Association for Information Systems
AIS Electronic Library (AISeL)

ECIS 2011 Proceedings

European Conference on Information Systems (ECIS)

Summer 10-6-2011

ENFORCING COMPLIANCE ON BUSINESS PROCESSES THROUGH THE USE OF PATTERNS

Oktay Turetken

Amal Elgammal

Willem-Jan van den Heuvel,

Mike Papazoglou

ENFORCING COMPLIANCE ON BUSINESS PROCESSES THROUGH THE USE OF PATTERNS

Turetken, Oktay, European Research Institute in Service Science (ERISS), Tilburg University, Warandelaan 2 K729, 5000LE, Tilburg, NL, o.turetken@uvt.nl

Elgammal, Amal, European Research Institute in Service Science (ERISS), Tilburg University, Warandelaan 2 K704, 5000LE, Tilburg, NL, a.f.s.a.elgammal@uvt.nl

van den Heuvel, Willem-Jan, European Research Institute in Service Science (ERISS), Tilburg University, Warandelaan 2K728, 5000LE, Tilburg, NL, w.j.a.m.vdnheuvel@uvt.nl

Papazoglou, Mike, European Research Institute in Service Science (ERISS), Tilburg University, Warandelaan 2 K711, 5000LE, Tilburg, NL, m.p.papazoglou@uvt.nl

Abstract

In the past recent years, business process compliance has become an area of significant concern to many organizations. Despite an increasing number of methods and tools, organizations are still facing difficulties in finding effective support to ensure that their business processes comply with the requirements set forth by regulations, laws, standards, etc. While manual solutions offer limited assurance for compliance, there is a lack of a comprehensive framework for semi-automatically managing compliance requirements and ensuring compliance throughout all the phases of business process lifecycle. One of the foundational building blocks of such a framework is a generic conceptual model that supports factoring compliance and its relation to business processes. This paper introduces a compliance conceptual model to capture and manage compliance requirements and to relate them to business processes in a transparent and verifiable manner. The model also incorporates a set of patterns to facilitate the specification of formal compliance rules to be used for automated compliance verification and monitoring. We have developed a set of integrated tools that supports our framework and partially validated the framework in two case studies involving industry companies.

Keywords: Business Process Compliance, Compliance Model, Compliance Requirement, Internal Control, Compliance Pattern, Regulatory Compliance.

1 Introduction

In today's business environment, organizations are required to cope with an increasing number of constraints originating from various compliance sources, such as Sarbanes-Oxley (SOX, 2002), IFRS (2001), HIPAA (1996), and EU Directive 2008/30/EC (2008). Such normative laws and requirements force organizations to establish internal control systems, continuously assess their business processes, and ensure their processes are compliant with them. Without business process compliance enforced by these internal control structures, organizations face litigation risks and even criminal penalties. Industry reports identified regulation and compliance as the top business risk according to executives and analysts of different industry sectors (Ernst & Young, 2010).

In a broader perspective, *compliance* is about unambiguously ensuring conformance to a set of prescribed and/or agreed upon norms. These norms may originate from various sources, including laws and regulations, standards, public and internal policies, partner agreements and jurisdictional provisions. Although these sources translate into numerous compliance requirements, they deliberately refrain from providing any specific recommendation on how an organization achieves compliance. As such, many organizations typically achieve compliance on a *per-case basis* resulting into myriad *ad-hoc* solutions. In practice, these solutions are generally handcrafted and tailored for a particular compliance problem, resulting into process (control) definitions with a contrived and fragile layout. They lack the flexibility needed to rapidly adapt to ever-changing business environment, as they usually involve hard-coded requirements across multiple systems. With such ad-hoc solutions, it is difficult to verify compliance of business process specifications, and to monitor and ensure compliance of their execution. Having recognized these problems and their ramifications for organizations, major enterprise software vendors have developed commercial products that provide a bundled set of compliance solutions (e.g., Oracle GRC Accelerators, SAP BusinessObjects GRC, IBM OpenPages). However, these commercial products usually suffer from being highly proprietary (vendor lock-in) and technology-specific as they are usually tightly coupled with vendor provided enterprise systems.

Compliance is a multidimensional concern that applies to the entire business process (BP) lifecycle phases impacting not only the BP control and data flow, but also management and governance aspects. This, in combination with the above-mentioned impediments of existing compliance methods and tools, points towards a holistic approach and a generic framework for managing compliance throughout all BP lifecycle phases. A key component of such a framework is a *generic conceptual model* that effectively supports the compliance management practices and captures compliance information, which essentially comprises compliance requirements and relevant concepts (such as their sources, significances, etc.), as well as their interaction with business processes. Business process and compliance specifications should be decoupled as they are typically formulated by different stakeholders and have different lifecycles (Sadiq, Governatori and Namiri, 2007). Decoupling involves specifying and managing compliance information as a separate entity - starting from abstract constraints and their sources to concrete and organization specific rules- and requires them to be linked to relevant business processes to enable round-trip traceability. The explicit separation of these models helps to manage and independently evolve not only BPs but also compliance requirements, which usually change and are extended over time. It helps analysing the impact of such changes and enables the link between a BP and its compliance requirements to be scrutinized for deviations.

A conceptual model, such as the one discussed above, should unambiguously define not only concepts such as compliance requirements that are typically in *textual* forms but also dedicated constructs that are expressed using *formal languages*. Formal specification of compliance rules is necessary in order to bring about the automated assurance of business processes to compliance requirements for the entire BP lifecycle. This enables BP specifications and their executions to be automatically verified, monitored and analysed against these rules. Partially or fully automating manual business process inspections and audits would substantially reduce the overall cost of compliance. Formal models of such requirements provide an accurate and unambiguous specification facilitating a lifetime assurance of business process-based systems. Although the need for formal models is clearly shown by several studies (e.g., Liu, Muller and Xu, 2007; Sadiq et al., 2007), the knowledge required for their use and the complexity of the formalisms remain as significant obstacles for their widely adoption by compliance and business experts (Kharbili and Keil, 2010). Using

patterns supports shielding the complexity of formalisms off from business and compliance experts and facilitate their specifications in the abstract (Namiri and Stojanovic, 2007; Yu, Manh, Han and Jin, 2006).

This paper has two main contributions. *First*, it proposes a *generic conceptual model* for capturing compliance concepts and cross-correlating them to business processes and enterprise applications to establish a conceptual foundation for a BP compliance management framework. *Second*, it introduces a set of *patterns* for defining domain-specific representations of compliance requirements and generating corresponding formal statements based on these representations.

Patterns are high-level abstractions of frequently used logical formulas, which help non-technical users to represent desired properties and constraints without going into the complex details of the underlying formal language. Grounded on (Dwyer, Avrunin and Corbett, 1998), we have developed, integrated, and experimented with a new breed of *patterns* to capture frequently recurring compliance requirements. We also developed a mapping scheme from patterns to corresponding formalisms to facilitate the automated transformation of pattern-based representations into a set of logical formulas, thereby fostering the adoption of formalisms in the BP compliance domain. To investigate the feasibility and applicability of our compliance conceptual model and patterns, we have developed a set of integrated software tools and employed them in *two case studies* that deal with real-life business processes of companies in two different industry sectors. The development of these tools was carried out as a part of an ongoing work on the implementation of a comprehensive software environment for BP compliance management. The tools assist the definition of compliance concepts and their relations to BP specifications, and allow constructing graphical pattern-based expressions of compliance requirements and automatically generating corresponding formal rules.

The remainder of the paper is organized as follows: Section 2 briefly summarizes related research on BP compliance focusing on studies emphasizing compliance and relevant conceptual models. Section 3 gives an overview of the proposed approach for managing BP compliance and outlines the context of the work in this paper. Section 4 introduces the case process used as the running example throughout the paper. Subsequently, Section 5 describes the compliance conceptual model. Section 6 presents and discusses the implementation of the approach and the case studies conducted. Conclusions and research outlook are outlined in Section 7.

2 Related Work

With the increase in attention paid to the role of compliance within business processes, several works have been produced in the area of compliance management, attempting to address the current needs of organizations. Notably, the COSO (1994) framework is an early work introduced as a key guidance to establish internal control mechanisms in organizations. The COSO framework does not propose a model to describe compliance concepts, however, it elucidates the way the organization progresses from objectives, abstract requirements, to controls instituted into the processes. Other initiatives, such as COBIT (2007) and OCEG's GRC (2009) provide a governance model with control objectives for particular domains to help organizations to refine concrete controls. However, similarly these models do not provide explicit guidance addressing how compliance concepts and their interrelationships are defined and integrated. Risk management is a key component in managing compliance (ERM, 2004) and there are few works on conceptualizing risks and processes (Strecker, Heise and Frank, 2010; Zur Muehlen and Rosemann, 2005). However, these works fail to address several concepts, such as compliance requirements, sources and concerns, which are particular to BP compliance.

On the specification of compliance requirements, Sadiq et al. (2007) proposes an approach for modelling control objectives within BP structures. Their work is one of the few works that actually introduce a basic model to capture compliance requirements. In order to realize what Sadiq et al. (2007) refers to as 'compliance-by-design', BP models are enriched with, so-called, 'control tags'. They propose a modal logic based approach using Formal Contract Language (Governatori, Milosevic and Sadiq, 2006), which separates the prescriptive modelling of processes and the descriptive nature of compliance requirements. However, the complexity of the adopted formal language poses critical problems in practice. Similarly, a number of approaches and technologies have been developed, proposing a separate BP modelling and compliance requirements modelling phases, which is followed by a model checking based approach for compliance verification (Ghose and Koliadis, 2007; Liu et al., 2007; Namiri and Stojanovic, 2007). In general, BP models

are expressed using a language or notation, such as BPMN (OMG, 2010) or WS-BPEL (OASIS, 2007), which are then transformed into formal models and verified against formally specified requirements by means of model-checking technology. Process mining techniques are also applied on process event logs or real-time data to monitor the behaviour of processes (van der Aalst, Beer and van Dongen, 2005). Possible deviations with process definition and compliance requirements can then be detected and resolved.

The majority of the approaches mentioned above are limited to certain phases of the BP lifecycle, and locked into specific technologies/languages used for specifying BPs and compliance requirements. In general, the research on compliance has predominantly focused on exploratory studies, rather than proposition of solutions that can assist organizations in their compliance management regimens (Abdullah, Indulska and Sadiq, 2009).

In general, works on the use of patterns to facilitate formal specifications concentrate on specific BP concerns or a certain BP lifecycle phase. Namiri et al. (2007) focus on BP execution phase and propose a set of patterns for monitoring compliance during runtime. For static verification during design-time, Yu et al. (2006) extended the original Dwyer's patterns (1998). Extensions to original patterns are also studied for rendering real-time related properties (Gruhn and Laue, 2005).

It is also important to point out that majority of existing compliance solutions automate some part of compliance detection by generating audit reports based on specific, pre-defined checks against data pulled from enterprise applications (Sadiq et al., 2007). One of the drawbacks of these approaches is that such checks take place after a violation has occurred. Clearly, there is a need for a comprehensive framework that harmonizes automated static verification, runtime monitoring and retrospective reporting. In order to establish a sustainable solution, the framework should be based on a generic conceptual model that takes into account the requirements of various enterprise systems supporting organizations' processes, and can be applied through the BP lifecycle.

3 Business Process Compliance Management Approach

To provide a brief overview of the business process compliance management approach that we propose and to explain the role and application of the concepts introduced in this paper, this section briefly discusses the key elements of the approach with respect to the BP lifecycle. Figure 1 depicts an overview of its main practices, and highlights the parts that outline the scope of this paper.

There are two generic roles involved in this approach: (i) a *business expert,* who is responsible for defining and managing service-enabled business processes in an organization while taking compliance constraints into account, and (ii) a *compliance expert,* who is responsible for the refinement, specification and management of compliance requirements stemming from external and internal sources. In doing so, the compliance expert works *in close collaboration with* the business expert.

The approach encompasses two logical repositories: the *business process repository* and the *compliance repository*, which may reside in a same shared physical environment supported by database technology. Business process models and relevant elements (that constitute BP models) including service descriptions are defined and maintained in the business process repository, while concepts relevant to compliance requirements are managed in the compliance repository.

In the overall, the approach starts either with the BP lifecycle (the upper-part of Figure 1) or with compliance management practices (depicted in the lower part of Figure 1), which afterwards align and run together exchanging inputs and outputs. BP lifecycle starts with the analysis and design of the processes (Papazoglou and Heuvel, 2007). This involves the analysis of existing processes and the design of 'to-be' processes taking into account various factors, such as business objectives, risks, industry best practices and frameworks, and compliance requirements. The processes can be designed as high-level models using notations, such as BPMN, which is followed by the construction of detailed-level executable business process and service specifications using, for instance, WS-BPEL. Once these BP specifications are tested and have reached a steady state, they are deployed (e.g., on BP execution engines and other runtime environments) and executed. The BP executions are subsequently monitored by tracking the progress of individual process instances, so that statistics on their state and performance can be provided. Monitoring information and changes in the business environment may trigger another iteration of the cycle starting from the analysis and design phase.

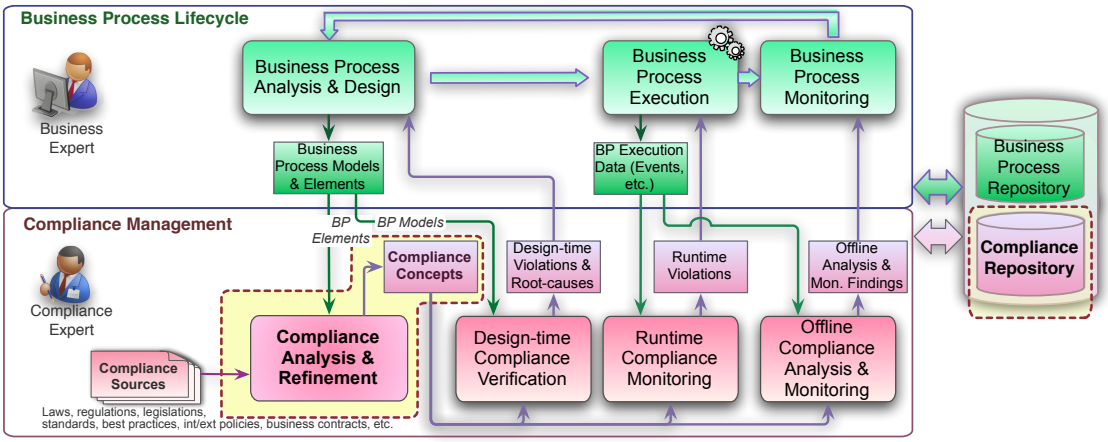


Figure 1. Overview of the Business Process Compliance Management Approach

On the other side, the compliance management practices commence with the *compliance analysis and refinement*, which involves first, the analysis of the sources of compliance requirements, such as laws, regulations, standards, policies, etc. that state the norms mandating or impacting the way the business processes are executed; second, the transformation of these abstract norms into a set of concrete concepts relevant to compliance management. Analysis and refinement requires not only compliance but also BP domain knowledge. Hence, to define and iterate an effective set of compliance concepts, the compliance expert may work together with legal experts as well as business domain experts, as the refinement requires individuals possessing the necessary knowledge of a company's business. The key output of this step is the *compliance concepts* that are stored and managed in the *compliance repository*. Among other elements, these concepts include formal compliance rules, which are the main inputs for the subsequent automated compliance assurance activities. We will elaborate compliance refinement and the specification of compliance concepts in Section 5.

The approach depicted in Figure 1 encompasses three main compliance assurance activities each having a corresponding BP lifecycle stage. *Design-time compliance verification* involves the static verification of business process models against formal compliance rules. Design-time compliance violations and possible root-causes provide key input for BP analysis and design activities to ensure that BP models progressing to execution are compliant by-design. Compliance checks at design-time are critical, as they are less costly than corresponding checks at later phases (Ly, Rinderle-Ma, Göser and Dadam, 2009). However, it is not always feasible to enforce compliance with all constraints imposed on a process models at design time. *Runtime compliance monitoring* (i.e., verifying compliance dynamically during BP execution) and *offline compliance analysis and monitoring* (i.e., verifying compliance after BP execution) are vital for a holistic view ensuring compliance throughout the remaining phases of the BP life span. During runtime compliance monitoring, the execution of the BP model instances is observed against runtime compliance rules. Like design-time verification, runtime compliance monitoring is preventive in nature aiming to detect violations before they occur. Thus, during runtime the BP execution is often stopped or paused preventing violations to arise. Violations can also be conditionally relaxed during runtime. Offline monitoring, on the other hand, involves *after-the-fact* reporting. Possible violations and trends resulting from the analysis of the execution data with respect to (offline) compliance rules are presented on monitoring dashboards in the form of various indicators.

As discussed above, the formal specification of constraints allows automated compliance verification and monitoring techniques to be applied. However, not all compliance requirements can be formally specified. For these cases, the compliance is complemented by manual controls and inspections that involve full or partial human intervention. For example, checking if "a certain type of document is secured stored in locked cabinets" involves a manual control. The approach depicted in Figure 1 presents only those assurance activities that employ automated verification and monitoring practices.

This paper focuses on the topics that are relevant to the parts in Figure 1 that are enclosed within dotted lines, i.e., the compliance analysis and refinement, and compliance concepts and repository. Our work on the other components of the approach is kept outside the scope of this paper due to reasons of scope and space.

4 Running Scenario

In this section, we introduce one of the case studies, which we shall use as a running scenario throughout this paper. The general context in which the case study takes place is the e-business applications domain, and particularly, banking applications. Taking into account the demands for strong regulation compliance schemes, such as SOX (2002), ISO 27000 (2009), internal policies and sometimes contradictory needs of the different stakeholders, such processes raise several interesting compliance requirements.

The process flow may be basically described as follows: Once a customer loan request is received, the *credit broker* checks if customer's banking privileges are suspended. If privileges are not suspended, the credit broker accesses the customer information and checks if all loan conditions are satisfied. Next, a loan threshold is calculated, and if the threshold amount is less than 1 million (M) Euros, the *post-processing clerk* checks the credit worthiness of the customer by conducting the credit bureau service. Next, the *post-processing clerk* initializes the form and approves the loan. If the threshold amount is greater than 1M Euros, the *clerk supervisor* is responsible for performing the same activities instead of the post-processing clerk. Next, the *manager* evaluates the loan risk, after which she normally signs the loan form and sends the form to the customer to sign.

Examples of compliance requirements relevant to the scenario are given in discussed in the next section.

5 Refining and Internalizing Compliance Constraints

Compliance sources are often abstract, complex and ambiguous. As a result they typically require expert interpretation and translation to concrete and organization-specific requirements. There are only few approaches that guide organizations to advance through this refinement process. Below we introduce a brief approach based on the COSO framework (1994), which is recognized by regulatory bodies as a de facto standard for establishing internal control systems. The approach involves the following major steps:

1. Analyse and interpret compliance sources in order to identify and elucidate the requirements with which the organization has to comply. These sources prescribe requirements in a range of abstraction levels from vague compliance constraints to precisely described (control) objectives.
2. Perform a 'risk assessment' to identify the risks to the achievement of these compliance requirements.
3. Design actions/statements (referred as 'controls') to mitigate the compliance risks identified.
4. Use *(compliance) patterns* to represent controls and generate (formal) rules for those controls that can be realized with a formal language and be effectively used for automated compliance assurance.

As organizations typically deal with a number of diverse compliance sources, there is a need for a structured means for capturing and organizing compliance information to support business and compliance experts particularly in the compliance analysis and refinement process. The *compliance conceptual model* proposed in this paper provides a generic and comprehensive structure tailored for compliance concepts and their relation to business processes, and establishes the underpinnings of the BP compliance management approach depicted in Figure 1. The proposed model clearly separates two domains -business processes and compliance- while establishing their relation and traceability. Managing the traceability mainly involves tracing compliance requirements *back* to their sources, or *forward* to the processes that enforce them (Koliadis and Ghose, 2008). Bi-directional traceability is important as it helps to recognize the implications of changing requirements and processes. It allows analysing why a particular decision in a process was made and what the implications of changing these specifics are in relation to the compliance requirements. Figure 2 depicts an overview of the key concepts in compliance model. In the following, we describe the key constructs both in the business process and compliance domains in line with the refinement approach we depict above.

As the focus of this paper is on compliance, we assume a *generic* model for the BP domain also in order for the proposed model not to be constrained for a particular BP modelling notation or approach. We assume *business processes* are designed as a collection of *process elements*. An organization might want to capture different aspects of their processes. Depending on these aspects, process elements may take diverse forms including the *basic elements*, such as activities, events, and business objects; *and others*, such as roles, org. units, software systems, and goals. Business processes and process elements are instantiated during process

execution to achieve process goals. These instantiations are also subject to compliance requirements (through the elements they are instantiated from) as many of these requirements are dependent on runtime conditions.

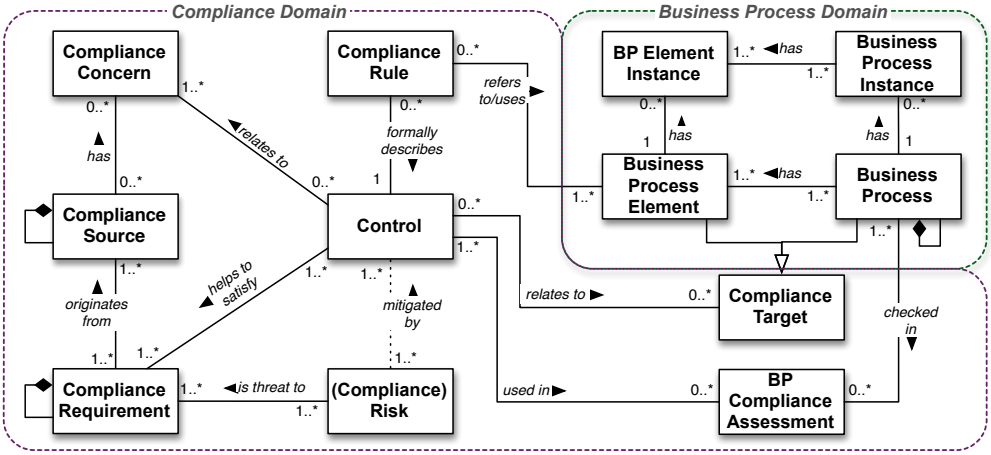


Figure 2. Key concepts of the business process compliance model

A *compliance source* is the origin of compliance requirements. It can be in the form of a regulation, legislation or law, such as SOX (2002), HIPAA (1996); standards or code of practices, such as ISO/IEC 27000 (2009) series; internal or external policies; or business partner contracts. A source typically consists of a set of sections or clauses in a hierarchical form. Interpretation of these sources results into *compliance requirements* expressed at various abstraction levels. A *compliance requirement* is a constraint or assertion that prescribes a desired result or purpose to be achieved by factoring actions or control procedures in processes. It can be prescribed in the form of abstract constraints or control objectives (COBIT, 2007).

Performing risk assessment to identify the *risks* influencing the achievement of these requirements/objectives is one of the key components of compliance management. A *risk* is the probability of occurrence of an event that might influence the achievement of certain goals (Strecker et al., 2010), which in turn might impair the organization's business model, reputation and/or financial condition. A risk is usually measured as a combination of impact and probability of occurrence (COBIT, 2007). Risk assessment also generates a set of *(internal) controls* to *mitigate* the risks and to ensure an effective implementation of the compliance requirements (COSO, 1994). A control describes the restraining or directing influence to check, verify or enforce rules to satisfy one or more compliance requirements. Failure to address controls increases the likelihood of a (*compliance) risk* to materialize. Controls are typically concrete and organization-specific.

Table 1 presents a selection of compliance requirements and elements including risks, controls and sources applicable to the loan origination process that we introduced in Section 4, which are resulted from the compliance analysis and refinement activities performed in our first case study.

ID	Control	Comp. Requirement	Risk	Comp. Source
C1	Customer bank privilege verifications (to be performed by Credit Broker) are segregated from credit worthiness checks	Duties in *processing loans* shall be adequately segregated	- Loan granted with inadequate level of assurance - Fraud/misuse - Financial loss	- SOX Sec. 404 - ISO/IEC 27002-10.1.3 - Internal Policy
C2	If the loan request's credit is above 1 million Euros, the Clerk Supervisor of Credit Operations checks the credit worthiness of the customer.			
C3	The branch office Manager checks risks and profitability of the loan request and makes the final approval (or rejection) of the request.			
C4	The customer receives an automated email notification when his personal data is collected by the "Credit Bureau service".	Customer's personal data shall be handled confidentially	- Legal penalty due to non-compliance with laws	- EU Directive 95/46/EC (Data protection)
C5	The offer in the signed loan contract is valid for 7 working days and afterwards it is closed.	Customers shall receive a certain period of time for responding to offers	- Financial loss due to changing conditions	- Internal policy

Table 1. Selected compliance requirements from the running case scenario

Controls are related to different aspects of compliance and can be grouped into *compliance concerns*, such as security, privacy, segregation-of-duties, access-rights/authorizations, management reviews, etc. Concerns often crosscut business processes. A compliance source may enforce controls in diverse concerns, and solutions addressing a certain concern usually handle relevant controls in similar ways.

A key *touchpoint* between the compliance and the business process domains is the link between the *controls* and the *compliance targets* (as shown in Figure 2). A *compliance target* is an abstract concept representing a generic 'object' of compliance requirements. They are in the form of business processes or process elements. A control applies to compliance targets and their properties.

A *BP compliance assessment* is performed to verify and ascertain that an organization is designing and executing processes that satisfy the compliance requirements applicable to them. It involves checking (during design-time, runtime, and offline) whether a set of compliance targets conforms to applicable rules with the purpose of identifying if and how a target can be changed to make it (more) compliant.

Only those controls that can be implemented and checked effectively through automated compliance assurance should be formally specified into *compliance rules.* A rule usually takes the form of 'if-then' statement consisting of a set of conditions as its antecedents, and one or more conclusions. BP elements and their attributes are the building blocks of the conditions and conclusions in rules. For example, a *control* mandating "orders above 10,000 Euros to be approved by supervisors" can be formally specified using Linear Temporal Logic (LTL) as follows: $G(Order.Amount > 10000 \rightarrow F(ApproveOrder.Role(Supervisor))$.

Despite their undeniable value-added for *automated* compliance assurance, formal specifications are difficult for end-users to understand, specify and adapt. To help in hiding the complexity of formalisms, we advocate the use of *(compliance) patterns*. In the next section, we introduce a set of patterns that help compliance/business experts with minimal/no knowledge on formalisms, to specify semi-formal representations of controls using patterns and to generate formally specified compliance rules.

5.1 Using Patterns to aid Formal Specification

Patterns are high-level domain-specific templates used to represent desired properties and constraints. For example, the expression "Create_Order *LeadsTo* Approve_Order" is built using the *LeadsTo* pattern so it respects the norm that requires 'Approve_Order' to follow 'Create_Order' activity. Compliance/business experts use these patterns to design *pattern-based expressions* as an intermediate specification between *controls* and formal *compliance rules.* These expressions are agnostic to any particular formal language, yet can be automatically transformed into a set of formal statements based on the mapping scheme designed between the patterns and an appropriate formal language of choice.

Pattern-based expressions are basically defined using *patterns* and *operands*. Operands take the form of BP elements (such as activities, events, business objects, etc.), their attributes, or conditions on them. Following the same example, the expression "Create_Order *LeadsTo* Approve_Order" has two operands (activities in this case) connected via the *LeadsTo* pattern. Expressions and operands can be combined and nested via Boolean operators (e.g., and, or, xor) to address the definition of complex controls. A pattern can be *atomic*, dealing with the occurrence and ordering of process elements; *composite,* which is built up from combinations of two or more atomic patterns via Boolean operators; or *timed*, used in combination with atomic & composite patterns to handle time-depended constraints.

Table 2 lists a set of key patterns introduced in this paper (due to space limitations only commonly used key patterns are included). The *atomic* patterns listed in Table 2 (upper part) are founded on property specification patterns proposed by Dwyer (1998) (the exception is the *XleadsTo* pattern that we introduce in this paper). Atomic patterns can be used directly for the specification of certain types of compliance rules. However, we analysed a number of compliance sources and frameworks, and worked on a wide range of requirements in the case studies we conducted (refer to Section 6) in order to discover and design new patterns for recurring types of constraints particularly in the compliance domain. These additional patterns are listed in Table 2 as *composite* and *timed patterns*.

	Pattern	Description (Given *P*, *Q*, *Y* and *Z* as states representing certain BP elements (activity, event, business object, etc.) or conditions based on them (e.g., createOrder, 'Order.Amount > 10,000').
ATOMIC	P Exists	Describes a condition that necessities the existence of *P*.
	P *Absent*	Describes a condition that necessitates a portion of a system to be free of *P*.
	P *Universal*	Indicates that *P* occurs/valid throughout the scope of the compliance target.
	P *Precedes* Q	A given *P* must always be preceded by a given *Q*.
	P *LeadsTo* Q	P must always be followed by *Q*.
	P XLeadsTo Q	A strict case of the LeadsTo pattern, which requires P to be directly followed by *Q*.
COMPOSITE	P *SegregatedFrom* Q	(Activities) *P* & *Q* should be assigned to different roles and performed by distinct actors.
	P *SSE* Q	(Activities) *P* and *Q* should be performed by different actors (SSE: Second Set of Eyes).
	Contrary-To-Duty (CTD)	The expression 'P *LeadsTo* Q *CTD* Y *CTD* Z …' indicates: If condition *P* is true then *Q* should take place; if *Q* cannot be satisfied, then *Y* should take place (which compensates the violation of *Q*), and if *Y* is violated then *Z* should take place and so on.
	P *Inclusive* Q	The presence of *P* mandates that *Q* is also present.
	P *Prerequisite* Q	The absence of *P* mandates that *Q* is also absent.
	P *Exclusive* Q	The presence of *P* mandates the absence of *Q* and vice versa.
	P *MutexChoice* Q	Either *P* or *Q* exists but not any of them or both of them.
TIMED	*Within k*	Specifies the occurrence of *P within* a given time span *k*. E.g., "P *LeadsTo* Q *Within* k" indicates that (activity) *Q* has to follow *P* within *k* time units after (the completion of) P.
	AtLeastAfter k	Specifies the occurrence of P *after* a given time span. E.g., "P *LeadsTo* Q *AtLeastAfter* k" indicates that (activity) *Q* has to follow *P after k* time units after (the completion of) *P*.
	ExactlyAt k	Specifies the occurrence of *P at* a certain time. E.g., "P *Exists ExactlyAt* k" indicates that *P* must occur at time *k*, starting from the start state of the business process model.

Table 2. A selected set of atomic, composite, and timed patterns

The mapping scheme between patterns and a formal language enables automated transformation of pattern-based expressions into a set of formal statements as compliance rules. Each rule is then used as input to subsequent design-time, runtime, and/or offline compliance assurance activities. Table 3 gives an example of a control, its pattern-based expression, and generated formal compliance rules (due to space limitations, only one of the controls –C1- from Table 1 is considered). LTL is used as the formalism to exemplify the representation of the rules. LTL is a logic used to formally specify temporal properties of software or hardware designs (Pnueli, 1977). In LTL, *G* and *F* correspond to the temporal operators 'always' and 'eventually' respectively. *G* denotes that formula *f* must be true in all the states of the BP model. *F* indicates that formula *f* will be true at some state in the future.

Control	Pattern-Based Expression	Generated Compliance Rule (in LTL)	Comp. Rule Description
C1: Customer bank privilege verifications (to be performed by Credit Broker) are segregated from credit worthiness checks	P1.1: VerifyBankingPrivilege.Role(CreditBroker) ***SegregatedFrom*** CheckCreditWorthiness	*CR1.1: F(*VerifyBankingPrivilege.*Role(*CreditBroker*))*	Activity1 shall exist
		CR1.2: F(CheckCreditWorthiness)	Activity2 shall exist
		*CR1.3: G(*VerifyBankingPrivilege.*Role(*CreditBroker*)* → *F(*CheckCreditWorthiness*)*	Act.1 shall lead to Act.2
		CR1.4: G(¬CheckCreditWorthiness*) W* VerifyBankingPrivilege.*Role(*CreditBroker*))*	Act.2 shall precede Act.1
		*CR1.5: G(*VerifyBankingPrivilege.*Role(Role1)* → *G(*¬*(*CheckCreditWorthiness.*Role(Role1)))*	Two activities shall be assigned to different roles
		CR1.6: G((VerifyBankingPrivilege.*Actor*(*Actor 1*)→ *G(*¬(CheckCreditWorthiness.*Actor*(*Actor1*))	Two activities shall be performed by different actors

Table 3. Pattern-based representation of a control and generated rules (for the running scenario)

The mapping that we have defined also takes *implicit logical rules* into consideration to facilitate the analysis of violations and to provide better insight into their root-causes and remedies for their avoidance or recovery. The details regarding the mapping scheme including the generation of implicit rules and the analysis of root-causes of violations are described in detail in (Elgammal, Turetken, van den Heuvel and Papazoglou, 2010).

6 Case Studies and the Implementation

The utility of a design artefact must be rigorously demonstrated via well-executed evaluation methods (Hevner, March, Park and Ram, 2004). Observational methods, such as case studies and field studies, allow an in-depth analysis of the artefact and the monitoring of its use in multiple projects within the technical

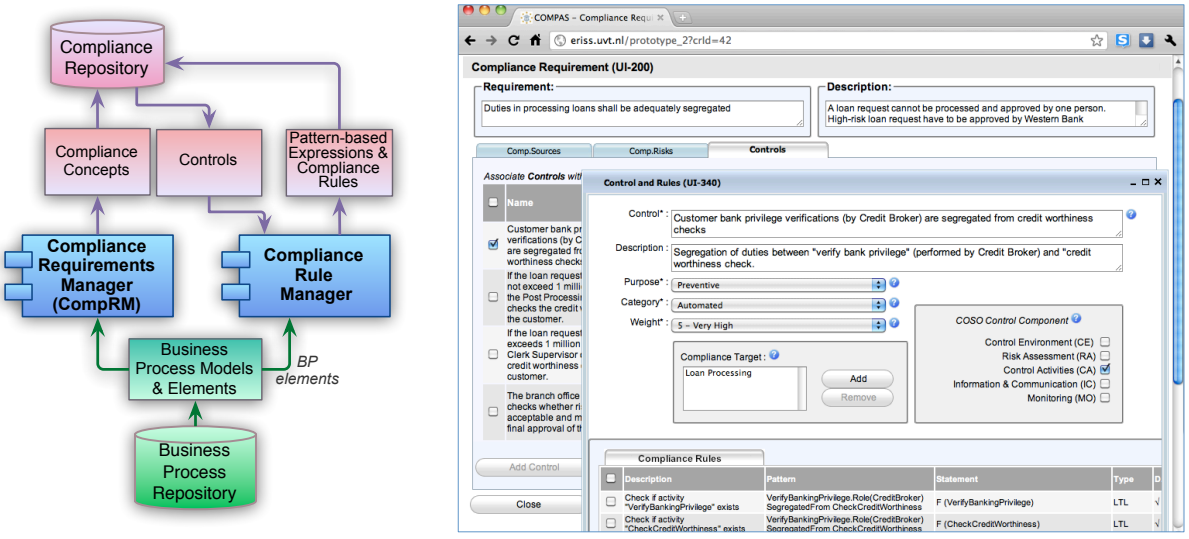
infrastructure of the business environment. We considered the 'case study' as an appropriate research strategy to help ascertaining the soundness of the concepts proposed in this paper and conducted two case studies in the *e-business* and *banking* domains. We selected these cases mainly due to the strong and diverse regulatory compliance regimes exerted in these industry sectors.

As a part of the work for implementing a comprehensive software environment for the BP compliance management framework (discussed in Section 3), we have developed a set of integrated software tools and applied them in the case studies. The first tool component -Compliance Requirements Manager *(CompRM)*- is a web-based application (accessed through http://eriss.uvt.nl/compas) for defining, storing and managing compliance requirements and relevant concepts in a compliance repository. The tool is implemented using 'PHP' as the main scripting language and Oracle database (ver.9i) for the compliance repository. The second tool – the *Compliance Rule Manager*- is a standalone application used for building graphical representations of pattern-based expressions and automatically generating corresponding formal statements. The Compliance Rule Manager is built in Microsoft Visual Studio environment using C# programming language. Figure 3(a) puts these components in context and depicts their relationships between the repositories.

In Section 4, we have introduced the first case study on 'loan processing' that takes place in the banking environment as the running scenario. The overall process involved 7 high-level requirements originating from diverse compliance sources -some of which are listed in Table 1 as examples. The second case study covered a wider range of processes (in particular: order processing, invoicing, payments, delivery, ledger maintenance) performed within an Internet reseller company that offers products through online systems. The processes are constrained by 52 requirements stemming mainly from SOX (2002), ISO/IEC 27000 (2009) and internal policies, and of diverse compliance concerns including access rights, segregation-of-duties, security, etc.

The team involved in the case studies comprised three compliance and two business experts, who worked together and followed the approach (proposed in Section 5) to refine and internalize the compliance requirements imposed on the processes and to develop graphical pattern-based expressions by using the software components mentioned above. The main objectives of the case studies were (i) to investigate the applicability of the compliance conceptual model together with the use of CompRM tool for compliance refinement and specification, and (ii) to examine the expressiveness of the patterns and to observe the feasibility of using the *Rule Manager* for building graphical representations of pattern-based expressions.

The approach followed by the case study team resulted 127 controls in total refined from 59 compliance requirements. The participants used the web-based CompRM tool to store and manage these concepts. Figure 3(b) presents a user interface from CompRM tool depicting one of the compliance requirements and a control (C1 in Table 3) defined in the first case study.



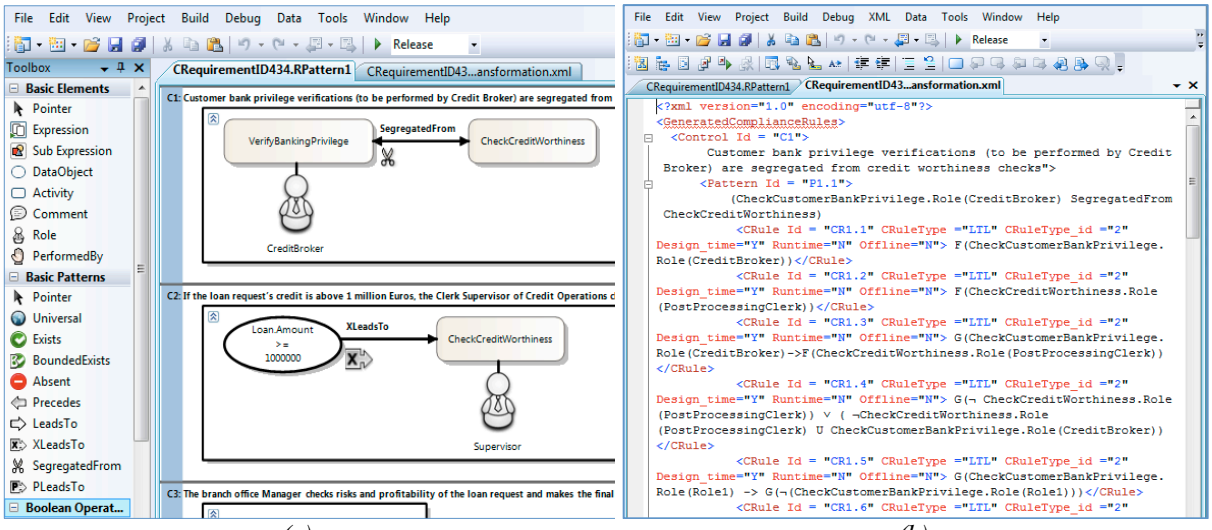
(a) *(b)*

Figure 3. *(a) A general architectural view of the CompRM and Compliance Rule Manager Components (b) A user interface from the CompRM*

Next, the compliance experts analysed the controls that were identified in order to assess if pattern-based expressions of these controls can be constructed using the patterns that we proposed. The participants were able to express 118 controls (out of 127, i.e., 93%) using the Rule Modeller tool. Nine controls that could not be expressed using available patterns required either manual inspections or were relevant to certain compliance concerns; in particular, data integrity, encryption, and data retention (e.g., controls that require customer data to be retained for a certain period). Further analysis of these cases revealed that these concerns are often not encoded explicitly within BP specifications but are typically addressed by means of crosscutting solutions (e.g., use of specific tools for data retention policies and encryption, etc., which are often implemented as 'application controls' (COBIT, 2007)). In addition, among 118 controls that were expressed using patterns, there were 28 controls that could be checked automatically but required human intervention for guaranteed assurance. This type of a control, for instance, requires a document to be reviewed by management for accuracy. A formal rule can enforce this review to be performed but cannot fully assure the completeness and correctness of the check.

Despite the limitations discussed above, we may conclude that proposed patterns are effective means for expressing compliance requirements of diverse concerns. In particular, the concerns relevant to control-flow, information (business objects, etc.), resource (roles, actors, etc.) and temporal aspects of BPs were accurately expressed with patterns. In our case studies, these types of requirements constituted 93% of the requirements. Figure 4(a) presents a screenshot of the user interface of the Rule Manager, which shows the graphical representation of controls C1 and C2 listed in Table 1. The generated rules for the same examples that are transferred to the compliance repository are shown in Figure 4(b). Empirical tests on the usability of the Rule Manager were not conducted, as one of the main objectives of the case study was to investigate the expressive power of the patterns. However, in view of the early responses from the case study participants, we expect the graphical environment and automated transformation features of the Rule Manager to bring about significant increases on the degrees of the efficiency and usability over the use of formal languages in practice.

The case studies also revealed some limitations of the overall approach. It is confirmed that the type and coverage of the formal compliance rules that can be used for automated compliance verification and monitoring depend not only on the expressive power of the patterns but also on the extent of the information encoded within BP specifications. For example, a compliance rule implementing a control that involves *roles* or other organizational units cannot be verified if the BP specification under consideration does not incorporate process elements that capture these aspects. Thus, the granularity and formality level of the BP specifications in different phases of the lifecycle and the languages used for their specifications pose limits on the rules that can be used for their verification and monitoring.



(a) *(b)*

Figure 4. *(a) A screenshot from the C. Rule Manager: Building graphical pattern-based expressions (b) Generated compliance rules to be transferred to the compliance repository*

7 Conclusions and Outlook

Business processes constitute the foundation for organizations, and as such, are impacted by industry regulations. To cope with the challenges brought by these directives, an integrated BP compliance management framework that provides effective means to ensure compliance for the full BP lifecycle becomes increasingly important. Unlike ad-hoc solutions, such a framework should not hinder but augment the ability of the organization to effectively react to changing business environment.

As one of the foundational building blocks of this framework, this paper proposes a generic *compliance conceptual model* for the definition of compliance concepts, and introduces a set of *patterns* to facilitate the specifications of formal compliance rules to be used for automated compliance verification and monitoring. In order to evaluate the applicability and feasibility of the proposed concepts, we implemented a set of tools and used them to conduct two case studies involving enterprise processes and various requirements of different compliance concerns. In general, the compliance model, patterns and tools were successfully employed in the case studies allowing compliance information to be effectively captured and majority of the compliance requirements to be expressed using the proposed patterns. The construction of pattern-based expressions is supported by a tool enabling graphical representations and automated transformations from patterns to formal rules, which we believe helped to improve the degree of usability to the end users. The case studies also uncovered limitations of the overall approach. For instance, patterns failed to address requirements in certain compliance concerns. Identification and integration of new patterns to address these concerns and provide a full coverage is an intrinsic part of our ongoing research work.

The contributions of this paper form the core building blocks for the overall approach discussed in Section 3. Our research and development work is ongoing in several directions to support the entire framework. The validation of the proposed concepts, and the usability and efficacy of the tools should further be intensified by their application on various case studies and empirical tests on prospective users. Compliance entails different aspects and requires knowledge of various domains. Future case studies should also consider further evaluating the efficacy of the solutions in different domains (such as health, safety, environment, etc.) to better address their unique requirements.

The model proposed in this paper incorporates not only 'pure' compliance concepts but also related concepts, such as process and risk, which have been studied and defined in-depth in frameworks such as ARIS (Scheer, 2000) and MEMO (Frank, 2002). Future work will be directed towards factoring the compliance model into such frameworks, making them "compliance-aware" and capable of addressing a wider spectrum of organizational knowledge.

Acknowledgements

The authors gratefully acknowledge PricewaterhouseCoopers (NL), Thales Services (FR), and other COMPAS Project Partners (Ref.FP7-215175) for their effort in providing and participating in the case studies and scenarios, and their valuable contributions. The authors are also grateful to Mathijs van der Paauw for his effort in the development of the software components of the compliance management environment.

References

Abdullah, N. S., Indulska, M. and Sadiq, S. (2009). *A study of compliance management in information systems research*, ECIS'09, Verona, Italy, pp. 1-10.

COBIT (2007). *Control Objectives for Inf. and related Technology - COBIT, 4.1. IT Governance Institute.*

COSO (1994). *Internal Control – Integrated Framework. The Committee of Sponsoring Organizations of the Treadway Commission*.

Directive 2008/30/EC (2008). Directive 2008/30/EC: Amd. to 2006/43/EC on statutory audits of annual accounts and consolidated accounts, The European Parliament & The Council of the EU.

Dwyer, M., Avrunin, G. and Corbett, J. (1998). Property Specification Patterns for Finite-State Verification, 2nd International workshop on Formal Methods on Software Practice, USA, pp. 7-15.

Elgammal, A., Turetken, O., van den Heuvel, W. and Papazoglou, M. (2010). *Root-Cause Analysis of Design-time Compliance Violations on the basis of Property*, ICSOC'10, San Francisco, pp. 17-31.
ERM (2004). *Enterprise Risk Management – Integrated Framework. The Committee of Sponsoring Organizations of the Treadway Commission*.
Ernst & Young (2010). *The Top 10 Risks For Business*, The Ernst & Young Business Risk Report 2010.
Frank, U. (2002). *Multi-perspective enterprise modeling: Conceptual framework and modeling languages*, The Annual Hawaii Int. Conference on System Sciences (HICSS), Honululu, HI, pp. 72–82.
Ghose, A. and Koliadis, G. (2007). *Auditing Business Process Compliance,* In Service-Oriented Computing – ICSOC07, Vol. 4749, Springer Berlin / Heidelberg, pp. 169-180.
Governatori, G., Milosevic, Z. and Sadiq, S. (2006). Compliance Checking Between Business Processes and Business Contracts, EDOC'06, Hong Kong pp. 221-232.
Gruhn, V. and Laue, R. (2005). Specification Patterns for Time-Related Properties, 12th Int'l Symposium on Temporal Representation and Reasoning, USA, pp. 198-191.
Hevner, A., March, S., Park, J. and Ram, S. (2004). *Design Science in Information Systems Research,* MIS Quarterly*,* 28 (1), pp. 75-105.
HIPAA (1996). *The Health Insurance Portability and Accountability Act, U.S. Congress*.
IFRS (2001). *International Financial Reporting Standards. International Accounting Standards Board.*
ISO/IEC (2009). *ISO/IEC 27000:2009 - Information security management systems — Overview and vocab*.
Kharbili, M. and Keil, T. (2010). *Bringing Agility to Business Process Management: Rules Deployment in an SOA,* In Emerging Web Services Technology, Vol. III, Birkhäuser Basel, pp. 157-170.
Koliadis, G. and Ghose, A. K. (2008). *Service Compliance: Towards Electronic Compliance Programs*, Tech. Rep. TR2008-01, Decision Systems Lab, University of Wollongong.
Liu, Y., Muller, S. and Xu, K. (2007). *A Static Compliance-Checking Framework for Business Process Models,* IBM Systems Journal*,* 46.
Ly, L., Rinderle-Ma, S., Göser, K. and Dadam, P. (2009). *On enabling integrated process compliance with semantic constraints in process management systems,* Information Systems Frontiers*,* 1-25.
Namiri, K. and Stojanovic, N. (2007). *Pattern-based Design and Validation of Business Process Compliance,* In Lecture Notes in Computer Science, Vol. 4803, pp. 59-76.
OASIS (2007). *Web Services Business Process Execution Language (WS-BPEL), Version 2.0*.
OCEG (2009). *GRC Capability Model, Ver 2.0. Open Compliance and Ethics Group*.
OMG (2010). *Business Process Model and Notation (BPMN), Version 2.0 (Beta 2)*.
Papazoglou, M. P. and Heuvel, W.-J. v. d. (2007). *Business process development life cycle methodology,* Commun. ACM*,* 50 (10), pp. 79-85.
Pnueli, A. (1977). The Temporal Logic of Programs, 18th IEEE Symposium on Foundations of Computer Science, Providence, pp. 46–57.
Sadiq, S., Governatori, G. and Namiri, K. (2007). Modeling Control Objectives for Business Process Compliance, Business Process Management, Brisbane, Australia, pp. 149-164.
Scheer, A.-W. (2000). *ARIS: Business Process Modeling, 3rd. Ed.,* Springer.
SOX (2002). *Sarbanes-Oxley Act of 2002, U.S. Congress*.
Strecker, S., Heise, D. and Frank, U. (2010). *RiskM: A multi-perspective modeling method for IT risk assessment,* Information Systems Frontiers*,* 1-17, DOI: 10.1007/s10796-010-9235-3.
van der Aalst, W. M. P., Beer, H. and van Dongen, B. F. (2005). Process Mining and Verification of Properties: An Approach based on Temporal Logic, International Conference on Cooperative Information Systems (CoopIS'05), Cyprus, pp. 130-147.
Yu, J., Manh, T., Han, J. and Jin, Y. (2006). Pattern Based Property Specification and Verification for Service Composition, Web Information Systems Engineering (WISE06), China, pp. 156-168.
Zur Muehlen, M. and Rosemann, M. (2005). Integrating Risks in Business Process Models, Australasian Conf. on Information Systems (ACIS 2005), 29 Nov – 2 Dec 2005, Sydney, Australia, pp. 62–72.