institute and describes the loan application process. Approaches S6, S7, S8, S9, S11, S14, S15, S18, S25, S28, S33, S34, S38, S40, S42, S44, S45, S49, S52 and S53.
- **BPIC13** - this log is from Volvo IT Belgium and describes incident and problem management process within VINST system. It is used for evaluating S11, S21, S29, S42, S48 and S55.
- **BPIC14** - this log is from Rabobank Group ICT and describes the work of service desk, namely interaction management, incident management, and change management process. Approaches S11 and S53 are tested with this log.

The following list gives overview of the contexts that are used for evaluation.

- **Automotive** - The usage of BPIC13 log
- **Administrative** - United States Patent Classification patent applications process for category 435 between 2000 and 2005 (used in S4), Dutch municipality logs (events that correspond to citizens objecting to the valuation of their houses and events that correspond to citizens that request for building permits in two different logs) and public works log (events related to invoices at a provincial office of the Dutch national public works) (used in S8), project applications handling in the Belgian research funding agency IWT for the applied biomedical research funding program containing events from 2009 to 2012 (used in S16), CoSeLog project logs (non-public logs of a building permit approval process in five municipalities used in S18, not specified in S20), De Lijn log (events related to the customer-complaint process in a company, used in S21), WABO1BB (a log from a building permit approval process of a Dutch municipality, used in S25), event traces for four different applications to get a license to ride motorbikes or to drive (used in S30), product recall process defined by Wynn et al. (used in S31), booking flight system log (used in S37), a log pertaining to a road traffic fines management process (used in S48), log related to the process for handling house-building permit applications in a Dutch municipality (used in S50)
- **Health care** - The usage of BPIC11 log, diagnostic and treatment events from a hospital department (used in S8), an eHealth process which contains 18 activities from [GG14] (used in S40) and a log of baby feedings using a smart baby bottle equipped with various sensors that was developed by Philips (used in S49)
- **Insurance** - The usage of Suncorp, an Australian insurance company, log (commercial insurance claims handling process executed in 2012)

- **University systems** - Data generated by the students usage of S35
- **Banking** - BPIC14 log and multilayer banking system log (used in S37)
- **Financial** - BPIC12 log, a fraud detection process (used in S21)
- **Benchmark** - Logs used in process mining community for benchmarking, such as caise2014, complex, confdimblocking, documentsflow, fhmexamplen5, incident, purchasetopay, receipt, BigLog1, Log1, Log2, DigitalCopier, softwarelog, incidenttelco, svn log, telecom.
- **Logistics** - eight event logs from the foodstuff supply information system used in several ports across China (used in S24), the circulation of buses data across a network of two neighbouring Italian cities, Cosenza and Rende, during the year 2012 (used in S31), a port logistics process of landside transport (used in S51)
- **Services** - log of a large Dutch agency that rents houses and apartments representing the cancellation of a current rental agreement and subsequent registration of a new rental agreement (used in S1)
- **Hardware** - This domain includes industrial machinery in S41 and test events for the deployment of high-tech equipment, containing both in-factory tests and on-site test events (used in S8)
- **Software** - SAP/R3 collection and IBM BIT collection (used in S43), method calls executed by Rapid Miner (used in S38), Robostrike, log representing complex logic of a real-world online game service (used in S2), events from a web server (used in S8), logs describing of 20 successive days of an application server behaviour (used in S17), click-stream data from a dot-com start-up (used in S38) and Gazelle data, real life data set used in the KDD-CUPâĂŹ2000 containing customers web click-stream data provided by the Blue Martini Software company(used in S44)

This covers the body of testing done with real-life logs.

Name	Domains													
	Automotive	*Administrative*	*Health care*	*Insurance*	*University systems*	*Banking*	*Financial*	*Benchmark*	*Logistics*	*Software*	*Software*	*Software*	*Services*	*Hardware*	*Software*
S1													✓		
S2											✓				
S4		✓													
S6			✓				✓								
S7			✓				✓								
S8		✓	✓				✓			✓				✓	
S9							✓								
S11	✓		✓			✓	✓								
S12															
S13															
S14							✓								
S15							✓								
S16		✓		✓											
S17										✓					
S18		✓	✓				✓								
S20		✓													
S21	✓	✓					✓								
S23															
S24									✓						
S25		✓					✓								
S28							✓								
S29	✓														
S30		✓													
S31		✓							✓						
S32			✓												
S33							✓								
S34							✓								
S35					✓										
S36								✓							
S37		✓				✓									
S38			✓				✓			✓		✓			
S40			✓				✓								
S41														✓	
S42	✓						✓								
S43															✓
S44							✓			✓					
S45			✓				✓								
S47								✓							
S48		✓													
S49			✓				✓								
S50		✓		✓											
S51									✓						
S52							✓								
S53						✓	✓								
S55	✓														

Table 5: Overview of applied domains

5 Evaluation

In this section we describe a comparative evaluation of process discovery approaches with domain experts. In the subsection 5.1 the general description of the log used is given. In 5.2 the list of miners used and their settings are specified. In 5.3 the process for creating a refined log and using it for model discovery is described. In 5.4 the setup for the domains experts evaluation is described. In 5.5 the methods for conducting statistical analysis, are presented. Finally, in 5.6 the results of the evaluation are presented and discussed. In general, the purpose of this evaluation is to assess of usable are the business process models created by automated approaches for the domain experts with respects to four quality metrics mentioned in 5.4.

5.1 The log used

In our evaluation, a process log from a software company was used. It described the process of incident handling and originated from a task list software called Product Idea Management. It consists of three years data: from 2014 to the end of 2016. It has 265562 events, 49549 traces, and 33 activities. Since we had the information that the incident handling process was restarted every year and our intention was not to evaluate the detection of concept drifts, we decided to use only data from 2016. The log did not have explicit information about activities performed. Therefore 5.3 describes how the data was extracted from the log, and how the log was made acceptable for process mining algorithms.

5.2 Miners used

The selection of the miners is based on two criteria. At first, it needed to be present in the SLR. The second criteria was that the implementation needed to be freely available.To create models the, following miners were used:

- **BPMN Miner** - Heuristics Miner 6 was selected to mine the initial structure and CreatorDept was selected as primary key candidate. All other fields were left as default. The implementation in ProM 6.5.1 was used.
- **Structured Miner** - Heuristics Miner was used for the mining, the Java Virtual Machine maximal heap size was increased to 6GB, all other fields were left default. Implementation downloaded from the Apromore was used.
- **Causal Net Mining** - Default settings were used, in variant two creates null pointer exception
- **Heuristics Miner 6** - Default settings were used

- **Heuristics Miner 5** - Default settings were used
- α**\$** - Default settings were used, exceeds two hours in variant two
- **Hybrid ILP Miner** - Fuzzy miner was used to find the causal structure, other settings were left default. Version in the ProM 6.5.1
- **Inductive Miner - incompleteness** - Default settings were used
- **Inductive Miner - all operations** - Default settings were used
- **Evolutionary Tree Miner (ETMd)** -> Time limit was setted to two hours, also 1000 generations limit was used, fitness limit 0.90 (if the candidate had lower value, it was dropped). All other settings were left default. ProM 6.5.1

In all the miners the latest available implementation was used. In the evaluation we used only procedural approaches. A comparison between procedural and declarative approaches is out of the scope of this thesis.

5.3 From log to the models

As mentioned before in section 5.1 the log was not suitable for the evaluation. So we needed to pre-process it. We converted the .XLSX Excel worksheet provided by company into .CSV format. Then we added a new column to the log, since the current format was not suitable to identify activities. The added field consisted of the name of the function that the assignee of a task had in a company and the country code, e.g. UK Head. This information was available for us in a separate Excel worksheet. Then we filtered the log to only have data from year 2016. In addition, we removed all traces that were shorter than five events. The resulting .CSV file was translated to an XES event log. The resulting log had 52629 events, 5551 traces and 29 activities. When applying algorithms to the log, all the procedural models were spaghetti like. Therefore we decided to further filter this log to isolate more frequent behaviours. We used Disco for this. We created nine separate logs ranging from a log all behaviour present up to a log containing behaviour that is shared by at least 9 cases. We could not filter more, because otherwise we would have had less than 500 traces which would have been a threat to validity. These logs were then used to produce a model. If the model was not natively BPMN, it was converted to BPMN. Petri nets were converted by the algorithm available in the ProM 6.5.1 by Raffaele Conforti (the same algorithm, that is used in the BPMN Miner, called "**Convert Petrinet to BPMN**"), and process trees were converted by using algorithm available in the ProM 6.5.1 by A. Kalenkova (named "**Convert Process tree to BPMN diagram**"). In this

set of experiments, the three most promising models were mined by Structured Miner, Inductive Miner - all operations, and Evolutionary Tree Miner when using log with at least 9 repeating cases. The other models were clearly spaghetti like and not suitable for a user evaluation. So these three: i)Structured Miner (model B); ii)Evolutionary Tree Miner (model A); and iii)Inductive Miner - all operations (model C) were selected for evaluation with domain experts.

5.4 Evaluation set-up

We decided to carry out a two stage evaluation. With non-domain experts, we decided to carry out one stage evaluation. The non-experts were represented by the first year students of University of Tartu from Conversion Master in IT masters curriculum. The team leads, Product maintenance team, QA and a mix of departments from the company were in the role of the domain experts.

The first stage of the evaluation was a questionnaire. The questions were available for the individual answering through Google Forms. With this we avoided the effect of authority influence on the group decisions. The domain experts were asked questions relating about if and how they recognised their processes from the models. For the non-experts, the questions were a bit modified, since they don't have any underlying knowledge about the company's processes. Nevertheless, these questions were about the same model quality metrics. The questions for domain experts are in Appendix A and for students in Appendix B. In both cases all questions were required to be answered. The participants also had an option to insert comments after each model quality question. The model quality questions answer options were presented in a form of a grid. On the X-axis are the numbers from 1 to 7 as possible answers and on the Y-axis are the three models, A, B and C. In each row the answer was required. The grid is illustrated in the Figure 9.

These questions correspond to the following business process models quality metrics:

- **correctness** - Question 4 in experts case, not used in the non-experts case;
- **precision** - In experts case question 5, not used for non-expert
- **understandability** - Questions 1, 2 and 3 in both cases. For non-experts case also question 4.
- **usefulness** - Questions 5 and 7 in non-experts case, in experts case question 6.

For each question we had three variants as A, B or C representing the miner used to create the model. We used such codified letters to avoid biases (e.g. somebody

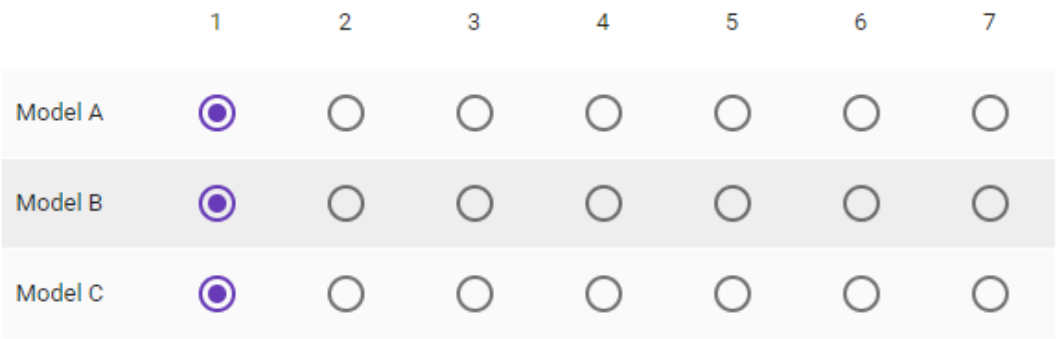

Figure 9: The grid used in questions

could prefer a one miner from previous experience). We restricted ourselves to three variants also to avoid getting meaningless results due relativity low amount of participants. If we would have more than three models, the answers could be distributed in a way where it wouldn't be possible to make any statistically strong conclusions.

In the second stage, we carried out a workshop. This second stage was directly followed after first stage. It was in open form, so the participants could express their feelings and give quantifiable feedback about the models presented. We recorded the workshop. For the questions used to moderate the workshop see the Appendix C.

The experiments for the domain experts and non-experts were carried out separately. At first with non-experts and then, on a separate day, with domain experts.

The models for evaluation are in the Appendix D and all other models generated in the Appendix E.

5.5 Description of statistical analysis methods used

With statistical analysis we wanted to discover if there is any differences of the ratings between the models. Before the analyses, the data was formatted to be suitable for data analysis with the free software R. The answers of the questionnaire was extracted from Google Forms as .CSV file. At first the name of the questions in the header of the .CSV file were renamed to correspond to the respective metrics. If there was more than one question for a metric, the ratings of the subquestions were divided by the number of subquestions and then summed

to reflect the general rating of the respective process model quality metrics.

To be able to compare the models based on the different aspects, we needed the scales of each of the metrics to be comparable. So for each of the metrics, rating of 1 meant that the model do not correspond to the respective characteristic at all and rating of 7 meant that the model fully corresponds to the respective characteristic. So if the model is evaluated based on correctness, understandability, and usefulness, the best model should have highest score. For precision, the original scale from 1 to 7 indicated a model that is too specific or too general, respectively. So the scale was changed so that rating 4 was rated as 7 (model is not too specific or too general) and rating 1 or 7 was rated as 1 (both too specific and too general model has the lowest rating). It was done to be able to compare the different metrics on the same scale. To calculate the overall scores for the models, in experts case we used formula **(1/4 * understandability + 1/4 * usefulness + 1/4 * correctness + 1/4 * precision)**. For students, the formula was **(1/2 * understandability + 1/2 * usefulness)**.

We formulated the following hypotheses pairs:

- **The null hypothesis**: There is no difference in the mean rating of the models **The alternative hypothesis**: There is at least one model that is different from the others
- **The null hypothesis**: There is no difference in the mean rating of the subgroups **The alternative hypothesis**: There is at least one subgroup that is differently rated from the others
- **The null hypothesis**: There is no significant interaction between the model type and metrics subgroup **The alternative hypothesis**: There is an interaction between the model type and metrics subgroup, meaning that the rating of metrics subgroup depends on the model type.

We tested the hypotheses only within experts, because the students were used as a control group to ensure the validity of the questionnaire.

The hypotheses were tested using the two-way ANOVA. If we assume the independence of the observations, then the ANOVA model additionally assumes that the residuals are normally distributed for each combination of the groups and the residuals have the same variance (homogeneity of variances) for each combination of the groups. Normality assumption was assessed with Shapiro-Wilk test and QQ-plots, homogeneity of variances was assessed with Levene's test. A significant ANOVA test was followed by Tukey HSD test to perform multiple pairwise-comparison between the means to determine the statistically significant pairs of groups.

All the analysis was done in R by using RStudio and the figures were made with ggplot2. Violin plots were used due their expressiveness of median value, interquartile range, and kernel density estimations.

5.6 Evaluation results

In the students case, we had 21 persons present out of 41 students registered to that curriculum. Seven team leads, six product maintenance team members, six QA members, and five persons from the mixed group were present during the domain experts evaluation. So all together four groups and 25 domain experts out of ca. 50 employees of the company.

In the students case, the fastest response to the questionnaire took 12 minutes and 52 seconds, the slowest response took 32 minutes and 37 seconds, and the average questionnaire filling time was 20 minutes and 56 seconds. In domain experts case, the fastest time was 6 minutes and 56 seconds, the slowest time 34 minutes and 4 seconds, and the average time over all groups was 19 minutes and 49 seconds.

The discussion about background questions is available at Appendix F.

At first glance, the students found the model A rather difficult to understand, since the median value is 2.0 and the mean value 2.048, model B was found as rather medium to understand, since the median value is 4.0 and mean value 3.381, and model C appeared to be somewhat easy to understand, since the mean value is 4.762 and median value 5. Also students would rate at first glance model C to be easiest with higher probability than model B. Model A would rated with score under 4 at first glance with high probability. Experts found model A difficult to understand at first glance, since the median value is 3 and mean value 3.25, model B as as medium to understand, since the median value is 4.5 and mean value 4.542, and model C as somewhat easy to understand, since the median value is 5 and mean value 4.792. In experts case, the model would be again rated to be easy with highest probability. When answering the question 2, the students found model A to have rather medium understandability, since the median value is 4 and the mean value 3.429, model B to have somewhat easy understandability, since the median value is 5 and the mean value 4.476, and model C to have rather easy understandability, since the median value is 6 and mean value 4.905. The experts found model A to have somewhat difficult understandability, since the median value is 3 and mean value 3.667, model B to be somewhat easy to understand, since the median value is 5.0 and mean value 5.167, and C to have easy understandability, since the median value is 6.0 and mean value 5.417. In students case, in the context of Q2,

the model A would have ratings from 1 to 6 with almost same probability, and model C would be rated as best one with highest probability. Experts would rate model A with rating between 2 and 4 with highest probability, and model C with rating around 6 with highest probability. We can also see that for experts Q2, the model B would be rated with rating from 4 to 6.5 with highest probability. When answering Q3, the students found model A to have somewhat difficult understandability, since the median score is 3.0 and mean 2.714, model B to have somewhat easy understandability, since the median score is 5 and mean 4.095, and model C to have also somewhat easy understandability, since the median score is 5 and mean 4.571. Experts found model A to have somewhat difficult understandability, since the median value is 3 and the mean value 3.375, model B to have easy understandability, since the median value is 6 and mean value 5.333, and model C to have easy understandability, since median value is 6 and mean value 5.25. It is interesting to see that in Q3 the experts would rate model B with rating 6 with higher probability than in model's C case. For understanding what type of process is described in a model, the students found model A to be somewhat difficult, since the median value is 3.19, and models B (mean 4 and median 4) and C to be medium (median 4 and mean 3.857). The students ratings are shown in the Figure 10 and experts understandability ratings in Figure 11.

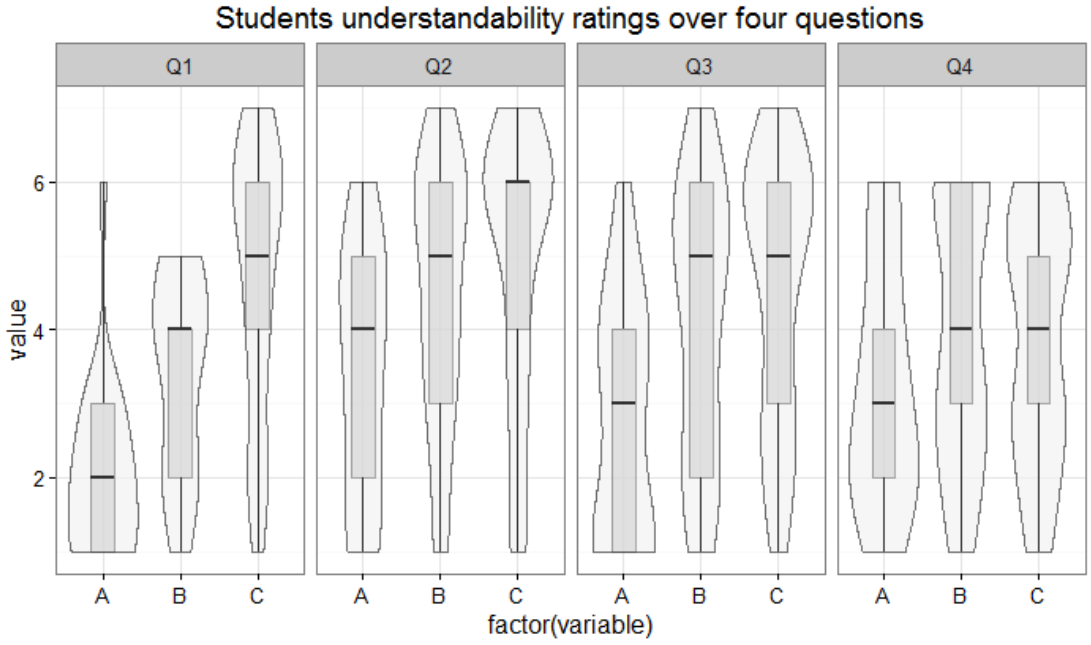

Figure 10: Students ratings for each understandability question

Experts overall rating for each model in the sense of understandability is shown

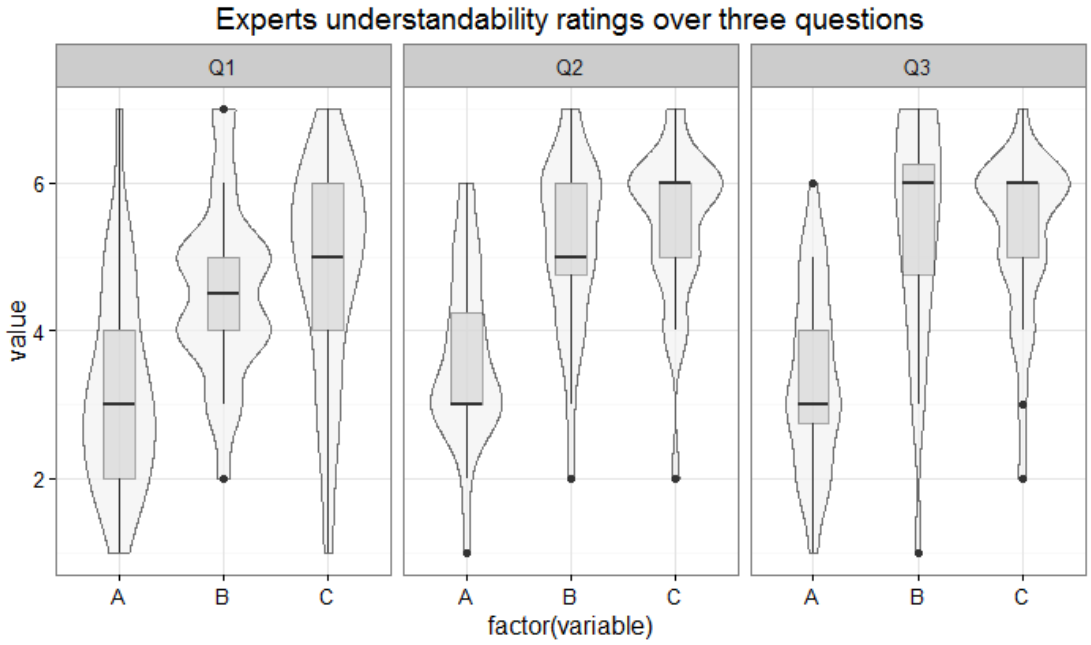

Figure 11: Experts ratings for each understandability question

in the Figure 13 and students overall understandability rating in Figure 12. From these models we can see, the students would rate model A with somewhat difficult understandability, model B with rather medium understandability, and model C with rather somewhat easy understandability. We can also see that in students case, the model A would have a rating between 2 and 4 with highest probability, model B would have rating between 6 to 3 with highest probability, and model C would have rating over 4 with highest probability. Experts found model A to somewhat difficult to understand, model B to somewhat easy to understand, and model C to be rather easy to understand. When selecting the best one in the context of understandability, we used mean values, as described in the Section 5.5. Students found that model C is the best (mean value 4.524) in the terms of understandability, second would be model B with mean value 3.988 and third model A with mean value 2.845. Experts found model C to be best one in the context of understandability, with mean value 5.153, followed by B, with mean value 5.014, and as third best, model A, with mean value 3.431.

In the sense of informativeness, the students found model A to be somewhat useless, B to be somewhat useful and C to be neither useful nor useless, whereas domain experts found models B and C to be somewhat useful and model A as somewhat useless. Students opinion is presented in the Figures 14 and 15, and experts opinion in Figure 16. In students case, the best one in the term of usefulness

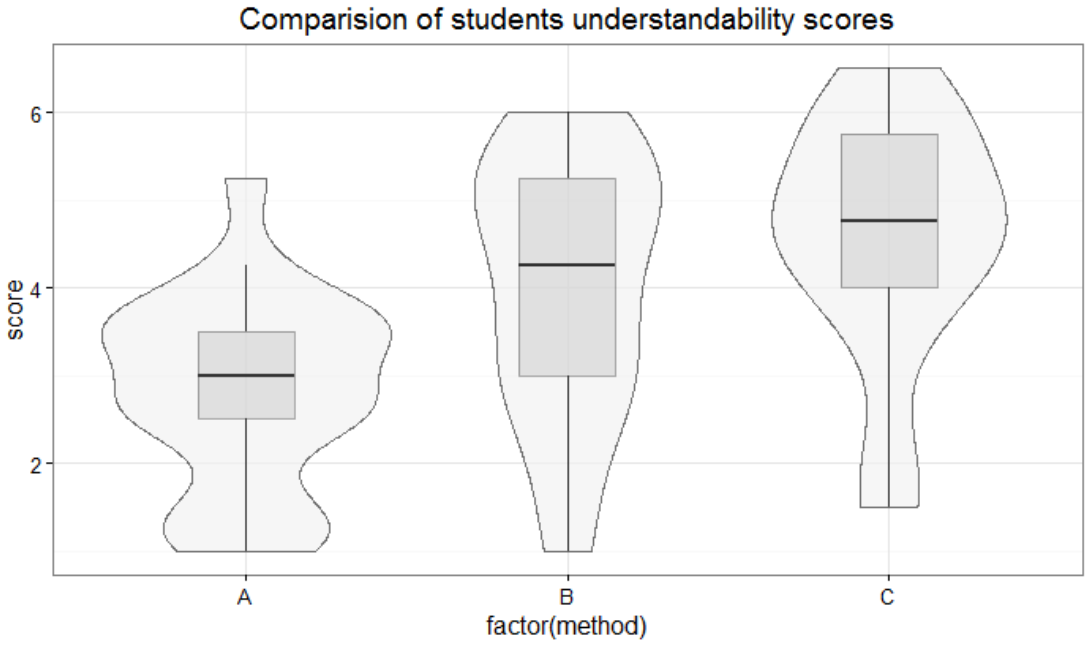

Figure 12: Students overall understandability rating

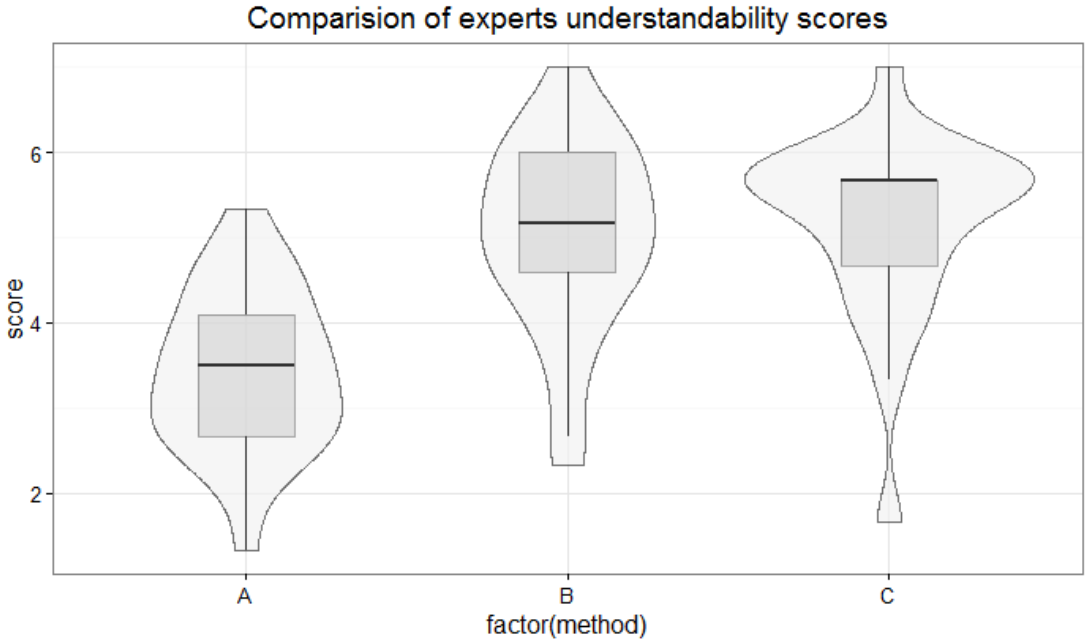

Figure 13: Experts overall understandability rating

is model B with mean value 4.238, followed by model C with mean value 3.833, and as third one, model A with mean value 3.524. Experts would rate to be most useful the model C with mean value 5.083, followed by model B with mean value 4.25, and as third one model A with mean value 3.625.

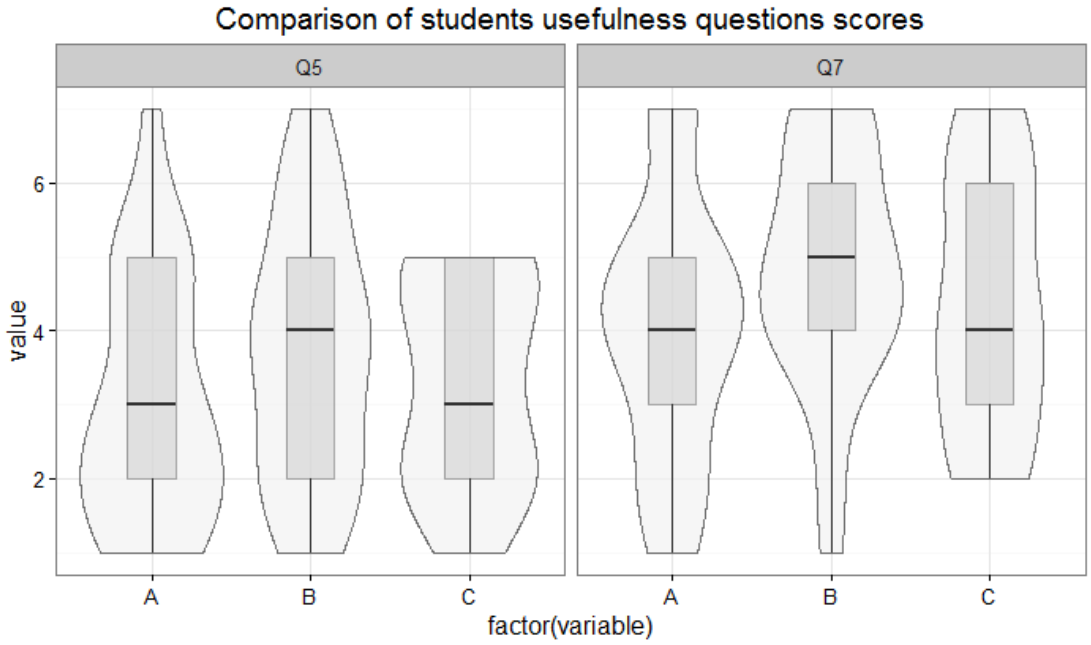

Figure 14: Students usefulness rating

Domain experts found all the models to be somewhat correct as shown in the Figure 17. In their opinion the most correct one is model C with mean value 4.958, then model B, with mean value 4.875, and then model A with mean value 4.208.

In precision perspective, the domain experts would rate model A with score under 5 with highest probability, model B with score around five with highest probability, and model C with score over 5 with highest probability. It is shown in the Figure 18. To evaluate this and compare with other three metrics, the scale was modified as mentioned in Section 5.5. The most precise would be the model C with mean value 5.25, followed by model B with mean value 4.833, and as third, model A with mean value 4.667. We can also conduct that in case of model a the experts were unsure about the precision of model A, because the answers all scattered around the spectrum from 3.5 to 7, when considering the shape of violin.

Based on the Figure 19 and the mean scores, the best model for experts is B with mean value 4.639, followed by with model C with mean value 4.549, and as third best, model A with mean value 3.753. Based on the Figure 20 and mean values, the best model for students was model C with mean value 4.179, followed

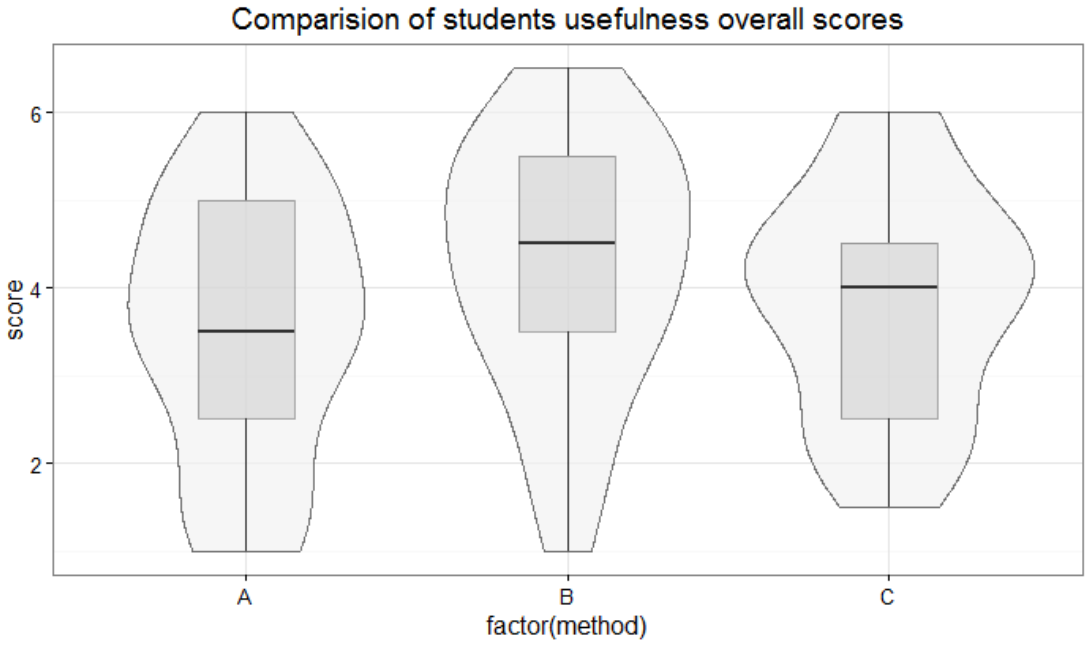

Figure 15: Students usefulness overall rating

Figure 16: Experts usefulness rating

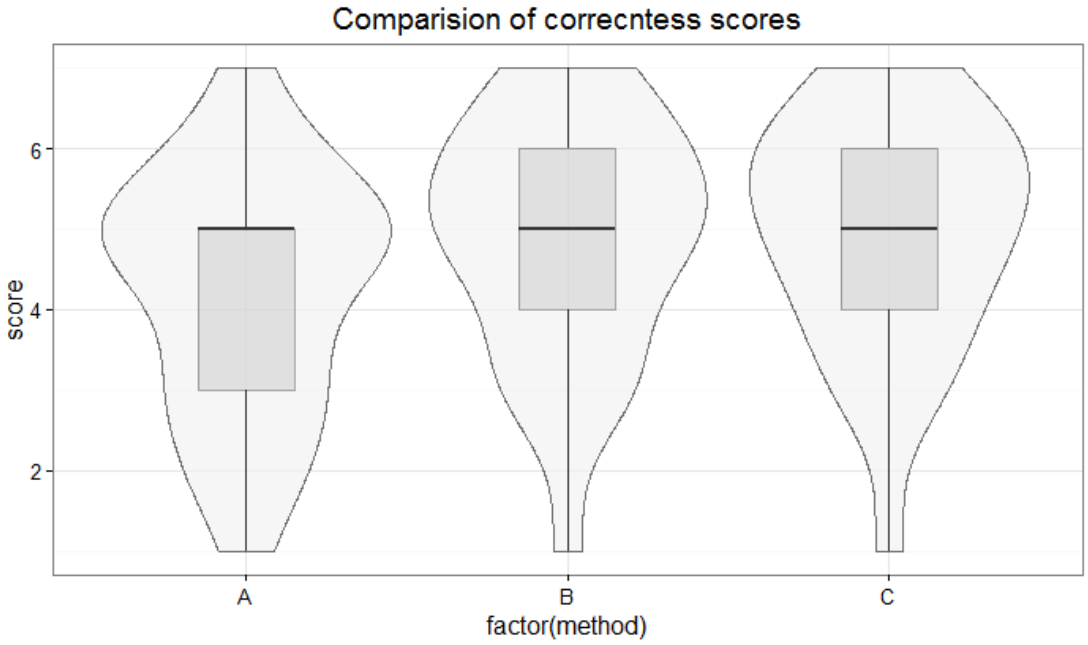

Figure 17: Experts correctness rating

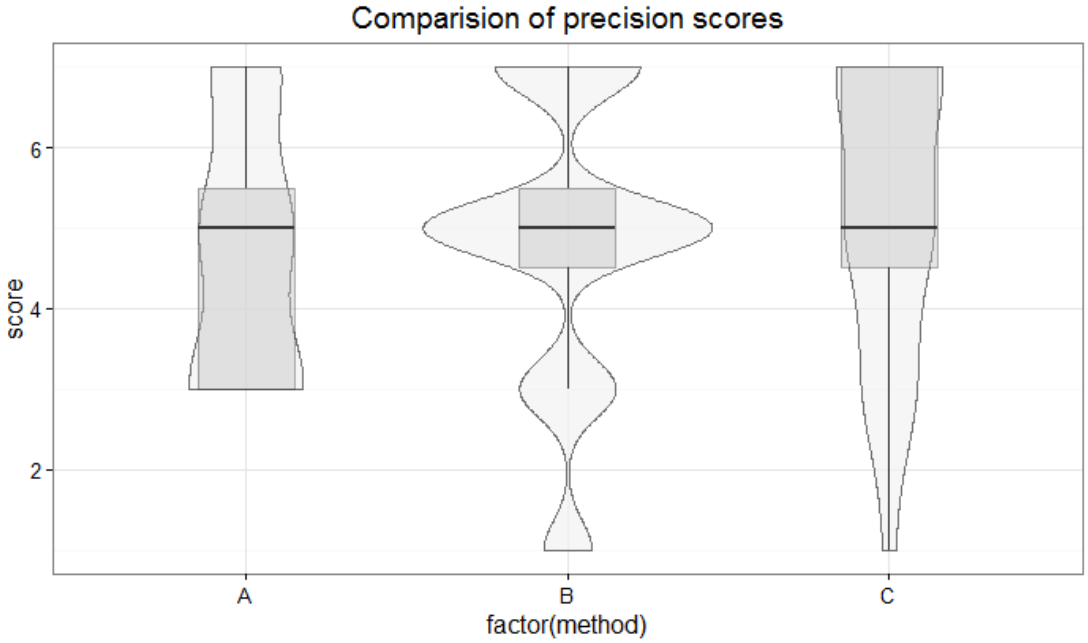

Figure 18: Experts Q5 opinion

by model B with mean value 4.113, and as third, model A, with mean value 3.185. Figure 21 illustrates the comparison of overall ratings of models between students and experts. To do this, we only considered the same quality metrics covered, usefulness and understandability. Figure 22 compares the experts and students rating in the context of understandability. The Figure 23 compares the experts and students rating in the context of usefulness. In these models, students scores are with grey colour(the box and violin), and experts scores are with red(violin) and blue(box). From Figure 23 we can conclude that experts rate models A and B with similar usefulness than students. In model C case, the experts found it to be more useful then students do. From Figure 22 we can conclude that experts and students would rate model A in similar way, that experts would give higher scores to model B than students, and in model C the experts scores are concentrated around 6 and are less distributed than in students case.

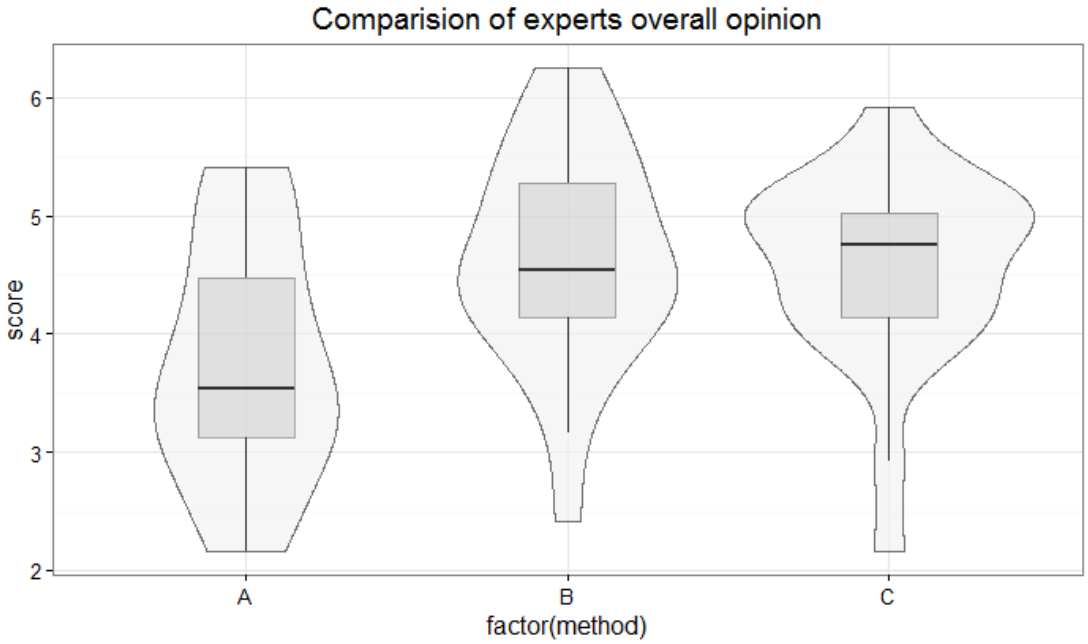

Figure 19: Experts overall rating

Figure 24 shows the experts opinion for each category in plot. From that figure we can see, that the group understandability is with biggest difference. From the interaction plot (Figure 25) we can see that in the model A case, the effect of understandability and usefulness is different from the the effect of correctness and precision. We can also see that in models B case, the effect of usefulness is different from the effects of precision, correctness and understandability. In models C case there is no difference between effects of metrics.

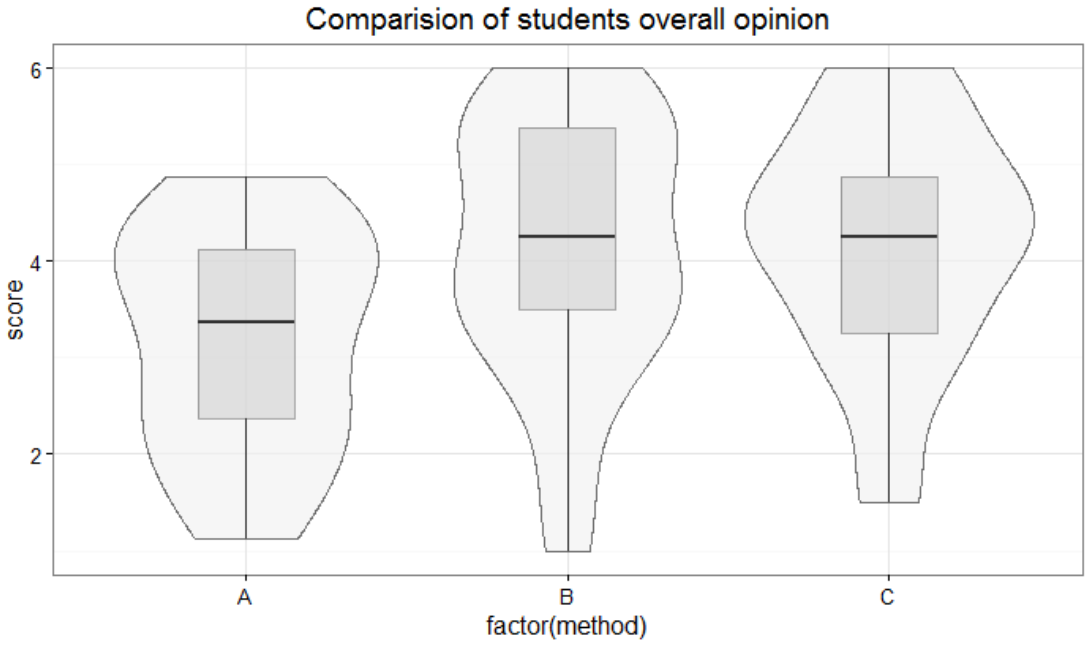

Figure 20: Students overall rating

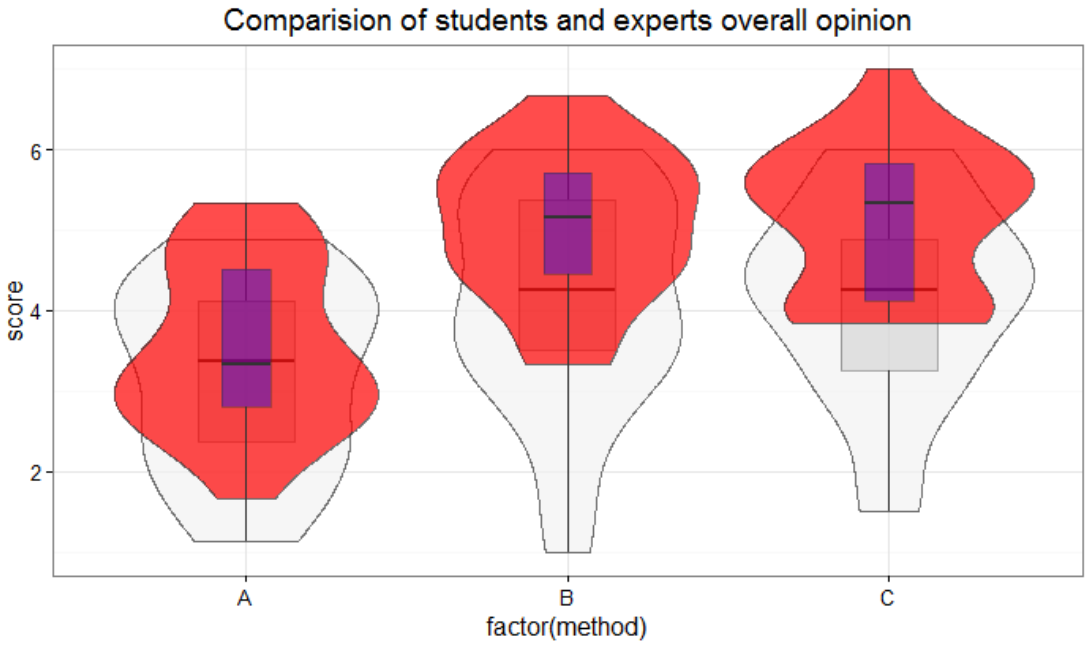

Figure 21: Students vs experts overall rating

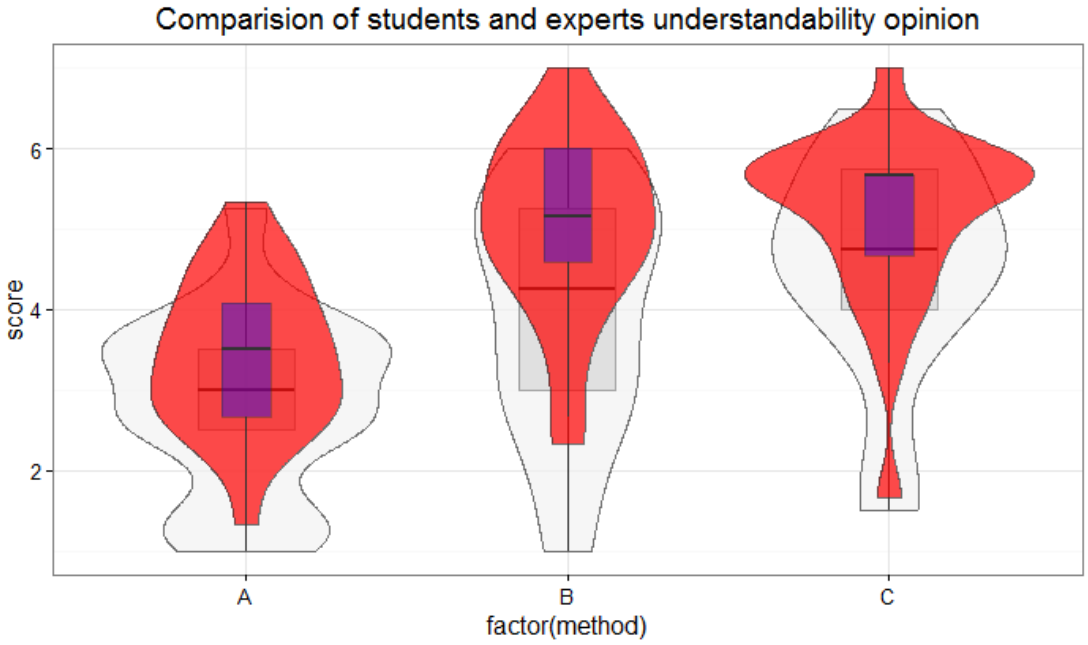

Figure 22: Students vs experts understandability rating

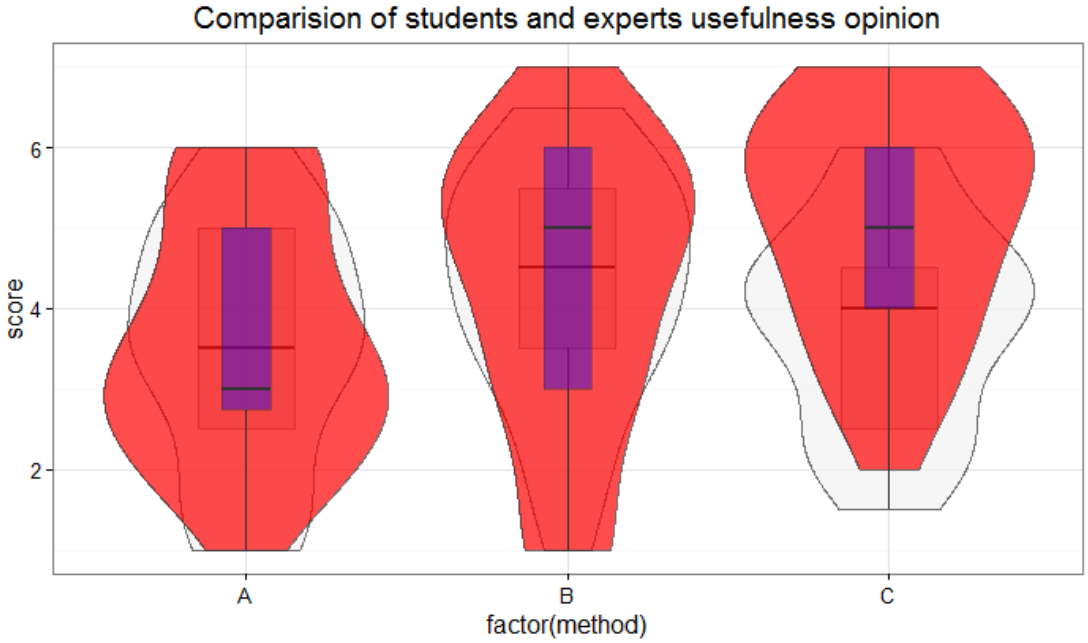

Figure 23: Students vs experts usefulness rating

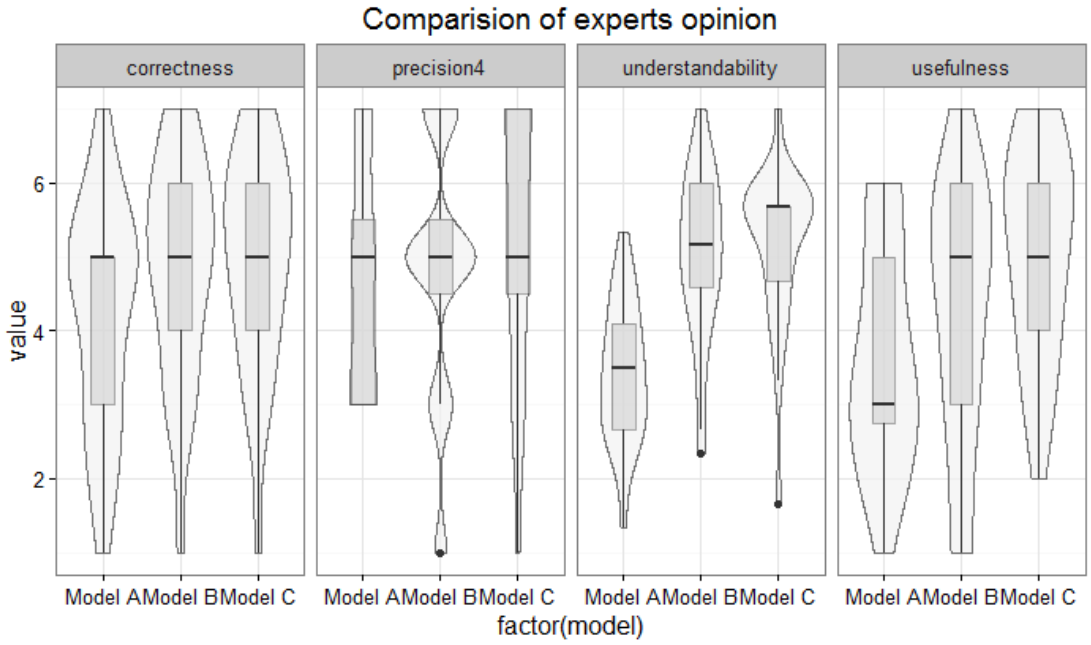

Figure 24: Experts all questions score

At first the hypotheses null : *i*) There is no difference in the mean rating of the subgroups, and *ii*) There is no significant interactions, were tested. Results indicated that there is no significant interaction, since the p value was 0.2211. The p-value for the subgroups (precision, correctness, understandability, usefulness) was 0.1083 which indicates that there is not enough evidence in the data to conclude that the different aspects of the model are differently valued. Based on this, we have to to stay on the two null hypotheses mentioned before.

The null hypothesis "There is no difference in the mean rating of the models" was also tested. The group model p-value was 1.259e-06 that indicates that there are significant differences between the models. We also found that model B is statistically significantly different from model A over all categories. Then that model C is statistically significantly different from model A over all categories. Also, that models B and C are not evaluated statistically significantly different over all categories. This indicates that models B and C are statically significantly different form model A, and that there is no statistically significant difference between models B and C. So we can reject null hypothesis "There is no difference in the mean rating of the models" and accept alternative hypothesis "There is at least

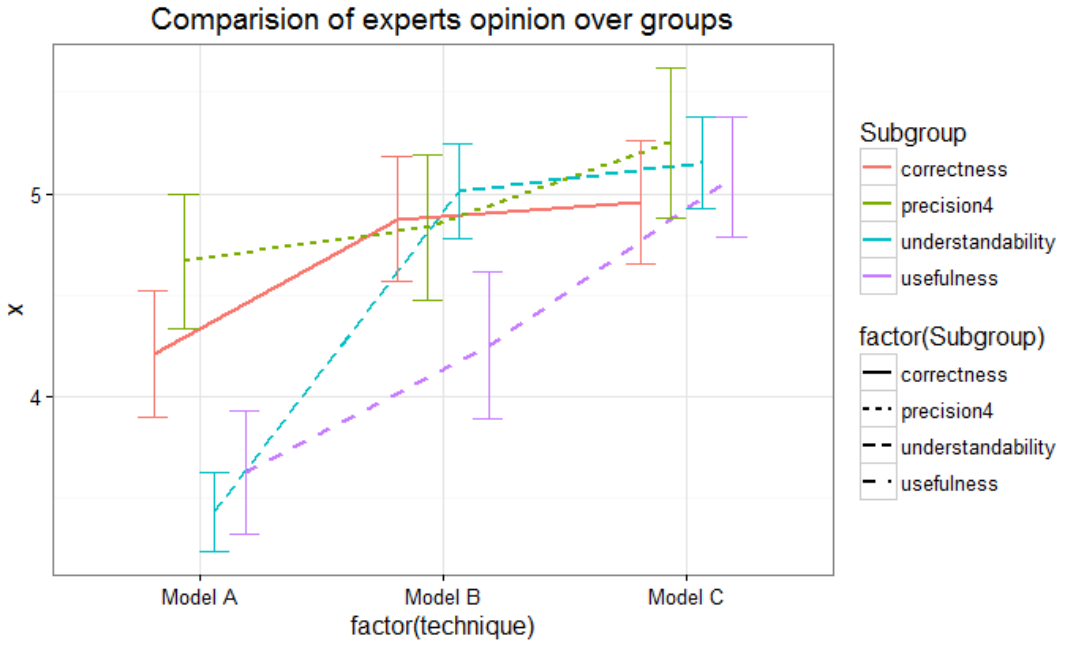

Figure 25: Experts opinion interaction

one model that is different from the others".

Our finding are valid, since the variability test Pr-value was 0.2751, which higher than 0.05, which indicates that variabilities are same over the groups. Also since the homogeneity of variance assumption plot Figure 26 and normality assumption QQ-plot(Figure 27) were acceptable. And also due the fact that Shapiro-Wilk normality test p-value was 0.0007942.

We also carried out workshops with domain experts. Following aspects raised from their workshops:

1. Models should have information about the frequency of paths taken (i.e. different colour, probability numbers at gateways, bolder paths, frequency numbers, or a heat map)

2. The models should have option to follow the flow of a process when a loop occurs. The order of tasks should shown in that case.

3. Models could be split into sub-models to increase readability and understandability. It was suggested as by frequency information, or as by in which category the tasks belongs to

4. Discarding infrequent paths could be dangerous

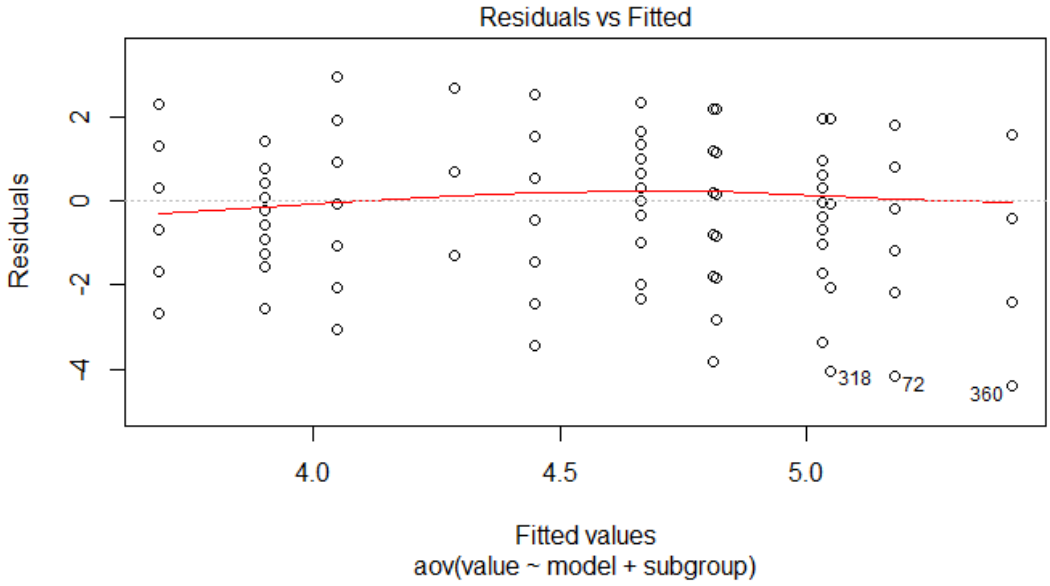

Figure 26: Homogeneity check

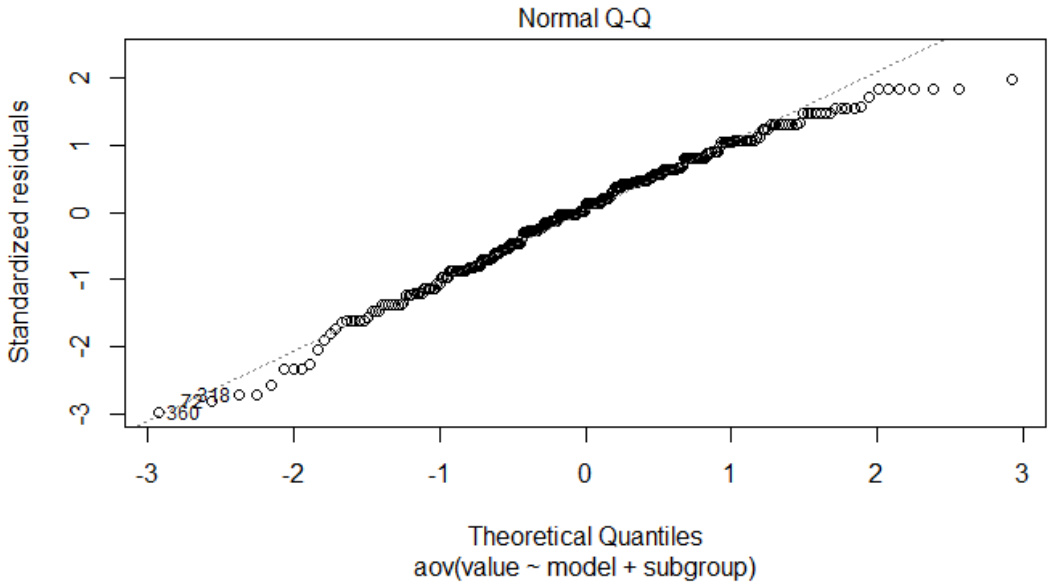

Figure 27: Normality check

5. There could be a scale or an axis that shows in which stage (for example support,enhancement, development) the process is in at model.

6. Models were good for high level analysis for managers and team leads, but for others they were too general. Non-managers found that the models wouldn't help to decide what to do in they everyday work.

7. Each model as a partial representation of the process.

8. The time perspective should also be present at the models, time taken by each task on average, maximally and at best case, and also the overall time spent for the process.

9. Models could be more simple - some of the gateways could be removed

10. Model C was found to be the best in the sense of reflecting company's everyday work. A was found to be too detailed, and B was found not to reflect the reality.

11. It was also pointed that C allows a pass without taking any path.

From the students quick verbal feedback we learned that the splitting gateways should have a textual description attached. That could make models more understandable than without the information. From textual feedback we extracted following information:

1. Students found that model A is the most detailed, but also hardest to follow. The same finding was also in the experts case.

2. Experts pointed out that models do represent possible situations, but not all of them.

In addition to the subjective opinion of the two groups, we carried out an objective evaluation of the models with using Calculate BPMN Metrics plug-in at ProM 6.5.1 from BPMN Miner package.

Table 6 shows the overall results of the evaluation. We can see that for the experts the best model was B, for the students model C and in metrics wise, the best model were B and C. In all the cases the model A was the worst one. For overall scores (metric + students + experts) we cannot differentiate between model B and C. They are both equally usable. This conforms to our findings from that, that there is no statistically significant difference between models B and C.

The size means number of elements in model (the smaller, the better), CFC stands for control-flow complexity (the smaller, the better).

Table 6: Rating table

	Model A	**Model B**	**Model C**	Rank A	Rank B	Rank C
Size	67	36	31	3	2	1
CFC	71	30	31	3	1	2
Metrics ranking	3	1-2	1-2			
Students ranking	3	2	1			
Experts ranking	3	1	2			
Overall ranking	3	1-2	1-2			

Model A stands for Evolutionary Tree Miner, Model B for Structured Miner, and model C for Inductive Miner - all operations.

6 Related work

A previous survey and empirical evaluation of automated process discovery methods was done by De Weerdt et al. [WBVB12]. This survey covered 27 approaches altogether, all of which are included in the studies identified during our systematic literature review prior to filtering. That empirical evaluation by De Weerdt et al. in [WBVB12] includes seven approaches, namely AGNEsMiner,α\$+, α\$++, Genetic Miner (and a variant thereof),Flower Heuristics Miner and ILP Miner. In comparison, our evaluation includes three, Structured Miner, Evolutionary Tree Miner, and Inductive Miner - all operations. Another difference with respect to [WBVB12] is that in this thesis the evaluation was done with domain experts and based on their opinion conclusions were made. So we targeted the usability of the automated business process discovery methods for the industry, whereas the evaluation in [WBVB12] was rather a functional testing.

Another previous survey in the field is outdated [vdAvDH+03], and a more recent one is not intended to be comprehensive [CP12], but rather focuses on plug-ins available in the ProM.

Another related study done by Augusto et al. [ACD+17] is similar to this thesis, because the review part of this thesis was used there. In that paper instead of a user-evaluation the authors propose a benchmark analysis.

7 Conclusion

This thesis has presented a Systematic Literature Review (SLR) of automated process discovery methods and a comparative evaluation of existing implementations of these methods using an real-life event log from an international software engineering company, four quality metrics, and feedback from domain experts.

From the literature analysed, we can conclude that automated process discovery is a "hot" research topic, due the number of papers published. Automated process discovery can be divided into frequency based, genetic based, theory of regions based, probabilistic based approaches. With RQ1 we found 55 different approaches for automated process discovery. With RQ2 we identified that 9 declarative approaches, one hybrid approach, 43 procedural approaches, and 2 approaches that produce multiple different type models. With RQ3 we found that all approaches can deal with sequences, that three approaches can discover all constraints, one approach cannot discover AND construct, one approach cannot discover XOR, two approaches cannot discover loops, 9 discover Declare constraints, and one discovers WoMan formalism. With RQ4 we discovered that there is 5 platforms for process discovery plug-ins, and that 24 approaches have standalone implementations. With RQ5 we found that most of the approaches (78%) have been evaluated with real-life logs. With RQ6 we found that most of the approaches have been evaluated at administrative or financial domain.

From the statistical analysis we discovered that there exist a model that is statistically different from the others, the model A. Domain experts found Structured Miner (model B) to be the best one, closely followed by Inductive Miner - all operations (model C), but the difference is not statistically significant between these two.

Finally, domain experts found that the automated process discovery methods at current state are not acceptable for their goals. This opens up new directions for the research, like adding frequency information to the models, splitting the models, adding tracking scales, and adding time information. In domain experts opinion these make models more usable for them.

References

[ACD+16] Adriano Augusto, Raffaele Conforti, Marlon Dumas, Marcello La Rosa, and Giorgio Bruno. Automated discovery of structured process models: Discover structured vs. discover and structure. pages 313–329, 2016.

[ACD+17] Adriano Augusto, Raffaele Conforti, Marlon Dumas, Marcello La Rosa, Fabrizio Maria Maggi, Andrea Marrella, Massimo Mecella, and Allar Soo. Automated discovery of process models from event logs: Review and benchmark. *arXiv preprint arXiv:1705.02288*, 2017.

[ACN14] Banafshe Arbab-Zavar, John N. Carter, and Mark S. Nixon. On hierarchical modelling of motion for workflow analysis from overhead view. *Mach. Vis. Appl.*, 25(2):345–359, 2014.

[AK14] Mari Abe and Michiharu Kudo. Business monitoring framework for process discovery with real-life logs. In *Business Process Management - 12th International Conference, BPM 2014, Haifa, Israel, September 7-11, 2014. Proceedings*, pages 416–423, 2014.

[Bay11] İ Bayraktar. The business value of process mining bringing it all together. *Eindhoven University of Technology, Eindhoven*, 2011.

[BCFM16] Mario Luca Bernardi, Marta Cimitile, Chiara Di Francescomarino, and Fabrizio Maria Maggi. Do activity lifecycles affect the validity of a business rule in a business process? *Inf. Syst.*, 62:42–59, 2016.

[BDB+15] Gabriele Bleser, Dima Damen, Ardhendu Behera, Gustaf Hendeby, Katharina Mura, Markus Miezal, Andrew Gee, Nils Petersen, Gustavo Maçães, Hugo Domingues, et al. Cognitive learning, monitoring and assistance of industrial workflows using egocentric sensor networks. *PloS one*, 10(6):e0127769, 2015.

[BMDB16] Dominic Breuker, Martin Matzner, Patrick Delfmann, and Jörg Becker. Comprehensible predictive models for business processes. *MIS Quarterly*, 40(4):1009–1034, 2016.

[BRL10] Elena Bellodi, Fabrizio Riguzzi, and Evelina Lamma. Probabilistic declarative process mining. In *Knowledge Science, Engineering and Management, 4th International Conference, KSEM 2010, Belfast, Northern Ireland, UK, September 1-3, 2010. Proceedings*, pages 292–303, 2010.

[BvDvdA14] Joos C. A. M. Buijs, Boudewijn F. van Dongen, and Wil M. P. van der Aalst. Quality dimensions in process discovery: The importance of fitness, precision, generalization and simplicity. *Int. J. Cooperative Inf. Syst.*, 23(1), 2014.

[CC14] Josep Carmona and Jordi Cortadella. Process discovery algorithms using numerical abstract domains. *IEEE Trans. Knowl. Data Eng.*, 26(12):3064–3076, 2014.

[CCP14] Sofie De Cnudde, Jan Claes, and Geert Poels. Improving the quality of the heuristics miner in prom 6.2. *Expert Syst. Appl.*, 41(17):7678–7690, 2014.

[CDGR14] Raffaele Conforti, Marlon Dumas, Luciano García-Bañuelos, and Marcello La Rosa. Beyond tasks and gateways: Discovering BPMN models with subprocesses, boundary events and activity markers. In *Business Process Management - 12th International Conference, BPM 2014, Haifa, Israel, September 7-11, 2014. Proceedings*, pages 101–117, 2014.

[CDGR16] Raffaele Conforti, Marlon Dumas, Luciano García-Bañuelos, and Marcello La Rosa. BPMN miner: Automated discovery of BPMN process models with hierarchical structure. *Inf. Syst.*, 56:284–303, 2016.

[Cel17] Celonis. Celonis prouduct description, 2017.

[CLM+09] Federico Chesani, Evelina Lamma, Paola Mello, Marco Montali, Fabrizio Riguzzi, and Sergio Storari. Exploiting inductive logic programming techniques for declarative process mining. volume 2, pages 278–295. 2009.

[CLRS10] Massimiliano Cattafi, Evelina Lamma, Fabrizio Riguzzi, and Sergio Storari. Incremental declarative process mining. In *Smart Information and Knowledge Management: Advances, Challenges, and Critical Issues*, pages 103–127. 2010.

[CP12] Jan Claes and Geert Poels. Process mining and the prom framework: An exploratory survey. In *Business Process Management Workshops - BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised Papers*, pages 187–198, 2012.

[dLStHvdA16] Massimiliano de Leoni, Suriadi Suriadi, Arthur H. M. ter Hofstede, and Wil M. P. van der Aalst. Turning event logs into process

movies: animating what has really happened. *Software and System Modeling*, 15(3):707–732, 2016.

[dLvdA13] Massimiliano de Leoni and Wil M. P. van der Aalst. Data-aware process mining: discovering decisions in processes using alignments. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13, Coimbra, Portugal, March 18-22, 2013*, pages 1454–1461, 2013.

[DM13a] Claudio Di Ciccio and Massimo Mecella. Mining artful processes from knowledge workers' emails. *IEEE Internet Computing*, 17(5):10–20, 2013.

[DM13b] Claudio Di Ciccio and Massimo Mecella. A two-step fast algorithm for the automated discovery of declarative workflows. In *IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2013, Singapore, 16-19 April, 2013*, pages 135–142, 2013.

[dMB08] Leonardo Mendonça de Moura and Nikolaj Bjørner. Z3: an efficient SMT solver. In *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, pages 337–340, 2008.

[DMM16] Claudio Di Ciccio, Fabrizio Maria Maggi, and Jan Mendling. Efficient discovery of target-branched declare constraints. *Inf. Syst.*, 56:258–283, 2016.

[DMMM15] Claudio Di Ciccio, Fabrizio Maria Maggi, Marco Montali, and Jan Mendling. Ensuring model consistency in declarative process discovery. In *Business Process Management - 13th International Conference, BPM 2015, Innsbruck, Austria, August 31 - September 3, 2015, Proceedings*, pages 144–159, 2015.

[Eve16] Joerg Evermann. Scalable process discovery using map-reduce. *IEEE Trans. Services Computing*, 9(3):469–481, 2016.

[Fer14] Stefano Ferilli. Woman: Logic-based workflow learning and management. *IEEE Trans. Systems, Man, and Cybernetics: Systems*, 44(6):744–756, 2014.

[FGG+14] Philippe Fournier-Viger, Antonio Gomariz, Ted Gueniche, Azadeh Soltani, Cheng-Wei Wu, and Vincent S. Tseng. SPMF: a java open-source pattern mining library. *Journal of Machine Learning Research*, 15(1):3389–3393, 2014.

[FGP15] Francesco Folino, Massimo Guarascio, and Luigi Pontieri. On the discovery of explainable and accurate behavioral models for complex lowly-structured business processes. pages 206–217, 2015.

[GG14] Mykola Galushka and Wasif Gilani. Drugfusion - retrieval knowledge management for prediction of adverse drug events. In *Business Information Systems - 17th International Conference, BIS 2014, Larnaca, Cyprus, May 22-23, 2014. Proceedings*, pages 13–24, 2014.

[GGLP15] Gianluigi Greco, Antonella Guzzo, Francesco Lupia, and Luigi Pontieri. Process discovery under precedence constraints. *TKDD*, 9(4):32:1–32:39, 2015.

[GvdA07] Christian W. Günther and Wil M. P. van der Aalst. Fuzzy mining - adaptive process simplification based on multi-perspective metrics. In *Business Process Management, 5th International Conference, BPM 2007, Brisbane, Australia, September 24-28, 2007, Proceedings*, pages 328–343, 2007.

[GWW+15] Qinlong Guo, Lijie Wen, Jianmin Wang, Zhiqiang Yan, and Philip S. Yu. Mining invisible tasks in non-free-choice constructs. In *Business Process Management - 13th International Conference, BPM 2015, Innsbruck, Austria, August 31 - September 3, 2015, Proceedings*, pages 109–125, 2015.

[HK12] Zan Huang and Akhil Kumar. A study of quality and accuracy trade-offs in process mining. *INFORMS Journal on Computing*, 24(2):311–327, 2012.

[Kit04] Barbara Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26, 2004.

[KMDF16] Taavi Kala, Fabrizio Maria Maggi, Claudio Di Ciccio, and Chiara Di Francescomarino. Apriori and sequence analysis for discovering declarative process models. In *20th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2016, Vienna, Austria, September 5-9, 2016*, pages 1–9, 2016.

[LBvdA10] Jiafei Li, R. P. Jagadeesh Chandra Bose, and Wil M. P. van der Aalst. Mining context-dependent and interactive business process maps using execution patterns. In *Business Process Management Workshops - BPM 2010 International Workshops and Education Track, Hoboken, NJ, USA, September 13-15, 2010, Revised Selected Papers*, pages 109–121, 2010.

[LFvdA13] Sander J. J. Leemans, Dirk Fahland, and Wil M. P. van der Aalst. Discovering block-structured process models from event logs containing infrequent behaviour. In *Business Process Management Workshops - BPM 2013 International Workshops, Beijing, China, August 26, 2013, Revised Papers*, pages 66–78, 2013.

[LFvdA14a] Sander J. J. Leemans, Dirk Fahland, and Wil M. P. van der Aalst. Discovering block-structured process models from incomplete event logs. In *Application and Theory of Petri Nets and Concurrency - 35th International Conference, PETRI NETS 2014, Tunis, Tunisia, June 23-27, 2014. Proceedings*, pages 91–110, 2014.

[LFvdA14b] Sander J. J. Leemans, Dirk Fahland, and Wil M. P. van der Aalst. Exploring processes and deviations. In *Business Process Management Workshops - BPM 2014 International Workshops, Eindhoven, The Netherlands, September 7-8, 2014, Revised Papers*, pages 304–316, 2014.

[LFvdA15a] Sander J. J. Leemans, Dirk Fahland, and Wil M. P. van der Aalst. Scalable process discovery with guarantees. In *Enterprise, Business-Process and Information Systems Modeling - 16th International Conference, BPMDS 2015, 20th International Conference, EMMSAD 2015, Held at CAiSE 2015, Stockholm, Sweden, June 8-9, 2015, Proceedings*, pages 85–101, 2015.

[LFvdA15b] Sander J. J. Leemans, Dirk Fahland, and Wil M. P. van der Aalst. Using life cycle information in process discovery. In *Business Process Management Workshops - BPM 2015, 13th International Workshops, Innsbruck, Austria, August 31 - September 3, 2015, Revised Papers*, pages 204–217, 2015.

[LFvdA16] Sander J. J. Leemans, Dirk Fahland, and Wil M. P. van der Aalst. Scalable process discovery and conformance checking. *Software & Systems Modeling*, pages 1–33, 2016.

[LHEZ12] Robert Lorenz, Markus Huber, Christoph Etzel, and Dan Zecha. SYNOPS - generation of partial languages and synthesis of petri nets. In *Proceedings of the International Workshop on Petri Nets and Software Engineering, Hamburg, Germany, June 25-26, 2012*, pages 237–252, 2012.

[LHFL15] Jan Ladiges, Christopher Haubeck, Alexander Fay, and Winfried Lamersdorf. Learning behaviour models of discrete event production systems from observing input/output signals. *IFAC-PapersOnLine*, 48(3):1565–1572, 2015.

[LvdA14] Maikel Leemans and Wil M. P. van der Aalst. Discovery of frequent episodes in event logs. In *Data-Driven Process Discovery and Analysis - 4th International Symposium, SIMPDA 2014, Milan, Italy, November 19-21, 2014, Revised Selected Papers*, pages 1–31, 2014.

[LYC15] Veronica Liesaputra, Sira Yongchareon, and Sivadon Chaisiri. Efficient process model discovery using maximal pattern mining. In *Business Process Management - 13th International Conference, BPM 2015, Innsbruck, Austria, August 31 - September 3, 2015, Proceedings*, pages 441–456, 2015.

[MBvdA13] Fabrizio Maria Maggi, R. P. Jagadeesh Chandra Bose, and Wil M. P. van der Aalst. A knowledge-based integrated approach for discovering and repairing declare maps. In *Advanced Information Systems Engineering - 25th International Conference, CAiSE 2013, Valencia, Spain, June 17-21, 2013. Proceedings*, pages 433–448, 2013.

[MCB16] Andrey Mokhov, Josep Carmona, and Jonathan Beaumont. Mining conditional partial order graphs from event logs. volume 11, pages 114–136. 2016.

[MDGM13] Fabrizio Maria Maggi, Marlon Dumas, Luciano García-Bañuelos, and Marco Montali. Discovering data-aware declarative process models from event logs. In *Business Process Management - 11th International Conference, BPM 2013, Beijing, China, August 26-30, 2013. Proceedings*, pages 81–96. 2013.

[MMDM16] Fabrizio Maria Maggi, Marco Montali, Claudio Di Ciccio, and Jan Mendling. Semantical vacuity detection in declarative process mining. In *Business Process Management - 14th International Conference, BPM 2016, Rio de Janeiro, Brazil, September 18-22, 2016. Proceedings*, pages 158–175, 2016.

[MRD+15] Thomas Molka, David Redlich, Marc Drobek, Xiao-Jun Zeng, and Wasif Gilani. Diversity guided evolutionary mining of hierarchical process models. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2015, Madrid, Spain, July 11-15, 2015*, pages 1247–1254, 2015.

[MSR14] Fabrizio Maria Maggi, Tijs Slaats, and Hajo A. Reijers. The automated discovery of hybrid processes. In *Business Process Management - 12th International Conference, BPM 2014, Haifa, Israel, September 7-11, 2014. Proceedings*, pages 392–399, 2014.

[NSCB11] Hamid R. Motahari Nezhad, Régis Saint-Paul, Fabio Casati, and Boualem Benatallah. Event correlation for process discovery from web service interaction logs. *VLDB J.*, 20(3):417–444, 2011.

[PCvB15] Hernán Ponce de León, Josep Carmona, and Seppe K. L. M. vanden Broucke. Incorporating negative information in process discovery. In *Business Process Management - 13th International Conference, BPM 2015, Innsbruck, Austria, August 31 - September 3, 2015, Proceedings*, pages 126–143, 2015.

[PFD15] Viara Popova, Dirk Fahland, and Marlon Dumas. Artifact lifecycle discovery. *Int. J. Cooperative Inf. Syst.*, 24(1), 2015.

[PS14] Gustavo Pizarro and Marcos Sepúlveda. Experimenting with an OLAP approach for interactive discovery in process mining. In *Business Process Management Workshops - BPM 2014 International Workshops, Eindhoven, The Netherlands, September 7-8, 2014, Revised Papers*, pages 317–329, 2014.

[RMG+14] David Redlich, Thomas Molka, Wasif Gilani, Gordon S. Blair, and Awais Rashid. Dynamic constructs competition miner - occurrence- vs. time-based ageing. In *Data-Driven Process Discovery and Analysis - 4th International Symposium, SIMPDA 2014, Milan, Italy, November 19-21, 2014, Revised Selected Papers*, pages 79–106, 2014.

[RW11] J. T. S. Ribeiro and A. J. M. M. Weijters. Event cube: Another perspective on business processes. In *On the Move to Meaningful Internet Systems: OTM 2011 - Confederated International Conferences: CoopIS, DOA-SVI, and ODBASE 2011, Hersonissos, Crete, Greece, October 17-21, 2011, Proceedings, Part I*, pages 274–283, 2011.

[SBVA15] Ashwin Srinivasan, Michael Bain, Deepika Vatsa, and Sumeet Agarwal. Identification of transition models of biological systems in the presence of transition noise. In *Inductive Logic Programming - 25th International Conference, ILP 2015, Kyoto, Japan, August 20-22, 2015, Revised Selected Papers*, pages 200–214, 2015.

[SJYM16] Wei Song, Hans-Arno Jacobsen, Chunyang Ye, and Xiaoxing Ma. Process discovery from dependence-complete event logs. *IEEE Trans. Services Computing*, 9(5):714–727, 2016.

[SRC+16] Stefan Schönig, Andreas Rogge-Solti, Cristina Cabanillas, Stefan Jablonski, and Jan Mendling. Efficient and customisable declarative process mining with SQL. In *Advanced Information Systems Engineering - 28th International Conference, CAiSE 2016, Ljubljana, Slovenia, June 13-17, 2016. Proceedings*, pages 290–305, 2016.

[TSHvdA16] Niek Tax, Natalia Sidorova, Reinder Haakma, and Wil M. P. van der Aalst. Mining local process models. *J. Innovation in Digital Ecosystems*, 3(2):183–196, 2016.

[vdA13a] Wil M. P. van der Aalst. Process cubes: Slicing, dicing, rolling up and drilling down event data for process mining. In *Asia Pacific Business Process Management - First Asia Pacific Conference, AP-BPM 2013, Beijing, China, August 29-30, 2013. Selected Papers*, pages 1–22, 2013.

[VdA13b] Wil MP Van der Aalst. Decomposing petri nets for process mining: A generic approach. *Distributed and Parallel Databases*, 31(4):471–507, 2013.

[vdAAdM+11] Wil M. P. van der Aalst, Arya Adriansyah, Ana Karla Alves de Medeiros, Franco Arcieri, Thomas Baier, Tobias Blickle, R. P. Jagadeesh Chandra Bose, Peter van den Brand, Ronald Brandtjen, Joos C. A. M. Buijs, Andrea Burattin, Josep Carmona, Malú Castellanos, Jan Claes, Jonathan Cook, Nicola Costantini, Francisco Curbera, Ernesto Damiani, Massimiliano de Leoni, Pavlos Delias, Boudewijn F. van Dongen, Marlon Dumas, Schahram Dustdar, Dirk Fahland, Diogo R. Ferreira, Walid Gaaloul, Frank van Geffen, Sukriti Goel, Christian W. Günther, Antonella Guzzo, Paul Harmon, Arthur H. M. ter Hofstede, John Hoogland, Jon Espen Ingvaldsen, Koki Kato, Rudolf Kuhn, Akhil Kumar, Marcello La Rosa, Fabrizio Maria Maggi, Donato Malerba, R. S. Mans,

Alberto Manuel, Martin McCreesh, Paola Mello, Jan Mendling, Marco Montali, Hamid R. Motahari Nezhad, Michael zur Muehlen, Jorge Munoz-Gama, Luigi Pontieri, Joel Ribeiro, Anne Rozinat, Hugo Seguel Pérez, Ricardo Seguel Pérez, Marcos Sepúlveda, Jim Sinur, Pnina Soffer, Minseok Song, Alessandro Sperduti, Giovanni Stilo, Casper Stoel, Keith D. Swenson, Maurizio Talamo, Wei Tan, Chris Turner, Jan Vanthienen, George Varvaressos, Eric Verbeek, Marc Verdonk, Roberto Vigo, Jianmin Wang, Barbara Weber, Matthias Weidlich, Ton Weijters, Lijie Wen, Michael Westergaard, and Moe Thandar Wynn. Process mining manifesto. In *Business Process Management Workshops - BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I*, pages 169–194, 2011.

[vdAdMW05] Wil M. P. van der Aalst, Ana Karla A. de Medeiros, and A. J. M. M. Weijters. Genetic process mining. In *Applications and Theory of Petri Nets 2005, 26th International Conference, ICATPN 2005, Miami, USA, June 20-25, 2005, Proceedings*, pages 48–69, 2005.

[vdAKRV15] Wil M. P. van der Aalst, Anna Kalenkova, Vladimir Rubin, and Eric Verbeek. Process discovery using localized events. In *Application and Theory of Petri Nets and Concurrency - 36th International Conference, PETRI NETS 2015, Brussels, Belgium, June 21-26, 2015, Proceedings*, pages 287–308, 2015.

[vdAvDH+03] Wil M. P. van der Aalst, Boudewijn F. van Dongen, Joachim Herbst, Laura Maruster, Guido Schimm, and A. J. M. M. Weijters. Workflow mining: A survey of issues and approaches. *Data Knowl. Eng.*, 47(2):237–267, 2003.

[vdAWM04] Wil M. P. van der Aalst, Ton Weijters, and Laura Maruster. Workflow mining: Discovering process models from event logs. *IEEE Trans. Knowl. Data Eng.*, 16(9):1128–1142, 2004.

[vESvdA16] Maikel L. van Eck, Natalia Sidorova, and Wil M. P. van der Aalst. Discovering and exploring state-based models for multi-perspective processes. In *Business Process Management - 14th International Conference, BPM 2016, Rio de Janeiro, Brazil, September 18-22, 2016. Proceedings*, pages 142–157, 2016.

[VML15] Borja Vázquez-Barreiros, Manuel Mucientes, and Manuel Lama. Prodigen: Mining complete, precise and minimal structure process models with a genetic algorithm. *Inf. Sci.*, 294:315–333, 2015.

[VSL14] Olegas Vasilecas, Titas Savickas, and Evaldas Lebedys. Directed acyclic graph extraction from event logs. In *Information and Software Technologies - 20th International Conference, ICIST 2014, Druskininkai, Lithuania, October 9-10, 2014. Proceedings*, pages 172–181, 2014.

[VvdA12] H. M. W. (Eric) Verbeek and Wil M. P. van der Aalst. An experimental evaluation of passage-based process discovery. In *Business Process Management Workshops - BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised Papers*, pages 205–210, 2012.

[VvdA14] H. M. W. Verbeek and Wil M. P. van der Aalst. Decomposed process mining: The ILP case. In *Business Process Management Workshops - BPM 2014 International Workshops, Eindhoven, The Netherlands, September 7-8, 2014, Revised Papers*, pages 264–276, 2014.

[vZvDvdA15] Sebastiaan J. van Zelst, Boudewijn F. van Dongen, and Wil M. P. van der Aalst. Ilp-based process discovery using hybrid regions. pages 47–61, 2015.

[WBVB12] Jochen De Weerdt, Manu De Backer, Jan Vanthienen, and Bart Baesens. A multi-dimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs. *Inf. Syst.*, 37(7):654–676, 2012.

[WvDADM06] AJMM Weijters, Wil MP van Der Aalst, and AK Alves De Medeiros. Process mining with the heuristics miner-algorithm. *Technische Universiteit Eindhoven, Tech. Rep. WP*, 166:1–34, 2006.

[YSB+16] Bernardo Nugroho Yahya, Minseok Song, Hyerim Bae, Sung-ook Sul, and Jei-Zheng Wu. Domain-driven actionable process model discovery. *Computers & Industrial Engineering*, 99:382–400, 2016.

A Domain expert questions

Following questions were used for **domain experts**:

- **General Questions**
  1. How many process models have you analysed or read within the last 12 months?
  2. How many process models have you created or edited within the last 12 months?
  3. How many activities did all these models have on average?
  4. Overall, I am very familiar with BPMN
- **Questions about model quality**
  1. Rate how easy it is for you to understand the process models (1 means very difficult, 7 means very easy).
  2. Take one path and follow it from the beginning to the end. Rate how easy it is for you to follow your chosen path (1 means very difficult, 7 means very easy).
  3. Rate how easy it is for you to distinguish the paths in models (1 means very difficult, 7 means very easy).
  4. Can you recognise any processes you work with in the models? (1 means not at all, 7 means yes, clearly, everything is there)
  5. In your estimation, rate how well the models describe your processes (1 means that the model is too specific so to exclude some paths that are possible in reality, 7 means that the model is too general so to allow process paths that are not possible in reality).
  6. If you were to improve your business processes, which model would you find most useful for this purpose? (1 means useless, 7 means very useful)

All the general questions were multiple choice where participant could only mark one variant. For the general questions 1 and 2 the answer variants were none, 1 to 5, 6 to 15, and more than 15. For general question 3 the answer variants were I have not worked with process models during last year, 2 to 10, 11 to 20, and more than 20. For general question 4 the answer variants were Strongly agree, agree, somewhat agree, neutral, somewhat disagree, disagree, and strongly disagree. The purpose of the general questions was to collect background knowledge about the participants.

B Students questions

Following questions were used for **non-experts**:

- **General questions**

  1. How many process models have you read or analysed within the past 12 months?
  2. How many process models have you created or edited within the past 12 months?
  3. On average, how many activities did each process model have?

- **Questions about model quality**

  1. Rate how easy it is for you to understand the process models (1 means very difficult, 7 means very easy)
  2. Take one path and follow it from the beginning to the end. Rate how easy it is for you to follow your chosen path (1 means very difficult, 7 means very easy)
  3. Rate how easy it is for you to distinguish the different paths in the process models (1 means very difficult, 7 means very easy)
  4. Rate how easy it is for you to determine what type of process the models are representing (for instance insurance, health care, manufacturing etc.) (1 means very difficult, 7 means very easy)
  5. Rate how informative for you the process models are (1 means uninformative, 7 means very informative)
  6. Rate how general the models are (how many process behaviours they allow) (1 very specific, 7 means very general)
  7. If you had to improve this business process, rate how useful would be the process models for you (1 means useless, 7 means very useful)

All the general questions were multiple choice where participant could only mark one variant. For the general questions 1 and 2 the answer variants were none, 1 to 5, 6 to 15, and more than 15. For general question 3 the answer variants were I have not worked with process models during last year, 2 to 10, 11 to 20, and more than 20.

C Workshop questions

Following questions were used for moderating the workshop.

- Which of the models were the best? Why?
- How did the models look like in general?
- What could be developed?
- Did the models filled your expatiations?
- Would you consider using these algorithms in your company? And process discovery?
- What lacks are present in the models?

D Used models

Models used in the evaluation.

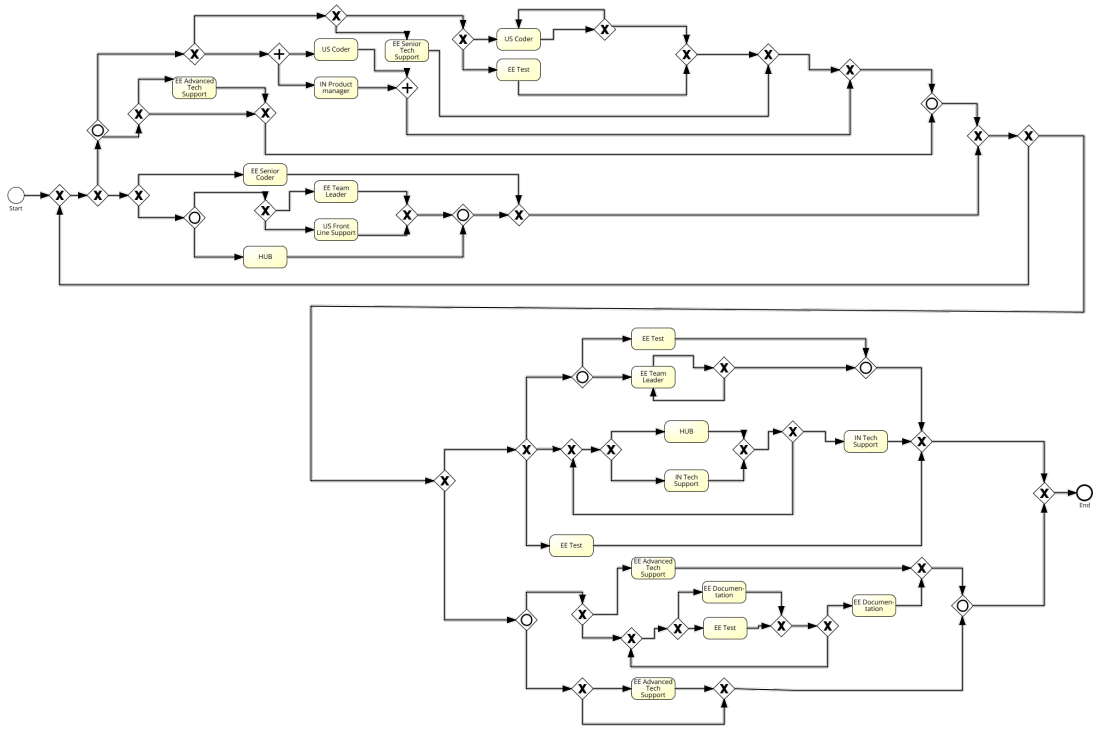

Figure 28: Model A

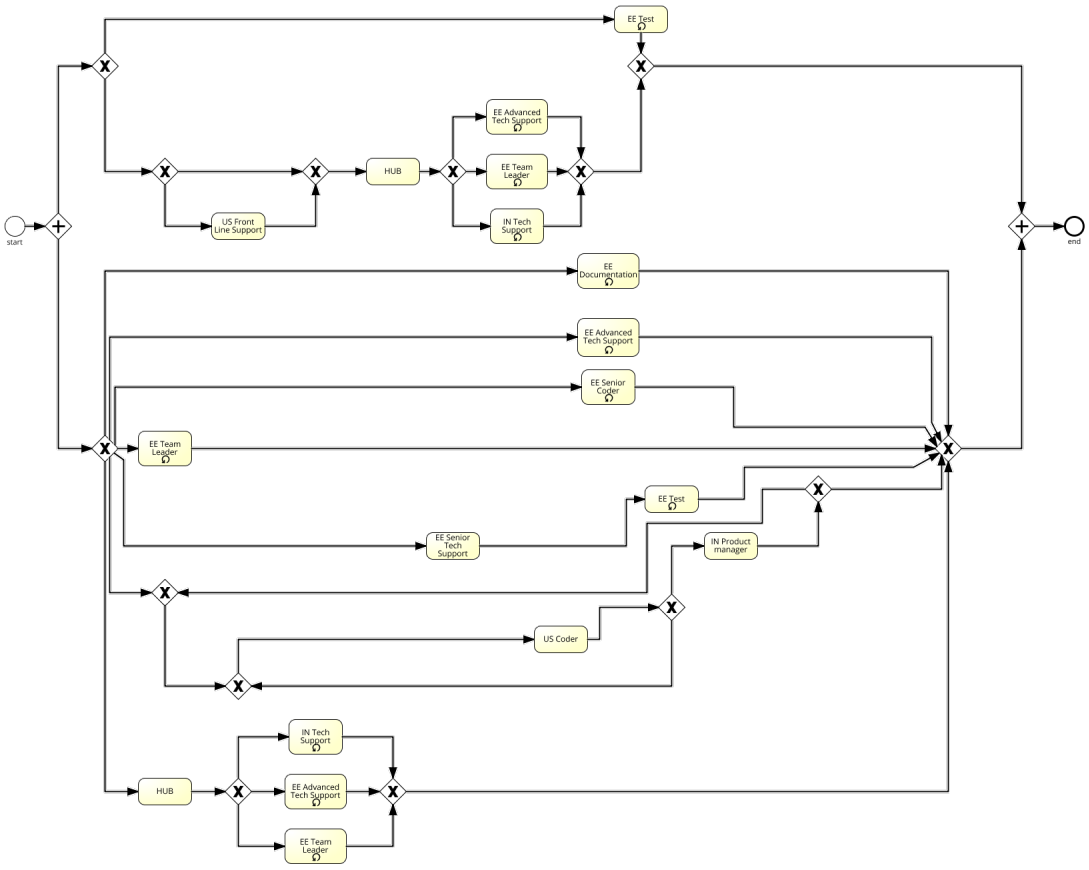

Figure 29: Model B

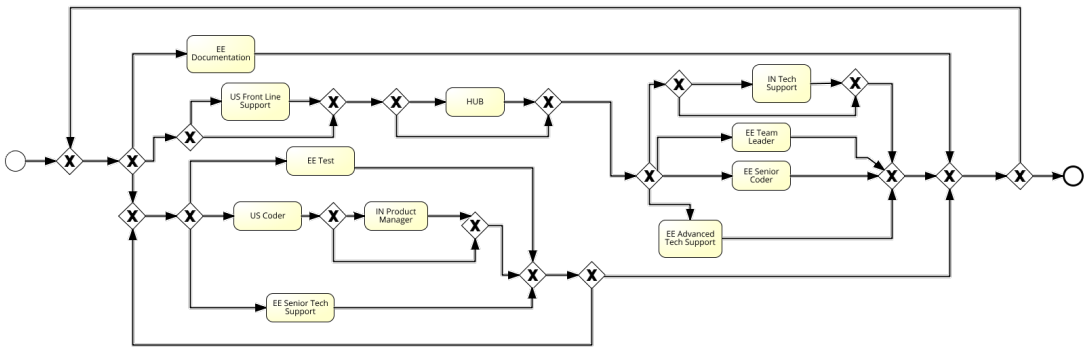

Figure 30: Model C

E Models generated

Models generated during testing.

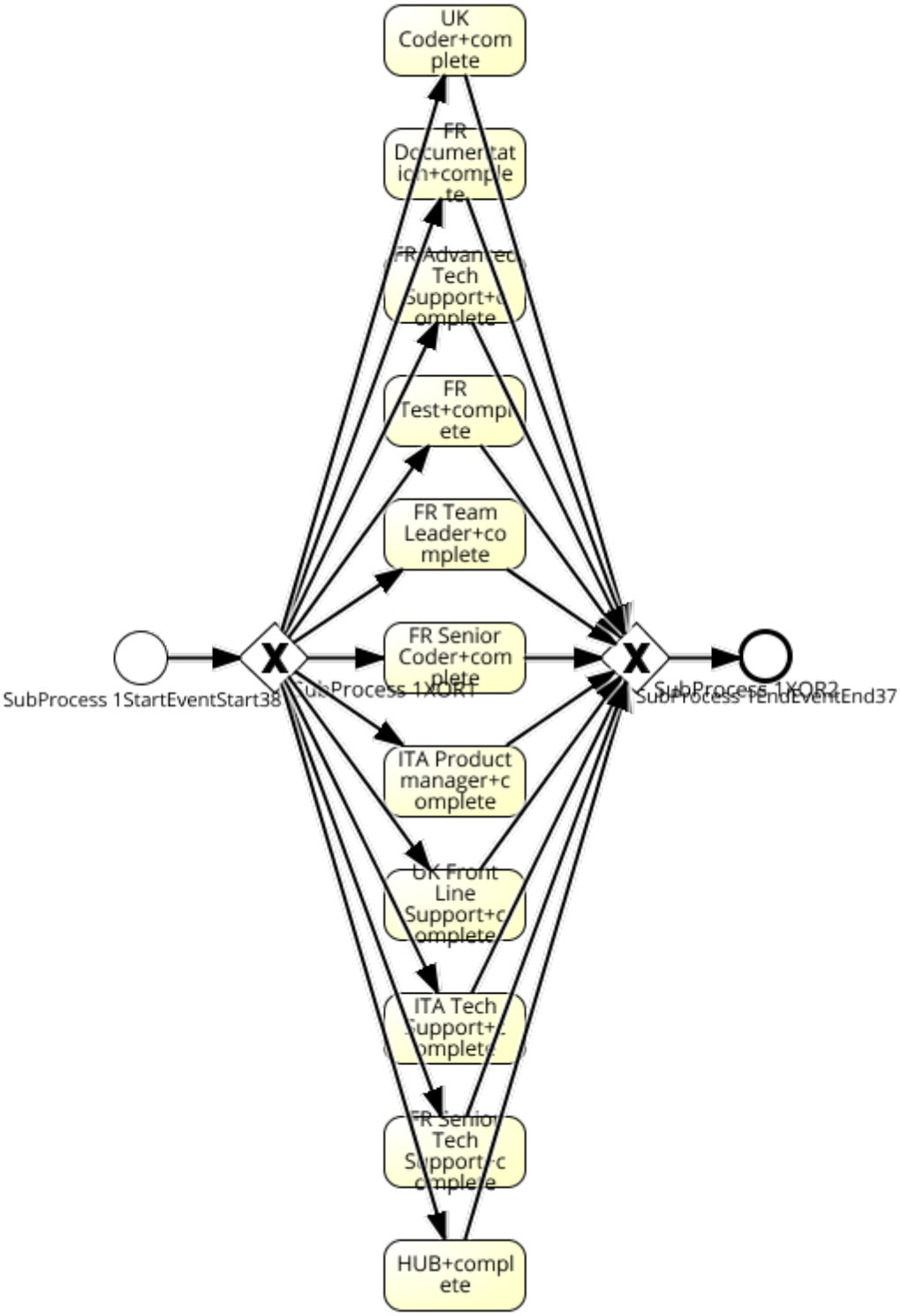

Figure 31: BPMN Miner

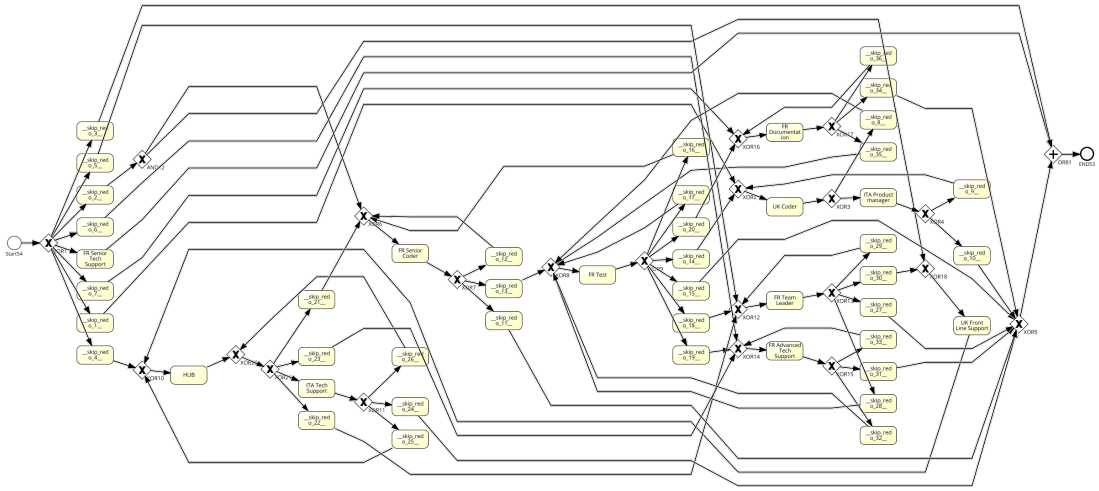

Figure 32: Alpha dollar

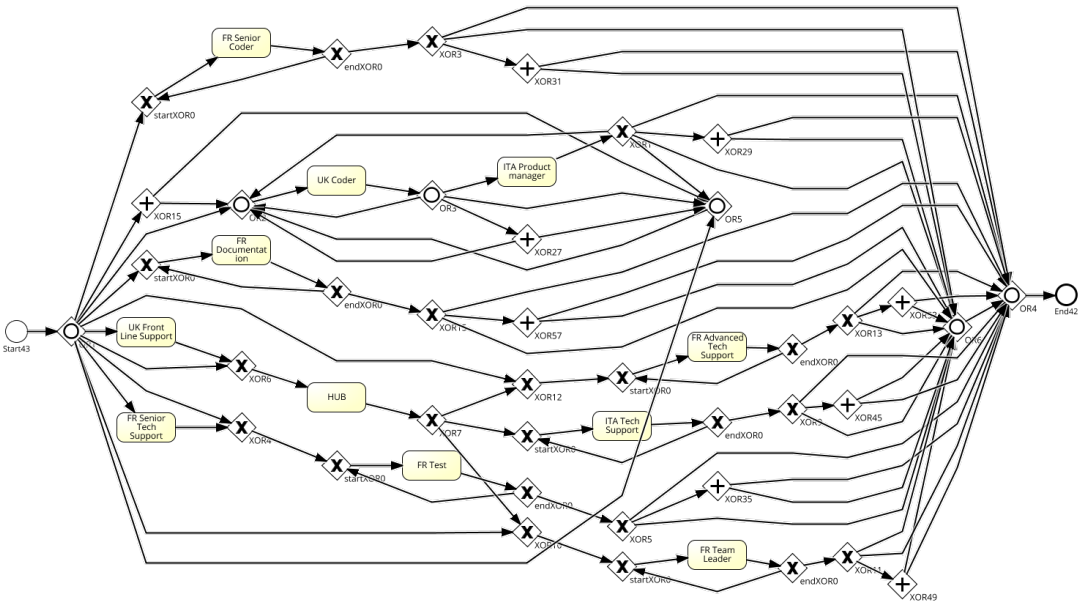

Figure 33: HM5

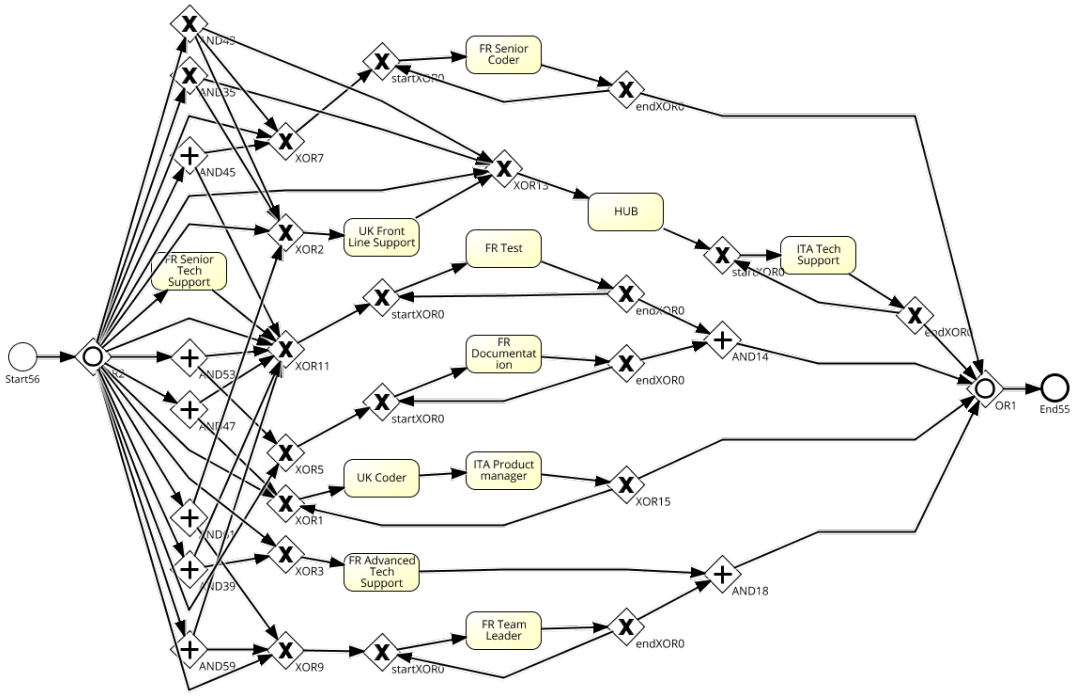

Figure 34: HM 6

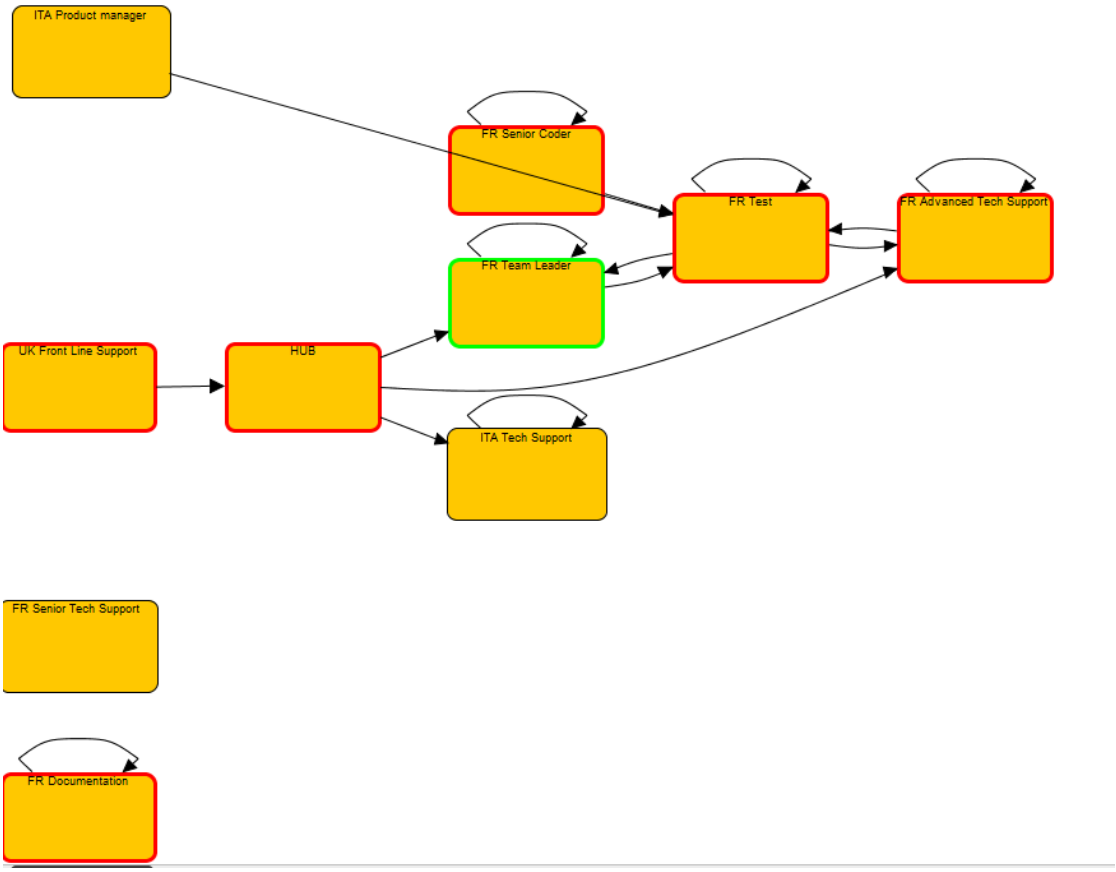

Figure 35: CNM

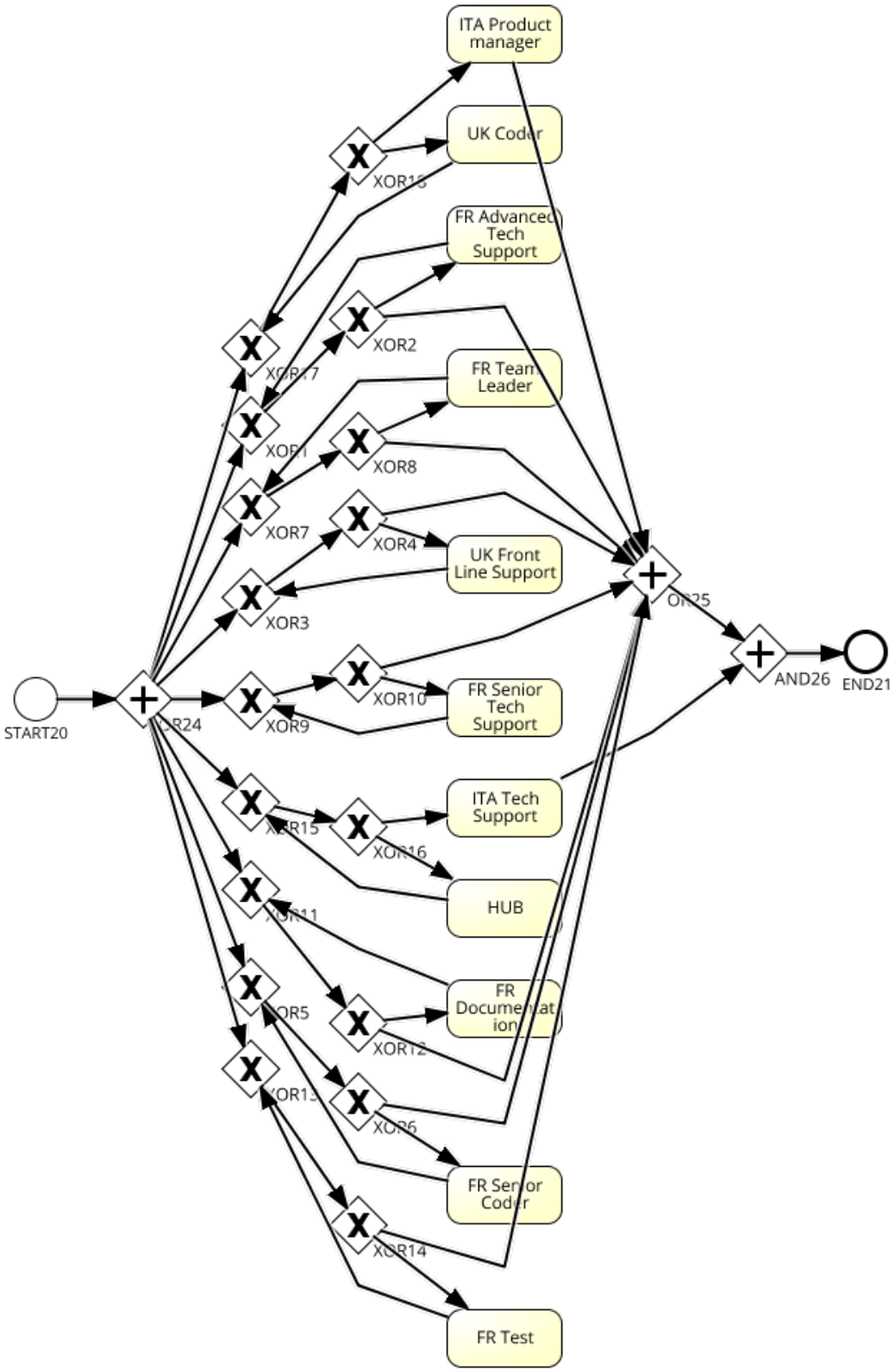

Figure 36: HILP

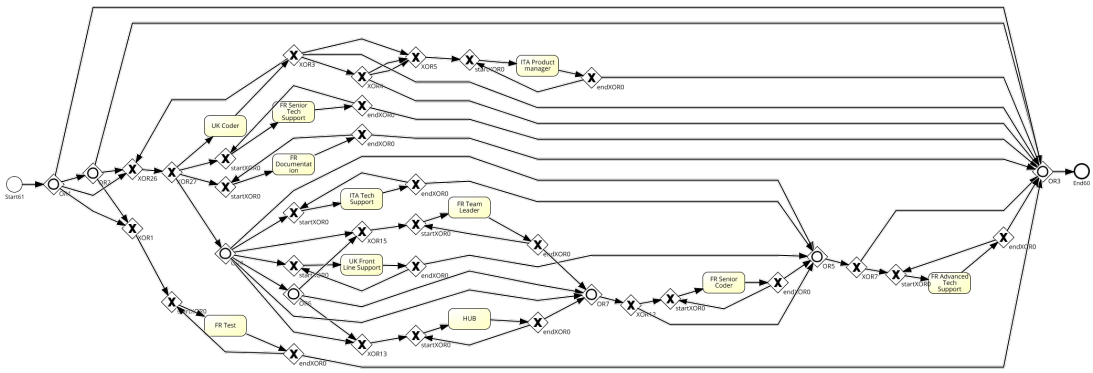

Figure 37: IMc

F General questions discussion

All the students have read or analysed, and created or edited at least one process model. It is presented in Figures 39 and 41. In domain experts case, 50% of participants (12 persons) have not read or analysed any process models within last year as shown in Figure 40. Also, in the experts case, 62.5% (15 persons) have not created or edited any process models within last year as shown in Figure 42. Moreover, 50% of domain experts have not worked with process models within last year, as shown in Figure 44. In the students case, 71.4% (15 persons) have worked with models having 2 to 10 activities, and 28.6% (6 persons) have worked with models having 11 to 20 activities (See Figure 43). In experts case, as mentioned before, 50% have not worked with models within last year, 41.7% (10 persons) have worked with models having 2 to 10 activities, and 8.3% (2 persons) have worked with models having more than 20 activities. In the students case, we knew that they were familiar with the BPMN due their curriculum. In experts case, 16.7 (4 persons) strongly disagree that they are very familiar with BPMN, 25% (6 persons) find that they disagree that they are very familiar with BPMN, 12.5% (3 persons) somewhat disagree that they are very familiar with BPMN, 20.8% (5 persons) position themselves as neutral, 20.8% (5 persons) somewhat agree that they are very familiar with BPMN, and 4.2% (1 persons) agrees that with question as shown in Figure 45. None of the domain experts strongly agreed with the questions.

Students found model A to be rather specific, model B to be neutral, and model C to be rather a bit general. It is presented in the Figure 38.

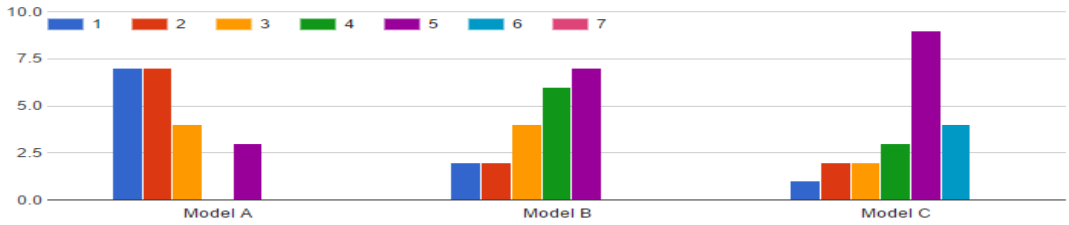

Figure 38: Students Q6 opinion

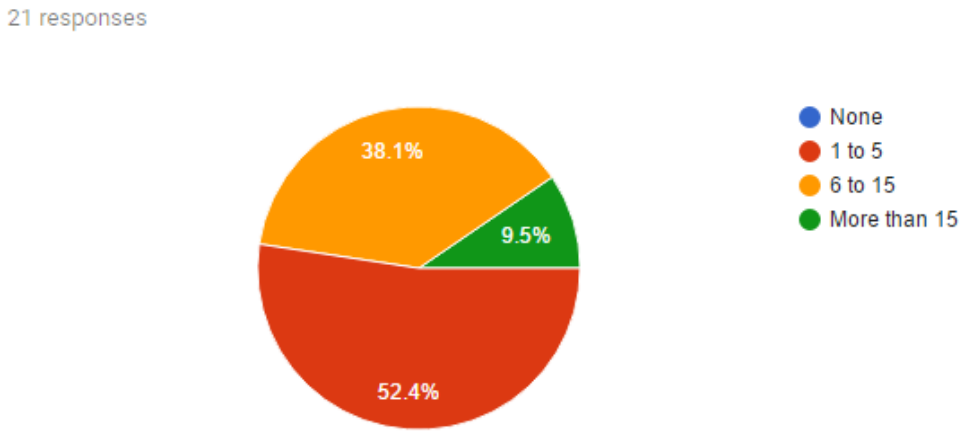

Figure 39: Models read or analysed within the past 12 months by students

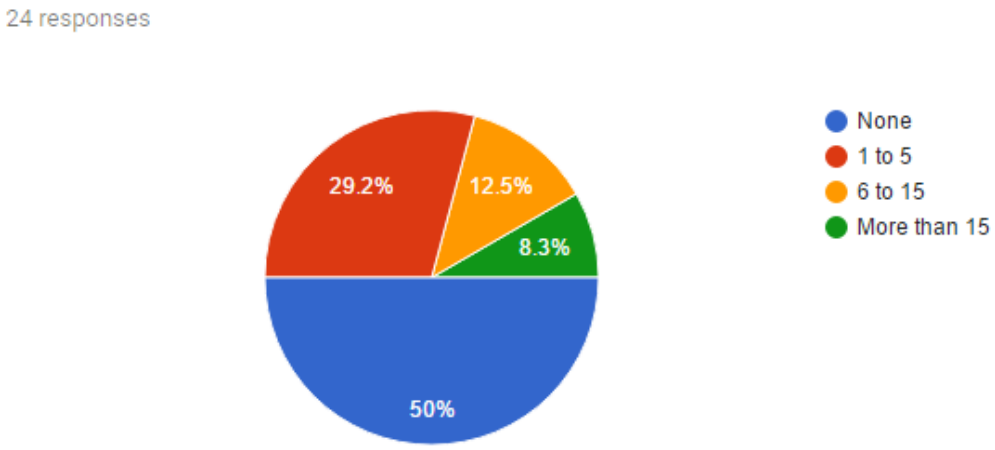

Figure 40: Models read or analysed within the past 12 months by experts

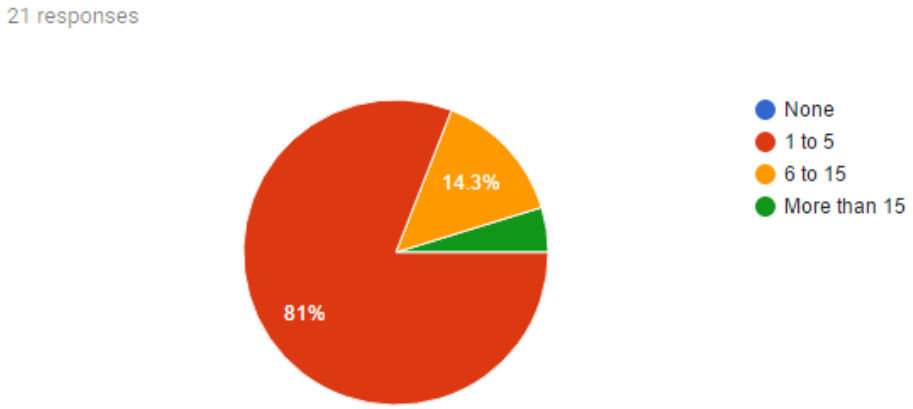

Figure 41: Models created or edited within the past 12 months by students

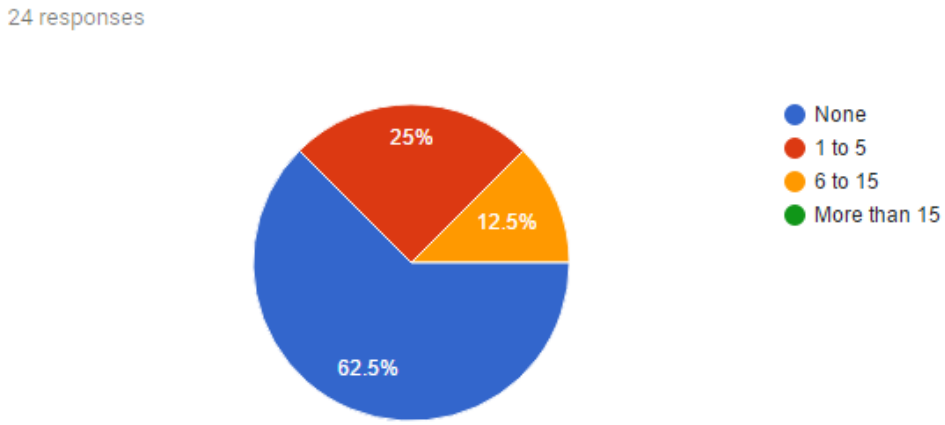

Figure 42: Models created or edited within the past 12 months by experts

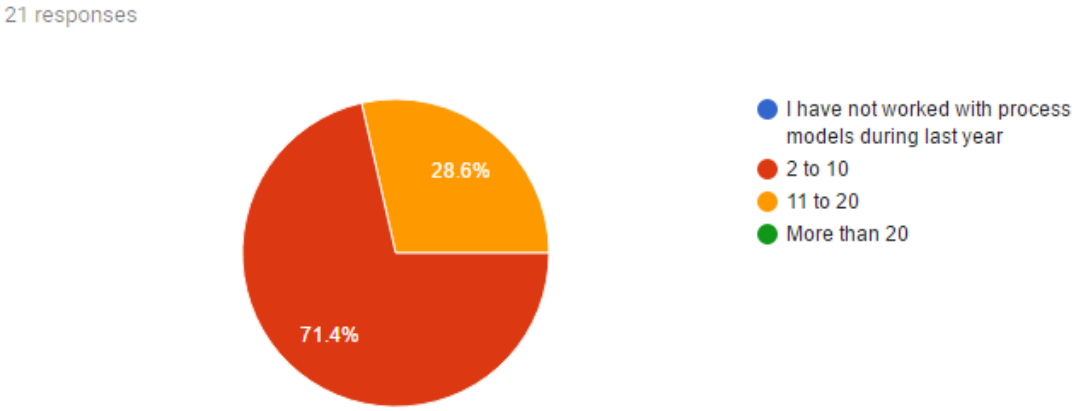

Figure 43: Students models sizes

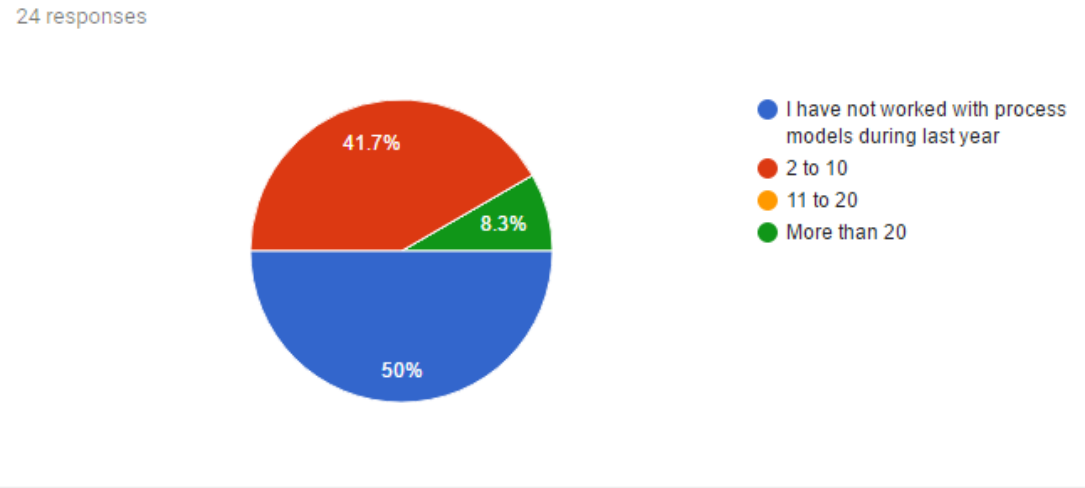

Figure 44: Experts models sizes

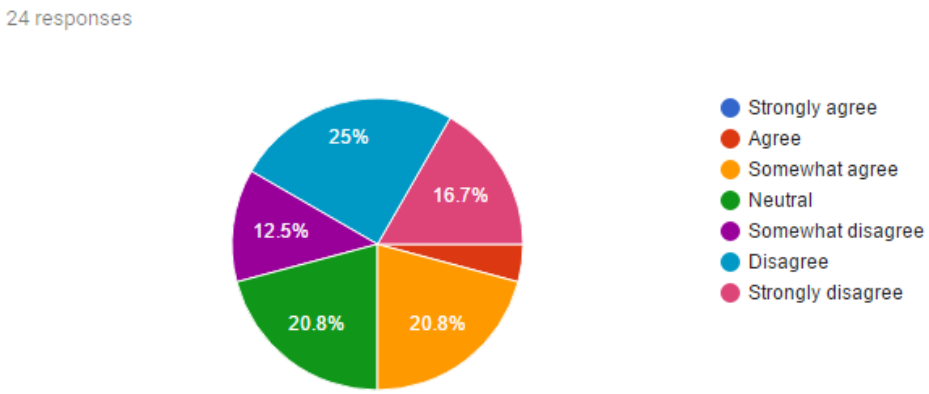

Figure 45: Experts familiarity with BPMN

G Licence

Non-exclusive licence to reproduce thesis and make thesis public

I,Allar Soo (date of birth: 12th of December 1991),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

1.2 make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

Automated process discovery: A literature review and a comparative evaluation with domain experts

supervised by Fabrizio Maria Maggi, Fredrik Payman Milani, Andrea Marrella and Massimo Mecella.

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 18.05.2017