Towards Visually Monitoring Multiple Perspectives of Business Process Compliance*

David Knuplesch[1], Manfred Reichert[1], and Akhil Kumar[2]

[1] Institute of Databases and Information Systems, Ulm University, Germany
{david.knuplesch,manfred.reichert}@uni-ulm.de
[2] Smeal College of Business, Pennsylvania State University, PA, USA
akhilkumar@psu.edu

Abstract. A challenge for enterprises is to ensure conformance of their business processes with imposed compliance rules. Usually, the latter may constrain multiple perspectives of a business process, including control flow, data, time, resources, and interactions with business partners. Like in process modeling, visual languages for specifying compliance rules have been proposed. However, business process compliance cannot be completely decided at design time, but needs to be monitored during run time as well. This paper introduces an approach for visually monitoring business process compliance. In particular, this approach covers all relevant process perspectives. Furthermore, compliance violations cannot only be detected, but also be visually highlighted emphasizing their causes. Finally, the approach assists users in ensuring compliant continuations of a running business process.

Keywords: business process compliance, compliance monitoring

1 Introduction

Correctness issues of business process models have been intensively studied for more than a decade. While early work focused on syntactical correctness and soundness (e.g., absence of deadlocks and lifelocks), recent approaches have focused on how to ensure the compliance of business processes with semantic constraints. Usually, respective *compliance rules* stem from domain-specific requirements, like, for example, corporate standards or legal regulations [1], and need to be ensured in all phases of the process life cycle [2].

In this context, approaches addressing the compliance of running business process instances are covered by the notion of *compliance monitoring* [3–5]. In general, events of running process instances need to be considered to detect and report run-time violations of compliance rules (cf. Fig. 1). Thereby, *reactive* and *proactive* monitoring need to be distinguished. Regarding the former, compliance violations are reported once they have occurred. In turn, proactive monitoring

* This work was done within the research project C^{3}Pro funded by the German Research Foundation (DFG) under project number RE 1402/2-1.

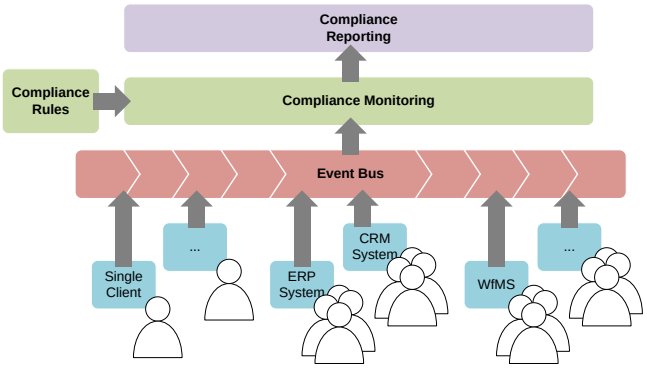

Fig. 1: Compliance Monitoring [7, 9]

aims to prevent compliance rule violations; e.g., by suggesting appropriate tasks that still need to be executed to meet a compliance rule. While early approaches for monitoring compliance focused on the control flow perspective, more and more, additional process perspectives have been considered as well (e.g. [6]). In particular, the data, resource and time perspectives as well as interactions with business partners have been addressed. Other advanced work has dealt with the traceability of compliance violations [7, 8]. However, existing approaches do not provide a satisfactory solution that combines an expressive compliance rule language with full traceability [9].

As example consider the event log from Fig. 2 that refers to an order-to-

#	date	time	type	id	details
⋮					
37	1/7/2013	15:27	receive	124	Request
38	1/7/2013	15:27	write	124	customer = Mr.Smith
39	1/7/2013	15:27	write	124	amount = 15.000€
39	1/7/2013	15:27	end	124	Request
⋮					
55	1/7/2013	18:03	receive	592	Request
56	1/7/2013	18:03	write	592	customer =Mrs.John
57	1/7/2013	18:03	write	592	amount = 27.000€
58	1/7/2013	18:03	end	592	Request
⋮					
77	2/7/2013	15:43	start	234	SolvencyCheck (Mrs. Brown)
78	2/7/2013	15:43	read	234	customer = Mr.Smith
79	2/7/2013	15:54	write	234	rating= high
80	2/7/2013	15:55	end	234	SolvencyCheck
⋮					
91	2/7/2013	18:13	start	453	Approval (Mr. Muller)
92	2/7/2013	18:14	read	453	customer = Mr.Smith
93	2/7/2013	18:14	read	453	rating = high
94	2/7/2013	18:17	write	453	result= granted
95	2/7/2013	18:18	end	453	Approval
96	2/7/2013	18:19	start	642	Approval (Mrs. Brown)
97	2/7/2013	18:20	read	642	customer = Mrs.John
98	2/7/2013	18:23	write	642	result = granted
99	2/7/2013	18:23	end	642	Approval
⋮					

Compliance rules
c_1 When a request item with an amount greater than 10,000 is received from an agent, the request must not be approved unless the solvency of the respective customer was checked. The latter task must be started at max three days after the receipt. Further, task approval and task solvency check must be performed by different staff members.
c_2 After approval of a request item, the agent must be informed about the result within one days.
c_3 After starting the production related to a particular order the latter may only be changed by the head of production.

Fig. 2: Event log of order-to-delivery processes and compliance rules

delivery process. Compliance rule c_1, which is shown on the right, is satisfied in one case, but violated in another. In particular, the depicted log refers to two different request items related to customers *Mr. Smith* and *Mrs. John*. These items, in turn, trigger two different instances of compliance rule c_1. In both cases, the amount is greater than 10,000 € and hence a solvency check is required. However, the latter was only performed for the request item of Mr. Smith, but not for the one of *Mrs. John* (i.e., c_1 is violated in the latter case). Besides the violation of c_1, compliance rule c_2 is violated twice as well. While the violated instance of c_1 can never be successfully completed, the violations of c_2 still can be healed by informing the *agent*. The rule examples further indicate that solely monitoring control flow dependencies between tasks is not sufficient to ensure compliance at run time. In addition, constraints in respect to the data, time and resource perspectives of a business process must be monitored as well as the interactions this process has with partner processes [10, 11, 9]. For example, the data perspective of compliance rule c_1 is addressed by activity *request item* and its data *amount*. Receiving the *request item*, in turn, represents an interaction with a business partner. Furthermore, the phrase *by different staff members* deals with the resource perspective, whereas the condition *at maximum three days* refers to the time perspective. To meet practical demands, compliance monitoring must not abstract from these process perspectives.

This paper sketches an approach for visually monitoring multiple perspectives of business process compliance. For this purpose, we annotate the visual *extended Compliance Rule Graph (eCRG)* language [11, 12] with text, markings and symbols to highlight the current state of a compliance rule. The annotations not only indicate compliance violations, but may also be utilized for recommending the next process steps required to restore compliance. Furthermore, they allow us to clearly distinguish between fulfilled and violated instances of an eCRG. Note that the eCRG language adequately supports the time, resource and data perspectives as well as interactions with business partners.

The remainder of this paper is structured as follows: The approach for visually monitoring multiple perspectives of business process compliance is outlined in Section 2. Section 3 concludes the paper and provides an outlook on future research.

2 eCRG Compliance Monitoring

This paper utilizes the extended Compliance Rule Graph (eCRG) language for compliance monitoring [11, 12]. The eCRG language is a visual language for modeling compliance rules. It is based on the Compliance Rule Graph (CRG) language [7]. As opposed to the latter, the eCRG language not only focuses on the control flow perspective, but additionally provides integrated support for the resource, data and time perspectives as well as for the interactions with business partners. Fig. 3 provides an overview of eCRG elements, which are applied in Fig. 4 in order to model the compliance rules from Fig. 2.

In the following, we sketch the approach towards visually monitoring multiple perspectives of business process compliance at runtime. As discussed in Sect. 1,

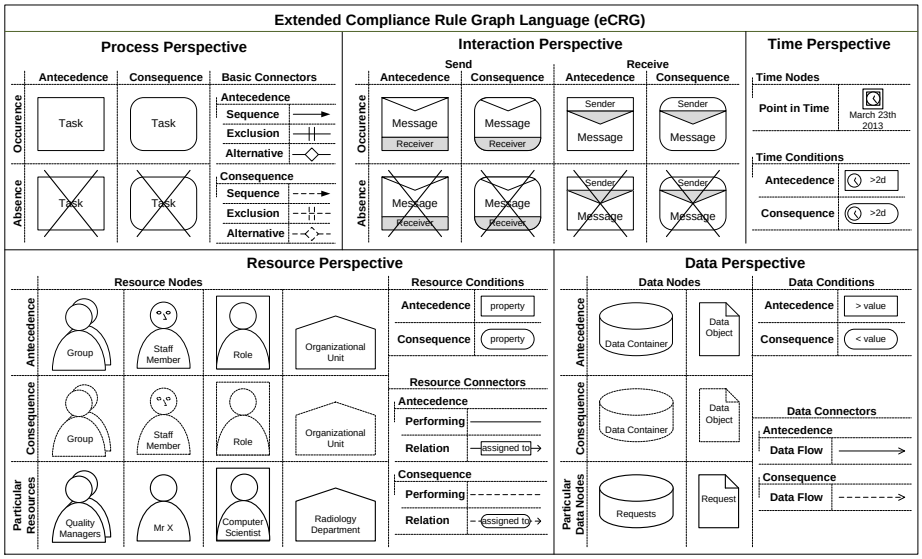

Fig. 3: Elements of the eCRG language

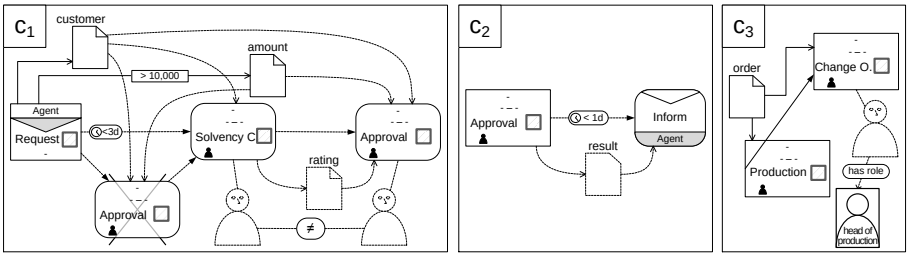

Fig. 4: Modeling compliance rules $c_1 - c_3$ with the eCRG language

compliance monitoring is based on streams of events, which occur during the execution of business processes, and aims to determine or prevent compliance violations. For this purpose, we extend the approach presented in [7] and annotate the elements of an eCRG when processing events.

Events The processing of event logs requires a well-defined set of events. As the approach enables compliance monitoring for multiple process perspectives, we not consider only events referring to the start and end of tasks, but additionally monitor data flow events as well as events that correspond to the sending and receipt of messages. Furthermore, events may include temporal information as well as information about involved resources. Table 1 summarizes the supported event types. Each event refers to the occurrence time as well as a unique id. The latter enables us to identify correlations between the start, end and data flow events of the same task or message.

Table 1: Supported Events

Task events	Message events	Data flow events
`start`(time, id, tasktype, performer)	`send`(time, id, message)	`write`(time, id, value $\xrightarrow{\text{param}}$ source)
`end`(time, id, tasktype, performer)	`receive`(time, id, message)	`read`(time, id, value $\xleftarrow{\text{param}}$ source)
	`end`(time, id, message)	

eCRG Markings To monitor the state of a compliance rule, we annotate and mark eCRG elements with symbols, colors and text (cf. Figs. 5). Such a *marking of an eCRG* results in an annotated eCRG highlighting whether or not the events corresponding to a particular node have occurred so far. Furthermore, it describes whether the conditions of edges and attachments are satisfied, are violated or have not been evaluated yet.

Event Processing We exemplarily describe how events are processed for an eCRG (cf. Fig. 6) and refer to [13] for a formal specification of the operational semantics of the eCRG langauge. First, all markings are updated to the point in time of the specific event. Second, the effects of the update (i.e., adapted annotations) are propagated to succeeding as well as skipped elements. Third, the actual *event handling* takes place depending on the type of the current event. Finally, the effects of the latter step are propagated as well.

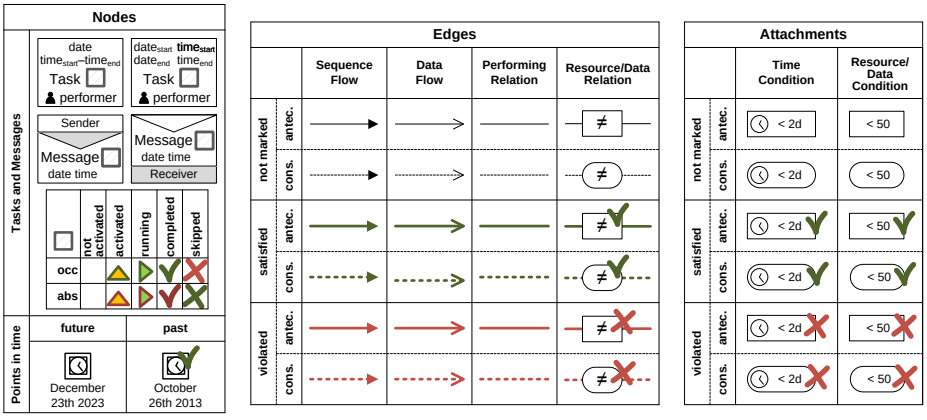

Fig. 5: Annotations of eCRG elements

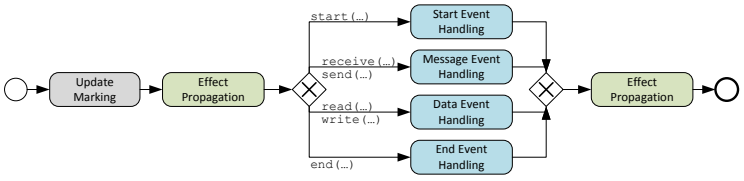

Fig. 6: Processing of start, message, data and end events

Fig. 7b illustrates the handling of a message event. In particular, the marking of the activated message node *request*, which matches the event, changes to ▶. Accordingly, the starting time is set. Start events are processed like message events, but additionally store information about the responsible actor. Start and message events are handled non-deterministically; i.e., the changes are applied to a copy of the original marking. Fig. 7c shows the handling of data events. In particular, the corresponding data flow edge is annotated with the data value passed. Fig. 7e illustrates the handling of an end event. In particular, the annotations of *request* is changed to ✓ and its ending time is set accordingly.

Fig. 7g illustrates how a marking is updated to the current point in time. In particular, the annotations of point in time nodes, which now refer to the past, change to ✓. Finally, time conditions on running task nodes or sequence flow edges are skipped (×) if they are no longer satisfiable (cf. Fig. 7g).

According to Fig. 6, effects of events and updates must be propagated to indirectly affected eCRG elements in order to ensure correct annotations (e.g., activation of subsequent task nodes) as well as to detect contradictory annotations related to the data and resource perspectives. In particular, data values are propagated from writing and reading data flow edges to dependent data object and data container nodes (cf. Fig. 7d). In turn, resources are propagated from task nodes to dependent resource nodes via the connecting resource edges. The propagation fails, if a resource, data object or container node was set to a different value before. In this case, the respective edge is skipped (×). Furthermore, conditions and relations are evaluated as soon as possible (cf. Fig. 7d). If any element of the eCRG corresponding to a task or message node is skipped (e.g., due to a failed data/resource propagation or a violated condition), the corresponding task or message node will be skipped as well. Then, outgoing sequence flows of completed nodes are marked as satisfied, whereas non-marked incoming edges of already started nodes are skipped. Sequence flow edges from and to skipped nodes are skipped as well. Task and message nodes, in turn, become activated when all incoming sequence flows they are depending on are satisfied. Additionally, task or message nodes will be skipped if they depend on sequence flows that were skipped as well. Note that the latter might require skipping further sequence flow edges, and so forth (cf. Fig. 7h). Finally, Fig. 7i provides a marking that fulfills c_1 for the request of Mr. Smith. In turn, Figs. 7h+7k highlight conflicts regarding time and resource perspectives.

The non-deterministic processing of start and message events may result in sets of markings. Therefore, single fulfilling or conflicting markings do not imply (non-)compliance with the related rule. For this purpose, [13] provides mechanisms to evaluate such sets and to select the most meaningful markings.

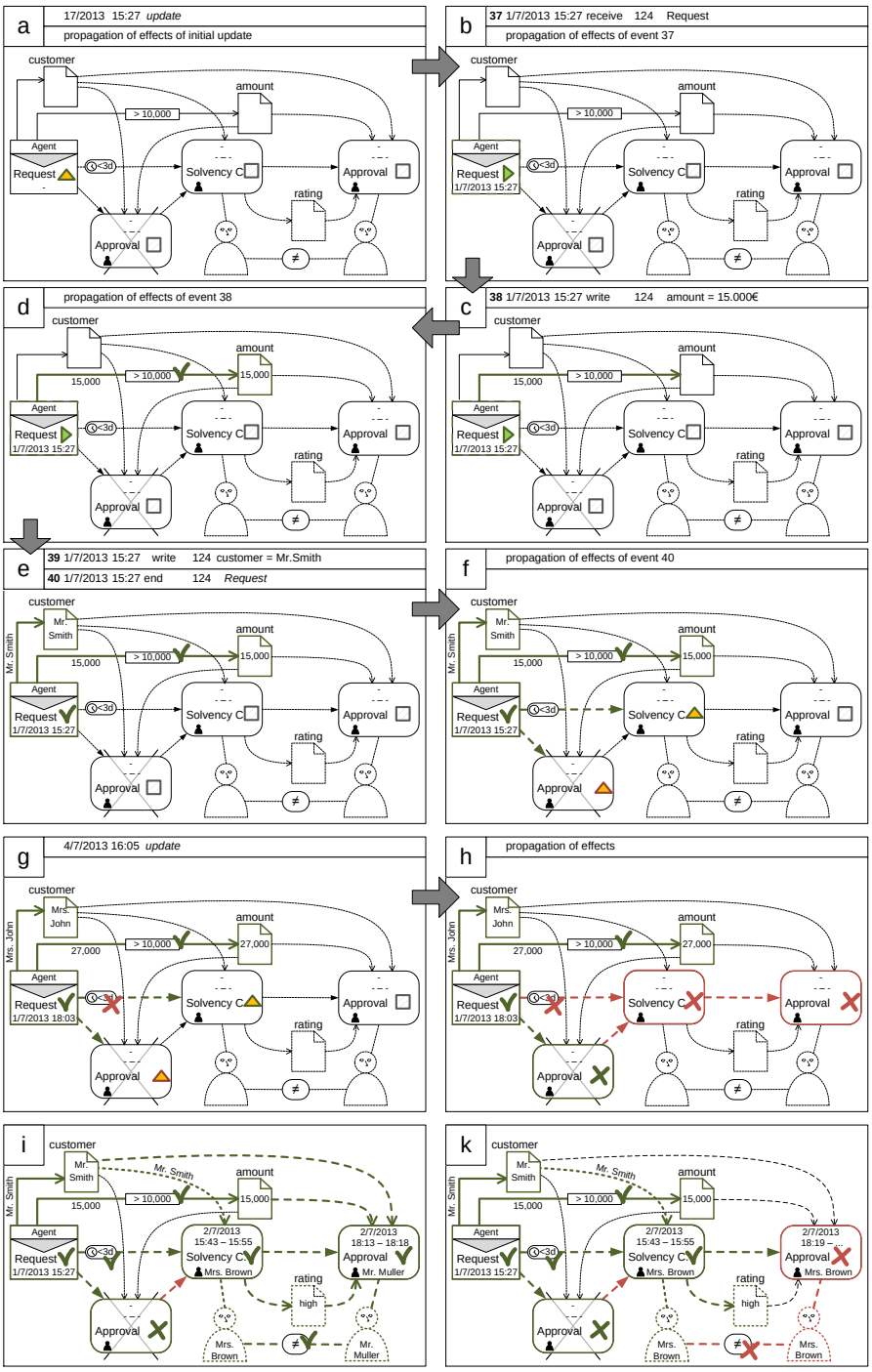

Fig. 7: eCRG markings and handling of events

3 Summary and Outlook

Business process compliance has gained increasing interest during the last years. Several approaches focus on compliance monitoring at run time [3–5]. However, existing approaches do not provide a satisfactorily solution combining an expressive language with full traceability [9]. This paper, therefore, has sketched an approach for visually monitoring multiple perspectives of business process compliance. For this purpose, we utilize the extended compliance rule graph (eCRG) language [11] that enables the visual modeling of compliance rules with support of the control flow, data, time, and resource perspectives as well as the interactions with partners. We annotate eCRGs with text, colors and symbols to visually highlight the current compliance state as well as to indicate its evolution during process execution. Note that we formally specified this operational semantics of the eCRG language in a technical report [13]. As opposed to existing approaches, we aim to combine full traceability with an expressive visual notation.

As next step, we will investigate algorithms for eCRG-based compliance monitoring utilizing the eCRG operational semantics we presented in [13]. Finally, we will provide a proof-of-concept implementation to evaluate the approach.

References

1. Sadiq, S., Governatori, G., Naimiri, K.: Modeling control objectives for business process compliance. In: BPM'07, Springer (2007) 149–164
2. Knuplesch, D., Reichert, M.: Ensuring business process compliance along the process life cycle. Technical Report 2011-06, Ulm University (2011)
3. Namiri, K., Stojanovic, N.: Pattern-Based design and validation of business process compliance. In: CAiSE'07, Springer (2007) 59–76
4. Alles, M., Kogan, A., Vasarhelyi, M.: Putting continuous auditing theory into practice: Lessons from two pilot implementations. Inf Sys **22**(2) (2008) 195–214
5. van der Aalst, W.M.P., et al: Conceptual model for online auditing. Decision Support Systems **50**(3) (2011) 636–647
6. Montali, M., et al.: Monitoring business constraints with the event calculus. ACM Trans. Intell. Syst. Technol. **5**(1) (2014) 17:1–17:30
7. Ly, L.T., Rinderle-Ma, S., Knuplesch, D., Dadam, P.: Monitoring business process compliance using compliance rule graphs. In: CoopIS'11. (2011) 82–99
8. Maggi, F., Montali, M., Westergaard, M., van der Aalst, W.M.P.: Monitoring business constraints with linear temporal logic: an approach based on colored automata. In: BPM'11. (2011) 132–147
9. Ly, L.T., et al.: A framework for the systematic comparison and evaluation of compliance monitoring approaches. In: EDOC'13, IEEE (2013) 7–16
10. Knuplesch, D., et al.: Towards compliance of cross-organizational processes and their changes. In: BPM'12 Workshops, Springer (2013) 649–661
11. Knuplesch, D., et al.: Visual modeling of business process compliance rules with the support of multiple perspectives. In: ER'2013, Springer (2013) 106–120
12. Semmelrodt, F., Knuplesch, D., Reichert, M.: Modeling the resource perspective of business process compliance rules with the extended compliance rule graph. In: BPMDS'14, Springer (2014) 48–63
13. Knuplesch, D., et al.: An operational semantics for the extended compliance rule graph language. Technical Report 2014-6, Ulm University (2014)