Automated Certification for Compliant Cloud-based Business Processes

Cloud Computing workflows need to adhere to a variety of rules and offer high flexibility. This is at odds with the compliance certification currently being carried out in a manual fashion. The paper presents Comcert, an approach for the automated analysis of workflows. If a workflow does not adhere to the given rules, re-usable rule patterns are used to pinpoint the workflow vulnerabilities. The results of this design time analysis can be used as certificate by Cloud providers to signal compliance. Auditors can check the rule adherence of workflows before workflow execution, and thanks to the rule patterns certification is open to scrutiny by customers. Companies which so far have refrained from Cloud Computing because of Compliance concerns can use the new analysis approach to check for rule adherence, and Cloud providers can demonstrate compliance through certificates.

DOI 10.1007/s12599-011-0155-7

The Authors

Dr. Rafael Accorsi (✉)
Dipl.-Inf. Lutz Lowis
IIG Telematics
University of Freiburg
Friedrichstr. 50
79098 Freiburg
Germany
accorsi@iig.uni-freiburg.de

Yoshinori Sato MSc
Yokohama Research Laboratory
Hitachi
292 Yoshida-cho, Totsuka-ku,
Yokohama
244-0817 Kanagawa
Japan
yoshinori.sato.uw@hitachi.com

Received: 2010-07-01
Accepted: 2011-02-09
Accepted after three revisions by Prof. Dr. Müller.

This article is also available in German in print and via http://www.wirtschaftsinformatik.de: Accorsi R, Lowis L, Sato Y (2011) Automatisierte Compliance-Zertifizierung Cloud-basierter Geschäftsprozesse. WIRTSCHAFTSINFORMATIK. doi: 10.1007/s11576-011-0269-z.

1 Automating the Certification of Business Processes

Certifying the adherence of business processes to compliance requirements is a key issue in the large-scale deployment of reliable Cloud Computing (Hayes 2009; ENISA 2009; CSA 2009, 2010). It must be checked that moving (parts of) business processes into the Cloud does not violate existing rules, and that the processes also adhere to new rules which the use of Cloud Computing may bring along. Currently, many companies refrain from Cloud Computing because of compliance considerations, most of all security and privacy related ones.

Sustainable Cloud Computing includes the capability of provably keeping control over processes' compliance. However, two issues prevail on the way to achieving certification. Firstly, tools for automating certification procedures are missing. In consequence, certification is a long-winded, error-prone procedure. This is at odds with the increased flexibility stemming from the realization of Clouds as a means for companies to timely adapt their business processes on demand. Secondly, the simultaneous consideration of a multitude of regulations and contractual rules increases the complexity of checking compliance. A uniform way of expressing the resulting requirements has not yet been established and research results are lacking (Breaux 2009 is a notable exception). Overall, this situation and the associated risks of noncompliance inhibit enterprises from outsourcing their tasks onto the Cloud (Chow et al. 2009) and, in general, prevent the full realization of the economical potential of Cloud Computing (Etro 2009).

Addressing these two issues, this paper presents a thorough classification of compliance rules. Drawn from major compliance regulations, the resultant requirements categorization comprises three rule classes for which formal patterns are provided. Building upon this classification, the paper introduces Comcert, an approach for the automated compliance certification of business processes. Comcert employs Petri nets (Murata 1989) as a notation-independent, uniform formalism to specify both business processes and compliance rules. Similar to security automata (Schneider 2000), Petri net patterns modeling the rules are analyzed in parallel to the process specification, flagging when the process structure indicates a rule violation. This gives Comcert a constructive character as it automatically identifies design vulnerabilities in process specifications.

Comcert thus contributes to automating the certification of processes in the Cloud and, in consequence, fostering its wider deployment. Allowing workflows to be checked for compliance rules, including security and privacy rules, Comcert helps to make Cloud Computing a real option for companies. Due to the use of a general Petri net formalism, the approach does not rely on a particular process notation, such as Business Process Modeling Notation (BPMN), Busi-

Published online: 20 April 2011

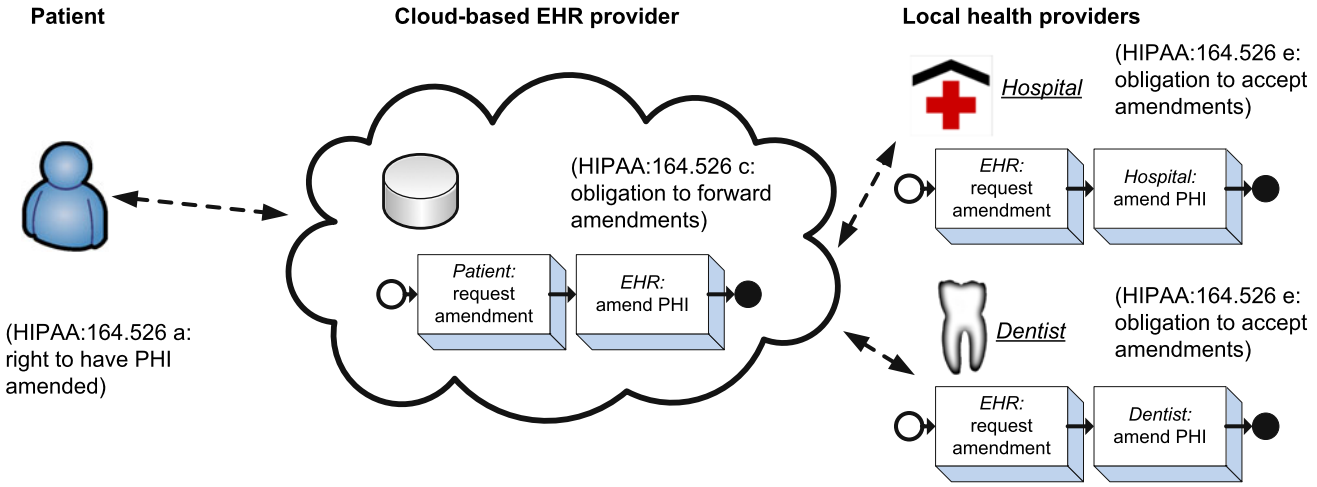

Fig. 1 Health care scenario with exemplary workflows

ness Process Execution Language (BPEL) or Event-driven Process Chain (EPC). Checking both the control flow and the data flow for rule adherence, Comcert provides well-founded, reliable evidence of (non)compliance.

Section 2 presents the motivation behind Comcert. Along with a Cloud-based E-Health scenario with HIPAA rules as running example, a description of Comcert's use cases is given. Section 3 summarizes a selection of compliance texts and extracts a classification of compliance rules. From the classification, rule patterns will be derived in Sect. 4, which also introduces the Comcert analysis algorithm. Section 5 concludes the paper.

2 Scenario Illustration

Of the many regulations that include usage control requirements, the U.S. Health Insurance Portability and Accountability Act (HIPAA 1996) is one of the most detailed ones. HIPAA regulates the use of protected health information (PHI) in business processes. **Figure 1** illustrates the E-Health scenario, using exemplary workflows and HIPAA rules. The scenario comprises a Cloud-based provider of electronic health records (EHR) and local health providers such as hospitals and dentists. Each party has its own workflows, and some of the workflows are connected to workflows of other parties. Note that **Fig. 1** is not an actual process model, but a schematic representation of the involved parties, their workflows, and according HIPAA rules.

A *workflow* is a discrete and case-based business process, i.e., it has a defined start and end point and handles a specific instance of a business process. The two main characteristics of a workflow are its *control* flow and its *data* flow. The control flow describes which *activities* happen in what *order*, and the data flow describes which *data* and *resources* are exchanged between these activities. For example, the Cloud-based EHR provider in **Fig. 1** runs a workflow that contains a control flow from the "request amendment" activity to the "amend PHI" activity. The data flow consists of the incoming request and the PHI amendment between the two activities.

In **Fig. 1**, a patient uses an online EHR, such as Google Health or Microsoft HealthVault offer. The EHR provider interacts with third parties such as hospitals and dentists. However, based on the patient's right to have her PHI amended (HIPAA:164.526a), the patient may request – in this case, she does so through the EHR provider – that third parties update their records accordingly. Third parties, in turn, are obliged to forward the performed amendments to all other relevant parties (HIPAA:164.526c). According to HIPAA:165.526e, these third parties must accept the amendment and perform the corresponding updates.

The general scenario is that of a company which wants to use Cloud services. Three questions result, one each regarding the company, the auditor(s) and the company's customers. First, the company would like to identify the parts of its business processes that can be outsourced without violating rules regarding, for example, the location of service execution. Second, auditors have to check the rule adherence of the (partially) outsourced business processes. Third, customers should have the ability to assess whether the "new" business processes treat personal data in a way that conforms to privacy policies.

Current workflow analysis techniques chiefly focus on control flow, i.e. on the detection of activity errors caused by flawed workflow design, e.g. van der Aalst (2003), Ehrig et al. (2007), Wong and Gibbons (2008). Ghose and Koliadis (2007) – and similarly Governatori et al. (2009) – present an approach for auditing workflows for compliance that focuses on activities and time limits. These approaches employ modal temporal logic for the analysis of workflows specified in BPMN. Focusing exclusively on data, Meda et al. (2010) introduce an approach for checking data flows, and Atluri et al. (2001) present an approach for reasoning about confidentiality properties in workflows. Monakova et al. (2009), Liu et al. (2007), and Trčka et al. (2009) present an approach for the automated verification of BPEL processes which combines both data flow and control flow perspectives. These approaches neglect usage control and, hence, cannot check for all of the compliance requirements (see Sect. 3). For this reason, Comcert includes usage control checks.

Given a workflow model and a set of rules, Comcert can answer all three questions. Technically, there really is only one question: "Does workflow W adhere to rule R?" From an organizational perspective, the three views company, auditor, and customer differ because of the information being available to each. The

company has all details at hand. The auditor sees what the company gives him – this can, but may not be every detail. The customer sees little to none of the business process' details. In the following, we focus on the technical rule adherence check, assuming that the relevant pieces of information are available. Third parties, checking company details on behalf of the customers and creating compliance certificates for public display, will not be further discussed in this article.

3 Compliance Regulation Survey and Classification

Given the strong demand for compliance, two questions emerge. First, which rules do workflows have to adhere to? Second, how can the rule adherence be checked? The remainder of this section presents a survey of compliance regulations to answer the first question. Section 4 will answer the second question by introducing the Comcert approach.

3.1 Compliance Regulation Survey

OECD Guidelines The "OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data" (1980) define eight principles: the collection limitation, data quality, purpose specification, use limitation, security safeguards, openness, individual participation and the accountability principle. Organizational processes are necessary to, e.g., achieve high data quality through record updates, or to follow the openness principle by informing customers about the uses of personal data. Regarding workflows, the principles mainly concern the data flow. Data collection should be limited ("collection limitation") to the amount required for a specific purpose ("data quality"), and data usage should not exceed that specific purpose ("use limitation").

European Community Directive 95/46/EC Within the European Community, the Directive 95/46/EC regulates "the protection of individuals with regard to the processing of personal data" and "the free movement of such data" (European Commission 1995). Workflows adhere to this directive if they use data only for a specific and specified purpose (95/46/EC:6.1.b) and do not use more data than necessary for that purpose (95/46/EC:6.1.c). Companies shall keep identifying personal data only as long as necessary for a certain purpose (95/46/EC:6.1.e), shall obtain the customer's consent to use personal data (95/46/EC:7.a), may not use certain kinds of personal data (95/46/EC:8.1), and have to inform customers about how the companies use personal data (95/46/EC:10,12). In general, the appropriate security mechanisms have to be used (95/46/EC:17.1), and personal data may only be transmitted to other countries if those countries provide an appropriate level of security and privacy for the data (95/46/EC:25).

German Tele Media Act (TMG) The TMG describes how and what personal data may be collected and used by online services (TMG 2009). Specifically, personal data ("personenbezogene Daten") may only be used in workflows if either the law requires their use or the customer has given her consent (TMG:12). Companies must announce their privacy policy (TMG:13). Service usage must be possible without being observed by others, and profiles may, if at all necessary, only be created using pseudonyms (TMG:13.4). Only the required personal data shall be collected (TMG:15.1), and billing shall not reveal usage details unless otherwise agreed upon by the customer.

German Federal Data Protection Act (BDSG) The BDSG (2009) refines the Directive 95/46/EC and allows the processing of personal data only if permitted by law or the explicit consent of a customer (BDSG:4). Transmission of personal data to third parties, possibly in other countries, may only happen if an appropriate level of security is given (BDSG:4.b). Customers have the right to have their records updated or deleted (BDSG:6). Technical and organizational security must be provided by the processing company, mainly access control and usage control (BDSG:9). Third parties providing additional processing must adhere to the same rules (BDSG:11). Also, processing must follow a certain purpose (BDSG:28), and companies must inform their customers about how personal data is being used (BDSG:34).

Gramm-Leach-Bliley Act (GLB) Focusing on the financial sector, the GLB mostly describes affiliations between banks, securities firms, and insurance companies (GLB 1999). GLB:501 *a* and *b* demand that so-called "nonpublic personal information" (NPPI) must be protected by establishing appropriate security standards. GLB:502 forbids disclosing NPPI to third parties, neither directly nor through affiliates. GLB:503 demands that financial institutions inform customers about the current privacy policy. The related, cross-sector regulation Sarbanes-Oxley-Act (SOX 2002) is even more abstract from a workflow point of view and does not contain any rules directly pertaining to workflows.

Health Insurance Portability and Accountability Act (HIPAA) The HIPAA Administrative Rule regulates the use of health data. Of the presented compliance sources, HIPAA is the most detailed one regarding workflow rules. For this reason, it serves as our running example and concludes this survey.

While the security part (subpart C of part 164) demands general security precautions, the privacy part (subpart E of part 164) describes how PHI may be used in workflows. Based upon a "default deny" approach (HIPAA:164.502.a), PHI may only be used or disclosed for treatment, payment, or health care operations purposes (HIPAA:164.502.a.1). A "minimum necessary" (HIPAA:164.502.b) requirement holds for both the sender and the receiver of PHI: If a contracted third party receives PHI which it does not require by the contract, the receiving side is obliged to inform the sending side (HIPAA:164.504.e.2.ii.C). As usual, the patient's consent must be obtained before using or disclosing any PHI (HIPAA:164.506.a.1) and if PHI is used or disclosed without consent, this must be documented and resolved later (HIPAA:164.506.a.3.C.ii). Certain PHI may be used without authorization, e.g., for directory purposes within a hospital (HIPAA:164.510.a.1), but other PHI require extra authorization to be used/disclosed, e.g., psychotherapy notes (HIPAA:164.508.a.2). Patients may request to restrict the use and disclosure of their PHI to certain elements (HIPAA:164.502.c), have a right to be informed about their PHI (HIPAA:164.524.a.1) within a certain amount of time (30 days, see HIPAA:164.524.b.2.i). Also, patients have the right to be provided with a list of their PHI's disclosures over the last six years (HIPAA:164.528), and may ask to have their PHI amended (HIPAA:164.526.a.1).

Table 1 Classification of Compliance Rules

		HIPAA	95/45/EC	OECD	BDSG	TMG	GLB
↔	Inform customers about policy/usage	√	√	√	√	√	√
↔	Obtain customer consent	√	√	√	√	√	
↔	Check third parties	√	√		√		√
↔	Update customer records	√		√	√		
↔	Delete after use	√	√				
↘	Treat special data separately	√	√				
↘	Use for specific purpose	√	√	√	√	√	
▯	Use pseudonyms or de-identify	√			√	√	
▯	Limit to minimum necessary	√	√	√			

Such amendments must then be forwarded by the amending institution (HIPAA:164.526.c), e.g., a hospital, to all involved parties which must accept that amendment (HIPAA:164.526.e) and update their records accordingly.

3.2 Compliance Rule Classification

The workflow related sections of the above compliance regulations can be organized in a few basic classes of rules, as shown in **Table 1**.

The rule categories in the second left column were obtained by sifting through the compliance sources in the top row and listing all rules that directly pertain to either the control flow (e.g., some activity has to happen before another) or the data flow (e.g., treatment of documents and their content) of workflows. After consolidating those rules that only used slightly different wording to describe the same requirement, the nine categories remained.

Given these nine categories, we transformed the corresponding rules into a Petri net representation as described in Sect. 4. Then, we analyzed the representations for similarities and differences. It showed that most of the rules (5 out of 9) referred to the order of activities, and two each referred to the workflow's branching conditions and the data being processed.

The resulting three rule classes are symbolized by the icons in the very left column. A double headed arrow for categories that require certain activities to (not) be performed before or after other activities. A single headed branching arrow for categories describing the flow of data. Finally, a rectangular label stands for categories directly relating to data elements.

Access control is important for Cloud-based workflows, but it is not sufficient for covering all the above rule classes. While access control covers the first release of data, it neither covers further releases nor additional uses. In the E-Health scenario, a health professional could obtain PHI with the stated purpose of treatment, and then use it for another purpose, e.g., advertising, in a following activity. In contrast to other classifications and taxonomies of compliance requirements, such as (Breaux and Antón 2008), (Wagner 2002) and (Sadiq et al. 2007), which address only a single legislation, the classification presented draws its categorization from different legislations, thereby being more suitable for Cloud providers, which have to simultaneously comply with multiple regulations and contractual terms (service-level agreements). In particular, Comcert checks usage control, without which many compliance requirements cannot be covered. This way, the new approach extends previous work in usage control (Park and Sandhu 2004 provide a model to capture usage control rules, Pretschner et al. 2006 present an enforcement mechanism) to the analysis at designtime.

4 The Comcert Approach

Petri nets are well-suited for reasoning about workflows (van der Aalst 1998), as they provide an adequate formal semantics for workflow specifications in BPMN, BPEL and EPC (Lohmann et al. 2009). Building upon their standard definition (Murata 1989), Comcert uses Petri nets to formalize both workflows and rules. Under the assumption that, with their graphical representation, Petri nets are more intuitive than the complex formulas of, e.g., linear temporal logic, this offers the advantage of making the compliance check comprehensible for a wider group of users while still delivering sound results.

The main idea behind Comcert's analysis is to transform a workflow into one Petri net and the rules into additional ones. Rule adherence is determined by traversing each path of the workflow and triggering the according transitions in the rule Petri net (RPN) for each workflow activity. Similar to security automata (Schneider 2000), each RPN contains special places which, after the workflow has been traversed completely, either indicate rule adherence or a violation.

Comcert is suitable for the automated detection of design vulnerabilities at workflow level. Regarding other levels, such as software and hardware, different approaches are required (Lowis and Accorsi 2010). Overall, certification and verification techniques are tailored for a particular workflow notation. Comcert abstracts away from the particular notation and employs a Petri net meta-model for both the workflow and the rules, resp. compliance requirements representation.

Details of Comcert's Petri net definitions will be given when introducing the analysis patterns. First, we discuss the transformation of workflows and rules into Petri nets.

4.1 Workflow Representation and Transformation

Workflow nets are employed as a target meta-model to formalize workflows (van der Aalst 1998). A workflow net is a special kind of Petri net with distinct source and sink places where all the nodes lie on some path between the source and the sink place. A token in the source place denotes a new execution (so-called *case*), whereas a token in the sink place denotes a complete execution (end of a case). The following illustrates the transformation of workflows into workflow nets.

Example 1 BPMN offers three main elements for the formalization of a workflow's control flow: activities, events, and gateways. **Figure 2a** illustrates some of these elements. The boxes stand for activities (or tasks), x-gateways for the exclusive choice (left-hand side) and simple merge (right-hand side). The circle in the left denotes the start event, whereas the bold circle in the right stands for the end event.

The corresponding workflow net derived from **Fig. 2a** is depicted in **Fig. 2b**.

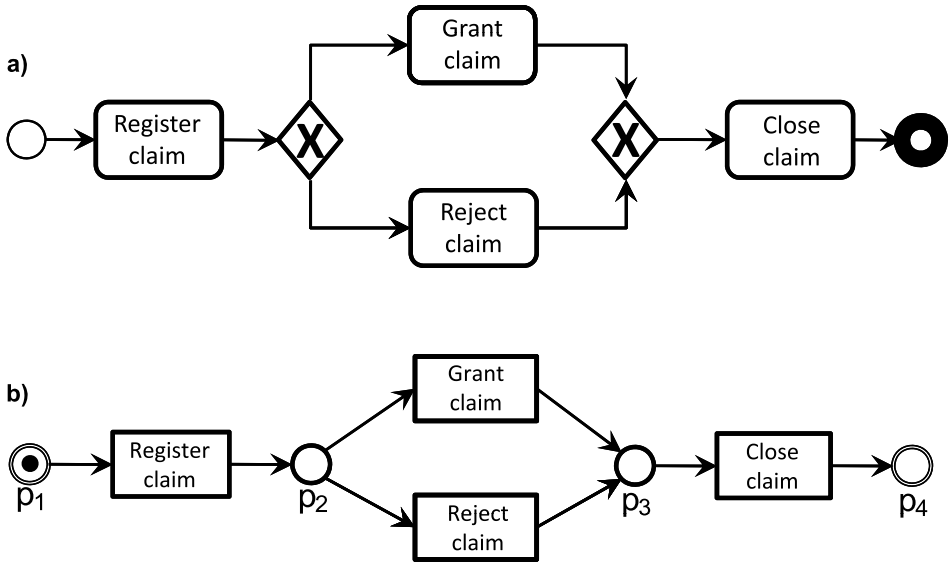

Fig. 2 Exemplary BPMN and workflow net representation

Place p_2 is called an "OR-split" and p_3 an "OR-join".

The transformation of workflow descriptions into workflow nets can be generally automated, e.g., with Oryx (2010) and SAP Galaxy (Saha 2008). The precision of the generated model varies according to the source workflow. For BPMN, for example, there are mappings which restrict the original model to having a single start and end event (Dijkman et al. 2008). Moreover, activities with multiple concurrent instances, as well as some kinds of gateways, such as OR-gateways, cannot generally be mapped into workflow nets. Similarly, only a particular class of EPC models can be mapped into workflow nets (van Dongen et al. 2007). In contrast to that, the "feature-complete" transformation of BPEL workflows is possible, e.g. with WofBPEL (Ouyang et al. 2005). Comcert uses workflow nets with extended annotations. It employs the tool *BW2PN* (IIG 2010) to transform a BPEL workflow with its WSDL specification into a Petri net stored as Petri Net Markup Language (PNML). *BW2PN* ignores certain BPEL features such as fault or exception handlers, so that compact, small Petri nets can be created. This is a trade-off between feature completeness and readability. If the scenario demands including exception handling, other tools can be used to create feature complete Petri nets.

4.2 Comcert Patterns for Compliance Rule Representation

Comcert assesses the compliance of a workflow by analyzing the five established elements required to check for rule adherence in workflows: *activities*, *data, location, resources*, and *time limits* (Curtis et al. 1992; Stohr and Zhao 2001; Sadiq et al. 2007; Svirskas et al. 2007; COMPAS 2008; Breaux 2009; Cabanillas et al. 2010).

A rule describes which activities may, must or must not be performed on what objects by which roles. In addition, a rule can further prescribe the order of activities, i.e. which activities have to happen before or after other activities.

The formalization of rules as Petri nets patterns has been proposed by Katt et al. (2009) and Huang and Kirchner (2009). In contrast to Katt et al., Huang and Kirchner cannot cope with the expression of usage control policies. Katt et al. employ Usage Control Colored Petri Nets (UCPN) for the formalization and enforcement of diverse types of obligations, i.e. actions to be performed before, during and after an activities. However, their approach assumes that the rules are integrated into the workflow, so that UCPN cannot be singled out for reuse for other workflows. Acting as a security automata (Schneider 2000), rules in Comcert are captured as Petri net patterns and are not integrated into the workflow. Together with the classification of compliance requirements, this makes it possible to organize compliance rules in categories according to their intent and semantics, thereby facilitating their formalization as re-usable Petri net patterns or templates in other modular policy languages for usage control.

Comcert captures rules as Petri net patterns. A rule Petri net (RPN for short) consists of the standard Petri net elements places P (depicted as circles), transitions T (rectangles), arcs A (lines) and tokens (dots). Arcs connect either P to T or T to P, and tokens are always contained in a place. Places immediately leading to a transition are this transition's *source places*, places immediately following a transition are this transition's *sink places*. A transition is *active* if its source places contain tokens; an active transition can *fire*. Upon firing, a transition *consumes* tokens from its source places and *produces* tokens in its sink places. Standard firing behavior is to consume one token of each source place and to produce one token in each sink place.

Within a Cloud-based workflow, multiple locations can be involved in one workflow, especially when services are outsourced to other countries. Each element of a net can carry predicates, which are used to describe further details, e.g., about the location of an activity or the emergency status of a patient. Such predicates are used to enable extended firing behavior.

Extended firing rules are introduced to make the firing behavior of some transitions deterministic (in **Fig. 3**, deterministic transitions resp. transitions with extended firing rules carry a dotted rectangle inside). These transitions fire in dependency of token and activity predicates, including tokens (not) being available in certain places. In **Fig. 3** for example, transition A in *Pattern a* fires in dependency of an activity's location, transition B in *Pattern e* fires in dependency of a token being present in the place *flag*. Using extended firing rules reduces the visual complexity of the RPNs because it requires (in some cases, much)

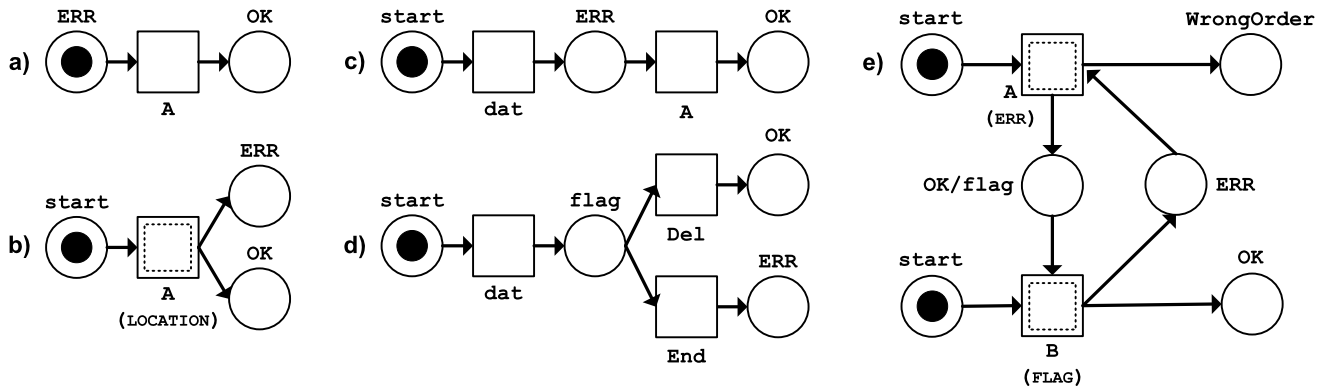

Fig. 3 Rule Petri net patterns (excerpt)

fewer places and transitions than a standard Petri net would require for the same tasks.

RPN places can have different semantics, the two most important ones are indicated by *OK* and *ERR* in **Fig. 3**. A token in *OK* signals rule adherence of the corresponding workflow path. Rule violations are identified through tokens in *ERR*. Section 4.3 will sketch how tokens and workflow paths are linked through the coloring resp. numbering of tokens.

Transforming and refining a natural language policy into a RPN is supported through RPN patterns. Rules for each of the five elements activities, data, locations, resources, and time limits can be transformed with these patterns. Because of space constraints, we only present a selection of patterns in the remainder of this section.

Pattern a in **Fig. 3** serves to detect the presence of required activities. As long as the activity *A* (transition *A* in the pattern) is not found in the workflow model, the RPN remains in its initial state, indicating an error through a token in the place *ERR*. If and when the activity *A* is encountered while scanning the workflow model, transition *A* in the RPN fires. This removes the token from *ERR* and creates a new one in *OK*. At the end of the workflow, the RPN then shows adherence to the rule that activity *A* must be present in the workflow: a token in *OK* signals adherence, a token in *ERR* means a violation. Inverting the semantics (*OK* becomes *ERR* and vice versa) allows checking for the absence of forbidden activities.

With the RPN *Pattern b*, it is straightforward to check for the location of an activity's execution. When the activity *A* is found in the workflow model, the transition *A* fires in dependency of the activity's location, which must be available through, e.g., an annotated predicate. If the location matches the required one, a token in *OK* is created, signaling rule adherence. A forbidden location causes the production of a token in *ERR*, indicating a rule violation. Because the workflow model may contain the activity *A* multiple times, the firing does not consume any tokens, which again is part of the extended firing behavior.

If data elements must be processed by a certain activity after their first occurrence in the workflow model, *Pattern c* can check adherence to this rule. The first occurrence of the data element dat causes the transition *dat* to fire and produce a token in *ERR*, where it remains as long as the activity *A* does not process the data element in the workflow model. Transition *A* will remove the token from *ERR* and create a new one in *OK* otherwise.

The deletion of a data element before the workflow's end can be checked for with *Pattern d*. When a token in *flag* shows that the data element dat has been used, an activity that deletes dat will trigger the transition *Del* which removes the flag token and creates a new one in *OK*. If dat has not been deleted when the workflow reaches its end, the transition *End* puts a token in *ERR*.

Typical *correctness* or *soundness* requirements for Petri nets include being free of deadlocks, leading from start to end on all or at least one path, and not leaving any tokens in non-end places (Trčka et al. 2009). Tokens are possibly left behind when an AND switch is followed by an OR join. The notion of "corresponding pairs" has been introduced for the according analyses (Liu and Kumar 2005). *Pattern d* has the same structure as the pattern for corresponding pairs of AND switches with AND joins. Assume the renaming of *dat* to *AND-switch*, *Del* to *AND-join*, and *End* to *OR-join*. For every opening AND switch in the workflow, a token in *flag* is created. If a closing AND join follows, the *flag* token is deleted and a new one is put in *OK*, indicating a corresponding pair. If a closing OR join follows, the *flag* token is deleted and *ERR* receives a new one, signaling that potentially, at this point a token has been created in the workflow model without being consumed before the end of the workflow.

Checking for the order of activities is more complex, as *Pattern e* shows. The rule is that activity *B* may only happen if activity *A* was performed before. In the pattern, transition *A* fires when activity *A* is found, producing a token in place *OK/flag*. Upon activity *B* in the workflow model, transition *B* fires. Both transitions *A* and *B* have extended firing rules. If a token in *OK/flag* is present, activity *A* has been performed before and *B* creates a token in *OK*. If *OK/flag* is empty, a token in *ERR* is created, showing that *B* occurs in the workflow model before *A*, which is against the rule. To further support the analyst, the pattern will also indicate if the activities *A* and *B* happen in the wrong order. In that case, *B* will have produced a token in *ERR* before *A* fires. If *A* finds a token in *ERR*, it produces a token in the place *WrongOrder* instead of *OK/flag*.

Some rules set time limits for starting or ending an activity, but actual execution times cannot be determined at designtime. Still, even at designtime evidence of possible violations can be obtained by assuming the use of control activities. Such control activities check the actual execution times during runtime and create a warning if a time limit rule is violated. Checking for the presence of such control activities in the workflow

```
 1: For each workflow path in WF from P_start to P_end {          // visit all WF paths
 2:    Identify current WF:transition
 3:    For each matching RPN:transition {                         // trigger applicable RPNs
 4:       Fire RPN:transition depending on WF:annotations
 5:    }
 6:    Move to next WF:transition(s), create unique tokens when branching
 7: }                                  // loop will end when all paths have been visited
 8: For each error place in RPN {                                  // list rule violations
 9:    For each token in the error place {
10:       Identify WF:path through control token ID
11:       Identify WF:transition responsible for violation
12:       Evaluate RPN:annotations and provide details to analyst
13:       For WrongOrder errors, identify and show both WF:transitions
14:    }
15: }                                  // loop will end when all violations have been analyzed
```

Fig. 4 Comcert analysis algorithm

model, Comcert indicates possible violations of time limits. With a pattern similar to *Pattern e*, Comcert checks whether a workflow model contains the required control activities in the desired order. Please note a small but important difference to *Pattern e*: While in *Pattern e*, the rule allows that *A* occurs without *B* following, in the pattern for control activities it is a rule violation if *A* occurs without an *afterward* control activity.

4.3 Comcert Analysis

The compliance survey in Sect. 3 shows that although compliance regulations contain various rules, the compliance or rule adherence of workflows can be determined by checking for a few types of rule classes. Analyzing the control flow and data flow in a workflow model with respect to the five elements activities, data, locations, resources, and time limits covers these classes. While presenting the analysis algorithm, the Petri net containing the workflow will be referred to as workflow model, and the Petri nets containing the rules will be referred to as RPNs.

Filling the patterns of Sect. 4.2 with actual values that correspond to a workflow leads to RPNs that are ready for analysis. The names of activities, data elements, locations and resources should be used consistently between the workflow, the workflow model, the natural language or XML-style rules, and the RPN. Without consistent naming, a mapping must be available so that the analysis algorithm can determine which element in the workflow model corresponds to what element in the RPN.

For the patterns that carry predicates, the workflow and rule attributes must be assigned to the transitions and tokens of a RPN. Comcert uses colored Petri net tokens to differentiate between control flow tokens and data flow tokens. In addition, data flow tokens have colors that indicate the type of data element and resource they represent.

Comcert analyzes a workflow model for rule adherence by following each path from the workflow's start to its end and triggering the corresponding RPN transitions along the way. Correspondence depends on several aspects. Basically, the names of workflow and rule transitions are compared. Upon encountering a certain workflow transition, the rule transitions with the same name are triggered, firing if the required tokens are available in the RPN.

The extended firing behavior mentioned above results from considering predicates and the available tokens. For each source place it can be defined how many (including null) tokens of which kind (control token, data elements token) are required for firing and will be consumed when firing. How many and which tokens will be produced in which sink place can be defined accordingly.

For some of the patterns, a transition name comparison is sufficient, e.g., for *Pattern a* that checks for the presence of required activities. Checking for data deletion as in *Pattern d* requires that either there is a transition with a unique name, e.g., "delete", or that a transition carries a "delete" annotation.

Some patterns will always yield a boolean decision, e.g., checking for a required order of activities with *Pattern e*. Other patterns depend on certain predicates being evaluable, i.e., certain workflow model attributes being available. For practical application, those patterns can be extended with an end place that signals an unresolved analysis result. For example, *Pattern b* would then indicate rule adherence with tokens in *OK*, rule violations with tokens in *ERR*, and occurrences of unresolved cases with tokens in a place called, e.g., *UNRES*. Comcert supports the indication of missing annotations by creating "unresolved" tokens of a certain color whenever the rule adherence of a workflow activity, data element, resource or time limit cannot be resolved.

While visiting each path through the workflow model, Comcert triggers the applicable RPNs, so that finally, the RPN tokens indicate rule adherence or rule violation, depending on the semantics of the place they reside in. Each path receives its own number, following the scheme of binary trees, but extended to more than two possible branches. For example, 3.4.1 marks the path that is reached when taking the third branch of the first split, then the fourth branch of the second split, and the first branch of the third split. For space reasons, further details have to be omitted. Analysis with this numbering scheme allows an unlimited number of branches and includes loop detection. Analysis complexity will be discussed below.

With a workflow model and a set of RPN at hand, the Comcert analysis algorithm is described by the pseudo code depicted in **Fig. 4**.

Example 2 In the Workflow 1 (top of **Fig. 5**), a hospital requests PHI for treatment purposes. The EHR provider then

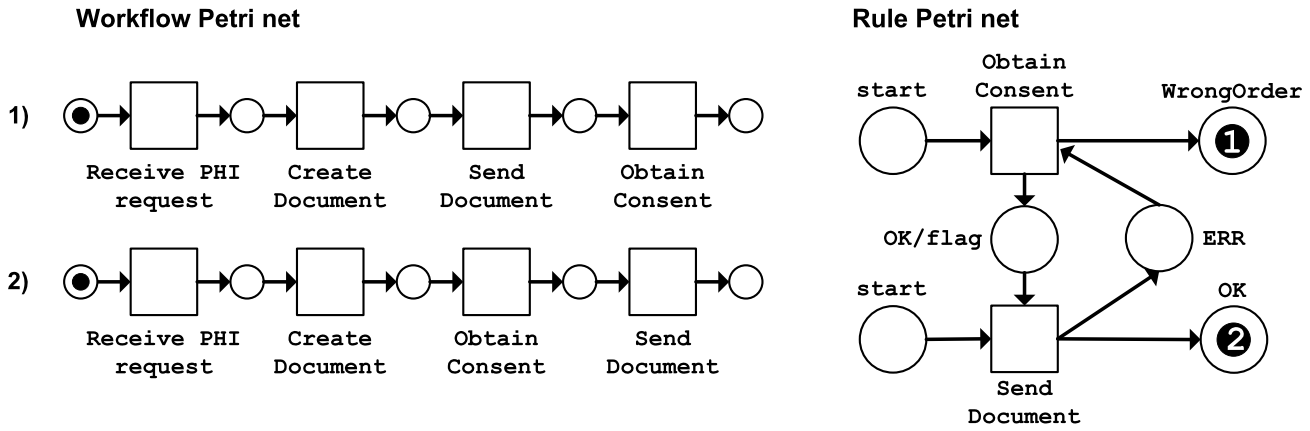

Fig. 5 Exemplary workflow with according rule Petri net

wants to create a document out of this PHI and send it to a third party. However, instantiating *Pattern e* for the order of activities, the RPN (right side of **Fig. 5**) rules that the activity "Obtain Consent" must be performed *before* the EHR provider is allowed to "Send Document". Upon finding the workflow transition "Obtain Consent" *after* "Send Document" in Workflow 1, Comcert sets the RPN to the state "WrongOrder" (indicated by the token "1" in the RPN). This information can be used to fix the workflow.

Now let us assume the workflow contains the required activities in the correct order as shown in Workflow 2 (bottom of **Fig. 5**). "Send Document" happens *after* "Obtain Consent", so a token (numbered "2" in **Fig. 5**) in the place *OK* is created. In this case, the workflow adheres to the rule and a corresponding certificate can be issued.

Some other approaches first create all traces (paths through the Petri net) and then analyze those traces in a second step (Meda et al. 2010). Comcert performs both tasks at once by creating unique control tokens for each workflow path within the Petri nets. Flooding the net in this way makes sure that every path is visited while at the same time allowing the analysis of single traces resp. paths.

Comcert, just like other Petri net analysis algorithms, has to cope with the state space explosion problem. In the worst case, the analysis runtime is exponential to the number of branching places and transitions in the workflow model. We do not expect this to be a problem in practice, because typical industrial workflows do not contain more than a few hundred activities. For example, the Comcert prototype analyzed a workflow model that contained more than 300 places and 300 transitions, resulting in more than 7000 workflow paths, in less than 2 seconds on a 1.2 GHz processor, using less than 6MB of memory. While tests have shown that nested loops considerably increase analysis time and memory usage, typical workflow models consisting of up to a few hundred places and transitions can be analyzed in seconds. In fact, small workflows of about 100 activities mostly required less than 15 ms to analyze.

The above example exhibits a useful property of Comcert. Besides the certification of a workflow, the approach also provides for pointers to flaws in the workflow, along with the affected part of the rule being violated. For example, the annotated error state that results from the analysis of the flawed Workflow 1 in **Fig. 5** identifies the missing activity "obtain consent" between Step 1 and Step 2 of the workflow. With patterns such as *Pattern e*, Comcert is able to even identify those activities in the workflow that occur too late, i.e., would make the workflow adhere to the rule if they occurred earlier. When activities occur in the wrong order, such as "obtain consent" and "send document" in **Fig. 5**, the required order of activities can be displayed to the analyst. Based on this automated identification, flaws can be corrected with automated process rewriting as described by Höhn (2009).

5 Summary

A chief requirement for the provision of sustainable Cloud Computing is the ability to certify the adherence of business processes to regulatory requirements. Currently, certification is manual, which is definitely at odds with the dynamics and flexibility offered by, e.g., the Software as a Service paradigm. The business models behind modern service technologies in general and Cloud Computing in particular essentially depend on business processes being tailored to the individual needs of customers. Failing to reliably support adaptivity will hamper the widespread adoption of Cloud Computing.

This paper provides two main contributions: A classification of compliance rules, and Comcert, an automated approach for the certification of business process compliance. Comcert is a well-founded approach based on Petri nets that allows a company, an auditor or a customer to check whether a business process, formalized in standard languages such as BPMN and BPEL, adheres to compliance requirements. In the event of noncompliance, the resultant evidence indicates the flawed fragments of the process. To further facilitate the certification process, this paper also classifies recurring types of compliance requirements, demonstrating how to formalize them as Petri net patterns for subsequent verification.

The analysis results do not formally prove that under all circumstances the workflow will adhere to all compliance rules at runtime. Rather, Comcert produces evidence of a workflow model's rule adherence, or, in the case of nonadherence, identifies design vulnerabilities. For example, in the health care scenario, compliance violations can be identified at a very early point in time, so that design vulnerabilities regarding the treatment of EHR data can be resolved before patients file HIPAA complaints. This avoids or reduces remediation costs,

which typically are very high when vulnerabilities are fixed later. Even though no formal proof in the strict sense is achieved, a complete analysis in the style of model checking is performed. The results are summarized in a compliance certificate that demonstrates the workflow's rule adherence.

Completeness of the checks is achieved in the sense that for all defined rule patterns, every occurrence of a pattern is detected within the workflows. However, rule pattern completeness is undecidable in general, because there is no formal way of proving that future attackers will not be able to think of new attacks against which additional rules would have to be defined. Overall, the goal of Comcert is to ease and automate compliance checks and provide adequate tool support. Companies can use Comcert to check workflows for rule adherence, either when about to put (parts of) their own workflows into the Cloud, or for certification to signal potential industrial or private customers that data will be processed in a compliant way.

Experiments carried out with Comcert demonstrate that it effectively detects possible compliance violations in workflows. While the rule patterns – capturing explicit information flow, i.e., control flow and data flow characteristics – account for the most relevant properties in practice, subtle structural vulnerabilities within the implicit information flow cannot yet be detected using Comcert. Such subtle vulnerabilities potentially lead to information leaks, which are not acceptable in high security scenarios. To also suit to such certification (Accorsi and Wonnemann 2011), we plan to extend Comcert with patterns to identify the so-called "covert-channels" (Lampson 1973).

References

Accorsi R, Wonnemann C (2011) Strong nonleak guarantees for workflow models. ACM, SAC, pp. 308–314

Atluri V, Chun SA, Mazzoleni P (2001) A Chinese wall security model for decentralized workflow systems. ACM conference on computer and communications security. ACM, New York, pp 48–57

BDSG (2009) Bundesdatenschutzgesetz. German Federal Ministry of Justice

Breaux TD, Antón AI (2008) Analyzing regulatory rules for privacy and security requirements. IEEE Trans Software Eng 34(1):5–20

Breaux TD (2009) Legal requirements acquisition for the specification of legally compliant information systems. PhD thesis, North Carolina State University

Cabanillas C, Resinas M, Ruiz-Cortés A (2010) Hints on how to face business process compliance. In: Resinas M, Ruiz-Cortés A, Pastor JA, Sancho MR (eds) Proc JISBD 4, pp 26–32

Chow R, Golle P, Jakobsson M, Shi E, Staddon J, Masuoka R, Molina J (2009) Controlling data in the cloud: outsourcing computation without outsourcing control. In: Proc 2009 ACM workshop on cloud computing security. ACM, New York, pp 85–90

COMPAS (2008) Compliance-driven models, languages, and architectures for services. EU FP7 Project 215175, deliverable 2.1 "State of the art in the field of compliance languages"

CSA (2009) Security guidance for critical areas of focus in cloud computing. Cloud Security Alliance. http://www.cloudsecurityalliance.org/. Accessed 2010-06-29

CSA (2010) Top threats to cloud computing. Cloud Security Alliance. http://www.cloudsecurityalliance.org/. Accessed 2010-06-29

Curtis B, Kellner MI, Over J (1992) Process modeling. Comm ACM 35(9):75–90

Dijkman R, Dumas M, Ouyang C (2008) Semantics and analysis of business process models in BPMN. Information & Software Technology 50(12):1281–1294

Ehrig M, Koschmider A, Oberweis A (2007) Measuring similarity between semantic business process models. ACS CRPIT 67:71–80

Etro F (2009) The economic impact of cloud computing on business creation, employment and output in Europe. Review of Business and Economics 54(2):179–218

European Commission (1995) Directive 95/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data

ENISA (2009) Cloud computing—benefits, risks and recommendations for information security. European Network Information and Security Agency

Ghose A, Koliadis G (2007) Auditing business process compliance. Springer LNCS 4749:168–180

GLB (1999) Gramm-Leach-Bliley Act. In: Congress of the USA

Governatori G, Hoffmann J, Sadiq SW, Weber I (2009) Detecting regulatory compliance for business process models through semantic annotations. Springer LNBPI 14:5–17

Hayes B (2009) Cloud computing. Comm ACM 51(7):9–11

HIPAA (1996) Health insurance portability and accountability act. In: Congress of the USA

Höhn S (2009) Model-based reasoning on the achievement of business goals. In: ACM symposium on applied computing. ACM, New York, pp 1589–1593

Huang H, Kirchner H (2009) Component-based security policy design with colored Petri nets. Springer LNCS 5700:21–42

IIG (2010) BW2PN: BPEL+WSDL to Petri net transformation. Software tool developed at the University of Freiburg, IIG Telematics. http://www.telematik.uni-freiburg.de/comcert/. Accessed 2010-06-29

Katt B, Zhang X, Hafner M (2009) Towards a usage control policy specification with Petri nets. Springer LNCS 5871:905–912

Lampson B (1973) A note on the confinement problem. Commun ACM 16(10):613–615

Liu Y, Müller S, Xu K (2007) A static compliance-checking approach framework for business process models. IBM System Journal 46(2):335–361

Abstract

Rafael Accorsi, Lutz Lowis, Yoshinori Sato

Automated Certification for Compliant Cloud-based Business Processes

A key problem in the deployment of large-scale, reliable cloud computing concerns the difficulty to certify the compliance of business processes operating in the cloud. Standard audit procedures such as SAS-70 and SAS-117 are hard to conduct for cloud-based processes. The paper proposes a novel approach to certify the compliance of business processes with regulatory requirements. The approach translates process models into their corresponding Petri net representations and checks them against requirements also expressed in this formalism. Being based on Petri nets, the approach provides well-founded evidence on adherence and, in case of noncompliance, indicates the possible vulnerabilities.

Keywords: Business process models, Cloud computing, Compliance certification, Audit, Petri nets

Liu R, Kumar A (2005) An analysis and taxonomy of unstructured workflows. Springer LNCS 3649:268–284

Lohmann N, Verbeek E, Dijkman RM (2009) Petri net transformations for business processes—A survey. Springer LNCS 5460:46–63

Lowis L, Accorsi R (2010) Vulnerability analysis in SOA-based business processes. IEEE Transactions on Services Computing (in press)

Meda HS, Sen AK, Bagchi A (2010) On detecting data flow errors in workflows. Journal of Data and Information Quality 2(1):1–31

Monakova G, Kopp O, Leymann F, Moser S, Schäfers K (2009) Verifying business rules using a SMT solver for BPEL processes. GI LNI 147:81–94

Murata T (1989) Petri nets: properties, analysis and applications. Proc IEEE 77(4):541–580

Organisation for Economic Co-Operation and Development (OECD) (1980) OECD guidelines on the protection of privacy and transborder flows of personal data

Oryx (2010) The Oryx project. http://bpt.hpi.uni-potsdam.de/Oryx/WebHome. Accessed 2010-06-29

Ouyang C, Verbeek E, van der Aalst WMP, Breutel S, Dumas M, ter Hofstede AHM (2005) WofBPEL: a tool for automated analysis of BPEL processes. Springer LNCS 3826:484–489

Park J, Sandhu R (2004) The UCONABC usage control model. ACM Transactions on Information and System Security 7:128–174

Pretschner A, Hilty M, Basin D (2006) Distributed usage control. Comm ACM 49:39–44

Sadiq S, Governatori G, Namiri K (2007) Modeling control objectives for business process compliance. Business process management. Springer LNCS 4714:149–164

Saha D (2008) A hitchhiker's guide to galaxy a.k.a. Netweaver business process modelling. http://www.sdn.sap.com/irj/scn/weblogs?blog=/pub/wlg/10947. Accessed 2010-06-29

Schneider F (2000) Enforceable security policies. ACM Trans Inf Syst Secur 3(1):30–50

SOX (2002) Sarbanes-Oxley act. In: Congress of the USA

Stohr EA, Zhao JL (2001) Workflow automation: overview and research issues. Information Systems Frontiers 3(3):281–296

Svirskas A, Courbis C, Molva R, Bedžinskas J (2007) Compliance proofs for collaborative interactions using aspect-oriented approach. IEEE Congress on Services 1:33–40

TMG (2009) Telemediengesetz. German Federal Ministry of Justice

Trčka N, van der Aalst WMP, Sidorova N (2009) Data-flow anti-patterns: discovering data-flow errors in workflows. Springer LNCS 5565:425–439

van der Aalst WMP (1998) The application of Petri nets to workflow management. Journal of Circuits, Systems, and Computers 8(1):21–66

van der Aalst WMP (2003) Challenges in business process management: verification of business processing using Petri nets. Bulletin of the EATCS 80:174–199

van Dongen BF, Jansen-Vullers MH, Verbeek HMW, van der Aalst WMP (2007) Verification of the SAP reference models using EPC reduction, state-space analysis, and invariants. Computers in Industry 58(6):578–601

Wagner G (2002) How to design a general rule markup language. GI LNI 14:19–37

Wong PYH, Gibbons J (2008) Verifying business process compatibility. In: International conference on quality software. IEEE, pp 126–131