Enterprise Regulatory Compliance Modeling using CoReL An Illustrative Example

Marwane El Kharbili*, Qin Ma*†, Pierre Kelsen*† and Elke Pulvermueller‡
**Laboratory for Advanced Software Systems*
†*Interdisciplinary Centre for Security Reliability and Trust*
University of Luxembourg, Luxembourg
‡*Department of Mathematics & Computer Science, Institute of Computer Science*
University of Osnabrueck, Germany

Abstract—**Regulatory compliance management is a critical and challenging task, especially in the context of Business Process Management. It requires a comprehensive framework for dealing with compliance requirements: elicitation, modeling, static and dynamic checking and reporting. We previously defined CoReL, a domain specific language for the domain of compliance decision-making. This paper shows how CoReL can be used to model compliance requirements using an illustrative example. In particular, we show how CoReL's agnosticism of logical formalisms and coverage of enterprise business aspects leverages the task of compliance modeling to the business user level.**

Keywords-**Regulatory Compliance, Business Process, Policies, Modeling, Domain Specific Language.**

I. INTRODUCTION

Regulatory Compliance Management (RCM) consists of ensuring that a given company follows all internal or external guidance contained in a regulation document and implements it properly. Concretely, this requires ensuring that a company's structure is defined and behavior runs according to these guidelines. The main difference in our work to mainstream definitions of RCM is that we regard a regulation not only as a set of constraints, but extend this definition to contain (i) a description of the enterprise business aspects impacted by the regulation, (ii) the violations of the constraints, and (iii) ways to recover from such violations.

A regulation can be of several kinds: a contract, internal guidelines, a law, etc. Regulations themselves possess a complexity that consists of a hierarchical structure and references to the same or other regulations. Moreover, companies are usually under the jurisdiction of several regulations at the same time, which complicates even more the challenge of a comprehensive and unified RCM framework. Additionally, these regulations can be country-specific (e.g., laws) or are not equally critical to every enterprise (e.g., some client contracts is more important to fulfill than violating internal guidelines).

In order to enable a comprehensive solution to compliance management, we have proposed a model-driven and policy-based framework in [EKMKP11], which contributed CoReL as a Domain Specific Language (DSL) for compliance decision-making. In this paper, we illustrate the use of CoReL on a job application process (JAP) and discuss the insights gained from this example and their impact on future research.

In the remainder of this paper, we first discuss the nature of compliance requirements in Section II. The JAP process and associated compliance requirements are then introduced and modeled in Section III. We conclude in Section IV with a discussion of CoReL's capabilities and planned future work.

II. BEYOND CONSTRAINT MODELING: COREL'S RATIONALE

Enterprise Models (EMs) do not only express the control flow behavior of an enterprise, but also contain other enterprise information such as data flow, resource flow, and organizational elements. Example languages for such EMs include eEPC [KNS92] and BPMN[OMG11], although the latter only considers data annotations to control flows. Our aim is to define and provide a solution to RCM seen in this broader context of EMs.

A. The Semantic Gap in Existing RCM Solutions

In Figure 1, we show that classical approaches proceed so that constraints are extracted directly from regulations and formally modeled in logical formalisms. The semantic gap between the regulation domain and the formalism domain, raises at least two issues. First, the jump from regulation documents to constraints risks loss of information such as consequence of violating the constraints, different types of violations, and quantifying compliance, that mere constraints may fail to express. Second, Business Users (BUs), as key stakeholders

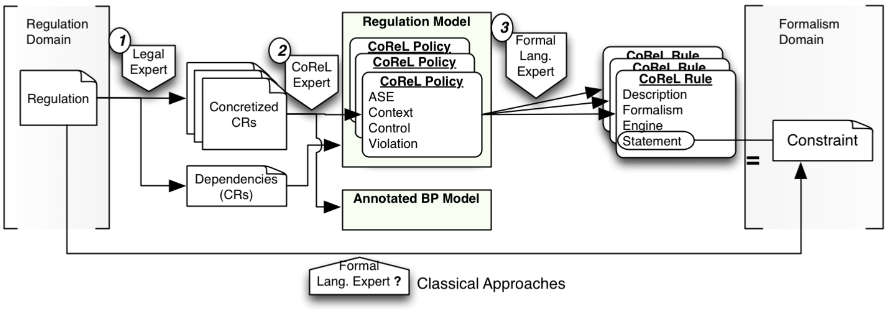

Figure 1. CoReL's Approach to RCM Modeling - Leveraging Modeling Abstraction to the Business User Level

in any RCM initiative are hardly able to work with formal languages.

In CoReL, regulation documents are first 'concretized', i.e., interpreted by legal experts and formulated for the particular enforcement context into a set of guidelines called Compliance Requirements (CRs). Every CR is modeled as a CoReL policy which is further implemented by CoReL rules. Roughly CoReL rules represent the constraints classical approaches produce. A regulation is complex because (i) logical formalisms used to express constraints are too complex to be used directly by BUs, and (ii) the number of CRs to be handled in RCM is usually overwhelming. To tackle this complexity, CoReL modularizes policies to facilitate reuse. CoReL policies consist of several building blocks to model CRs: (i) the impacted elements, (ii) the constraint placed on these elements, (iii) the associated violations and eventually, (iv) the required recovery mechanisms. In addition, CoReL provides a graphical notation and makes use of a repository to enable reuse of police building blocks (cf. Section III).

B. 2D Classification of CRs

After widely studying literature around the topic of RCM, we observed different types of CRs, which we collect and classify along two dimensions. First, CRs contain different types of constraints that require different logic formalism to be expressed, which gives rise to the definition of the dimension below.

Definition II.1 (CR Categories - Logical Formalism Dimension)**.**
1. Structural CRs *are constraints which hold over the static (structural) part of an EM. For example, a certain document's size must not exceed a certain number of pages. OCL is an example of language used to model these CRs.*
2. Temporal CRs *express temporal dependencies between execution states of a Business Process (BP). For example, if a customer deposits cash on his bank account, then this amount should eventually show up on his account balance. CTL and LTL are examples of formalisms used to model these CRs.*
3. Contractual CRs *express duties, rights and commitments that EM elements hold over each other. The obligation to pay a fee every month for a user of an online DVD rental service is an example. Contractual CRs are typically found in contracts and usually require modal logics (e.g., FCL[GZ05]) to be modeled.* □

Another challenge in RCM is the fact that grasping a company through its EM requires taking several business perspectives into consideration, such as the organizational structure, the usage of resources, the management of goals and objectives, etc. We refer to these aspects as the Enterprise Business Aspects (EBAs). Unfortunately, most existing approaches for CR modeling stay at a formal level, i.e., only one or no EBA is considered [ACPP11], [KL08], [SM02].

Definition II.2 (CR Categories - EBA Dimension)**.**
We list three types for illustration purpose:
1. Informational CRs *target the attributes describing an EM element, e.g., the size of a document that is transferred between process tasks.*
2. Resource Usage CRs *express constraints that must hold before, during and after using a resource. Examples of resources are web services, persons, databases, etc. Examples of usage are allocating, sending, printing, etc.*
3. Organizational CRs *express constraints on the organizational elements such as roles carrying out tasks, or departments where the processing of a task is located. The well-known segregation of duty (SoD) problem is an example.* □

Table I gives a broad qualitative evaluation of the maturity of existing approaches to RCM projected over the two dimensions elicited above.

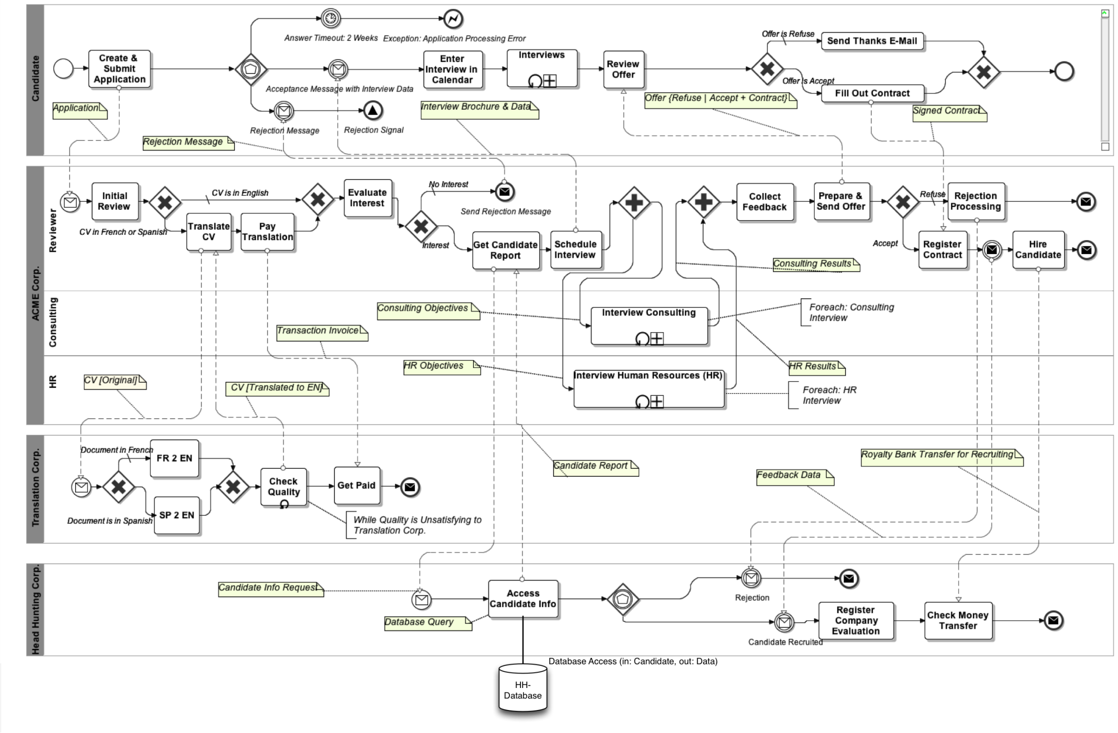

Figure 2. Job Application Process (JAP) - Example - BPMN 2.0 [OMG11]

Table I
DISTRIBUTION OF APPROACHES ALONG THE 2 DIMENSIONS

	Structural	Temporal	Contractual
Informational	● IS	◐ IT	◐ IC
Resource	◐ RS	◐ RT	○ RC
Organizational	◐ OS	◐ OT	○ OC
Legend: Degree of Satisfaction			
	●Great	◐Partial	○Unsatisfactory

III. MODELING THE JAP PROCESS CRS USING COREL

A. The JAP Example

Figure 2 presents the BPMN job application process. There are four pools representing respectively a candidate, the ACME corporation where the candidate would like to work, the Translation Corporation which is contracted by ACME to translate CVs and the Head Hunting Corporation which is contracted by ACME to deliver analysis of candidates.

We associate three CRs with the JAP process, which we introduce in Table II in natural language. Table II contains an informational temporal CR (cf. CR1), a resource usage structural CR(cf. CR2), and an organizational contractual CR (cf. CR3). We attempt in this way to provide a coverage of the 2D CR constraint space that is satisfying even though space is limited.

B. Using CoReL in the JAP Example

1) CoReL Constructs: In CoReL, every CR is represented as a policy π = (D, ASE, c_x, c_t, ν).

- D gives the modality of the policy to express whether it is a permission, a prohibition, an obligation, or a dispensation.
- ASE is a triple of three elements (a, s, e) of the EM that is be constrained by the policy, where the Subject s executes the Action a on Entity e. These elements come with data attributes describing them, which we call *qualifiers*.
- Context c_x describes the applicability condition of π.
- Control c_t describes the constraints carried by π . In case π applies (i.e., c_x is $\top$) and its constraints (i.e., c_t) are evaluated to $\top$, we say that π holds.
- Finally, ν describes the violations associated with π and the corresponding recovery mechanisms.

The context and control parts of a policy are implemented using a conjunction of rules. A rule in CoReL is a statement in one of languages supported by CoReL (e.g., CTL, OCL). Using

Table II
SET OF 3 EXAMPLES OF CONCRETIZED COMPLIANCE REQUIREMENTS

ID	Concretized Compliance Requirement
CR1	For any application coming from Europe, it always has to be the case that when applications are submitted and reviewed then these are either refused or a contract offer is made to the applicant. A violation of this will lead to an email sent to the head of recruitment of ACME. A more strict violation handling would be to force processing of the email by the head of recruitment and looking into the matter in a timespan of at most 30 days.
CR2	The applicants database of the head hunting company shall not be accessed more than 100 times a day by ACME. A violation of this will lead to an increase of the cost of head hunting services by 5% for ACME. Additionally, the process model will be modified as follows: the HH (Head Hunting Corp.) pool will be extended with a new task before accepting request from ACME, which checks the number of requests already made during the day.
CR3	For all candidates from Europe, employees holding an interviewer role are obliged to never hold a reviewer role and vice-versa. A violation of this will lead to a commission to process the case of the employee responsible for accumulating conflicting roles.

adequate verification engines, CoReL computes the ⊤ or ⊥ valuations for each rule. An user-defined compliance valuation function computes a violation value which CoReL[1] maps to a violation recovery. It is important to note that in CoReL, any deviation from the perfect behavior as specified exactly by constraints is called a violation.

Depending on which violation value has been computed for the policy, CoReL finds the associated violation recovery and executes the actions specified by the latter. This way, it is possible to associate several violation recoveries with a policy. Violation recoveries can either be handlings (penalty or reward), reparations, or compensations. An example of handling is to send an alert email or decrement the trustfulness value for an employee. The actions specified by a handling are from the BP model of the EM. Executing these actions do not change the BP model. In contrast, reparations are statements able to change the BP model itself, in a controlled attempt to impeach recurrent violations. For example, it might be possible to express that whenever a certain policy is violated ten times, then the BP model itself is extended with an additional inspection task. However, in order to keep the compliance checking process decidable, reparations must be constrained in certain ways. This however, is out of the scope of the paper. Another CoReL construct for violation recovery is compensations, which allow reacting to violations by introducing new policies, e.g., a new obligation that must be fulfilled. Informally, a policy π_1 having a policy π_2 as a violation part means that in case π_1 is violated, then π_2 must be enforced. This is the same as FCL's violation chains, which we investigate using to provide formal semantics for this part of CoReL.

Informally, a policy expresses a deontic modality over a decision on the execution of an ASE triple. For instance, an obligation (resp. dispensation) policy over an ASE triple means that when the constraint contained in the policy is evaluated to ⊤, then the ASE triple *must* (resp. *does not have to*) be executed (i.e., a possible obligation on ASE is removed). Symmetrically, a permission (resp. prohibition) policy over an ASE triple means that when the constraint contained in the policy is evaluated to ⊤, then the ASE triple *can* (resp. *must not*) be executed.

Table III
THE ACME RECRUITMENT GUIDELINES - COREL PART DEFINITION

Part	Type	Definition
		Policy Definition for CR1
ASE		(*Send*(Candidate, Application, ACME), Candidate, ACME)
Context		Application.Location in {'Europe'}
Control	IT [CTL]	**AG**[Application.reviewed() → **EF** (Contract.signed() or Offer.refused())]
		Policy Definition for CR2
ASE		(*Usage*, HeadHunting, Database)
Context		*Send*(ACME, Request, HH)
Control	RS [OCL]	**Context:** Database **Inv:** **this**.NumDayAccessCalls('ACME') ≤ 100
		Policy Definition for CR3
ASE		(ACME, nil, nil)
Context		nil
Control	OC [FCL]	∀ e:Employee ⊢ $\mathbf{OB}_e$ (Reviewer(e) → ¬'Interviewer(e)) ⊗ (ACME.Commission(e))

2) JAP CoReL Model: The modeling of the JAP CRs in CoReL is proceeded as follows. First, for each policy represents a CR, we determine the ASE triple, the context and the control. These parts are summarized in Table III. We modeled the 3 CRs using a variety of languages (OCL, CTL and FCL) on purpose, to show that some languages are

[1] When using CoReL, we mean the CoReL engine and the CoReL language interchangeably.

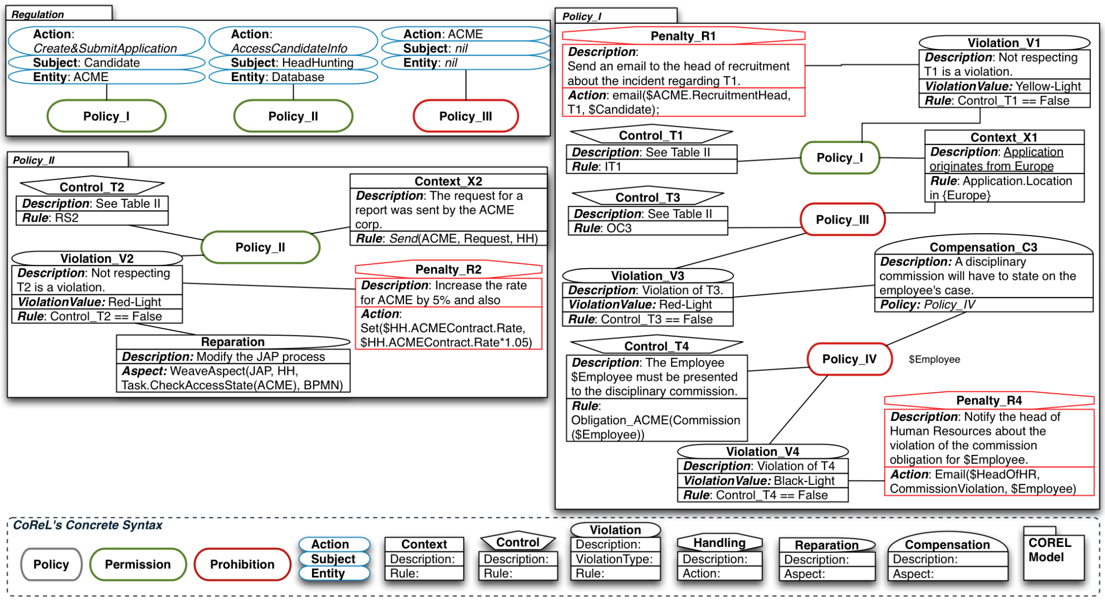
Regulation
Action: Create&SubmitApplication
Subject: Candidate
Entity: ACME
Action: AccessCandidateInfo
Subject: HeadHunting
Entity: Database
Action: ACME
Subject: nil
Entity: nil
Policy_I
Policy_II
Policy_III
Policy_II
Control_T2
Description: See Table II
Rule: RS2
Context_X2
Description: The request for a report was sent by the ACME corp.
Rule: Send(ACME, Request, HH)
Policy_II
Violation_V2
Description: Not respecting T2 is a violation.
ViolationValue: Red-Light
Rule: Control_T2 == False
Penalty_R2
Description: Increase the rate for ACME by 5% and also
Action: Set($HH.ACMEContract.Rate, $HH.ACMEContract.Rate*1.05)
Reparation
Description: Modify the JAP process
Aspect: WeaveAspect(JAP, HH, Task.CheckAccessState(ACME), BPMN)
Policy_I
Penalty_R1
Description: Send an email to the head of recruitment about the incident regarding T1.
Action: email($ACME.RecruitmentHead, T1, $Candidate);
Violation_V1
Description: Not respecting T1 is a violation.
ViolationValue: Yellow-Light
Rule: Control_T1 == False
Control_T1
Description: See Table II
Rule: IT1
Policy_I
Context_X1
Description: Application originates from Europe
Rule: Application.Location in {Europe}
Control_T3
Description: See Table II
Rule: OC3
Policy_III
Compensation_C3
Description: A disciplinary commission will have to state on the employee's case.
Policy: Policy_IV
Violation_V3
Description: Violation of T3.
ViolationValue: Red-Light
Rule: Control_T3 == False
Control_T4
Description: The Employee $Employee must be presented to the disciplinary commission.
Rule: Obligation_ACME(Commission ($Employee))
Policy_IV
$Employee
Penalty_R4
Description: Notify the head of Human Resources about the violation of the commission obligation for $Employee.
Action: Email($HeadOfHR, CommissionViolation, $Employee)
Violation_V4
Description: Violation of T4
ViolationValue: Black-Light
Rule: Control_T4 == False
CoReL's Concrete Syntax
Policy
Permission
Prohibition
Action
Subject
Entity
Context
Description:
Rule:
Control
Description:
Rule:
Violation
Description:
ViolationType:
Rule:
Handling
Description:
Action:
Reparation
Description:
Aspect:
Compensation
Description:
Aspect:
COREL Model

Figure 3. CoReL Model for the Concretized CRs in Table II

more adequate than others to express certain kinds of CRs. Then, in Figure 3 we capture the graphical CoReL model for the JAP example.

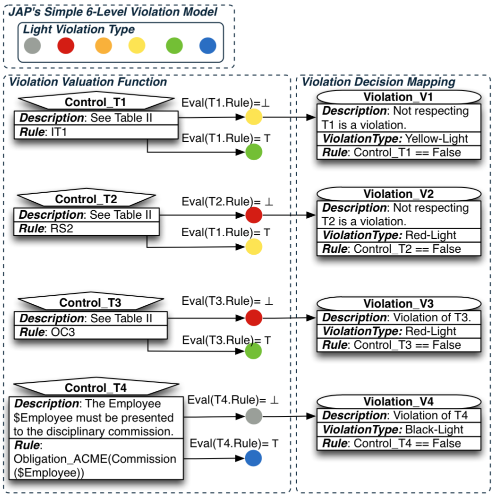
JAP's Simple 6-Level Violation Model
Light Violation Type
Violation Valuation Function
Control_T1
Description: See Table II
Rule: IT1
Eval(T1.Rule)=⊥
Eval(T1.Rule)= T
Control_T2
Description: See Table II
Rule: RS2
Eval(T2.Rule)= ⊥
Eval(T1.Rule)= T
Control_T3
Description: See Table II
Rule: OC3
Eval(T3.Rule)= ⊥
Eval(T3.Rule)= T
Control_T4
Description: The Employee $Employee must be presented to the disciplinary commission.
Rule: Obligation_ACME(Commission ($Employee))
Eval(T4.Rule)= ⊥
Eval(T4.Rule)= T
Violation Decision Mapping
Violation_V1
Description: Not respecting T1 is a violation.
ViolationType: Yellow-Light
Rule: Control_T1 == False
Violation_V2
Description: Not respecting T2 is a violation.
ViolationType: Red-Light
Rule: Control_T2 == False
Violation_V3
Description: Violation of T3.
ViolationType: Red-Light
Rule: Control_T3 == False
Violation_V4
Description: Violation of T4
ViolationType: Black-Light
Rule: Control_T4 == False

Figure 4. Violation Model, Violation Valuation (left part) and Violation Decision (right part) used in the example

In order to show the modeling functionalities of CoReL in extended fashion, we added mentions of context, penalty, compensation, and recovery statements to the concretized CRs given in Table II. We use the previously introduced CoReL building blocks (also called CoReL policy parts) to model these statements accordingly. For this, we will provide a simple violation model and a simple compliance valuation function for illustration purposes (see Figure 4). Let us model the regulation consisting of the set of three concretized CRs given in Table II.

In our example, there are no multiple violations. In Figure 4, we present the violation valuation function and violation decision mapping used by the CoReL policy given in Figure 3. The violation model described the set of violation values available for use. The valuation function maps every valuation of the controls to a violation value. In Figure 4, we only show the mappings for the ⊥ valuations of controls. Moreover, the control valuation is the same as the rule valuation as every control in our example contains one rule.

IV. Final Wrap-Up

A. Discussion & Future Work

In comparison to existing approaches, CoReL does not commit to a single rule formalism for expressing constraints. CoReL allows violations to happen, but modelers can operationally describe how to recover from violations. The example delivers early validation of CoReL, although it was not conducted as empirical research. The example allows to express no conclusions about the ability of CoReL to provide support for the two other RCM sub-problems: verification and reporting.

We illustrated how the graphical concrete syntax of CoReL helps conceptualizing the CRs as policies. Through the reusable parts of a policy's definition, a basic mechanism for reuse is offered to the modeler. We motivated the semantic difference between CRs referring to different EBAs in

Section II. We discovered for example that providing adequate support for resource usage control requires CoReL to provide a generic way of referring to resources and resource usage operations in the policy rules. We also illustrated the need for support of multiple formalisms.

One more problem this example uncovered is that the graphical concrete syntax of CoReL can be strengthened by making the rules also accessible to BUs. At the current stage of work, rules have to be provided by formal language experts and documented in order to be selected and reused by BUs in creating CoReL models. However, this rule selection process requires that the constraints expressed by the CoReL rules be part of a common understanding between the formal language expert and the BU.

Looking at research tackling this issue, we want to look at two alternatives. On the one hand, creating a repository of rules and allowing to search for adequate rules using the description field of each rule (string). On the other hand, existing research looks at creating pattern languages for rules, e.g., in LTL [ETvdHP10], and providing this pattern language with a graphical concrete syntax. Reusing results from this stream of research looks very promising to us in attaining our main objective of making regulation modeling more accessible to business users. This is not in conflict with the original aim of CoReL of providing a graphical language for compliance decision-making, which is distinct from rule modeling.

We are currently at the stage of formally defining the semantics of CoReL models, by defining an interpretation function capable of computing the violation types and deciding which recovery actions must be undertaken. However, CoReL's integration with a BP modeling language must still be defined, in particular for the eEPC and the BPMN notations. Our aim is to do static compliance checking, i.e., using model transformations, and to obtain formal representations of BPs (i.e., as Kripke models) in order to check policies which hold temporal rules. Also, CoReL still lacks an enforcement mechanism. which would allow it to do dynamic checking of policies, i.e., checking policies during the execution of a BP.

B. Conclusion

The example introduced here contains a variety of CRs. We showed how to model these CRs using our graphical compliance modeling language. We introduced some features of CoReL which shall enable business users to better apprehend the operational aspects of CR modeling, for the purpose of static and dynamic checking. This example leads us to acknowledge some strengths of CoReL. An empirical evaluation of CoReL using a modeling tool which is currently being implemented will be necessary to validate claims made about CoReL. This is planned future work, together with the static and dynamic checking of CoReL policies on BPs.

Acknowledgment

This work is done in the context of the MaRCo[2] financed by the FNR[3].

References

[ACPP11] Wihem Arsac, Luca Compagna, Giancarlo Pellegrino, and Serena Ponta. Security validation of business processes via model-checking. *Engineering Secure Software and Systems*, LNCS 6542, pp. 29–42, 2011.

[EKMKP11] M. El Kharbili, Qin Ma, Pierre Kelsen, and Elke Pulvermueller. Corel: Policy-based and model-driven regulatory compliance management. In *Proceedings of the 15th IEEE International EDOC Conference. To Appear.*, 2011.

[ETvdHP10] A. Elgammal, O. Turetken, W. van den Heuvel, and M. Papazoglou. Root-cause analysis of design-time compliance violations on the basis of property patterns. In *Proceedings of the 8th International Conference on Service-Oriented Computing (ICSOC10)*, pp. 17–31, 2010.

[GZ05] G. Governatori and Milosevic. Z. Dealing with contract violations: formalism and domain specific language. In *Proceedings of the 9th IEEE International EDOC Conference.*, pp. 46–57, 2005.

[KL08] Akhil Kumar and Rong Liu. A rule-based framework using role patterns for business process compliance. In *Proceedings of the International RuleML Symposium on Rule Interchange and Applications*, pp. 58–72, 2008.

[KNS92] G. Keller, M. Nuettgens, and A. W. Scheer. Semantische prozessmodellierung auf der grundlage ereignisgesteuerter prozessketten (epk). Technical Report Heft 89, Universitaet des Saarlandes, Saarbruecken, Germany, 1992.

[OMG11] OMG. Business process modeling notation (bpmn) specification. Technical report, Object Management Group (OMG), February 2011. http://www.omg.org/spec/BPMN/2.0/PDF.

[SM02] Andreas Schaad and Jonathan D. Moffett. A framework for organisational control principles. In *Proceedings of the 18th Annual Computer Security Applications Conference*, pp. 229–238, 2002.

[2] marco.gforge.uni.lu/

[3] Fond National de Recherche du Luxembourg