On the formal specification of business contracts and regulatory compliance

El Gammal, A.; Turetken, O.; van den Heuvel, W.J.A.M.; Papazoglou, M.

Published in:
Proceedings of the 4th International Workshop on Formal Languages and Analysis of Contract-Oriented Software (FLACOS '10)

Publication date:
2010

On the Formal Specification of Business Contracts and Regulatory Compliance

Amal Elgammal, Oktay Turetken, Willem-Jan van den Heuvel, Mike Papazoglou
European Research Institute in Service Science (ERISS), Tilburg University, Tilburg, the Netherlands
{a.f.s.a.elgammal, o.turetken, w.j.a.m.vdnheuvel, m.p.papazoglou}@uvt.nl

Today's business climate requires business processes to meet many compliance regulations, such as Sarbanes-Oxley (SOX) and to adhere to business partner contracts. In this paper, we report a comparative analysis between Linear Temporal Logic and Formal Contract logical languages, which have been successfully utilized in the literature as the formal basis of compliance requirements to enable their automatic verification.

1 Introduction

Today's business climate requires business processes to meet many compliance regulations, such as Basel II and Sarbanes-Oxley (SOX), and to adhere to business partner contracts. *Compliance* is mainly ensuring that business processes, operations and practices are in accordance with a prescribed and/or agreed on set of norms [1]. *Compliance requirement* is any explicitly stated rule or regulation that prescribes any aspect of an internal or cross-organizational business process. A comprehensive compliance management solution is of utmost importance, which must support compliance throughout all the stages of the complete business process lifecycle staring from business process design. The main focus of this paper is on *design-time* compliance management.

An automated verification of business process models against a set of relevant compliance requirements requires these requirements to be based on a formal foundation of an expressive logical language. In this paper, we report a comparative analysis between Linear Temporal Logic (LTL) and Formal contract language (FCL), which have been successfully utilized in the literature as the formal basis of compliance requirements (e.g. [1] and [2]). LTL belongs to the temporal logic family, while FCL belongs to the Deontic logic family. In Deontic logic, compliance requirements should be reduced to the set of obligations, permissions and prohibitions the enterprise has to follow in order to be considered as compliant. Deontic logic of violations (e.g. FCL) provides the ability to reason about violations, and the obligations arising in response to violation. Reparation to a specific violation can be captured by the Contrary-to-duty (CTD - $\otimes$) operator. On the other hand, in LTL each state has one possible future and can be represented using linear state sequences, which corresponds to describing the behavior of a single execution of a system. Temporal operators constitutes G, F, X, U that correspond to *Always*, *Eventually*, *Next* and *Until*, respectively. We assume prior knowledge of the syntax and semantics of the two logics (refer to [3] , [4]).

In assessing the applicability of these formal languages, we applied them on the specification of a wide range of compliance requirements emerging from different sources relevant to several industrial case studies. Our main focus is on compliance requirements stemming from legislation and regulatory bodies and business contracts. The comparative analysis is based on the capabilities and limitation of each language and a set of identified features that are discussed next. Based on the findings, we also infer which conclusions can be generalized to the whole families of Deontic and Temporal logic.

2 Required Features of a Compliance Specification Language

In order to reveal the features that should be possessed by a language to be used for the formal specification of compliance requirements, we analyzed [5] a wide range of compliance legislations and relevant frameworks including Basel II, Sarbanes-Oxley, IFRS, FINRA (NASD/SEC), COSO, COBIT and OCEG. The identified features can be summarized as follows:

- *Formality*: The specification language should be formal to pave the way for the application of future automatic analysis, reasoning and verification tools and techniques.
- *Expressiveness*: The specification language should be expressive enough to be able to capture the intricate semantics of compliance requirements emerging from different sources.
- *Usability*: The language should not be excessively complex to inhibit experts to understand and use it.
- *Consistency checks*: It is desirable for the language to provide mechanisms to identify and resolve the inconsistencies and conflicts that might arise between compliance rules.
- *Declarativiness*: Compliance requirements are commonly normative and descriptive, indicating what needs to be done [1]. Hence, a declarative language is more suited to capture these requirements.
- *Generic*: The language should enable the specification of the various types of compliance requirements (sequence, temporal, data validation and requirements, task allocation and data access rights).
- *Symmetricity*: refers to the ability to annotate business process models with compliance requirements. The annotation helps user to understand the interplay between the business and compliance specifications.
- *Non-monotonicity*: A violation to a compliance rule is not necessarily an error. Non-monotonic rules are open to violation to handle exceptional situations.
- *Normalization*: Cleaning-up of the requirements specification to identify and remove redundancies.

3 The Loan Approval Business Scenario

This scenario represents e-banking application, more specifically, the loan approval process, where compliance to strict regulations and legislations is prevalent. Two simplified compliance requirements are presented as:

R1: *Customer bank privilege check is segregated from credit worithness check* (compliance source: SOX Sec.404, ISO 17799)

R2: *If loan conditions are satisfied, the customer can check the status of her loan request infinitely often until the loan form is signed by the manager* (Bank's Internal Policy).

In FCL: $R1: CheckCustomerBankPrivilge.Role(Role1); CheckCreditWorthiness \vdash O_{Role1} \neg CheckCreditWorthiness$
$R2: Can't\ be\ represented\ in\ FCL$

In LTL: $R1': G((CheckCustomerBankPrivilge.Role(Role1) \rightarrow G(\neg(CheckCreditWorthiness.Role(Role1))$
$R2': FG(LoanConditions =' True' \rightarrow (GF(CheckLoanStatus)\ U\ SignLoanForm.Role('Manager')))$

Note that FCL is not able to express the *weak fairness* property of R2 (a constantly enabled event must occur infinitely often)[4], which is expressible in LTL. The same applies to the specification of *strong fairness* properties (an event that becomes enabled infinitely often must occur infinitely often).

4 A Business Partner Contract

A part of a sample business partner contract introduced in [3] is presented below. The business contract is between an ISP provider and a purchaser of ISP service

CONTRACT FOR SERVICES

… 2 Service Delivery

2.1 The (Supplier) shall ensure that the (Services) are available to the (Purchaser) under Quality of Service Agreement (http://supplier/qos1.htm). (Services) that do not conform to the Quality of Service Agreement shall be replaced by the (Supplier) within 3 days from the notification by the (Purchaser), otherwise the (Supplier) shall refund the (Purchaser) and pay the (Purchaser) a penalty of $1000.

2.2 If for any reason the conditions stated in 2.1 are not met, the (Purchaser) is entitled to charge the (Supplier) the rate of $ 100 for each hour the (Services) are not delivered.

In *FCL* (normalized)**:**

$r2.1 : Service \vdash O_S QualityOfService \otimes O_S Replace3days \otimes O_S Refund\&Penalty \otimes P_P ChargeSupplier$

In *LTL*:

$$\boldsymbol{r_{2.1'}}: G\Big(Service \rightarrow F\big(QualityOfService.Role(S) \wedge \neg Replace3days.Role(S) \wedge \neg Refund\&Penalty.Role(S) \wedge \neg ChargeSupplier.Role(P)\big)\Big) \vee$$
$$G\Big(Service \wedge \neg\, QualityOfService.Role(S) F\big(Replace3days.Role(S) \wedge \neg Refund\&Penalty.Role(S) \wedge \neg ChargeSupplier.Role(P)\big)\Big)$$
$$\vee\, G(Service \wedge \neg\, QualityOfService.Role(S) \wedge \neg Replace3days.Role(S) \rightarrow F(Refund\&Penalty.Role(S) \wedge \neg ChargeSupplier.Role(P))) \vee G(Service \wedge \neg\, QualityOfService.Role(S) \wedge \neg Replace3days.Role(S) \wedge \neg Refund\&Penalty.Role(S) \rightarrow F(ChargeSupplier.Role(P)))$$

As noticed from the formal specification of the contract clause presented above, LTL yields a complicated formula to capture the semantics of the CTD operator. Besides, the notion of permission is not expressible in LTL. However, the study in [6] proposes a plain extension (no change is required to the associated model-checkers) to LTL to support the specification of non-monotonic statements, which enables the specification of this requirement in LTL (e.g. *maybe* something can happen).

5 Comparative Analysis between FCL and LTL

Table 1 summarizes the results of the comparative analysis, which highlights the strengths and limitations of the two languages. The degree of support is denoted by; '+', indicating that the feature is satisfied, '-', indicating that the feature is not satisfied, and '±', indicating that the support is partial.

Table 1: Comparative Analyses of Compliance Request Languages

	FCL	Deontic Logic	LTL	Temporal Logic
1- Formality	+	+	+	+
2- Usability	-	-	-	-
3- Expressiveness	±	±	±	±
4- Declarativiness	+	+	+	+
5- Consistency Checks	+	?	-	-
6- Non-Monotonicity	+	?	±	-
7- Generic	±	?	±	?
8- Symmetric request	±	?	-	?
9- Normalization	+	?	-	-
10- Intelligible feedback	-	-	+	+
11- Tool Support	±	±	+	+

Some of these results can be generalized to the whole families of Deontic logic and Temporal Logic. For example, both FCL and LTL possess limitations in terms of *usability*. This result can be generalized to the whole families of Deontic and Temporal Logic. FCL and LTL have different expressive powers (e.g. the notion of permission is not expressible in LTL, while fairness properties are not expressible in FCL). This result is valid to all languages in both Deontic and Temporal families of logic; i.e. each language has different expressive language, and there is no language that can address all aspects. Deontic and Temporal families of logic are declarative by nature. Furthermore, FCL provides a mechanism for consistency checks by the means of the *superiority relation* of the *defeasible logic*, yet this result can't be generalized to the Deontic Logic family (denoted by '?' in Table 1). Temporal Logic family doesn't provide any support for checking consistency among formulas. Non-monotonic requirements can be expressed in FCL by means of the *superiority relation*. On the other hand, rules in temporal logic are monotonic by nature. The study in [6] extends LTL to support non-monotonic rules. FCL and LTL were capable to represent compliance requirements of the industrial case studies under consideration; however this result should be further

investigated by considering more different case studies (*Generic* metric). In FCL, by exploiting the results in [1], business process models can be visually annotated by compliance requirements using the notion of *control tags*. Model-checkers support the *counterexample tracing* facility that helps experts to resolve a compliance violation, thus providing the user with intelligent feedback, which is not addressed by the Deontic logic family. Finally, a basic strength of LTL and temporal logic in general lies in its maturity and availability of sophisticated verification tools that have been proven to be successful to verify complex systems [7].

5 Related Work and Conclusion

In [8], a comparison is conducted between three types of logics: (i) CL (Contract Language): Deontic logic, (ii) LTL and CTL: temporal logics and (iii) CSP(Communicating Sequential Processes): operational language, with respect to their expressiveness to represent three requirements. The main focus is on business contracts. Although we agree with the conclusion highlighting CL's power to represent the business contract under consideration, we disagree with the argument that states LTL's lack of support to some fairness properties. The comparative analysis conducted in this paper is generic and considers an extensive list of comparison criteria in addition to the expressiveness property.

Temporal and deontic families of logic have been successfully used in the literature as the formal foundation of compliance requirements. In this paper, we report a comparative analysis between LTL and FCL. The comparison surfaces the strengths and limitations of each language with respect to a set of identified features. Some of these conclusions can be generalized to the whole family of temporal or Deontic logic. The decision on which formal language is better used is context-dependent. Based on the nature, complexity and source of compliance requirements the user can make a decision. However from the comparative analysis, we can argue that Deontic logic is better suited for business contracts, on the other hand, temporal logic is more powerful to express regulatory compliance. An important strength in temporal logic is its maturity and its sophisticated tool support. Besides, the identified comparison criteria are not equally important. For example, the support of temporal logic to the *intelligible feedback* and *sophisticated tool support metrics* is significant. On the other hand, temporal logic for instance doesn't support *normalization*, which can't be so critical. An interesting ongoing research direction is to resolve the main problems of the temporal logic family, focusing on a specific logic (e.g. LTL) and considering the powerful features of other formalisms.

Acknowledgment: This work is a part of the research project "COMPAS: Compliance-driven Models, Languages and Architectures for Services", which is funded by the European commission, funding reference FP7-215175.

References

[1] Sadiq, S., Governatori, G., and Naimiri, K.: Modeling Control Objectives for Business Process Compliance, Proc. *BPM07*, 2007.
[2] Liu, Y., Muller, S., and Xu, K.: A Static Compliance-Checking Framework for Business Process Models, *IBM Systems Journal,* vol. 46, 2007.
[3] Governatori, G., Milosevic, Z., and Sadiq, S.: Compliance Checking Between Business Processes and Business Contracts, Proc. *EDOC 2006*, Hong Kong 2006.
[4] Pnueli, A.: The Temporal Logic of Programs Proc. *18th IEEE Symposium on Foundations of Computer Science*, Providence, pp. 46–57, 1977.
[5] COMPAS Project, Deliverable 2.1, 2008.
[6] Baral, C. and Zhoa, J.: Non-monotonic Temporal Logics for Goal Specifications, Proc. *IJCAI-07*, India, 2007.
[7] Vardi, M.: Branching vs. Linear Time: Final Showdown Proc. *7th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, Italy, pp. 1-22, 2001.
[8] Fenech, S., Okika, J., Pace, G., Ravn, A., and Schneider, G.: On the Specification of Full Contracts, Proc. *FESCA'09*, UK, 2009.