Towards a Formal Framework for Partial Compliance of Business Processes

Ho-Pun Lam[1], Mustafa Hashmi[1,2,3], and Akhil Kumar[4]

[1] Data61, CSIRO, Australia
[2] Federation University, Brisbane QLD 4000, Australia
[3] La Trobe Law School, La Trobe University, Melbourne, Australia
{brian.lam,mustafa.hashmi}@data61.csiro.au
[4] Smeal College of Business, Penn State University, University Park, PA 16802, USA
akhilkumar@psu.edu

Abstract. Binary "YES-NO" notions of process compliance are not very helpful to managers for assessing the operational performance of their company because a large number of cases fall in the grey area of partial compliance. Hence, it is necessary to have ways to quantify partial compliance in terms of metrics and be able to classify actual cases by assigning a numeric value of compliance to them. In this paper, we formulate an evaluation framework to quantify the level of compliance of business processes across different levels of abstraction (such as task, trace and process level) and across multiple dimensions of each task (such as temporal, monetary, role-, data-, and quality-related) to provide managers more useful information about their operations and to help them improve their decision making processes. Our approach can also add social value by making social services provided by local, state and federal governments more flexible and improving the lives of citizens.

Keywords: Partial compliance · Business process modelling · Compliance measures · Process compliance

1 Introduction

When designing business processes (BPs), practitioners always assume that the business model will be executed as planned. However, this is impractical in many situations. For example, cost fluctuations, equipment and resource availability, time constraints, and human errors can cause disruptions. In response to this, it is crucial for the practitioners to have a complete picture of the status of their running business processes — for taking strategic decisions on identifying, forecasting, obtaining and allocating required resources, and to be notified if any non-compliance issues are identified during execution.

Let us illustrate this idea by examining the payment process model as shown in Figure 1, which consists of a sequence of tasks to be performed. Accordingly, a customer is required to make the payment within 15 days upon receiving the invoice; if not, the invoice must be paid with 3% per day interest in addition

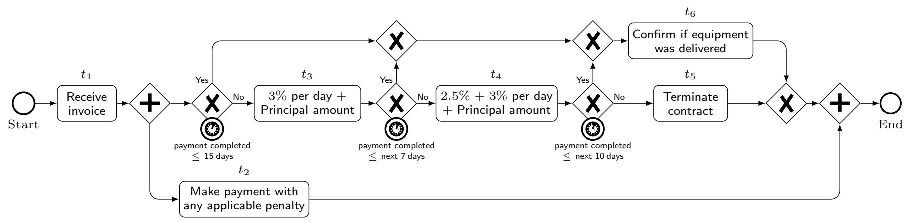

Fig. 1: Fragment of payment-making process (adopted from: [4])

to principal amount within the next 7 days. For any subsequent days hereafter within the next 10 days, an additional 2.5% interest will be added to the total payment as penalty, which will be calculated based on the principal amount. The contract will be terminated automatically upon 3 consecutive defaults.

Now, consider two compliant executions performed by two business customers of the company, customers A and B. Customer A strictly follows the *normal* sequence and makes the payment within 15 days after receiving the invoice. Customer B instead delayed the payment and paid the bill (together with interest and penalty) 3 weeks later. If we ascribe value to this process depending on the billing company's revenue, both executions positively contribute to it, as both customers did make their payments after receiving the invoices. However, the deferred payment of company B may affect the cash flow of the service provider company. Moreover, both these scenarios represent examples of partial compliance because there was a violation on the temporal dimension. Other violations may occur along other compliance dimensions such as: *money*, when monetary payments are not made according to agreements; *roles*, when individuals who perform certain tasks like approvals, etc., are not in the normal or authorized, or delegated, role; *data*, when the complete data required to perform a task is not available; and *quality*, when the quality of the work performed by a task is sub-standard. For each dimension, there are prescribed ranges of values or performance indicators in which a task is considered to be compliant on that dimension. If the indicator values within a narrow range are outside this normal range, then the task is said to be in partial compliance on that dimension. Finally, if the indicator does not fall into either of these two ranges then it is said to be non-compliant. A dimension can also be related to an attribute value. Thus, *payInDays* attribute represents the number of days within which payment is made after the invoice is sent to the customer. This attribute corresponds to the temporal dimension and can be used interchangeably with it.

Existing systems and compliance management frameworks (such as Declare [11], SeaFlows [10], COMPAS [14], etc.) only provide an *all-or-nothing* type of binary answer, i.e., *YES* if the BP is *fully* compliant; and *NO* if any non-compliant behavior has been detected at some point during execution, which is not *informative* and raises a simple yet significant question of whether the *whole* process is not compliant or *only* a part of it, and whether corrective actions

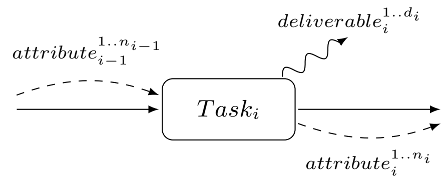

Fig. 2: Annotations of a task

should be performed from the point where the non-compliant behavior was detected, or from an earlier point.

Recently, some efforts coining the notion of *partial compliance* have been reported. For example, the approach in [9] returns the status of a BP as *ideal*, *sub-ideal*, *non-compliant* and *irrelevant*. Based on the notion of decision lattices [6], Morrison *et al.* [12] categorizes the compliance status as *Good*, *Ok*, and *Bad*. However, the issues remain similar as: *to what extent the process is compliant and how much (or what kind of) additional resources are required to resolve any detected non-compliance issues?*

To answer this question, in this paper, we present a formal framework for evaluating the levels of compliance of a BP at different levels of abstraction during execution and auditing phases, aiming to provide more clear and useful information to users concerned in facilitating their decision making process when any non-compliance issues arise.

The rest of the paper is organized as follows. Next, in Section 2, we provide necessary background information and terminologies following which we introduce our proposed framework in Section 3. Examples illustrating how the proposed framework works in practice are presented in Section 4. Related work is discussed in Section 5 before the paper is concluded with final remarks and directions for future work in Section 6.

2 Background and Problem Statement

In this section, we first introduce the necessary background and terminologies for the understanding of our proposed framework, and subsequently derive the problem statement.

Structure of a Business Process

A BP is represented as a temporally and logically ordered, directed graph in which the nodes represent tasks of the process that are executed to achieve a specific goal. It describes what needs to be done and when (*control-flows* and *time*), who is involved (*resources*), and what it is working on (*data*) [3]. Essentially, a BP is composed of various elements which provide building blocks for aggregating loosely-coupled (atomic) *tasks* as a sequence in a process aligned with the business goals.

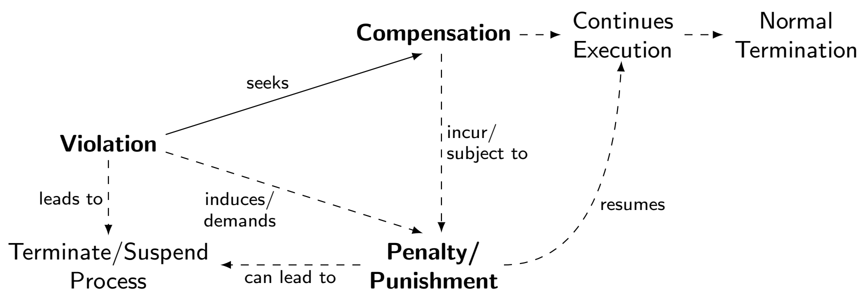

Fig. 3: Violation-compensation relationships of a partially compliant business process

Each task is an atomic unit of work with its own set of *input attributes*, which can be (partially) aggregated from *preceeding* task(s) or acquired from other sources, describing the *prerequisites* or *requirements* that the task has to comply with for its (full) execution, and *output attributes* that it has to produce upon execution and for propagation to *succeeding* tasks (as input), as shown in Figure 2. Note that, in Figure 2, the term *deliverable* is used to describe an output document or *artefact* that is produced by the task after execution and is not propagated to the next task. Technically, the values of attributes (both input and output) can have multiple dimensions, which may include information about time (or *temporal*), monetary, data, role, quality of service, or any combination of these. As an example, the value of payment and the payment due date of t_2 in Figure 1 are from temporal and monetary dimensions, respectively.

A sequence representing the execution order of tasks of a BP in a given case is called a trace (a.k.a. occurrence sequence). Typically, a BP can be executed in a number of ways. For instance, below is the set of traces that can be generated from the business model, from *start* to the *end*, as shown in Figure 1.

$$
\begin{aligned}
\mathfrak{T}^+ = \{ & \mathfrak{T}_1 = \langle t_1, t_2, t_3, t_4, t_5 \rangle, && \mathfrak{T}_2 = \langle t_1, t_3, t_2, t_6 \rangle, \\
& \mathfrak{T}_3 = \langle t_1, t_2, t_3, t_4, t_6 \rangle, && \mathfrak{T}_4 = \langle t_1, t_3, t_4, t_2, t_5 \rangle, \\
& \mathfrak{T}_5 = \langle t_1, t_2, t_3, t_6 \rangle, && \mathfrak{T}_6 = \langle t_1, t_3, t_4, t_2, t_6 \rangle, \\
& \mathfrak{T}_7 = \langle t_1, t_2, t_6 \rangle, && \mathfrak{T}_8 = \langle t_1, t_3, t_4, t_5, t_2 \rangle, \\
& \mathfrak{T}_9 = \langle t_1, t_3, t_2, t_4, t_5 \rangle, && \mathfrak{T}_{10} = \langle t_1, t_3, t_4, t_6, t_2 \rangle, \\
& \mathfrak{T}_{11} = \langle t_1, t_3, t_2, t_4, t_6 \rangle, && \mathfrak{T}_{12} = \langle t_1, t_3, t_6, t_2 \rangle, \\
& \mathfrak{T}_{13} = \langle t_1, t_6, t_2 \rangle \}
\end{aligned}
$$

While it is always desirable that a BP behave strictly in accordance with the prescribed conditions, this may not always be the case in practice. A BP may deviate from its desired behavior in unforeseen circumstances and violate some (or all) of the conditions attached to it during execution.

Figure 3 illustrates what can happen when a violation occurs in a BP. The divergent behavior may cause a temporary suspension or (in some cases) termination of the process, and may also induce penalties. A *penalty* is a punitive measure (e.g. monetary or in some other form) enforced by company policy or a rule for the performance of an action/act that is proscribed, or for the failure to carry out some required acts. However, a violation of (mandatory) conditions does not necessarily imply automatic termination/suspension of a BP that would prevent any further execution. Certain violations can be compensated for [5], where compensation can be broadly understood as a remedial measure taken to offset the damage or loss caused by the violation. In general, legal acts and contracts provide clauses prescribing penalties and remedial provisions which are triggered when the deviations from the contractual clauses occur. These provisions may prescribe conditions that are subject to some penalties or punishments. As mentioned in the previous section, an execution with compensated violations (as in Figure 1) leads to a sub-ideal situation [13], and is deemed partially-compliant. Nonetheless, the process can continue execution and complete normally once the compensatory actions are performed.

Next, we develop our framework in a formal manner.

3 Partial Compliance Framework

In this section we develop a partial compliance framework. The framework is based on the following principles or axioms underlying partial compliance:

Axiom 1. Compliance should not be binary 0/1 but should cover a spectrum of scenarios between 0 and 1.

Axiom 2. Partial compliance should be recognized and treated fairly.

Axiom 3. Partial compliance can be rectified by compensation mechanisms such as imposition of penalties, or sanctions that increase monotonically with the extent of the violation.

Axiom 4. The level of partial compliance decreases monotonically as the magnitude of the violation increases.

Throughout this section, we use the following notations: $\mathcal{T}$ is the set of unique task identifiers of tasks that appear in an instance of a trace $\mathfrak{T}$; and, $\mathcal{A}_n$ denote the set of attribute names of task t. Each attribute is mapped to a value v from a suitable numeric or categorical domain in a running instance. Thus, (a, v) is a attribute-value pair or tuple for an attribute in a task.

We introduce a partial compliance function ψ on task t to define partial compliance values for different attribute values under various compliance dimensions. Thus, $\psi_t(a, v, d)$ denotes the degree of partial compliance of attribute a of task t where the value of attribute a is equal to v, on compliance dimension d. This function maps attribute values of a task to a real-value in the $[0, 1]$ range that represents the degree of partial compliance, where 1 corresponds to full compliance. Thus, to formally describe the partial compliance for the running example in Figure 1, we can write, $\psi_{t_2}(payInDays, 10 \text{ days}, Time) = 1$. This means that the partial compliance of task T_2 in dimension $Time$ is equal to 1 if the $payInDays$ attribute has a value of 10 days, which also represents compliance of task T_2 on the temporal dimension.

Given a task t, we denote metric $\mathcal{D}_t = \{\mathcal{D}_t^1, \ldots, \mathcal{D}_t^n\}$ as the set of compliance dimensions that relate to t attributes and denote its size as $|\mathcal{D}|$. Thus, for the running example of Figure 1, $\mathcal{D}_{t_2} = \{\text{Monetary, Time, Percent}\}$ and $|\mathcal{D}| = 3$.

Hence, given a set of attribute names $\mathcal{A}_n$, it is necessary to determine which attributes relate to compliance and aggregate their individual compliance into a single metric of compliance. Thus, one can decide if a task is fully-, partially- or non-compliant. Accordingly, we introduce the following definitions:

Definition 1 (Aggregate attribute compliance metric). *Given a task $t \in \mathcal{T}$, $\mathcal{D}_t = \{\mathcal{D}_t^1, \ldots, \mathcal{D}_t^n\}$ as the set of n compliance dimension(s) of attributes of t, and an attribute aggregation operator $\odot$, then we define a compliance metric for attribute a of task t on dimension i be:*

$$M_t^d = \bigodot_{(a,v)|a \in \mathcal{A}_n} \psi_t(a, v, d)$$

be the aggregate compliance value across all task attributes for which $d \in \mathcal{D}_t$ is the dimension relevant to an attribute a in task t, v is the value of the attribute a in task t. In addition, we denote $\psi_t(a, v, d) = null$ if dimension d does not apply to attribute a in task t. Thus, any compliance dimension with a null value will be simply ignored from the aggregation. Finally, $M_t^i \in [0, 1]$.

Definition 2 (D-Compliant). *Given a task $t \in \mathcal{T}$, $\mathcal{D}_t = \{\mathcal{D}_t^1, \ldots, \mathcal{D}_t^n\}$ as the set of compliance dimensions that relate to its attributes, M_t^i as the aggregate attribute compliance metric per Definition 1, and $S^i, \Delta^i \in \mathbf{R}$, then,*

- *t is* non-compliant *on dimension $\mathcal{D}_t^i$ iff $M_t^i < S^i$;*
- *t is* partially compliant *on dimension $\mathcal{D}_t^i$ iff $S^i \leq M_t^i < S^i + \Delta^i$; and*
- *t is* fully compliant *on dimension $\mathcal{D}_t^i$ iff $M_t^i \geq S^i + \Delta^i$*

where S_t^i and Δ_t^i are the standard and threshold values for full and partial compliance, respectively. Note that the threshold represents a range or window around the standard value in which partial compliance is possible. These values are generally numeric constants provided by the domain experts to the analysts.

Definitions 1 and 2 define how the attribute metric value should be calculated and conditions for different levels of compliance, respectively. This means that a task is fully compliant if it is executing under some *ideal* situation; while a task is partially compliant if its attributes in $\mathcal{D}_t$ are to a large extent in accordance with the requirements specified but a few of them have been violated and remedial actions have been performed to repair/compensate the situation such that all violations identified have either been resolved or compensated; or a task is non-compliant otherwise.

The D-compliance score on dimension $\mathcal{D}_t^i$ is given by M_t^i and is a real value in $[0, 1]$. For a non-numeric value, the attribute dimension metric may be recorded on a qualitative scale such as a 3-point scale of (low, medium, high) or on a 5- or 7-point Likert scale. In this case, the points on the scale can be mapped uniformly to the 0-1 scale. Thus, by default, high would correspond to 1, medium to 0.67 and low to 0.33. Alternatively, a user-defined mapping function may be employed for this purpose. In general, rules can also be applied to determine a user-defined mapping function for nominal compliance values. Thus, given a task with an attribute a and a 3-point scale of (low, medium, high) in dimension d, a set of rules can be written as follows using three reasonable cut-off values:

$$\psi(a, v, d) = \begin{cases} 0.25 & \text{if } v = \text{“low”} \\ 0.50 & \text{if } v = \text{“medium”} \\ 0.90 & \text{if } v = \text{“high”} \end{cases}$$

Once the individual attribute value has been evaluated, they can be combined in different ways to obtain a dimension compliance score M_t^i.

Below are some alternative methods to compute the aggregation operator $\odot$ for an attribute.

1. *Average method.* Take a (weighted) average of attribute dimension metric values. This will give an average across the individual scores across all the applicable dimensions. For three dimensions with scores of 0.7, 0.9 and 1, the average would be 0.867. It is also possible to assign different weights to each dimension based on its importance.
2. *Product method.* Take the product of all attribute dimension metric values. In this case, we would multiply across all the $\psi(a, v, d)$'s. Thus, in the above example we would obtain 0.63. In general the product approach would lead to a lower value than the average approach.
3. *Rule-based method.* Apply a more general rule-based method to combine the individual metrics. Thus, a rule could be expressed as:

   $$\text{If } (\psi(a_1, v_1, d_1) < 0.5) \text{ AND } (\psi(a_2, v_2, d_2) < 0.5) \text{ then } M_t^i = 0.$$

   which states that if the partial compliance on metrics 1 and 2 is less than 0.5 then the task is non-compliant even though it is partially compliant on individual metrics, perhaps because these two metrics are very important.

The simplest implementation of M_t^i is to set $\odot$ to the (weighted) average of all non-null compliance values after evaluations, i.e., $M_t^i = \frac{1}{|\mathcal{D}_t^i|} \Sigma \psi_t(a)$. However, we should be cautious when selecting which function to use in computing M_t^i as setting $\odot = max$ would mean that whenever an attribute in a dimension is fully compliant, then the task will also be fully compliant in this particular dimension and similar will apply when we set $\odot = min$, which may not be something that we intended.

Example 1. A review loan application task has $S^i = 3$ days. $\Delta^i = 2$ days. If the task takes 4 days, it is partially compliant on the dimension $\mathcal{D}^T$.[5] But if it takes 6 days, it is non-compliant.

[5] From now on, we will use $\mathcal{D}^M$, $\mathcal{D}^T$, $\mathcal{D}^R$, $\mathcal{D}^D$, and $\mathcal{D}^Q$ to denote the monetary, time, role, data, and quality dimensions of a task, respectively.

Based on the definitions above, the level of compliance of tasks can be defined in terms of a metric outside a permitted range for one or more related dimensions, such as money, time, role, data, quality, etc. Thus, a task in the process may be required to be performed by a worker in a role using certain data inputs or documents. There is also a time limit for the completion of a task and a quality requirement. Finally, some tasks may also require the monetary payment of a fee (e.g. an application fee for admission to a school, processing fee for issuance of a passport or permit, etc.).

Definition 3 (T-compliance). *Given a task $t \in \mathcal{T}$, and $\mathcal{D}_t = \{\mathcal{D}_t^1, \ldots, \mathcal{D}_t^n\}$ as the compliance dimensions that correspond to its various attributes, then we define:*

- *t is* non-compliant *iff $\exists \mathcal{D}_t^i \in \mathcal{D}_t$, $\mathcal{D}$ is* non-compliant*;*
- *t is* fully compliant *iff $\forall \mathcal{D}_t^i \in \mathcal{D}_t$, $\mathcal{D}$ is* fully compliant*;*
- *otherwise, t is said to be partially-compliant meaning that some attributes are operating under* sub-ideal *conditions.*

Definition 4 ($\mathcal{P}_t$-Measure). *Given a task $t \in \mathcal{T}$, $\mathcal{D}_t = \{\mathcal{D}_t^1, \ldots, \mathcal{D}_t^n\}$; M_t^i the set of its attribute dimensions and dimension metrics as defined in Definition 1; and a dimension aggregation operator $\oplus$, then we define:*

$$\mathcal{P}_t = \bigoplus_{i \in [1, |\mathcal{D}_t|]} M_t^i$$

as the task compliance measure, or $\mathcal{P}_t$-Measure, of task t.

The dimension aggregation operator $\oplus$ here works much like the attribute aggregation operator $\odot$ in Definition 2. It aggregates dimension metrics that were calculated for each dimension and returns a single value that represents the overall level of task compliance. However, as discussed above, the aggregation function should be chosen with care.

Example 2. A loan application process consists of 5 activities from submit application to receive final decision. The standard amount of time for it is 15 days. If the threshold Δ is 5 days and it takes 18 days to finish the loan application process, then it is partially compliant, showing that even when some activity(ies) in the process instance may be non-compliant, the instance itself can be compliant.

Consequently, given an instance of trace $\mathfrak{T}$ of a BP, one can simply calculate the level of compliance of $\mathfrak{T}$ by directly aggregating/averaging the $\mathcal{P}_t$-Measure value of each task. However, this may have some drawbacks as the aggregated value may not necessarily reflect the real situation of the *whole* trace. This is due to the fact that the changes made after any non-compliance issues might introduce new attributes (and/or values), and changes to the task. Besides, during execution other tasks may also impact the value of the attribute, averaging these values might not give correct performance of the attribute, hence it would not make sense.

To overcome these issues, we define trace compliance and a trace compliance measure based on the attribute dimension metrics, as follow.

Definition 5 ($\mathfrak{T}$-compliance). *Given an instance of trace $\mathfrak{T}$ of a BP, we define:*

- $\mathfrak{T}$ *is* non-compliant *iff* $\exists t \in \mathcal{T}$, t *is* non-compliant*;*
- $\mathfrak{T}$ *is* fully compliant *iff* $\forall t \in \mathcal{T}$, t *is* fully compliant*;*
- *otherwise, $\mathfrak{T}$ is said to be partially-compliant meaning that $\mathfrak{T}$ has been executed under some* sub-ideal *(or* sub-optimal*) conditions.*

Definition 6 ($\mathcal{P}_{\mathfrak{T}}$-Measure). *Given an instance of trace $\mathfrak{T}$ of a BP; $\mathcal{T}$ the set of unique task identifiers of tasks; $\mathcal{D}_{\mathfrak{T}} = \{\mathcal{D}^1_{\mathfrak{T}}, \ldots, \mathcal{D}^n_{\mathfrak{T}}\}$; M^i_t the set of attribute compliance dimensions that appear in $\mathfrak{T}$; the aggregate compliance metric of task t as in Definition 1; and $\otimes$ the task dimension aggregation operator, then we define the trace partial compliance measure as:*

$$\mathcal{P}_{\mathfrak{T}} = \bigotimes_{\mathcal{D}^i_{\mathfrak{T}} | \mathcal{D}^i_{\mathfrak{T}} \in \mathcal{D}_{\mathfrak{T}}} \operatorname*{arg\,min}_{t | t \in \mathcal{T} \cap \mathcal{D}^i_{\mathfrak{T}} \in \mathcal{D}_t} (M^i_t > 0)$$

As execution progresses, the aggregate compliance metrics of each task will be updated accordingly. Hence, to reflect this situation, the compliance measure of a trace is defined by the aggregated dimension metrics (across all dimensions). Naturally, if all metrics of a particular dimension are 0 for a task, then a zero value will be returned. Note here that an instance of trace can be D-compliant on multiple dimensions, yet it does not mean that it will automatically be $\mathfrak{T}$-compliant at the end.

Lastly, we give the following definition for the overall compliance of a process log consisting of multiple traces to conclude our framework.

Definition 7 ($\mathcal{P}_P$-Measure). *Given a BP P; T_P the set of log trace instances obtained after executing P; and $|\mathrm{T}_P|$, its size, then the compliance measure for the process P is given by:*

$$\mathcal{P}_P = \frac{1}{|\mathrm{T}_P|} \Sigma_{\mathfrak{T} \in \mathrm{T}_P} \mathcal{P}_{\mathfrak{T}}$$

where $\mathcal{P}_{\mathfrak{T}}$ is the $\mathcal{P}_{\mathfrak{T}}$-Measure of the trace instance $\mathfrak{T}$.

Here, the compliance measure of a BP, $\mathcal{P}_P$-Measure, is defined as the average value of the $\mathcal{P}_{\mathfrak{T}}$-Measure across all traces since each trace represents an independent execution of P and will not affect other ones.

It is important to note that we have defined our metrics at three levels of aggregation in a hierarchical manner, i.e., at the task, trace and process log levels. Depending upon the user application and requirements, metrics at one or more levels can used in conjunction with each other to gain multiple perspectives. Besides, it is possible that a metric may be violated at one level but may still be satisfied at another or a higher level, or vice-versa. Moreover, some metrics along some dimensions like time may be more meaningful at the instance level as in Example 2 since the total instance duration is more important for the customer than the duration of individual tasks. Other metrics may be more relevant at the task level, such as the monetary amounts involved, etc. The process log level metrics can give insights into the overall compliance level for the entire log over a period of time, such as a week, month or quarter. Comparing such metrics across several successive periods can provide managerial insights into overall compliance trends.

4 Composite Measure Computations

Next, we discuss some scenarios in the context of a real-world example to illustrate how the proposed framework can be applied in practice to compute different levels of compliance by employing the *averaging method* discussed in the previous section. For this purpose, we consider the invoice payment example from Figure 1, and provide some notation for our computations. We consider the attribute aggregation operator $\odot$ to be the average of all attribute values projected in the dimension, i.e., $M_t^i = \frac{1}{|\mathcal{D}_t^i|}\Sigma\psi_t(a)$. Moreover, $\oplus$ is the compliance dimension aggregation operator averaging the dimension index values for each dimension in the task i.e., $M_t^i = \frac{1}{|\mathcal{D}_t^i|}\Sigma\psi_t(a)$, and $\otimes$ is the minimum of all values for each dimension.

Table 1 illustrates the attributes and their possible values in the context of Figure 1. Attributes such as *description*, $invoiceValue$ and $invoiceDate$ are meta information of the invoice and do not contribute to the compliance metrics. The attributes $equipmentDeliveryDays$ and $payInDays$ denote the number of days required to deliver the equipment(s) to the purchaser and the number of days within which *full* payment must be made after the invoice is issued, respectively. As shown, different values for these parameters are mapped to compliance levels based on the ψ projection function. Moreover, in this scenario, the partial compliance cut-off value S and the threshold Δ are set to 0.3 and 0.4, respectively. Thus, a compliance value between 0.3 and 0.7 $(0.3 + 0.4)$ is considered as *partially compliant*, and any value below 0.3 as *non-compliant*. Similarly, the attribute *paymentReceived* is the amount paid by the customer, which includes the principal plus any applicable interest and penalty. Notice from the table that a payment of less than half of the amount due is deemed as non-compliant, while other values of payment are considered as partially compliant. The two attributes, *interest* and *penalty* are meta information that will be used to calculate the penalty when violations occur.

Full Compliance: Consider a scenario where $equipmentDeliveryDays =$ 2 days, $payInDays = 10$ days and a payment of \$500 has been received from the purchaser, i.e., the equipment has been delivered and full payment has been received within the prescribed time frame. Hence, as an example, consider the trace $\mathfrak{T}_4 = \langle t_1, t_2, t_6 \rangle$ which contains the attributes $\langle paymentReceived, payInDays, equipmentDeliveryDays \rangle$, as illustrated in Table 1. To compute the aggregate metric across the compliance dimensions for an attribute, we first compute the individual compliance values along each dimension and then aggregate them. The compliance metric for the

Table 1: Attributes metric of business process in Figure 1

<table>
<tr><th>Attribute name (a)</th><th>Dimension</th><th>$\mathcal{A}_n$</th><th>$\psi(a)$</th><th>Cut-off (S)</th><th>Threshold (Δ)</th></tr>
<tr><td>$invoiceValue$ (P)</td><td>Monetary</td><td>$500</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>$invoiceDate$</td><td>Temporal</td><td>2019-04-01</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>$equipmentDeliveryDays$</td><td>Temporal</td><td>≤3 days
≤7 days
>7 days</td><td>1
0.5
0</td><td rowspan="2">0.3</td><td rowspan="2">0.4</td></tr>
<tr><td>$payInDays$</td><td>Temporal</td><td>≤15 days
≤22 days
≤32 days
>32 days</td><td>1
0.6
0.3
0</td></tr>
<tr><td>$interest$ (Int)</td><td>Percentage</td><td>0 %
3 %</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>$penalty$ (Pen)</td><td>Percentage</td><td>0
2.5 %</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>$paymentReceived$
($R = P + Int + Pen$)</td><td>Monetary</td><td>$< 50\,\% \times R$
$< 75\,\% \times R$
$< 80\,\% \times R$
$< R$
≥R</td><td>0
0.3
0.5
0.9
1</td><td>0.3</td><td>0.7</td></tr>
</table>

monetary (M) and temporal (T) dimensions for task t_2 are first computed as:

$$\begin{aligned} M^M_{t_2} &= \frac{1}{|\mathcal{D}^M_{t_2}|}\Sigma_{a\in\mathcal{D}^M_{t_2}}\psi_{t_2}(a) \\ &= \frac{1}{|\mathcal{D}^M_{t_2}|}(\psi_{t_2}(paymentReceived)) \\ &= \frac{1}{1}(1) \\ &= 1 \end{aligned}$$

$$\begin{aligned} M^T_{t_2} &= \frac{1}{|\mathcal{D}^T_{t_2}|}\Sigma_{a\in\mathcal{D}^T_{t_2}}\psi_{t_2}(a) \\ &= \frac{1}{|\mathcal{D}^T_{t_2}|}(\psi_{t_2}(payInDays)) \\ &= \frac{1}{1}(1) \\ &= 1 \end{aligned}$$

Further, by Definition 4, we have:

$$\begin{aligned} \mathcal{P}_{t_2}\text{-Measure} &= \frac{1}{|\mathcal{D}_{t_2}|}\Sigma_{i\in\mathcal{D}_{t_2}}M^i_{t_2} \\ &= \frac{1}{|\mathcal{D}_{t_2}|}(M^T_{t_2} + M^M_{t_2}) = \frac{1}{2}(1+1) \\ &= 1 \end{aligned}$$

Similarly, for task t_6, we have: $M^T_{t_6} = 1$ and $\mathcal{P}_{t_6}$-Measure $= 1$.

Hence, we have: $\mathcal{D}_{\mathfrak{I}_1} = \{\mathcal{D}^M_{\mathfrak{I}_1}, \mathcal{D}^T_{\mathfrak{I}_1}\}$, and $\arg\min(\mathcal{D}^M_{\mathfrak{I}_1}) = 1$, and $\arg\min(\mathcal{D}^T_{\mathfrak{I}_1}) = 1$.

Consequently, it follows that: $\mathcal{P}_{\mathfrak{I}_1}$-Measure $= \frac{1}{2}(1+1) = 1$.

Partial Compliance: Let us now turn to consider a different scenario where:
$equipmentDeliveryDays = 2$ days,

$payInDays = 20$ days
$paymentReceived = \$575$
In this scenario, the payable amount is now $\$500 + \$75 = \$575$, and has been fully paid by the customer.Thus, we can calculate the aggregate partial compliance measures for the tasks as follows:

$$\begin{aligned} interest &= (20 - 15) \times 0.03\,\% \times \$500 \\ &= \$75 \end{aligned}$$

and $penalty = 0$.
Accordingly, we have the following:

Dimensions	Attributes	t_2	t_3	t_6
Temporal	$equipmentDeliveryDays$	–	–	1
	$payInDays$	0	0.6	–
Monetary	$paymentReceived$	0	1	–
	$\mathcal{P}_t$-Measure	0	0.8	1

Hence, $\arg\min(\mathcal{D}^M_{\mathfrak{T}}) = 0.6$, and $\arg\min(\mathcal{D}^T_{\mathfrak{T}}) = 1$.
Therefore, $\mathcal{P}_{\mathfrak{T}}$-Measure $= \frac{1}{2}(0.6 + 1) = 0.8$.

Non-Compliance Lastly, consider the situation where no payment has been received after 32 days, i.e., the conditions of the contract have been violated and cannot be repaired. Thus, the contract will be deemed as terminated.

Dimensions	Attributes	t_2	t_3	t_4	t_6
Temporal	$equipmentDeliveryDays$	–	–	–	0
	$payInDays$	0	0	0	–
Monetary	$paymentReceived$	0	0	0	–
	$\mathcal{P}_t$-Measure	0	0	0	0

$\mathcal{P}_{\mathfrak{T}}$-Measure $= \frac{1}{2}(0 + 0) = 0$.
The partial compliance functions can also be introduced as mappings from a numeric domain of attribute values corresponding to the threshold window around the standard value for an attribute to the $[0, 1]$ range. These functions typically take a linear, concave or convex form depending upon how rapidly the distance from the standard value affects compliance. The shapes of typical functions have to be determined through empirical studies and this is out of the scope of the current work.

5 Discussion and related work

There are different ways to overcome various partial compliance scenarios as shown in Figure 4. For each kind of deviation or case of partial compliance,

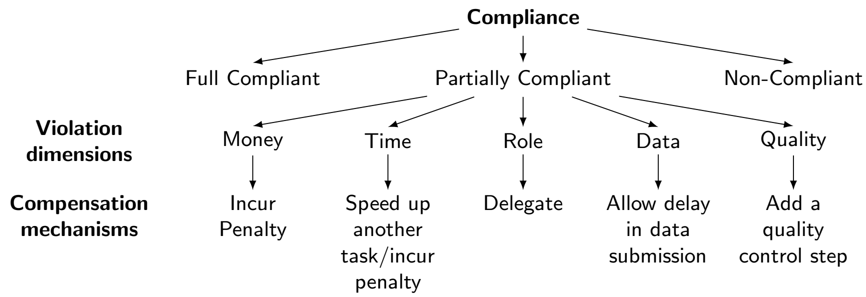

Fig. 4: Compliance dimensions and compensation mechanisms for partial compliance

one or more compensatory mechanisms may be provided for the task to resume execution.

For example, if a task is delayed it may be made up by speeding up a later task so that the customer of a service does not notice any increase in the total time for a process instance. A role violation occurs when an employee in the designated role is not available to perform a task. In such a case, a possible compensation is to assign the task to a delegate of the person who would normally perform it. For the data dimension, to process a passport application a user may be required to provide social security card or ID card and birth certificate, etc. If the user does not provide the birth certificate (or, say, one of three required documents), it may still be possible to process the application provided the missing document is submitted within one week of the application. Thus, the application may still be processed despite a minor violation. In the absence of such a mechanism, the application would have to be rejected, and then have to be resubmitted thus increasing the overall cost of processing it both for the citizen and for the governmental agency involved.

On the other hand, if the process instance itself takes longer than the standard time, then the customer has to be compensated by the service provider as per their agreement. With regard to the process model redo, restart, undo or abort the tasks can be other possible ways to overcome partial compliance issues. As they have their own complexities and impact on the execution of the individual tasks and the process on the whole, these are the topics of our further investigation.

Our framework determines partial compliance in a bottom up manner from the individual task level, to the trace level, and then a compliance score can thus be assigned to each trace. It is also possible to consider partial compliance at a still higher level of all process instances or cases in a day, week or month. The notion here would be to determine how many traces fall within a certain partial compliance level, say X% of instances have a compliance level more than 0.8 during a month. Moreover, a similar analysis may be done at the individual task level to determine what percentage of payment tasks had a compliance level of more than, say, 0.9 in a month. Such information can be very helpful to the management of a company. As we noted above, binary notions of compliance are not very useful from a management perspective for understanding the operational performance of a company. By introducing partial compliance in this way management can gain deeper insights into their operations. The values in Table 1 can be derived through empirical studies and from analysis of logs from previous executions of the business process model.

The problem of partial compliance has been studied widely in different domains. Gerber and Green [2] proposed the use of regression analysis scalable protocols to resolve some of the partial compliance issues that appear in field experiments. Jin and Rubin [7], on the other hand, proposed the use of principal stratification to handle partial compliance issues when analysing drug trials and educational testing. However, only a limited amount of work has targeted the area of improving business process compliance (BPC) management or measuring the level of compliance of a BP in a quantitative way. In the followings, we present some pertinent studies and discuss their strengths and limitations for the measurement of partial compliance.

In [13], Sadiq *et al.* introduced the notion of compliance distance as a quantitative measure of how much a process model may have to change in response to a set of rules (compliance objectives) at design time; or by counting the number of recoverable violations, how much an instance deviates from its expected behavior at runtime. This approach is extended in [9] to effectively measure the distance between compliance rules and organization's processes. To this purpose, the authors have divided the control objectives into four distinct classes of ideal semantics, namely: (i) *ideal*, (ii) *sub-ideal*, (iii) *non-compliant*, and (iv) *irrelevant*, and compute the degree to which a BP supports the compliance rules. Although their method provides computationally efficient means to analyze the relationships between the compliance rules and BPs, the heavily formalized rules have increased the complexity of the modelling process which is a potential obstacle to non-technical users.

Shamsaei [15] proposed a goal-oriented, model-based framework for measuring the level of compliance of a BP against regulations. In the paper, the author decomposed the regulations into different control rule levels, and then defined a set of key performance indicators (KPIs) and attributes for each rule to measure their level of compliance. The value of the KPIs can be provided either manually or from external data sources, and the satisfaction level of each rule is evaluated on a scale between 0 and 100 by considering the values of target, threshold, and worst, so that analysts can prioritize compliance issues to address suitably given the limited resources at their disposal.

Morrison *et al.* [12] proposed a generic compliance framework to measure the levels of compliance using *constraint semiring* (c-semiring) [1]. In their approach, imprecise or non-crisp compliance requirements will first be quantified by means of *decision lattices* (through the notion of *lattice chain*), which provides a formal setting to represent concept hierarchies and values preferences [6], such as $\{Good$, Ok, $Bad\}$ or $\{Good, Fair, Bad\}$. These values will then be combined and utilized as a decision-making tool by c-semirings to rank the level of compliance of the BPs. Essentially, the advantage of their framework lies in the ability in combining compliance assessments on various dimensions. Although the proposed approach is general enough to provide an abstract valuation at policy (business process in this case) level, the information about compliance at lower levels of abstraction is missing.

Kumar and Barton [8] discussed an approach for checking temporal compliance. They used a mathematical optimization model to check for violations. After a violation occurs it can also check whether the remaining process instance can be completed without further violations and determine the best way to do so. In this way, the level of compliance along the time dimensions can be managed.

6 Conclusions and Future Work

Compliance to policies, rules and regulations is usually treated in a rigid manner in business, government, and other kinds of organizations. Compliance pertains to matters that affect employees, customers, and just ordinary citizens. Rigid compliance means that either there is strict adherence to a rule or policy by an entity in which case the entity is compliant, else the entity is treated as being non-compliant or in violation of the rule or policy. In the real-world, however, such a binary approach is not very efficient because violations related to processing of applications, permits, invoices, fines, taxes, etc. may occur along a continuous spectrum and even minor violations may lead to cancellation of transactions or processes. Hence, it is important to recognize the extent of the violation and also allow for remediation or compensation mechanisms for them that are commensurate with the degree of the violation. This would enhance overall social value by reducing inefficiencies and cutting down wasteful work performed in the system.

In this paper, we propose notions of full-, partial- and non-compliance to describe the compliance levels of a business process during its execution, and, based on the information available on different compliance dimensions for the attributes of a task, we have proposed a metrics-based framework that can be used to measure the level of compliance and provide more information on the state of a BP instance during execution and auditing phases. The framework was developed from basic principles of partial compliance.

To realize the effectiveness of the proposed framework, from an implementation perspective, we are planning to implement it as a ProM Plugin[6] such that, given a process log, the application can automatically perform a compliance evaluation and analysis on the log, and generate a full report that shows compliance at multiple levels of aggregation. We would also like to test, validate and fine tune this tool by applying it to real-world logs, and seeking feedback from the domain experts about the perceived value they obtain from such an analysis. Further, it would be useful to extend the notions of compensation more formally.

[6] ProM Tools: `http://www.promtools.org/doku.php`

References

1. Bistarelli, S.: Semirings for Soft Constraint Solving and Programming. Springer Berlin Heidelberg (2004). https://doi.org/10.1007/978-3-540-25925-1
2. Gerber, A.S., Green, D.P.: Field Experiments: Design, Analysis, and Interpretation. W. W. Norton & Company (2012)
3. Governatori, G.: Business Process Compliance: An Abstract Normative Framework. it – Information Technology **55**(6), 231–238 (2013)
4. Hashmi, M.: A Methodology for Extracting Legal Norms from Regulatory Documents. In: 2015 IEEE 19th International Enterprise Distributed Object Computing Workshop. pp. 41–50. EDOCW 2015, IEEE, Adelaide, SA, Australia (Sep 2015)
5. Hashmi, M., Governatori, G., Wynn, M.T.: Normative requirements for regulatory compliance: An abstract formal framework. Information Systems Frontiers **18**(3), 429–455 (Jun 2016)
6. Huchard, M., Hacene, M.R., Roume, C., Valtchev, P.: Relational concept discovery in structured datasets. Annals of Mathematics and Artificial Intelligence **49**(1), 39–76 (Apr 2007)
7. Jin, H., Rubin, D.B.: Principal Stratification for Causal Inference With Extended Partial Compliance. Journal of the American Statistical Association **103**(481), 101–111 (2008). https://doi.org/10.1198/016214507000000347
8. Kumar, A., Barton, R.R.: Controlled violation of temporal process constraints – Models, algorithms and results. Information Systems **64**, 410 – 424 (2017)
9. Lu, R., Sadiq, S., Governatori, G.: Measurement of Compliance Distance in Business Processes. Information Systems Management **25**(4), 344–355 (2008)
10. Ly, L.T., Rinderle-Ma, S., Göser, K., Dadam, P.: On enabling integrated process compliance with semantic constraints in process management systems. Information Systems Frontiers **14**(2), 195–219 (Apr 2012)
11. Maggi, F.M., Montali, M., Westergaard, M., van der Aalst, W.M.P.: Monitoring Business Constraints with Linear Temporal Logic: An Approach Based on Colored Automata. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) Proceedings of the 9th International Conference on Business Process Management. pp. 132–147. BPM 2011, Springer Berlin Heidelberg, Clermont-Ferrand, France (Aug 2011)
12. Morrison, E., Ghose, A., Koliadis, G.: Dealing with imprecise compliance requirements. In: Proceedings of the 13th Enterprise Distributed Object Computing Conference Workshops. pp. 6–14. IEEE, Auckland, New Zealand (Sep 2009)
13. Sadiq, S., Governatori, G., Namiri, K.: Modeling Control Objectives for Business Process Compliance. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) Proceedings of the 5th International Conference on Business Process Management. pp. 149–164. BPM 2007, Springer Berlin Heidelberg, Brisbane, Australia (Sep 2007)
14. Schumm, D., Turetken, O., Kokash, N., Elgammal, A., Leymann, F., van den Heuvel, W.J.: Business Process Compliance through Reusable Units of Compliant Processes. In: Daniel, F., Facca, F.M. (eds.) Proceedings of the 10th International Conference on Current Trends in Web Engineering. pp. 325–337. ICWE 2010, Springer Berlin Heidelberg, Vienna, Austria (Jul 2010)
15. Shamsaei, A.: Indicator-based Policy Compliance of Business Processes. In: Boldyreff, C., Islam, S., Leonard, M., Thalheim, B. (eds.) Proceedings of the CAiSE Doctoral Consortium 2011. pp. 3–14. London, UK (Jun 2011)