UNIVERSITY OF TARTU
Institute of Computer Science
Software Engineering Curriculum

Etibar Hasanov

ENHANCING BPMN CONFORMANCE CHECKING WITH OR GATEWAYS AND DATA OBJECTS

Master's Thesis (30 ECTS)

Supervisor(s):
Fabrizio Maggi, Daniele Turato, Marco Pegoraro

Tartu 2017

Enhancing BPMN Conformance Checking with OR Gateways and Data Objects

Abstract

The Business Process Model and Notation is a developing standard for capturing business processes. Process models describe how the business process is expected to be executed. When a log is available from process executions, this situation raises the interesting question “Are the model and the log conformant?". Conformance checking, also referred to as conformance analysis, aims at the detection of inconsistencies between a process model and its corresponding execution log.

The BPMN conformance checker, as a part of a process mining tool, developed an Italian company called SIAV, however, this tool lacks some formal semantics. In particular, the previous conformance checking approach in Siav tends to focus on the control-flow in a process, while abstracting from data dependencies and process models containing OR gateways could not be used.

OR-join has an ambiguous semantics. The several formal semantics of this construct have been proposed for similar languages such as EPCs and YAWL. However, executing and verifying models using these semantics is computationally expensive. Therefore, in this thesis, we implemented enablement of an OR-join in linear time in the size of the workflow graph.

Data dependencies are also not considered in conformance checker developed in SIAV, which may lead to misleading conformance diagnostics. For example, a data attribute may provide strong evidence that the wrong activity was executed. That’s why the conformance checker should not only describe the process behaviour from the control flow point of view, but also from other perspectives like data or time. In the second part of the thesis, we enhanced the existing conformance checker with data attributes.

Keywords: Conformance checking with data, OR gateway, BPMN with data attributes

CERCS: P170 Computer Science, Numerical Analysis, Systems, Control

BPMN-i vastavusanalüüsi täiendamine OR väravate ja andmeobjektidega

Lühikokkuvõte

Äriprotsessimudel ja -notatsioon (BPMN) on arenev standard äriprotsesside graafilisesks kujutamiseks. Protsessimudel kirjeldab, kuidas äriprotsess peaks toimima. Kui äriprotsessi tegelikust käitamisest on saadaval ka sündmuste logi, on võimalik vastata küsimusele, kas protsessimudel vastab tegelikkusele. Vastavusanalüüs püüab tuvastada mittevastavusi protsessimudeli ja äriprotsessi käitamisel tekkinud sündmuste logi vahel. BPMN vastavuse

analüsaator on üks Itaalia ettevõtte SIAV-i poolt arendatud protsessikaeve tööriista osadest. Nimetatud tööriistal on aga puudujäägid formaalse semantika osas. Nimelt keskendub vastavusanalüüs järgnevuse voole (control-flow) protsessis, kuid jätab arvesse võtmata andmetevahelisi sõltuvusi. Lisaks ei ole vastavusanalüüsil võimalik kasutada protsessimudeleid, mis sisaldavad OR väravaid (OR gateway). OR-join omab mitmetähenduslikku semantikat. Se lle konstruktsiooni jaoks on pakutud mitmeid formaalseid semantikaid sarnastes keeltes, nagu EPCs ja YAWL. Nimetatud semantikate kasutatamine mudelite käitamisel ja vastavuse analaüüsil on aga arvutuslikult kulukas. Seega on käesolevas lõputöös implementeeritud OR värava aktiveerimine lineaarse ajalise sõltuvusega mudeli suuruse suhtes. Kuna SIAV-i vastavusanalüsaator ei võta arvesse andmetevahelisi sõltuvusi, võib puudulik analüüs viia vigase vastavusdiagnostikani. Näiteks võib andmeatribuut anda infot selle kohta, et käitati vale tegevus. Kirjeldatud põhjustel ei peaks vastavusanalüsaator tegelema vaid järgnevuse voo vastavuse analüüsiga, vaid peaks arvesse võtma ka andmeid ja nendevahelisi sõltuvusi ning aega. Käesoleva töö teises osas täiendati olemasolevat andmeanalüsaatorit andmeatribuutidega.

Võtmesõnad: Andmeanalüsaatorit andmeatribuutidega, OR värav, BPMN koos andmeatribuutidega

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine

Table of Contents

1 Introduction

1. Context and Motivation

If we ask people from different sectors "What is BPM for?" Probably we will get different answers [9]. Currently, BPM is used:

- By vendors. Some vendors use BPM for process improvement in some large companies. Some vendors use it for business process modelling and business process management
- By some consultants. The consultants use BPM for improvement and reengineering of business processes

Definition of BPM in [9] is: A management discipline focused on using business processes as a significant contributor to achieving the organisation's objectives through the improvement, ongoing performance management and governance of the essential business processes.

Business processes are the key instrument for organising activities and improving the understanding of their interrelationships. These activities can be enacted automatically without human involvement. Companies can reach their goals in an effective way only if people and other resources, such as information systems play together well [10].

Business Process Model and Notation (BPMN) is a widely used process modelling standard and development by Object Management Group in 2011 [5]. While most of the modelling languages focus on different levels of abstraction, such as from business level to a more technical level, the Business Process Model and Notation supports the complete range of abstraction levels, including business levels and software technology levels. "The primary goal of BPMN is to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes."[10]

2. Problem Statement

Modern organisations are aiming at delivering products in an efficient and effective manner. Organisations that operate at a higher process maturity level use formal/semiformal models to document their processes. In some cases, these models are used to configure process-aware information systems. However, in most organisations process models are not used to enforce a particular way of working. Instead, process models are used for discussion, performance analysis (e.g., simulation), certification, process improvement, etc. However, the reality may be different from such models. Companies tend to focus on idealised process models that have little to do with the reality. This illustrates the importance of conformance checking [11].

Conformance checking is a method which helps us for uncovering and measuring the discrepancies between models and executions. Conformance checking takes a process execution and a process model and measures the level of correspondence between the two. The differences help process engineers control over determining the severity of different types of discrepancies [25].

We develop our conformance checking technique on the top of the conformance checking tool developed in Siav, an Italian IT company.

Conformance Checking with OR Gateways

On conformance checking the following steps are done.

1) Token Replay
2) Comparing Footprints
3) Alignment

For conformance checking these steps need to be implemented. First, we get all the logs and we take each trace from a log file. With token replay, we execute each event from the log on the model and observe how the model will react. Later we compare footprints and find out if the model and log are aligned, or if there is a deviance. We will talk about footprints in section 2.4. In the end, we find an optimal alignment between the model and the log. We used [7] for token replay. In BPMN there are three types of gateways: inclusive, exclusive and parallel. In [7] all the gateways except inclusive gateway are implemented. OR gateway can be implemented by using AND and XOR gateways [8]. But it results in duplication of the same activities. If we need more OR gateways, BPMN model will get more complicated. OR-split is similar to the XOR-split, but the conditions on its outgoing branches do not need to be mutually exclusive, i.e. more than one of them can be true at the same time. OR-split activates one or more branches depending on which conditions are true. In terms of semantics, OR-split takes one token and generates at least one token, at most as much as the number of outgoing edges. Similar to the XOR-split gateway, an OR-split can also be equipped with a default flow, which is taken only when all other conditions evaluate to false [8].

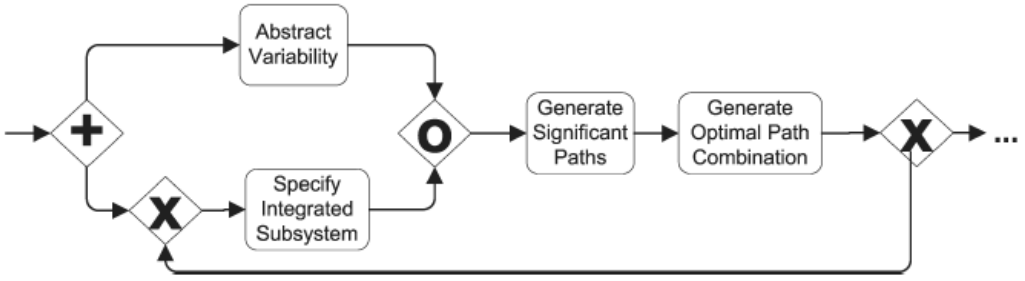

Figure. 1: Process fragment with an OR-join [3]

For each of its incoming branches, the OR-join will normally wait for a token indicating its completion; but if at some point in time it can be determined that no token will ever arrive along with a given incoming branch, the OR-join will not wait for a token along that branch.

OR-join normally waits for a token indicating its completion. However, sometimes it does not need to wait if ever a token will come from a definite branch or not. In figure 1 after completion of "Generate Optimal Path Combination" a choice can be made to "Specify Integrated Subsystem". After doing the task the OR-join gateway does not need to wait for a token from "Abstract Variability". OR-join gateway's enablement may depend on tokens in far places on a model. That's why the enablement for OR-join gateways using existing semantics is computationally very expensive. Another issue with OR-joins that enablement of OR-join may depend on enablement of another OR-join [3].

Running time of OR-join enablement is very important because, during the running of a trace, it may get called ten, even sometimes more hundreds or thousands time. Of course, it depends on the size of the model. Before implementing OR-join gateway we checked how many times AND-join and XOR-join get called.

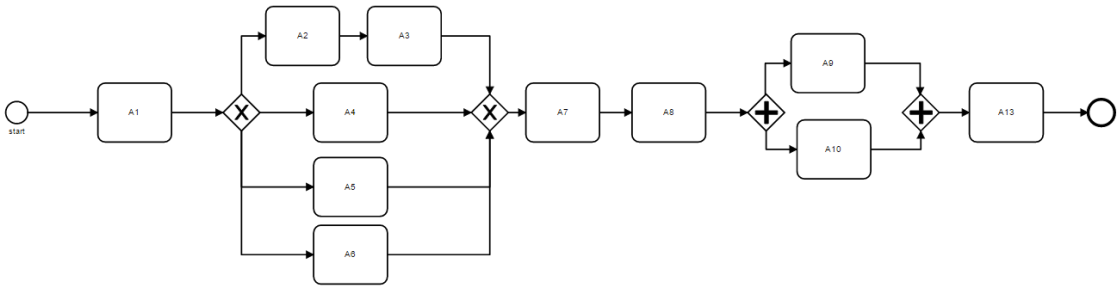

Figure. 2: BPMN model

We tried to measure how many times XOR-split is getting called while running alignment method between the BPMN model in figure 2 and the log {A1}. The XOR-split was called more than two thousand times. Later we designed the same model with AND gateways. And called an alignment method with the same log and AND-join was called more than three thousand times. For us, it was crucial to develop OR gateways which run in linear time.

Conformance Checking with Data

In order to reach the goal efficiently in the terms of cost and time, the processes need to be cost and time effective. In companies, process models are used for different reasons. In some companies, they may be used to enforce for a particular way of working. However, it is desirable to compare a model and an actual behaviour. The comparison can be used for performance analysis, process improvement and etc. This shows how important conformance checking is. Various conformance checking techniques have been proposed during recent years. Usually, they check by ordering of activities, control flow, however, they ignore data and resources [12].

One very important functionality that any process-aware information system should be able to support is conformance checking, i.e., the ability to verify whether the actual workflow is conformant with the business process model. There are some process constraints needs to be correct in order to consider aligned with the business goals. Multi-perspective constraints describe the expected behaviour, not only from the point of view of the control flow but also from other perspectives such as time and data.

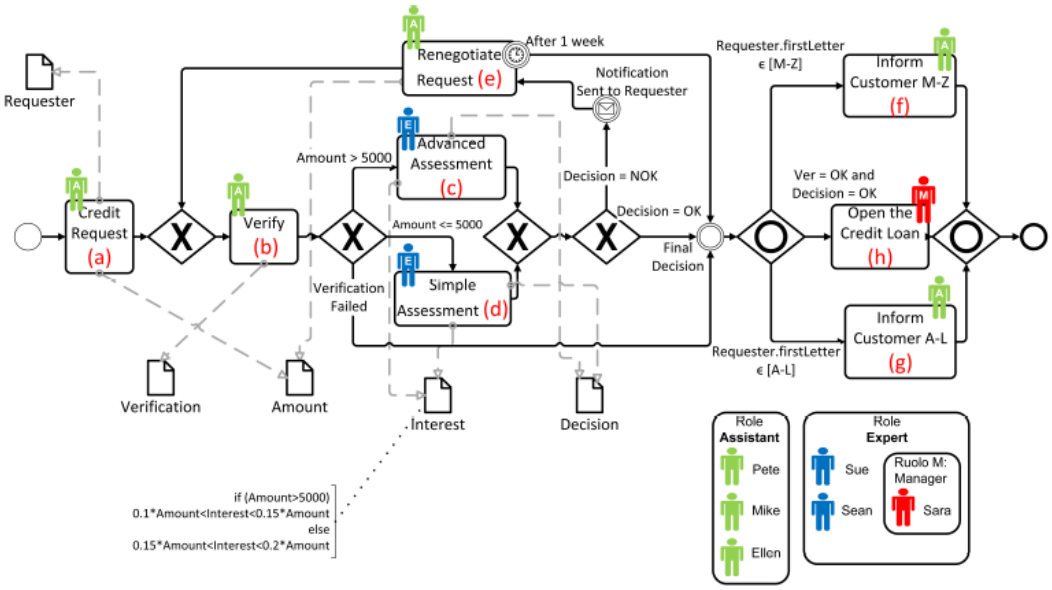

Figure 3: BPMN diagram of a data and resource-aware process to manage credit requests to buy home appliances. In the remainder, data objects are simply referred with the upper-case initials, e.g., V=Verification, and activity names with the letter in brackets, e.g. a=Credit Request [1]

Let us consider the following trace in figure 3:

((a; {A = 2000; R = Etibar; Ea = Pete; Ta = 31 Jan}); (b; {V = OK; Eb = Sue});

(c; {I = 250;D = OK; Ec = Sue ;Tb = 1 Feb});

If we do conformance checking by using control flow we do not detect any violation. However, conformance checking with data will find that Sue cannot verify because she is an expert, not an assistant.

Conformance checking which is developed in SIAV does conformance checking by control flow. And this conformance checking as we mentioned support neither OR-split nor OR-join. Apart from not supporting OR gateways, it needs to support conformance checking with data. In general, we need to implement multi-perspective conformance checker and give the possibility to users to use models including OR gateways for conformance checking.

3. Research Questions

We divided this thesis into two parts.

1) Enhancement of the conformance checker with OR gateways
2) Implementing a conformance checker with data perspective

In particular, we will try to find answers for the following questions.

1) Which approaches can be used for enhancing the conformance checker with the possibility for users to use BPMN models including OR gateway?
2) Which approaches can be used to have results of OR enablement in linear time?
3) Which approaches can be used for conformance checking with data?
4) How can we reduce execution time of conformance checking with data?

2 Related Work

The multi-perspective conformance checking describes process behaviour, not only from control flow point of view but also from other perspectives, such as date and time. Early works on conformance checking were focused on control flow perspective. These works were based on replaying logs on models and could tell which traces can be played, which not. We did research the works about multi-perspective conformance checking.

[37] Work distribution, temporal constraints and etc. can be encoded as data constraints. This paper demonstrates that data-aware decompositions can speed up conformance checking. As process models and event logs grow up in size, divide and conquer approaches are still needed to check conformance and diagnose problems.

In [24] and [25] first time the concept of conformance checking is investigated. Cook et al developed a technique for uncovering and measuring deviance between model and execution log and called it process validation. Two metric-oriented techniques were developed for process validation and the process validation implementation worked with finite-state machine models of the process. In [25] Rozinat and van der Aalst tried to answer the question "Do model and the log conform to each other?". The paper proposes an incremental proposes to check the conformance of a process model and an event log. The approach Rozinat and van der Aalst was, first the fitness between log and model is ensured, later the appropriateness of the model analysed with respect to the log. For addressing fitness one metric (f) was defined. Two metrics for structural appropriateness and two metrics for behavioural appropriateness was defined. All of them together are used for the quantification of conformance. Besides to quantification, Rozinat et al were able to locate the respective problem areas in the model and the log.

In conformance checking Imperative approaches capture allowed activities. As an alternative to the imperative paradigm, there is another paradigm which is called declarative paradigm. If we call imperative "close world" then declarative paradigm should be called "open world". Unlike imperative, a declarative model captures processes without limitation. As we can say everything is allowed as in "open world" unless it is forbidden by rule.

Imperative Paradigm

[26] Conformance checking finds out how good a model of a process is with respect to a log. In this paper, fitness dimension is focused. Given a process model and an event log, deviations in the fitness dimension manifest as either skipped or inserted activities. Skipped activities are the activities should be performed according to the model, but do not appear in the log. However, inserted activities are the ones occur in the log, but was not supposed to happen according to the model. In reality, the severity of skipping/inserting activities may depend on characteristics of the activity, e.g., some activities may be inserted without severe problems while the insertion of an important activity may lead to other significant problems Classical techniques measures penalize conformance for either skipped or inserted activities, however heuristics often result in an incorrect estimate of fitness. Therefore, the fitness does not correspond the real conformance. Adriansyah and et al proposes a cost-based replay technique that measures fitness and taking into account the cost of skipping and inserting individual activities. The technique is based on general framework uses A* algorithm to find the best fit of a case in a Petri. While replaying logs it doesn't only find unobservable activities, it finds inserted and skipped activities.

There are two kinds of compliance checking: forward compliance checking aims to design and implement processes where compliant behaviour is enforced, and backwards compliance checking aims to detect and localise non-compliant behaviour. [27] and [28] focuses on backward compliance checking based on event data. E. Ramezani Taghiabadi et al. provided and approach for data and resource aware compliance checking. Based on alignments, two techniques were provided to check the rules. The technique shows diagnostic information about violations of a compliance rule in a process instance, such as for each case which events violated temporary requirements and when the event should have happened. The techniques are feasible for compliance rules where data dependencies are between two or three attributes. For more complex rules additional pre-processing is needed.

In [29] while existing approaches mainly focus on the compliance checking part, this paper focuses on the pre- and post-processing steps that enable data-aware compliance checking by tackling the state explosion problem. In general, by abstracting from states of data objects irrelevant for the verification of a compliance rule, fewer cases need to be explored in the verification procedure. For doing automatic abstraction, three steps need to be accomplished. First, identifying data objects relevant to compliance rule and data conditions on them and data based gateways of a model. Second, identifying predicates for relevant data objects. Third, application of abstraction predicates to obtain abstract model and abstract compliance rule. Knuplesch and et al [29] showed how the state explosion can be reduced by conduction compliance checking for an abstract model and an abstract compliance rule.

In [30] for compliance checking new approach is suggested. Mainly approaches focus on verifying aspects related to control flow. However, giving useful feedback in the case of violation is ignored. By using BPMN-Q in the paper it is demonstrated how data can be incorporated into compliance rules. BPMN-Q is a visual language we developed for querying business process models, to express compliance requirements (compliance rules) regarding the execution ordering of activities (services) in process models. The approach showed in [30] has been implemented within the BPMN-Q query processor engine. The implementation covers mapping the rules defined in the paper into PLTL formula. PLTL is linear temporal linear logic (LTL) with past operators. LTL allows expressing formulae about the future states of the system. First, the data which caused violation is extracted, later this violation is visualised on the process model level. In order to explain violations, temporal logic querying is applied.

In [11], the paper presents conformance checking technique including data and resources. The proposition is heuristics-based, the approach is sub-linear. The main problem with the solution is that 70 percent of running time goes for parsing operations. For finding alignment A* is employed. The solution is similar to [2] which we discussed in the background section. The main difference is the calculation of heuristics function.

$$h(\gamma) = \kappa^{\min} \cdot (\|\sigma_L\| - \|\sigma'_L\|)$$

σ_L is the number of steps in the trace that have not been included, $\sigma_{L'}$ is the number of steps included in the execution of steps. K^{min} is the smallest number returned by a cost function.

Declarative Paradigm

Traditional business process model and notation (BPMN) and other related imperative approaches capture allowed activities. In [21] as an alternative to the imperative paradigm

which they tagged it as “closed world”, suggest different “open world” which is called declarative paradigm. Before several researchers exposed about imperative paradigm’s limitations. Unlike imperative, a declarative model captures, processes without limitation. As we can say everything is allowed as in ‘open world’ unless it is forbidden by rule. In the paper extending BPMN with declarative constructs is analysed. BPMN-D is a macro extension of BPMN. And it is shown in the paper that declare model can be transformed into BPMN-D. At the same time BPMN-D model can be transformed into BPMN, however, it is less readable and larger. And BPMN-D can be embedded into existing imperative process modelling notations without extending their existing semantics.

In [33] it has been investigated if imperative of declarative process modelling approach is better with respect to understanding matters. The models were compared reference to the understanding based on insights from cognitive research on programming. The paper suggests that imperative process models are better understandable than declarative models, either task type is sequential or circumstantial. A further insight concerns the theoretical axioms of the Cognitive Dimensions Framework, stating that tasks containing sequential information are better understandable using imperative languages, and tasks containing circumstantial information are better understandable using declarative languages. This could be confirmed partially. Apparently, sequential tasks are better understandable, regardless whether an imperative or declarative process model was used [33].

[31] Paper describes a controlled experiment conducted to compare two approaches for business process modelling: workflow and declarative. Silva et al [31] developed a tool which provides an interface for the creation and management of business rules based on templates of DECLARE, without relying on LTL implementation. DECLARE is a declarative language that combines semantics grounded in LTL with a graphical representation for users. They main research questions were:

- Compared to traditional workflow how easy is to model a declarative workflow?
- When does workflow fail declarative model successfully cope with unexpected situations?
- Is workflow better than declarative models when it comes to the execution of wrong paths?

40 students participated in the experiment. Students who used Declarative models reported difficulties in dealing with the growth of the number of rules. In the test declarative model showed higher quality compared to the workflow group, however, all the models required adjustments. The workflows required less effort to adjust than the declarative models. In the paper, it was concluded that Workflow models are easier to adjust. Although proponents of the declarative approach claim that declarative models are better for non-expected situations, no evidence was observed for supporting the claim.

In [32] it describes that with declarative models bring flexibility, however, it causes understandability problems and resulting maintainability problems of respective process models. The paper picks up this need and contributes a controlled experiment investigating the impact of the adoption of test cases on the maintenance of declarative process models. The result of the exploratory study shows that subjects read declarative process models in a sequential way. There were minor problems with single constraints, however, the combination of several constraints seems to be more challenging. Subjects failed to identify hidden dependencies.

F. Chesani et al. [34] describe a framework to check the compliance of process execution traces to declarative business rules. Three step methodology is proposed for developing and

applying rules. Under the formal framework, Prolog is used, due to its expressiveness. Prolog lets us complement tested rules with background knowledge composed of rules. This can help us expose new facts about the analysed traces. In order to use rules for reasoning, F. Chesani et al. sketched how rules can be mapped to Logic Programming, and it makes them possible to adopt Prolog for verification.

R. Masellis et al. [17] developed a monitoring tool for data-aware Declare constraints. The problem of verifying temporal constraints with data is theoretically challenging. It requires verification and database knowledge. Most of the literature on monitoring tools focus on checking propositional formulas, the database community studied offline analysis of temporal constraints. The logic is too expressive for supporting satisfiability because it is not possible to apply prediction mechanism of possible future evolutions. The framework represents an automate-based for the runtime (online) monitoring of process execution traces against dynamic, first-order constraints. And it can monitor Declare language extended with data-related aspects. The monitoring technique for conformance checking supports all the true values of RV-LTL (Runtime Verification of Linear Temporal Logic). But it doesn't support continuous verification. In other terms, after violation happens it stops providing verification.

In [18] a timed version of Declare was taken into consideration. It allows the use of time for more Declare constraints. First, by using timed automate, it was possible to warn users about a possible violation. In this way, the violation can be avoided. And this approach lets to detect early violations. The semantics are given in the terms of MTL, which is the timer version of LTL. Time spans between the tasks are considered even without duration taking place. The duration of the tasks is very important in this work. This is done by considering the beginning and completion of a task as separate events or by looking at tasks as signals not events. Interesting statistics can be found from these cases, such as how fast model can be executed given infinite resources, or how fast it can be executed using a given amount of resource, or how many resources needed to execute the model.
In [19] conformance checking with data for a declarative business process is done.
Advantage of the approach is

- It lets the BPs be specified in a declarative way
- Problems can be modelled easily
- It allows business data rules in process specification
- Gives detailed diagnostics while conformance checking
- Because of using constraint programming, it allows for efficient diagnostic process and conformance checking

Downside of the approach is that

- The business analysts must deal with a non-standard language for the declarative specification of BPs, therefore a period of training is required to let the business analysts become familiar with Declare specifications
- The considered constraint-based models deal with both control flow and data perspectives, but do not consider the resource perspective,
- Additional evaluations of our proposal in the context of real process executions are desired and are intended to be addressed as future work.

In [20] an approach conformance checking of events longs for multi-perspective Declare is proposed. The aim is proposing a conformance checking based on Declare that allows data and time. To this reason, MP-Declare is formally defined. The extended version of Declare allows for the definition of activation, correlation and time conditions to build constraints over the traces. The data perspective isn't fully integrated. Global variables must be true

during execution of the process and are disconnected from control flow. Another problem is that for declarative model efficient multi-perspective conformance checking isn't integrated.

In [35] the work explains converting a Declare model into an automaton and doing conformance checking of a log with the generated automaton. In the paper, new graphical language for modelling service is presented (DecSerFlow). For supporting rules like 'B' should be executed no later than 3 days after activity 'a' LTL is replaced by the real-time temporal logic. A logic can be translated timed automata. Later automata can be used for execution and verification of models with time perspective. The approach is based on the concept of alignment and as a result of the analysis, each trace is converted into the most similar trace that the model accepts. However, DecSerFlow does not support data. Extending the framework with data is a complex task. Data elements bring a lot of issues that need to be solved. It brings questions like "how to deal with data scope". Another complex issue is to find graphical representation dealing with data.

In [22] LTL checker first time presented, it is a language and a tool to enable the verification of properties based on event logs. In the paper, the main focus is verification, i.e., for given event log to verify certain properties. If there are events for submitting and accepting requests, the 4-eyes principle easily can verify it. When there is ordering for the presence of activities temporal logic is suggested. For temporal logic, there are two candidates: Computational Tree Logic and Linear Temporal logic. In this paper, the latter was picked. In LTL checker, L holds for the log, F is a formula. It evaluates for concrete parameter values. $\forall \pi \in L$ (check (F, π, 0)) returns true if F holds for the log. The plugin accepts logs in the MXML format which is tool independent format [22].

In [23] while checking conformance with respect to constraint it checks if constraints are violated or fulfilled. It lets a user find "healthiness" of the log. Π (a ->$\Diamond$ b) where 'b' is a target, let's say parameters 'a' and 'b' take the values "Create Questionnaire" and "Send Questionnaire". This constraint means that every action "Create Questionnaire" must eventually be followed by action "Send Questionnaire". The above characteristics make Declare very suitable for defining and analysing compliance models, i.e., checking whether the behaviour of a system (e.g., recorded in an event log of the system) complies with predefined regulations. Let's say in one trace 'a' is followed by 'b', and another one it isn't followed. So in the first one, it results with a fulfilment, in the second one with a violation. In [23] there are two terms for violation and fulfilment are used, violation ratio and fulfilment ratio. The ratio of violation and fulfilment are calculated over a number of activations. The diagnostics do not depend on the underlying LTL syntax. The paper shows that LTL constraints are not very readable for non-experts.

In multi-perspective conformance checkers time and data can be evaluated separately, in [20] it has been evaluated together as well. Conformance checking is done regarding MP-Declare (Multi-perspective) models. But conformance checker works with standard Declare models as well. In [20] the work is compared with [22], [23] and [20]. The test is done with 10, 20, 30 and 40 constraints, per trace 10, 20, 30 and 40 events used. Each log had 25_000, 50_000, 75_000 and 100_000 traces. For the small number of traces, the approach showed in [22] (van der Aalst et al., 2005) was faster, [20] performed better when the number of constraints per event and/or the number of traces per log and/or the number of logs increase.

e 33 A7 has right data attributes.

3 Conclusion and Future Work

The Business Process Modelling Notation is a developing standard for capturing business processes. Process models describe how the business process is expected or should be executed. Together with a log, this situation raises the interesting question “Are the model and the log conformant?". Conformance checking, also referred to as conformance analysis, aims at the detection of inconsistencies between a process model and its corresponding execution log. The BPMN conformance checking has been developed in SIAV lacked some formal semantics. Previous conformance checking approach in Siav tends to focus on the control-flow in a process while abstracting from data dependencies and process models containing OR gateways could not be used.

We investigated the enhancement of the conformance checker with OR gateways and implementation of the conformance checker with data attributes. About OR gateways we found out that the conformance checker can be enhanced with OR gateway linear time in the size of the workflow graph. Before implementing OR gateways while investigation, we found out that during the alignment process gateways can be called hundreds or thousands of times. That’s why the running time of OR gateway was crucial for our development. We implemented OR gateway in token replay and enhanced conformance checker with OR gateway as well. During the evaluation, we tested OR gateway with 10, 20, 30, 40, 50 links between two OR gateways. By applying linear regression methods we found out that the dependency between the size of workflow and the running time of OR-enablement can be expressed with a linear function.

Conformance checking with data techniques include Integer Linear Programming. We investigated different methodologies in conformance checking, including imperative and declarative models. And came to a conclusion that alignment methodology using integer linear programming is the best solution for us.

We developed a conformance checker with data and found out that running time of conformance checking with data for a single trace gets higher by the number of data attributes. However, while parsing logs we filtered unique traces. If all the traces with the same control flow, has missing data attributes or wrong value for the same data attribute or even missing data attribute, then we will consider all these logs the same. As we showed in the contribution section how it was handled.

4 References

[1] Massimiliano de Leoni, W.M.P. van der Aalst, "Aligning Event Logs and Process Models for Multi-Perspective Conformance Checking: An Approach Based on Integer Linear Programming" (2012).p 2.

[2] Felix Mannhardt, Massimiliano de Leoni, Hajo A. Reijers, Wil M. P. van der Aalst "Balanced multi-perspective checking of process conformance" (2015).

[3] Dumas, Marlon et al. "Semantics of standard process models with OR-joins." On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS (2007): 41-58.

[4] Völzer, Hagen. "A new semantics for the inclusive converging gateway in safe pro-cesses." Business Process Management (2010): 294-309.

[5] Dumas, M., La Rosa, M, Mendling, J. & Reijers, H.A.: "Fundamentals of Business Process Management". Springer-Verlag, 2013.

[6] W.M.P. van der Aalst, Process Mining—Discovery, Conformance and Enhancement of Business Processes (Springer, Berlin, 2011): 191-214

[7] I. Weber, A. Rogge-Solti, Chao Li, J. Mendling " CCaaS: Online Conformance Checking as a Service"

[8] Dumas, M., La Rosa, M, Mendling, J. & Reijers, H.A.: "Fundamentals of Business Process Management". Springer-Verlag, 2013. 72-74

[9] John Jeston, and Johan Nelis. Business process management, 2014.

[10] Weske, Mathias. Business process management: concepts, languages, architectures. Springer Science + Business Media

[11] Massimiliano de Leoni, Wil M. P. van der Aalst, and Boudewijn F. van Dongen " Data- and Resource-Aware Conformance Checking of Business Processes" 2015

[12] Andrea Burattin, Fabrizio M. Maggi, Alessandro Sperduti," Conformance Checking Based on Multi-Perspective Declarative Process Models"

[13] S.A. White, BPMN 1.0 business process modeling notation — OMG final adopted specification, on BPMN website, 2006. http://www.bpmn.org (May 2011).

[14] Dijkman, Remco M., Dumas, Marlon, & Ouyang, Chun (2008) Semantics and analysis of business process models in BPMN. Information and Software Technology

[15] W.M.P. van der Aalst "Process Mining Discovery, Conformance and Enhancement of Business Processes", Springer, 2011

[16] W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. IEEE Transactions on Knowledge and Data Engineering, 16(9):1128–1142, 2004.

[17] De Masellis, R., Maggi, F. M., & Montali, M. (2014). Monitoring data-aware business constraints with finite state automata. In International Conference on Software and Systems Process 2014, ICSSP (pp. 134–143).

[18] Westergaard, M., & Maggi, F. M. (2012). Looking into the future: Using timed automata to provide a priori advice about timed declarative process models. In Proc. of CoopIS LNCS. Springer

[19] Borrego, D., & Barba, I. (2014). Conformance checking and diagnosis for declarative business process models in data-aware scenarios. Expert Systems with Applications, 41 , 5340–5352

[20] Burattin, A., Maggi, F., Sperdutti, A..(2016). Conformance Checking Based on Multi-Perspective Declarative Process Models

[21] Giacomo, G., Dumas, M., Maggi, F. and Marco, M. (2015). Declarative Process Modeling in BPMN

[22] van der Aalst, W. M. P., de Beer, H. T., & van Dongen, B. F. (2005). Process mining and verication of properties: An approach based on temporal logic.

[23] Burattin, A., Maggi, F. M., van der Aalst, W. M. P., & Sperduti, A. (2012). Techniques for a Posteriori Analysis of Declarative Processes.

[24] Cook, J. E., & Wolf, A. L. (1999). Software process validation: Quantitatively measuring the correspondence of a process to a model. ACM Trans. Softw. Eng. Methodol., 8

[25] Rozinat, A., & van der Aalst, W. M. P. (2008). Conformance checking of processes based on monitoring real behavior.

[26] Adriansyah, A., van Dongen, B. F., & van der Aalst, W. M. P. (2011). Conformance checking using cost-based fitness analysis.

[27] Taghiabadi, E. R., Fahland, D., Van Dongen, B. F., & van der Aalst, W. M. P. (2013). Diagnostic information for compliance checking of temporal compliance requirements.

[28] Taghiabadi, E. R., Gromov, V., Fahland, D., & van der Aalst, W. M. P. (2014). Compliance Checking of Data-Aware and Resource-Aware Compliance Requirements.

[29] Knuplesch, D., Ly, L. T., Rinderle-ma, S., Pfeifer, H., & Dadam, P. (2010). On Enabling Data-Aware Compliance Checking of Business Process Models.

[30] Awad, A., Weidlich, M., & Weske, M. (2009b). Specication, Verication and Explanation of Violation for Data Aware Compliance Rules.

[31] Silva, N. C., de Oliveira, C. A. L., Albino, F. A. L. A., & Lima, R. M. F. (2014). Declarative versus imperative business process languages - A controlled experiment.

[32] Haisjackl, C., Zugal, S., So er, P., Hadar, I., Reichert, M., Pinggera, J., & Weber, B. (2013). Making sense of declarative process models: Common strategies and typical pitfalls.

[33] Pichler, P., Weber, B., Zugal, S., Pinggera, J., Mendling, J., & Reijers, H. A. (2011). Imperative versus declarative process modeling languages: An empirical investigation.

[34] Chesani, F., Mello, P., Montali, M., Riguzzi, F., Sebastianis, M., & Storari, S. (2009). Checking Compliance of Execution Traces to Business Rules.

[35] Montali, M., Pesic, M., van der Aalst, W. M. P., Chesani, F., Mello, P., & Storari, S. (2010). Declarative Specication and Verication of Service Choreographies.

[36] de Leoni, M., Maggi, F. M., & van der Aalst, W. M. (2014a). An alignment based framework to check the conformance of declarative process models and to preprocess event-log data.

[37] de Leoni, M., Munoz-Gama, J., Carmona, J., & van der Aalst, W. M. P. (2014b). Decomposing Alignment-Based Conformance Checking of Data-Aware Process Models.

[38] Dechter R, Pearl J (1985) Generalized best-first search strategies and the optimality of A*. J ACM (JACM) 32:505–536

[39] Vincenzi, B., Sperdutti, A., Turat, D. (2015). Allineamento del flusso di lavoro su modelli BPMN per l'analisi di conformita

5 Appendix

3. License

Non-exclusive licence to reproduce thesis

I, Etibar Hasanov,

(*author's name*)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for the purpose of preservation in the DSpace digital archives until expiry of the term of validity of the copyright

ENHANCING BPMN CONFORMANCE CHECKING WITH OR GATEWAYS AND DATA OBJECTS,

(title of thesis)

supervised by Fabrizio Maggi, Daniele Turato, Marco Pegoraro,

(supervisor's name)

2. Making the thesis available to the public is not allowed.

3. I am aware of the fact that the author retains the right referred to in point 1.

4. This is to certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu , **18.05.2018**