# Controlling Compliance of Collaborative Business Processes through an Integration Platform within an E-government Scenario

Laura González, Raúl Ruggia
Instituto de Computación, Facultad de Ingeniería, Universidad de la República
lauragon@fing.edu.uy, ruggia@fing.edu.uy

## Abstract

*Compliance management is gaining increasing interest in inter-organizational service-oriented systems, which are usually supported by integration platforms. Due to their mediation role and capabilities, these platforms constitute a convenient infrastructure for controlling compliance requirements affecting inter-organizational message exchanges, which may be carried out as part of collaborative business processes (CBPs). This paper addresses compliance requirements of CBPs within an e-government scenario, by using a policy-based compliance control solution for integration platforms which was introduced in our previous work.*

## 1. Introduction

Compliance management is gaining increasing interest in inter-organizational service-oriented systems, because of the large number of regulations that have emerged during the last decades. Compliance management aims to ensure that organizations act in accordance with multiple established regulations [1]. It comprises the modeling, implementation, maintenance, verification and reporting of compliance requirements extracted from different sources [2], such as laws, technical standards (e.g. SOAP), sectorial regulations (e.g. SOX) and service level agreements.

Controlling compliance requirements (i.e. assessing their fulfillment and acting accordingly) is a major issue in these scenarios because any compliance violation may lead to the malfunction of the whole system as well as to organizations facing litigation risks, criminal and financial penalties, and losses of reputation [1][3]. Organizations are thus required to develop solutions in their systems to control the applicable requirements given that, in general, increasing the frequency of compliance audits, monitoring and reporting leads to a more effective compliance management [4]. This is specially important in e-government scenarios where organizations have to provide public services with a satisfactory quality level as well as to guarantee respect of the rights of citizens (e.g. regarding data protection).

In turn, inter-organizational service-oriented systems can be supported by integration platforms, which are specialized infrastructures providing capabilities to facilitate the integration of heterogeneous systems. This way, systems in different organizations communicate with each other by invoking services through the platform via message exchanges, which may be processed by integration flows (e.g. to perform a message transformation) in order to solve heterogeneity issues. Due to their mediation role and capabilities, integration platforms constitute a convenient infrastructure for controlling compliance requirements that affect inter-organizational message exchanges [5]. For example, a transformation may remove sensitive data from messages in order to comply with data protection regulations.

In our previous work we proposed an approach to compliance management within inter-organizational service integration platforms [6]. The approach comprises a compliance control solution, which includes a system-level compliance control subsystem (SCC Subsystem) and a policy language [7]. The language provides the means to specify how requirements have to be controlled using the components of the SCC Subsystem. This subsystem is responsible for controlling compliance by processing all messages exchanged through the platform based on the deployed policies. This control may lead to compliance actions (e.g. remove data from messages) which are based on integration platforms mechanisms.

More concretely our previous work focused on:

- the comprehensive compliance management approach for integration platforms, which enables compliance management along all the phases of the proposed life cycle and across different compliance areas (e.g. data protection) [6]
- the main elements of the compliance control solution (i.e. the SCC Subsystem and the Compliance Policy Language) [7]

URI: https://hdl.handle.net/10125/63986
978-0-9981331-3-3

- the formalization of the compliance control solution [8] using the Event-B method [9]

However this previous work did not address compliance requirements of CBPs, which is an area of increasing interest [10]. In this context, this paper constitutes a step forward in our compliance management approach as well as in the area of business process compliance [11] and e-government, by addressing requirements of CBPs within e-government scenarios and using our compliance management approach. In particular, the work focuses on controlling requirements of CBPs concerning the order of messages (e.g. specified in a choreography) and provides guidelines on how other requirements may be addressed.

The remainder of the paper is organized as follows. Section 2 describes an e-government scenario and our compliance approach. Section 3 describes how to deal with requirements concerning message order using our solution. Section 4 provides guidelines for addressing other CBP requirements and Section 5 analyzes related work. Section 6 presents conclusions and future work.

## 2. Preliminaries

This section presents an e-government motivational scenario and describes our compliance control solution.

### 2.1. E-government Motivational Scenario

The scenario is inspired by the Uruguayan e-Government Interoperability Platform (egovIP) [12], which uses a general purpose integration platform. The egovIP enables and facilitates government organizations to offer business services leveraging the web services technology. These web services, which are usually hosted on organizations' infrastructure, are exposed and invoked through proxy services deployed on the egovIP. The platform is thus able to process all service invocations and apply mediation operations to them.

For example, as shown in Figure 1, the Technical Police National Directorate (Dirección Nacional de Policía Técnica, DNPT) offers the Judicial Records Certificate Service to other government organizations in the platform. This service has an operation (i.e. hasJudicialRecords), which receives a National Identification Number (NIN) and returns whether or not the citizen with this NIN has judicial records[1].

In this context, two or more organizations may carry out collaborative business processes (CBPs) by leveraging the services available in the platform. For example, a Passport Application CBP would enable

[1]https://www.gub.uy/agencia-gobierno-electronico-sociedad-informacion-conocimiento/comunicacion/publicaciones/certificado-antecedentes-judiciales

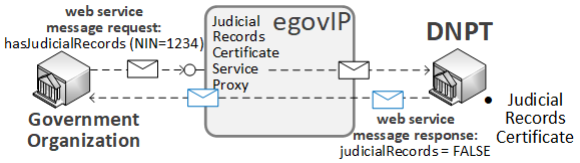


**Figure 1. egovIP - Death Certificates Service**

citizens to get or renew their passport. Figure 2 presents the message exchanges involved in this CBP using a BPMN2 choreography diagram.

The CBP starts when a citizen performs a request for an appointment through the citizens portal, hosted on the Uruguayan e-government agency (AGESIC), in order to get or to renew the passport. Then, AGESIC interacts with the Civil Identification National Directorate (DNIC) to get an appointment for the citizen, specifying a NIN, and DNIC returns a list of available dates and times (1). After the citizen selects a date and a time in the portal, AGESIC confirms these data to DNIC (2). If the appointment is not confirmed by DNIC, the collaboration ends (3). Otherwise, DNIC checks if the citizen has judicial records by interacting with DNPT (4). If DNIC does not receive a response from DNPT in twenty four hours or if the citizen has judicial records, the appointment is cancelled and DNIC informs AGESIC of this decision (7). Otherwise, DNIC informs AGESIC whether or not the citizen could get or renew the passport in the appointment (8).

Table 1 presents a summary of the services and operations involved in the Passport Application CBP.

**Table 1. Operations in Passport Application CBP**

| Organization | Service | Operation |
|---|---|---|
| DNIC | Passport | GetAvailableDates |
| | | ConfirmAppointment |
| AGESIC | Procedures Status | NotifyProcedureStatus |
| DNPT | Judicial Records Certificate | HasJudicialRecords |

Note that although most organizations, services and processes are currently part of the egovIP, the scenario was adapted for the purpose of this paper.

### 2.2. Compliance Control Solution

As shown in Figure 3, the compliance control solution proposed in our previous work [6][7][8] comprises: i) a system-level compliance control subsystem (SCC Subsystem), ii) a business-level compliance control subsystem (not relevant for this paper), iii) a compliance policy language, and iv) a formal model.

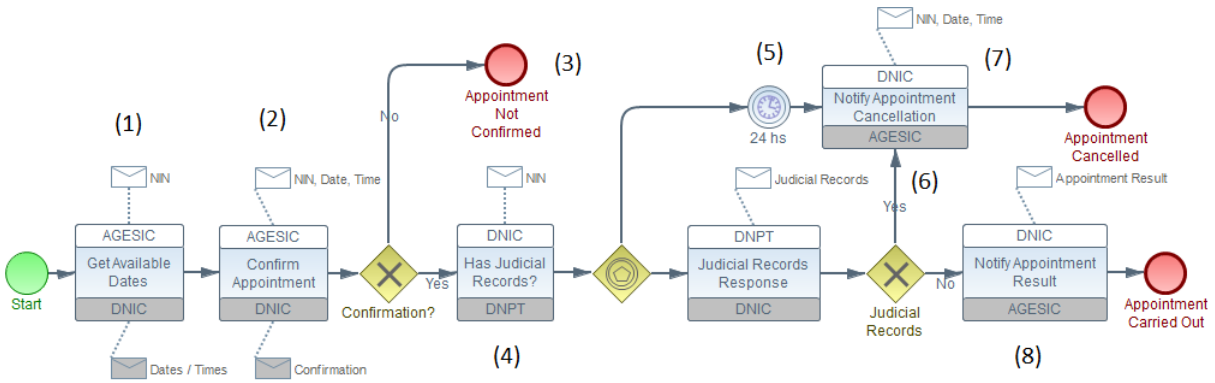


**Figure 2. Passport Application Collaborative Business Process (BPMN2 Choreography Diagram) [13]**

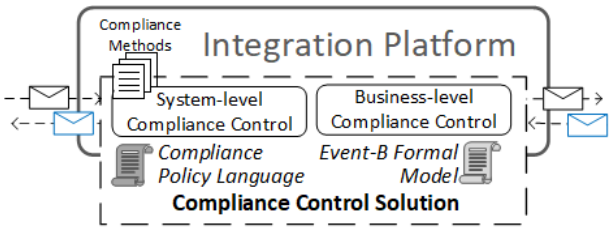


**Figure 3. Compliance Control Solution [13]**

The *SCC Subsystem* extends integration platforms and processes platform messages to control compliance at the message-level, based on compliance methods and following a policy-based approach. The SCC Subsystem has the typical architecture of policy-based solutions [14]. Its main components are: a Compliance Policy Enforcement Point (cPEP), a Compliance Policy Decision Point (cPDP), a Compliance External Services Point (cESP), a Compliance Event Monitoring Point (cEMP), a Compliance Logging Service (cLS) and a Compliance Actions Service (cAS).

The cPEP is responsible for processing messages and enforcing compliance based on the decisions made by the cPDP (e.g. reject messages). The cPDP renders compliance decisions based on compliance methods deployed on the platform and data included in cPEP requests. If the cPDP requires additional information to make decisions, it interacts with the cESP and the cEMP. When the cPEP receives a compliance response (e.g. accept) from the cPDP, it leverages other components in order to enforce compliance.

When the cPDP receives a monitored event from the cEMP or an asynchronous external service response from the cESP, the cPDP performs a similar processing as it does with compliance requests. This processing generates a compliance event which is sent from the cPDP to the cPEP and it is processed by the cPEP in a similar way as it does with compliance responses.

The SCC Subsystem supports seven decision values including: *accept* (the message exchange is compliant and it must be accepted), *allow* (the message exchange is not compliant but it must be allowed), and *verify* (the message exchange is going to be verified to check if it is compliant or not; in the meantime, it has to be allowed).

The *Policy Language for Compliance (PL4C)* [7] is geared towards enabling the specification of policy-based compliance methods that indicate how compliance requirements have to be controlled at the message level by the components of the SCC Subsystem. PL4C is inspired by XACML [14] and FACPL [15]. The abstract syntax of PL4C is specified by an Ecore metamodel and restrictions for model elements. The concrete syntax of PL4C was developed with Xtext and it is specified using the Xtext grammar language.

The *Formal Model* addresses the formalization of the SCC Subsystem using the Event-B method [9] and the Rodin platform [16]. It specifies how messages and events are processed by the SCC subsystem according to the different PL4C constructs [8].

Event-B is a modeling method for formalizing and developing systems that can be modelled as discrete transition systems [17]. It is centered around the notion of events (i.e. transitions) and its main purpose is to aid the development of systems that will be correct by construction [18]. The basis for the models in Event-B is first-order logic and a typed set theory.

The models described with Event-B are built by means of: i) contexts, which contain the static part of the system, and ii) machines, which contain the dynamic part of the system [16]. Events (i.e. transitions) describe the dynamics of machines [17]. They may contain parameters, guards (which specify the conditions under which an event is enabled) and actions (which describe how the state variables evolve when the event occurs).

The Event-B formalization of the SCC Subsystem [8] consists of a core model, which does not depend on the deployed methods, and extensions, which address the operation of the subsystem according to the deployed elements (e.g. to address requirements concerning message order). In particular, Event-B events have to be included for each method, policy and rule, among others.

In particular, for each compliance method and policy Event-B events have to be included in the model in order to evaluate its applicability and to obtain its compliance evaluation. In addition, three Event-B events have to be included in the model for each rule in order to evaluate them: i) one that fires when the condition of the rule is true, ii) one that fires when the condition of the rule is false, and iii) one that fires when there is an error that prevents from evaluating the condition of the rule.

## 3. Controlling Message Order in CBPs

This section describes how the solution presented in Section 2.2 can be used to deal with requirements of CBPs, concerning the order in which messages have to be exchanged in the context of these processes. The goal of this section is to describe the general operation of the SCC subsystem to deal with compliance requirements of CBPs as well as to propose a concrete compliance method for requirements concerning message order. In particular, we consider a compliance requirement which states that a message of type A has to be exchanged before a message of type B within each instance of a given CBP, such as the Passport Application CBP presented in Figure 2.

### 3.1. General Description of the Method

In order to control a requirement concerning message order within the SCC subsystem, a compliance method has to be implemented and deployed on the platform. In this case, the proposed method comprises four policies and two monitored events. Figure 4 presents pseudocodes of such policies and events as well as a sequence diagram showing the interactions between the components of the SCC subsystem, when controlling the requirement using the proposed method.

The first policy (i.e. PolicyForMsgA) applies to messages of type A and states the result *accept*. The policy also states that monitoring data have to be sent to the cEMP. The second policy (i.e. PolicyForMsgB) applies to messages of type B and states the result *verify*. The policy also states that monitoring data have to be sent to the cEMP. The third policy (i.e. PolicyEventNotMsgA) applies to a monitored event (i.e. EventNotA) and states the result *allow*. The last policy (i.e. PolicyEventMsgA) applies to the other monitored event (i.e. EventA) and states the result *accept*. The first event (i.e. EventNotA) is triggered when a message of type B is received and a message of type A was not exchanged before, within an instance of a given CBP. Otherwise, the second event (i.e. EventA) is triggered.

The sequence diagram in Figure 4 shows that when a message of type B is received, the cPDP returns *verify* and states a monitoring action to be performed by the cPEP. After that, an enforcement process takes place which lets the message pass through for later processing. Then, the cEMP detects one of the two monitored events and sends its information to the cPDP. According to the specification of PolicyEventNotMsgA and PolicyEventMsgA, the cPDP sends a compliance event to the cPEP with *allow* or *accept* as decisions, respectively. Finally, a second enforcement process takes place that sends compliance data to the cLS and to the BCC subsystem, which controls compliance at the business level.

### 3.2. PL4C Specification of the Method

Following the ideas of Section 3.1, we propose a compliance method in order to control the order of two types of messages (i.e. JudicialRecordsResponse, AppointmentResult) of the Passport Application CBP presented in Section 2.1. This compliance method is specified using PL4C and it is presented in Listing 1.

First, Lines 1-14 specify general properties of the method. Lines 3-6 establish that the method controls the CollaborationMessageBefore requirement over the PassportApplication CBP. Also, Lines 8-10 specify that the method is applicable to messages that are exchanged within this collaboration (i.e. CBP).

Second, lines 16-24 define two monitored events. The first one (lines 18-20) detects when a JudicialRecordsResponse message was not exchanged before an AppointmentResult message. The second one (lines 22-24) detects the opposite.

Then, Lines 28-53 specify two message policies. The first message policy applies to messages returned by the operation HasJudicialRecords (lines 28-38) and its evaluation is always *accept*. The policy also specifies a MonitoringAction (lines 36-38) in order to send monitoring data regarding the message (i.e. the message type, the collaboration instance) to the cEMP. This action has to be performed by the platform when the decision for the message is *accept*, *allow* or *verify*. The second message policy applies to messages sent to the operation NotifyProcedureStatus (lines 40-53). The policy has a rule (lines 46-53) which states *verify* as the result if the message corresponds

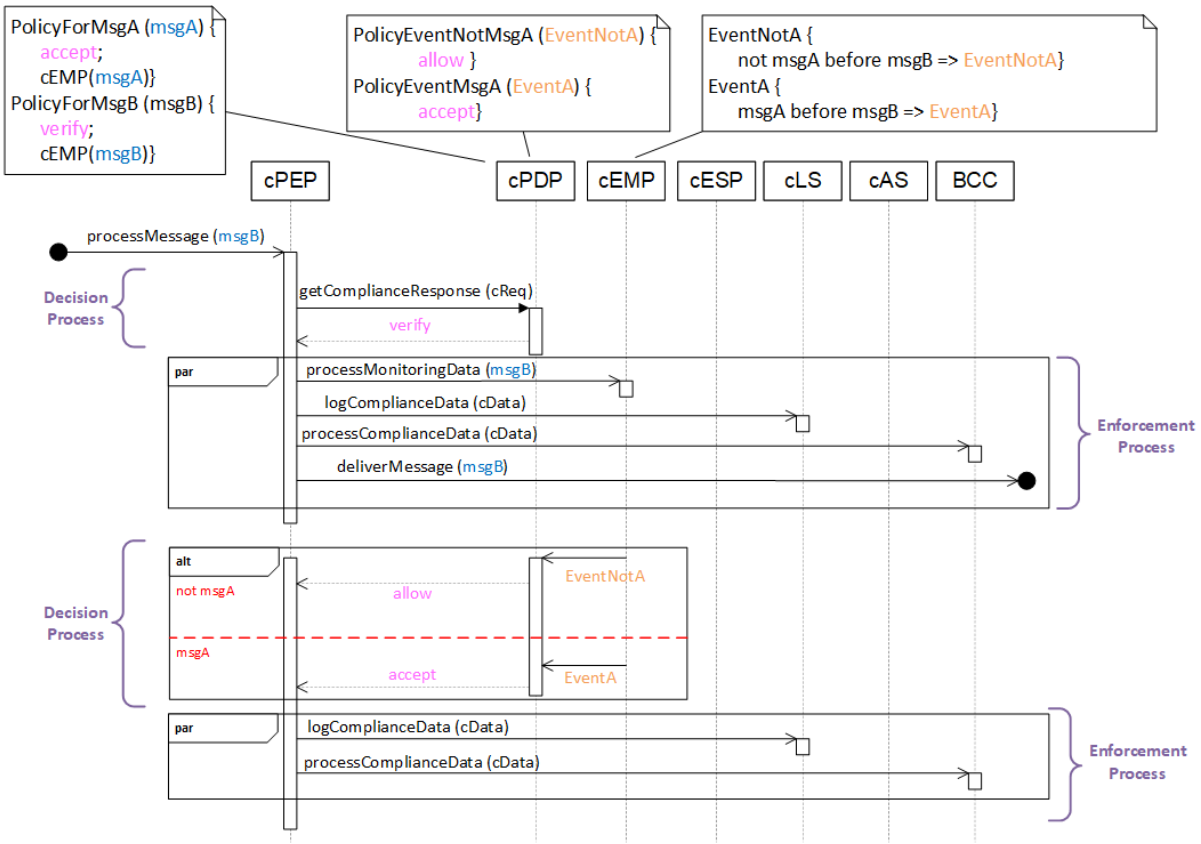


**Figure 4. SCC Subsystem - Controlling a Requirement concerning Message Order [13]**

to an ”AppointmentResult”, which is indicated by the element ”info” of the message. The policy also specifies a MonitoringAction (lines 51-53) in order to send monitoring data regarding the message to the cEMP. This action has to be performed by the platform when the decision for the messages is *accept*, *allow* or *verify*.

Finally, Lines 55-69 specify two monitored event policies. The first one (lines 55-61) applies to the NotJudicialRecordsBefore event and states *allow* as result. The second one (lines 63-69) applies to the JudicialRecordsBefore event and states *accept* as result.

### 3.3. Formal Specification of the Method

This section presents the formal specification of the proposed compliance method, which extends the SCC Subsystem formal model as summarized in Section 2.2. In particular, Listing 2 presents some of the extensions for the SCC subsystem formal model, according to the proposed compliance method (cf. Section 3.2).

The first Event-B event is used to evaluate the applicability of the proposed compliance method. The event is fired when the following conditions hold: the state of the machine is stGettingApplicableMethods (grd1), the applicability of the method has not been evaluated yet (grd2), the exchanged message is a valid one (grd3), and the message is exchanged within the PassportApplicaiton CBP. When fired, this event add the method to the set of applicable methods (act1).

The second Event-B event is used to evaluate the applicability of the first policy of the compliance method. The event is fired when the following conditions hold: the state of the machine is stGettingApplicablePolicies (grd1), the method to which the policy belongs has been evaluated as applicable (grd2), the applicability of the policy has not been evaluated yet (grd3), the exchanged message is a valid one (grd4), and the message comes from the hasJudicialRecords operation. When fired, this event adds the policy to the set of applicable policies (act2).

The third Event-B event is used to evaluate the applicability of the second policy of the compliance method and it is similar to the previous one.

The fourth Event-B event is used to evaluate the first rule of the method (i.e. RuleJudicialRecords) and it has similar guards to the previous events. Note that when this event fires, an evaluation for the rule (i.e. accept) is added to the set of rule evaluations (act2).

### Listing 1. PL4C Specification of Compliance Method [13]

```
ComplianceMethod ControlMessageOrder_PassportApplication

    controls:

        requirement "CollaborationMessageBefore"
        over object "PassportApplication"

    scope:

            toCollaboration "PassportApplication"

    algorithm:

            accept UnlessOther

    events:

        MonitoredEvent NotJudicialRecordsBefore {
            "not (Judicial Record Response)
            before (Appointment Result)"}

        MonitoredEvent JudicialRecordsBefore {
            "(Judicial Record Response)
            before (Appointment Result)"}

    policies:

        MessagePolicy PolicyForJudicialRecordsResponse
            scope:
                fromOperation "HasJudicialRecords"
            algorithm:
                accept UnlessOther
            rules:
                Rule RuleJudicialRecords (accept)
            actions:
                MonitoringAction (accept, allow, verify)
                        data(["message", "JudicialRecordsResponse"],
                            ["collabInstance", this.toCollabInstance])

        MessagePolicy PolicyForAppointmentResult
            scope:
                toOperation "NotifyProcedureStatus"
            algorithm:
                accept UnlessOther
            rules:
                Rule RuleAppointmentResult (verify
                    condition:
                        equal(getDataItem("\\info", this.msgSrvMessage),
                                "AppointmentResult")
                    actions:
                        MonitoringAction (accept, allow, verify)
                                data(["message", "AppointmentResult"],
                                    ["collabInstance", this.toCollabInstance]))

        MonitoredEventPolicy PolicyForNotJudicialRecordsBefore
            onEvent:
                NotJudicialRecordsBefore
            algorithm:
                accept UnlessOther
            rules:
                Rule RuleNotJudicialRecords (allow)

        MonitoredEventPolicy PolicyForJudicialRecordsBefore
            onEvent:
                JudicialRecordsBefore
            algorithm:
                accept UnlessOther
            rules:
                Rule RuleJudicialRecords (accept)
```

### Listing 2. Formalization of Compliance Method [13]

**metControlMessageOrderPassportApplicationApplicable** $\widehat{=}$
**WHEN**
grd1 : decisionState = stGettingApplicableMethods
grd2 : metControlMessageOrderPassportApplication $\notin$ applicableMethods $\cup$ notApplicableMethods
grd3 : message $\in$ messages
grd4 : toCollaboration(message) = PassportApplication // scope
**THEN**
act1 : applicableMethods := applicableMethods $\cup$ {metControlMessageOrderPassportApplication}
**END**

**polPolicyForJudicialRecordsResponseApplicable** $\widehat{=}$
**WHEN**
grd1 : decisionState = stGettingApplicablePolicies
grd2 : getMessagePolicyMethod(polPolicyForJudicialRecordsResponse) $\in$ applicableMethods
grd3 : polPolicyForJudicialRecordsResponse $\notin$ applicableMessagePolicies $\cup$ notApplicableMessagePolicies
grd4 : message $\in$ messages
grd5 : fromOperation(message) = HasJudicialRecords
**THEN**
act1 : applicableMessagePolicies := applicableMessagePolicies $\cup$ {polPolicyForJudicialRecordsResponse}
**END**

**polPolicyForAppointmentResultApplicable** $\widehat{=}$
**WHEN**
grd1 : decisionState = stGettingApplicablePolicies
grd2 : getMessagePolicyMethod(polPolicyForAppointmentResult) $\in$ applicableMethods
grd3 : polPolicyForAppointmentResult $\notin$ applicableMessagePolicies $\cup$ notApplicableMessagePolicies
grd4 : message $\in$ messages
grd5 : toOperation(message) = NotifyProcedureStatus
**THEN**
act1 : applicableMessagePolicies := applicableMessagePolicies $\cup$ {polPolicyForAppointmentResult}
**END**

**rulRuleJudicialRecordsApplicable** $\widehat{=}$
**WHEN**
grd1 : decisionState = stEvaluatingRules
grd2 : getRulePolicy(rulRuleJudicialRecords) $\in$ applicableMessagePolicies
grd3 : rulRuleJudicialRecords $\notin$ applicableRules $\cup$ notApplicableRules $\cup$ evaluationErrorRules
grd4 : message $\in$ messages
**THEN**
act1 : applicableRules := applicableRules $\cup$ {rulRuleJudicialRecords}
act2 : rulesEvaluations := rulesEvaluations $\cup$ {rulRuleJudicialRecords $\mapsto$ accept}
**END**

**rulRuleAppointmentResultApplicable** $\widehat{=}$
**WHEN**
grd1 : decisionState = stEvaluatingRules
grd2 : getRulePolicy(rulRuleAppointmentResult) $\in$ applicableMessagePolicies
grd3 : rulRuleAppointmentResult $\notin$ applicableRules $\cup$ notApplicableRules $\cup$ evaluationErrorRules
grd4 : message $\in$ messages
grd5 : info $\in$ dom(msgSrvMessage(message))
grd6 : msgSrvMessage(message)(info) = AppointmentResult
**THEN**
act1 : applicableRules := applicableRules $\cup$ {rulRuleAppointmentResult}
act2 : rulesEvaluations := rulesEvaluations $\cup$ {rulRuleAppointmentResult $\mapsto$ verify}
**END**

Finally, the fifth Event-B event is used to evaluate the second rule of the method (i.e. RuleAppointmentsResult) and it has similar guards to the previous events. Note that the last guard of the event (grd6) corresponds to the condition of the rule. In addition, when this event fires, an evaluation for the rule (i.e. verify) is added to the set of rule evaluations (act2).

The developed formal model enables the application of model animation mechanisms in order to verify the correct operation of the SCC subsystem in specific usage scenarios [19]. Model animation was performed using ProB[2]. In particular, the SCC subsystem model was animated in a scenario based on the compliance method presented in Listing 1 and considering the four messages shown in Table 2.

The results of the animation were the expected ones for both pairs of messages (i.e. msg11-msg12, msg13-msg14): i) all messages were delivered, ii) the evaluation of msg12 was *accept* given that a JudicialRecordsResponse message (i.e. msg11) was exchanged before within the same process instance (i.e. instance1), and iii) the evaluation of msg13 was *allow* given that a JudicialRecordsResponse message was not exchanged before within instance2.

## 4. Guidelines and Compliance Actions

This section provides guidelines for addressing other requirements of CBPs using the proposed approach. In addition, it describes how compliance actions can be used in compliance methods, in order to achieve more advanced compliance control functionalities.

### 4.1. Other Compliance Requirements of CBPs

Even though this paper focuses on requirements concerning message order, our compliance management approach may be used to control other compliance requirements of CBPs. In what follows, we analyze how different types of compliance requirements may be addressed by the proposed approach by developing policy-based compliance methods. In particular, we focus on the requirements that can be specified with the extended Compliance Rule Graph (eCRG) language proposed in the C[3]Pro project. Note that this analysis also enable a first assessment concerning the support provided by our solution in order to address compliance requirements identified in related work.

The eCRG language enables the visual modeling of compliance rules using multiple perspectives: control flow, interaction, resource, data and time [10].

The control flow perspective constrains the execution sequence as well as the occurrences of tasks within a process. The requirements of this perspective are somehow similar to the requirements concerning message order addressed in this paper, if the occurrences of tasks are considered as message exchanges. Therefore, they may be controlled with a compliance method similar to the one proposed in Listing 3.2.

The interaction perspective constrains the interactions a process may have with external partner processes. For example, it provides constructs for specifying the occurrence or absence of message exchanges as well as message flows. Compliance requirements specified with the constructs of this perspective may also be controlled by our approach in a similar way as the control flow perspective.

The time perspective provides elements for specifying constraints that require modeling points in time and time conditions. Compliance requirements specified with the constructs of this perspective may also be controlled by the proposed approach. In particular, PL4C provides time-related data types (i.e. dateTime, dateTimeDuration), monitored events and a timestamp property in messages. For example, a compliance method may be implemented to control that the temporal distance between two types of messages within an instance of a CBP is not greater than 1 hour.

The data perspective enables the specification of requirements referring to data (e.g. data conditions). These requirements are likely to be controlled by the proposed approach if they refer to data within message exchanges given that PL4C provides operators to obtain message elements as well as operators to perform comparisons. For example, a compliance method may be implemented to control that the age of a person specified in a message exchange is greater than 18 years.

Finally, the resource perspective covers several kinds of human resources (e.g. staff member, role, group, and organizational unit) as well as their relations and it enables the specification of constrains concerning the assignment of resources to tasks. Compliance requirements specified with constructs of this perspective are also likely to be controlled by our approach if they refer to message exchanges. In particular, PL4C directly supports various of the resources defined by this perspective: staff member (i.e. user), roles and organizational unit (which may be mapped to organization). For example, a compliance method may be implemented in order to control that two consecutive message exchanges within an instance of a CBP are not carried out by the same user.

[2]https://www3.hhu.de/stups/prob/

Table 2. Messages for Model Animation and Checking

| Prop. / Mess. | msg11 | msg12 | msg13 | msg14 |
|---|---|---|---|---|
| idMessageId | 11 | 12 | 13 | 14 |
| fromOrganization | DNPT | DNIC | DNIC | DNPT |
| fromService | JudicialRecords Certificate | | | JudicialRecords Certificate |
| fromOperation | HasJudicialRecords | | | HasJudicialRecords |
| toCollaboration | PassportApplication | PassportApplication | PassportApplication | PassportApplication |
| toCollabInstance | instance1 | instance1 | instance2 | instance2 |
| toService | | ProceduresStatus | ProceduresStatus | |
| toOperation | | NotifyProcedureStatus | NotifyProcedureStatus | |
| msgSrvMessage | | info: AppointmentResult | info: AppointmentResult | |
| msgTimestamp | 700 | 750 | 800 | 850 |

### 4.2. Compliance Requirements in other Areas

There are compliance requirements in other areas that although they are not directly related to CBPs, they may apply to them. For instance, in our previous work we identified the following compliance areas: Quality of Service (e.g. response time, maximum throughput, compliance with SOAP 1.2, availability), Data Quality (e.g. completeness, consistency, accuracy) and Data Protection (e.g. privacy) [6].

Several of these requirements may apply to CBPs. For instance, a compliance requirement may specify that the ”country” included in messages exchanged within a CBP has to be syntactically correct according to the Alpha-3 codes of the ISO 3166-1 standard. In [8] we proposed a compliance method to control this requirement for a single service. In order to use this method for a CBP, the scope of the method has to be changed to a CBP (e.g. Passport Application CBP).

It is important to note that our compliance management approach (cf. [6]) enables the homogeneous management of compliance requirements within different areas (e.g. quality of service, data quality, CBPs) as well as applying to different objects of inter-organizational integration platforms (e.g. the whole platform, organizations, CBPs, services, operations of services, message types). This characteristic provides a holistic view of compliance issues within inter-organizational service integration platforms. In addition, it may also contribute to detect conflicts between requirements in different areas and / or applying to different objects.

### 4.3. Adaptation and Compensation Actions

The compliance method proposed in this paper does not include any adaptation or compensation action given that it only ”detects” when the requirement is not fulfilled. Our compliance approach enables corrective as well as preventive compliance control at runtime by means of adaptation and compensation actions.

For instance, the compliance method proposed in [7], which controls a maximum throughput requirement, uses an adaptation action that defers a message a number of seconds if a service is about to receive more requests that the ones it can handle. In addition, the compliance method proposed in [8], which controls a data quality requirement, uses a compensation action that logs a non-compliance for later processing. Finally, a compliance method to deal with privacy requirements may specify an adaptation action that removes sensitive data from messages, if the person has not provided the required consents to share this information.

Listing 3 presents how these actions can be specified in compliance methods (uses:) and then leveraged in policies or rules (actions:).

Listing 3. Adaptation and Compensation Actions

```
uses:

    AdaptationActionMechanism (”DeferSeconds”)
      deferSeconds

    CompensationActionMechanism (”LogCompliance”)
      logCompliance

actions:

    CompensationAction (allow) logCompliance
        ([”certNumber”,”123”])

    AdaptationAction (repair)
        deferSeconds([”seconds”, ”1”])
```

In particular, the logCompliance action is stated when the result is *allow* and the deferSeconds action is stated when the result is *repair*.

Note that this types of adaptation and compensation actions can also be used for controlling compliance of CBPs.

## 5. Related Work

One of the most relevant related work is the COMPAS project[3], which defined a model-driven approach for runtime compliance governance in

[3]https://cordis.europa.eu/project/rcn/85292/en

the context of a process-driven SOA [1]. The approach proposed languages and tools for modeling compliance requirements in different areas (e.g. QoS, licensing, security), linking them to business processes, monitoring process execution using complex event processing (CEP) mechanisms, displaying the current state of compliance and analyzing cases of non-compliance [20]. Compared to our work this project did not focus on CBPs nor in solutions based on integration platforms, policies or e-government.

Other relevant related work is the C[3]Pro Project[4], which focused on providing a theoretical framework for enabling change and compliance of CBPs. The proposal enables compliance control at design time using model checking as well as based on the specified processes, interaction models and compliance rules [21]. It also proposed mechanisms to visually monitor business process compliance at runtime [22] and to perform a-posteriori compliance control (i.e. after execution) by processing execution logs [10]. Even though this project focused on CBPs, it did not provide solutions to control compliance based on integration platforms nor on policy-based mechanisms.

In [23] an architectural framework to verify the compliance of the overall sequence of inter-organizational choreography operations is proposed. Compared to our approach, this work does not leverage a centralized integration platform but it uses components to be deployed on each organizations. In addition, our proposal is not restricted to control the sequence of interaction as discussed in Section 4.

The Marco project[5] focused on producing a visual environment for: i) specifying norms and business processes with an underlying formal model, and ii) checking regulatory compliance of business processes. The proposal defines a Compliance Representation Language (CoReL) which is a compliance decision modeling language based on policies [24]. Compared to our approach, this project did not focus on CBPs or integration platforms. Also, the CoReL policy language focuses on business-level concerns while PL4C is geared towards specifying how components of the SCC Subsystem control compliance at the system-level.

The MASTER project[6] proposed xESB: an enhanced version of an Enterprise Service Bus (ESB) for access and usage control policy enforcement [25][26]. The ESB monitors and enforces preventive as well as reactive policies. The enforcement semantics of xESB provides the means not only to reject ESB messages that violate a policy but also to compensate that violation. Although this project proposed a specific type of integration platform to control compliance (i.e. xESB), it did not focus on CBPs and it is mainly restricted to security related requirements.

As a summary, the distinguishing characteristics of our proposal with respect to related work are: the focus on compliance requirements of CBP, the use of integration platforms capabilities (e.g. message transformation) to control compliance, policy language for specifying how compliance have to be controlled at the system-level, and the comprehensive nature of the approach which may enable organizations to deal with different compliance requirements of CBPs as well as of other elements (e.g. services) in a holistic way.

## 6. Conclusions and Future Work

This paper proposed solutions for addressing compliance requirements of collaborative business processes (CBPs) within an e-government scenario, by leveraging a policy-based compliance control solution for integration platforms. The paper focused on controlling the order of messages exchanged within CBPs according to a choreography agreed between organizations. The proposed solutions were specified with a compliance policy language (i.e. PL4C) and formalized with the Event-B method. The formalization enabled us to perform model animation in order to verify the correct operation of the SCC subsystem and the proposed compliance method.

The main contribution of this work are proposals for addressing compliance requirements of CBPs leveraging our compliance management approach within a real world e-government context. In particular, the paper detailed how the mechanisms provided by the approach (e.g. SCC Subsystem, policy language) can be used to deal with such requirements, proposed concrete solutions for requirements concerning message order within CBPs and analyzed how the approach may be used in order to address other compliance requirements of CBPs. It is important to highlight that this work extends the scope of the compliance control mechanisms for integration platforms defined in our previous work [7][8] by enabling compliance control of CBPs. It also enabled us to validate the genericity and reusability of the compliance control solution described in Section 2.2.

Future work includes the automatic generation of compliance methods for controlling requirements concerning message order based on choreography specifications (e.g. such as the one presented in Figure 2), addressing other compliance requirements of CBPs (e.g. the ones identified in [27]), developing prototypes

[4] https://www.uni-ulm.de/in/iui-dbis/forschung/laufende-projekte/c3pro/

[5] https://marco.gforge.uni.lu/

[6] https://cordis.europa.eu/project/rcn/85559/factsheet/es

in order to continue advancing in the evaluation of the technical feasibility of the proposals, and evaluating the approach with government organizations integrated to the egovIP.

## References


[1] H. Tran, U. Zdun, T. Holmes, E. Oberortner, E. Mulo, and S. Dustdar, "Compliance in service-oriented architectures: A model-driven and view-based approach," *Information and Software Technology*, vol. 54, 6 2012.

[2] M. El Kharbili, "Business process regulatory compliance management solution frameworks: A comparative evaluation," in *Proceedings of the Eighth Asia-Pacific Conference on Conceptual Modelling - Volume 130*, APCCM '12, (Darlinghurst, Australia, Australia), pp. 23–32, Australian Computer Society, Inc., 2012.

[3] A. Elgammal, O. Turetken, W.-J. van den Heuvel, and M. Papazoglou, "Formalizing and appling compliance patterns for business process compliance," *Software & Systems Modeling*, vol. 15, pp. 119–146, 2 2016.

[4] S. Sackmann, M. Kähmer, M. Gilliot, and L. Lowis, "A classification model for automating compliance," in *2008 10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services*, IEEE, 7 2008.

[5] L. González and R. Ruggia, "On controlling compliance requirements within adaptive integration platforms," in *Proceedings of the 19th Workshop on Adaptive and Reflexive Middleware - ARM 18*, ACM Press, 2018.

[6] L. González and R. Ruggia, "A comprehensive approach to compliance management in inter-organizational service integration platforms," in *Proceedings of the 13th International Conference on Software Technologies*, SCITEPRESS, 2018.

[7] L. González and R. Ruggia, "Policy-based compliance control within inter-organizational service integration platforms," in *2018 IEEE 11th Conference on Service-Oriented Computing and Applications (SOCA)*, IEEE, 11 2018.

[8] L. González and R. Ruggia, "Formalizing a policy-based compliance control solution with event-b," in *Proceedings of the 14th International Conference on Software Technologies*, SCITEPRESS - Science and Technology Publications, 2019.

[9] J.-R. Abrial, *Modeling in Event-B: System and Software Engineering*. Cambridge University Press, 1st ed., 2010.

[10] D. Knuplesch, M. Reichert, and A. Kumar, "A framework for visually monitoring business process compliance," *Information Systems*, vol. 64, pp. 381–409, 3 2017.

[11] M. Reichert and B. Weber, "Business process compliance," in *Enabling Flexibility in Process-Aware Information Systems*, pp. 297–320, Springer Berlin Heidelberg, 2012.

[12] L. González, R. Ruggia, J. Abin, G. Llambías, R. Sosa, B. Rienzi, D. Bello, and F. Álvarez, "A service-oriented integration platform to support a joined-up e-government approach: The uruguayan experience," in *Advancing Democracy, Government and Governance*, vol. 7452 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2012.

[13] L. González, "A comprehensive and policy-based approach to compliance management within inter-organizational service integration platforms," 2019. PhD Thesis.

[14] OASIS, "eXtensible Access Control Markup Language (XACML) version 3.0," 2013.

[15] A. Margheri, M. Masi, R. Pugliese, and F. Tiezzi, "A rigorous framework for specification, analysis and enforcement of access control policies," *IEEE Transactions on Software Engineering*, pp. 1–1, 2017.

[16] J.-R. Abrial, M. Butler, S. Hallerstede, T. S. Hoang, F. Mehta, and L. Voisin, "Rodin: an open toolset for modelling and reasoning in event-b," *International Journal on Software Tools for Technology Transfer*, vol. 12, pp. 447–466, 4 2010.

[17] A. Romanovsky and M. Thomas, eds., *Industrial Deployment of System Engineering Methods*. Springer Berlin Heidelberg, 2013.

[18] J.-R. Abrial, "On b and event-b: Principles, success and challenges," in *Lecture Notes in Computer Science*, pp. 31–35, Springer International Publishing, 2018.

[19] I. Ait-Sadoune and Y. Ait-Ameur, "Animating event b models by formal data models," in *Communications in Computer and Information Science*, pp. 37–55, Springer Berlin Heidelberg, 2008.

[20] A. Birukou, V. D'Andrea, F. Leymann, J. Serafinski, P. Silveira, S. Strauch, and M. Tluczek, "An integrated solution for runtime compliance governance in SOA," in *Service-Oriented Computing*, pp. 122–136, Springer Berlin Heidelberg, 2010.

[21] D. Knuplesch, M. Reichert, R. Pryss, W. Fdhila, and S. Rinderle-Ma, "Ensuring compliance of collaborative and distributed workflows," in *Proceedings of the 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, ICST, 2013.

[22] D. Knuplesch, M. Reichert, and A. Kumar, "Visually monitoring multiple perspectives of business process compliance," in *Lecture Notes in Computer Science*, pp. 263–279, Springer International Publishing, 2015.

[23] A. Baouab, O. Perrin, and C. Godart, "An event-driven approach for runtime verification of inter-organizational choreographies," in *2011 IEEE International Conference on Services Computing*, IEEE, 7 2011.

[24] M. E. Kharbili, Q. Ma, P. Kelsen, and E. Pulvermueller, "CoReL: Policy-based and model-driven regulatory compliance management," in *2011 IEEE 15th International Enterprise Distributed Object Computing Conference*, IEEE, 8 2011.

[25] G. Gheorghe, S. Neuhaus, and B. Crispo, "xESB: An enterprise service bus for access and usage control policy enforcement," in *IFIP Advances in Information and Communication Technology*, pp. 63–78, Springer Berlin Heidelberg, 2010.

[26] G. Gheorghe, P. Mori, B. Crispo, and F. Martinelli, "Enforcing UCON policies on the enterprise service bus," in *On the Move to Meaningful Internet Systems, OTM 2010*, pp. 876–893, Springer Berlin Heidelberg, 2010.

[27] D. Knuplesch and M. Reichert, "A visual language for modeling multiple perspectives of business process compliance rules," *Software & Systems Modeling*, vol. 16, pp. 715–736, 4 2016.