SeaFlows Toolset – Compliance Verification Made Easy*

Linh Thao Ly[1], David Knuplesch[1], Stefanie Rinderle-Ma[2], Kevin Göser[3],
Manfred Reichert[1], and Peter Dadam[1]

[1] Institute of Databases and Information Systems
Ulm University, Germany
{thao.ly,david.knuplesch,manfred.reichert,peter.dadam}@uni-ulm.de

[2] Faculty of Computer Science
University of Vienna, Austria
stefanie.rinderle-ma@univie.ac.at

[3] AristaFlow GmbH, Germany
kevin.goeser@aristaflow.com

Abstract. In the light of an increasing demand on business process compliance, the verification of process models against compliance rules has become essential in enterprise computing. The SeaFlows Toolset featured in this tool demonstration extends process-aware information system by compliance checking functionality. It provides a user-friendly environment for modeling compliance rules using a graph-based formalism. Modeled compliance rules can be used to enrich process models. To address a multitude of verification settings, SeaFlows Toolset provides two compliance checking components: The *structural compliance checker* derives structural criteria from compliance rules and applies them to detect incompliance. The *data-aware compliance checker* addresses the state explosion problem that can occur when the data dimension is explored during compliance checking. It performs context-sensitive automatic abstraction to derive an abstract process model which is more compact with regard to the data dimension enabling more efficient compliance checking. Altogether, SeaFlows Toolset constitutes a comprehensive and extensible framework for compliance checking of process models.

Key words: Compliance rules, Process verification, Tool support, Data-awareness

1 Introduction

In the light of an increasing demand on business process compliance [1], the verification of process models within process-aware information systems against

* This work was done in the research project SeaFlows which is partially funded by the German Research Foundation (DFG).

compliance rules has become essential in enterprise computing. To ensure compliance with imposed rules and policies, compliance audits for process models are necessary. Due to increasing complexity of process models [2] manual compliance verification is hardly feasible. Tool support is particularly needed in order to deal with changes at different levels. On the one hand, changes and evolution of regulatories and policies may occur, leading to changes in implemented compliance rules. On the other hand, changes to business processes may take place, resulting in changes of implemented process models. This further necessitates tool support for (semi-)automatic compliance verification.

The toolset featured in this tool demonstration resulted from our research in the SeaFlows project. In this project, we aim at providing techniques to enable compliance with imposed regulatories throughout the process lifecycle. This inludes compliance cheking of business process models at buildtime but also compliance monitoring of process instances at runtime [3]. With the implementation of SeaFlows Toolset, so far, we have realized concepts addressing compliance checking of process models at buildtime. The particular components shown in this tool demonstration enable modeling compliance rules as visual compliance rule graphs as well as verifying process models against imposed compliance rules [4]. To support a variety of verification scenarios and to exploit their specific properties, SeaFlows Toolset comprises several verification components: a *structural compliance checker*, enabling efficient compliance verification for block-structured process models and a *data-aware compliance checker*, enabling data-aware compliance checking using model checking techniques.

In the following, the particular components of SeaFlows Toolset are introduced. Related work is discussed in Sect. 3 before we close the paper with an outlook on future developments in Sect. 4

2 SeaFlows Toolset

SeaFlows Toolset extends process-aware information system (PAIS) by compliance checking functionality. Fig. 1 depicts the interplay between existing infrastructure stemming from PAIS (e.g., activity repository, process modeling tool, and process model repository) and components introduced by SeaFlows Toolset[2].

The SeaFlows Graphical Compliance Rule Editor (cf. Fig. 1) allows to model compliance rules over process artifacts as *compliance rule graphs* [4] (cf. Sect. 2.1). By interacting with the activity repository responsible for organizing and managing process artifacts relevant within a business domain, the Graphical Compliance Rule Editor enables compliance rule modeling over exactly the process artifacts available in the domain. Thus, we can enrich process models by compliance rules that are imposed on the corresponding business process. This can be done at an early stage, when the process is modeled to enable *compliance by design.* Compliance rules may be also assigned to a completed or released process model to perform compliance audits.

[2] The Rule Graph Execution Engine for executing compliance rule graphs is currently under implementation

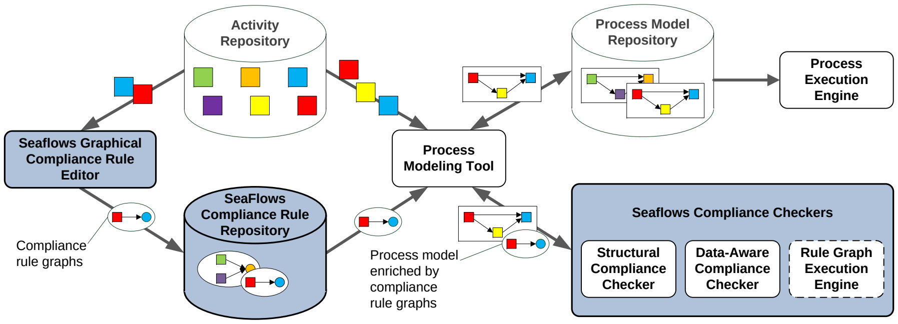

Activity Repository
Process Model Repository
Process Execution Engine
Seaflows Graphical Compliance Rule Editor
Process Modeling Tool
SeaFlows Compliance Rule Repository
Compliance rule graphs
Process model enriched by compliance rule graphs
Seaflows Compliance Checkers
Structural Compliance Checker
Data-Aware Compliance Checker
Rule Graph Execution Engine

Fig. 1. Overall infrastructure around the SeaFlows Toolset

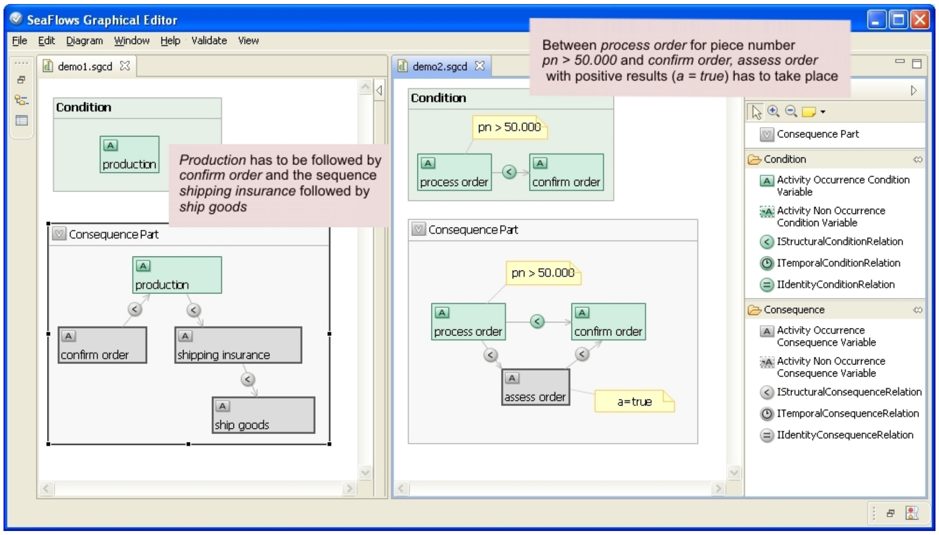
SeaFlows Toolset currently comprises two compliance checking components to verify process models (cf. Fig. 1), namely the Structural Compliance Checker and the Data-aware Compliance Checker. By interacting with the process modeling tool of PAIS, the SeaFlows compliance checkers enable the process designer to verify process models already during process design. Meaningful compliance reports help the process designer to identify incompliant process behaviour. Based on them, the process designer may further modify the process model until incompliance is resolved.

To transfer our concepts into a comprehensive prototype, we opted to base our implementation on the commercial process management system AristaFlow BPM Suite orginated from research activities in the ADEPT project [5]. AristaFlow BPM Suite provides a powerful API which enables us to extend existing PAIS functionality by compliance checking mechanisms in an elegant manner. Thus, SeaFlows compliance checking components are smoothly integrated into the process modeling environment of AristaFlow BPM Suite. In the following, the components of SeaFlows Toolset (cf. Fig. 1) and underlying concepts are discussed in more detail.

2.1 Graphical Compliance Rule Editor and Compliance Rule Repository

We developed a graph-based compliance rule specification language that enables modeling compliance rules in a manner similar to process modeling. Designed to support intuitive compliance rules modeling, *compliance rule graphs* are modeled by linking nodes representing absence and occurrence of activity executions of certain types [4]. In particular, (sub-)graphs are used to respresent an antecedent pattern that activates the compliance rule and corresponding required consequence patterns. This enables modeling frequent compliance rule patterns [6, 7] in a straightforward manner. Further, compliance rule graphs can be enriched with annotations of temporal constraints (e.g., minimal temporal distance) as well as data conditions.

The Graphical Compliance Rule Editor provides a user-friendly environment for modeling compliance rule graphs (cf. Fig. 2). Nodes of compliance rule graphs



Between *process order* for piece number *pn > 50.000* and *confirm order, assess order* with positive results (*a = true*) has to take place

Production has to be followed by *confirm order* and the sequence *shipping insurance* followed by *ship goods*

SeaFlows Graphical Editor
File Edit Diagram Window Help Validate View
demo1.sgcd
Condition
production
Consequence Part
production
confirm order
shipping insurance
ship goods
demo2.sgcd
Condition
pn > 50.000
process order
confirm order
Consequence Part
pn > 50.000
process order
confirm order
assess order
a=true
Consequence Part
Condition
Activity Occurrence Condition Variable
Activity Non Occurrence Condition Variable
IStructuralConditionRelation
ITemporalConditionRelation
IIdentityConditionRelation
Consequence
Activity Occurrence Consequence Variable
Activity Non Occurrence Consequence Variable
IStructuralConsequenceRelation
ITemporalConsequenceRelation
IIdentityConsequenceRelation

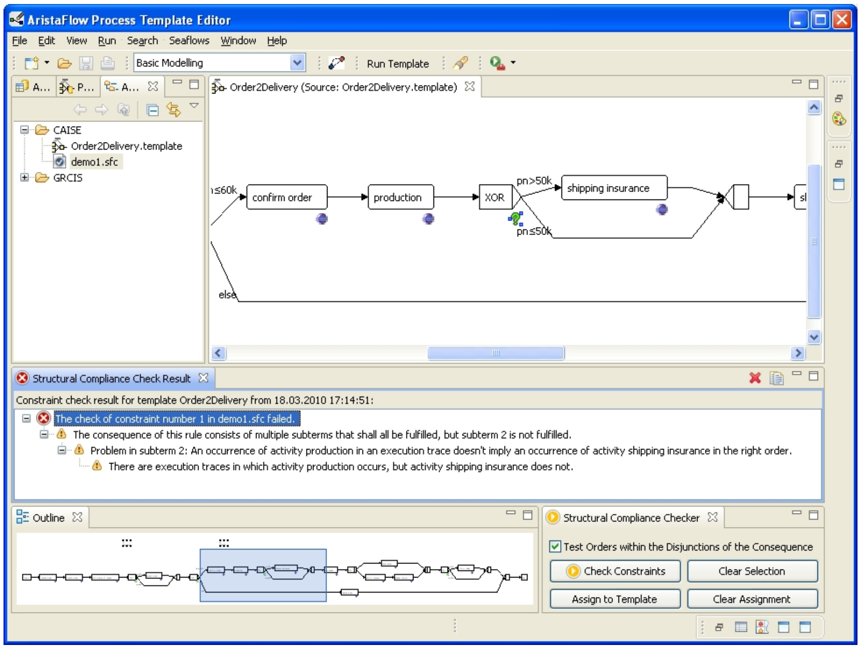
Fig. 2. The SeaFlows Graphical Compliance Rule Editor

are assigned to activity types available in the activity repository (cf. Fig. 1). Modeled compliance rule graphs are exported as separate XML-files which enables their organization within rule sets in the Compliance Rule Repository. In addition, versioning of compliance rules is also supported by the repository. Being implemented based on Eclipse Modeling Framework, modeled compliance rule graphs are based on a defined data object model that facilitates their import and processing in compliance checking tools.

2.2 Structural Compliance Checker

The basic idea underlying the Structural Compliance Checker is to efficiently verify process models by automatically deriving *criteria on the process structure* from compliance rules [8]. Following the dynamic programming paradigm, for each compliance rule a set of simple binary structural criteria (such as "A excludes B") whose satisfaction ensure compliance with the corresponding rule is derived. By checking the process model for compliance with these derived criteria, we can identify the criteria not fulfilled by the process model. This is useful information to generate intelligible textual feedback in case incompliance is detected. Based on the results of checking the structural criteria, the Structural Compliance Checker is able to provide detailed diagnosis that is helpful to locate incompliance (cf. Fig. 3). For example, the feedback Fig. 3 indicates that *shipping insurance* is optional to *production* in the process model. This detailed diagnosis can further be applied to resolve incompliance.

By making assumptions on the verification setting (e.g., unique label assumption) and exploiting the block-structure of process models the Structural Compliance Checker identifies incompliance in an efficient manner.



Constraint check result for template Order2Delivery from 18.03.2010 17:14:51:

- The check of constraint number 1 in demo1.sfc failed.
  - The consequence of this rule consists of multiple subterms that shall all be fulfilled, but subterm 2 is not fulfilled.
    - Problem in subterm 2: An occurrence of activity production in an execution trace doesn't imply an occurrence of activity shipping insurance in the right order.
      - There are execution traces in which activity production occurs, but activity shipping insurance does not.

☑ Test Orders within the Disjunctions of the Consequence

Fig. 3. The SeaFlows Structural Compliance Checker integrated into AristaFlow Process Template Editor

The Structural Compliance Checker is implemented as Eclipse-plug-in for AristaFlow Process Template Editor and thus, is smoothly integrated into the process modeling environment. Therefore, compliance checks "on the fly" during process modeling can be carried out to support compliance by design.

2.3 Data-aware Compliance Checker

The Data-aware Compliance Checker is able to deal with data-aware compliance rules and data conditions in process control flow. The challenge with data-aware compliance checking is that the exploration of the data dimension during compliance checking can lead to state explosion and thus, to intractable complexity. To tackle this, we developed a process-meta-model-independent approach for automatic context-sensitive (i.e., rule-specific) abstraction (cf. Fig. 4 B). By analyzing the data conditions contained in the compliance rule and in the process model, it reduces the state space of the data dimension to be explored during verification. The obtained *abstract process model* and *abstract compliance rule* are given as input to the actual compliance checking procedure (cf. Fig. 4 A). For compliance checking we used a model checker. In case of violation, the counterexample obtained from the model checker is conretized to yield not only the incompliant execution but also the its data conditions.

The Data-aware Compliance Checker first performs automatic abstraction, then transforms the abstract process model into a state space representation.

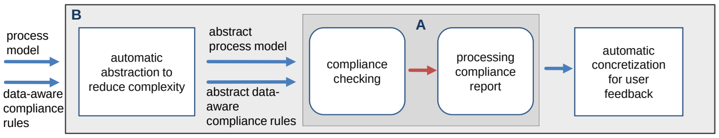
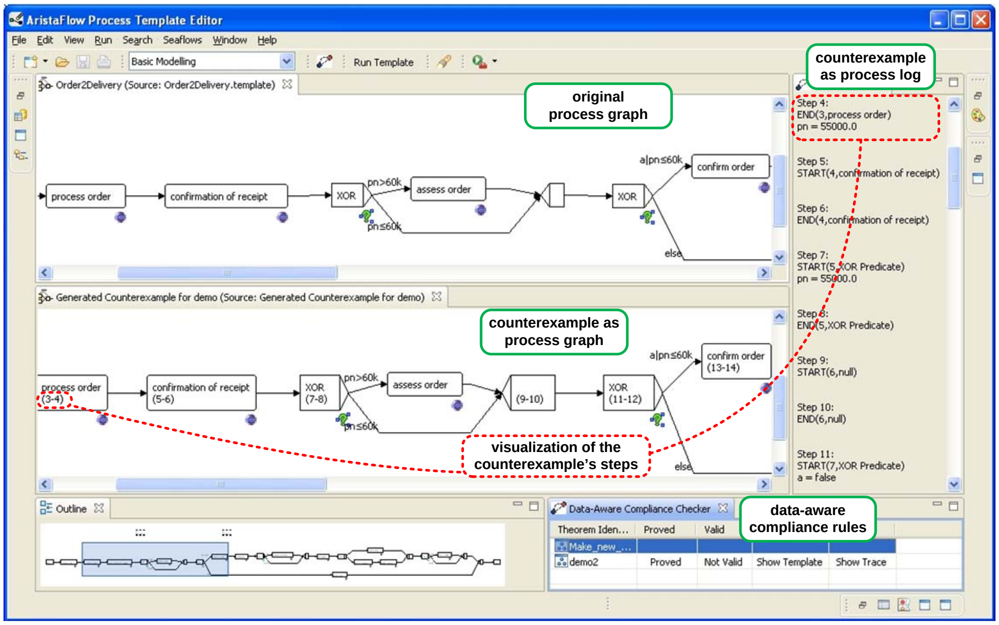

B
process model
data-aware compliance rules
automatic abstraction to reduce complexity
abstract process model
abstract data-aware compliance rules
A
compliance checking
processing compliance report
automatic concretization for user feedback

Fig. 4. Abstraction and concretization as pre- and postprocessing steps to the actual data-aware compliance checking

The latter is then passed to the model checker SAL [9], which carries out the exploration of the abstract process model. In case compliance violation is detected, the Data-aware Compliance Checker retransforms the counterexample output of the model checker and visualizes it as an execution trace and as process graph.

Similar to the Structural Compliance Checker, the Data-aware Compliance Checker is directly integrated into the process modeling environment. 17.000 lines of code and the class hierarchy comprising about 70 interfaces and 210 classes indicate its complexity. Automatic abstraction is supported for domains of numbers and for large domains of object references.


AristaFlow Process Template Editor
File Edit View Run Search Seaflows Window Help
Basic Modelling
Run Template
Order2Delivery (Source: Order2Delivery.template)
original process graph
process order
confirmation of receipt
XOR
pn>60k
pn≤60k
assess order
XOR
a|pn≤60k
confirm order
else
Generated Counterexample for demo (Source: Generated Counterexample for demo)
counterexample as process graph
process order (3-4)
confirmation of receipt (5-6)
XOR (7-8)
pn>60k
pn≤60k
assess order
(9-10)
XOR (11-12)
a|pn≤60k
confirm order (13-14)
else
visualization of the counterexample's steps
counterexample as process log
Outline
Data-Aware Compliance Checker
data-aware compliance rules

```
Step 4:
END(3,process order)
pn = 55000.0

Step 5:
START(4,confirmation of receipt)

Step 6:
END(4,confirmation of receipt)

Step 7:
START(5,XOR Predicate)
pn = 55000.0

Step 8:
END(5,XOR Predicate)

Step 9:
START(6,null)

Step 10:
END(6,null)

Step 11:
START(7,XOR Predicate)
a = false
```

Theorem Iden...	Proved	Valid		
Make_new_...				
demo2	Proved	Not Valid	Show Template	Show Trace

Fig. 5. The Data-aware Compliance Checker visualizes the counterexample as execution trace and process graph

3 Related Work

Three major challenges arise from compliance verification of process models: compliance rule modeling, verification techniques, and feedback generation. The concepts implemented in SeaFlows Toolset address all three issues. Existing approaches for modeling compliance rules range from rather informal annotations of process models with compliance rules, over formal languages [10], to visual patterns and languages [11, 12, 13]. With the compliance rule graphs, we opted for a compositional graph-based modeling formalism that supports the typical antecedent-consequence-structure of rules.

For compliance verification, model checking is often applied in literature [12, 13, 10]. As advantage we obtain an approach that is not specific to a particular process meta-model or process modeling notation. One challenge of model checking, however, is the generation of meaningful feedback from the report (e.g., counterexample) provided by the model checker. SeaFlows Toolset implements two compliance checking approaches, one based on model checking and another based on structural criteria, that complement each other.

Some approaches address the verification of data-aware compliance rules [11, 12]. However, the state explosion problem arising from exploration of the data dimension is not addressed by these approaches. In SeaFlows Toolset we implemented an abstraction approach that serves as preprocessing step to the actual data-aware compliance checking to limit state explosion.

[7] addresses visualization of incompliance by querying the process model for anti-patterns that are defined for each compliance rule pattern. In our approach, structural criteria are automatically derived from the compliance rule by the Structural Compliance Checker. Checking the structural criteria allows for identifying precisely the structural reason for incompliance.

Similar to DECLARE [14], the declarative process management system, SeaFlows enables to model graphical compliance rules. In DECLARE constraints are mapped onto formula in temporal logic and then to finite automata in order to execute constraint-based workflows. In contrast, SeaFlows compliance rule graphs are used to verify process models.

SeaFlows Toolset can be further complemented by other process analysis tools, such as the process mining framework ProM [15] to provide comprehensive support of compliance checking a priori as well as a posteriori.

4 Summary and Outlook

SeaFlows Toolset featured in this tool demonstration extends process-aware information system by compliance checking functionality. It enables modeling compliance rule as graphs independently from specific process models by making use of an activity repository. Process models can be enriched by compliance rules for documentation purposes and for compliance verification. Two compliance checkers, the Structural Compliance Checker and the Data-aware Compliance Checker, addressing specific compliance verification scenarios (e.g., data-awareness) complement each other and thus, ensure broad applicability.

In our future work, we will further extend SeaFlows Toolset to provide support for compliance checking during process execution (cf. the SeaFlows Rule Graph Execution Engine in Fig. 1). In addition, SeaFlows Toolset will be extended by a visualization and explanation component to provide advanced user feedback.

References

1. Sadiq, S., Governatori, G., Naimiri, K.: Modeling control objectives for business process compliance. In: Proc. BPM '07. (2007) 149–164
2. Bobrik, R., Reichert, M., Bauer, T.: View-based process visualization. In: Proc. of the 5th Int'l Conf. on Business Process Management (BPM'07). Volume 4714 of LNCS. (2007) 88–95
3. Ly, L.T., Rinderle-Ma, S., Dadam, P.: On enabling integrated process compliance with semantic constraints in process management systems. Information Systems Frontiers , accepted for publication.
4. Ly, L.T., Rinderle-Ma, S., Dadam, P.: Design and verification of instantiable compliance rule graphs in process-aware information systems. In: The 22nd Int'l Conf. on Advanced Information Systems Engineering (CAiSE'10). (2010) accepted for publication.
5. Dadam, P., Reichert, M.: The ADEPT project: a decade of research and development for robust and flexible process support. Computer Science-Research and Development **23**(2) (2009) 81–97
6. Dwyer, M.B., Avrunin, G.S., Corbett, J.C.: Patterns in property specifications for finite-state verification. In: Proc. ICSE'99. (1999) 411 – 420
7. Awad, A., Weske, M.: Visualization of compliance violation in business process models. In: Proc. BPM Workshops. (2009)
8. Ly, L.T., Rinderle-Ma, S., Dadam, P.: Integration and verification of semantic constraints in adaptive process management systems. Data & Knowledge Engineering **64** (2007) 3–23
9. Bensalem, S., et al.: An overview of SAL. In: Proc. of the Fifth NASA Langley Formal Methods Workshop, NASA Langley Research Center (2000) 187–196
10. Ghose, A., Koliadis, G.: Auditing business process compliance. In: ICSOC '07. Volume 4749 of LNCS., Springer (2007) 169–180
11. Awad, A., Weidlich, M., Weske, M.: Specification, verification and explanation of violation for data aware compliance rules. In: Proc. ICSOC/ServiceWave'09. (2009) 500–515
12. Liu, Y., Müller, S., Xu, K.: A static compliance-checking framework for business process models. IBM Systems Journal **46**(2) (2007) 335–261
13. Förster, A., et al.: Verification of business process quality constraints based on visual process patterns. In: Proc. 1st Joint IEEE/IFIP Symposium on Theoretical Aspects of Sofware Engineering. (2007)
14. Pesic, M., Schonenberg, M., Sidorova, N., van der Aalst, W.: Constraint-based workflow models: Change made easy. In: OTM 2007, Part I. Number 4803 in LNCS, Springer-Verlag (2007) 77–94
15. van der Aalst, W., et al.: ProM 4.0: Comprehensive support for real process analysis. In: Proc. ICATPN'07. Volume 4546 of LNCS., Springer (2007) 484–494