

2009

Dealing With Imprecise Compliance Requirements

Evan D. Morrison
University of Wollongong

Aditya K. Ghose
University of Wollongong, aditya@uow.edu.au

George Koliadis
University of Wollongong

Publication Details

This conference paper was originally published as E. Morrison, A. K. Ghose and G. Koliadis. Dealing With Imprecise Compliance Requirements. In Proc. of the 2nd International Workshop on Dynamic and Declarative Business Processes (DDBP 2009), IEEE Computer Society Press, 2009.

Dealing With Imprecise Compliance Requirements

Abstract

Business process compliance management is a field of study involving the co-ordination of business process management and compliance systems. A compliance system is an organisation wide tool that links legislative and business rules to organization policies and processes. The objective of such a system is to promote a self sustaining level of operations that minimizes the losses caused to the business through breaches of laws or internal misappropriations. We view a compliance system in a similar fashion to that of an accounting system where each process is treated as a transaction. Each process may be monitored and valuations of costing and benefits associated to each task. Both high order policy creation as well as low order transactional histories of single processes must be considered to obtain a complete picture of current operations. In this paper we discuss benefits and shortcomings in some of the currently implemented compliance schemes and present a method for measuring the degree of compliance that each business process may achieve.

Disciplines

Physical Sciences and Mathematics

Publication Details

This conference paper was originally published as E. Morrison, A. K. Ghose and G. Koliadis. Dealing With Imprecise Compliance Requirements. In Proc. of the 2nd International Workshop on Dynamic and Declarative Business Processes (DDBP 2009), IEEE Computer Society Press, 2009.

Dealing With Imprecise Compliance Requirements

Evan Morrison, Aditya Ghose, George Koliadis
Decision Systems Laboratory,
University of Wollongong
{edm92,aditya,gk56}@uow.edu.au

Abstract—Business process compliance management is a field of study involving the co-ordination of business process management and compliance systems. A compliance system is an organisation wide tool that links legislative and business rules to organization policies and processes. The objective of such a system is to promote a self sustaining level of operations that minimizes the losses caused to the business through breaches of laws or internal misappropriations. We view a compliance system in a similar fashion to that of an accounting system where each process is treated as a transaction. Each process may be monitored and valuations of costing and benefits associated to each task. Both high order policy creation as well as low order transactional histories of single processes must be considered to obtain a complete picture of current operations. In this paper we discuss benefits and shortcomings in some of the currently implemented compliance schemes and present a method for measuring the degree of compliance that each business process may achieve.

Keywords-Compliance, Business Process Management, c-semirings

I. INTRODUCTION

A compliance system is an organisation wide tool that links legislative and business rules to organization policies and processes. The objective of such a system is to promote a self sustaining level of operations that minimizes the losses caused to the business through breaches of laws or internal misappropriations [1–4]. .

We have reviewed a number of recommendations for best-practices and measures that have been proposed and implemented in various industry sectors [5–9], focusing on the fulfillment of legislative objectives for laws such as the Sarbanes Oxley Act(SOX) [10], the Corporate Law Economic Reform Program [11] (amending the Australian Corporations Act), and more recently the Anti-Money Laundering laws [12] implemented in a number of countries. A consequence of the introduction of some of these laws (in particular SOX) there is a legislative need for organizations to implement internal control measures (Section 404 of SOX).

In this review of current IT management literature, we have identified a number of frameworks that offer an extended level of compliance monitoring to that of a traditional accounting approach to compliance management. The strong point of many of these frameworks is in the segregation of activities as a fraud prevention tool. The pain point

of these frameworks is that they are semiformal and lack rigorous feedback devices; a great deal of reliance is placed on an organization doing the right thing and estimating their level of compliant operations. The implementation of a formal system that aids in the measurement of the degree of compliance of an organization will help in identifying weak and strong strategies and their implementing processes, developed to meet the requirements of compliance frameworks.

A business process is a series of activities created to fulfill the daily operations of an organization. Each time a business process is executed it may vary depending on the variables associated with the execution instance. Business processes are complex adaptive systems that have rules placed on them. Each business input should trigger the execution of a new process execution instance. In the execution of a new process instance as resulting effect will occur and the instance can be measured for compliance. The execution instance will have a preference evaluation based on its execution.

A compliance system is a tool that can be used to identify compliance requirements and then map them against the business process execution instances. A compliance system may then be used to measure the ‘degree’ of compliance for a process execution.

It is clear that a fundamental basis for assessing effective compliance as a complete system is needed, rather than creation of independent monitoring sets and varying policies recommendations for each new law that is enacted. We have developed a general framework for compliance management using constraint semirings(c-semirings) [13] to aid in the dissemination of the compliance in operation for processes. We have also used decision lattices [14–16] to aid in the creation of viable process compliance rankings.

Using the combination of c-semirings and lattices we aim to answer the following questions, if there is a case that a company must comply with legislation, how does it achieve compliance at a minimal cost for maximum benefit? Is there a method to implement a broad measurement device for satisfying compliance requirements?

This paper is broken down in the following manner. Section 2 provides an introduction to the current state of the art in compliance and a breakdown of the components that make up a compliance framework. In section 3 we show how processes and compliance preferences may be constructed

within an algebraic structure for analysis. Section 4 gives an overview of a preference valuation system. In section 5 we introduce a notion of non-compliant process repair based on work in section 3 and section 4. Section 6 is the conclusion of this paper and hints at further research questions in the domain.

II. BACKGROUND

There are a number of compliance methodologies and best-practice frameworks that apply to specific areas in industry. These methodologies attempt to define procedures that can be used to model and meet compliance requirements of legislation. All of these methodologies attempt to achieve the same result of helping a business create a compliant operation level.

Balanced scorecards [9] are an assessment device. A business ranks each process it completes on a matrix to find roughly where it stands compared to the overall business strategy.

Standardized policy creation [2,6–8,17] is important in defining compliant operational procedures. This is the creation and adherence of ‘best-practices’ and standards [3,17] involving regular revision of policies and continuous feedback to standards committees. We have reviewed some of the more popular frameworks including COSO, COBIT, and ISO17799/27002. The COSO framework defines a number of internal controls, standards and criteria against which companies and organisations can measure themselves. The approach involves the development of a control framework for IT systems, recognizing that IT systems are now fundamental for the success of a company. COSO is broken into five components that summarize best-practices for: A Control Environment, Risk Assessment, Control Activities, Information and Communication, and Monitoring [7,17]. Cobit is a framework that has been designed to strengthen the relationship between financial reporting, IT functions, and security. It bridges the gaps between business risks, control needs and technical issues. The Cobit framework is said to be Sarbanes Oxley Compliant. This is achieved through: planning and organizing, acquiring and implementing, delivering and supporting, monitoring, management guidelines, and maturity models [8]. In the various implementations of COBIT there are both a balanced scorecard(COBIT3.1) and a methodology for general compliance management(COBIT4.1). The ISO 17799/27002 standard for risk based compliance management has a similar structure to achieving compliance. Control, monitoring, risk analysis, support and feedback form the foundation of this compliance system [2,6]. The pain point of many of these frameworks is that during implementation continuous business operations are not considered. These frameworks are very broad and do well in providing an overview of an organizations strategic level of compliance. There is a contention that these frameworks may provide a false

security to organizations that believe these frameworks are completely accurate [18]. This weakness stems from the lack of a formal analysis tool. These frameworks tend to focus directly on specific legislation. It is due to this we believe the market fear of increased legislations is well founded [19,20].

An alternate to these frameworks is the divide and conquer based hierarchical role breakdown within an organisation that aids in the enforcement of desirable compliance traits. Systems are beginning to adopt the idea of divide and conquer in regard to organisational process definitions [5]. In this work each organizational unit is refined to the atomic processes and input, processing and output are checked for compliance to organizational policies. Similar work has been conducted by Ghose and Koliadis [21] in the atomic breakdown of processes for compliance management. In their work various process activities can be combined and checked for inconsistencies in process design.

In [22] Rozinat and Aalst have investigated the conformance of runtime transactions against the designs of the process to be checked. In this work Rozinat et. al. consider both the process sequence fit as well as the structural fit of transaction instances to their design counterparts using metrics. Aalst has continued this work in [23] through definitions of precise translation devices from SOAP messages overlaying a formal Petri-nets to ensure conformance. Similar work has been conducted in [24] through semantic and structural comparisons between BPMN models.

A. Algebraic Frameworks

Algebraic Frameworks have been defined to act as a translation and reasoning devices for activity level compliance checking against contractual requirements as in [4,25]. Work has been completed in the formalization of contract languages such as FCL [25–27].

Governatori et al [25,26] provide a reparation chain mechanism for representing statements such as condition C generates an obligation O_1 , failing which a reparation obligation O_2 is generated, failing which a reparation obligation O_3 is generated etc. Clearly a reparation chain can be viewed as an elaboration of an imprecise compliance requirement using a linearly ordered preference structure. The focus of our work is in assessing degree of compliance of process instances with imprecise compliance requirements that have not been elaborated. The use of our technique leads to an incremental elaboration of an imprecise compliance requirement, via the mapping of process instance to a partially-ordered preference structure.

III. RUN TIME COMPLIANCE

In this section, we shall define a means for auditing and measuring levels of compliance. Effectively, we wish to understand the *degree* to which the processes of an organization complied with the applicable set of compliance

requirements, over a given audit period. We propose to use a simple algebraic mechanism for specifying degrees of compliance - the framework of c-semirings. This framework is particularly useful because it permits us to combine assessments on multiple dimensions and on a mixture of qualitative and quantitative rules. C-semirings have been used to formalize soft constraint problems [28] and [29] where different tuples in a constraint satisfy the constraint to varying degrees.

Definition 1: (c-semiring) [13]

A c-semiring is a tuple $\langle A, +, \times, 0, 1 \rangle$ such that:

- (i) A is a set and $0, 1 \in A$
- (ii) $+$ is called the comparison operation.
- (iii) \times is called the combination operation.
- (iv) \times distributes over $+$ (i.e. $a \times (b + c) = ab + ac$)
- (v) Each c-semiring has a partial order \leq_s over the set of values A where $a \leq_s b$ implies $a + b = b$ in c-semiring S .

As a c-semiring contains all value combinations of a compliance system, all combinations are ordered specifically based on the comparison operator. This is a trait of c-semirings that allows us to rank and rate various levels of compliance. This ranking is a partial order of the set of values.

We consider a compliance requirements as a function: $R: 2^{\mathcal{L}} \rightarrow A$ where $2^{\mathcal{L}}$ is the set of all well formed sentences in the underlying language in which process instances are described. \mathcal{L} is the language in which process instances are described. A is the set of preference values associated to the process. R is a compliance preference measurement function. Each compliance function maps requirements into c-semiring values.

A. Degree Measurement Framework

Within the c-semiring framework [13] it is possible to substitute any c-semiring instance into a degree measurement framework. For example a Boolean c-semiring can be used to show how it is possible to model simple business process statements. A process can be given a preference of either good (G) or bad (B); this preference is based on whether the process was completed. For example, given the two processes:

“Creation of a new user on the DBMS” and “A quarterly activity report is filed”.

A representation is shown as follows.

Example 1: (c-semiring example)

Consider the following instance of a c-semiring:

$\langle \{G, B\}, \vee, \wedge, B, G \rangle$ where (G) is a good process and (B) is a bad process and the operators (\vee/\wedge) are representative of logical or/and functions. So that elements being compared are either (G)ood or (B)ad, and when elements are combined they form tuples $\langle (G)ood, (G)ood \rangle$ which is (G)ood and (G)ood.

We can represent various problems where the instance of a business process may perform well (G) or badly (B) using compliance requirements preferences.

Following on, we create an application framework to cover business run time compliance by grouping multiple processes and their activities that can be tailored to a specific application area.

Definition 2: (Instance Measure)

For a process P , with a set of instances $\{p_1, \dots, p_n\}$ over a compliance audit period (this could be the full set of instances over the audit period, or a random sample), the *degree of compliance with R* , denoted by $C_R(P)$ where R is a compliance requirement, is given by $R(p_1) \times R(p_2) \times \dots \times R(p_n)$.

Using a process of “Creation of a new user on the DBMS” as an example.

Example 2: If this process is conducted a number of times in an audit period then a set of instance transaction logs could be as follows:

New username:John, New password: secret, (Saved)

New username:Andrew, New password: password, (Saved)

New username:John, New password: fido, (Not saved)

A compliance requirement function (R) could return the assessment of (G)ood, (G)ood, (B)ad for each log entry retrospectively.

The set of instances have a *degree of compliance with R* that is:

$$C_R(P) = G \times G \times B = G \wedge G \wedge B = B$$

Given this (B)ad *degree of compliance with R* an compliance analyst may quickly find non-compliant process instances and fixes to the system can be made.

It can be shown that various requirements can act on a single process. If we consider the process statement from above as a single process then the requirements on each activity completed for that process are the set of requirements for a single process.

The degree of compliance of a single process with all its requirements can be formulated as follows:

Definition 3: (Requirement Measure)

For a process P , and a set of compliance requirements $RS = \{r_1, r_2, \dots, r_n\}$, the *degree of compliance of P with RS* , denoted by $C_{RS}^{\Sigma}(P)$, is given by $C_{r_1}(P) \times C_{r_2}(P) \times \dots \times C_{r_n}(P)$.

This degree is shown by saying

$$C_{RS}^{\Sigma}(P) = C_{r_1}(P) \times C_{r_2}(P) \times \dots \times C_{r_n}(P)$$

The degree of compliance on a single process with all activity requirements is a combination of the requirements. If one requirement contradicts another activities requirements then there is a problem with the process definition or we will find an error in the execution.

Next we show the overall measure of compliance by combining multiple requirements onto multiple process instances. This is done by combining all process instances with all requirements acting on each instance.

Definition 4: (Compliance Measure)

For a set of processes $PS = \{P_1, P_2, \dots, P_n\}$ and a set of compliance requirements RS, the compliance measure of PS with respect to RS, denoted by $CM_{RS}(PS)$, is given by $C_{RS}^\Sigma(P_1) \times C_{RS}^\Sigma(P_2) \times \dots \times C_{RS}^\Sigma(P_n)$

For example, taking both process definitions, the activities involved in each, the requirements and instances of running we have:

Degree of Compliance for P1 “Creation of a new user on the DBMS” = Requirement(A valid username should be chosen) \times Requirement(A ‘good’ password must be chosen) \times Requirement(The user and password must be saved)

P1.I1 : New username:John, New password: secret, (Saved)

P1.I2 : New username:Andrew, New password: password, (Saved)

P1.I3 : New username:John, New password: fido, (Not saved)

Degree of Compliance for P2 “A quarterly activity report is to be filed” = Requirement(an activity report must be made) \times Requirement(A quarterly activity report must be printed and mailed to the regulatory body) \times Requirement(complete in reasonable time)

P2.I1 : New report written on the 01-01-2008, Printed and Mailed on 15-01-2008, Reasonable time.

P2.I2 : New report written on the 01-03-2008, Printed and Mailed on 10-03-2008, Reasonable time.

P2.I3 : New report written on the 01-06-2008, Not printed, Unsatisfactory time

Here the Compliance Measure becomes

$$CM_{RS}(PS) = ((P1.I1 \times P1.I2 \times P1.I3) \times (P2.I1 \times P2.I2 \times P2.I3))$$

If we use a rating system of good and bad from before then we find that $CM_{RS}(PS) = (B)ad$ as there are errors in P1.I3 and P2.I3.

Next we will show how when we combine instance measures with compliance measure it is possible to gain a full picture of compliance operations.

An important property of c-semirings is that they can be combined into an aggregate structure. In the following example we show how a business process requirement of performance can be combined with a requirement of completion. In the example we have a compliance requirement that values a process on a good/bad scale using a c-semiring. We also have a fuzzy c-semiring that values a process instance on completion.

So if the first process of creating a user on a system is based on the scale of good and bad and the second process of filing a quarterly activity report is based on a fuzzy scale of preference for completion such that activity reports filed faster are better then:

Example 3: (Combination example)

Given a binary c-semiring S and a fuzzy c-semiring T;

$$S = \langle \{B, G\}, \vee, \wedge, B, G \rangle \text{ and}$$

$$T = \langle \{0, .5, 1 \in [0, 1]\}, max, min, 0, 1 \rangle$$

The combination of S and T is:

$$\langle G, 0 \rangle; \langle G, .5 \rangle; \langle G, 1 \rangle; \langle B, 0 \rangle; \langle B, .5 \rangle; \langle B, 1 \rangle$$

Which is showing an ordered relationship between good

processes and completion levels. The tuple $\langle B, 0 \rangle$, is the worst outcome where the user creation is performed badly and quarterly activity statements are not completed. The top right value $\langle G, 1 \rangle$, is the best value tuple where users are created on the system well and the quarterly activity statement is filed within a ‘reasonable amount of time’. The tuple $\langle G, .5 \rangle$, shows a mid level completion of a compliance objective, the user is created in the system in an amount of time that longer than ‘reasonable’.

Using this tuple as a scale we can begin to compare process instance measures to see how well a process is doing in contrast to possible outcomes.

To this point we have worked with both crisp Boolean requirements and imprecise requirements. In the next section we will provide a method for determining values for imprecise compliance requirements using a decision lattice. Decision lattices have been used as the values obtained can be implemented in a c-semiring.

IV. DEGREE OF COMPLIANCE

Compliance requirements and rules can be broken into contractual rules [30] that can be accumulated across business process models [27]. In this section we will define a spectrum of crisp and imprecise compliance requirements.

We need to be able to determine, for each process instance, and for each compliance requirement, the “*degree of compliance*” of that instance with that requirement. For crisp compliance requirements, the assessment of degree of compliance is Boolean, i.e. a process instance either does or doesn’t comply. For more imprecise or vague compliance requirements, the assessment involves greater complexity. Consider a compliance requirement that states: “*quarterly activity statements must be filed within a reasonable time frame*”. Clearly an activity statement filed immediately after the end of a quarter satisfies the requirement entirely, while one that is never filed violates it entirely [31,32]. A statement filed 10 weeks after the end of the quarter satisfies the requirement partially. A statement filed 12 weeks after the end of the quarter also satisfies the requirement partially, but to a lesser degree than the statement filed 10 weeks after the end of the quarter. A mechanism for assessing degrees of compliance that sit between the two extremes of full and partial compliance is therefore required.

A. Identifying Imprecise Compliance Requirements

When we obtain various departmental policies, such as “Secure the DBMS from well-known attacks” and “Ensure that financial reports are signed in triplicate”, we are faced with the problem of how does each policy compare to the other?

There is existing work that seeks to monetize “prescriptive policies” [33], i.e., attach monetary penalties for non-compliance. In principle, this could be extended to deal with our problem, by associating differential monetary penalties

to varying degrees of non-compliance. Unfortunately monetary penalties alone may not be sufficient to completely describe a policy that has been defined to meet objectives such as “Increase customer satisfaction” or “Reduce environmental impact”. We suggest that a monetary valuation system can be tricky to negotiate and estimates of projected growth of trading can be *manufactured to sway* audit systems. This problem has been addressed in [32] briefly, with no immediate solution given to the above mentioned monetary valuations.

Our framework is general enough to aid in providing an abstract valuation for the completion of activities that may be combined to produce a policy level compliance value. This is done through use of decision lattices [14–16]. The benefit of using these tools is that they may be combined with the use of c-semirings [13].

A non-crisp compliance requirement is a process requirement that has multiple acceptance criteria that are difficult to *evaluate* with a simple monetary values. As an example, in the process of ‘processing a form’ with a non-crisp requirement that the ‘form must be processed within a *reasonable amount of time*’, each person facing the problem may offer conflicting answers. The person who sent the form may consider a reasonable amount of time to be ‘2 days’ as they require the resulting processed form to complete other business processes. The person processing the form may consider a reasonable amount of time to be ‘4 weeks’ as the load of incoming is increasing and it is not possible to process every form in such a short period of time. The marketing team behind the processing team may believe that a reasonable amount of time is ‘30 minutes’. It is due to the fuzzy nature of *imprecise* requirement results that these types of compliance requirements may have different values depending on the *instance* of completion [33].

If we consider that ‘Form processing in under 4 weeks’ to be a satisfactory achievement of compliance, and that ‘Form processing in under 2 weeks’ is a good achievement, and ‘Form processing in under 30minutes’ to be a great achievement we can start to categorize various processes with a *degree of compliance*. We call the *degree* of how a process satisfies its *compliance requirement* the *compliance preference* of a process.

B. Preference Valuation System

During run time, as processes are completed, a preference valuation system can be consulted to obtain a valuation of degree of compliance of the active process instance. There are systems already defined that expand this notion in the area of penalty addition to behaviour patterns [32,33] as well as cost-benefit analysis systems based on semantic QOS frameworks [28,29,34]

A set of standard business process can be provided in formalized best-practices guidelines within many industry domains [3,5–9].

C. Engineering Compliance Requirements

In this section we present a methodology for acquiring, maintaining and using potentially vague and imprecise requirements.

A process is a set of activities that are completed in order to meet various business strategies. For example if a goal is to create a new user on a computer system then a process is a set of the activities that can be completed for the actual fulfillment of the goal. When referring to a process instance, we refer to a single execution of a selected process. A compliance requirement is a measurement function that returns a value of performance based on the process instance that has just completed.

For all of the examples shown we will use two processes.

- 1) The “user creation” process. Shown in Fig.3.
- 2) The “lodgment of quarterly activity report” process.

These processes have associated requirements of:

- 1) “Selection of a valid username” - A new user must not be allocated a username that currently exists in the username data store.
- 2) “Selection of a ‘good’ password” -The password must meet the requirements of a ‘good’ password. This may be an imprecise requirement if ‘good’ is not defined.
- 3) “The user management system must be updated to include the new user”
- 4) “A quarterly report is to be filed within a reasonable amount of time” - This is a KPI based requirement that is set to limit delays in processing. This may be an imprecise requirement if ‘reasonable’ is not defined.
- 5) “A quarterly activity report must be typed using the company report template” - This is a crisp restriction to the material used to produce the report.
- 6) “A quarterly activity report must be printed and mailed to the regulatory body” - This is a crisp communication requirement, indicating email is not to be used.

For non-crisp rule values such the previously mentioned “quarterly activity statements must be filed within a reasonable time frame”, consider each possible value as a preference for degree of compliance i.e. given the requirement we associate a c-semiring value:

R(‘Activity report is lodged in 24hours’) → *⟨Good⟩*
 R(‘Activity report is lodged in 10 weeks’) → *⟨Ok⟩*
 R(‘Activity report is lodged in 9 weeks 6days’) → *⟨Fair⟩*
 R(‘Activity report is not lodged’) → *⟨Bad⟩*

This is the assertion that if a statement is filed in 24 hours, it is fulfilled and has an associate preference rating of *Good*. If a statement is not filed then it is not fulfilled and has a *Bad* preference value on it.

Rules and preference values can be added at the process design time [28,29]. An analyst writing compliance requirements gives examples of possible activity executions with associated values. During process instance execution a human based comparison can be made on the current activity

in comparison to the examples provided. Using the previous example requirements, if we find that “the quarterly activity statement is actually filed in 7 days” then an appropriate value between $\langle Good \rangle$ and $\langle Ok \rangle$ can be given (method for this activity is described in the next subsection) [15,16]. The more examples given during the design time process policy creation with preference values, the more accurate our values and measures will be at run time. Note that each preference may have associated with it an n-ary preference.

Example 4: An example of giving process policies preferences and example cases. For the process “User creation” the process policies are:

- “Selection of a valid username” which has the requirement that a valid username is a username that does not already exist on the computer system.
 - If a username exists on the system then a new username should be selected e.g. ‘John’ exists so the next new username maybe ‘John.Smith’ this has a preference $\langle Good \rangle$ and is more preferred over non completion
 - If a username exists and the new user is not created then the rest of the process can not be completed. This is $\langle Bad \rangle$.
- “Selection of a ‘good’ password” at this point ‘good’ is not a complete requirement, but the selection of a password is. This is a combination of a crisp and non-crisp rules.
 - A good password should stand up against possible dictionary attacks, it should not be the same as the users name and should not be easily guessable. A password of mixed case alphanumeric characters (e.g. a-z A-Z 0-9) should be used and the length should be greater than 8 characters e.g. pR0z@c99. If a password meets these standards then it is $\langle 100\% \rangle$
 - If a password is not at least 8 characters long but contains mixed case alphanumeric characters (e.g. f1D0) then it is considered $\langle 50\% \rangle$
 - If a password is 8 characters but does not contain mixed characters (e.g. passwords) then it is considered $\langle 10\% \rangle$
 - If a password is not selected then the compliance is $\langle 0\% \rangle$
- “Update the computer system with details of new user” this activity is undertaken to save the user to the computer system and the requirement is that it must be completed to complete the process. On completion this is $\langle Good \rangle$, if there is an error this is $\langle Bad \rangle$.

Implementing crisp rules with an added imprecise preference values gives a method for checking the degree of compliance of various processes. Each measure can be formed to meet the specification of a specific company or industry, an example of QoS based metrics can be seen in

[28,29].

D. Method for Obtaining Compliance Preference

To rank and show *compliance preferences* we have chosen to use a *lattice* [14]. A lattice is a structure that can be used to diagrammatically represent a partially ordered set. The use of a lattice provides a formal setting to represent concept hierarchies and value preferences [15,16].

Returning to our lodgment of quarterly activity report example:

- R(‘Activity report is lodged in 24hours’) $\rightarrow \langle Good \rangle$
- R(‘Activity report is lodged in 10 weeks’) $\rightarrow \langle Ok \rangle$
- R(‘Activity report is lodged in 9 weeks 6days’) $\rightarrow \langle Fair \rangle$
- R(‘Activity report is not lodged’) $\rightarrow \langle Bad \rangle$

We say that R produces a set of compliance preferences R_p . During the determination of compliance preferences a relation of equality should be defined such that for the dyad $\{Good, Ok\}$ there exists a partial order \leq between elements (i.e. $Good \leq Ok$ - $Good$ is more preferred to Ok). Note: If $a, b \in R_p; a \leq b$ and $b \leq a$; then $a \equiv b$.

For each set of compliance preferences R_p where $\forall a, b \in R_p, a \leq b$ or $b \leq a$ the set has a total order. A chain of compliance preferences in the set R_p is called a chain RC_p^i if $\forall a, b \in RC_p^i, a < b$ or $b < a$. A chain of compliance preferences contains elements that can be compared with each other.

For each element in the example we show partial order as follows: $\{Good \leq Ok\}$, $\{Good \leq Fair\}$, $\{Ok \equiv Fair\}$, $\{Fair \leq Bad\}$, $\{Ok \leq Bad\}$. This is shown in Fig. 1. In Fig. 1(a) we show a point labeled ‘Good’ with a rising line to a second point in space ‘Ok’, this represents the order that ‘Good’ is better than ‘Ok’.

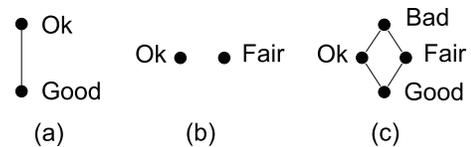


Figure 1. Preference Lattice

In Fig. 1(b) we show a point labeled ‘Ok’ and a point labeled ‘Fair’. In our definition $Ok \equiv Fair$. When represented on a lattice the two points are non-comparable so they exist at the vertical position as each other with no connecting line. In Fig. 1(c) we show 4 points ‘Bad’, ‘Ok’, ‘Fair’, and ‘Good’. ‘Good’ is at the bottom of the diagram as it is the best value. A rising line connects ‘Good’ to ‘Ok’ and another rising line connects ‘Good’ to ‘Fair’. These rising lines represent the next in order. There is a rising line from ‘Ok’ to bad, this line shows that ‘Ok’ is better than ‘Bad’ (similarly ‘Fair’ to ‘Bad’).

Previously we introduced the formalism for Lattice Chains. When we examine Fig. 1(c) we can see that 2 maximum length chains exist (each chain may be decomposed

into smaller chains). Each maximum length chain can be found by taking the best element (in our example ‘Good’) and following a single path up along rising lines until there is no worse element (stop at the worst element ‘Bad’). This is shown in Fig. 2.

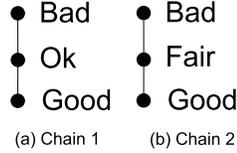


Figure 2. Preference Lattice Chains

When compiling a new compliance preference system each new value element can be added to the value set one by one as in [15,16]. For small preference systems this is a straight forward task. For larger preference systems with large quantities of preference values we have provided a method for introducing new preference values.

Step 1: Identify each lattice chain. As the compliance preference structure is built a preference lattice will be generated.

Step 2: For each lattice chain identify if there exists preference values v_1, v_2 in each lattice chain and process instances p_1, p_2 such that $R(p_1) = v_1$ and $R(p_2) = v_2$ with a defined order of $v_1 \leq v_2$ if a new instance p is between p_1, p_2 such that $p_1 \preceq_c p \preceq_c p_2$ then $R(p)$ can be any value v_p s.t. $v_1 \leq v_p \leq v_2$. Where \preceq_c is the process compliance ordering.

For example if $p_1 \Rightarrow$ ‘Statement is filed in 24hours’ and $p_2 \Rightarrow$ ‘Statement is not filed’. When introducing a new instance $p \Rightarrow$ ‘Statement is filed in 10 weeks’ the value (v_p) associated to p must be between $v_1 \Rightarrow$ <Good> and $v_2 \Rightarrow$ <Bad>. When we introduce the new value ‘Ok’ it stands that ‘Good’ \leq ‘Ok’ \leq ‘Bad’.

Step 3: If no such lattice chain exist, then we identify, for each lattice chain with at least one assigned value, either:

- The greatest value v_i s.t. $R(p_i) = v_i$ (over all p_i ’s that satisfy this constraint) and $p_i \preceq_c p$
- The least value v_j s.t. $R(p_j) = v_j$ and $p \preceq_c p_j$.

For example if $p_1 \Rightarrow$ ‘Statement is filed in 24hours’ and $p_2 \Rightarrow$ ‘Statement is filed in 10 weeks’. When introducing a new instance $p \Rightarrow$ ‘Statement is filed in 9 weeks 6days’, a determination that p and p_2 are non comparable as they are saying the same thing. By determining p is not in a chain with p_2 we say the element $p \notin RC_p^i$ if $p_2 \in RC_p^i$, and if $p \not\preceq_c p_1$, we create a new lattice chain RC_p^j consisting of the elements $p_1 \Rightarrow v_1 \leq p \Rightarrow v$.

This method can be used to devise preference values for completing each process instance. Multiple preference tables may be used to represent varying business goals. We now provide an example of the above method used to identify the preferences for the process of “user creation” (shown in Fig.3)

Example 5: Example of Preference Methodology. For the following example we will refer to Fig.3, this is a business process model of the user creation process. The process begins with an administrator requesting a username and password from the user. The user then returns a username and password selection. The administrator logs into the DBMS and runs the create user function - CreateUserCmd(). A username and password are supplied as input and then the DBMS is saved. The requirements for this process are listed in example 4. As per our methodology step one is to identify chains of sequence. For this we consider the sequence described above with the possible execution splits encountered in the requirements description. The username can be rated as either <good> or <bad> and the password either <100%>, <50%>, <10%>, or <0%>. From this we will begin to build chains. Starting with a combination of <good,100%> to represent a valid username name and a password greater than 8 mixed characters that is not the same as the username or a word in the dictionary. We continue to add the values <good,50%>, <good,10%>, <good,0%> above the point of origin as the proceeding values are ‘worse’ than the original value. This is shown in Fig.4A. Following this we introduce <bad,100%> to the lattice. This value is less than <good,100%> but is hard to compare to <good,50%>. We create a new chain for <bad,100%> is ranked worse than <good,100%> and is at the level of <good,50%> but non comparable, this is showing Fig.4B. This placement is dependant on organizational policy as each organization may value security over non-completion of duties.

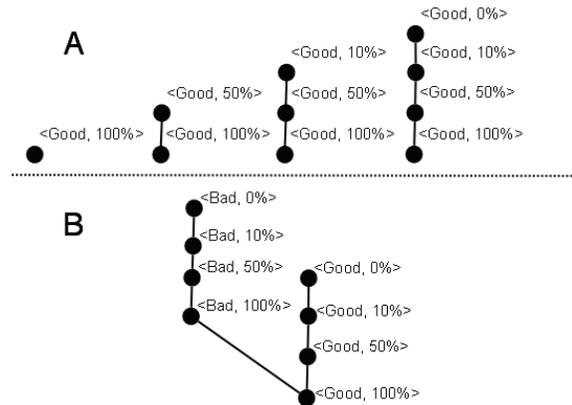


Figure 4. Compliance Scale

V. PRIORITIZING REPAIR OF NON-COMPLIANT PROCESSES

We use two notions of *compliance-driven process repair*, these are *design repair* and *execution repair*.

In the measurement of degree of compliance we investigate the possibility of analyzing each measure, and its variance from other similar processes. For a single process

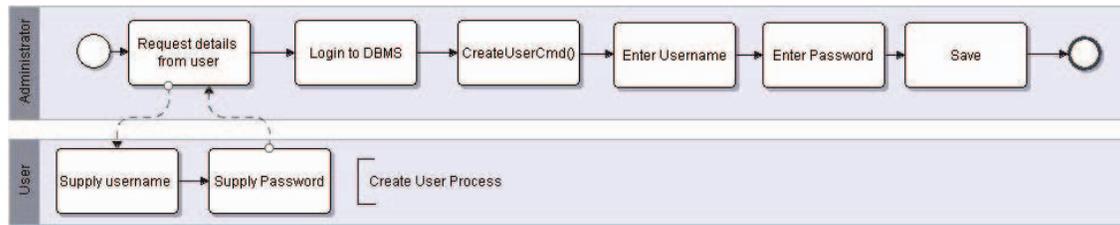


Figure 3. User Creation Process

instance, it is an easy thing for an organisation or departmental unit to follow policy and perform at an optimal compliance level when auditors are watching. For auditing run time compliance there is a need to assess the overall level of compliance even in instances where a process is non-compliant with the requirement policies.

If a process is non-compliant and the number of instances that perform poorly outweigh the number of instances that perform well then it can be assumed that a policy requirement may need to be amended or the activity definitions may need alteration. This idea is called design repair.

Example 6: If we added two requirements that said “all new users must be created on the system based on their details of the company HR report” and “the company HR report is to be created based on the details stored on current employees from the computer system user database” an execution of this process would produce errors as new users could never be created. We would find the instance measure would look like:

Degree of Compliance = $R(\text{all new users must be added based on their details in the company HR report}) \times R(\text{the company HR report is to be created based on the details stored on current employees from the computer system user database})$.

Producing combinations where $R(\text{the company HR report is to be created based on the details stored on current employees from the computer system user database})$ and $R(\text{all new users must be added based on their details of the company HR report})$ will not align and a bad preference would always be associated with this process.

On the other side of the scale, we review example 3 where of three process instances, two process were performed correctly and one was not completed. The result of the instance measure in the example was that the combination was a bad process. The process requirements are consistent with each other and it is possible to complete the process; however there is a problem in the activity instances. It is intuitive that if a process is consistently designed then in order to improve performance of the process execution, incentives could be given to complete activities at a higher performance rate.

There are also times within industry where some level of failure is acceptable and further work in defining statistical

analysis measures should be undertaken as in [35] to interpret outlier activities and determine a solid variance acceptable for each industry.

Once a business has defined its operational domain and a list of potential activities, there should be a level of consistency in actual completion of processes provided the business logic is correct.

VI. SUMMARY AND CONCLUSIONS

In this work we have presented a method for determining a degree of compliance for business processes. A framework has been provided that can be used in the formation of a general compliance system. We have provided a method for identifying both crisp and imprecise compliance requirements and applying preference values for process instance compliance evaluation. When implemented as a monitoring device on existing compliance frameworks our system can be used to provide transactional monitoring and show valuations of costing and benefits associated to each process.

If an organization were to adopt these rigorous standards for auditing and preparing the granular run-time transaction statements as we have shown, we would expect that there would be alleviation on further unpredictable behaviour within the organization. The degree of compliance for processes could also be utilized to identify and repair processes that exhibit non-compliance behaviours.

REFERENCES

- [1] Donleavy, G.D.: The law of requisite variety applied to corporate governance issues. In: Proceedings of ICSSSM. (2005) 63–68
- [2] IT-012, J.T.C.: ISO/IEC 27002:2006 Information Technology - Security Techniques - Code of Practice for Information Security Management (2006)
- [3] Berns, S., Baron, P.: Company law and governance: An Australian Perspective. Oxford University Press (1998)
- [4] Udipi, Y.B., Singh, M.P.: Governance of cross-organizational service agreements: A policy-based approach. In: SCC. (2007) 36–43
- [5] Bayne, J.S.: A theory of enterprise command and control. In: MILCOM. (2006) 1–8
- [6] Calder, A., Watkins, S.: International IT Governance. Kogan Page (2006)

- [7] Deloitte, T.: Internal Control Issues in Derivatives Usage, An Information Tool for Considering the COSO Internal Control - Integrated Framework in Derivatives Applications. COSO (1996)
- [8] IT Governance Institute: Cobit 4.1 (2007)
- [9] Niven, P.R.: Balanced Scorecard Diagnostics (Maintaining Maximum Performance). John Wiley and Sons (2005)
- [10] United States of America in Congress: Sarbanes-oxley act (2002)
- [11] Australian Securities and Investment Commission: Corporate law economic reform program act 1999 (2003) Accessed 08/01/09.
- [12] Commonwealth Government of Australia: Anti-money laundering and counter-terrorism financing act 2006 (2009)
- [13] Bistarelli, S.: Semirings for soft constraint solving and programming. In: Springer Berlin / Heidelberg (2004) 21–50
- [14] Donnellan, T.: Lattice theory. Pergamon Press (1968)
- [15] Carpineto, C., Romano, G.: Galois: An order-theoretic approach to conceptual clustering. In: In Proceedings of the Machine Learning Conference. (1993) 33–40
- [16] Huchard, M., Hacene, M., Roume, C., Valtchev, P.: Relational concept discovery in structured datasets. *Annals of Mathematics and Artificial Intelligence* **49**(1) (2007) 39–76
- [17] McGill, R., Sheppey, T.: Sarbanes-Oxley : building working strategies for compliance. Palgrave Macmillan (2007)
- [18] American Electronics Association: Sarbanes-oxley section 404: The 'section' of unintended consequences and its impact on small business. (2005)
- [19] Engel, E., Hayes, R.M., Wang, X.: The sarbanes-oxley act and firms' going-private decisions. *Journal of Accounting and Economics* **44**(1-2) (2007) 116 – 145
- [20] A.T. Kearney: 2007 foreign direct investment confidence index. Technical report, Global Business Policy Council (2007)
- [21] Ghose, A., Koliadis, G.: Auditing business process compliance. In: Service-Oriented Computing ICSOC 2007. Springer Berlin / Heidelberg (2007) 169–180
- [22] Rozinat, A., van der Aalst, W.M.P.: Conformance testing: Measuring the fit and appropriateness of event logs and process models. *Business Process Management Workshops* **3812/2006** (2006) 163–176
- [23] van der Aalst, W.M.P., Dumas, M., Ouyang, C., Rozinat, A., Verbeek, E.: Conformance checking of service behavior. *ACM Trans. Interet Technol.* **8**(3) (2008) 1–30
- [24] Morrison, E., Menzies, A., Koliadis, G., Ghose, A.: Business process integration: Method and analysis. In: Proceedings of the Sixth Asia-Pacific Conference on Conceptual Modelling (APCCM09). (2009)
- [25] Governatori, G., Milosevic, Z., Sadiq, S.: Compliance checking between business processes and business contracts. In: EDOC 2006, IEEE Computing Society (2006) 221–232
- [26] Governatori, G., Hoffmann, J., Sadiq, S., Weber, I.: Detecting regulatory compliance for business process models through semantic annotations. In: 4th Int. Workshop on Business Process Design. (2008)
- [27] Lu, R., Sadiq, S., Governatori, G.: Compliance aware business process design. In: 3rd International Workshop on Business Process Design (BPD'07), Springer (2007) 120–131
- [28] Hirsch, D., Tuosto, E.: Shreq: Coordinating application level qos. In: SEFM05, Washington, DC, USA (2005) 425–434
- [29] Jiang, Y.: A basic stochastic network calculus. *SIGCOMM Comput. Commun. Rev.* **36**(4) (2006) 123–134
- [30] Governatori, G., Milosevic, Z.: A formal analysis of a business contract language. *Int. Journal of Cooperative Information Systems* **15**(4) (2006) 659–685
- [31] Becker, G.S.: Crime and punishment: An economic approach. *The Journal of Political Economy* **76**(2) (1968) 169–217
- [32] Wyner, A.: Rudiments of computational jurisprudence. In: IWCS-6. (2005) 416–420
- [33] Hohfeld, W.N.: Fundamental Legal Conceptions as Applied in Judicial Reasoning. Volume 23-5. Yale University Press (1923)
- [34] García, J., Ruiz, D., Ruiz-Cortés, A., Martín-Díaz, O., Resinas, M.: An hybrid, qos-aware discovery of semantic web services using constraint programming. In: ICSOC07. Springer Berlin / Heidelberg (2007) 69–80
- [35] Kshirsagar, S., Blaschke, T., Sheiner, L., Krygowski, M., Acosta, E., Verotta, D.: Improving data reliability using a non-compliance detection method versus using pharmacokinetic criteria. *Journal of Pharmacokinetics and Pharmacodynamics* **34**(1) (2007) 35–55