Modeling Control Objectives for Business Process Compliance

Shazia Sadiq[1], Guido Governatori[1], Kioumars Naimiri[2]

[1] School of Information Technology and Electrical Engineering,
The University of Queensland, St Lucia QLD 4072.
Brisbane, Australia
[2] SAP Research Centre CEC Karlsruhe, SAP AG, Vincenz-Prießnitz-Str.1
76131 Karlsruhe, Germany
{shazia, guido}@itee.uq.edu.au; kioumars.namiri@sap.com

Abstract. Business process design is primarily driven by process improvement objectives. However, the role of control objectives stemming from regulations and standards is becoming increasingly important for businesses in light of recent events that led to some of the largest scandals in corporate history. As organizations strive to meet compliance agendas, there is an evident need to provide systematic approaches that assist in the understanding of the interplay between (often conflicting) business and control objectives during business process design. In this paper, our objective is twofold. We will firstly present a research agenda in the space of business process compliance, identifying major technical and organizational challenges. We then tackle a part of the overall problem space, which deals with the effective modeling of control objectives and subsequently their propagation onto business process models. Control objective modeling is proposed through a specialized modal logic based on normative systems theory, and the visualization of control objectives on business process models is achieved procedurally. The proposed approach is demonstrated in the context of a purchase-to-pay scenario.

Keywords: Compliance, Risk, Internal Controls, Business Process Design

1 Introduction

The importance of compliance has dramatically increased over the last few years for businesses in several industry sectors. Essentially, compliance is ensuring that business processes, operations and practice are in accordance with a prescribed and/or agreed set of norms. Compliance requirements may stem from legislature and regulatory bodies (e.g. Sarbanes-Oxley, Basel II, HIPAA), standards and codes of practice (e.g. SCOR, ISO9000) and also business partner contracts. Compliance related software and services is expected to reach a market value of over $27billion this year [17]. The boost in business investment is primarily a consequence of regulatory mandates that emerged as a result of recent events that led to some of the

5th International Conference on Business Process Management

largest scandals in corporate history such as Enron (USA) and HIH (Australia). In spite of mandated deadlines there is evidence that many organizations are still struggling with their compliance initiatives. A recent report [4] identifies the gap between management focus on compliance related issues and IT's lack of ability to implement the critical policies and procedures.

A number of compliance service/solution providers are currently available. Traditionally these are large consulting firms such as PriceWaterhouseCoppers, Deliotte etc. However software vendors are also emerging ranging from large corporations with products such as IBM Lotus workplace for Business Controls & Reporting, Microsoft Office Solutions Accelerator for Sarbanes-Oxley, SAP GRC (Governance, Risk and Compliance) Solution, as well as niche vendors such as OpenPages, Paisley Consulting, Qumas Inc and several others.

Compliance is predominantly viewed as a burden, although there are indications that businesses have started to see the regulations as an opportunity to improve their business processes and operations. Industry reports [17] indicate that up to 80% of companies said they expected to reap business benefits from improving their compliance regimens. This has opened a new but complex set of challenges for enterprise software vendors.

Currently there are two main approaches towards achieving compliance. First is *retrospective reporting*, wherein traditional audits are conducted for "after-the-fact" detection, often through manual checks by expensive consultants. A second and more recent approach is to provide some level of automation through *automated detection*. The bulk of existing software solutions for compliance follow this approach. The proposed solutions hook into variety of enterprise system components (e.g. SAP HR, LDAP Directory, Groupware etc.) and generate audit reports against hard-coded checks performed on the requisite system. These solutions often specialize in certain class of checks, for example the widely supported checks that relate to Segregation of Duty violations in role management systems. However, this approach still resides in the space of "after-the-fact" detection. Although, the assessment time is reduced, and correspondingly the time to remediation and/or mitigation of control deficiencies is also improved. This improvement is much sought after as is evident from the heavy investment in compliance software during the last few years.

A major issue with the above approaches (in varying degrees of impact) is the lack of sustainability. Even with automated detection facility, the hard coded check repositories can quickly grow out of control making it extremely difficult to evolve and maintain them for changing legislatures and compliance requirements. In addition to external pressures, there is often a company internal push towards quality of service initiatives for process improvement which have similar requirements. The complexity of the situation is exasperated by the presence of dynamically changing collaborative processes shared with business partners. The diversity, scale and complexity of compliance requirements warrant a highly systematic and well-grounded approach.

We believe that a sustainable approach for achieving compliance should fundamentally have a preventative focus. As such, we envisage an approach that provides the capability to capture compliance requirements through a generic requirements modeling framework, and subsequently facilitate the propagation of these requirements into business process models and enterprise applications, thus achieving *compliance by design*.

In light of the heavy socio, economic and environmental costs of non-compliance, a priori embedding of requisite checks and triggers into the enterprise applications is clearly desirable but also extremely difficult given that the technology landscape of today's organizations is disparate, and distributed. This is further complicated by several factors, legacy systems, distributed operations, outsourcing, and imperfect work practices to name a few.

Business process models may seem the most natural venue for the modeling of compliance related controls. However, our study indicates that an attempt to prematurely load business process models with compliance controls will be highly problematic from a practical standpoint. This is the basic premise of our approach.

In this paper, our objective is two fold. We will firstly present in section 2, a detailed discussion on the problem space of business process compliance, identifying major technical and organizational challenges. The scale of the problem space is beyond the scope of one paper, however, in this paper we tackle a part of the overall space, which deals with the effective modeling of control objectives (in section 3), and subsequently its interplay with business process models (in section 4). We present a review of current literature in section 5, followed by an outlook on future challenges in section 6.

2 The Problem Space

Business process management is well recognized as a means to enforce corporate policy. Regulatory mandates also provide policies and guidelines for business practice. One may argue why a separate requirements modeling facility is required to capture compliance requirements for business processes. We identify the following reasons against this argument:

Firstly, the source of these two objectives will be distinct both from an ownership and governance perspective, as well as from a timeline perspective. Where as businesses can be expected to have some form of business objectives, control objectives will be dictated by mostly external sources and at different times.

Secondly, the two have differing concerns, namely business objectives and control objectives. Thus the use of business process languages to model control objectives may not provide a conceptually faithful representation. Compliance is in essence a normative notion, and thus control objectives are fundamentally descriptive, i.e. indicating *what* needs to be done (in order to comply). Business process specifications

are fundamentally prescriptive in nature, i.e. detailing *how* business activity should take place. There is evidence of some developments towards descriptive approaches for BPM, but these works were predominantly focused on achieving flexibility in business process execution (see e.g. [18], [20])

Thirdly, there is likelihood of conflicts, inconsistencies and redundancies within the two specifications. The intersection of the two needs to be carefully studied.

In summary we present in Figure 1, the interconnect between Process Management and Controls Management. The two are formulated by different stakeholders and have different lifecycles. The design of controls will impact on the way a business process is executed. On the other hand, a (re)design of a business process causes an update of the risk assessment, which may lead to a new/updated set of controls. Additionally, business process monitoring will assess the design of internal controls and serve as an input to internal controls certification.

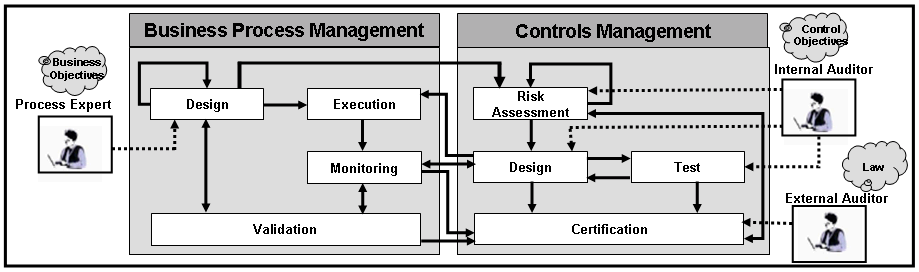

Fig. 1. Interconnect of Process Management and Controls Management

Given the scale and diversity of compliance requirements and additionally the fact that these requirements may frequently change, business process compliance is indeed a large and complex problem area with several challenges. Following our initial premise that business and control objectives are (or should be) designed separately, but must converge at some point, we present below a list of essential methods and techniques that need to be developed to tackle this overall problem.

2.1 Control Directory Management

Regulations and other compliance directives are complex, vague and require interpretation. Often in legalese, these mandates need to be translated by experts. For example the COSO framework [6] is recognized by regulatory bodies as a defacto standard for realizing controls for financial reporting. A company-specific interpretation results in the following (textual) information being created:

<control objective, risk, internal control [1]*>*

For example:

Control objective:*prevent unauthorized use of purchase order process*

Risk:*unauthorized creation of purchase orders and payments to non-existing suppliers*

Internal control:*The creation and approval of purchase orders must be undertaken by two separate purchase officers*

The above example is typical of the well known segregation of duty constraint (one individual does not participate in more than one key trading or operational function) mandated by Sarbanes-Oxley 404.

However, business will typically deal with a number of regulations/standards at one time. Thus there is a need to provide a structured means of managing the various interpretations within regional, industry sector and organizational contexts. We identify this as a need for a *controls directory***.** Control directory management could be supported by database technology, and/or could present some interesting content management challenges, but will be an essential component in the overall solution. There is some evidence in industry reports [e.g. SAP GRC Repository] that large solution vendors are producing repositories of control objectives (and associated parameters) against the major regulations.

2.2 Ontological Alignment

Interpretation of regulations from legal /financial experts comes in the form of textual descriptions (see example in section above). Establishing an agreement on terms and usage between these descriptions and the business processes and constituent activities/transactions is a difficult but essential aspect of the overall methodology.

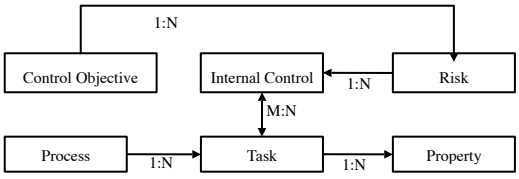

Fig. 2. Relationships between Process Modeling and Control Modeling Concepts

[1] "Internal control is broadly defined as a process, effected by an entity's board of directors, management and other personnel designed to provide reasonable assurance regarding the achievement of objectives in the following categories: Effectiveness and efficiency of operations; Reliability of financial reporting; and Compliance with applicable laws and regulations." [6]

In the Fig 2, we present the relationships between the basic process modeling and control modeling concepts. Clearly the relationship between process task and internal controls is much deeper than shown as it would require alignment between embedded concepts e.g. task identification, particular data items, roles and performers etc. However, it is evident that several controls may be applicable on a task, and one control may impact on multiple tasks as well. What tools and techniques are utilized to provide an effective alignment between the two conceptual spaces is not the focus of this paper, but none the less an important question at hand.

2.3 Modeling Control Objectives

The motivation to model control objectives is multifaceted: Firstly, a generic requirements modeling framework for compliance by design will provide a substantial improvement over current after-the-fact detection approaches. Secondly, it will allow for an analysis of compliance rules thus providing the ability to discover hidden dependencies, and view in holistic context, while maintaining a comprehensible working space. Thirdly, a precise and unambiguous (formal) specification will facilitate the systematic enrichment of business processes with control objectives.

A fundamental question in this regard is the *appropriate formalism* to undertake the task. In the next section we will deliberate further on this question, and also provide a discussion of complementary approaches in the section on related work.

2.4 Process Model Enrichment

In this context, we use the term process model enrichment as the ability to enhance enterprise models (business processes) with compliance requirements. This is essentially provided as *process annotation* (see section 4). The resultant visualization of control objectives on the process model, facilitates a better understanding of the interaction between the two specifications for both stakeholders (process owners as well as compliance officers).

However, the visualization is only a first step. The new checks introduced within the process model, can in turn be used to analyse the model for measures such as *compliance distance* that can provide a quantification of the effort required to achieve a compliant process model. Eventually, process models may need to be modified to include the compliance requirements.

2.5 Event Monitoring

The support provided in the design of compliant processes through process annotation and analysis and resultant process changes, will eventually lead to a *model driven enforcement of compliance controls* (where process management systems are in place). However, it is naïve to assume that all organizations have the complete implementation of the BPM lifecycle, and hence the process models and underlying applications may be disconnected. In this case, it is important to provide support for compliance through run time monitoring. This has been the agenda for several vendors in this space targeting the so called *automated detection,* described earlier. In general event monitoring is a well studied research topic [see e.g. www.complexevents.com], and although has not been widely/explictly associated with the compliance issue (notably excepting [10]), its usage in fraud detection and security is closely related.

Although, our work is primarily targeted at achieving *compliance by design* by adopting a preventative approach facilitated by business process models, the work on formal modeling of control objectives has taken into account the violations and resultant reparation policies that may surface at runtime (see next section).

3 Modeling Control Objectives

Our observation is that a compliance requirement (or its translation into a control objective and subsequently internal controls) can be reduced to the identification of what obligations an enterprise has to fulfill to be deemed as compliant. Initial work in this area [12] in the context of business contracts (a special case of compliance) has already provided the basic concepts leading to the adoption of formal models of normative systems as a candidate representation for control objectives.

In general a formal model of a normative system provides a precise and unambiguous account of the obligations, permissions, prohibitions as well as other normative positions an entity is subject to in the context where the normative system applies. To formalize normative systems one has to capture the logical properties of the notions of the normative concepts (e.g., obligations, prohibitions, permissions, violations, …) and how these relate to the entities in an organization and to the activities to be performed. Deontic logic is the branch of logic that studies normative concepts such as obligations, permissions, prohibitions and related notions. Over the years many different deontic logics have been proposed to capture the intuitions behind these notions. Standard Deontic Logic (SDL) offers a very idealized and abstract conceptual representation of the basic normative notions [5], but at the same time it suffers from several drawbacks given its high level of abstraction. One of the main limitations in this context is its inability to reason with violations, and the obligations arising in response to violations [19].

We propose FCL-Formal Contract Language [15] as formalism to express normative specifications. FCL is a combination of an efficient non-monotonic formalism (defeasible logic) and a deontic logic of violations [14] offering the right trade off between expressive power and computational complexity. The key idea of the logic of violations, backed-up by current views of legal theory, is that a normative document consists of a set of (normative) clauses regulating the intended behaviour of a system, and given the non-monotonic nature of normative systems (i.e., normative concepts admit exceptions), it is not possible to consider the clauses of the normative document in isolation, but the normative documents must be conceived as a whole (often clauses in apparently unrelated sections of the document can have mutual effects on each other).

In addition the document specifies only explicit behaviors. The basic mechanism of the logic of violations [14] takes a modular approach to the problem and it recursively deduces new clauses from the existing clauses in a module and combines clauses related to violations and obligations generated in response to violations. Then it recursively merges the clauses in different modules and computes new clauses resulting from the interaction among modules. The modularity of the mechanism used by FCL is of particular relevance for compliance since the architecture of modern enterprise systems is based on the composition of diverse components. In this way it is possible to revise the specifications of a component of a business process or a section in the normative specifications without being forced to perform a complete revision of the representation of the business process or of the normative document as it is often the case with hard-coded solutions.

Furthermore, the reasoning mechanism of defeasible logic is based on constructive proofs, thus for any conclusions it is possible to have a trace of the derivation, which then can be used to provide an explanation of the reasons why the conclusion has been obtained. This property is very important for compliance and auditing, since we are not only interested that a process is not compliant but we want the reasons why it does not comply.

In the following sections, we will provide an illustration on the use of FCL through a purchase-to-pay scenario which is often impacted by several regulations and best practice standards depending on the industry sector, region and organizational setup. A representative list of possible control objectives for the scenario is also provided. The FCL encoding is intended to demonstrate the natural fit of the proposed formalism for control objectives, and in turn provide the basis for business process model enrichment and analysis, which will be discussed in section 4.

3.1 Purchase-to-Pay (P2P) Scenario

Purchase-to-Pay is a well known process within procurement applications. A simplified version of the process is given in Figure 2. The assumption is that the

design of this process was governed primarily by business (improvement) objectives. Figure 2 provides the supplier perspective as well (in the lower half) for completeness.

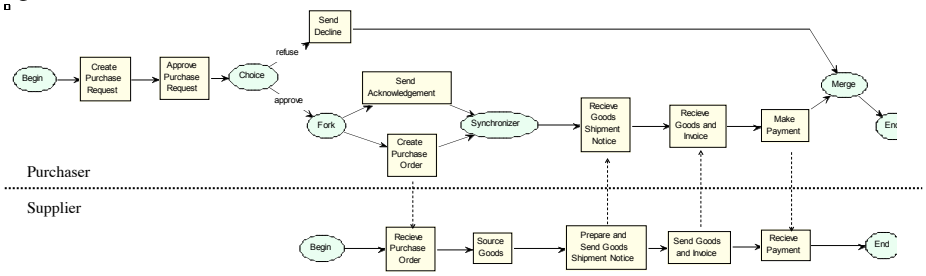

Fig. 3. Purchase-to-Pay Scenario

The generic P2P process may be subject to a number of control objectives emerging from compliance requirements (regional regulations, commercial standards, partner obligations etc.). In the table below we present a selected set of control objectives. Each of these objectives will have a corresponding risk statement, as well as a translation to an internal control indicating *effective* implementation [6] of the control objective.

Table 1. Control Objectives for Purchase-to-Pay Scenario

Control Objective	Risk	Internal Control
Prevent unauthorized use of purchase order process	Unauthorized creation of purchase orders and payments to non-existing suppliers	The creation and approval of purchase requests must be undertaken by two separate purchase officers
	Misappropriation of goods	Every Invoice must contain a valid Purchase Order Number
Ensure adequate supply of materials	Production delays due to lack of resources/ materials	Supplier can be charged a penalty if goods not received within *k* days of receipt of goods shipment notice
Timely and efficient P2P Process	Production delays due to lack of resources/ materials	Purchase requests not closed (declined or converted to Purchase Orders) within *k* days should raise an alert to purchasing manager

3.2 FCL Basics

In this section we outline the basic elements of FCL in order to illustrate how to use this formalism to represent and reason about “normative” specifications relative to a business process. For detailed presentation of the formalism we refer to [15], [12].

A rule in FCL is an expression of the form $r{:}A_1,..., A_n \Rightarrow B$, where r is the name of the rule (unique for each rule), $A_1,..., A_n$ are the premises, (propositions in the logic), and B is the conclusion of the rule (again B is a proposition of the logic).

The propositions of the logic are built from a finite set of atomic propositions, and the following operators: ¬(negation), O(obligation), P(permission), ⊗(violation/ reparation). The formation rules are as follows:

- every atomic proposition is a proposition;
- if p is an atomic proposition, then $\neg p$, is a proposition;
- if p is a proposition then Op is an obligation proposition and Pp is a permission proposition; obligation propositions and permission propositions are deontic propositions
- if $p_1,...,p_n$ are obligation propositions and q is a deontic proposition, then $p_1 \otimes ... \otimes p_n \otimes q$ is a reparation chain;

A simple proposition corresponds to a factual statement. The deontic operators are then indexed by the subject of the normative position corresponding to the operator. Thus $O_s SendInvoice$ means that the supplier s has the obligation to send the invoice to the purchaser, and $P_p ChargePenalty$ means that the purchaser p is entitled (permitted) to charge a penalty to the supplier. A reparation chain, for example

$$O_s ProvideGoodsTimely \otimes O_s OfferDiscout \otimes P_p ChargePenalty$$

captures obligations and normative positions arising in response to violations of obligation. Thus the expression above means that the supplier has the obligation to send the goods in a timely manner, but in case she does not comply with this (i.e., she violates the obligation do so) then she has the "secondary" obligation to offer a discount for the merchandise, and in case that she fails to fulfill this obligation (i.e., we have a violation of the possible reparation of the "primary" obligation), then, finally, the purchaser can charge the supplier with the penalty.

As usual in normative reasoning we have two types of rules: definitional rules and normative rules. A definitional rule gives us the conditions that assert a factual statement, while a normative rule allows us to conclude a normative positions (i.e., an obligation, a permission or a prohibition, where a prohibition is $O\neg$ or equivalently $\neg P$). According to the above distinction in definitional rules the conclusion is a proposition, and in normative rules the conclusion is either a deontic proposition or a reparation chain. In both cases the premises are propositions and deontic propositions, but not reparation chains.

FCL offers two reasoning modules: (1) a normaliser to make explicit rules that can be derived from explicitly given rules by merging their normative conclusions, to remove redundancy and identify conflicts rules; and (2) an inference engine to derive conclusions given some propositions as input.

Finally to incorporate the temporal dimension we timestamp all propositions in the language, and we adopt the persistence mechanism devised in [16] to deal with temporalised normative positions. Essentially if we can assert the conclusion $p{:}t_0$, i.e.,

p holds at time t_0, then we can continue to assert p for all $t'>t_0$, until we have an event such that we can terminate the validity of p.

3.3 Encoding

Below we provide FCL encoding for the internal controls specified in Table 1.

The creation and approval of purchase requests must be undertaken by two separate purchase officers

$c1$:*CreatePR*(x,y):t, *PurchaseOfficer*(y):t, *PurchaseOfficer*(z):t', $y \neq z$:$t' \Rightarrow O_p$*ApprovedPR*(x,z) :t'

The predicate *CreatedPR*(x,y):t means that at time t, y has created a Purchase Request whose Id is x; the meaning of *ApprovedPR* is similar. The predicate *PurchaseOfficer*(x) states that at the time of the timestap t, x plays the role of purchase officer.

Every Invoice must contain a valid Purchase Order Number.

$$c2\text{:}Invoice(x,y)\text{:}t,\ PurchaseOrderNumber(x,z)\text{:}t \Rightarrow O_s\ Include(y,z)\text{:}t$$

This internal control gives rise to two rules in FCL. The meaning of the predicates is as follows: *Invoice*(x,y):t means at time t, the object with Id y is the invoice for some purchase order x"; *PurchaseOrderNumber*(x,z):t means at time t, z is the purchase order number for order x"; *Include*(y,z):t means the object z is included in the object z.

Supplier can be charged a penalty if goods not received within k days of receipt of goods shipment notice

$$c3\text{: }GoodShipmentNotice(x,y)\text{:}t \Rightarrow O_s\ SendGood(x)\text{:}t+k \otimes P_p ChargePenalty$$

Notice that this internal control presupposes the existence of a primary obligation to provide the goods within k days (*SendGood*). In case this provision in violated then the purchaser is entitled to charge the supplier with the established penalty.

Purchase requests not closed (declined or converted to Purchase Orders) within k days should raise an alert to purchasing manager

$$c4.1\text{: }CreatePR(x,y)\text{:}t \Rightarrow O_p ClosePR(x)\text{:}t+k \otimes O_p AlertPurchaseManager\text{:}t+k$$

Here *ClosePR*(x):$t+k$ gives the deadline to change the status of a purchase request from open ($\neg$ *ClosePR*(x)) to closed. Beside the normative provision give by rule c4.1, this internal control gives conditions under which we can change the status of a request from open to close.

$$c4.2\text{: }ApprovePR(x,y)\text{:}t' \Rightarrow ClosePR(x);t'$$

$$c4.3\text{: }Decline(x)\text{:}t' \Rightarrow ClosePR(x);t'$$

Notice that the last two rules are definitional rules and not normative rules. However an additional rule is needed to set the status of a purchase request as open when it is created. Thus we introduce the rule

$$c4.4: \mathit{CreatePR}(x,y):t' \Rightarrow \neg\, \mathit{ClosePR}(x);t'$$

In this case we make use of the persistence condition discussed at the end of the previous section to maintain the state of the request as open until we can close as result of firing either $c4.2$ or $c4.3$. If this does not happened before $t+k$ we have a violation of the primary obligation of rule $c4.1$, and thus we fire the obligation to alert the purchase manager as response of this violation.

4 Process Model Enrichment

The example presented in Fig. 3 follows a simple language which can be mapped to several commercial/standard (e.g. BPMN) and formal (e.g. Petri-nets) languages. We use the notation only for its graphic simplicity. The following basic concepts provide basics for the language.

The **process model** P = <N, F> is a directed graph where N is a finite set of nodes, F is a flow relation $F \subseteq N \times N$. Flows show the control flow of the process. Nodes are classified into tasks (T) and coordinators (C), where $N = C \cup T$ and $C \cap T = \phi$. For each node $n \in N$, following basic attributes are defined:

nodeType[*n*] ∈ { TASK, COORDINATOR } represents type of *n*.

coordinatorType[*n*] ∈ { begin, end, choice, merge, fork, synchronizer }

A task $t \in T$ is not a mere node in the process graph, but has rich semantics which are defined through its properties, such as process relevant application data, temporal constraints, resources requirements etc.

Given these basic and well known concepts, the task ahead is to introduce the concepts relating to control objectives into the process while still maintaining a clear separation of concerns. To achieve this, we introduce a new concept of *control tags*.

4.1 Control Tags

We identify four types of control tags. Each tag will represent a control objective, and (one of) its corresponding internal control.

- **Flow Tag**: A flow tag represents a control objective that would impact on (the flow of) the business activities, e.g. approval of leave must occur before payment for travel.
- **Data Tag**: A data tag identifies the data retention and lineage requirements, e.g. a medical practice must retain the time of commencement of pathology tests.

- **Resource Tag**: A resource tag represents controls relating to access, role management and authorization, e.g. persons performing cash application and bank reconciliation must be different as it allows differences between cash deposited and cash collections posted to be covered up.
- **Time Tag**: A time tag identifies controls for meeting time constraints such as deadlines and maximum durations, e.g. a water leakage complaint must be investigated within 12 hours of lodging.

Control tags are constructed through parsing[2] of FCL expressions, representing normative rules. Each control tag is thereby represented by the schema shown in Table 2. The propositions related to checking of conditions are listed under *state* and represent new checks that may need to be incorporated into the process. The *operation* relates to the deontic operations in the expressions and identify new actions that may have to be undertaken within the process. As the final step in the control modeling phase, the operations in the control tags are type linked, resulting in the values listed under the *type* column in Table 2.

Table 2. Control Tags for Purchase-to-Pay Scenario

Rule	State	Task	Operation	Type	Task
*c*1	*CreatePR*(*x*,*y*):*t*, *PurchaseOfficer*(*y*):*t*, *PurchaseOfficer*(*z*):*t*', *y*≠ *z*:*t*'	Create Purchase Request	O_p*ApprovePR*(*x*,*z*):*t*'	Resource	Approve Purchase Request
*c*2	*Invoice*(*x*,*y*):*t*, *PurchaseOrderNumber*(*x*,*z*):*t*	Send Goods and Invoice	O_s*Include*(*y*,*z*):*t*	Data	Send Goods and Invoice
*c*3	*GoodShipmentNotice*(*x*,*y*):*t*	Send Goods Shipment Notice	O_s*SendGood*(*x*):*t*+*k*	Time	Send Goods and Invoice
			P_p*ChargePenalty*	Flow	Make Payment
*c*4.1	*CreatePR*(*x*,*y*):*t*	Create Purchase Request	O_p*ClosePR*(*x*):*t*+*k*	Time	Create Purchase Request
			O_p*AlertPurchaseManager*:*t*+*k*	Flow	Create Purchase Request

Lastly, an alignment of the terms used within the two specifications, namely process model (P) and control model (FCL expressions) is required. As discussed previously, it is unrealistic that the two specifications will always be constructed in synch, simply because of their disparate lifecycles, stakeholder groups and purpose within the organization. However, the overall approach presented in this paper

[2] FCL encodings can be mapped to RuleML, and consequently provide an automated means of processing. For details on RuleML mapping see [13].

(section 2), proposes a systematic way to converge the two. Table 2 provides an illustration of such an alignment in the context of the Purchase-to-Pay scenario. For each control tag, the effected process *tasks* are identified.

It is trivial to observe that the above alignment may be implicitly undertaken at the time of FCL encoding through appropriate tools (as discussed in section 2.2) that allow writing of FCL propositions to use naming consistent to process model task (and task property) names.

4.2 Process Annotation

Given Table 2, the annotation of the process model with control tags, can be done programmatically leading to automatic visualization of the control tags on business process models. Fig 4 shows a subset of the control tags given in Table 2. The annotation distinguishes two aspects of control tags: All propositions related to *state* are annotated as "check" and all deontic operations are annotated as "perform". Furthermore, the *type* of the control tag is visualized through a representative symbol.

Checks as well as perform actions of type resource, data and time represent possible modification of the effected tasks (i.e. their underlying properties). However perform actions of type flow represent possible changes to the task set and/or order of execution.

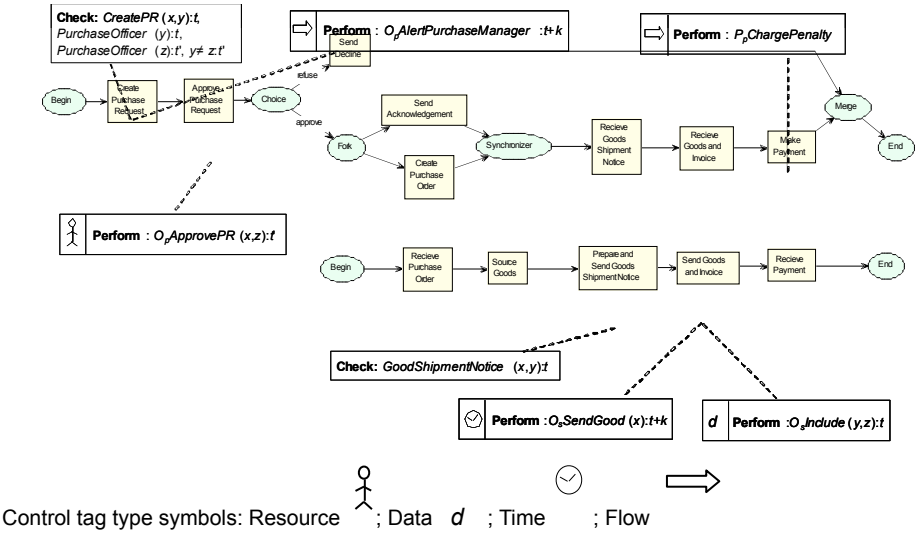

Fig. 4. Visualization of Control Tags

Process annotation allows the process designers to import and visualize control objectives within the process modeling space. In addition to the support provided to process designers through the above, we also propose the use of analysis tools. These can provide e.g. support for identifying conflicts and redundancies between the two

specifications. Similarly, they can provide an evaluation of the measure of compliance. To this effect, we introduce the notion of *compliance distance*. Compliance distance is basically a quantitative measure of *how much* a process model may have to be changed in response to a given set of control objectives. This design time analysis can be undertaken based on FCL encoding and its alignment with process tasks as given in Table 2. We base our notion of compliance distance on the two aspects of control tags: Namely the checks (*state* related propositions) and perform actions (deontic *operations)* as derived from the FCL expressions. An FCL rule is of the form $r{:}c_1,..., c_n \Rightarrow p_1 \otimes ... \otimes p_m$, where C represents the set of checks and P represents the set of perform actions. Given r rules against a given set of control objectives, a simplistic compliance distance can be computed as a sum of the number of elements in C and P. For example, for the Purchase-to-Pay scenario, the compliance distance is computed as 13 (7 checks and 6 performs).

The notion of compliance distance can be also used at run time to measure how much a particular instance deviates from the expected behavior. For example this can be done by simply counting the number of recoverable violations (i.e., unfulfilled obligations not in the last position of a reparation chain) that occurred in the process instance. However, this method considers all potential violation at the same level, thus a more realistic way would be to associate to each potential violation a cost, and then the compliance distance of a process instance from the expected ideal behavior is the sum of the cost of all actual violations in the process.

In summary, the purpose of the annotation and analysis is to provide design time support to process owners to create compliant business processes. The proposed methods provide a structured and systematic approach to undertaking changes in the process model in response to compliance requirements. Fig. 5 summarizes the overall methodology.

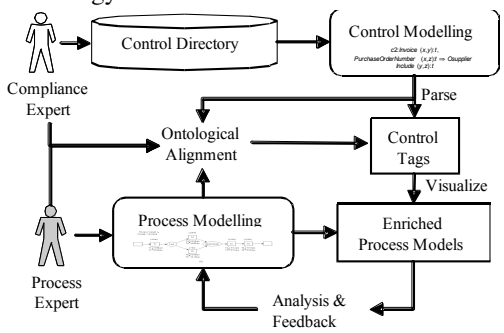

Fig. 5 Summary of Overall Methodology

5 Related Work

Both process modeling as well as modeling of normative requirements are well studied fields independently, but until recently the interactions between the two have been largely ignored [7], [21]. Some notable exceptions are the relationships between the execution (performance) of business contract based on their formal representation [9]. Research on closely related issues has also been carried out in the field of autonomous agents [8], [2].

A plethora of proposals exist both in the research community on formal modelling of rules, as well as in the commercial arena through business rule management systems (see e.g. ilog.com). It is obvious that the modelling of control objectives will be undertaken as rules, although the question of appropriate formalism is still under studied. We have proposed FCL as a candidate which has proved effective due to its ability to reason with violations, but we acknowledge that further empirical study is necessary to effectively evaluate the appropriateness of FCL.

Another closely related area is process monitoring. This is a widely studied area, which has several commercial solutions (business activity monitoring, business intelligence etc). Noteworthy in research literature is the synergy with process mining techniques [1] which provide the capability to discover runtime process behavior (and deviations) and can thereby assist in detection of compliance violations.

There have been recently some efforts towards support for business process modelling against compliance requirements. In particular the work of [22] provides an appealing method for integrating risks in business processes. The proposed technique for "risk-aware" business process models is developed for EPCs (Event Process Chains) using an extended notation. Similarly [11] present a logical language PENELOPE, that provides the ability to verify temporal constraints arising from compliance requirements on effected business processes. Distinct from the above works, the contribution of this paper has been on firstly providing an overall methodology for a model driven approach to business process compliance, and secondly on a structured technique for process model enrichment based on formal modelling of control objectives.

Lastly, significant research exists on the modelling of control flow in business processes, particularly in the use of patterns to identify commonly used constructs [www.workflowpatterns.com]. On a similar note, [10] provide temporal rule patterns for regulatory policies, although the objective of this work is to facilitate event monitoring rather than the usage of the patterns for support of design time activities.

6 Conclusions and Outlook

Process and control modeling represent two distinct but mutually dependent specifications in current enterprise systems. In this paper, we take the view that the two specifications, will be created somewhat independently, at different times, and by different stakeholders, using their respective conceptually faithful representation schemes. However the convergence of the two must be supported in order to achieve business practices that our compliant with control objectives stemming from various regulatory, standard and contractual concerns. This convergence should be supported with a systematic and well structured approach.

We have proposed such an approach. The approach allows a formal representation of control objectives in FCL, a language suitable to capture the declarative nature of compliance requirements. In turn we have introduced the concept of control tags that can be derived from FCL, and used to visually annotate and analyze typical graph based process models. We argue that such process enrichment and associated analysis capability will be instrumental in the (re) design of compliant business processes.

Next steps in our work entail the development of demonstrable methods to parse FCL to derive control tags and provide improved process annotation and analysis. The notion of compliance distance for process analysis also poses interesting research questions. We also plan to pursue the evaluation of the suability of FCL from various angles which includes an empirical study to assess the usability FCL, further theoretical analysis of its expressiveness and processing scalability, and investigation of FCL rules as an instrument for identification of runtime control violations.

References

1. W. M. P. van der Aalst, B.F. van Dongen, J.Herbst, L. Maruster, G. Schimm, A.J.M.M. Weijters (2003). Workflow Mining: A Survey of Issues and Approaches. Data & Knowledge Engineering, 47: 237 – 267
2. M. Alberti, F. Chesani, M. Gavanelli, E. Lamma, P. Mello and P. Torroni (2006). Compliance verification of agent interaction: A logic based tool. Applied Artificial Intelligence, 20 (2-4):133–157
3. G. Antoniou, D. Billington, G. Governatori, and M.J. Maher (2001). Representation results for defeasible logic. ACM Transactions on Computational Logic, 2, 2: 255-287.
4. BPM Forum (2006) CEE: The Future. Building the Compliance Enabled Enterprise. Report produced by GlobalFluency in partnership with: AXS-One, Chief Executive Magazine and IT Compliance Institute.
5. J. Carmo and A.J.I. Jones (2002) Deontic Logic and Contrary-to-Duties. In Handbook of Philosophical Logic, 2nd Ed. Volume, 8, 265-344. Kluwer.
6. COSO - The Committee of Sponsoring Organizations of the Treadway Commission. (1994) Internal Control – Integrated Framework. May 1994.

7. N. Desai, A.U. Mallya, A.K. Chopra, M.P. Singh (2005) Interaction Protocols as Design Abstractions for Business Processes. IEEE Transaction on Software Engineering 31(12): 1015-1027
8. V. Dignum, J. Vázquez-Salceda, F. Dignum (2004) OMNI: Introducing Social Structure, Norms and Ontologies into Agent Organizations. PROMAS 2004: 181-198.
9. A. D. H. Farrell, M.J. Sergot, M. Sallé, C. Bartolini (2005). Using the event calculus for tracking the normative state in contracts. International Journal of Cooperative Information Systems 14 (2-3): 99-129.
10. C. Giblin, S. Muller, B. Pfitzmann (2006) From regulatory policies to event monitoring rules: Towards model driven compliance automation. IBM Research Report. Zurich Research Laboratory. Oct 2006.
11. S. Goedertier, J. Vanthienen (2006) Designing Compliant Business Processes with Obligations and Permissions. J eder, S. Dustdar et al. (Eds.) BPM 2006 Workshops, LNCS 4103, pp 5-14. Springer Verlag 2006.
12. G. Governatori, Z. Milosevic, S. Sadiq (2006) Compliance checking between business processes and business contracts. Proceedings of the 10th IEEE Conference on Enterprise Distributed Object Computing, Hong Kong, 16-20 Oct 2006.
13. G. Governatori. (2005) Representing Business Contracts in RuleML. International Journal of Cooperative Information Systems. 14 (2-3): 181-216.
14. G. Governatori and A. Rotolo (2006) Logic of Violations: A Gentzen System for Reasoning on Contrary-To-Duty Obligations. Australasian Journal of Logic, 4, 193–215.
15. G. Governatori and Z. Milosevic. (2006) A Formal Analysis of a Business Contract Language. International Journal of Cooperative Information Systems, 15, 4: 659-685, 2006
16. G. Governatori, A. Rotolo, and G. Sartor. Temporalised normative positions in defeasible
17. logic. In A. Gardner, editor, Procedings of the 10th International Conference on Artificial Intelligence and Law, pages 25-34. ACM Press, 2005.
18. J. Hagerty (2006) SOX Spending for 2006. AMR Research, Boston USA. Nov 29, 2007.
19. M. Pesic and W.M.P. van der Aalst. (2006) A Declarative Approach for Flexible Business Processes. In J. Eder and S. Dustdar, editors, Business Process Management Workshops, Workshop on Dynamic Process Management (DPM 2006), volume 4103 of *Lecture Notes in Computer Science*, pages 169-180. Springer-Verlag, Berlin, 2006.
20. G. Sartor (2005) Legal Reasoning: A Cognitive Approach to the Law. Springer.
21. S. Sadiq, W. Sadiq, M. Orlowska (2005) A Framework for Constraint Specification and Validation in Flexible Workflows. Information Systems, 30, 5: 349-378
22. V. Padmanabhan, G. Governatori, S. Sadiq, R. Colomb and A. Rotolo. (2006) Process Modeling: The Deontic Way. In M. Stumptner, S. Hartmann and Y. Kiyoki, editors, Australia-Pacific Conference on Conceptual Modeling 2006, CRPIT 53, pp. 75-84.
23. M. zur Muehlen, M. Rosemann (2005) Integrating Risks in Business Process Models. 16th Australasian Conference on Information Systems. 29 Nov – 2 Dec. Sydney, Australia.