

# Automatisierte Compliance-Zertifizierung Cloud-basierter Geschäftsprozesse

Die bisher manuelle Compliance-Zertifizierung von Workflows kann besonders im Cloud-Computing den Ansprüchen an Einhaltung verschiedenster Regularien und hohe Flexibilität nicht gerecht werden. Mit Comcert wurde in diesem Beitrag ein Verfahren zur automatisierten Analyse von Workflows vorgestellt. Hält ein Workflow die aufgestellten Regeln nicht ein, lokalisiert Comcert die Schwachstellen im Workflow anhand von wiederverwendbaren Regelmustern. Die Ergebnisse der Analyse zur Design-Zeit können von Cloud-Providern als Zertifikat zur Signalisierung der Compliance eingesetzt werden. Auditoren können mit Comcert vor Ausführung der Workflows die Regeleinholung überprüfen, und für Kunden ist die Zertifizierung dank der wiederverwendbaren Muster nachvollziehbar. Unternehmen, die bisher wegen ungeklärter Compliance-Fragen dem Cloud-Computing skeptisch gegenüberstehen, können mit Comcert die Regeleinholung überprüfen, und Cloud-Provider können die Regeleinholung durch Zertifikate nachweisen.

DOI 10.1007/s11576-011-0269-z

## Die Autoren

**Dr. Rafael Accorsi** (✉)  
**Dipl.-Inf. Lutz Lewis**  
 IIG Telematik  
 Universität Freiburg  
 Friedrichstr. 50  
 79098 Freiburg  
 Deutschland  
[accorsi@iig.uni-freiburg.de](mailto: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](mailto:yoshinori.sato.uw@hitachi.com)

Eingegangen: 2010-07-01  
 Angenommen: 2011-02-09  
 Angenommen nach drei Überarbeitungen durch Prof. Dr. Müller.

This article is also available in English via <http://www.springerlink.com> and <http://www.bise-journal.org>: Accorsi R, Lewis L, Sato Y (2011) Automated Certification for Compliant Cloud-based Business Processes. *Bus Inf Syst Eng*. doi: [10.1007/s12599-011-0155-7](https://doi.org/10.1007/s12599-011-0155-7).

© Gabler Verlag 2011

## 1 Automatisierte Zertifizierung von Geschäftsprozessen

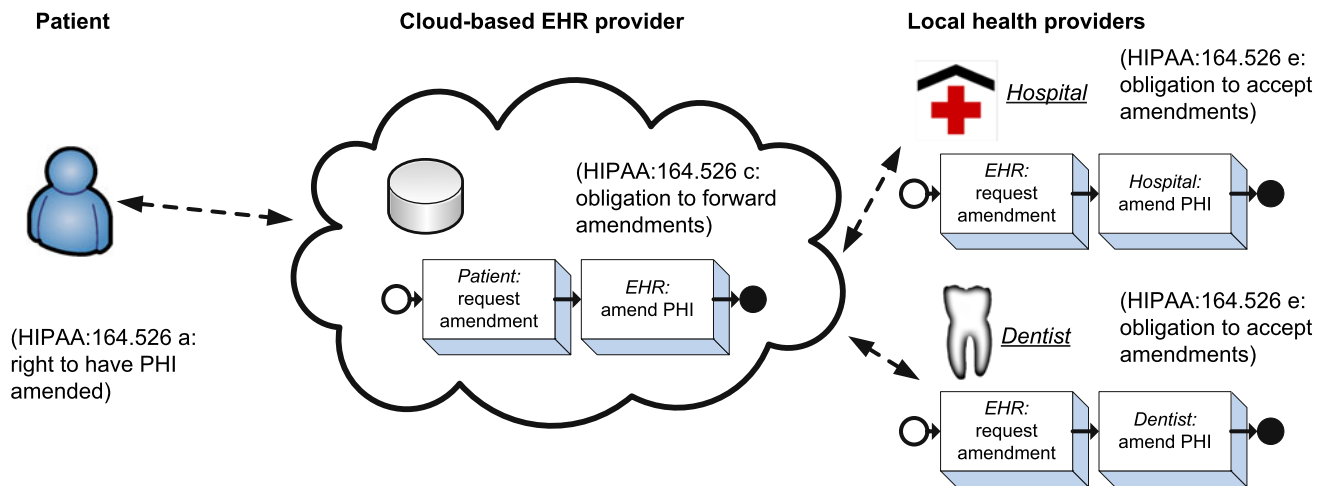
Die Compliance-Zertifizierung von Geschäftsprozessen ist eines der Schlüsselemente für nachhaltiges Cloud-Computing (Hayes 2009; ENISA 2009; CSA 2009, 2010). Wenn (Teile der) Geschäftsprozesse in die Cloud ausgelagert werden, müssen nicht nur die bisherigen Regeln eingehalten werden, sondern auch durch die Cloud-Nutzung neu hinzukommende Regeln. Wegen der aktuell meist ungeklärten Regeleinholung halten sich viele Unternehmen von Cloud-Computing fern. Hauptgrund sind Bedenken insbesondere bezüglich der Sicherheit und Privatsphäre.

Die Kontrolle über Prozesse im Cloud-Computing muss also nachweisbar sein. Die für die Zertifizierung nötige Prüfung findet allerdings bisher hauptsächlich manuell statt, was einen Wider-

spruch zu der durch Clouds versprochenen schnellen und flexiblen Anpassbarkeit von Geschäftsprozessen darstellt. Welche Aussagekraft hat eine jährliche Prüfung von Prozessen, die sich mehrmals im Jahr ändern?

Automatisierte Prüfverfahren werden nicht nur der Flexibilität wegen benötigt, sondern auch weil die Anzahl geltender Regeln die Prüfkomplexität erhöht. Gesetzliche und vertragliche Vorgaben werden zukünftig wohl nicht nur in ihrer Anzahl, sondern auch in ihrer Strenge wachsen. Eine einheitliche Notation der aus den Vorgaben resultierenden Regeln fehlt bisher (erste Schritte macht Breaux 2009). Die nicht zuletzt wegen fehlender Automatisierung mangelnde Prüfung und das damit verbundene Risiko von Compliance-Verstößen hält Unternehmen davon ab, Prozess(teil)e in die Cloud zu legen (Chow et al. 2009). Dadurch wiederum wird die Entfaltung des ökonomischen Potentials von Cloud-Computing verhindert (Etro 2009).

In diesem Artikel wird die Automatisierung und, als benötigte Grundlage dafür, die einheitliche Notation von Compliance-Regeln angestrebt. Zunächst erfolgt die Regelklassifikation, die aus einer Analyse bestehender Regularien abgeleitet wird. Für die drei sich ergebenden Oberklassen von Regeln werden formale Muster definiert. Anschließend wird das neue Verfahren Comcert vorgestellt, das die Muster zur automatisierten



**Abb. 1** E-Health-Szenario mit Beispiels-Workflows

Compliance-Prüfung von Geschäftsprozessen einsetzt. Comcert verwendet *Petrinetze* (Murata 1989) sowohl zur Notation der Geschäftsprozesse als auch für die Notation von Regeln. Die aus den Regeln gewonnenen Petrinetz-Muster werden während der Prozessanalyse, den Sicherheitsautomaten (Schneider 2000) ähnlich, je nach Prozessablauf geschaltet und zeigen strukturelle Schwachstellen bereits zur Design-Zeit an, also noch vor der tatsächlichen Ausführung der Prozesse.

Damit leistet Comcert einen Beitrag zur Automatisierung der Compliance-Zertifizierung von Cloud-basierten Geschäftsprozessen, wodurch wiederum der breite Cloud-Einsatz unterstützt wird. Unternehmen können mit dem Verfahren die Einhaltung von Regeln, besonders Sicherheits- und Privatsphären-Regeln prüfen und damit Bedenken gegen die Cloud-Nutzung abbauen. Dank der universell einsetzbaren Petrinetz-Darstellung kann Comcert unabhängig von der für die jeweiligen Geschäftsprozesse verwendeten Notation wie Business Process Modeling Notation (BPMN), Business Process Execution Language (BPEL) oder Ereignis-Prozessketten (EPK) eingesetzt werden. Aus der Prüfung sowohl des Kontrollals auch des Datenflusses innerhalb eines Prozesses werden fundierte Evidenzen für Compliance erzeugt.

Der zweite Abschnitt motiviert die Entwicklung des Comcert-Verfahrens anhand eines E-Health-Szenarios im Cloud-Computing, bei dem HIPAA-Regeln eingehalten werden müssen. Nach der Beschreibung des Anwendungsfalls fasst der dritte Abschnitt eine Auswahl

von Compliance-Regularien zusammen und stellt eine Klassifikation der enthaltenen Regeln vor. Aus diesen Regeln werden im vierten Abschnitt Regelmuster abgeleitet, und darauf aufbauend wird das Comcert-Analyseverfahren eingeführt und diskutiert. Der fünfte Abschnitt schließt den Artikel.

## 2 Exemplarischer Anwendungsfall

Der U.S. Health Insurance Portability and Accountability Act (HIPAA 1996) verfügt im Vergleich zu vielen anderen Compliance-Regularien über ein sehr detailliertes Regelwerk und eignet sich daher hervorragend als laufendes Beispiel. HIPAA reguliert die Verwendung sogenannter „protected health information (PHI)“ in Geschäftsprozessen. **Abbildung 1** illustriert das E-Health-Szenario mit Workflow-Beispielen und HIPAA-Regeln. Ein Cloud-Provider bietet Dienste für „electronic health records (EHR)“ an, die von Gesundheitsdienstleistern wie Krankenhäusern und Zahnärzten genutzt werden, um Patientendaten zu verwalten. **Abbildung 1** stellt kein einzelnes Prozessmodell dar, sondern veranschaulicht das Zusammenspiel der Parteien und ihrer Workflows.

Ein *Workflow* ist ein fallbasierter, zur Automatisierung geeigneter Geschäftsprozess, d. h. er hat einen definierten Start- und Endpunkt und stellt eine konkrete Instanz eines Geschäftsprozesses dar. Hauptsächlich beschreibt ein Workflow einen *Kontroll-* und einen *Datenfluss*. Der Kontrollfluss legt fest, welche *Aktivitäten* in welcher *Reihenfolge*

stattfinden. Der Datenfluss legt fest, welche *Daten* zwischen den Aktivitäten ausgetauscht werden. Im Beispiel der **Abb. 1** läuft beim EHR-Provider ein Workflow, dessen Kontrollfluss von der Aktivität „request amendment“ zur Aktivität „amend PHI“ übergeht. Der Datenfluss besteht aus der eingehenden Anfrage und den Daten der geänderten PHI.

Im Beispielszenario verwenden Patienten eine Online-Gesundheitsakte, wie sie etwa durch Google Health oder Microsoft HealthVault angeboten wird. Der EHR-Provider wiederum ist direkt mit weiteren Parteien wie Krankenhäusern und Zahnärzten verbunden. Patienten haben das Recht, die über sie gespeicherten Daten anpassen zu lassen (HIPAA:164.526a). In einem solchen Fall müssen die Änderungen an alle beteiligten Parteien weitergeleitet werden (HIPAA:164.526c). Jede einzelne Partei wiederum ist verpflichtet, die Änderung aufzunehmen und die Daten anzupassen (HIPAA:165.526e).

Allgemein stellen sich drei Fragen, wenn ein Unternehmen Cloud-Computing einsetzt. Erstens will das Unternehmen wissen, welche (Teile) seiner Geschäftsprozesse ausgelagert werden können ohne Compliance-Regeln beispielsweise bezüglich der Verarbeitungs- und Speicherorte zu verletzen. Zweitens müssen Auditoren die Regeleinhaltung vorab prüfen und brauchen dazu Ansatzpunkte. Drittens müssen Kunden die Möglichkeit bekommen, die angebotenen Cloud-Dienste daraufhin einzuschätzen, ob persönliche Daten regelkonform, insbesondere die Privatsphäre während verarbeitet werden.

Bestehende Analyseverfahren zielen vor allem auf den Kontrollfluss ab, z. B. (van der Aalst 2003; Ehrig et al. 2007; Wong und Gibbons 2008). Ghose und Koliadis (2007) – ähnlich wie Governatori et al. (2009) – legen bei ihrem Compliance-Audit den Fokus auf einzelne Aktivitäten und Zeitlimits. Bei diesen Ansätzen kommt modale temporale Logik zum Einsatz. Meda et al. (2010) analysieren explizit den Datenfluss eines Workflows. Atluri et al. (2001) untersuchen die Vertraulichkeit von Daten in Workflows. Monakova et al. (2009), Liu et al. (2007) und Trčka et al. (2009) machen erste Schritte in Richtung der kombinierten Kontroll- und Datenflussanalyse von BPEL-Prozessen. Allerdings werden Usage-Control-Regeln dabei nicht untersucht, weswegen nicht alle der im dritten Abschnitt vorgestellten Regeln abgedeckt werden können. Comcert hingegen umfasst auch Usage-Control-Regeln bzw. -Regelmuster.

Die obigen drei Fragen kann Comcert beantworten, wenn ein Workflow-Modell und die einzuhaltenden Regeln vorliegen. Technisch reduzieren sich die drei Fragen auf eine, nämlich: „Hält der Workflow *W* die Regel(n) *R* ein?“ Organisatorisch gesehen unterscheiden sich die drei Blickwinkel „Cloud-Provider, Auditoren, Kunden“ vor allem durch die jeweils verfügbaren Informationen. Den Providern liegen alle Details vor. Die Auditoren sehen was die Unternehmen ihnen zur Verfügung stellen; das kann, muss aber nicht alles sein. Die Kunden sehen wenige bis keine Details der Geschäftsprozesse. Nachfolgend liegt der Fokus auf der technischen Sicht, wobei davon ausgegangen wird, dass die zur Prüfung benötigten Workflow-Details vorliegen. Dritte Parteien, die im Auftrag von Kunden die Prüfung durchführen sollen, werden im Folgenden nicht betrachtet.

### 3 Compliance-Regularien: Übersicht und Klassifikation

Vor einer Compliance-Prüfung muss einerseits klar sein, welche Regeln eingehalten werden müssen. Andererseits muss festgelegt werden, wie die Einhaltung überprüft werden soll. In diesem Abschnitt werden Compliance-Regularien und die enthaltenen Regeln vorgestellt.

#### 3.1 Übersicht über Compliance-Regularien

*OECD-Richtlinien* In den „OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data“ (1980) werden acht Datenschutzprinzipien definiert: Begrenzung der Datensammlung, Qualitätsansprüche an Daten, Zweckbindung, Begrenzung der Verwendung, Sicherheitsvorgaben, Offenheit, individuelle Beteiligung und Zurechenbarkeit. Durch Updates der Datensätze soll eine hohe Datenqualität erreicht werden, und Benachrichtigungen an die Benutzer über die Verwendung ihrer Daten soll Offenheit erreichen. Aus Workflowsicht handelt es sich um Datenflussvorgaben. Datensammlungen sollen auf die für einen bestimmten Zweck nötigen Umfang begrenzt werden und nur die direkt beteiligten Parteien sollen die Daten erhalten.

*EU-Direktive 95/46/EC* In der Europäischen Union reguliert die Direktive 95/46/EC den Umgang mit persönlichen Daten und die Übertragung dieser (European Commission 1995). Workflows halten die Direktive ein, wenn sie Daten nur für einen bestimmten Zweck verwenden (95/46/EC:6.1.b) und nur die unbedingt nötige Menge an Daten erheben und verarbeiten (95/46/EC:6.1.c). Unternehmen dürfen persönliche Daten nur so lange wie für den angegebenen Zweck nötig aufbewahren (95/46/EC:6.1.e), sollen beim Kunden die Einwilligung zur Verwendung persönlicher Daten einholen (95/46/EC:7.a), dürfen bestimmte persönliche Daten gar nicht verwenden (95/46/EC:8.1) und müssen Kunden über die Verwendung persönlicher Daten informieren (95/46/EC:10,12). Allgemein müssen angemessene Schutzmechanismen eingesetzt werden (95/46/EC:17.1), und persönliche Daten dürfen nur dann in andere Länder übertragen werden, wenn diese Länder ein angemessenes Schutzniveau hinsichtlich Sicherheit und Privatsphäre bieten (95/46/EC:25).

*Telemediengesetz (TMG)* Das TMG beschreibt, welche persönlichen Daten wie gesammelt und von Onlinediensten verwendet werden dürfen (TMG 2009). Personenbezogene Daten dürfen nur dann in Workflows Verwendung finden, wenn das gesetzlich erforderlich ist oder Kunden ihre Einwilligung gegeben

haben (TMG:12). Unternehmen müssen ihre Datenschutzregeln veröffentlichen (TMG:13). Kunden müssen Dienste benutzen können, ohne dass die Benutzung für andere ersichtlich ist, und Kundenprofile dürfen wenn überhaupt dann nur erzeugt werden, wenn Pseudonyme verwendet werden (TMG:13.4). Die Datensammlung muss auf das Nötigste beschränkt werden (TMG:15.1), und Abrechnungen dürfen keine Nutzungsdetails enthalten, sofern der Kunde es nicht explizit anders wünscht.

*Bundesdatenschutzgesetz (BDSG)* Das BDSG (BDSG 2009) setzt die EU-Direktive 95/46/EC um. Es erlaubt die Verwendung persönlicher Daten nur bei gesetzlicher Notwendigkeit oder mit expliziter Kundeneinwilligung (BDSG:4). Persönliche Daten dürfen nur dann an – u. U. im Ausland befindliche – dritte Parteien übertragen werden, wenn ein ausreichendes Schutzniveau sichergestellt ist (BDSG:4.b). Kunden haben das Recht, ihre Daten ändern oder löschen zu lassen (BDSG:6). Technische und organisatorische Sicherheitsmaßnahmen müssen durch die Unternehmen getroffen werden, hauptsächlich Zugriffs- und Nutzungskontrolle (BDSG:9). Dritte Parteien müssen dieselben Regeln einhalten wie die Unternehmen mit direktem Kundenkontakt (BDSG:11). Außerdem darf die Verwendung nur zu einem bestimmten Zweck stattfinden (BDSG:28), und Unternehmen müssen ihre Kunden über die Datenverwendung informieren (BDSG:34).

*Gramm-Leach-Bliley Act (GLB)* Ein Regularium aus dem Finanzsektor ist der GLB, worin Zusammenschlüsse von Banken und Versicherungsfirmen behandelt werden (GLB 1999). GLB:501 *a* und *b* fordern, dass sogenannte „nonpublic personal information“ (NPPI) durch angemessene Sicherheitsstandards geschützt werden. GLB:502 verbietet die Offenlegung und Weitergabe von NPPI an dritte Parteien. GLB:503 verlangt von Finanzinstitutionen, die Kunden über die jeweiligen Datenschutzregeln zu informieren. Der thematisch ähnliche, aber sektorübergreifende Sarbanes-Oxley-Act (SOX 2002) ist aus Workflowsicht noch abstrakter formuliert und enthält keine direkt auf Workflows bezogenen Vorgaben.

*Health Insurance Portability and Accountability Act (HIPAA)* Der HIPAA-Abschnitt „Administrative Rule“ reguliert die Verwendung von Gesundheitsdaten. HIPAA enthält sehr detaillierte Regeln für den Umgang mit Daten („protected health information“, PHI) und dient deswegen im Folgenden als laufendes Beispiel.

Im Sicherheitsteil (164 C) werden allgemeine Sicherheitsmaßnahmen gefordert. Im Teil zur Privatsphäre (164 E) wird hingegen beschrieben, wie PHI in Workflows verwendet werden dürfen. Ausgehend vom „default deny“-Ansatz (HIPAA:164.502.a) dürfen PHI nur für die Behandlung, Abrechnung oder den Betrieb von Gesundheitseinrichtungen verwendet werden (HIPAA:164.502.a.1). Sowohl für den Sender wie für den Empfänger von PHI gilt die „minimum necessary“-Anforderung (HIPAA:164.502.b, vgl. Datensparsamkeit). Wenn eine an der Verarbeitung beteiligte dritte Partei PHI erhält, die sie nicht zwingend benötigt, ist sie verpflichtet, den Absender darüber zu informieren (HIPAA:164.504.e.2.ii.C). Wie üblich muss die Einwilligung der Patienten vorliegen bevor PHI verwendet werden (HIPAA:164.506.a.1). Falls PHI doch ohne Einwilligung verwendet werden, muss der Fall dokumentiert und nachträglich aufgeklärt werden (HIPAA:164.506.a.3.C.ii). Gewisse PHI dürfen ohne Einwilligung verwendet werden, um beispielsweise ein Patientenverzeichnis zu führen (HIPAA:164.510.a.1). Andere PHI hingegen dürfen nur mit spezieller, zusätzlicher Genehmigung verwendet werden, so etwa Angaben zur Psychotherapie (HIPAA:164.508.a.2). Patienten dürfen die Verwendung ihrer PHI auf bestimmte Datenelemente beschränken lassen (HIPAA:164.502.c). Sie haben das Recht über die zu ihnen gespeicherten PHI informiert zu werden (HIPAA:164.524.a.1), und zwar innerhalb einer festgesetzten Frist (30 Tage, siehe HIPAA:164.524.b.2.i). Außerdem muss Patienten auf ihren Wunsch hin eine Liste der Fälle zur Verfügung gestellt werden, in denen in den letzten sechs Jahren ihre PHI weitergegeben wurden (HIPAA:164.528). Patienten können die Anpassung der über sie gespeicherten PHI fordern (HIPAA:164.526.a.1). Solche Anpassungen müssen dann von der betroffenen Einrichtung an alle weiteren beteiligten Parteien weitergegeben werden (HIPAA:164.526.c). Diese Parteien wiederum sind verpflichtet, die Anpassungen aufzunehmen (HIPAA:164.526.e).

**Tab. 1** Klassifikation von Compliance-Regeln

	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	✓	✓	✓			

**3.2 Klassifikation von Compliance-Regeln**

Die auf Workflows bezogenen Abschnitte der obigen Compliance-Regularien lassen sich auf wenige Regelklassen reduzieren, wie die **Tab. 1** zeigt. Der Kürze halber werden die englischen Regelbezeichnungen verwendet.

Die Regelklassen in der zweiten Spalte wurden abgeleitet, indem die Compliance-Regularien der ersten Zeile auf Regeln durchsucht wurden, die sich entweder auf den Kontrollfluss (z. B. Reihenfolge von Aktivitäten) oder den Datenfluss (Verwendung von Daten) beziehen. Aus der Zusammenfassung nur in der Formulierung unterschiedlicher Regeln entstanden die neun Regelklassen.

Für jede dieser Regelklassen wurden dann die entsprechenden Regeln in eine Petrinetz-Darstellung transformiert, wie sie in Abschn. 4 dieses Beitrags vorgestellt wird. Bei der Untersuchung auf Übereinstimmungen und Unterschiede zwischen den Mustern stellte sich heraus, dass die Mehrzahl der Regelklassen (5 von 9) auf die Reihenfolge von Aktivitäten abzielen und jeweils zwei auf die Verzweigung innerhalb eines Workflows und auf die zu verarbeitenden Daten.

Die resultierenden drei Oberklassen werden durch die Icons in der linken Spalte symbolisiert. Ein Pfeil mit zwei Spitzen für Regeln, die die Ausführung einer Aktivität vor oder nach einer anderen Aktivität fordern. Ein verzweigender Pfeil für Regeln, die die bedingte Verarbeitung von Daten betreffen, sowie ein Rechteck für Regeln, die sich auf einzelne Datenelemente beziehen.

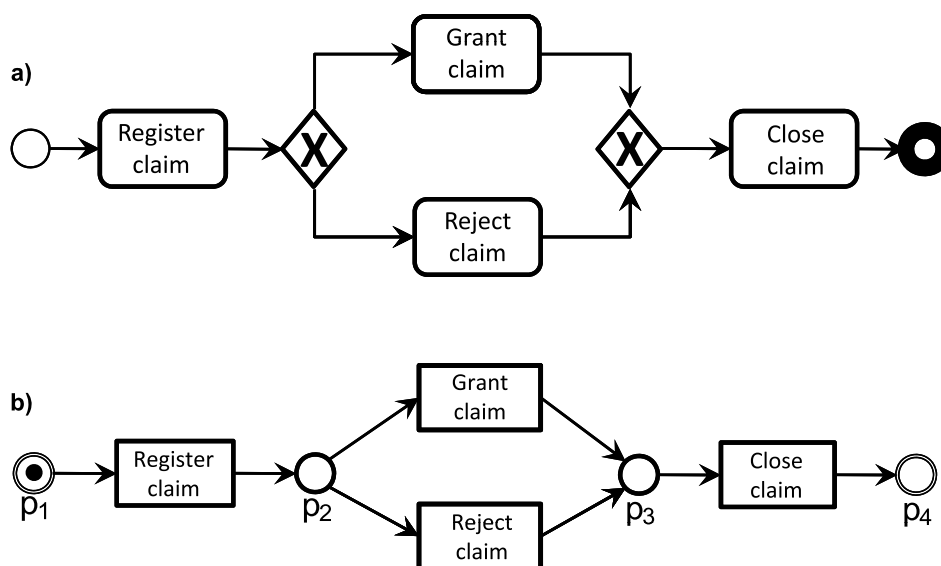
Zugriffskontrolle („access control“) ist unabdingbar für Cloud-basierte Workflows, aber nicht ausreichend um alle Regeln abzudecken. Zwar wird durch Zu-

griffskontrolle der erste Zugriff auf Daten gesichert, aber weder werden spätere Weitergaben noch andere Verwendungen abgedeckt. Im E-Health-Szenario könnte bei reiner Zugriffskontrolle ein Arzt auf Daten zu Behandlungszwecken zugreifen und diese anschließend zu Werbezwecken verwenden. Im Gegensatz zu anderen Compliance-Klassifikationen wie (Breux und Antón 2008; Wagner 2002) und (Sadiq et al. 2007) wurden bei der oben vorgestellten Untersuchung verschiedene Regularien berücksichtigt. Für Cloud-Computing-Anbieter ist es wichtig, Compliance zu einer Vielzahl von Regularien nachweisen zu können, da sie neben vertraglichen Vereinbarungen (z. B. aus Service-Level-Agreements) auch die Regeln einhalten müssen, die für die jeweiligen Sektoren gelten, aus denen der Nutzerkreis ihrer Dienste stammt. Insbesondere prüft Comcert neben Zugriffskontroll- auch Usage-Control- bzw. Nutzungskontroll-Regeln, ohne die viele Compliance-Anforderungen nicht erfüllt werden können. In dieser Hinsicht erweitert der vorgestellte Ansatz bestehende Usage-Control-Ansätze um die Analyse zur Design-Zeit (Park und Sandhu 2004 bieten ein Modell zur Erfassung von Usage-Control-Regeln, Pretschner et al. 2006 präsentieren einen Durchsetzungsmechanismus für die Ausführung).

**4 Analyse mit Comcert**

Dank ihrer formalen Semantik sind Petrinetze gut für die Analyse von Workflows geeignet (van der Aalst 1998). Workflows aus verschiedenen Spezifikationen wie BPMN, BPEL und EPC können damit erfasst und untersucht werden (Lohmann et al. 2009). Aufbauend

**Abb. 2** Beispiel einer BPMN- und Workflow-Net-Darstellung



auf die Standarddefinition der Petrinetze (Murata 1989) werden sowohl Workflows wie auch Regeln von Comcert in Petrinetz-Form verarbeitet. Die Annahme lautet dabei, dass Petrinetze dank ihrer graphischen Darstellung leichter zugänglich sind als die komplexen Formeln beispielsweise temporaler Logik. Davon ausgehend wird die Compliance-Prüfung für eine größere Gruppe von Anwendern nachvollziehbar, ohne dass die Qualität der Analyse sinkt.

Die Hauptidee bei der Comcert-Analyse ist es, einen Workflow in ein Petrinetz zu transformieren, die Regeln in weitere Petrinetze zu transformieren und dann die Einhaltung der Regeln durch Gegenüberstellung der Netze zu prüfen. Dazu wird jeder Pfad durch den Workflow bzw. durch das Workflow-Petrinetz untersucht, um beim Auftreten bestimmter Elemente die entsprechende Regel-Petrinetze (RPN) zu schalten. Ähnlich dem Ansatz der Sicherheitsautomaten (Schneider 2000) enthält jedes RPN spezielle Plätze, an denen sich der Status der Regeleinholung ablesen lässt. Nach dem Workflow-Durchlauf zeigen diese Plätze die Regeleinholung oder etwaige Regelverstöße an.

Comcert eignet sich damit zur Aufdeckung von Schwachstellen im Workflow zur Design-Zeit und auf Workflow-Ebene. Zur Untersuchung anderer Ebenen, wie etwa der Software und Hardware, müssen andere Verfahren herangezogen werden (Lowis und Accorsi 2010). Bisherige Analyse- und Zertifizierungsverfahren sind auf eine spezielle Workflow-Notation ausgerichtet. Comcert abstrahiert von speziellen Notationen und stellt sowohl Workflows wie

auch die geltenden Regeln als Petrinetze dar.

Die Details der von Comcert verwendeten Petrinetze werden bei der Vorstellung der Analysemuster diskutiert. Zunächst wird die Transformation der Workflows und Regeln in Petrinetze erläutert.

#### 4.1 Transformation und Modellierung von Workflows

Workflows werden also sogenannte „Workflow-Nets“ modelliert (van der Aalst 1998). Ein Workflow-Net ist ein Petrinetz mit speziellen Plätzen als Quellen und Senken. Jedes andere Element liegt zwischen Quellen und Senken. Ein Token in einer Quelle zeigt eine neue Workflow-Ausführung an, einen sogenannten „case“. Tokens in einer Senke zeigen eine beendete Ausführung an, also das Ende eines Case. Das folgende Beispiel veranschaulicht die Transformation eines BPMN-Workflows in ein Workflow-Net.

*Beispiel 1* BPMN formalisiert den Kontrollfluss eines Workflows mit drei Elementen: Aktivitäten, Ereignisse und Gateways. **Abbildung 2a** enthält ein Beispiel. Die Rechtecke repräsentieren Aktivitäten, die Rauten stehen für Gateways („exclusive choice“ links und „simple merge“ rechts). Der einfache Kreis links steht für das Ereignis „Start“, der breite Kreis rechts für das Ereignis „Ende“. **Abbildung 2b** zeigt das aus **Abb. 2a** abgeleitete Workflow-Net. Der Platz  $p_2$  wird als „OR-split“ bezeichnet,  $p_3$  ist ein „OR-join“ (entsprechend „exclusive choice“ und „simple merge“).

Die automatisierte Transformation von Workflow-Beschreibungen in Workflow-Nets kann beispielsweise mit Oryx (2010) oder SAP Galaxy (Saha 2008) erfolgen. Dabei hängt die Präzision des erzeugten Modells natürlich vom Ausgangs-Workflow ab. Für BPMN beispielsweise wurden Transformationen definiert, bei denen das Ausgangsmodell nur jeweils genau ein Start- und ein Endereignis haben darf (Dijkman et al. 2008). Bestimmte Aktivitäten mit mehrfachen, gleichzeitigen Instanzen sowie einige spezielle Arten von Gateways finden in Workflow-Nets keine direkte Entsprechung und müssen durch Hilfskonstrukte abgebildet werden. Gleiches gilt für EPK. Nur eine bestimmte Art von EPK kann direkt in Workflow-Nets übertragen werden (van Dongen et al. 2007). Im Gegensatz dazu kann die Transformation von BPEL-Workflows „feature-complete“, also ohne Verlust an Ausdrucksstärke erfolgen, z. B. mit WofBPEL (Ouyang et al. 2005). Comcert verwendet Workflow-Nets mit zusätzlichen Annotationen, die durch *BW2PN* (IIG 2010) erzeugt werden. BPEL-Workflows mitsamt ihrer WSDL-Spezifikation werden dafür in das Format der Petri Net Markup Language (PNML) transformiert. *BW2PN* ignoriert standardmäßig gewisse BPEL-Angaben zum Fault- bzw. Exception-Handling, um kompakte, übersichtliche Petrinetze zu erzeugen. Es handelt sich dabei um einen Trade-Off zwischen „feature completeness“ und Les- bzw. Nachvollziehbarkeit. Sollte es für das jeweilige Szenario unabdingbar sein, vollständiges Exception-Handling zu berücksichtigen, können entsprechend komplexe Petrinetze erzeugt werden.

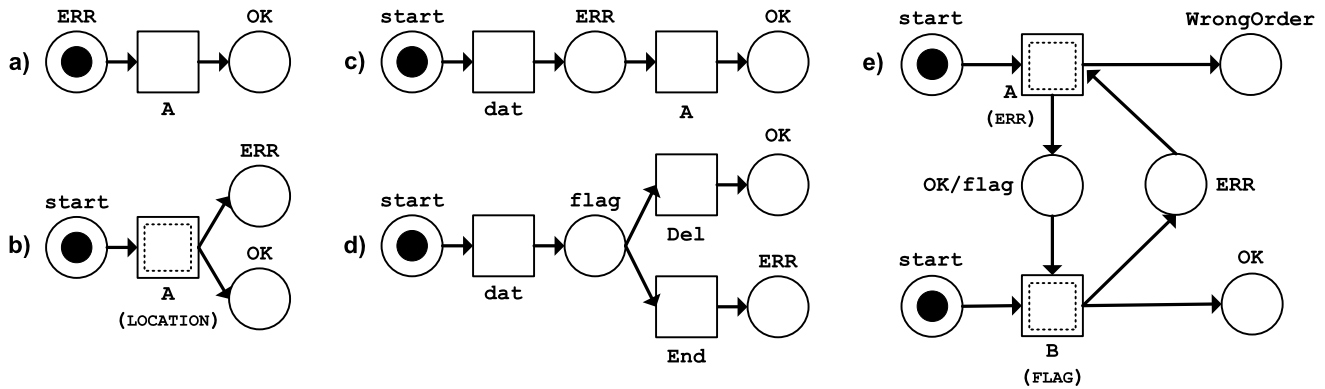


Abb. 3 Regel-Petrinetz-Muster (Auszug)

#### 4.2 Comcert-Muster für Compliance-Regeln

Comcert prüft die Einhaltung von Compliance-Regeln innerhalb eines Workflows durch die Analyse der für die Regeleinhaltung zentralen, fünf anerkannten Hauptelemente eines Workflows: *Aktivitäten, Daten, Orte, Ressourcen* und *Zeitlimits* (Curtis et al. 1992; Stohr und Zhao 2001; Sadiq et al. 2007; Svirskas et al. 2007; COMPAS 2008; Breaux 2009 und Cabanillas et al. 2010).

Eine Regel beschreibt, welche Aktivitäten von welchen Rollen mit welchen Daten (nicht) ausgeführt werden dürfen oder müssen. Zusätzlich kann eine Regel eine Reihenfolge von Aktivitäten festlegen, so dass bestimmte Aktivitäten vor oder nach anderen stattfinden müssen.

Die Formalisierung von Regeln durch Petrinetze wurde von Katt et al. (2009) sowie von Huang und Kirchner (2009) vorgeschlagen. Im Gegensatz zu Katt et al. berücksichtigen Huang und Kirchner in ihrer Arbeit keine Usage-Control-Policies. Katt et al. setzen Usage Control Colored Petri Nets (UCPN) zur Formalisierung ein und erlauben damit die Durchsetzung von Obligationen, also von Bedingungen, die vor, während oder nach der Ausführung bestimmter Aktivitäten erfüllt werden müssen. Allerdings gehen sie davon aus, dass die Regeln direkt in den Workflow integriert wurden. Diese Vermischung von Workflows und Regeln erschwert oder verhindert die Weiterverwendung von Regeln zur Prüfung anderer Workflows. Dem Prinzip der Sicherheitsautomaten folgend (Schneider 2000) werden Regeln von Comcert als einzelne Petrinetze vorgehalten und nicht in den Workflow integriert. In Kombination mit der Klassifikation von Compliance-Regeln können

die Regeln so in Kategorien eingeteilt und leicht wiederverwendet werden.

Ein Regel-Petrinetz (RPN) besteht aus den Standardelementen eines Petrinetzes: Plätze  $P$  (als Kreise dargestellt), Transitionen  $T$  (Rechtecke), Kanten  $K$  (Linien) und Tokens (Punkte). Kanten verbinden entweder  $P$  mit  $T$  oder  $T$  mit  $P$ , und Tokens liegen immer innerhalb eines Platzes. Die Plätze direkt vor einer Transition sind deren *Eingangsplätze*, die Plätze direkt hinter der Transition *Ausgangsplätze*. Eine Transition ist *aktiv*, wenn ihre Eingangsplätze Tokens enthalten, und eine aktive Transition kann *feuern*. Beim Feuern *verbraucht* eine Transition die Tokens ihrer Eingangsplätze und *produziert* neue Tokens in ihren Ausgangsplätzen. Das Standard-Feuerverhalten verbraucht einen Token aus jedem Eingangsplatz und erzeugt einen Token in jedem Ausgangsplatz.

Im Cloud-Computing können Teile eines Workflows an verschiedenen Orten ausgeführt werden, besonders wenn Dienste in andere Länder ausgelagert werden. Um die entsprechenden Regeln prüfen zu können, tragen die Petrinetz-Elemente Prädikate, aus denen die nötigen Details ausgelesen werden können. Der Ausführungsort einer Aktivität ist ein Beispiel dafür, ein anderes Beispiel ist der Notfallzustand eines Patienten. Diese Prädikate werden verwendet, um ein erweitertes Feuerverhalten zu steuern.

Erweitertes Feuerverhalten wird von Comcert verwendet, um das Verhalten bestimmter Transitionen deterministisch zu machen (in Abb. 3 sind deterministische Transitionen bzw. solche mit erweitertem Feuerverhalten durch ein gepunktetes Rechteck gekennzeichnet). Diese Transitionen feuern in Abhängigkeit der vorhandenen Tokens und der Prädikatbelegung. Auch das Nicht-Vorhandensein von Tokens in gewissen

Plätzen wird berücksichtigt. In Abb. 3 beispielsweise feuert die Transition  $A$  im Muster  $a$  in Abhängigkeit vom Ausführungsort einer Aktivität und Transition  $B$  im Muster  $e$  feuert in Abhängigkeit von der Anwesenheit eines Tokens im Platz *flag*. Durch das erweiterte Feuerverhalten wird die visuelle Komplexität von RPN verringert weil (oftmals sehr) viel weniger Plätze und Transitionen zur Modellierung benötigt werden als in klassischen Petrinetzen.

Die Plätze eines RPN können unterschiedliche Semantik besitzen. Die beiden wichtigsten Bedeutungen sind in Abb. 3 durch *OK* und *ERR* gekennzeichnet. Ein Token in *OK* signalisiert die Regeleinhaltung auf dem entsprechenden Workflow-Pfad. Regelverstöße werden durch Tokens in *ERR* angezeigt. Im Abschn. 4.3 wird vorgestellt, wie Tokens und Workflow-Pfade durch die Färbung bzw. Nummerierung von Tokens verknüpft werden.

Die Umwandlung einer natürlichsprachlichen Regel in ein RPN wird durch RPN-Muster unterstützt. So können Regeln bezüglich der fünf Elemente Aktivitäten, Daten, Orte, Ressourcen und Zeitlimits übersetzt werden. Aus Platzgründen wird nachfolgend nur eine Auswahl der Muster vorgestellt.

Mit dem Muster  $a$  der Abb. 3 kann das Vorkommen geforderter Elemente im Workflow-Modell geprüft werden. Solange die Aktivität  $A$  (Transition  $A$  im Muster) im Workflow nicht gefunden wurde, bleibt das RPN in seinem Ausgangszustand und zeigt die Nicht-Einhaltung der Regel durch ein Token im Platz *ERR* an. Wenn und sobald die Aktivität  $A$  beim Durchsuchen des Workflows gefunden wird, feuert die Transition  $A$  des RPN. Dadurch wird das Token aus *ERR* entfernt und ein neues wird in *OK* abgelegt.

Nach der Analyse des Workflows zeigt das RPN dann an, ob die geforderte Aktivität *A* im Workflow gefunden wurde. Ein Token in *OK* bestätigt die Regeleinhaltung, ein Token in *ERR* zeigt eine Regelverletzung. Anhand der Umkehrung der Semantik (*OK* wird *ERR* und umgekehrt) kann auf die Abwesenheit von verbotenen Aktivitäten geprüft werden.

Ortsvorgaben bezüglich der Ausführung von Aktivitäten können mit dem *Muster b* überprüft werden. Wenn die Aktivität *A* im Workflow-Modell gefunden wird, feuert die Transition *A* in Abhängigkeit vom annotierten Ort. Entspricht der Ort den Vorgaben, wird ein Token in *OK* erzeugt, was die Regeleinhaltung anzeigt. Widerspricht der Ort den Vorgaben, wird ein Token in *ERR* produziert und zeigt einen Regelverstoß an. Da die Aktivität *A* mehrfach im Workflow auftreten kann, muss dieser Umstand bei der Analyse berücksichtigt werden, was dank des erweiterten Feuerverhaltens aber keine zusätzlichen Plätze im Petrinetz beansprucht.

Wenn Datenelemente nach ihrer ersten Verwendung im Workflow durch eine bestimmte Aktivität verarbeitet werden müssen, kann diese Regel mit *Muster c* geprüft werden. Beim ersten Auftreten des Datenelements *dat* feuert die Transition *dat* und erzeugt ein Token in *ERR*, wo das Token einen Regelverstoß signalisiert solange die geforderte Verarbeitung durch die Aktivität *A* nicht stattfindet. Wird die Aktivität vor dem Ende des Workflows gefunden, feuert die Transition *A*, entfernt das Token aus *ERR* und erzeugt ein neues in *OK*.

Häufig wird die Löschung von bestimmten Daten vor dem Ende eines Workflows gefordert. *Muster d* dient zur Prüfung auf die Einhaltung dieser Regel. Ein Token in *flag* zeigt an, dass das Datenelement *dat* im Workflow verwendet wurde. Folgt eine dieses Element löschende Aktivität vor Workflow-Ende, wird die Transition *Del* ausgelöst, die das Token aus *flag* entfernt und ein neues in *OK* ablegt. Wurde bis zum Workflow-Ende das Element *dat* nicht gelöscht, produziert die Transition *End* ein Token in *ERR*.

Korrektheitsanforderungen an Petrinetze bestehen typischerweise aus der Deadlock-Freiheit, der Forderung, dass alle oder zumindest ein Pfad vom Anfang bis zum Ende des Workflows führen und dass am Workflow-Ende keine Tokens in Plätzen zurückbleiben, die nicht als Endzustände gekennzeichnet sind (Trčka et

al. 2009). Tokens bleiben möglicherweise zurück, wenn eine UND-Verzweigung von einer ODER-Vereinigung gefolgt wird. Für die entsprechende Analyse wurde von Liu und Kumar 2005 die Idee der „corresponding pairs“ vorgestellt. Das *Muster d* hat dieselbe Struktur wie das *Muster b* für korrespondierende Paare aus UND-Verzweigungen mit UND-Vereinigungen; angenommen sei die Umbenennung von *dat* zu *AND-switch*, *Del* zu *AND-join* und *End* zu *OR-join*. Für jede UND-Verzweigung im Workflow wird ein Token in *flag* abgelegt. Wenn eine UND-Vereinigung folgt, wird das Token aus *flag* entfernt und ein neues in *OK* produziert, was ein korrespondierendes Paar anzeigt. Folgt hingegen – regelwidrig – eine ODER-Vereinigung, wird das Token aus *flag* entnommen und ein neues in *ERR* abgelegt. Damit wird signalisiert, dass an dieser Stelle möglicherweise Tokens zurückbleiben und die Korrektheitsanforderungen verletzen.

Die Reihenfolgeüberprüfung benötigt ein komplexeres *Muster*, wie am *Muster e* zu sehen. Die dadurch umgesetzte Regel lautet, dass die Aktivität *B* nur auftreten darf, wenn die Aktivität *A* vorher im Workflow-Modell gefunden wurde. Die Transition *A* feuert wenn Aktivität *A* gefunden wird, und legt ein Token in *OK/flag* ab. Wird Aktivität *B* gefunden, feuert Transition *B*. Sowohl Transition *A* wie auch *B* besitzen erweitertes Feuerverhalten. Wenn ein Token in *OK/flag* liegt, wurde die Aktivität *A* bereits gefunden und die Transition *B* erzeugt ein Token in *OK*. Wenn *OK/flag* kein Token enthält, erzeugt *B* ein Token in *ERR*, wodurch angezeigt wird dass die Aktivität *B* im Workflow vor Aktivität *A* gefunden wurde. Die Analyse geht sogar noch einen Schritt weiter und zeigt an, ob die Aktivitäten in der falschen Reihenfolge auftreten (im Unterschied zu Aktivität *B* ohne folgende Aktivität *A*). In diesem Fall wird nämlich von *B* ein Token in *ERR* erzeugt bevor *A* feuert. Wenn *A* beim Feuern ein Token in *ERR* entdeckt, wird ein Token in *WrongOrder* produziert und nicht in *OK/flag*.

Manche Regeln legen einen bestimmten Zeitpunkt für den Start oder das Ende einer Aktivität fest. Die tatsächlichen Ausführungszeitpunkte können aber bei der Analyse zur Design-Zeit nicht ermittelt werden. Dennoch können bereits zur Design-Zeit Hinweise auf mögliche Verstöße solcher Zeitlimit-Regeln abgeleitet werden, wenn sogenannte Kontrollaktivitäten eingesetzt werden. Diese müssen

an all denjenigen Stellen im Workflow-Modell stehen, an denen die tatsächlichen Ausführungszeitpunkte überwacht werden müssen, um eine Warnung bei Verstößen erzeugen zu können. Comcert prüft auf die Einhaltung von Regeln bezüglich der Zeitlimits, indem die Anwesenheit von Kontrollaktivitäten untersucht wird. Mit einem *Muster* ähnlich dem *Muster e* wird überprüft, ob die Kontrollaktivitäten in der geforderten Reihenfolge im Workflow modelliert wurden. Ein kleiner aber wichtiger Unterschied zum *Muster e*: das *Muster e* erlaubt die Aktivität *A* ohne die folgende Aktivität *B*, wohingegen es bei Zeitlimits einen Regelverstoß darstellt wenn die Aktivität *A* auftritt ohne dass eine *nachfolgende* Kontrollaktivität gefunden wird.

### 4.3 Comcert-Analyse

Die Untersuchung der Compliance-Regularien in Abschn. 3 zeigt, dass trotz der Vielfalt von Compliance-Regeln die Compliance bzw. Regeleinhaltung in Workflows durch die Prüfung auf einige wenige Regeltypen abgedeckt werden kann. Dazu muss der Kontroll- und der Datenfluss in einem Workflow-Modell in Bezug auf die fünf Elemente Aktivitäten, Daten, Orte, Ressourcen und Zeitlimits analysiert werden. Bei der nachfolgenden Vorstellung des Analysealgorithmus wird das den Workflow darstellende Petrinetz als Workflow-Modell bezeichnet und die die Regeln enthaltenden Petrinetze als Regel-Petrinetze (RPN).

Zur Vorbereitung der Analyse müssen die in Abschn. 4.2 vorgestellten *Muster* mit den Werten belegt werden, die dem zu analysierenden Workflow entsprechen. Die Namen von Aktivitäten, Datenelementen, Orte und Ressourcen müssen einheitlich verwendet werden, also weder im Workflow-Modell noch in den natürlichsprachlichen oder XML-basierten Regeln oder den RPN unterschiedlich sein. Kann eine konsistente Namensgebung nicht erreicht werden, muss es einen automatisiert auslesbaren Index geben, so dass der Analysealgorithmus feststellen kann, welches Element im Workflow-Modell welchem Element im RPN entspricht.

Die *Muster*, bei denen Prädikate zum Einsatz kommen, setzen weiterhin voraus, dass die Werte aus dem Workflow-Modell den Transitionen und Tokens innerhalb der RPN zugeordnet werden. Comcert verwendet farbige Tokens zur

```

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

```

**Abb. 4** Comcert Analysealgorithmus

Unterscheidung von Kontroll- und Datenfluss. Auch einzelne Datenelemente und Ressourcen können so unterschieden werden.

Comcert überprüft ein Workflow-Modell auf Regeleinhaltung, indem jeder Pfad durch den Workflow von Anfang bis Ende durchlaufen wird und dabei die entsprechenden Transitionen in den RPN geschaltet werden. Entsprechende Transitionen werden anhand mehrerer Faktoren identifiziert. Hauptsächlich wird der Name von Workflow- und Regel-Transitionen verglichen. Bei gleichnamigen Transitionen wird die Regel-Transition angestoßen, so dass sie – wenn alle nötigen Tokens im RPN vorliegen – feuert.

Das bereits erwähnte, erweiterte Feuerverhalten basiert auf der Berücksichtigung von Prädikaten und der Art und Anzahl vorhandener Tokens. Für jeden Eingangsplatz einer Transition kann festgelegt werden, wieviele Tokens von welcher Sorte (Kontrollfluss, Datenfluss bzw. welches Datenelement) vorhanden sein müssen, um die Transition aktiv werden zu lassen. Ebenso kann festgelegt werden, wieviele und welche Tokens in welchem Ausgangsplatz produziert werden.

Für einige Muster genügt der Namensvergleich von Transitionen, so z. B. für das *Muster a* zur Prüfung auf die Anwesenheit geforderter Aktivitäten. Für die Prüfung auf Daten-Löschung mit *Muster d* muss entweder eine Aktivität mit einem eindeutigen Namen wie „Löschen“ modelliert werden oder aber eine Aktivität muss durch ein Prädikat anzeigen, dass sie Datenelemente löscht.

Muster wie etwa das *Muster e* zur Reihenfolgeprüfung ergeben immer eine Ja-/

Nein-Entscheidung über die Regeleinhaltung. Bei anderen Mustern hängt die Entscheidung von der Verfügbarkeit gewisser Prädikate bzw. zusätzlicher Informationen über den Workflow ab. Zur besseren praktischen Anwendbarkeit können diese Muster um einen speziellen End-Platz erweitert werden, der anzeigt, dass die Regeleinhaltung wegen fehlender Informationen nicht eindeutig bestimmt werden konnte. Das *Muster b* beispielsweise würde dann die Regeleinhaltung mit Tokens in *OK* anzeigen, Regelverstöße mit Tokens in *ERR* und ungeklärte Fälle mit Tokens in dem Platz *UNRES* („unresolved“). Comcert unterstützt die Anzeige fehlender Annotationen bzw. Prädikate durch die Erzeugung eines speziellen Token-Typs „unresolved“. Diese Tokens werden immer dann erzeugt, wenn eine Regel einen Wert vorschreibt, die Einhaltung im Workflow aber wegen fehlender Informationen nicht geklärt werden kann, z. B. bezüglich des Ausführungsorts einer Aktivität oder der ausführenden Rolle.

Comcert schaltet beim Durchlaufen aller Workflow-Pfade die jeweils entsprechenden RPNs. Dabei erhält jeder Pfad eine eindeutige Kennung, dem Schema binärer Bäume entsprechend, aber an mehr als nur jeweils zwei mögliche Zweige angepasst. Die Kennung 3.4.1 beispielsweise steht für den Pfad über den dritten Zweig der ersten Verzweigung, den vierten Zweig der zweiten Verzweigung und den ersten Zweig der dritten Verzweigung. Nähere Ausführungen fallen aus Platzgründen weg. Mit diesem Schema kann die Analyse beliebig viele Pfade und Verzweigungen erfassen und auch Schleifen im Workflow berücksich-

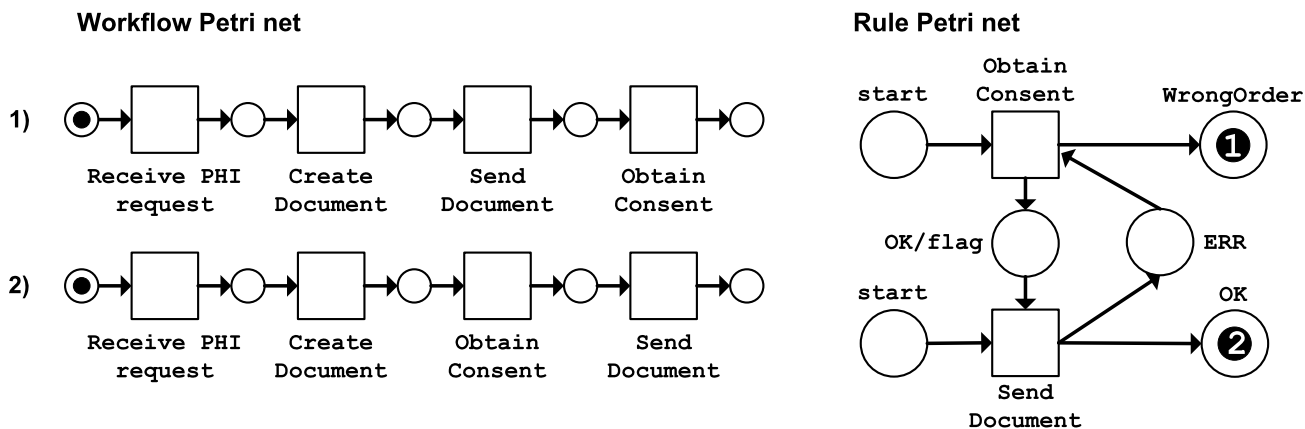
tigen. Die Analysekomplexität wird weiter unten diskutiert.

Wenn das Workflow-Modell und die passenden RPN vorliegen, beginnt Comcert die Analyse dem in **Abb. 4** durch Pseudo-Code beschriebenen Algorithmus entsprechend.

*Beispiel 2* Im Workflow 1 (oben in **Abb. 5**) fordert ein Krankenhaus PHI zu Behandlungszwecken an. Der EHR-Provider will daraufhin ein PHI-Dokument erzeugen und weiterleiten. Das instanziierte *Muster e* für die Reihenfolge von Aktivitäten (siehe RPN rechts in **Abb. 5**) fordert, dass die Aktivität „Obtain Consent“ ausgeführt werden muss *bevor* der EHR-Provider die Aktivität „Send Document“ ausführen darf. Die Transition „Obtain Consent“ steht im Workflow aber *hinter* „Send Document“ im Workflow 1, weswegen Comcert das RPN in den Zustand „WrongOrder“ schaltet und mit dem Token „1“ den Regelverstoß anzeigt. Die Information über die falsche Reihenfolge kann zur Reparatur des Workflows verwendet werden.

Im Gegensatz zu Workflow 1 enthält der Workflow 2 (unten in **Abb. 5**) die Aktivitäten in der geforderten Reihenfolge. „Send Document“ wird erst *nach* „Obtain Consent“ gefunden, so dass ein Token („2“ in **Abb. 5**) in *OK* erzeugt wird. Die so gezeigte Regeleinhaltung kann durch ein entsprechendes Zertifikat ausgewiesen werden.

Andere Verfahren wie z. B. von Meda et al. (2010) erzeugen erst alle sogenannten Traces (Pfade durch den Workflow) und



**Abb. 5** Beispiel-Workflow mit zugehörigem Regel-Petrinetz

analysieren diese Pfade in einem zweiten Schritt. Comcert erledigt diese beiden Schritte gleichzeitig in einem durch die Erzeugung eindeutig gekennzeichneten Tokens. Durch das Fluten des Petri-netzes wird sichergestellt, dass jeder Pfad besucht wird, und eindeutige Kennungen erlauben die Unterscheidung einzelner Pfade.

Genau wie alle anderen Verfahren zur Petri-netz-Analyse steht Comcert vor dem Problem der Zustandsraumexplosion („state explosion problem“). Im schlimmsten Fall steigt die Analysezeit exponentiell zur Anzahl der Verzweigungen im Workflow. Typischerweise enthalten Workflows aber deutlich weniger als die kritische Anzahl einiger hundert Aktivitäten bzw. Verzweigungen, so dass die Analyse in kurzer Zeit durchgeführt werden kann. Um ein Beispiel zu nennen: Der Comcert-Prototyp hat die über 7000 Pfade eines aus über 300 Plätzen und über 300 Transitionen bestehenden Workflow-Modells auf einem 1,2 GHz-Prozessor in weniger als zwei Sekunden analysiert und dabei weniger als 6 MB Arbeitsspeicher beansprucht. Bei Tests hat sich gezeigt, dass verschachtelte Schleifen sowohl die Analysezeit wie auch den Speicherbedarf deutlich erhöhen. In den von Industriepartnern bereitgestellten Workflows traten solche Mehrfachverschachtelungen aber nur sehr selten auf. Kleine Workflows mit etwa 100 Aktivitäten konnten häufig in weniger als 15 ms analysiert werden.

Das obige Beispiel 2 zeigt ein nützliches Nebenprodukt der Analyse mit Comcert. Über die Zertifizierung bezüglich der Regeleinhaltung hinaus werden Schwachstellen im Workflow lokalisiert. Die Analyse des Workflows 1 (oben in **Abb. 5**) weist auf die zwischen Schritt 1 und 2

fehlende Aktivität „Obtain consent“ hin. Mit Mustern wie dem *Muster e* kann Comcert in der falschen Reihenfolge auftretende Aktivitäten erkennen und die genaue Stelle im Workflow lokalisieren. Anhand dieser Informationen können Workflows automatisiert repariert werden. Höhn (2009) beschreibt ein passendes Verfahren zum Process-Rewriting.

## 5 Zusammenfassung

Nachhaltiges Cloud-Computing muss nachweisen können, dass die vielfältigen Compliance-Regeln in den angebotenen Workflows eingehalten werden. Bisher findet die Zertifizierung manuell statt, was einen bremsenden Gegensatz zu der Dynamik und Flexibilität von Paradigmen wie „Software as a Service“ darstellt. Die Geschäftsmodelle hinter modernen Diensten und besonders im Cloud-Computing funktionieren nur, wenn Workflows auf die individuellen Bedürfnisse zugeschnitten werden können. Wenn diese Anpassung nicht zuverlässig im Sinne der Regeleinhaltung geschieht, schränkt das die Nutzbarkeit von Cloud-Diensten ein.

Zwei Beiträge wurden hier zur automatisierten Zertifizierung vorgestellt. Erstens eine Klassifikation von Compliance-Regeln, und zweitens Comcert, ein automatisiertes Verfahren für die Zertifizierung der Compliance von Workflows. Comcert erlaubt Unternehmen, Auditoren und Kunden, einen beispielsweise in BPMN oder BPEL vorliegenden Workflow auf die Einhaltung von Compliance-Regeln zu überprüfen. Bei Verstößen wird die auslösende Aktivität im Workflow exakt lokalisiert. Für

den Petri-netz-basierten Ansatz wurde außerdem gezeigt, wie die klassifizierten Compliance-Regeln in wiederverwendbare Muster transformiert werden können. Anhand solcher Muster kann die Zertifizierung leicht nachvollzogen werden.

Die Analyseergebnisse stellen keinen Beweis dafür dar, dass ein Workflow unter allen Umständen alle Compliance-Regeln zur Laufzeit einhält. Comcert liefert statt solcher Beweise Evidenzen für die Regeleinhaltung bzw. eventuelle Regelverstöße. Im Gesundheitsumfeld beispielsweise kann die Analyse zur Designzeit helfen, mögliche Probleme im Workflow früh zu erkennen und eventuelle Klagen wegen der (HIPAA-) regelwidrigen Verarbeitung von Gesundheitsdaten zu vermeiden. Etwaige Behebungskosten werden dadurch reduziert oder sogar ganz vermieden. Vor dem Hintergrund, dass die Behebung umso teurer ist, je später ein Fehler gefunden wird, ist dies sehr positiv zu bewerten. Auch wenn aus dem Verfahren kein formaler Beweis im engeren Sinne resultiert, wird eine vollständige Analyse in der Art des Model-Checkings durchgeführt. Die Ergebnisse werden in einem Zertifikat zusammengefasst, dass die Regeleinhaltung des untersuchten Workflows bestätigt.

Die Prüfung ist in dem Sinne vollständig, dass für alle definierten Regelmuster jedes Vorkommen auf jedem Workflow-Pfad entdeckt wird. Im allgemeinen Fall ist aber nicht entscheidbar, ob *alle* nötigen Regeln aufgestellt wurden, da es keinen Beweis dafür geben kann, dass zukünftige Angreifer keine neuen Angriffe entdecken können, gegen die zusätzliche Regeln aufgestellt werden müssen. Insgesamt ist das Ziel von Comcert, die Compliance-Prüfung zu auto-

## Zusammenfassung / Abstract

Rafael Accorsi, Lutz Lowis, Yoshinori Sato

### Automatisierte Compliance-Zertifizierung Cloud-basierter Geschäftsprozesse

Die in wachsendem Ausmaß geforderte Compliance von Geschäftsprozessen kann beim Cloud-Computing bisher – wenn überhaupt – nur mit großem, manuellem Aufwand nachgewiesen werden. Ohne entsprechende Zertifizierung werden aber viele Interessenten keine Cloud-basierten Geschäftsprozesse einsetzen. Der Beitrag stellt ein neues Verfahren zur automatisierten Compliance-Prüfung von Prozessen vor, bei dem Prozesse und Regeln als Petrinetze formalisiert werden. Außerdem wird eine Klassifikation von Compliance-Anforderungen aufgestellt, aus der Muster für die Prüfung abgeleitet werden. Anhand der Muster werden Evidenzen bezüglich der Regelinhaltung automatisiert erzeugt.

**Schlüsselwörter:** Geschäftsprozessmodelle, Cloud-Computing, Compliance-Zertifizierung, Audit, Petrinetze

### 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

matisieren und durch Werkzeugunterstützung zu vereinfachen. Unternehmen können mit Comcert die Regeleinhaltung ihrer Workflows prüfen, und zwar sowohl wenn es um die Auslagerung von (Teilen der) Workflows in die Cloud geht wie auch zur Zertifizierung. Die Zertifizierung dient dann als Signal an mögliche Industriepartner oder Kunden, dass Daten nur unter Einhaltung der Compliance-Regeln verarbeitet werden.

Mit Comcert durchgeführte Tests zeigen, dass das Verfahren zuverlässig die möglichen Regelverstöße in Workflows aufdeckt. Die vorgestellten Regelmuster, die den expliziten Informationsfluss erfassen, decken die in der Praxis am häufigsten auftretenden Schwachstellen ab. Verstöße, die aus dem impliziten Informationsfluss resultieren, können bisher nicht mit dem Verfahren aufgedeckt werden. Die daraus möglicherweise entstehenden Informationslecks müssen aber zumindest in Hochsicherheitsszenarien erkannt werden. Für die entsprechende Zertifizierung ist geplant, Comcert um die Entdeckung der problematischen „covert-channels“ (Lampson 1973) zu erweitern (Accorsi und Wonnemann 2011).

## Literatur

- Accorsi R, Wonnemann C (2011) Strong non-leak guarantees for workflow models. *ACM, SACS*, S. 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, S 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 (Hrsg) *Proc JISBD 4*, S 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, S 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/>. Abruf am 2010-06-29

- CSA (2010) Top threats to cloud computing. Cloud Security Alliance. <http://www.cloudsecurityalliance.org/>. Abruf am 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 LNBP* 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, S 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/>. Abruf am 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
- 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
- Lewis 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>. Abruf am 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>. Abruf am 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, S 126–131