Proposition de recherche doctorale

Process and rules integration: towards semi-unattended compliance management

Mots clés :
- Directeur de thèse : engin KIRDA
- Co-encadrant(s) :
- Unité de recherche : Laboratoire de recherche d'EURECOM
- École doctorale : École Doctorale Informatique, Télécommunications, Électronique de Paris
- Domaine scientifique principal: Divers

Résumé du projet de recherche (Langue 1)

1 Problem description The need to model the organizational or business setting, in which information systems are intended to operate, is nowadays well recognized. The design process of business processes requires several different experts/stakeholders to be involved as business processes cover different aspects of an enterprise [14, 29]. Business managers, technical and business experts, juridical experts and operators that handle day-to-day operations are typically involved in the design and analysis phase of the Business Process Management (BPM) life cycle. The different people involved in the design process think and describe the same process in a complete different way. This requires different view/perspectives on the system and at the same time the stakeholders need to communicate with each other using one common language [30]. Designing a business process that satisfies the requirements of all the different persons involved is not a trivial task [31]. The result of the design process are models that support different perspectives [32]; for example the functional, behavioral, organizational and the informational perspectives [10, 7]. Designing and implementing a business process that is compliant with regulations and legislations such as those described in Basel II, COBIT, COSO, ISO and Sarbanes-Oxley is especially difficult. These regulations and legislations are often imposed by external entities such as the government. Implementing these regulations and legislations is difficult as they are expressed at an high-level of abstraction that is organization-centric, business-centric, information-centric, legal-aspects centric and human-centric [15]. This means that the abstract and high-level requirements need to be translated into business rules, compliance and security policies that can be enforced by the underlying infrastructure. This mapping is always done manually as there are no tools available to automate this process. For these reasons, designing a business process that satisfies laws, rules and legislations and implementing it on top of an IT-infrastructure is a time consuming, costly and error-prone process. 1 In order to survive in today’s business world which is characterized by fact-paced market development, sudden emergence of disruptive technologies, increased time-to-market pressure and shortened product life cycles, enterprises need to be agile with respect to business processes, partners and relations. Thus, business processes are continuously adapted as business objectives and the environment are continuously evolving. In many cases, regulations, legislations and business objectives change independently. Business processes that are compliant to rules and regulations, are designed and managed through separate activities and by several different experts which have different domain knowledge [13]. Furthermore, the mapping of abstract and high-level compliance requirements to implementable rules and policies is a manual process. Therefore, managing compliance in an ‘agile’ company is not only time consuming, costly and error-prone but also maintenanceintensive [15]. A scalable, robust and powerful solution is desired to solve the above issues. 2 Approach Several approaches exist for achieving and maintaining compliance. Broadly speaking, we can distinguish two main approaches of compliance management. The first approach is a reactive based solution. In this approach, traditional audits are carried out resulting in audit reports. Non-compliant behavior is detected after it has happened; detection takes place “after the fact”. This reporting approach can be done manually; this requires many audit checks which have to be performed by expensive consultants. In recent years, software solutions came on the market that provide some automation of this process (e.g. SAP Governance Risk and Compliance). The software solutions hook into existing ERP systems and hard-coded checks are performed against the systems [28]. Business activity monitoring and data mining are technologies used in this approach. The second main approach has a preventative focus meaning that the intention is to avoid non-compliant behavior by collecting compliance requirements using a generic requirements engineering framework and propagating these requirements into the business processes and the underlying IT-landscape, thus the approach focuses on achieving compliance by design [17]. Formal methods, tools and languages are used in this approach to check whether the business process model satisfies the compliance rules. We believe that the two approaches are complementary. A reactive based solution can detect non-compliant behavior at runtime while a preventive based solution is not able to do that; due to the inherent possibility of human or system error, compliance must also be monitored and enforced at run-time. In this thesis, we will focus on preventive based solution(s) as we believe that a company’s goal is first to avoid/prevent non-compliant behavior.

Résumé du projet de recherche (Langue 2)
2.1 Goal
The goal of our research is to develop tools that support people in achieving compliance by design. We envisage an automated tool which facilitates the development of business processes that satisfy both business requirements as well as compliance requirements. A business process model refers to a set of business requirements and compliance requirements can be expressed as a set of business rules. Modeling a business process and defining business rules can be considered as separate activities in the design process and they are carried out by different people which have different views and understandings. However, the resulting executable business process model should be a fusion of the initial business process model and the business rules which are defined by compliance experts. The automated tool will support this integration process.

2.2 Compiler
In the last two decades, several languages for process modeling have been developed or enhanced. The Business Process Modeling Notation (BPMN) is nowadays considered as being the de-facto standard for business process modeling [8]. The goal of this language is to provide a comprehensive and integrated notation that is understandable by all business users, from strategic decision makers and domain analysts to the technical developers who are responsible for implementing the technology that will support the business processes. Although BPMN has many advantages, support for process execution is considered to be one of its weaknesses. Meanwhile, the Business Process Execution Language (WS-BPEL) has been introduced to support the specification of Abstract and Executable business processes; WS-BPEL is an executable language that supports the implementation of business processes on top of a Service-Oriented Architecture (SOA) [25]. WS-BPEL code can be executed by so-called WS-BPEL engines. These engines interpret BPEL code and orchestrate Web Services accordingly. One of the strengths of WS-BPEL is that the language can be extended. An example is the BPEL4People specification which allows the specification of human tasks in a workflow [32].

WS-BPEL does not enforce the rules previously defined. To do this, WS-BPEL can encapsulate these rules (business rules) and generate the WS-BPEL code. Then, the resulting WS-BPEL code is adapted according to the initial regulations and legislations that are imposed by the government or other (external) entities. The task of the compiler is to apply or to weave the regulations or legislations, that are expressed as business rules, in the business process model resulting in executable WS-BPEL code. The compiler can return one of the following answers: • The BPMN diagram and the business rules are conflicting. The compiler is not able to find a solution that solves the conflict. The compiler returns no answer. • The BPMN diagram and the business rules are conflicting. The compiler finds one solution to resolve the conflict. The compiler proposes the solution which is represented by executable WS-BPEL code. The BPMN diagram and the business rules are conflicting. The compiler finds several alternatives to solve. For this reason, the compiler returns all the alternatives, each of them represented by executable WS-BPEL code. • The BPMN diagram and the business are not conflicting. The BPMN diagram satisfies all the business rules. The compiler translates the BPMN diagram and returns executable WS-BPEL code. The compiler may return WS-BPEL code and in some cases it will return several possible alternatives of WS-BPEL code. User involvement is then necessary to select the correct alternative. In that situation, business process experts and compliance experts should talk and in the end they should agree on one alternative. Research is needed to investigate how to translate regulations and legislations into business rules and how to weave business rules into business process models such that WS-BPEL code can be generated. We foresee that 4 not all of the business rules can be enforced in a Service-Oriented Architecture using WS-BPEL code only. However, once we have WS-BPEL code that is based on BPMN diagrams and business rules, we have knowledge about the context (business activities and the Web Services implementing them). Then, the compiler can use the same set of business rules, together with the BPEL code, to produce WSDL [6] descriptions and XACML [19] policies. The WSDL descriptions specify how a Web Service has to be invoked. Access control aspects of a compliance requirement can be expressed in one or more XACML policies which can be deployed in a Service-Oriented Architecture. We believe that these description languages and similar languages can be used to describe, specify, model, implement and enforce a subset of compliance requirements at the (Web) Service level. Further research is needed to investigate how this can be done. In many cases, it is not possible to enforce all of the business rules directly via WS-BPEL code, WSDL descriptions or XACML policies. In some cases, the compiler can simply not evaluate the business rule and then there will be no answer. In other situations, there is not enough information available in order to be able to evaluate the rule. A solution to this problem is to model the evaluation of the rule as a separate activity in the business process. This activity can be implemented by a Web Service. The Web Service can encapsulate a rule engine which enforces the rules derived from the original business rule. The compiler we proposed, is able to generate the BPEL code that belongs to this separate activity. Besides that, the compiler can generate the code of the Web Service that encapsulates and invokes a rule engine. It can even produce a framework of low-level rules which implement a high-level business rule. However, this framework will not provide a complete implementation. For instance, specifying how to collect the fact data that is necessary for the rule engine is a manual process. With manual intervention, the framework can be fully implemented and deployed on the rule engine. Although, the compiler is not able to produce a Web Service that implements a compliance control, this approach provides support to the stakeholders in the development of such Web Service. Further research is required to examine to which extend low-level rules can be generated from high-level business rules when contextual information (e.g. business process) is known.