"The final authenticated version is available online at
https://doi.org/10.1007/978-3-030-62522-1_14"

Assessing the Compliance of Business Process Models with Regulatory Documents

Karolin Winter[1], Han van der Aa[2], Stefanie Rinderle-Ma[1], Matthias Weidlich[3]

[1] Faculty of Computer Science, University of Vienna, Vienna, Austria
{karolin.winter|stefanie.rinderle-ma}@univie.ac.at
[2] Data and Web Science Group, University of Mannheim, Mannheim, Germany
han@informatik.uni-mannheim.de
[3] Department of Computer Science, Humboldt-Universität zu Berlin, Berlin, Germany matthias.weidlich@hu-berlin.de

Abstract. Implementing regulatory documents is a recurring, mostly manual and time-consuming task for companies. To establish and ensure regulatory compliance, constraints need to be extracted from the documents and integrated into process models capturing existing operational practices. Since regulatory documents and processes are subject to frequent change, the constant comparison between both is mandatory. Additionally, new regulations must be integrated and checked against existing process models. To address these challenges, we provide an approach that uses natural language processing to automatically support compliance assessment between regulatory documents and process model repositories. The outcome is a pairwise matching between parts of a regulatory document and process models from a repository. This matching can be used to either determine the coverage of regulations by a process model or to guide compliance assessment by ranking models based on their fitness and cost. The approach is implemented and applied in two real-world case studies: one from the energy domain and the other based on the General Data Protection Regulation.

Keywords: Compliance Assessment, Regulatory Documents, Business Process Models, Natural Language Processing

1 Introduction

Due to the potentially enormous fines for non-compliance with regulations such as the GDPR, establishing and monitoring regulatory compliance is a crucial task for companies. Although many companies have their business processes captured in process models using, e.g., Business Process Model and Notation (BPMN) [15], in practice, checking and ensuring their compliance with regulatory documents is still mostly conducted manually [11]. This might lead to errors and high costs when implementing the regulations. Moreover, regulatory documents are constantly subject to change and new regulatory documents come into effect regularly [11].

In literature, several approaches for checking regulatory compliance based on process models exist [10]. However, existing compliance checking approaches abstract from the regulatory documents themselves and, instead, assume the existence of a set of extracted compliance constraints that are already captured using formalisms such as LTL or Event Calculus [10]. Although some approaches aim at extracting process descriptions [3] or compliance constraints [21] from regulatory documents, an approach to directly assess compliance between process models and regulatory documents during design time, is missing, despite providing the following advantages: i) it avoids misunderstanding, misinterpretation, and errors in the extraction and formalization of compliance constraints [11]; ii) it facilitates the monitoring of changes in the regulatory documents; iii) it enables the implementation of new regulatory documents; and iv) it supports sanity checks for process models with respect to regulatory compliance. This work aims to fill this gap by addressing the following research questions:

- *RQ1* How to identify which parts of a regulatory document relate to which process models in a repository?
- *RQ2* How to measure and assess compliance violations between a process model and regulatory constraints?

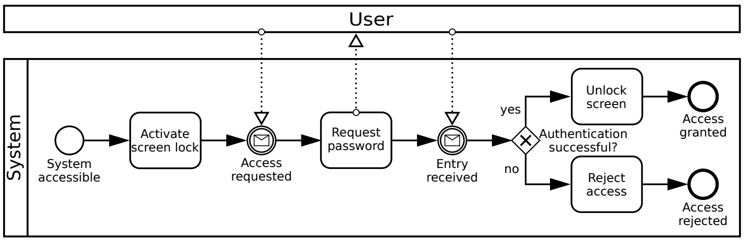
Addressing these research questions involves several challenges, like differences in granularity between documents and process models. As shown in previous work, regulatory documents are extensive and structured along topics, making it advisable to fragment the documents into paragraphs [21]. However, certain regulations might be implemented in various process models [16]. Hence, we aim to support complex many-to-many relations between parts of regulatory documents and process models. To assess compliance, we provide a matching of paragraph-model pairs using a fitness score ($\mapsto$ *RQ1*) and a cost score ($\mapsto$ *RQ2*). The *fitness score* determines the likelihood that a paragraph pertains to a particular process model, whereas the *cost score* quantifies the amount of detected process compliance violations. Our work particularly targets the detection of control-flow violations in terms of mandatory activities that are missing in a process, as well as activities that are performed in the wrong order. Furthermore, we detect resource-related violations in terms of activities that are not performed by the correct organizational entity.

Based thereon, we support two compliance assessment scenarios. When a matching between paragraphs and process models is pre-defined by a user, the cost score measures compliance, whereas the fitness score enables conclusions on the coverage of regulations by a model. In the absence of a given matching, the scores guide compliance assessment by ranking models that show high coverage, but low compliance, with respect to certain parts of a regulatory document.

The remainder of the paper is structured as follows. Section 2 illustrates the problem and highlights the main challenges addressed in this work. Section 3 describes our compliance assessment approach. Section 4 presents an evaluation based on two real-world case studies, followed by a discussion in Section 5. After a review of related work in Section 6, the paper concludes in Section 7.

P_1: A screen lock provides the possibility to conceal the information currently displayed on the screen. In order that access to an IT system is reliably prevented during a short absence of the IT user, it should only be possible to disable a screen lock after successful user authentication **[R_1]**. It should be possible for the user to activate the screen lock manually **[R_2]**. In addition, the screen lock should be automatically initiated after a predefined period of inactivity **[R_3]**.

(a) Description of Screen-lock Regulation



(b) Process model M_1: Screen-lock protection

Fig. 1: Running Example of an IT Security Scenario, Based on [16]

2 Problem Illustration and Challenges

This section presents an exemplary scenario to illustrate the challenges imposed by the problem of assessing the compliance between regulatory documents and business process models. To this end, we consider IT security regulations stemming from the *IT Baseline Catalogues* of the German Federal Bureau of Security in IT[4]. As a starting point, we consider a full regulatory document, consisting of multiple paragraphs, and a repository of process models. However, for illustrative purposes, we here present a single regulatory paragraph stemming from the English version of the document, as well as an accompanying process model.

The respective paragraph of the regulatory document, P_1, is presented in Fig. 1a and describes various rules associated with the protection of IT systems using screen locks[5]. Given such a paragraph, as part of a larger document, automated compliance assessment involves the following steps: (1) Extracting the compliance rules that paragraph P_1 imposes on processes, (2) identifying that the rules from P_1 relate to process model M_1, depicted in Fig. 1b, and (3) determining whether the rules from P_1 are properly implemented by M_1.

To perform these steps, a variety of challenges need to be addressed. For instance, step (1) requires an approach to differentiate between the sentences describing actual constraints and those providing additional clarification or context. To determine correspondences between a paragraph and a model in step (2), an automated approach needs to be able to deal with the inherently flexible nature of natural language, which allows the same constraints to be expressed in a broad range of manners, can lead to considerable differences in terminology

[4] `www.bsi.bund.de/grundschutz`

[5] The identifiers (R1 to R3) have been inserted for clarity.

between paragraph and model, as well as to differences in granularity [2]. Finally, step (3) requires the assessment of the exact constraint that a compliance rule imposes on a process and to consider how this constraint should be reflected in a corresponding process model.

3 Compliance Assessment Approach

Figure 2 presents an overview of our compliance assessment approach illustrated using the running example (cf. Section 2). The input consists of a process model repository $\mathcal{M}$ and a set of paragraphs $\mathcal{P}$ stemming from a regulatory document. We assume that the paragraphs included in $\mathcal{P}$ contain actual regulations, i.e., introductory sections and reference lists are omitted. As shown in Fig. 2, our approach first parses the process models in $\mathcal{M}$ and the paragraphs in $\mathcal{P}$. The former is straightforward, whereas the latter involves the identification and extraction of constraints imposed by the regulatory document, as well as, e.g., control flow aspects like strict orders between activities. The output is a pairwise matching of process models and paragraphs based on fitness and cost scores.

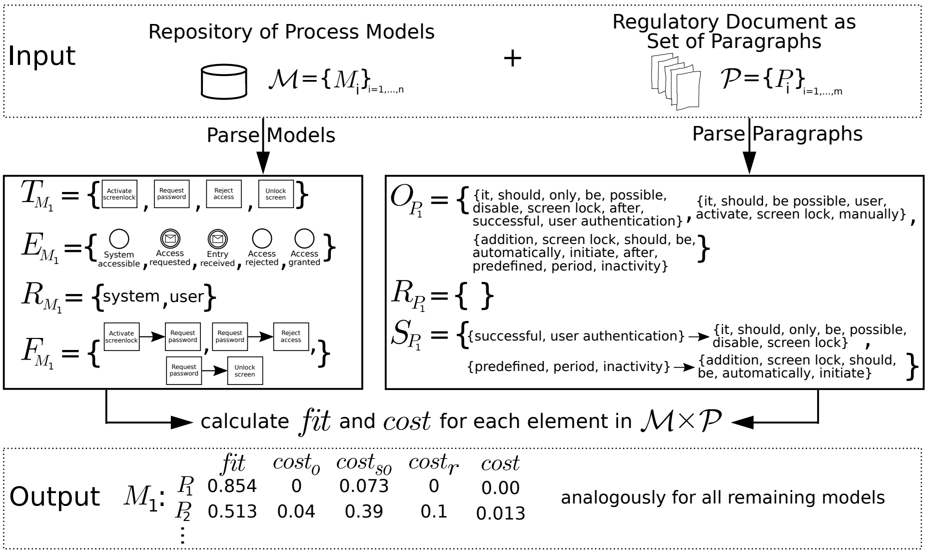


Fig. 2: Compliance Assessment Approach Illustrated Based on Fig. 1

The *fitness score* $\mathit{fit}(P, M)$ quantifies the likelihood that a paragraph $P \in \mathcal{P}$ pertains to a model $M \in \mathcal{M}$. The *cost score* $\mathit{cost}(P, M)$ quantifies the distance (i.e., cost), between the obligations expressed in paragraph P and the process implemented by model M. Our work targets both control-flow and resource-related compliance violations in terms of three violation types: (**V1**) an obligatory activity is not incorporated in the model, (**V2**) activities are executed in the wrong order and (**V3**) activities are not performed by the correct resource.

By combining $fit(P, M)$ with $cost(P, M)$, our approach is able to detect those paragraph-model pairs that have a strong semantic relation (i.e., P specifies rules that are relevant to model M), but are also likely subject to compliance issues.

3.1 Parsing Process Models

Process models can be created using a variety of modeling languages, such as Petri nets, EPCs, and the Business Process Model and Notation (BPMN). Since our work is independent of the specific notation used to define a process model, we define process models using a generic definition, given in Definition 1.

Definition 1. *A process model is a tuple M that consists of:*

- $N_M = T_M \cup E_M \cup G_M$ *is a finite set of nodes, with* T_M *a set of tasks,* E_M *a set of events, and* G_M *a set of gateways,*
- $F_M \subseteq N_M \times N_M$ *is the flow relation, s.t.* (N_M, F_M) *is a connected graph,*
- $t_M : G_M \to \{and, xor, or\}$ *is a function that maps each gateway to a type,*
- R_M *is a finite set of resources,*
- $u_M : R_M \nrightarrow T_M \cup E_M$ *is a partial function that maps resources to tasks and events.*

In the process model-parsing step, our approach parses all models in a repository (e.g., from JSON files) into a collection $\mathcal{M}$, where each model $M \in \mathcal{M}$ fits the provided definition. Whereas notations such as BPMN can be directly and fully mapped to this format, other notations may only result in a partially populated process model definition. For instance, a process model notation that lacks resources will result in an empty set R_M. This naturally prevents the detection of compliance violations involving resources (V3). Furthermore, we note that sub-processes can be flattened into one process model.

3.2 Parsing Regulatory Documents

In this step, our approach aims to extract constraints specified in paragraphs of a regulatory document together with those elements of constraints that enable the detection of compliance violations. In particular, for each paragraph $P \in \mathcal{P}$, the approach extracts a set O_P of obligatory activities that must be performed according to P, a strict order relation $S_P \subseteq O_P \times O_P$ that indicates pairs of activities that must be executed in a specific order, a set R_P of described resources, and a partial function $u_P : R_P \nrightarrow O_P$ that specifies which resources must execute which activities. These are extracted as follows.

Extracting Obligatory Activities $\mathbf{O_P}$. The extraction of obligatory activities described in a paragraph P represents a two-step procedure. Our approach first distinguishes between sentences in P that describe actual process constraints, i.e., sentences that contain mentions of obligatory activities, and those that provide contextual information, e.g., *A screen lock provides the possibility to conceal the information currently displayed on screen.* To discern between these two types of sentences, we recognize that obligatory activities correspond to mentions of some action (typically a verb), associated with a closed-class of signal words, including *must, should, shall*, and *has to* (cf. [7]), whereas contextual sentences lack such obligations. We refer to the set of sentences with mentions of obligations, identified in this manner, as $P' \subseteq P$.

From the sentences in P', our approach then aims to extract individual activities. The extraction of activities using tailored techniques based on heuristics (cf. [8, 17]) is known to be error prone for long, complex sentences [1]. Therefore, our approach rather splits each sentence in P' into one or more clauses, which can be achieved by employing existing NLP techniques, cf., spaCy [12]. Each clause is then added as an activity to O_P. For instance, for a complex sentence like: *The electrical supply must be interrupted, an entry in the logbook of the terminal is generated and the status is transmitted to the central system.*, our approach recognizes three clauses and, thus, three activities: [(1) *The electrical supply must be interrupted*, (2) *an entry in the logbook of the terminal is generated*, and (3) *the status is transmitted to the central system*].

Extracting Order Restrictions $\mathbf{S_P}$. To extract order restrictions between the activities in O_P, we build on existing work on the extraction of process models from natural language text [8]. This extraction procedure identifies signal words, e.g., *then, after* or *afterwards*, to detect specific orders in which activities must be performed. For the running example, our approach recognizes two such order restrictions, both based on the marker *after*, resulting in S_P containing two restrictions: 1. *successful user authentication* $\rightarrow$ *it should only be possible to disable screen lock* and 2. *a predefined period of inactivity* $\rightarrow$ *screen lock should be automatically initiated.* Both examples indicate clear restrictions about which activity must be executed first (e.g., successful user authentication), before the next activity is allowed (e.g., disabling a screen lock).

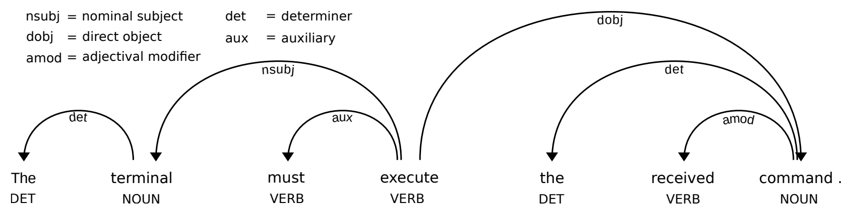


Fig. 3: Sentence with Dependency Tree and Part-of-Speech Tags

Extracting Resource Responsibilities $\mathbf{R_P}$. To identify the resources R_P associated with obligatory activities, we again employ NLP techniques such as dependency parsing and POS tagging. Figure 3 depicts a sentence with its corresponding dependency tree and POS tags obtained using spaCy [12]. In the displayed example, the resource corresponds to *the terminal*, identified as the nominal subject via the corresponding dependency label *nsubj*. However, given the flexibility of natural language, the same constraint could also have been expressed as, e.g., *The received command must be executed by the terminal*. This makes the resource the *agent* of the sentence. The corresponding element in R_P would then be given as $\{$*terminal* $\rightarrow \{$*must, execute, received, command*$\}\}$.

3.3 Matching Paragraph-Model Pairs Based on Fitness and Cost

The compliance assessment between a process model repository and a regulatory document is based on the computation of fitness and cost scores between each paragraph-model pair in $\mathcal{P} \times \mathcal{M}$. The *fitness score* $\mathit{fit}(P, M)$ quantifies the likelihood that paragraph $P \in \mathcal{P}$ pertains to model $M \in \mathcal{M}$. The *cost score* $\mathit{cost}(P, M)$ quantifies the distance (i.e., cost), between the process constraints imposed by paragraph P and the actual process implemented by model M, computed as the total number of violations detected between P and M.

Fitness Score. The fitness score aims to quantify the likelihood that a paragraph P relates to a model M. For instance, we use fitness to recognize that paragraph P_1 from Fig. 1, which relates to screen lock protection, should be paired with model M_1, rather than, e.g., a process model related to password management. Our approach achieves this by first identifying correspondences between each obligatory activity in O_P and its most similar process model element (task or event) in $T_M \cup E_M$. We denote the similarity between some $o \in O_P$ and $t \in T_M \cup E_M$ using $\mathit{sim}(o, t) \in [0, 1]$. In our approach, we use existing *semantic similarity* measures for this quantification [12], which allows the approach to recognize when comparable process steps are described using synonymous terms. Based on these similarity values, we define a set $\mathcal{C}_{P,M} \subseteq O_P \times T_M \cup E_M$ containing the correspondences with the highest similarity scores for each $o \in O_P$. Thus, $(o, t) \in \mathcal{C}_{P,M}$ denotes that t is the process model element with the highest similarity to o. To omit unimplemented obligatory activities from consideration in the fitness computation, we introduce a threshold $\gamma \in [0, 1]$ that filters out correspondences with a low similarity score. We shall use $C_{P,M,\gamma} = \{(o, t) \in C_{P,M} \mid \mathit{sim}(o, t) > \gamma\}$ to denote the set of correspondences above this threshold. Based on this set, we define fitness as the average similarity values obtainable for the obligatory activities in O_P that are greater than γ, as given in Eq. 1.

$$\mathit{fit}(P, M, \gamma) = \frac{\sum_{(o,t) \in C_{P,M,\gamma}} \mathit{sim}(o, t)}{|C_{P,M,\gamma}|} \tag{1}$$

Consequently, $\mathit{fit}(P, M)$ receives a high value if there is a strong relation between the obligatory activities described in the paragraph P and the tasks and events contained in the process model M.

Cost Score. The cost score provides a quantitative assessment for compliance violations $\mathbf{V1} - \mathbf{V3}$. For a model $M \in \mathcal{M}$ and a paragraph $P \in \mathcal{P}$, the cost score $\mathit{cost}(P, M, \gamma, \delta) \in [0, 1]$ is defined as follows:

$$\mathit{cost}(P, M, \gamma, \delta) := w_o \mathit{cost}_o(P, M, \gamma) + w_{so} \mathit{cost}_{so}(P, M, \gamma) + w_r \mathit{cost}_r(P, M, \gamma, \delta) \tag{2}$$

where $w_o, w_{so}, w_r \in [0, 1]$ with $w_o + w_{so} + w_r = 1$ are weights that allow users to alter the relative importance of the violation types. Parameters γ and δ represent similarity thresholds that are explained below. Note that a high violation in one of the three cost scores can be weakened by the weights, e.g., if $cost_o = 1, cost_{so} = cost_r = 0$ and $w_o = w_{so} = w_r = \frac{1}{3}$ then the overall costs evaluate to $cost = \frac{1}{3}$. Another possibility is to take the maximum function instead of a weighted sum to compute the overall cost score. However, this can also be achieved by setting the weights accordingly, i.e., for the example, $w_o = 1, w_{so} = w_r = 0$.

Missed Obligatory Activities (V1). We define $cost_o(P, M)$ to quantify the amount of obligatory activities from P that are not implemented in model M. To recognize such cases, we define a threshold $\gamma \in [0, 1]$ that captures the minimal similarity value that is required for an obligatory activity o to be recognized as implemented in a model through task (or event) t, i.e., if $sim(o, t) < \gamma$ for $(o, t) \in \mathcal{C}_{P,M}$, the obligation imposed by activity o is considered violated in model M. The cost score between model and paragraph is then computed as follows:

$$cost_o(P, M, \gamma) := \frac{|\{(o, t) \in \mathcal{C}_{P,M} \mid sim(o, t) < \gamma\}|}{|O_P|} \tag{3}$$

Strict Order Violations (V2). Strict order violations occur when a regulatory document specifies that two activities should be executed in a specific order, whereas this order is not enforced in the model. To recognize such cases, we compare the strict order relation S_P extracted from a paragraph P to the flow relation F_M of the model. In particular, let $(o, o') \in S_P$ be a strict order constraint and let $(o, t), (o', t') \in \mathcal{C}_{P,M}$. Then, this strict order constraint is only satisfied if there is a path from t to t', i.e., $(t, t') \in F_M^+$, and not vice versa, i.e., $(t', t) \notin F_M^+$. However, the enforcement of such a constraint in the model shall only be assessed, if the model indeed includes sufficiently similar tasks (or events), as again determined by a parameter γ, for the obligatory activities o, o'. We therefore limit the relation S_P to these constraints as $S_{P,M,\gamma} = \{(o, o') \in S_P \mid \exists\ (o, t), (o', t') \in \mathcal{C}_{P,M} : sim(o, t) > \gamma \land sim(o', t') > \gamma\}$. Then, the cost score between model and paragraph is computed as follows:

$$cost_{so}(P, M, \gamma) := \frac{|\{(o, o') \in S_{P,M,\gamma} \mid \exists\ (o, t), (o', t') \in \mathcal{C}_{P,M} : (t, t') \in F_M^+ \land (t', t) \notin F_M^+\}|}{|S_{P,M,\gamma}|} \tag{4}$$

Resource Responsibility Violations (V3). Resource responsibility violations occur when the regulatory document specifies that an activity must be performed by a specific resource, whereas a different resource executes the corresponding process model task. Given an obligatory activity o and a process model task t, such that $(o, t) \in \mathcal{C}_{P,M}$, we compare the resource $r_P \in R_P$ assigned by u_P to o to the resource $r_M \in R_M$ assigned by u_M to t. Here, we consider that a resource responsibility is satisfied if r_P and r_M are sufficiently similar, i.e., if $sim(r_P, r_M) \geq \delta$. This way, we are able to recognize comparable resource descriptions, e.g., a *supervisor* to a *manager*. Let $R_{P,M,\gamma}$ be the resources in the document assigned to an activity for which there exists a γ-similar task in the process model, i.e., $R_{P,M,\gamma} = \{r \in r_P \mid \exists\ (o, t) \in \mathcal{C}_{P,M} : u_P(r) = o \land sim(o, t) > \gamma\}$. Furthermore, let $\phi(r, r', \gamma, \delta)$ be a predicate that holds true for resources $r \in R_{P,M,\gamma}$ and $r' \in R_M$ if their assigned documents and tasks are γ-similar, $(u_P(r), u_M(r')) \in \mathcal{C}_{P,M}$ and $sim(u_P(r), u_M(r')) > \gamma$, and the resources are δ-similar, $sim(r, r') > \delta$. Then, the cost score between model and paragraph is computed as follows:

$$cost_r(P, M, \gamma, \delta) := \frac{|\{r \in R_{P,M,\gamma} \mid \exists\, r' \in R_M : \phi(r, r', \gamma, \delta)\}|}{|R_{P,M,\gamma}|} \tag{5}$$

Compliance Assessment. The computed fitness and cost scores are applicable in two different scenarios that will be picked up in the evaluation.

The first application scenario presumes that the correct paragraph-model pairs are already known. Assessing a paragraph-model pair via the above scores then enables the following conclusions: The lower the costs, the better the compliance of a model to a paragraph. The fitness score, in turn, serves as a quality indicator of the process model. It indicates the coverage of the paragraph. Moreover, if an additional paragraph has a high fitness score, this paragraph is likely to contain compliance constraints that also refer to the model.

In a second application scenario, pre-defined paragraph-model pairs are not given. In this case, our approach acts as a sort of recommender system that displays a top-k list of paragraph-model correspondences. Here, k can, e.g., depend on the median of all fitness values, meaning that only paragraph-model pairs having a fitness score higher than the median are considered. The top-k pairs are then ranked based on their cost score, in order to highlight the pairs with the most compliance issues. Note that this result suggests a paragraph-model matching that shall serve as a basis for manual assessment by a domain expert.

4 Evaluation

The evaluation experiments comprise two real-world cases, intended to show the effectiveness of our compliance assessment approach. A prototype was implemented in Python3 taking as input a collection of .bpmn-files, e.g., exported from Signavio, and a collection of .txt-files, corresponding to paragraphs of a regulatory document. For the NLP tasks, e.g., parsing, analysis of grammatical relations, and similarity computation, we employ the spaCy library [12].

Section 4.1 reports on experiments conducted for Austrian's energy providers and demonstrates the first application scenario, i.e., the assessment of compliance between an already known paragraph-model matching. Section 4.2 presents a case study for the second scenario in the context of GDPR implementation, i.e., the detection of compliant paragraph-model pairs from scratch. While the former experiments are based on proprietary data, the GDPR data collection and the employed implementation are publicly available.[6] In both case studies, we assign equal values to the weights in the cost calculation ($w_o = w_{so} = w_r = 1/3$), set the resource threshold to $\delta = 0.8$, but vary γ. We conducted the experiments using an Intel Core i5-7200U @2.50GHz processor (4 cores) and 8GB RAM.

[6] `http://gruppe.wst.univie.ac.at/projects/RegMiner/index.php?t=prototypes`

Each experiment required at most 16 minutes, illustrating the feasibility of our approach from a computational perspective.

4.1 Smart Meters for Austrian Energy Providers

This case study demonstrates the first application scenario, i.e., compliance assessment for already known paragraph-model pairs.

Input. The input consists of a repository of 12 process models, which were established and verified by domain experts in the context of earlier work [6] and have one-to-one correspondences to 12 paragraphs of the related regulatory document for smart electricity meters.[7]

Evaluation Strategy. For each process model $M \in \mathcal{M}$, we establish a ranking $rank(M, \mathcal{P})$ in which all paragraphs in $\mathcal{P}$ are ranked according to their fitness score. Furthermore, we use $rank(M, P)$ to denote the ranking of paragraph P in $rank(M, \mathcal{P})$, with 1 being the highest rank and $\hat{P}_M \in \mathcal{P}$ to refer to the paragraph that actually corresponds to model M.

We quantify our approach's accuracy using common measures for recommender systems [18]: The *average precision* (AP) per model, as well as the *mean average precision* (MAP) for the whole model repository, defined as:

$$AP(M) = \frac{1}{rank(M, \hat{P}_M)}, \qquad MAP(\mathcal{M}) = \frac{\sum_{M \in \mathcal{M}} AP(M)}{|\mathcal{M}|}$$

Since there is only a single relevant paragraph $\hat{P}_M$ per model, the AP value corresponds to the inverse of its rank. The mean value MAP is computed by averaging the AP values for all 12 models in $\mathcal{M}$. As such, it provides an overall quantification of how good the fitness score performs for this particular document collection. The cost score is evaluated in a qualitative manner.

Configurations. We employ two configurations for this use case that vary in the way that obligatory activities are extracted. One configuration employs the exact method defined in Section 3.2, yielding a set of obligatory activities O_P. The other configuration skips the filtering of sentences and, thus, transforms the clauses from all sentences into obligatory activities, yielding a set O'_P. This latter configuration is introduced given the procedural style present in this particular regulatory document, which uses fewer words to explicitly denote obligations.

Results. For the fitness score, the AP and MAP results are, respectively, depicted in Fig. 4 and Tab. 1. As displayed in Fig. 4, the O_P configuration achieves the best result for $\gamma = 0.8$.[8] In this case, 6 out of 12 models are matched to the correct paragraph (i.e., the paragraph is ranked in the first spot). However, for the remaining models, the average precision indicates that the correct paragraph was actually ranked low on the list.

[7] `https://oesterreichsenergie.at/files/Downloads%20Netze/Oesterreich%20Use%20Cases%20Smart%20Metering_14122015_Version_1-1.pdf`

[8] Due to limited space within the figures and since the values for AP and MAP are lower than for $\gamma = 0.8$ the results for $\gamma = 0.9$ are omitted.

The O'_P configuration performs better, in particular for $\gamma = 0.6$. In this case, for 9 out of 12 models the correct paragraphs are ranked in first place, while for the remaining models the correct paragraphs are ranked in second place. These results emphasize the potential of the fitness score, achieving a perfect correspondence for 75% and a top-2 ranking for all models. As shown in Tab. 1, this results in a mean average precision of $MAP = 0.875$. Furthermore, the results reveal that the O'_P configuration performs better than O_P, due to the aforementioned descriptive nature of the regulatory document.

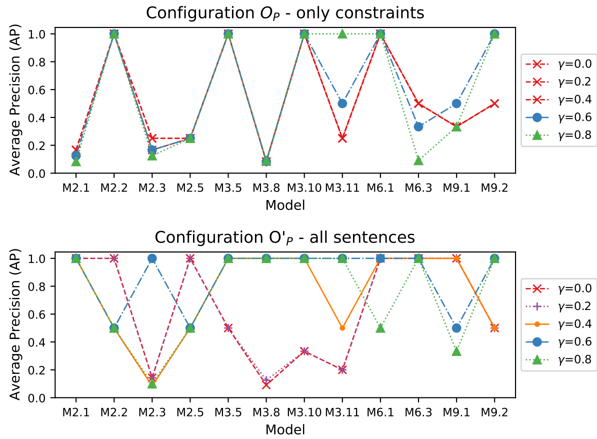


Fig. 4: Average Precision per Model for Smart Meters Case Study

Table 1: Mean Average Precision for Smart Meters Case Study for O_P and O'_P

Configuration	$\gamma = 0.0$	$\gamma = 0.2$	$\gamma = 0.4$	$\gamma = 0.6$	$\gamma = 0.8$
O_P	0.528	0.528	0.528	0.579	0.581
O'_P	0.647	0.650	0.760	0.875	0.745

Regarding the cost score, the expected observations are confirmed, i.e., for higher γ values, the $cost_o$ increases while $cost_{so}$ and $cost_r$ decrease. Moreover, the costs are mostly zero, correctly indicating that manually verified process models have few compliance violations.

4.2 General Data Protection Regulation – GDPR

The second case study corresponds to the second application scenario, i.e., the identification of compliant paragraph-model pairs.

Input. As input we take seven process models (taken from [4]), which capture how the main privacy constraints of the GDPR can be implemented within processes. Aside these seven process models, we consider GDPR Articles 5 to 50, since Articles 1 to 4 contain introductory statements, whereas Article 51 and onwards apply to supervisory authorities rather than organizations.

Evaluation Strategy. We exemplary describe the results for the model depicting a pattern for a *data breach*. In order to verify whether the approach works well, we identified in a manual analysis relevant articles. It turned out that Articles 33 and 34 contain most information on the situation of a data breach and Article 40 contains one constraint referring to that topic.

Results. As within the first case study, we tested several values for γ. For $\gamma = 0.0$, Article 29 has the highest fitness (0.87508)[9] and Article 16 has the lowest fitness score (0.73057). The median of the fitness score is at 0.79285 and except for three articles having a fitness greater than the median, the cost score evaluates to zero. Among these, Article 34 has the second highest fitness score, Article 33 the third highest fitness score while having both $cost = 0$. If we want to detect from scratch which paragraph-model pairs are most compliant, we would therefore have the correct ones on top of the list regarding fitness. For $\gamma = 0.4$ the same situation holds while for $\gamma = 0.6$ both articles are shifted down by one position, i.e., in third resp. fourth place and for Article 33 $cost_{so} = 1.0$. For $\gamma = 0.8$ both articles are further shifted down in the ranking and costs for Article 33 additionally increase while Article 34 has now $cost_o = 0.05$. In contrast to the first case study, the ideal γ is at 0.4 for this model-paragraph matching.

5 Discussion and Limitations

Within this section a reflection and discussion of results as well as limitations and suggestions how to resolve these are outlined.

The evaluation demonstrated the impact of parameter γ on the results. In particular, the higher γ, the fewer obligations are identified with corresponding obligatory activities, resulting in a lower MAP. However, MAP also decreases if too many smaller similarities are allowed. For the cost score, similar effects occur, i.e., a stricter threshold γ can lead to an increase of costs. In scenarios where a model-paragraph alignment is available, a suitable value for γ can be chosen by selecting the value that achieves the highest MAP. When such an alignment is not available, the parameter can be set based on the γ value that leads to the highest overall fitness score. For the case reported on in Section 4.1, this would result in $\gamma = 0.6$, which corresponds to the configuration with the second highest MAP value for the O_P configuration and to the highest MAP for the O'_P one. This thus suggests that the fitness score could be a useful proxy value for the parameter selection.

A limitation arises if resources have contrary names within the model and paragraph since semantic similarity would probably fail to identify them as similar. Adapting the threshold δ is one possibility but this would allow for undesired behaviour, i.e., resource cost could increase tremendously. Having a user defined mapping would be more feasible. By now, our approach focuses on mandatory tasks but there might be optional constraints within a paragraph, e.g., indicated by *can*. As future work we plan to adapt the cost score such that compliance violations caused by optional constraints are considered as well.

[9] Article 29 has one of the highest fitness scores for almost each of the process models since this article just consists of one single constraint, i.e., the chance of having a high semantic similarity with one obligatory activity from a process model is high.

6 Related Work

Various approaches provide (semi-)automated support for business process compliance checking (see [11] for an overview). Our work targets so-called design-time compliance analysis, which aims at detecting compliance issues during development and implementation of a business process [13]. Most techniques in this regard require regulations to be first transformed into a formal representation, e.g., temporal logic [10], rather than operating on the regulatory document itself.

The extraction of process constraints from natural language text is typically conducted as part of a broader use case and is a core requirement for approaches that automatically extract process models from process-oriented texts [8, 9, 20]. Other approaches aim to elicit process constraints from rule-oriented texts, such as the extraction of requirements from documents [5, 7, 19] and declarative process constraints [1]. More broadly, constraint extraction from regulatory documents is related to requirements elicitation from text. According to the survey in [14], most of the existing approaches are manual or semi-automatic. Other related work compares textual process descriptions to process models [2, 17]. However, those works assume that a textual process description relates to exactly one, already known process model, whereas the regulatory documents considered by our work can be subject to complex, many-to-many correspondences.

7 Conclusion

This work enables the automatic assessment of the compliance between process models and regulatory documents. For this, we presented a notion of fitness to identify which of the compliance constraints extracted from a regulatory document concern a certain model. In addition, we proposed a cost function to measure the distance between these constraints and the process as captured by the model, thereby highlighting potential violations. The effectiveness of our approach has been demonstrated in two case studies.

Our work supports companies to cope with frequent changes of regulations and of process models through automated support. As the first approach of its kinds, our work opens several avenues for future research. The approach can be expanded to incorporate additional types of compliance violations, such as those stemming from *prohibitive* rather than *obligatory* statements in regulatory documents, as well as those covering the data and time perspectives of processes. Furthermore, our proposed cost function can be employed to go beyond the detection of violations by helping to assess which process change operation may be employed to ensure compliant process execution.

References

1. Van der Aa, H., Di Ciccio, C., Leopold, H., Reijers, H.A.: Extracting declarative process models from natural language. In: CAiSE. pp. 365–382 (2019)

2. Van der Aa, H., Leopold, H., Reijers, H.A.: Comparing textual descriptions to process models–the automatic detection of inconsistencies. Inf. Syst. 64, 447–460 (2017)
3. Van der Aa, H., Leopold, H., Reijers, H.A.: Checking process compliance against natural language specifications using behavioral spaces. Inf. Syst. 78, 83–95 (2018)
4. Agostinelli, S., Maggi, F. M., Marrella, A., Sapio, F.: Achieving GDPR Compliance of BPMN Process Models. Inf. Syst. Eng. in Responsible Inf. Syst. 10–22 (2019)
5. Bajwa, I.S., Lee, M.G., Bordbar, B.: SBVR business rules generation from natural language specification. In: AAAI spring symposium. pp. 2–8 (2011)
6. Böhmer K., Stertz F., Hildebrandt T., Rinderle-Ma S., Eibl G., Ferner C. Burkhart S., Engel D.: Application and Testing of Business Processes in the Energy Domain. Big Data Management Systems in Business and Industrial Applications (2017)
7. Dragoni, M., Villata, S., Rizzi, W., Governatori, G.: Combining NLP approaches for rule extraction from legal documents. In: MIREL (2016)
8. Friedrich, F., Mendling, J., Puhlmann, F.: Process model generation from natural language text. In: CAiSE. pp. 482–496 (2011)
9. Ghose, A., Koliadis, G., Chueng, A.: Process discovery from model and text artefacts. In: Services. pp. 167–174 (2007)
10. Hashmi, M., Governatori, G.: Norms modeling constructs of business process compliance management frameworks: a conceptual evaluation. Artif. Intell. Law 26(3), 251–305 (2018)
11. Hashmi, M., Governatori, G., Lam, H.P., Wynn, M.T.: Are we done with business process compliance: state of the art and challenges ahead. Knowl. and Inf. Syst. 57(1), 79–133 (2018)
12. Honnibal, M., Montani, I.: spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing (2017)
13. Kharbili, M.e., Medeiros, A.K.A.d., Stein, S., van der Aalst, W.M.: Business process compliance checking: Current state and future challenges. MobIS (2008)
14. Meth, H., Brhel, M., Maedche, A.: The state of the art in automated requirements elicitation, Inf. Softw. Technol. 55(10):1695–1709 (2013)
15. Omg, O., Parida, R., Mahapatra, S.: Business process model and notation (BPMN) version 2.0. Object Management Group (2011)
16. Rinderle-Ma, S., Kabicher-Fuchs, S.: An indexing technique for compliance checking and maintenance in large process and rule repositories. EMISAJ 11, 2–1 (2016)
17. Sànchez-Ferreres, J., van der Aa, H., Carmona, J., Padró, L. Aligning textual and model-based process descriptions. Data & Knowledge Eng., 118 (2018)
18. Schröder, G., Thiele, M., Lehner, W.: Setting goals and choosing metrics for recommender system evaluations. UCERSTI2 (2011)
19. Selway, M., Grossmann, G., Mayer, W., Stumptner, M.: Formalising natural language specifications using a cognitive linguistic/configuration based approach. Inf. Syst. 54, 191–208 (2015)
20. Sinha, A., Paradkar, A.: Use cases to process specifications in business process modeling notation. In: Web Services. pp. 473–480 (2010)
21. Winter, K., Rinderle-Ma, S.: Deriving and combining mixed graphs from regulatory documents based on constraint relations. In: CAiSE. pp. 430–445 (2019)