David Karlin

IT-GESTÜTZTES COMPLIANCE MANAGEMENT FÜR GESCHÄFTSPROZESSE

KIT Scientific Publishing

David Karlin

IT-gestütztes Compliance Management für Geschäftsprozesse

IT-gestütztes Compliance Management für Geschäftsprozesse

von
David Karlin

Scientific Publishing

Dissertation, Karlsruher Institut für Technologie (KIT)
Fakultät für Wirtschaftswissenschaften, 2015

Eingereicht unter dem Titel: Compliance Management für Geschäftsprozesse
Tag der mündlichen Prüfung: 9. Juni 2015
Referenten: Prof. Dr. Andreas Oberweis, Prof. Dr.-Ing. Eric Sax

Impressum

KIT Scientific Publishing

Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

www.ksp.kit.edu

Print on Demand 2016

ISBN 978-3-7315-0507-5
DOI 10.5445/KSP/1000053663

IT-gestütztes Compliance Management für Geschäftsprozesse

Zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften

(Dr.-Ing.)

von der Fakultät für Wirtschaftswissenschaften
des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

Dipl.-Wi.-Ing. David Karlin

Tag der mündlichen Prüfung:	09.06.2015
Referent:	Prof. Dr. Andreas Oberweis
Korreferent:	Prof. Dr.-Ing. Eric Sax

Vorwort

An dieser Stelle möchte ich mich bei all denen bedanken, die das Gelingen dieser Arbeit ermöglicht und auf unterschiedliche Art und Weise unterstützt haben.

Zuallererst möchte ich meinem Doktorvater Herrn Professor Dr. Andreas Oberweis für seine wertvollen Anregungen und die Betreuung meiner Dissertation danken. Bei Herrn Professor Dr.-Ing. Eric Sax bedanke ich mich für sein konstruktives Feedback und die Übernahme des Korreferats. Darüber hinaus danke ich meinen Prüfern Herrn Professor Dr. Hagen Lindstädt und Herrn Professor Dr.-Ing. habil. Thomas Lützkendorf. Mein Dank gilt außerdem Herrn Professor em. Dr. Dr.h.c. Wolffried Stucky für seine Unterstützung und das Korrekturlesen der Arbeit.

Meinen Kolleginnen und Kollegen des Forschungsbereichs Software Engineering (SE) am FZI Forschungszentrum Informatik in Karlsruhe möchte ich sowohl für den fachlichen Austausch als auch die gute Zusammenarbeit danken: Sascha Alpers, Christoph Becker, Oliver Denninger, Esmahan Eryilmaz, Dr.-Ing. Henning Groenda, Dr.-Ing. Michael Hauck, Dr. Stefan Hellfeld, Jörg Henß, Georg Hinkel, Matthias Huber, Martin Küster, Dr.-Ing. Benjamin Klatt, Dr.-Ing. Klaus Krogmann, Jochen Rill, Stephan Seifermann, Christian Stier, Patrik Scheidecker, Dr. Thomas Schuster und Emre Taspolatoglu. Dr.-Ing. Nicole Groß und Jan Wiesenberger danke ich für die aufmunternden Gespräche insbesondere während der Schreibphase der Dissertation.

Ein besonderer Dank gilt meinen Eltern, meiner Familie und meinen Freunden, die mich bei meinem Promotionsvorhaben uneingeschränkt unterstützt haben.

Karlsruhe, im März 2016

David Karlin

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Zunächst wird anhand einer Beschreibung der Ausgangslage auf relevante Problemstellungen eingegangen und das Thema der vorliegenden Arbeit motiviert. Nachfolgend wird die Zielsetzung der Arbeit beschrieben. Abschließend werden der Aufbau und die Struktur der Arbeit vorgestellt und (grafisch) veranschaulicht.

1.1 Ausgangslage, Motivation und Problemstellung

Bereits in der Mitte der siebziger Jahre stellte die für die Kontrolle des Wertpapierhandels in den Vereinigten Staaten zuständige Börsenaufsichtsbehörde, die United States Securities and Exchange Commission (SEC) fest, dass ein bedeutender Anteil der amerikanischen Unternehmen in Bestechungsskandale in Übersee verwickelt war (Tarantino, 2008). Um das Vertrauen in die amerikanische Wirtschaft und die Glaubwürdigkeit der Unternehmen wiederherzustellen, wurde im Dezember 1977 mit dem Foreign Corrupt Practices Act ein Gesetz verabschiedet, das im historischen Rückblick als eine der ersten Compliance-Richtlinien gilt, die Anforderungen definierte, um das verantwortungsvolle Handeln von Unternehmen zu sichern (Stemper, 2008).

In den darauffolgenden Jahren folgten eine Reihe weiterer Unternehmensskandale – sowohl in den Vereinigten Staaten als auch in Europa. Zu den bekanntesten zählen neben dem Zusammenbruch der Barings Bank, der durch riskante und unerlaubte Währungsspekulationen ausgelöst wurde (1995), die Insolvenz von Enron nach der Aufdeckung von Bilanzfälschungen (2001), dem von Worldcom eingeleiteten Börsenskandal (2002), die bei Tyco aufgedeckten Bilanzmanipulationen (2004) oder die zuletzt im Rahmen der globalen Finanzkrise durch die Société Generale verursachten Milliardenverluste (2008) (Tarantino, 2008). Auch Fälle deutscher Unternehmen gelangten in den letzten Jahren an die Öffentlichkeit, darunter die bei Siemens aufgedeckten Bestechungsskandale (2008) (Stemper, 2008).

In der Folge reagierten die Gesetzgebung sowie die zuständigen Überwachungsorgane mit neuen oder überarbeiteten Gesetzen sowie Richtlinien, die die gesamte Bandbreite an unternehmerischen Aktivitäten abdecken: von der Steuerzahlung und Finanzberichterstattung bis zur Überprüfung der Fertigungsqualität und Überwachung der Materialnutzung (Klotz & Dorn, 2008; Stemper, 2008).

Zu den bedeutendsten und bekanntesten Vorschriften zählen neben dem Health Insurance Portability and Accountability Act (HIPAA), dem Sarbanes-Oxley Act (SOX) oder Basel I-III die EU-Datenschutzrichtlinie oder die Markets in Financial Instruments Directive (MiFID) (Klotz & Dorn, 2008; Stemper, 2008). Auf IT-Seite werden sie in Deutschland u.a. vom Telekommunikationsgesetz (TKG), dem Bundesdatenschutzgesetz (BDSG), den Grundsätzen ordnungsmäßiger DV-gestützter Buchführungssysteme (GoBS) sowie den Grundsätzen zum Datenzugriff und zur Prüfbarkeit digitaler Unterlagen (GDPdU) begleitet und durch Standards wie dem vom Bundesamt für Sicherheit in der Informationstechnik (BSI) herausgegebenen IT-Grundschutz ergänzt (Klotz & Dorn, 2008; Stemper, 2008).

Darüber hinaus gewinnen für Unternehmen internationale Standards, z.B. für das Qualitätsmanagement (ISO 900x), die IT-Sicherheit (ISO 2700x), Rahmenwerke wie das Enterprise Risk Management (ERM)-Framework der Committee of Sponsoring Organizations of the Treadway Commission (COSO) oder das Control Objectives for Information and Related Technology (COBIT), sowie Best Practices wie das Supply Chain Operations Reference-Modell (SCOR) zunehmend an Bedeutung (Klotz & Dorn, 2008; Stemper, 2008; Tarantino, 2008).

Neben den für die Einhaltung bzw. Erfüllung dieser Vorschriften benötigten Aufwendungen bzw. Investitionen sind die Bewältigung der aufgrund der Vielfalt stetig zunehmenden Komplexität, der dadurch bedingten zunehmenden Intransparenz bei den zu beachtenden Gesetzen und Richtlinien sowie der daraus resultierenden Unsicherheit in Bezug auf die Risiken potenzieller Regelverstöße die größten Herausforderungen, denen sich insbesondere international tätige Unternehmen stellen müssen. So existieren in verschiedenen Ländern häufig entsprechend unterschiedliche Bestimmungen, die nicht selten sogar widersprüchlich zu denen anderer Länder oder Regionen sind und die Unternehmen in eine unangenehme Zwangslage bringen können (Klotz & Dorn, 2008; Stemper, 2008).

Neue oder überarbeitete Gesetze und Richtlinien stellen stetig neue oder veränderte Anforderungen an die Unternehmen, die sich kontinuierlich den wechselnden Rahmenbedingungen anpassen und ihre Geschäftsprozesse sowie organisatorische Strukturen und unternehmerische Ziele danach ausrichten müssen.

In einem zunehmend IT-zentrierten Umfeld ist die Umsetzung dieser regulatorischen Anforderungen kritisch für den Unternehmenserfolg geworden. In der Folge müssen Unternehmen eine Reihe verschiedener Maßnahmen ergreifen, um eine verbesserte Kontrolle ihrer Geschäftsprozesse zu ermöglichen. Dabei sollen diese Maßnahmen nicht nur die Einhaltung bzw. Erfüllung von Gesetzen und Richtlinien sicherstellen, sondern

gleichzeitig auch dazu beitragen, die eigenen operationellen Ziele effektiver und effizienter zu erreichen (Türetken, Elgammal, van den Heuvel, & Papazoglou, 2012).

In der Wissenschaft wird das Themengebiet rund um den Begriff Compliance in unterschiedlichen Fachgebieten und Disziplinen, darunter den Rechtswissenschaften, der Betriebswirtschaftslehre und der Informatik (Angewandte Informatik, Wirtschaftsinformatik) aus verschiedenen Perspektiven und Blickwinkeln betrachtet und mit unterschiedlichen Schwerpunkten sowie mit verschiedenen Ansätzen, Methoden und Verfahren erforscht (Klotz & Dorn, 2008).

Seit Beginn dieses Jahrtausends befassen sich wissenschaftliche Arbeiten sowohl mit Governance, Risk Management und Compliance (GRC) bzw. IT-GRC als auch verstärkt mit der Untersuchung von Compliance für Geschäftsprozesse unter dem Schlagwort Business Process Compliance (Management) (Ghose & Koliadis, 2007; Karagiannis, Mylopoulos, & Schwab, 2007; Governatori & Sadiq, 2008; Rinderle-Ma, Ly, & Dadam, 2008). Darüber hinaus finden sich zum Teil verwandte Ansätze in Arbeiten zum Process Mining (van der Aalst, de Beer, & van Dongen, 2005).

Als Grundlage zur Bewältigung bzw. Sicherstellung von Business Process Compliance gelten im Allgemeinen die aus verschiedenen Quellen stammenden regulatorischen Anforderungen und die damit verbundenen Kontrollen. Zur Umsetzung dieser Kontrollen bzw. zur Prüfung der Anforderungen dienen wiederum Regelwerke, die sich aus den verschiedenen Anforderungen ableiten lassen und sowohl manuell als auch automatisierbar zu prüfende Regeln umfassen können.

Des Weiteren werden Methoden und Verfahren unterschieden, die eine Compliance-Prüfung bereits zur Entwurfszeit der Geschäftsprozesse, also vor deren Ausführung, während deren Ausführung, als auch nach deren Ausführung ermöglichen. Hierfür sind in der Regel verschiedene Voraussetzungen zu erfüllen: einerseits müssen zur (standardisierten) Beschreibung der Geschäftsprozesse bestimmte Modellierungssprachen verwendet sowie entsprechende Modellierungskonventionen und -regeln beachtet werden. Andererseits müssen Aufzeichnungen bzw. Nachweise sowohl in Form einer fachlichen Dokumentation als auch in Form von standardisierten, technischen (Ablauf-)Protokollen existieren und vorgehalten werden, die Angaben zu erreichten Ergebnissen ausgeführter Tätigkeiten bereitstellen und auf deren Basis entsprechende interne sowie externe Audits durchgeführt werden können (Sackmann, 2008; Becker, Delfmann, Eggert, & Schwittay, 2012).

Aufgrund der Vielzahl regulatorischer Anforderungen werden Methoden und Werkzeuge benötigt, die neben einer geeigneten Verwaltung von Compliance-Anforderungen

und -Kontrollen, eine (semi-)automatisierte Compliance-Prüfung von Geschäftsprozessen, d.h. von Geschäftsprozessmodellen und/oder von in technischen (Ablauf-) Protokollen aufgezeichneten bzw. gespeicherten Geschäftsprozessinstanzen, ermöglicht. Zur Compliance-Prüfung sollten dabei generische Compliance-Regeln verwendet werden, um eine möglichst allgemeingültige und domänenunabhängige sowie breite Anwendbarkeit gewährleisten zu können. Des Weiteren soll im Anschluss eine zeitnahe Aufbereitung, Darstellung und Analyse der im Rahmen von Audits durchgeführten Compliance-Prüfungen und damit der Auditergebnisse ermöglicht werden. Darüber hinaus bedarf es neben einer geeigneten organisatorischen Einbettung des Compliance Managements für Geschäftsprozesse einer flexiblen und auf offenen Standards basierenden Werkzeugunterstützung, um Unternehmen und ihre Mitarbeiter bei der Erfüllung der unterschiedlichen Anforderungen geeignet zu unterstützen (Sackmann, 2008; Becker, Delfmann, Eggert, & Schwittay, 2012; El Kharbili, 2012; Rodríguez, et al., 2013). Motiviert durch die in diesem Kapitel vorgestellte Ausgangslage sowie die zuvor beschriebenen Herausforderungen und Problemstellungen, soll diese Arbeit einen anwendungsorientierten Beitrag zur informationstechnologischen Unterstützung von Unternehmen und ihren Mitarbeitern beim Compliance Management für Geschäftsprozesse leisten.

1.2 Zielsetzung der Arbeit

Im Folgenden wird die Zielsetzung der vorliegenden Arbeit beschrieben und diese im Anschluss auf Teilziele heruntergebrochen. Das Gesamtziel lässt sich wie folgt formulieren:

Ziel der Arbeit ist es, eine Methode und Werkzeuge für eine informationstechnologische Unterstützung des Compliance Managements für Geschäftsprozesse zu entwickeln.

Um den im vorangegangen Kapitel beschriebenen Problemstellungen begegnen zu können, werden die folgenden Teilzeile definiert, die im Rahmen der Arbeit adressiert werden sollen:

- Teilziel 1: Es soll ein musterbasierter Ansatz zur semiformalen Spezifikation von Compliance-Regeln entwickelt werden. Des Weiteren sollen verschiedene generische Compliance-Regeln identifiziert, vorgestellt sowie eine Einordnung der Regeln anhand von Kategorien vorgenommen werden. Dieses Teilziel wird maßgeblich in Kapitel 5 sowie in Kapitel 7 adressiert.

- Teilziel 2: Es soll ein neuer Ansatz zur ereignisprotokollbasierten Compliance-Prüfung entwickelt werden, der eine Prüfung der in Ereignisprotokollen zu Geschäftsprozessinstanzen festgehaltenen Informationen ermöglicht. Sowohl die Aufzeichnung und Speicherung von Ereignisdaten in Ereignisprotokollen als auch die Ergebnisse der Compliance-Prüfung sollen in einem Format vorliegen, das auf offenen Standards basiert. Dieses Teilziel wird maßgeblich in Kapitel 6 sowie in Kapitel 7 adressiert.
- Teilziel 3: Es soll ein Ansatz zur Aufbereitung, Darstellung und Analyse der Ergebnisse einer im Rahmen von Audits durchgeführten Compliance-Prüfung entwickelt werden. Darüber hinaus soll eine visuelle und interaktive Analyse der Ursachen potentieller Compliance-Verletzungen ermöglicht werden. Dieses Teilziel wird maßgeblich in Kapitel 7 adressiert.

Neben diesen Teilzielen soll im Rahmen dieser Arbeit zudem ein Ansatz konzipiert werden, der eine Verwaltung von in natürlicher Sprache vorliegenden Compliance-Anforderungen erlaubt. Darüber hinaus soll eine organisatorische Einbettung der zu entwickelnden Lösung sowohl in das Compliance Management von Unternehmen als auch in Teilbereiche des Geschäftsprozessmanagements möglich sein. Die zu entwickelnden Methoden und Werkzeuge sollen auf offenen Standards beruhen, flexibel und einfach erweiterbar sein.

1.3 Aufbau und Struktur der Arbeit

Die vorliegende Arbeit umfasst neun Kapitel und gliedert sich wie folgt: In Kapitel 1 wird auf die Ausgangslage, Motivation und Problemstellung sowie die Zielsetzung der Arbeit eingegangen. Aufbauend auf den Grundlagen von Governance, Risk Management und Compliance (Kapitel 2), dem Geschäftsprozessmanagement (Kapitel 3) sowie der Geschäftsprozess-modellierung (Kapitel 4) wird in Kapitel 5 der Begriff der Compliance-Anforderung definiert sowie ein Lebenszyklus für das Compliance Management präsentiert. Nachfolgend wird ein musterbasierter Ansatz zur semiformalen Spezifikation von Compliance-Regeln vorgestellt, es werden verschiedene generische Regeln identifiziert und es wird eine Einordnung anhand verschiedener Kategorien vorgenommen. Es wird ein konzeptuelles Modell entwickelt, das in der Folge als Basis für die Konzeption und Umsetzung einer informationstechnologischen Unterstützung verwendet wird. In Kapitel 6 wird eine Einordnung verschiedener in der Literatur diskutierter Ansätze für die Compliance-Prüfung von Geschäftsprozessen vorgenommen und es werden ausgewählte Arbeiten vorgestellt. Im Anschluss wird auf die Ereignisprotokollierung im Kontext von Geschäftsprozessen eingegangen und ein XML-basierter Standard für Ereig-

nisprotokolle vorgestellt. Nachfolgend wird ein Ansatz zur ereignisprotokollierten Compliance-Prüfung hergeleitet und eine beispielhafte Implementierung der in Kapitel 5 beschriebenen Regeln präsentiert. In Kapitel 7 werden drei prototypische Software-Komponenten konzipiert bzw. umgesetzt, die Funktionalitäten zur Verwaltung von Compliance-Anforderungen, -Kontrollen und -Regeln, zur Durchführung von Compliance-Audits sowie zur Compliance-Prüfung von Ereignisprotokollen zur Verfügung stellen. Darüber hinaus wird eine Aufbereitung, Darstellung und Analyse der Auditergebnisse ermöglicht. In Kapitel 8 wird die entwickelte Lösung anhand eines Anwendungsbeispiels evaluiert. Im Anschluss werden Grenzen aufgezeigt und die Ergebnisse der Evaluation kritisch diskutiert. Die Arbeit schließt mit der Zusammenfassung der Ergebnisse und es wird ein Ausblick auf zukünftige Forschungsarbeiten gegeben (Kapitel 9). Die Struktur der Arbeit ist in Abbildung 1.1 übersichtsartig dargestellt:

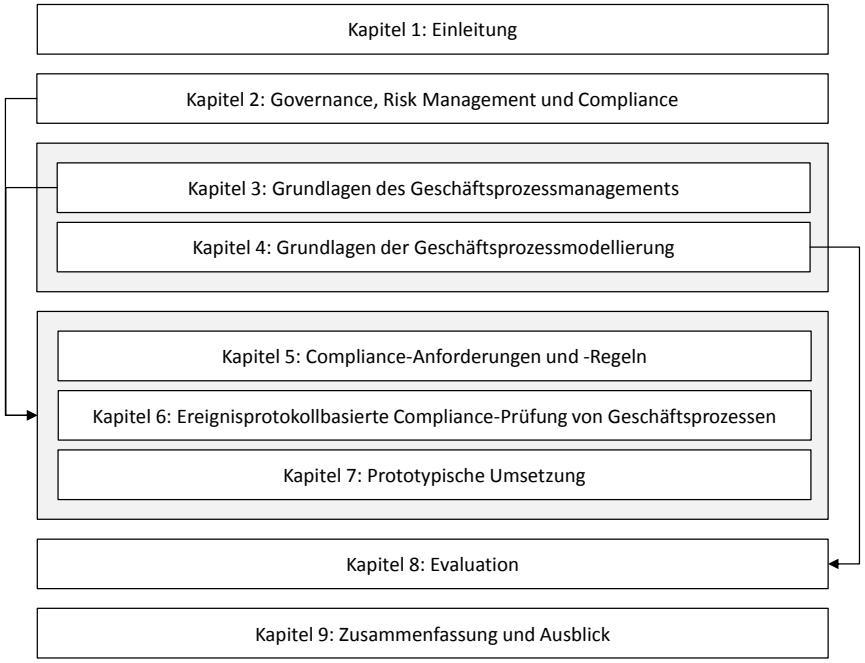

Abbildung 1.1: Struktur der Arbeit

2 Governance, Risk Management und Compliance

In diesem Kapitel wird zunächst ein Überblick über den Stand der Forschung und eine Definition des Begriffes (Corporate) Governance, Risk Management und Compliance gegeben. Im Anschluss werden die einzelnen Disziplinen separat vorgestellt und wesentliche Begriffe definiert. Abschließend wird auf GRC im Kontext von Informationstechnologien eingegangen.

Seitdem PriceWaterhouseCoopers im Jahr 2004 den Begriff Governance, Risk Management und Compliance und damit das Akronym GRC maßgeblich prägte (PwC, 2004), haben die dahinter stehenden Disziplinen für Unternehmen nicht nur kontinuierlich an Bedeutung gewonnen, es wurden auch immer neue Definitionen, u.a. von Beratungs- oder Marktforschungsunternehmen wie AMR Research, Forrester Research (McClean & Rasmussen, 2007; McClean, 2009; McClean, McNabb, & Dill, 2009; McClean, 2011; McClean, 2014) und Gartner (Hagerty, Gaughan, & Verma, 2008; Proctor, Caldwell, & Eid, 2008; Caldwell, Eid, & Casper, 2009; Caldwell, Scholtz, & Hagerty, 2011; Caldwell & Wheeler, 2013), Software-Herstellern sowie anderen Organisationen präsentiert, die sich sowohl im Umfang als auch im Integrationsgrad der Betrachtung zumeist erheblich unterscheiden und häufig auf bestimmte Produkte und Dienstleistungen zugeschnitten waren. Dabei lag bereits bei der Entstehung des Begriffs die Erkenntnis zu Grunde, dass es sich bei GRC nicht um eine vollkommen neue Disziplin handelte, sondern die einzelnen, hinter dem Akronym stehenden Bestandteile bereits seit jeher in einem grundlegenden Interesse von Unternehmen bzw. der Unternehmensführung liegen. Als Neuheit gilt vielmehr der damit aufgekommene Ansatz einer integrierten Betrachtung einer Anzahl von Konzepten, die, ganzheitlich sowie innerhalb eines Unternehmens angewandt, einen signifikanten Mehrwert und damit einen entscheidenden Wettbewerbsvorteil bieten können (Chatterjee & Milam, 2008; Racz, Weippl, & Seufert, 2010a).

Dabei kann neben der horizontalen Integration von GRC (also einer Integration der drei Disziplinen untereinander) die vertikale Integration (also die Integration von GRC mit Geschäftsprozessen von Unternehmen) unterschieden werden (Rosemann & zur Muehlen, 2005).

Traditionell fand die Umsetzung von GRC bzw. GRC-Aktivitäten in Unternehmen nicht selten in voneinander isolierten Nischen statt bzw. war über viele einzelne „Silos“

hinweg verstreut, was zu redundanten Aufwänden und Lösungen sowie höheren Kosten und gestiegenen Risiken führte und entsprechend negative Auswirkungen auf die Transparenz und Entscheidungsfindung hatte (Volonino, Gessner, & Kermis, 2004; Fisher, 2007; Gill & Purushottam, 2008).

Aufgrund der zunehmenden Globalisierung und der damit sowohl in Anzahl, Komplexität als auch Bedeutung gestiegenen regulatorischen Anforderungen, waren Unternehmen gezwungen, unterschiedliche Maßnahmen zu ergreifen, um die Einhaltung von regulatorischen Anforderungen und freiwillig auferlegten Verpflichtungen sicherzustellen (Menzies, 2006).

Eine Vielzahl von Autoren beschäftigt sich seither mit der Untersuchung von GRC sowie zunehmend mit Ansätzen zur Integration der einzelnen Disziplinen. Während sich verschiedene (wissenschaftliche) Publikationen mit der Entwicklung und dem Entwurf konzeptioneller Rahmenwerke, Referenzmodellen und -architekturen befassen (Tapscott, 2006; Mitchell, 2007; Frigo & Anderson, 2009; Marefika & Nissen, 2009; Paulus, 2009; Racz, Weippl, & Seufert, 2010a; Vicente & Mira da Silva, 2011; Nissen & Marefika, 2014), legen andere einen Schwerpunkt auf die Operationalisierung GRC-bezogener Methoden und Konzepte, deren informationstechnologische Umsetzung im betrieblichen Kontext (u.a. mit GRC-Softwarelösungen) und den damit verbundenen Implementierungsprozessen (Gericke, Fill, Karagiannis, & Winter, 2009; Racz, Weippl, & Seufert, 2010b; Wiesche, Schermann, & Krcmar, 2011; Spanaki & Papazafeiropoulou, 2013).

Ausgehend von der Erkenntnis, dass sich bis dato nur wenige Autoren bzw. Publikationen mit der integrierten Betrachtung von GRC und einer wissenschaftlichen Definition auseinandergesetzt haben, präsentieren (Racz, Weippl, & Seufert, 2010a) einen Ansatz, der die Vielzahl der in der Literatur verfügbaren unterschiedlichen Begriffe, Begriffsbildungen, Begriffsbestimmungen, Auslegungen oder Erklärungsversuche in einer umfassenden sowie möglichst allgemeingültigen Definition zusammenfasst bzw. zusammenführt:

Definition 2.1: (Corporate) Governance, Risk (Management) und Compliance (GRC) ist ein integrierter, ganzheitlicher und unternehmensweiter Ansatz, der sicherstellen soll, dass Unternehmen ethisch korrekt und in Übereinstimmung mit ihrer Risikoneigung, internen Richtlinien und externen Regularien durch die Ausrichtung von Strategie, Prozessen, Technologie und Menschen handeln und damit (ihre) Effizienz und Effektivität verbessern (Racz, Weippl, & Seufert, 2010a).

Ergänzend zu dieser Definition verbindet ein Bezugsrahmen (frame of reference) die einzelnen Begriffe bzw. Disziplinen und setzt diese in Beziehung zueinander. Dieser Bezugsrahmen (auch GRC-Dreieck bzw. GRC-Trias) dient zur Veranschaulichung der wesentlichen Elemente bzw. Bestandteile eines integrierten GRC-Verständnisses (Abbildung 2.1) (Klotz & Dorn, 2008; Racz, Weippl, & Seufert, 2010a).

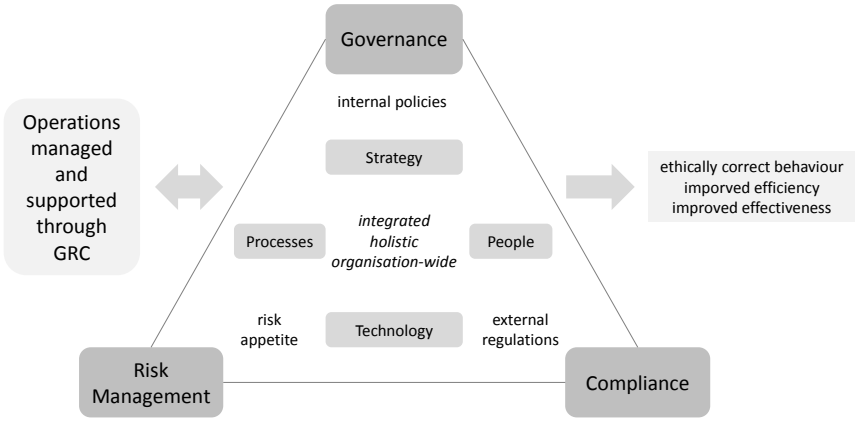

Abbildung 2.1: GRC-Bezugsrahmen

Die einzelnen Disziplinen (Corporate) Governance, Risk Management und Compliance stellen dabei die Kernelemente dar. Jedem Kernelement werden die vier grundlegenden Komponenten Strategie (strategy), Prozesse (processes), Technologie (technology) und Menschen (people) zugeordnet. Die Risikoneigung (risk appetite) des Unternehmens, dessen interne Richtlinien (internal policies) sowie externe Verordnungen bzw. Vorschriften (external regulations) beschreiben die für GRC geltenden Regeln (rules) von GRC. Dabei sind die Kernelemente, Komponenten und Regeln integriert (integrated), ganzheitlich (holistic) und unternehmensweit (organisation-wide) zu betrachten und mit dem Geschäftsbetrieb (business operations) abzustimmen. Ziel des Ansatzes ist es, Unternehmen die Erreichung der GRC-Ziele (objectives) zu ermöglichen: ein ethisch korrektes Verhalten sowie eine verbesserte Effizienz und Effektivität bezogen auf alle im Bezugsrahmen dargestellten Elemente bzw. Komponenten (Racz, Weippl, & Seufert, 2010a). Eines der umfassendsten und in der betrieblichen Praxis am weitesten verbreiteten sowie anerkanntesten Rahmenwerke für integriertes GRC basiert auf den Arbeiten von (Mitchell, 2007) und wird seitdem von der Open Compliance & Ethics Group (OCEG) gepflegt und weiterentwickelt. Die OCEG ist eine nichtgewinnorientierte Organisation, die Unternehmen dabei helfen möchte, eine sogenannte Principled Perfomance zu erreichen:

„OCEG is a nonprofit think tank that helps organizations achieve Principled Performance. We provide standards, resources and a hub around which many professionals can collaborate“ (OCEG, 2014).

Unter Principled Performance versteht die OCEG die Erreichung von (Unternehmens-) Zielen unter Beachtung verpflichtender sowie freiwillig auferlegter Grenzen (Mitchell, 2007; OCEG, 2014).

Das OCEG-Rahmenwerk definiert GRC als ein System von Menschen, Prozessen und Technologien, das ein Unternehmen unter anderem dazu in die Lage versetzt,

- die Erwartungen der Interessensgruppen zu verstehen und zu priorisieren,
- die Geschäftsziele übereinstimmend mit seinen Werten und Risiken aufzustellen,
- die Ziele zu erreichen, dabei das Risikoprofil zu optimieren und Werte zu schützen,
- innerhalb rechtlicher, vertraglicher, interner, sozialer und ethischer Grenzen zu arbeiten,
- den zuständigen Interessensgruppen relevante, zuverlässige und zeitnahe Informationen zukommen zu lassen und
- die Performanz und Effektivität des Systems messbar zu machen (OCEG, 2014).

Wesentlicher Bestandteil des Rahmenwerks – das auch als Red Book bezeichnet wird – ist das GRC Capability Model. Es beschreibt zentrale Komponenten, Elemente und Praktiken zur Umsetzung und zum Management von GRC-Aktivitäten. Eine GRC-Aktivität fasst dabei sämtliche Prozesse und Aktivitäten die zu dem System beitragen oder Teil des Systems sind, zusammen (Abbildung 2.2) (OCEG, 2009).

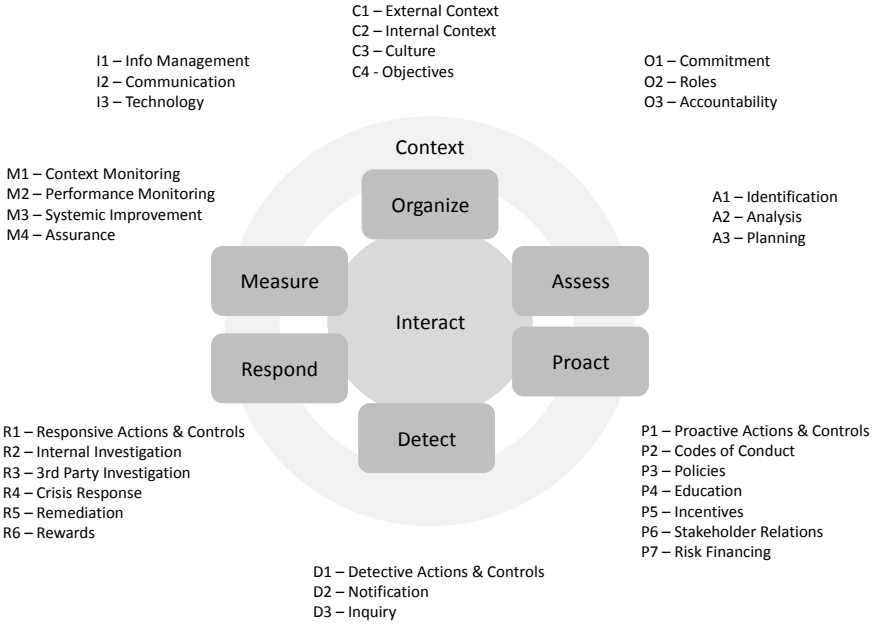

Abbildung 2.2: Komponenten und Elemente des GRC Capability Model

Das GRC Capability Model besteht derzeit (Version 2.1) aus acht Komponenten (Context, Organize, Assess, Proact, Detect, Respond, Measure, Interact), die abhängig vom bereits vorhandenen Reifegrad sowohl sequentiell als auch in beliebiger Abfolge, grundsätzlich jedoch kontinuierlich, angewendet werden sollten (OCEG, 2012).

Jede Komponente umfasst dabei eine Anzahl integrierter Elemente (insgesamt 33), die zur Unterstützung allgemeingültiger und unternehmensspezifischer Ziele neben grundlegenden Prinzipien und kritischen Erfolgsfaktoren verschiedene Praktiken (insgesamt 136) sowie Unter-Praktiken (insgesamt 699) beschreiben, um zu einem übergeordneten Erfolg der GRC-Aktivitäten beizutragen (OCEG, 2012).

Darüber hinaus werden zahlreiche proaktive, detektive sowie responsive Maßnahmen und Kontrollen aufgeführt, die Unternehmen bei der Umsetzung ihrer GRC-Aktivitäten beachten bzw. adressieren und in einem geeigneten „Mix“ umsetzen sollten, um eine Principled Performance zu erreichen (OCEG, 2012).

Neben dem GRC Capability Model bietet die OCEG zusätzliche Materialen an, die weitere Verfahren zur Bewertung des Reifegrads enthalten (GRC Assessment Tools – auch Burgundy Book) sowie dazu beitragen sollen, technologische Lösungen zu identi-

fizieren, die zur Unterstützung der einzelnen Elemente des GRC Capability Models geeignet sind und zum Einsatz kommen könnten (GRC Technology Solutions Guide) (OCEG, 2012).

2.1 Corporate Governance

Die grundlegenden Konzepte, die hinter dem Begriff Corporate Governance (engl. für Unternehmensführung) stehen, lassen sich bis in das 16. Jahrhundert zurückverfolgen. Insbesondere in den letzten 20 Jahren erlangten die Konzepte rund um den Begriff Corporate Governance aufgrund eines Rufs nach stärkerer Regulierung eine zunehmende Aufmerksamkeit (Tarantino, 2008). Ausgelöst wurde dieser Ruf durch eine Reihe verschiedenster Ereignisse, darunter die Unternehmensskandale von Enron und WorldCom im Jahr 2001 sowie anderer, darauf folgender Unternehmensskandale, die letztendlich zu der Verabschiedung des Sarbanes-Oxley Acts (SOX) führten (Tarantino, 2008).

Definition 2.2: Corporate Governance beschreibt „die Prozesse, Systeme und Kontrollen, mit dem Unternehmen die Interessen ihrer Stakeholder verteidigen. Dies kann sowohl Mitglieder des Aufsichtsrats, Vorstände, Führungskräfte, Mitarbeiter, Aktionäre, Lieferanten, Kunden als auch das Umfeld, in dem sich das Unternehmen befindet, umfassen“ (Tarantino, 2008).

Die OCEG definiert Corporate Governance als „die Kultur, Werte, Mission, Struktur und Ebenen von Grundsätzen, Prozessen und Maßnahmen, anhand derer ein Unternehmen geführt und kontrolliert wird […]“ (OCEG, 2009).

Der durch eine Regierungskommission im Jahr 2002 verabschiedete Deutsche Corporate Governance Kodex (DCGK) versteht unter Corporate Governance das „Führungssystem zur Leitung und Überwachung deutscher börsennotierter Gesellschaften“ (DCGK, 2013).

Nach dem im Jahr 2002 vom Verband der Schweizer Unternehmen genehmigten Swiss Code of Best Practice for Corporate Governance (Swiss Code) ist Corporate Goverance „die Gesamtheit der auf das Aktionärsinteresse ausgerichteten Grundsätze, die unter Wahrung von Entscheidungsfähigkeit und Effizienz auf der obersten Unternehmensebene Transparenz und ein ausgewogenes Verhältnis von Führung und Kontrolle anstreben“ (economiesuisse, 2007).

Unabhängig von nationaler Rechtsprechung, Rechtssystemen und Zuständigkeiten sowie anderer lokaler Gegebenheiten lassen sich einige allgemeingültige Prinzipien

bzw. Praktiken von Corporate Governance beschreiben, die sich über die vergangenen Jahre durchgesetzt haben und entsprechend anerkannt sind:

- Respektierung der Aktionärsrechte,
- Definition von Rollen und Verantwortlichkeiten des Aufsichtsrats bzw. Vorstands,
- eine Kultur ethisch korrekten und professionellen Verhaltens,
- finanzielle Transparenz und Vertraulichkeit und
- Einführung sowie Etablierung interner Kontrollen (Tarantino, 2008).

Eines der am kontroversesten diskutierten Gesetze der letzten Jahre ist der im Jahr 2002 in Folge der Unternehmenskandele durch den Kongress der Vereinigten Staaten verabschiedete Sarbanes-Oxley Act. Im Jahr 2003 erfolgte die Umsetzung des Gesetzes durch die Securites and Exchange Commision (SEC) in Form der Section 404. Im darauffolgenden Jahr 2004 wurde unter der Section 404 der Auditing Standard Number 2 for Internal Controls vom Public Company Accounting Oversight Board (PCAOB) veröffentlicht, der im Jahr 2007 durch den Auditing Standard Number 5 ersetzt wurde (PCAOB, 2007). Hieraus resultieren u.a. die Verpflichtungen zur Einführung von internen Kontrollen bzw. eines internen Kontrollsystems (PCAOB, 2007; Tarantino, 2008; Racz, Weippl, & Seufert, 2010b). Diese Regelung betrifft nicht nur US-amerikanische Unternehmen, sondern alle Aktiengesellschaften, deren Wertpapiere in den USA gehandelt werden und die somit bei der SEC registriert sind (Fiege, 2006).

In Deutschland hatte bereits das 1998 in Kraft getretene Gesetz zur Kontrolle und Transparenz im Unternehmensbereich (KontTRaG) u.a. die Regelung des § 91 Abs. 2 AktG zur Folge, der den Vorstand verpflichtet „geeignete Maßnahmen zu treffen, insbesondere ein Überwachungssystem einzurichten, damit die den Fortbestand der Gesellschaft gefährdenden Entwicklungen früh erkannt werden.“ Ähnliche Pflichten zur Gestaltung interner Kontroll- und Risikomanagementsysteme ergeben sich aus dem im Jahr 2009 verabschiedeten Gesetz zur Modernisierung des Bilanzrechts (BilMoG) (Klotz & Dorn, 2008).

Seit jeher ist der Begriff der internen Kontrolle in der Finanzberichterstattung weit verbreitet. Mit der Veröffentlichung des von dem Committee of Sponsoring Organizations of the Treadway Commission (COSO) im Jahr 1992/1994 entwickelten und seitdem gepflegten, gleichnamigen Rahmenwerk existiert heute ein de-facto-Standard, der zur Umsetzung eines internen Kontrollsystems herangezogen werden kann und der von der SEC neben anderen Rahmenwerken als Referenz anerkannt ist (Protiviti, 2007; Tarantino, 2008; McNally, 2013).

Definition 2.3: Eine interne Kontrolle (internal control) ist ein vom Aufsichtsrat bzw. Vorstand eines Unternehmens, dessen Management oder anderen Personen festgelegter Prozess, der sicherstellen soll, dass die auf die Geschäftstätigkeit (operations), die Berichterstattung (reporting) und die Einhaltung von Gesetzen und Richtlinien (compliance) bezogenen Ziele erreicht werden (COSO, 1992; COSO, 2013).

Dabei verfolgt das Rahmenwerk drei maßgebliche Ziele (objectives):

- die effiziente und effektive Geschäftstätigkeit, insbesondere bezogen auf die operationellen und finanziellen Leistungsziele sowie die Absicherung des Anlagevermögens gegen Verluste,
- die Zuverlässigkeit, Aktualität, Pünktlichkeit und Transparenz der (Finanz-) Berichterstattung, wie sie von Gesetzgebern, anerkannten Standardisierungseinrichtungen oder Unternehmensrichtlinien vorgegeben werden und
- die Einhaltung von Gesetzten und Richtlinien.

Das Rahmenwerk umfasst fünf integrierte Komponenten (components): Control Environment, Risk Assessment, Control Activities, Information and Communication, und Monitoring Activities, die wiederum insgesamt 17 Prinzipien zugeordnet sind. Diese beschreiben fundamentale Konzepte, um eine effektive interne Kontrolle sicherzustellen (COSO, 2013).

Die Beziehungen zwischen den Zielen und Komponenten des COSO-Rahmenwerks sind in Abbildung 2.3 in Form eines Würfels veranschaulicht. Die Organisationsstruktur eines Unternehmens ist in der dritten Dimension dargestellt (COSO, 2013):

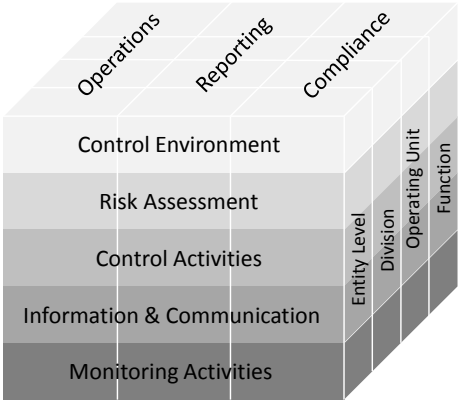

Abbildung 2.3: COSO-Würfel

Neben COSO existieren weitere Rahmenwerke, darunter das vom Canadian Institute of Chartered Accountants (CICA) im Jahr 1995 veröffentlichte Guidance on Control by the Criteria of Control Commitee (CoCo), das auf das COSO-Rahmenwerk aufbaut, der Standard ISO/IEC 38500:2008 – Corporate governance of information technology sowie das von der Information Systems Audit and Control Association (ISACA) veröffentlichte Control Objective for Information and Related Technology (COBIT), das zusammen mit der IT Infrastructure Library (ITIL) als der derzeitige de-facto-Standard in der IT-Industrie anerkannt ist (Racz, Weippl, & Seufert, 2010b).

2.2 Risk Management

Für den Begriff des Risikos existiert seit jeher eine Vielzahl von Definitionen (OCEG, 2009). Der ISO Guide 73 definierte Risiko zunächst als „die Möglichkeit oder Wahrscheinlichkeit von Verlust." Mit der Veröffentlichung der Normenreihe ISO 31000 im Jahr 2009 erfuhr die Definition von Risiko eine Überarbeitung, die nun sowohl positive als auch negative Auswirkungen in die Betrachtung mit einbezieht. So definiert das überarbeitete und harmonisierte Vokabular des ISO Guide 73:2009 den Begriff Risiko wie folgt:

Definition 2.4: Ein Risiko ist „die Auswirkung von Unsicherheit auf das Erreichen von Zielen."

Eine ähnliche, daran angelehnte Definition für Risiko liefert die OCEG: „Risiko ist das Maß für die Wahrscheinlichkeit des Eintretens eines Ereignisses, das eine Auswirkung auf das Erreichen von Zielen hat" (OCEG, 2009).

Abhängig von landes- oder industriespezifischen Einflüssen bzw. Faktoren sowie dem Globalisierungsgrad eines Unternehmens können verschiedene Risikokategorien unterschieden werden. Das Gesamtrisiko setzt sich dabei aus dem Marktrisiko, dem Kreditrisiko, dem Liquiditätsrisiko, dem Rechtsrisiko sowie dem operationellen Risiko zusammen. Darüber hinaus können interne und externe Risiken unterschieden werden (Tarantino, 2008).

Der Begriff des operationellen Risikos hat insbesondere im Bankwesen aufgrund der in den Jahren 2004 bis 2006 veröffentlichten Eigenkapitalvorschriften Basel II: Intenational Convergence of Capital Measurement and Capital Standards an Bedeutung gewonnen. Basel II löste im Jahr 2007 die bis dahin gültige Vereinbarung Basel I: International convergence of capital measurement and capital standards aus dem Jahr 1988 ab. Hierin definiert der Basler Ausschuss für Bankenaufsicht das operationelle

Risiko als „die Gefahr von Verlusten, die infolge der Unangemessenheit oder des Versagens von internen Verfahren, Menschen und Systemen oder in Folge von externen Ereignissen eintreten. Diese Definition schließt Rechtsrisiken ein, beinhaltet aber keine strategischen Risiken oder Reputationsrisiken“ (Basel Committee on Banking Supervision, 2006). Basel II unterscheidet sieben maßgebende Kategorien für operationelle Risiken:

- Internal fraud – Verluste, die durch eine betrügerische Absicht, Veruntreuung von Eigentum oder die Umgehung von Gesetzen oder Unternehmensrichtlinien durch die Beteiligung mindestens einer internen Partei realisiert werden,
- External fraud – Verluste, die durch eine betrügerische Absicht, Veruntreuung von Eigentum oder die Umgehung von Gesetzen oder Unternehmensrichtlinien durch die Beteiligung einer externen Partei realisiert werden,
- Employment practices and Workplace Safety – Verluste, die durch Inkonsistenzen mit Beschäftigungs-, Gesundheits- oder Sicherheitsvorschriften, der Begleichung von Personenschäden oder durch Diversität oder Diskriminierung hervorgerufen werden,
- Clients, Products, and Business Practices – Verluste, die durch einen unbeabsichtigten oder fahrlässigen Leistungsverzug oder einer anderen beruflichen Verpflichtung gegenüber bestimmten Kunden hervorgerufen werden (einschließlich treuhänderischer oder Eignungsanforderungen) oder in der Natur bzw. dem Design eines Produktes begründet liegen,
- Damage to physical assets – Verluste, die aus dem Verlust oder der Beschädigung von Anlagegütern stammen, die durch Naturkatastrophen oder andere Ereignisse hervorgerufen werden,
- Business disruptions and system failures – Verluste, die von der Unterbrechung der Geschäftstätigkeit oder Systemausfällen hervorgerufen werden,
- Execution, Delivery and Process Management – Verluste aus fehlgeschlagenen Transaktionen, Prozessen oder Geschäftsbeziehungen zu Lieferanten (Tarantino, 2008).

Mit Hilfe des Risikomanagements sollen die Risiken eines Unternehmens identifiziert, eingeschätzt, gemessen und bewertet werden, um im Anschluss adäquate Gegenmaßnahmen entwickeln bzw. ergreifen zu können. Da Risiken in der Regel nicht vollständig vermieden werden können, gilt es, die möglichen Auswirkungen so gut wie möglich abzuschwächen bzw. zu minimieren (Tarantino, 2008).

Eine Überprüfung der Effektivität risikobezogener interner Kontrollen kann u.a. die folgenden Elemente umfassen:

- Identifizierung von Geschäftsprozessen, insbesondere solcher, die das Finanzberichtswesen betreffen,
- Identifizierung assoziierter Risiken,
- Identifizierung von internen Kontrollen,
- Hierarchisierung von Geschäftsprozessen, Risiken und Kontrollen,
- Identifizierung von Tests bzw. Testfällen, die zur Bestimmung der Effektivität der internen Kontrollen herangezogen werden können,
- Überprüfung bzw. Test der internen Kontrollen sowie Veröffentlichung der (Test-)Ergebnisse,
- Bereitstellung von Einschätzungen hinsichtlich der Effektivität der Kontrollen,
- Empfehlungen bezüglich der Anpassung bzw. Überarbeitung ineffektiver Kontrollen sowie erneute Durchführung von Tests,
- Erstellung und Pflege einer Dokumentation der Prozesse, Risiken, Kontrollen, Tests, Ergebnisse sowie Nachbesserungen,
- Durchlaufen eines Zertifizierungsprozesses und Zertifizierung effektiver interner Kontrollen, u.a. durch externe Auditoren (Tarantino, 2008).

Aufbauend auf der überarbeiteten Definition von Risiko beschreibt der ISO Guide 73:2009 den Begriff Risikomanagement wie folgt:

Definition 2.5: Das Risikomanagement umfasst „koordinierte Aktivitäten zur Lenkung und Kontrolle einer Unternehmens hinsichtlich dessen Risiken."

Die OCEG definiert Risikomanagement als „die systematische Anwendung von Prozessen und Strukturen, die ein Unternehmen dazu in die Lage versetzen, Risiken zu identifizieren, evaluieren, analysieren, optimieren, überwachen, verbessern oder zu übertragen und diese Risiken sowie entsprechende Entscheidungen an die Stakeholder zu kommunizieren."

Aufgrund der durch neue Gesetzte, Richtlinien und Standards kontinuierlich gestiegenen Anforderungen nahm auch der Bedarf für ein Rahmenwerk, das ein unternehmensweites Risikomanagement ermöglicht und neben grundlegenden Prinzipien und Konzepten ein gemeinsames Vokabular sowie klare Anweisungen und Hilfestellungen zur Verfügung stellt, zu (COSO, 2004). Als Antwort auf die gestiegenen Anforderungen veröffentlichte die COSO im Jahr 2004 eine Erweiterung des gleichnamigen Rahmenwerks Internal Control – Integrated Framework aus dem Jahr 1992 unter dem Namen Enterprise Risk Management – Integrated Framework (COSO ERM).

In diesem Zusammenhang definiert das Rahmenwerk den Begriff des Enterprise Risk Management als „einen durch den Aufsichtsrat bzw. Vorstand eines Unternehmens, dessen Management sowie anderer Personen beeinflussten und in einem unternehmensweiten Rahmen angewendeten Prozess, der dazu beiträgt, potentielle Ereignisse, die auf das Unternehmen Auswirkungen haben können, zu identifizieren, das Risiko in Abstimmung mit der Risikoneigung des Unternehmens zu steuern und eine angemessene Absicherung hinsichtlich des Erreichens der Unternehmensziele zur Verfügung zu stellen“ (COSO, 2004).

Das COSO ERM-Rahmenwerk besteht aus acht zusammenhängenden und in gegenseitiger Beziehung zueinander stehenden Komponenten: Internal Environment, Objective Setting, Event Identification, Risk Assessment, Risk Response, Control Activities, Information and Communication und Monitoring. Es soll dazu beitragen, die Unternehmensziele zur erreichen, die in die folgenden vier Kategorien eingeteilt werden (COSO, 2004):

- Strategic: Ziele auf der obersten Ebene, die mit der Mission des Unternehmens abgestimmt sind und diese unterstützen
- Operations: die effektive und effiziente Nutzung der Unternehmensressourcen
- Reporting: die Zuverlässigkeit des Berichtswesens
- Compliance: die Einhaltung von Gesetzen und Richtlinien

Dabei versteht das COSO ERM-Rahmenwerk Enterprise Risk Management (ERM) als einen multidirektionalen sowie iterativen Prozess, in dem sich alle Komponenten gegenseitig beeinflussen, sowohl zu festen Zeitpunkten, ereignisgesteuert als auch kontinuierlich zur Anwendung kommen sowie angepasst und überwacht werden (COSO, 2004; Racz, Weippl, & Seufert, 2010b).

In Anlehnung an das COSO-Rahmenwerk werden die Beziehungen zwischen den Zielen und den Komponenten des COSO ERM-Rahmenwerks ebenfalls in Form eines Würfels veranschaulicht (Abbildung 2.4) (COSO, 2004):

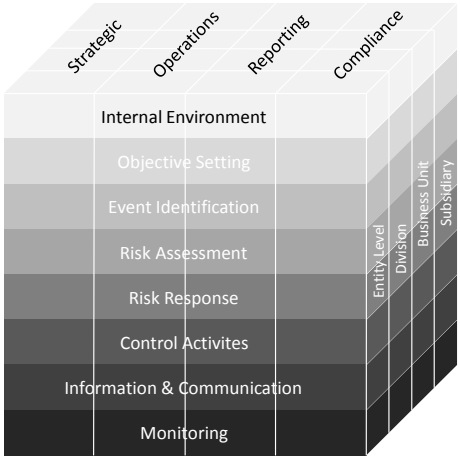

Abbildung 2.4: COSO ERM-Würfel

Neben dem COSO ERM und der Normenreihe ISO 31000:2009 (die der Nachfolger des Australischen Standards AS/NZS 4360:2004 – Risk management, ist) existieren weitere, IT-spezifische Rahmenwerke, wie das von der Information Systems Audit and Control Association (ISACA) veröffentlichte [The] Risk IT Framework sowie der Standard ISO/IEC 27005:2008 – Information security risk management, der Bestandteil der Normenreihe ISO/IEC 27000 ist.

Weitere IT-bezogene Veröffentlichungen umfassen die NIST Special Publication 800-30 – Guide for Conducting Risk Assessments des National Institute of Standards and Technology (NIST) sowie der IT-Grundschutz des Bundesamts für Sicherheit in der Informationstechnik (BSI).

2.3 Compliance

Unter dem Begriff (Corporate) Compliance wird im Allgemeinen die Einhaltung von Regeln (auch Regeltreue oder Regelkonformität) in Unternehmen verstanden (IDW, 2010).

Definition 2.6: Compliance ist „das Handeln in Übereinstimmung mit bestehenden Gesetzen, Verordnungen, Protokollen, Standards und Spezifikationen“ (Tarantino, 2008).

Dies umfasst einerseits von einem (externen) Gesetzgeber erlassene Gesetze, von Aufsichts- oder Regulierungsbehörden erlassene Verordnungen sowie verschiedene (unternehmens-)interne Regelungen und Verfahrensanweisungen (Tarantino, 2008).

Die OCEG definiert Compliance als „den Akt sowie die Befähigung zur Demonstration der Einhaltung verpflichtender Anforderungen, die durch Gesetze und Vorschriften beschrieben werden, sowie freiwilliger Anforderungen, die sich aus vertraglichen Verpflichtungen sowie internen Richtlinien ergeben“ (OCEG, 2009).

Der Australian Standard – Compliance programs (AS 3806-2006) versteht unter Compliance die „Einhaltung der Anforderungen von Gesetzen, industriespezifischer und unternehmerischer Standards und Kodizes, die Prinzipien guter Unternehmensführung sowie akzeptierter gesellschaftlicher und ethischer Grundsätze“ (Standards Australia, 2006).

Nach dem DCGK ist Compliance „[…] die Einhaltung der gesetzlichen Bestimmungen und der unternehmensinternen Richtlinien […]“, für die der Vorstand im Rahmen seiner Aufgaben und Zuständigkeiten zu sorgen und auf deren Beachtung durch die Konzernunternehmen er hinzuwirken hat (DCGK, 2013).

Als Querschnittsthema betrifft Compliance alle Bereiche und Funktionen eines Unternehmens und sollte daher mit den übergeordneten strategischen Zielen abgestimmt werden (Standards Australia, 2006; TÜV Rheinland, 2011). Darüber hinaus steht Compliance für die Forderung, potenzielle Regelverstöße als Risiken zu betrachten, die in Abstimmung mit der Corporate Governance, den internen Kontrollen sowie dem Risikomanagement eines Unternehmens mittels geeigneter organisatorischer, technischer und/oder personeller Maßnahmen zu behandeln sind (Klotz & Dorn, 2008; Tarantino, 2008; Vicente & Mira da Silva, 2011). Diese Maßnahmen werden in der Regel durch ein entsprechend geeignetes eigenständiges Managementsystem unterstützt (TÜV Rheinland, 2011).

Definition 2.7: Das Compliance Management bzw. ein Compliance Managementsystem (CMS) umfasst „die auf der Grundlage der von den gesetzlichen Vertretern festgelegten Ziele eingeführten Grundsätze und Maßnahmen eines Unternehmens, die auf die Sicherstellung eines regelkonformen Verhaltens der gesetzlichen Vertreter und der Mitarbeiter des Unternehmens sowie ggf. von Dritten abzielen, d.h. auf die Einhaltung bestimmter Regeln bzw. die Verhinderung von wesentlichen Verstößen (Regelverstöße)“ (IDW, 2010).

Ein CMS kann sich dabei entweder auf alle oder nur einzelne Geschäftsbereiche, operative Prozesse oder bestimmte Rechtsgebiete (abgegrenzte Teilbereiche) beziehen (IDW, 2010). Ziel eines CMS ist es, systematisch die Voraussetzungen in der Organisation dafür zu schaffen, dass Verstöße gegen Compliance-Anforderungen vermieden bzw. wesentlich erschwert und eingetretene Verstöße erkannt und behandelt werden können (TÜV Rheinland, 2011). Neben der Förderung einer günstigen Compliance-Kultur sowie der Festlegung von Compliance-Zielen sollte ein CMS darüber hinaus den Aufbau der Compliance-Organisation (Aufbau- und Ablauforganisation), Prozesse zur Feststellung und Analyse der Compliance-Risiken durch das Unternehmen, Prozesse zur Erstellung des Compliance-Programms, die Entwicklung von Kommunikationsprozessen sowie Verfahren zur Überwachung und Verbesserung umfassen (IDW, 2010).

Da Compliance-Anforderungen in der Regel nicht statisch sind und häufigen Änderungen unterliegen können, ist für die Realisierung und ständige Verbesserung eines CMS ein iterativer Prozess erforderlich (Abbildung 2.5) (TÜV Rheinland, 2011).

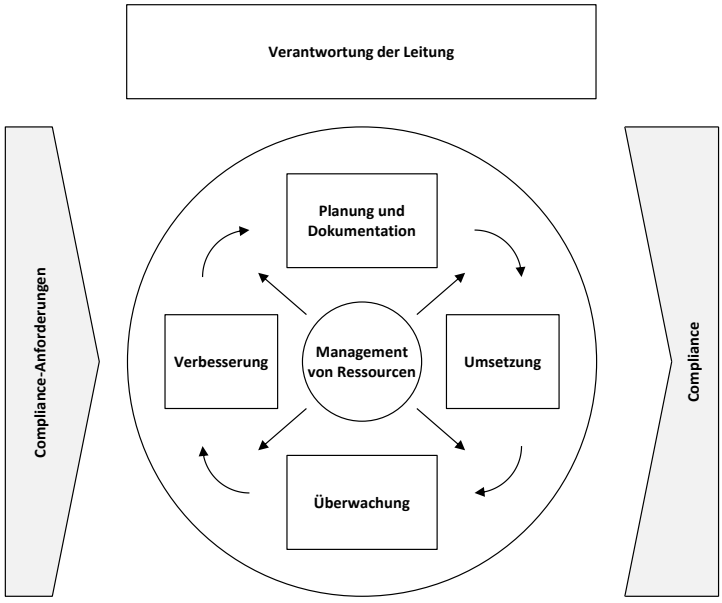

Abbildung 2.5: Modell eines prozessorientierten Compliance Management Systems

In diesem Zusammenhang hat insbesondere in den letzten Jahren die Auditierung von (Management-)Systemen mit Bezug zu GRC zunehmend an Bedeutung gewonnen. Dies betrifft sowohl die Überprüfung interner Kontrollen (z.B. im Rahmen der internen

Revision bzw. interner Audits) als auch die Überprüfung durch externe Auditoren, einschließlich der Überprüfung (und ggf. Zertifizierung) des zugrundeliegenden internen Kontrollsystems bzw. CMS.

2.4 GRC und IT

Mit der stetig wachsenden Anzahl von für Unternehmen gültigen Gesetzen, Verordnungen und Richtlinien nimmt auch die Anzahl der die IT betreffenden Regeln zu. Gleichzeitig ergibt sich aus der steigenden Anzahl der zu beachtenden Regeln eine besondere Herausforderung, diese zu steuern, zu kontrollieren und zu überwachen und somit die Regelkonformität – der IT sowie des Unternehmens – sicherzustellen. Dabei stehen insbesondere solche Gesetze, Verordnungen und Richtlinien im Fokus der Betrachtungen, die Auswirkungen auf die IT bzw. auf die durch IT unterstützten Geschäftsprozesse eines Unternehmens haben (Böhm, 2008).

Im Kontext von GRC und IT können hierbei zwei Perspektiven unterschieden werden: IT-GRC (auch GRC für IT) und IT für GRC (Racz, Weippl, & Seufert, 2010b).

IT-GRC betrachtet eine Untermenge von GRC, die alle GRC-Aktivitäten umfasst, die sich auf die IT-Landschaft bzw. den IT-Betrieb eines Unternehmens beziehen sowie die damit verbundenen Geschäftsprozesse betreffen (Abbildung 2.6) (Racz, Weippl, & Seufert, 2010b).

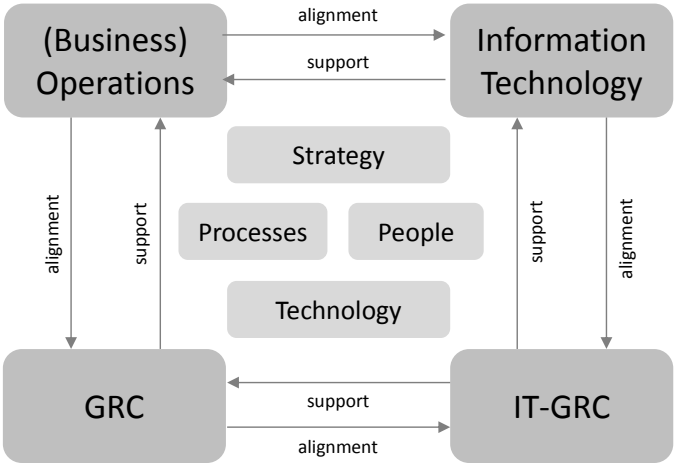

Abbildung 2.6: GRC, IT-GRC und IT

Analog zu GRC sollte IT-GRC mit den übergeordneten GRC-Aktivitäten des Unternehmens, mit dessen IT sowie indirekt mit dessen Geschäftstätigkeit unter Einbeziehung der Strategie des Unternehmens, den Prozessen, den Menschen und der Technologie abgestimmt werden (Racz, Weippl, & Seufert, 2010b).

Hierzu umfasst IT-GRC die jeweils IT-spezifischen Ausprägungen der einzelnen GRC-Disziplinen: die IT-Governance, das IT-Risikomanagement und die IT-Compliance (Abbildung 2.7) (Racz, Weippl, & Seufert, 2010b). Ziel von IT-GRC ist es, Unternehmen auf Basis eines geeigneten IT-Governance-Rahmenwerks einerseits Informationen über IT-Risiken sowie IT-Compliance-Anforderungen bereitzustellen, andererseits die effektive und effiziente Ausführung der Geschäftsprozeese unter Berücksichtigung von IT-Risiken und IT-Compliance-Anforderungen geeignet zu unterstützen (Racz, Weippl, & Seufert, 2010b).

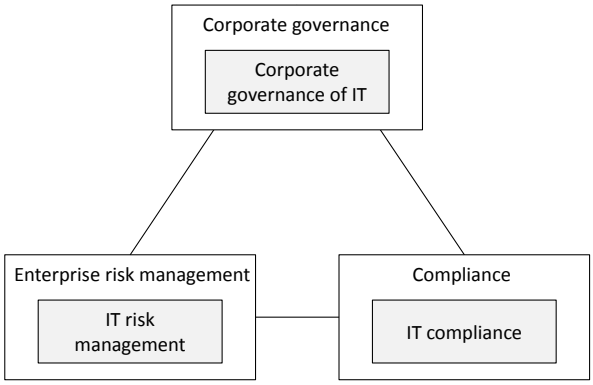

Abbildung 2.7: IT-GRC als Untermenge von GRC

Für die Disziplin IT-Governance (auch ITG oder corporate governace of IT) finden sich in der Literatur zwei unterschiedliche Denkschulen bzw. Ansätze, die diese aus unterschiedlichen Perspektiven betrachten: Eine Sichtweise leitet sich aus der Corporate Goverance ab und versteht die IT-Goverance maßgeblich als Instrument zur Unterstützung der sich daraus ergebenden Anforderungen. In diesem Kontext legt die IT-Governance die Rahmenbedingungen für die IT fest. Eine andere Sichtweise beschäftigt sich mit der möglichst wirtschaftlichen Gestaltung von IT sowie den damit verbundenen organisatorischen Strukturen und Prozessen. Diese werden häufig auch unter dem Begriff IT-Management zusammengefasst (Weill & Ross, 2004; De Haes & van Grembergen, 2005; Knolmayer & Loosli, 2006; Lewis & Millar, 2010).

Der Standard ISO/IEC 38500:2008 – Corporate governance of IT definiert IT-Governance als „das System, mit dem sowohl der gegenwärtige als auch zukünftige IT-Einsatz gesteuert und kontrolliert wird. Es umfasst die Bewertung, Steuerung und Überwachung der für den IT-Einsatz notwendigen Pläne zur Unterstützung des Unternehmens. Des Weiteren beinhaltet es die Strategie und Regeln für den IT-Einsatz in einem Unternehmen“ (Abbildung 2.8) (ISO, 2008). Diese Definition schließt ausdrücklich die Identifizierung von Prozessen für die Steuerung und Kontrolle der Nutzung von IT auf der Unternehmensleitungsebene ein (Lewis & Millar, 2010).

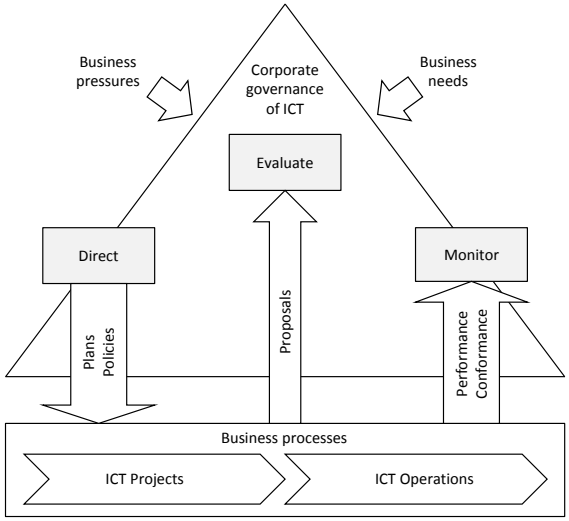

Abbildung 2.8: Das ISO/IEC 38500:2008-Modell für IT-Governance

Der Standard empfiehlt drei Prozessschritte – evaluate, direct und monitor – die entlang der sechs Prinzipien responsibility, strategy acquisition, performance, conformance und human behaviour anzuwenden sind (Racz, Weippl, & Seufert, 2010b). Darüber hinaus wird die Ergänzung eines weiteren Prozessschritts – reporting to stakeholder – diskutiert (Ohki, Harada, Kawaguchi, Shiozaki, & Kagaua, 2009; Racz, Weippl, & Seufert, 2010b).

Das von der ISACA im Jahr 1998 gegründete IT-Governance Institute (ITGI) definiert IT-Governance (auch Governance and Management of enterprise IT, GEIT) als eine „Struktur von Beziehungen und Prozessen zur Steuerung und Kontrolle des Unternehmens, um die Unternehmensziele durch das Schaffen von Mehrwert und gleichzeitig einen Ausgleich zwischen Risiko und Rendite bezogen auf die IT und die Geschäftsprozesse zu erreichen.“

Eines der am weitesten verbreiteten Rahmenwerke für die IT-Governance stellt das ebenfalls von der ISACA veröffentlichte Control Objectives for Information and Related Technology (COBIT) dar.

Im Gegensatz zu der ersten Sichtweise von IT-Governance rückt der Ansatz von (Weill & Ross, 2004) das Management der IT in den Fokus der Betrachtungen. In diesem Kontext wird IT-Governance als die „Festlegung von Entscheidungsrechten und Verantwortlichkeiten, um ein gewünschtes Verhalten bezüglich des Einsatzes von IT zu fördern“ definiert (Weill & Ross, 2004; Lewis & Millar, 2010).

Eines der bekanntesten Rahmenwerke auf dem Gebiet des IT-Management, insbesondere für das sogenannte IT Service Management (ITSM), ist die Information Technology Infrastructure Library (ITIL). Darüber hinaus sind Rahmenwerke wie PRINCE2 oder das Reifegradmodell CMMI zu nennen (Lewis & Millar, 2010).

Zur Integration der einzelnen Rahmenwerke existieren verschiedene Ansätze, die ein Mapping zwischen COBIT, ITIL, CMMI sowie verschiedenen ISO-Standards vornehmen (ISACA, 2008; ISACA, 2011). Hierbei wird deutlich, dass beide Denkschulen trotz zunächst unterschiedlicher Ausrichtungen bzw. Perspektiven und Schwerpunkten viele Gemeinsamkeiten aufweisen (Lewis & Millar, 2010; Sylvester, 2011).

IT-Risiken sind ein Teil der operationellen Risiken eines Unternehmens. Sie werden im Rahmen des IT-Risikomanagement adressiert, das immer häufiger Bestandteil des unternehmensweiten Risikomanagements (Enterprise Risk Management, ERM) ist und von den dabei zum Einsatz kommenden Rahmenwerken abgedeckt wird. Es existieren jedoch auch Rahmenwerke, die sich ausschließlich mit dem Management von IT-Risiken befassen bzw. entsprechende, IT-spezifische Komponenten aufweisen. Hierzu zählen u.a. die Informationssicherheit, der Datenschutz, IT-Sicherheitsaspekte sowie die Gewährleitung eines kontinuierlichen IT-Betriebs. Aufgrund der inhaltlichen Nähe sowie der fortschreitenden Integration von ERM und IT-Risikomanagement verschmelzen diese Disziplinen zunehmend miteinander (Racz, Weippl, & Seufert, 2010b; Racz, Weippl, & Seufert, 2010c).

Definition 2.8: IT-Compliance versteht die IT als Träger von Compliance-Anforderungen, die sowohl von externen Gesetzen und Verordnungen als auch (unternehmens-) internen Richtlinien stammen können (Klotz & Dorn, 2008).

IT-Compliance-Risiken ergeben sich somit als Schnittmenge von IT-Risiken und Risiken, die sich aus der Nichteinhaltung bzw. Verletzung von Regeln für die IT er-

geben. Im Rahmen der IT-Compliance werden somit nicht alle, sondern nur ein Teil der IT-Risiken eines Unternehmens betrachtet bzw. adressiert (Klotz & Dorn, 2008).

Im Gegensatz zu IT-GRC betrachtet eine zweite Perspektive (IT für GRC) die IT schwerpunktmäßig als Instrument zur Unterstützung von GRC bzw. der GRC-Aktivitäten eines Unternehmens (Racz, Weippl, & Seufert, 2010b).

Definition 2.9: IT-gestützte Compliance bedeutet, dass die IT als Mittel zum Erreichen von Compliance genutzt wird. Dies kann sich gleichermaßen auf die Corporate Compliance als auch auf die IT-Compliance beziehen (Klotz & Dorn, 2008).

In diesem Zusammenhang stehen seit einiger Zeit Ansätze zur Umsetzung einer IT-gestützten Compliance im Fokus des wissenschaftlichen Interesses (Klotz & Dorn, 2008). So soll die IT-gestützte Compliance neben einem besseren Überblick über die Compliance-Anforderungen eines Unternehmens eine Automatisierung von Compliance-Prüfungen, eine kontinuierliche Überwachung sowie eine bessere Unterstützung im Rahmen von internen und externen Audits ermöglichen (Sackmann, 2008). Ein besonderer Schwerpunkt liegt dabei auf dem Management von Compliance für die Geschäftsprozesse eines Unternehmens (Rinderle-Ma, Ly, & Dadam, 2008).

3 Grundlagen des Geschäftsprozessmanagements

Die Wurzeln des Geschäftsprozessmanagements gehen auf eine Bewegung der 1990er Jahre zurück, in deren Rahmen ein Ansatz entwickelt wurde, Unternehmen auf Basis ihrer Geschäftsprozesse zu strukturieren und zu organisieren (Weske, 2012; Dumas, La Rosa, Mendling, & Reijers, 2013). Dieser Trend basierte wiederum auf den frühen Arbeiten von Fritz Nordsieck, Erich Kosiol und Frederick Taylor. Nordsieck betrachtete schwerpunktmäßig die zielorientierte Zusammenarbeit von Arbeitern; darauf aufbauend entwickelte Kosiol organisatorische Prinzipien für Unternehmen (Nordsieck, 1932; Kosiol, 1962).

In seinem Buch Competitive Advantage betrachtet Michael Porter Unternehmen, deren Funktionen sowie die Beziehungen zwischen Geschäftspartnern aus einer übergeordneten und ganzheitlichen Perspektive. Hierfür werden die einzelnen Funktionen eines Unternehmens mit Hilfe des dafür entwickelten Ansatzes der Wertschöpfungskette (value chain) beschrieben und zueinander in Beziehung gesetzt. Aus den Beziehungen der Wertschöpfungsketten der Geschäftspartner zueinander ergibt sich wiederum ein Wertschöpfungssystem (value system). Um den Beitrag einer Wertschöpfungskette zu den Geschäftszielen und dem Unternehmenserfolg analysieren zu können, werden die einzelnen Funktionen von Unternehmen im Anschluss in feingranulare Einheiten heruntergebrochen (Porter, 1998).

Michael Hammer und James Champy beschreiben in ihrem Buch Reengineering the Corporation den Ansatz einer radikalen Neugestaltung der Geschäftsprozesse von Unternehmen. In diesem Zusammenhang definieren sie Geschäftsprozesse als „eine Menge von Aktivitäten, die eine oder mehrere Eingaben entgegennehmen und als Ausgabe etwas erzeugen, das für einen Kunden von Wert ist“ (Hammer & Champy, 1993).

Es zeigte sich jedoch, dass eine evolutionäre und kontinuierliche Verbesserung der Geschäftsprozesse unter Berücksichtigung organisatorischer, menschlicher und soziologischer Faktoren in der Regel einer radikalen Neugestaltung vorzuziehen ist (van der Aalst & Stahl, 2011; Weske, 2012). Dieser Ansatz der kontinuierlichen Verbesserung ist auch unter dem Begriff continuous process improvement (CPI) bekannt (Harrington, 1991).

Thomas Davenport definiert einen Geschäftsprozess als „eine Menge logisch verbundener Aufgaben, die ausgeführt werden, um ein bestimmtes Ergebnis für einen bestimmten Kunden oder Markt zu erreichen“ (Davenport, 1992). Darüber hinaus beinhaltet ein Geschäftsprozess „eine bestimmte zeitliche sowie örtliche Reihenfolge von Arbeitsschritten, die einen Anfang und ein Ende sowie eindeutig identifizierbare Eingaben und Ausgaben besitzen“ (Davenport, 1992). Darüber hinaus können Geschäftsprozesse sowohl interne als auch externe Kunden haben und unternehmensintern sowie unternehmensübergreifend ausgeführt werden (Davenport, 1992).

Definition 3.1: Ein Geschäftsprozess (business process) ist „eine Menge von manuellen, teilautomatisierten oder automatisierten Aktivitäten, die in einem Unternehmen ausgeführt werden. Durch die Ausführung dieser Aktivtäten wird ein definiertes Ziel unter Berücksichtigung bestimmter Regeln angestrebt. Die Aktivitäten werden durch Ressourcen ausgeführt. Es wird zwischen personellen und nicht-personellen (maschinellen) Ressourcen differenziert. Eine Aufgabe ist erfüllt bzw. nicht erfüllt, wenn eine oder mehrere Aktivitäten ausgeführt werden. Führen mindestens zwei Ressourcen eine Aktivität aus, so liegt ein kollaborativer Geschäftsprozess vor. Ein verteilter Geschäftsprozess ist dadurch gekennzeichnet, dass er nicht lokal, sondern an zwei geographisch unterschiedlichen Orten ausgeführt wird. Wird ein Geschäftsprozess ausschließlich innerhalb eines einzelnen Unternehmens ausgeführt, so beschreibt dies einen innerbetrieblichen Geschäftsprozess. Sind mindestens zwei Unternehmen an der Ausführung eines Geschäftsprozesses beteiligt, so bezeichnet dies einen überbetrieblichen Geschäftsprozess“ (Oberweis, 1996).

Eine ähnliche Definition liefert Matthias Weske, der einen Geschäftsprozess als eine Menge von Aktivitäten definiert, die „in einem organisatorischen und technischen Umfeld koordiniert und ausgeführt werden, um gemeinsam ein bestimmtes Geschäftsziel zu erreichen. Jeder Geschäftsprozess wird von einem einzigen Unternehmen ausgeführt, kann aber mit den Geschäftsprozessen anderer Unternehmen interagieren“ (Weske, 2012).

Die im Jahr 1993 gegründete Workflow Management Coalition definiert einen Geschäftsprozess als „eine Menge von zusammenhängenden Verfahren oder Aktivitäten, die ein gemeinsames Geschäftsziel oder eine Unternehmenspolitik im Rahmen einer organisatorischen Struktur, die funktionale Rollen und Beziehungen vorgibt, umsetzen“ (WFMC, 1999).

Eine mögliche Einordnung und Unterscheidung von Geschäftsprozessen kann anhand verschiedener Abstraktionsebenen vorgenommen werden (Abbildung 3.1) (Weske, 2012).

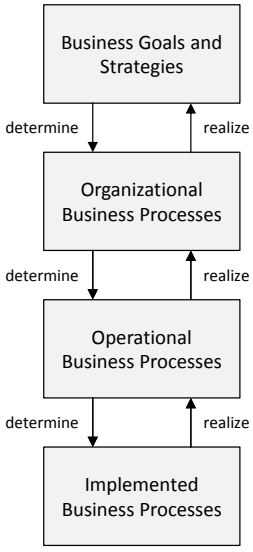

Abbildung 3.1: Abstraktionsebenen für Geschäftsprozesse

Auf der obersten Ebene befinden sich die langfristig angelegten Geschäftsziele sowie die zu deren Erreichung benötigten Geschäftsstrategien. Auf der zweiten Ebene liegen die organisatorischen Geschäftsprozesse, die in der Regel mit Hilfe von informellen (natürlichsprachlichen bzw. textuellen) sowie semiformalen Techniken beschrieben werden. Darunter befinden sich die operationellen Geschäftsprozesse, im Rahmen derer die einzelnen Aktivitäten und ihre Beziehungen näher spezifiziert und durch Geschäftsprozessmodelle abgebildet werden. Auf der untersten Ebene werden die zuvor beschriebenen, operationellen Geschäftsprozesse bzw. deren Geschäftsprozessmodelle umgesetzt bzw. implementiert und gegebenenfalls mit Hilfe von Informationssystemen ausgeführt (Weske, 2012; Dumas, La Rosa, Mendling, & Reijers, 2013).

Darüber hinaus können unternehmensinterne und unternehmensübergreifende Geschäftsprozesse nach dem Automatisierungsgrad, der Wiederholungshäufigkeit sowie dem Grad der Strukturierung unterschieden werden. Aufbauend auf diese Einordnungen und Unterscheidungen beinhaltet das Geschäftsprozessmanagement noch weitere, zusätzliche Aktivitäten (Weske, 2012).

Definition 3.2: Das Geschäftsprozessmanagement (business process management) umfasst „Konzepte, Methoden und Techniken zum Entwurf, der Verwaltung, der Konfiguration, der Ausführung und der Analyse von Geschäftsprozessen. Darüber

hinaus stellt die kontinuierliche Verbesserung der Geschäftsprozesse einen wichtigen Bestandteil des Geschäftsprozessmanagements dar“ (Weske, 2012).

Während das Geschäftsprozessmanagement auf einer strategischen bzw. organisatorischen Ebene eine wesentliche Voraussetzung dafür ist, die grundlegende Funktionsweise eines Unternehmens zu verstehen, spielt es darüber hinaus eine zunehmend wichtige Rolle beim Entwurf und der Umsetzung operationeller Geschäftsprozesse, insbesondere mit Hilfe von Informationssystemen (van der Aalst, 2011).

Sowohl die Prozessorientierung als auch das Geschäftsprozessmanagement können im Kontext der Entwicklung von Informationssystemen und damit verbundener IT-Architekturen betrachtet werden. Diese Entwicklungen gehen einerseits auf die von Edgar Dijkstra vorgestellten, grundlegenden Prinzipien der Trennung von Belangen (separation of concerns), andererseits auf den von David Parnas geprägten Modulbegriff und das damit verbundene Geheimnisprinzip (information hiding) zurück (Parnas, 1972; Dijkstra, 1982). Beide Ansätze tragen zu einer stärkeren Modularisierung von Funktionalitäten und somit zu einer höheren Flexibilität von Informationssystemen in Hinblick auf dynamische Marktanforderungen sowie technologische und rechtliche Veränderungen bei (Weske, 2012).

Aufgrund des durch die Globalisierung gestiegenen Wettbewerbdrucks hat die informationstechnologische Unterstützung von Geschäftsprozessen, insbesondere im Rahmen des Geschäftsprozessmanagements, kontinuierlich an Bedeutung gewonnen (Draheim, 2010). So können Aktivitäten, die traditionell manuell ausgeführt wurden, inzwischen zu großen Teilen mit Hilfe von Softwaresystemen abgebildet und umgesetzt werden (Weske, 2012).

Definition 3.3: Ein Geschäftsprozessmanagementsystem (business process management system) ist „ein generisches Softwaresystem, das durch die Verwendung expliziter Geschäftsprozessrepräsentationen die Ausführung von Geschäftsprozessen koordiniert“ (Weske, 2012).

Die explizite Repräsentation der Geschäftsprozesse erfolgt in Form von Geschäftsprozessmodellen, die die Grundlage für eine anschließende technische Implementierung und Ausführung mit Hilfe von Geschäftsprozessmanagementsystemen darstellen. Die Ausführung der Geschäftsprozesse kann zudem durch organisatorische Strukturen und Maßnahmen unterstützt werden oder durch diese erfolgen (Weske, 2012).

In diesem Zusammenhang werden in der Literatur häufig die Begriffe Workflow, Workflow Management und Workflow Management System verwendet (Draheim, 2010; van der Aalst, 2011).

Definition 3.4: Ein Workflow ist „die Automatisierung eines Geschäftsprozesses, im Ganzen oder in Teilen, während dessen Ablaufs Dokumente, Informationen oder Aufgaben anhand einer Menge von prozeduralen Regeln von einem Teilnehmer an einen anderen zur Bearbeitung übergeben werden“ (WFMC, 1999).

Ergänzend zum Geschäftsprozessmanagement, das maßgeblich auf den oberen Abstraktionsebenen von Geschäftsprozessen zur Anwendung kommt (Abbildung 10), liegt der Schwerpunkt des Workflow Managements auf der expliziten Repräsentation der Geschäftsprozesse in Form von Geschäftsprozessmodellen sowie der darauf basierenden technischen Implementierung und kontrollierten Ausführung mit Hilfe von Workflow Management Systemen (van der Aalst, 2011; Weske, 2012). Dabei bedient sich das Workflow Management (WFM) in der Regel eines modellgetriebenen Ansatzes (Draheim, 2010).

Definition 3.5: Ein Workflow Management System (WFMS) ist „ein Softwaresystem, das die Ausführung von Workflows durch den Einsatz von Software spezifiziert, erstellt und verwaltet. Die Ausführung erfolgt auf einer oder mehreren Workflow Engines, die in der Lage sind, die expliziten Repräsentationen der Geschäftsprozesse zu interpretieren, mit verschiedenen Workflow-Teilnehmern zu interagieren und wo nötig IT-Werkzeuge und -Anwendungen aufzurufen“ (WFMC, 1999).

Es können verschiedene Arten von Workflows unterschieden werden: single-application workflows bestehen aus kausal zusammenhängenden und zeitlich geordneten Aktivitäten, die von einem Anwendungssystem umgesetzt bzw. ausgeführt werden. Werden die Aktivitäten von mehreren, integrierten Anwendungssystemen realisiert, wird von multiple-application workflows gesprochen. System workflows bestehen aus Aktivitäten, die von einem Softwaresystem ohne Einbeziehung eines Anwenders ausgeführt werden. Im Gegensatz dazu wird bei einer aktiven Beteiligung von Anwendern in der Interaktion mit Informationssystemen von human interaction workflows gesprochen (Weske, 2012).

Das Zusammenspiel der im Rahmen des Geschäftsprozessmanagements zum Einsatz kommenden Konzepte, Methoden und Technologien wird häufig mit Hilfe eines Geschäftsprozesslebenszyklus (business process lifecycle) veranschaulicht (Abbildung 3.2) (Weske, 2012):

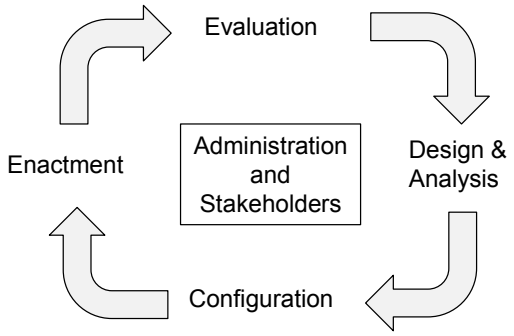

Abbildung 3.2: Geschäftsprozesslebenszyklus

Ein Geschäftsprozesslebenszyklus besteht aus mehreren in Beziehung zueinander stehenden Phasen, die zur Veranschaulichung ihrer logischen Abhängigkeiten in einer zyklischen Struktur angeordnet sind, jedoch keine strenge zeitliche Abfolge beschreiben, in der die Phasen ausgeführt werden. So können viele Entwurfs- und Entwicklungsaktivitäten bereits während jeder dieser Phasen erfolgen (Draheim, 2010; Weske, 2012; Dumas, La Rosa, Mendling, & Reijers, 2013).

- Design & Analysis: In der Entwurfs- und Analysephase werden die Geschäftsprozesse eines Unternehmens identifiziert, bewertet, validiert und mit Hilfe von Geschäftsprozessmodellen veranschaulicht. Hierfür können verschiedene Modellierungs-, Verifikations-, Simulations- und Validierungstechniken zum Einsatz kommen.
- Configuration: In der Konfigurationsphase werden die zuvor entworfenen Geschäftsprozesse (technisch) umgesetzt. Die Umsetzung kann einerseits durch eine Menge an Regeln und Arbeitsanweisungen erfolgen; andererseits können die Geschäftsprozesse mit Hilfe eines Softwaresystems realisiert bzw. unterstützt werden. Hierzu werden die Geschäftsprozessmodelle mit technischen Informationen versehen, die die Ausführung auf einem Geschäftsprozessmanagementsystem bzw. Workflow Management System ermöglichen. Die Konfiguration betrifft neben der Spezifikation der Interaktion der Mitarbeiter mit dem System zudem dessen Integration mit anderen (bereits bestehenden) Systemen des Unternehmens (z.B. ERP- und CRM-Systemen).
- Enactment: In der Ausführungsphase erfolgt die Ausführung der zuvor konfigurierten Geschäftsprozesse. Hierfür werden Instanzen der Geschäftsprozesse bzw. Geschäftsprozessmodelle erzeugt und gesteuert. Häufig verfügt ein Geschäftsprozessmanagementsystem über eine Monitoring-Komponente, die detaillierte Statusinformationen zu den einzelnen Geschäftsprozessinstanzen

bereithält und diese mit Hilfe verschiedener Visualisierungstechniken (z.B. mit Dashboards) darstellt. Während der Ausführung werden in der Regel Daten – häufig in Form von Protokolldateien – erzeugt. Diese Daten können als Grundlage für weitere Untersuchungen verwendet werden.

- Evaluation: In der Evaluationsphase werden die zu den Geschäftsprozessen bzw. Geschäftsprozessinstanzen vorliegenden bzw. zur Verfügung stehenden Informationen auf mögliche Regelverletzungen und Verbesserungspotentiale untersucht. Um die Qualität der Geschäftsprozesse analysieren und beurteilen zu können werden u.a. Protokolldateien herangezogen. Die Analyseergebnisse können wiederum in die vorgelagerte Entwurfs- und Analysephase einbezogen werden.

Im Rahmen der Verwaltung (Administration) werden die in den einzelnen Phasen des Geschäftsprozesslebenszyklus anfallenden und in unterschiedlichen Abstraktionsebenen vorliegenden Artefakte strukturiert und organisiert. Dies kann einerseits Geschäftsprozessmodelle, andererseits Informationen über einzelne Geschäftsprozessinstanzen umfassen, die wiederum in Form von Protokolldateien vorliegen können und in entsprechenden Datenbanken (Repositories) gespeichert werden (Weske, 2012).

Des Weiteren sind verschiedene Interessensgruppen und Akteure (Stakeholder) in die einzelnen Phasen des Geschäftsprozesslebenszyklus einzubinden, die über unterschiedliches Wissen sowie unterschiedliche Fähigkeiten und Kompetenzen verfügen (Weske, 2012; Dumas, La Rosa, Mendling, & Reijers, 2013):

- Der Chief Process Officer (CPO) ist für die Standardisierung und Harmonisierung der Geschäftsprozesse des Unternehmens verantwortlich. Dies schließt die Weiterentwicklung der Geschäftsprozesse unter Einbeziehung der gegenwärtigen und zukünftigen Marktanforderungen ein.
- Business Engineers sind Geschäftsbereichs- bzw. Domänenexperten eines Unternehmens. Sie sind für die Festlegung der strategischen Ziele sowie die Gestaltung der Geschäftsprozesse des Unternehmens verantwortlich.
- Process Designer verantworten die Modellierung der Geschäftsprozesse eines Unternehmens. Hierzu stehen sie in engem Austausch mit den Geschäftsbereich- bzw. Domänenexperten.
- Process Participants (Prozessbeteiligte bzw. -teilnehmer) führen die im Rahmen der einzelnen Geschäftsprozessinstanzen anfallende Arbeit aus. Sie verfügen über tiefergehende Einblicke in die operationellen Geschäftsprozesse des Unternehmens.

- Knowledge Worker (Wissensarbeiter) sind Prozessbeteiligte bzw. -teilnehmer, die Softwaresysteme zur Ausführung von Aktivitäten in einem Geschäftsprozess einsetzen. Sie verfügen in der Regel über detailliertes bzw. spezifisches Wissen über die Anwendungsdomäne und können häufig einzelne Aktivitäten oder auch Teile eines Geschäftsprozesses eigenständig bearbeiten.
- Process Owner (Prozesseigentümer) verantworten die korrekte und effiziente Ausführung eines Geschäftsprozesses oder mehrerer Geschäftsprozesse. Darüber hinaus sind sie für die Erkennung von Ineffizienzen als auch für die Weiterentwicklung und Verbesserung zuständig. Sie arbeiten eng mit den Prozessbeteiligten und den Prozessentwicklern (Process Designer) zusammen.
- System Architects (Systemarchitekten) sind für den Entwurf, die Entwicklung und die Konfiguration von Geschäftsprozessmanagementsystemen verantwortlich. Sie stellen sicher, dass die zugrundeliegenden Geschäfts-prozessmodelle korrekt instanziiert und in der IT-Landschaft bzw. den Softwaresystemen des Unternehmens ausgeführt werden.
- Software Developer (Softwareentwickler) sind für die technische Umsetzung bzw. Implementierung von Geschäftsprozessen und daraus hervorgehender Anforderungen in Form von Softwareartefakten verantwortlich. Darüber hinaus entwickeln sie Schnittstellen zu anderen Softwaresystemen.

Um die Geschäftsziele und -strategien eines Unternehmens zu erreichen und die organisatorischen und operationellen Geschäftsprozesse erfolgreich umsetzen zu können, ist eine kontinuierliche Zusammenarbeit aller beteiligten Interessensgruppen und Akteure erforderlich. Der Geschäftsprozesslebenszyklus stellt hierfür eine grundlegende Orientierung und Strukturierung relevanter Konzepte, Methoden und Technologien bereit (Weske, 2012).

4 Grundlagen der Geschäftsprozessmodellierung

In diesem Kapitel wird zunächst eine Einführung in die Grundlagen der Modellierung gegeben. Im Anschluss wird eine Auswahl von Modellierungssprachen zur Geschäftsprozessmodellierung vorgestellt. Abschließend wird auf das Konzept der Workflow Patterns eingegangen.

Modelle werden zu Erklärung und Gestaltung realer Systeme eingesetzt. Anhand der Ähnlichkeit, die zwischen dem realen System und dem Modell als Abbild dieses Systems bestehen, können verschiedene Erkenntnisse über Zusammenhänge bestimmter Sachverhalte gewonnen werden (Becker, Rosemann, & Schütte, 1995; Adam, 1996). In diesem Kontext stellen Modelle materielle oder immaterielle Repräsentationen eines Originals für Zwecke eines Subjekts (eines Konstrukteurs oder Modellierers) dar und können als Abstraktionen bzw. Abbildungen der Realwelt (des Objektsystems) verstanden werden (Becker, Rosemann, & Schütte, 1995). Von welchen Dingen der Realwelt abstrahiert wird, richtet sich nach dem Zweck, den das Modell erfüllen soll. Daher können die Anforderungen an die Modellierungsaspekte wie Art und Umfang der dargestellten Sachverhalte und die Detaillierung der Modelle je nach Zweck sehr unterschiedlich sein (Becker, Probandt, & Vering, 2012).

Modelle bilden sowohl im Rahmen der Organisations-, der Informationssystem- als auch der Anwendungssystemgestaltung eine Schnittstelle zwischen den betrieblichen bzw. betriebswirtschaftlichen Anforderungen und der konkreten Umsetzung bzw. Implementierung dieser Anforderungen (Becker, Rosemann, & Schütte, 1995; Becker, 1998; Becker, Probandt, & Vering, 2012). Diese immateriellen Modelle (auch Informationsmodelle) stellen eine Konkretisierung des allgemeinen Modellbegriffs für Zwecke des Organisations- und Informationsgestalters (des Subjekts) dar (Becker, Rosemann, & Schütte, 1995; Becker, Probandt, & Vering, 2012). Je nach Ausprägung bzw. Konkretisierung kann es sich dabei entweder um Referenz- oder um unternehmensspezifische Informationsmodelle handeln (Becker, Rosemann, & Schütte, 1995).

In Anlehnung an die Grundsätze ordnungsmäßiger Buchführung (GoB) veröffentlichten (Becker, Rosemann, & Schütte, 1995) im Jahr 1995 die Grundlagen ordnungsmäßiger Modellierung (GoM), die verschiedene Ziele und Modellierungskonventionen beschreiben, um sowohl Aussagen über die Qualität von Informationsmodellen treffen zu können als auch eine Verbesserung der Modellqualität zu erreichen. Die GoM verfolgen

hinsichtlich der Vorgabe von Zielen einen normativen Ansatz, dem ein kundenorientiertes Modellqualitätsverständnis zu Grunde liegt, d.h. die Qualität des Modells ist umso höher zu bewerten, je geringer die Differenz zwischen den Anforderungen des Modelladressaten und der tatsächlichen Eignung des Modells zur Problemlösung bzw. je höher der Abdeckungsgrad bezüglich der Anforderungen der Modellnutzer ist. Es werden die folgenden sechs Grundsätze unterschieden (Becker, Probandt, & Vering, 2012):

- Grundsatz der Richtigkeit: ein Modell ist dann syntaktisch richtig, wenn alle Regeln, die eine Modellierungssprache (z.B. in einem Metamodell) vorgibt, eingehalten werden bzw. die Anwendung des Modells konsistent zum Metamodell ist. Semantische Richtigkeit unterstellt einen konsensorientierten Richtigkeitsbegriff, d.h. ein Modell ist dann semantisch richtig, wenn im Diskurs der Gutwilligen und Sachkundigen eine Einigung erzielt werden kann.
- Grundsatz der Relevanz: es sollten nur die Sachverhalte modelliert werden, die für den zugrunde liegenden Modellierungszweck relevant sind. Es wird absichtlich vom Grundsatz der Vollständigkeit, der im Rahmen der GoB postuliert wird, abstrahiert.
- Grundsatz der Wirtschaftlichkeit: ein Modell soll so lange verfeinert werden, bis die zusätzlichen Kosten der Verfeinerung dem zusätzlichen Nutzen, der aus der Verfeinerung resultiert, entsprechen. Hierbei ist insbesondere die Nutzung von Referenzmodellen, die best practices oder common practices dokumentieren und (den Modellierer) bei der Erstellung unternehmensspezifischer Modelle anleiten bzw. unterstützen, zu berücksichtigen.
- Grundsatz der Klarheit: ein Modell ist dann klar, wenn eine leichte Lesbarkeit, Anschaulichkeit und Verständlichkeit gegeben ist. So sind neben der Hierarchisierung der Modelle z.B. Layoutkonventionen einzuhalten. Darüber hinaus können bestimmte Modellelemente bedarfsabhängig ein- oder ausgeblendet sowie Begriffs-, Struktur- und Referenzbausteine (gleiche Dinge erhalten den gleichen Bezeichner bzw. gleiche Abläufe die gleiche Struktur mit gleichem Namen) verwendet werden.
- Grundsatz der Vergleichbarkeit: eine Vergleichbarkeit von Modellen ist einerseits dann gegeben, wenn Abläufe, die in der Realwelt oder der Darstellungswelt gleich sind und innerhalb einer Modellierungssprache dokumentiert bzw. beschrieben wurden, auch im Modell identisch sind. Andererseits ist eine Vergleichbarkeit von in unterschiedlichen Modellierungssprachen dokumentierten bzw. beschriebenen Modellen gegeben, wenn diese auf ähnlichen Konstrukten aufbauen und/oder mit Hilfe einer (Meta-) Modelltransformation ineinander überführbar sind.

- Grundsatz des systematischen Aufbaus: zur Sicherstellung der Konsistenz des Gesamtmodells sollten Sachverhalte der Realwelt oder der Darstellungswelt, die aus unterschiedlichen Sichten beschrieben werden, in den Modellen gleichermaßen abgebildet werden.

Um die in ihrer Grundform allgemein gehaltenen GoM in der betrieblichen Praxis anwenden zu können, bedarf es ihrer Ausgestaltung in Form operativ umsetzbarer Handlungsanweisungen und Vorgaben. Hierfür ist – basierend auf initial zu formulierenden Modellierungszielen – zunächst festzulegen, welche Sachverhalte modelliert werden sollen (Becker, Probandt, & Vering, 2012). Im Anschluss werden relevante Sichten (z.B. Funktionssicht, Organisationssicht, Prozesssicht, Datensicht) ausgewählt und je Sicht geeignete Modelltypen (z.B. Funktionsmodelle, Organisationsmodelle, Prozessmodelle, Datenmodelle) bestimmt (Becker, Rosemann, & von Uthmann, 2000; Weske, 2012). Dabei lassen sich den mit den GoM spezifizierten Zielen gegebenenfalls mehrere konkrete Modellierungskonventionen zuweisen (Becker, Probandt, & Vering, 2012).

Angelehnt an die GoM stellen (Mendling, Reijers, & van der Aalst, 2010) sieben speziell für die Prozessmodellierung gültige Richtlinien bzw. Konventionen (Seven Process Modeling Guidelines, 7PMG) vor. Sie sollen die Konsistenz und Integrität der Modelle insbesondere bei der Erstellung bzw. Pflege durch mehrere Personen sicherstellen. Gleichzeitig sollen die Richtlinien die Lesbarkeit und Vergleichbarkeit der Modelle verbessern und eine effiziente Analyse der Modelle ermöglichen. Die Richtlinien lassen sich in Namenskonventionen, Modellierungskonventionen sowie Restriktionen unterscheiden (Dumas, La Rosa, Mendling, & Reijers, 2013):

- Es sollten so wenige Elemente wie möglich verwendet werden.
- Die Anzahl der Pfade pro Element sollte minimiert werden.
- Es sollte ein Start- und ein Endereignis geben.
- Bei der Modellierung sollte so strukturiert wie möglich vorgegangen werden.
- Die Verwendung von Oder-Verzweigungen sollte vermieden werden.
- Für Aktivitäten sollten Verb-Objekt-Beschriftungen verwendet werden.
- Modelle mit mehr als 30 Elementen sollten weiter verfeinert werden.

Eine Qualitätsbeurteilung von Modellen erfolgt in der Regel in einem konkreten Kontext, d.h. unter Berücksichtigung der beabsichtigten Verwendungszwecke (Becker, Probandt, & Vering, 2012). In diesem Zusammenhang kann die syntaktische, die semantische und die pragmatische Qualität unterschieden werden. Die Überprüfung bzw. Sicherstellung der Modellqualität erfolgt im Rahmen der Verifikation, der Validierung oder der Zertifizierung (Abbildung 4.1) (Dumas, La Rosa, Mendling, & Reijers, 2013).

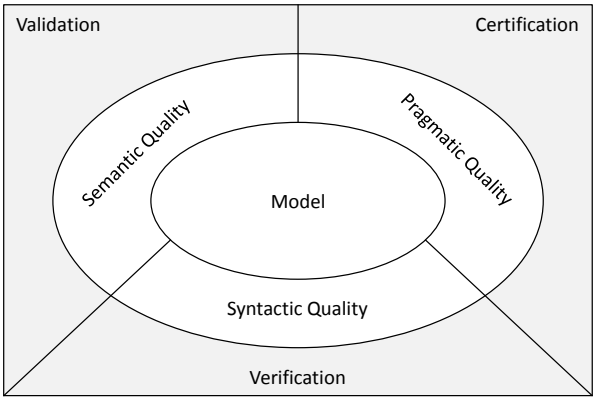

Abbildung 4.1: Qualitätsaspekte und Qualitätssicherungsaktivitäten

Die syntaktische Qualität bezieht sich auf die Konformität eines Modells zu syntaktischen Regeln und Richtlinien. Die formalen Eigenschaften eines Modells können ohne Kenntnis der Realwelt überprüft werden. Im Rahmen der Prozessmodellierung werden einerseits die strukturelle Korrektheit, andererseits die Korrektheit des Verhaltens unterschieden (Desel, 2002; Dumas, La Rosa, Mendling, & Reijers, 2013).

Ziel der semantischen Qualität sind Modelle, die die Realwelt unter Berücksichtigung eines bestimmten Abstraktionsgrades wahrheitsgemäß abbilden. Da hierfür keine formalen Regeln oder Richtlinien existieren, erfolgt die Validierung (der Modelle) durch einen Vergleich mit dem tatsächlichen Verhalten. Dabei stellen die Validität (validity) und die Vollständigkeit (completeness) die beiden grundlegenden Aspekte bzw. Bestandteile der semantischen Qualität dar (Desel, 2002; Dumas, La Rosa, Mendling, & Reijers, 2013).

Die pragmatische Qualität verfolgt das Ziel, eine einfache Benutzbarkeit der Modelle zu gewährleisten. Eine Überprüfung und Bewertung kann im Rahmen einer Zertifizierung erfolgen. Wesentliche Aspekte bzw. Bestandteile umfassen die Verständlichkeit (understandability) und die Wartbarkeit (maintainability) sowie die Güte der Modelle hinsichtlich der Abbildung der realen Welt (learning) (Dumas, La Rosa, Mendling, & Reijers, 2013).

Weitere Ansätze zur Bewertung bzw. Beurteilung der Qualität von Prozessmodellen sind das Semiotic Quality Model (SEQUAL) sowie das 3QM-Rahmenwerk (Lindland, Sindre, & Solvberg, 1994; Overhage, Birkmeier, & Schlauderer, 2012).

4.1 Sprachen zur Geschäftsprozessmodellierung

Die Dokumentation von Geschäftsprozessen erfolgt in der Regel mit Hilfe von Modellierungssprachen. Diese sollen einerseits so gut verständlich sein, dass sie den unterschiedlichen Ansprüchen der Interessensgruppen gerecht werden, andererseits sollen sie so formal sein, dass sie als Grundlage für die anschließende Gestaltung sowie Umsetzung bzw. Implementierung der Geschäftsprozesse (und gegebenenfalls damit verbundener Informationssysteme) verwendet werden können (Becker, Probandt, & Vering, 2012).

Im Folgenden wird eine Auswahl semiformaler und formaler Modellierungssprachen vorgestellt: Ereignisgesteuerte Prozessketten (EPK), die Business Process Model and Notation (BPMN) und Petri-Netze. Des Weiteren wird auf eine spezielle Klasse der Petri-Netze, die sogenannten Workflow-Netze, eingegangen. Darüber hinaus existieren eine Reihe weiterer Modellierungssprachen und -notationen, darunter UML-Aktivitätsdiagramme (UML AD), die XML Process Defintion Language (XPDL), Yet Another Workflow Language (YAWL) oder die Business Process Execution Language (WS-BPEL) (Weske, 2012).

4.1.1 Ereignisgesteuerte Prozessketten

Ereignisgesteuerte Prozessketten (EPK) sind eine semiformale Notation zur Modellierung von Geschäftsprozessen. EPK sind Teil eines ganzheitlichen Modellierungsansatzes, der von einer Arbeitsgruppe unter der Leitung von August-Wilhelm Scheer im Jahr 1992 entwickelt wurde. Ein wesentlicher Bestandteil des als Architektur Integrierter Informationssysteme (ARIS) bekannten Rahmenwerks ist das ARIS-Haus, das verschiedene Sichten und Ebenen unterscheidet. Das Dach des Hauses repräsentiert die Organisationssicht, während die Säulen die Datensicht, die Steuerungssicht sowie die Funktionssicht darstellen (Abbildung 4.2) (Scheer, Nüttgens, & Zimmermann, 1995).

Die Organisationssicht beschreibt die organisatorische Struktur eines Unternehmens und beinhaltet eine Beschreibung der Organisationseinheiten einschließlich der Beziehungen von Positionen, Rollen, Fähigkeiten und der damit verbundenen Ressourcen bzw. Personen. Zur Modellierung kommen in der Regel Organigramme zum Einsatz. Darüber hinaus können auch organisatorische Aspekte der IT-Landschaft betrachtet bzw. einbezogen werden. In der Datensicht werden alle geschäftsprozessrelevanten Datenobjekte eines Unternehmens beschrieben, die mit Entity-Relationship-Diagrammen modelliert werden. In der Funktionssicht werden die betriebswirtschaftlichen Ziele des Unternehmens beschrieben und die Beziehungen zu denjenigen Funktionen des Unter-

nehmens abgebildet, die zur Erreichung der jeweiligen Ziele beitragen. In der Steuerungssicht werden die betrieblichen Abläufe des Unternehmens als Geschäftsprozesse modelliert. Dabei wird beschrieben, welche Organisationseinheiten welche Funktionen darstellen bzw. ausführen und welche Daten dazu verwendet werden. Die Organisationssicht verbindet und integriert die einzelnen Sichten bzw. deren Artefakte miteinander. Jede Sicht umfasst darüber hinaus drei Abstraktionsebenen: eine konzeptionelle Ebene (Fachkonzept, requirements specification), eine Architekturebene (Datenverarbeitungskonzept, design specification) und eine Implementierungsebene (implementation description) (Scheer, Nüttgens, & Zimmermann, 1995).

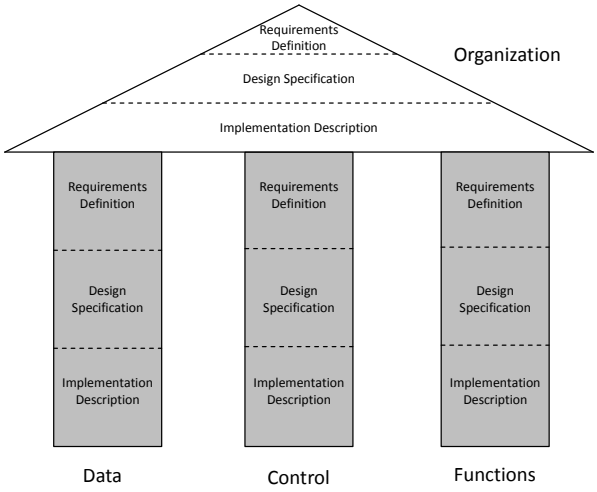

Abbildung 4.2: ARIS-Haus

Die Modellierung der betrieblichen Abläufe des Unternehmens erfolgt mit Hilfe der Ereignisgesteuerten Prozessketten, die aus den Grundelementen Ereignisse (events), Funktionen (functions), Konnektoren (connectors) und gerichteten Kanten (control flow), bestehen (Abbildung 4.3) (Scheer, 1992).

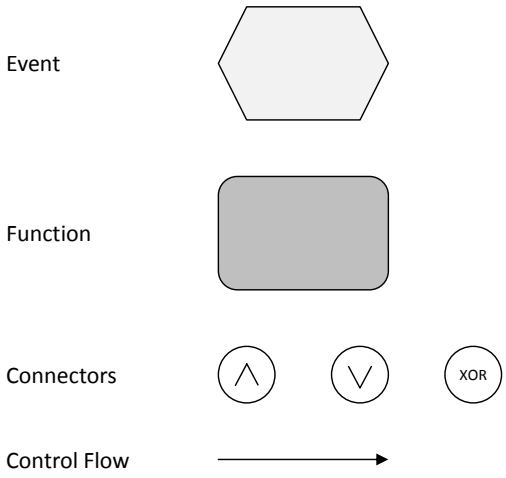

Abbildung 4.3: Grundelemente der Ereignisgesteuerten Prozessketten

Jede EPK beginnt und endet mit mindestens einem Ereignis. Ereignisse sind passive Elemente, die keine Entscheidungen treffen können. Auf Ereignisse folgen Funktionen. Funktionen sind aktive Elemente, die eine Eingabe erwarten und diese in eine Ausgabe transformieren. Funktionen werden einerseits durch Ereignisse ausgelöst, andererseits löst die Beendigung einer Funktion wiederrum Ereignisse aus. Funktionen können Entscheidungen treffen, die das Verhalten des Prozesses mit Hilfe von Konnektoren beeinflussen. Abhängig von der Anzahl eingehender und ausgehender Kanten werden die Konnektoren zur Verzweigung oder zur Zusammenführung des Kontrollflusses eingesetzt (Scheer, 1992).

Erweiterte Ereignisgesteuerte Prozessketten (eEPK) führen zusätzliche Elemente ein, die die Grundelemente der EPK um Organisations-, Informations- und Datenobjekte erweitern (Weske, 2012).

4.1.2 Business Process Model and Notation

Die Business Process Model and Notation (BPMN) wurde im Jahr 2002 von Stephen A. White zunächst unter dem Namen Business Process Modeling Notation entwickelt und im Jahr 2004 von der Business Process Management Initiative (BPMI) veröffentlicht (Weske, 2012). Im darauffolgenden Jahr wurde die Koordination der Pflege und Weiterentwicklung sowie die Standardisierung von BPMN an die Object Management Group (OMG) übergeben. Im Jahr 2011 wurde die Version 2.0 von BPMN veröffentlicht; die derzeit aktuelle Version 2.0.2 stammt aus dem Jahr 2013 (OMG, 2013).

Zur Abbildung von Geschäftsprozessen unterscheidet BPMN die folgenden drei Modelltypen, die wiederum mit Hilfe unterschiedlicher Diagramme (business process diagrams) abgebildet werden (OMG, 2013):

- Prozesse (auch workflows oder orchestration) umfassen sowohl private (auch interne) als auch öffentliche (public) Geschäftsprozesse. Private Geschäftsprozesse können ausführbar oder nicht-ausführbar sein und werden in der Regel innerhalb eines einzelnen „Schwimmbeckens“ (pool) dargestellt. Öffentliche Geschäftsprozesse stellen die Interaktionen zwischen einem privaten Geschäftsprozess und einem anderen (externen) Geschäftsprozess bzw. Beteiligten (participant) dar. Dabei werden nur die Aktivitäten dargestellt, die zur Kommunikation mit dem anderen Geschäftsprozess benötigt werden. Öffentliche Geschäftsprozesse werden in der Regel in Form von Kollaborationen dargestellt.
- Kollaborationen (collaborations) stellen die Interaktionen zwischen zwei oder mehreren Geschäftseinheiten dar. Hierfür enthalten Kollaborationen in der Regel zwei oder mehrere „Schwimmbahnen“, die die Beteiligten der Kollaboration darstellen. Der Nachrichtenaustausch zwischen den Beteiligten wird mit Hilfe eines Nachrichtenflusses (message flow) dargestellt, der die „Schwimmbecken“ miteinander verbindet und die Berührungspunkte zwischen öffentlichen Geschäftsprozessen darstellt.
- Choreographien (choreographies) definieren das erwartete Verhalten in Form eines prozeduralen Vertrags zwischen den Beteiligten eines Prozesses. Im Gegensatz zu Prozessen, die innerhalb eines „Schwimmbeckens“ dargestellt werden, finden Choreographien zwischen „Schwimmbecken“ statt. Dabei repräsentieren die Aktivitäten einer Choreographie den Austausch von Nachrichten zwischen den Beteiligten.

Die informelle Beschreibung von Kollaborationen wird darüber hinaus in Form von Konversationen abgebildet. Diese stellen eine logische Beziehung zwischen dem Nachrichtenaustausch und den beteiligten Geschäftsobjekten dar (OMG, 2013).

Um die Komplexität von Geschäftsprozessmodellen beherrschbar zu machen, werden die graphischen Elemente von BPMN in fünf Kategorien eingeordnet: Flussobjekte (flow objects), Daten (data), Verbindungsobjekte (connecting objects), Schwimmbahnen (swimlanes) und Artefakte (artifacts) (Abbildung 4.4) (OMG, 2013).

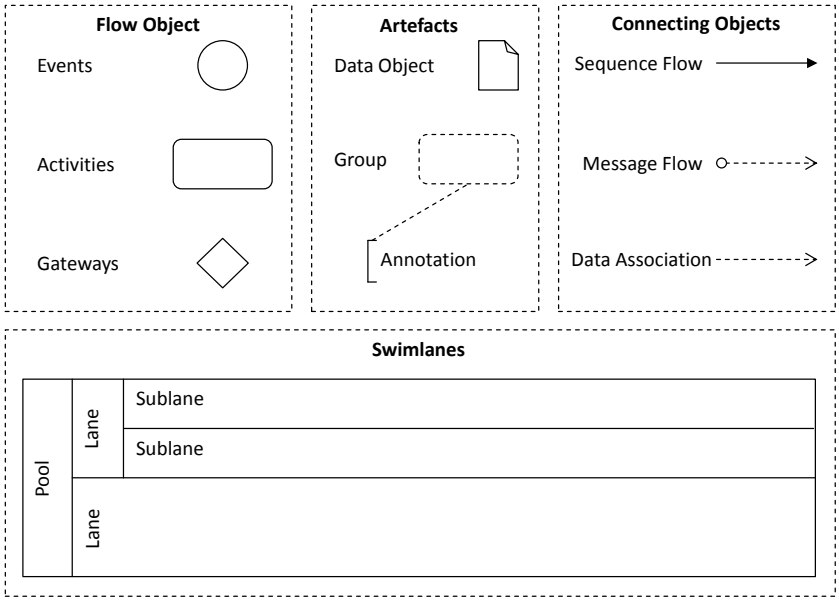

Abbildung 4.4: Kategorien und Grundelemente von BPMN

Mit Hilfe der Flussobjekte wird das grundlegende Verhalten eines Geschäftsprozesses definiert. Neben Datenobjekten werden verschiedene Verbindungsobjekte unterschieden, die die einzelnen Flussobjekte miteinander verbinden. Schwimmbecken repräsentieren die Beteiligten eines Geschäftsprozesses während Schwimmbahnen zur Organisation und Kategorisierung von Aktivitäten eingesetzt werden. Artefakte stellen Elemente zur Gruppierung und Annotation bereit (OMG, 2013).

Zusätzlich zu den Grundelementen von BPMN existieren eine Reihe weiterer Modellierungselemente, die zur Modellierung komplexerer Sachverhalte eingesetzt werden können (OMG, 2013).

Zu den wesentlichen Neuerungen von BPMN 2.0 zählen – neben einer Formalisierung der Ausführungssemantik für alle BPMN-Elemente und damit einhergehend derMöglichkeit einer direkten Ausführung von BPMN-Modellen – die Definition eines Metamodells sowie die Standardisierung eines XML-basierten Austauschformats. Hierfür werden verschiedene XSD- und XMI-Schemata sowie XSLT-Transformationen bereitgestellt. Darüber hinaus existiert ein Mapping zur Überführung von BPMN-Modellen in WS-BPEL (OMG, 2013).

4.1.3 Petri-Netze

Petri-Netze gehen auf die im Jahr 1962 veröffentlichte Dissertation „Kommunikation mit Automaten“ von Carl Adam Petri zurück, in der ein neuer Modellierungsansatz zur Ablaufbeschreibung vorgestellt wird, der neben einer grafischen Repräsentation eine mathematische Formalisierung besitzt (Petri, 1962).

Klassische Petri-Netze sind bipartite, gerichtete Graphen. Sie bestehen aus Stellen (places), Transitionen (transitions) sowie Kanten (arcs), die Stellen und Transitionen miteinander verbinden. Stellen sind die passiven, Transitionen die aktiven Elemente eines Petri-Netzes. Mit Hilfe von Marken (tokens) wird der Zustand eines Petri-Neztes abgebildet (Abbildung 4.5) (van der Aalst, 1998).

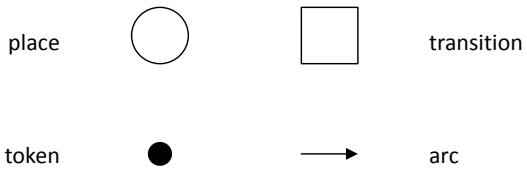

Abbildung 4.5: Grafische Repräsentation der Elemente von Petri-Netzen

Definition 4.1: Ein Petri-Netz ist ein Tripel (P, T, F) mit

1. einer endlichen Menge von Stellen P,
2. einer endlichen Menge von Transitionen T, mit $(P \cap T) = \emptyset$,
3. einer Flussrelation $F \subseteq (P \times T) \cup (T \times P)$.

Eine Stelle p wird dabei als eingehende Stelle einer Transition t bezeichnet, wenn es eine gerichtete Kante von p nach t gibt. Eine Stelle p wird als eine ausgehende Stelle einer Transition t bezeichnet, wenn es eine gerichtete Kante von t nach p gibt. Die Menge der eingehenden Stellen einer Transition t wird mit $\bullet t$ gekennzeichnet, während $p \bullet$ und $\bullet p$ die Mengen der Transitionen kennzeichnen, die sich p als jeweils eingehende und ausgehende Stelle teilen (van der Aalst, 1998).

Petri-Netze werden häufig zur Modellierung dynamischer Systeme unter Verwendung statischer Strukturen eingesetzt. Die Verteilung der Marken in den Stellen stellt den Zustand des Petri-Netzes und damit den Zustand des modellierten Systems dar (van der Aalst, 1998; Weske, 2012).

Definition 4.2: Der Zustand (auch Markierung) M eines Petri-Netzes (P, T, F) wird als die Verteilung der Marken über die Stellen definiert, d. h. $M: P \rightarrow \mathbb{N}$.

Das dynamische Verhalten wird mit Hilfe von Zustandsveränderungen modelliert. Dabei wird die Position der Marken mit Hilfe von Schaltregeln verändert. Die auf den Schaltregeln basierende Veränderung der Markenposition wird auch als Markenspiel (token play) bezeichnet (van der Aalst, 1998).

Definition 4.3: Das Schalten einer Transition repräsentiert eine Zustandsänderung des Petri-Netzes.

1. Eine Transition t ist aktiviert, wenn sich in jeder eingehenden Stelle p von t mindestens eine Marke befindet.
2. Eine aktivierte Transition kann schalten. Wenn die Transition t schaltet, konsumiert bzw. entfernt sie eine Marke aus jeder eingehenden Stelle p von t und produziert bzw. erzeugt eine Marke in jeder ausgehenden Stelle p von t.

Sei (P, T, F) ein Petri-Netz und M eine Markierung. Es gilt:

1. $M \overset{t}{\rightarrow} M'$: der Zustand des Petri-Netzes verändert sich durch das Schalten der aktivierten Transition t von M nach M'
2. $M \rightarrow M'$: es gibt eine Transition t, sodass gilt $M \overset{t}{\rightarrow} M'$
3. $M_1 \overset{*}{\rightarrow} M_n$: es gibt eine Schaltfolge $\sigma = t_1, t_2, \ldots, t_{n-1}$, die vom Zustand M_1 zum Zustand M_n führt, d.h. $M_1 \overset{t_1}{\rightarrow} M_2 \overset{t_2}{\rightarrow} \ldots \overset{t_{n-1}}{\longrightarrow} M_n$
4. Ein Zustand M' ist vom Zustand M erreichbar, wenn gilt $M \overset{*}{\rightarrow} M'$

Aufbauend auf der grundlegenden Definition klassischer Petri-Netze existieren weitere Klassen bzw. Erweiterungen, die sich hinsichtlich ihres Schaltverhaltens oder der Struktur ihrer Marken unterscheiden und sich daher zur Modellierung komplexerer Prozesse eignen (z.B. höhere Petri-Netze) (van der Aalst, 1995; Oberweis, 1996; van der Aalst, 1998; Weske, 2012).

4.1.4 Workflow-Netze

Petri-Netze werden häufig zur Abbildung von Prozessen eingesetzt. Die Abbildung von komplexeren Prozessen, insbesondere von Geschäftsprozessen, erfordert jedoch fortgeschrittene Modellierungsmechanismen (Weske, 2012).

Die von Will van der Aalst entwickelten Workflow-Netze stellen eine Erweiterung der klassischen Petri-Netze um Konzepte und eine Notation dar, um die Abbildung von Geschäftsprozessen zu erleichtern. Gleichzeitig führen Workflow-Netze strukturelle Restriktionen ein, die sich für die Modellierung von Geschäftsprozessen als nützlich erwiesen haben (van der Aalst, 1998; Weske, 2012).

Der Schwerpunkt von Workflow-Netzen liegt in Anlehnung an klassische Petri-Netze auf der Abbildung des Kontrollflusses von Prozessen. Ereignisse werden als Stellen und Aktivitäten als Transitionen dargestellt; Prozessinstanzen (cases) werden durch unterscheidbare Marken repräsentiert. Eine mögliche hierarchische Strukturierung von Workflow-Netzen kann durch eine Verfeinerung von Transitionen vorgenommen werden. Verfeinerungen werden häufig mit Hilfe eines doppelten Rahmens veranschaulicht (Abbildung 4.6) (van der Aalst, 1998; Weske, 2012).

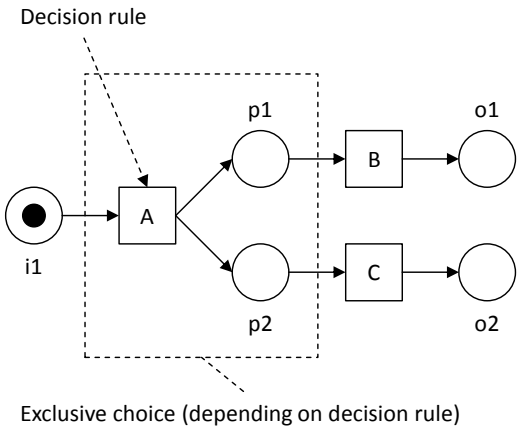

Abbildung 4.6: Grundlegende Konzepte von Workflow-Netzen

Workflow-Netze werden wie folgt definiert (van der Aalst, 1995; van der Aalst, 1998):

Definition 4.4: Ein Petri-Netz $PN = (P, T, F)$ ist ein Workflow-Netz (WF-Netz), wenn

1. es eine spezielle erste Stelle $i \in P$ gibt, die keine eingehende Kante besitzt, $\bullet i = \emptyset$,
2. es eine spezielle letzte Stelle $o \in P$ gibt, die keine ausgehende Kante besitzt, $o \bullet = \emptyset$,
3. jede Stelle p und jede Transition t auf einem Pfad von der ersten Stelle i zur letzten Stelle o liegt.

Bei Definition 4.4 handelt es sich um die Minimalanforderungen an ein Workflow-Netz. Auch wenn diese Anforderungen erfüllt werden, lassen sich Prozesse modellieren, die u.a. deadlocks oder livelocks enthalten können. Daher wird für WF-Netze in der Regel eine weitere Eigenschaft gefordert (van der Aalst, 1998):

Definition 4.5: Ein WF-Netz $PN = (P, T, F)$ ist intakt (sound, auch wohlgeformt), wenn

1. für jeden von i erreichbaren Zustand M eine Schaltfolge existiert, die vom Zustand M zum Zustand o führt,
2. der Zustand o der einzige Zustand ist, der vom Zustand i erreichbar ist und mindestens eine Marke enthält,
3. es keine toten Transitionen in (PN, i) gibt.

Neben der Unterstützung der von der WfMC identifizierten Workflow-Konstrukte (sequenzielles Routing, paralleles Routing, bedingtes Routing und iteratives Routing) führt (van der Aalst, 1998) eine spezielle Notation zur Darstellung des parallelen und bedingten Routings ein (Abbildung 4.7):

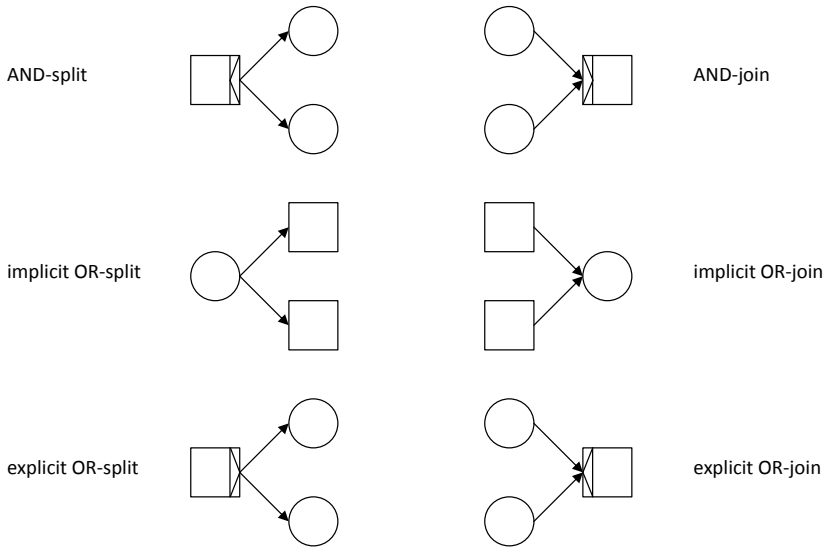

Abbildung 4.7: Paralleles und bedingtes Routing in Workflow-Netzen

Eine wesentliche Erweiterung der Workflow-Netze sind die sogenannten Auslöser (trigger), die Annotationen zu Transitionen darstellen und Informationen darüber liefern, wer oder was für die Aktivierung sowie die Ausführung einer Aktivität verantwortlich bzw. zuständig ist (Abbildung 4.8) (van der Aalst, 1998).

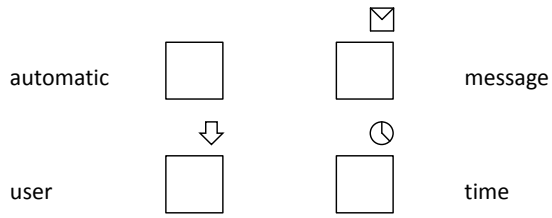

Abbildung 4.8: Auslöser in Workflow-Netzen

Die Abbildung organisatorischer Aspekte (z.B. Rollen oder Ressourcen) sowie von Daten oder (Ausführungs-)Bedingungen wird nicht explizit unterstützt und liegt in der Verantwortung des Modellierers bzw. ist mit Hilfe geeigneter Modellierungstechniken bzw. -werkzeugen abzudecken (van der Aalst, 1998; Weske, 2012).

4.2 Workflow Patterns

Die Verwendung von Mustern zur Kategorisierung wiederholt auftretender Probleme und Lösungen innerhalb einer bestimmten Domäne sowie das Konzept zur Beschreibung von Beziehungen zwischen bestimmten Mustern geht auf Christopher Alexander zurück (Alexander, Ishikawa, & Silverstein, 1978). Obwohl die Arbeit auf dem Gebiet der Architektur angesiedelt ist, kommt das zugrundliegende Konzept in verschiedenen anderen Domänen zur Anwendung (Russell, ter Hofstede, van der Aalst, & Mulyar, 2006).

Im Anschluss an die Gründung der Workflow Patterns Initiative im Jahr 1999, wurde im darauffolgenden Jahr ein musterbasierter Ansatz (pattern-based approach) zur Identifizierung grundlegender und im Rahmen der Geschäftsprozessmodellierung wiederholt auftretender Anforderungen, vorgeschlagen. Ziel des Ansatzes war es, diese Anforderungen ausreichend generisch sowie möglichst sprach- und technologieunabhängig zu beschreiben (van der Aalst, Barros, ter Hofstede, & Kiepuszewski, 2000). In der Folge wurden die ursprünglich identifizierten Workflow-Konstrukte sukzessive erweitert und im Jahr 2003 in der Form von 20 Kontrollflussmustern (control-flow patterns) vorgestellt (van der Aalst, ter Hofstede, Kiepuszewski, & Barros, 2003).

Im Jahr 2006 wurde eine systematische Überarbeitung der bestehenden Kontrollflussperspektiven vorgenommen. Hierzu wurden die ursprünglich identifizierten 20 Kontrollflussmuster zunächst kritisch bewertet und im Anschluss um 23 neue Kontrollflussmuster ergänzt. Neben einer formalen Beschreibung wurden darüber hinaus detaillierte Voraussetzungen und Evaluationskriterien vorgestellt, die die Implementierung der Kontrollflussmuster in kommerziellen Workflowsystemen sowie Geschäftsprozessmodellierungsformalismen und Geschäftsprozessausführungssprachen untersuchen und bewerten (Russell, ter Hofstede, van der Aalst, & Mulyar, 2006).

Die 43 Kontrollflussmuster werden in acht Kategorien eingeteilt: Basis-Kontrollflussmuster (basic control-flow patterns), fortgeschrittene Verzweigungs- und Synchronisationsmuster (advanced branching and synchronization patterns), strukturelle Muster (structural patterns), Mehrfach-Instanzmuster (multiple instance patterns), statusbasierte Muster (state-based patterns), Abbruch- und Abschlussmuster (cancellation and force completion patterns), Iterationsmuster (iteration patterns), Beendigungsmuster (termination patterns) sowie Auslösmuster (trigger patterns). Basis-Kontrollflussmuster umfassen dabei die grundlegenden Aspekte der Steuerung von Prozessen und orientierten sich an den ursprünglich vorgeschlagenen Konzepten der WFMC (Russell, ter Hofstede, van der Aalst, & Mulyar, 2006).

Im Folgenden werden die fünf grundlegenden Basis-Kontrollflussmuster sowie ein statusbasiertes Muster vorgestellt und mit Hilfe von Petri-Netzen veranschaulicht. Eingehende Stellen werden mit $i_1 \dots i_n$, ausgehende Stellen mit $o_1 \dots o_n$, interne Stellen mit $p_1 \dots p_n$ und Transitionen mit $A \dots Z$ beschriftet (Russell, ter Hofstede, van der Aalst, & Mulyar, 2006).

Bei der Sequenz (sequence) wird nach der Beendigung einer Aktivität eine nachfolgende Aktivität in demselben Prozess gestartet (Abbildung 4.9).

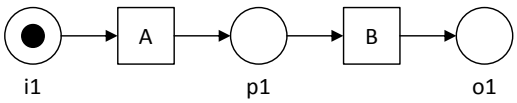

Abbildung 4.9: Kontrollflussmuster 1 (Sequenz)

Bei der parallelen Verzweigung (auch paralleles Routing, parallel split) wird eine Kante in zwei oder mehrere parallele Kanten verzweigt, die nebenläufig ausgeführt werden können (Abbildung 4.10).

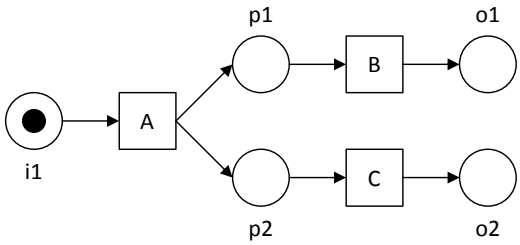

Abbildung 4.10: Kontrollflussmuster 2 (Parallele Verzweigung)

Bei der Synchronisierung (synchronization) werden zwei oder mehrere Kanten in einer einzigen ausgehenden Kante zusammengeführt. Die Kontrolle wird an die ausgehende Kante übergeben, sobald alle eingehenden Kanten aktiviert wurden (Abbildung 4.11).

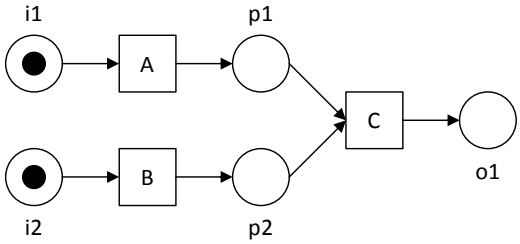

Abbildung 4.11: Kontrollflussmuster 3 (Synchronisierung)

Beim exklusiven (auch expliziten) Oder (bedingtes Routing, exclusive choice) wird eine eingehende Kante in zwei oder mehrere ausgehende Kanten verzweigt. Nachdem die eingehende Kante aktiviert wurde, wird die Kontrolle basierend auf einer mit der Kante verbundenen Entscheidungsregel bzw. dem Ergebnis eines logischen Ausdrucks umgehend an genau eine ausgehende Kante übergeben (Abbildung 4.12).

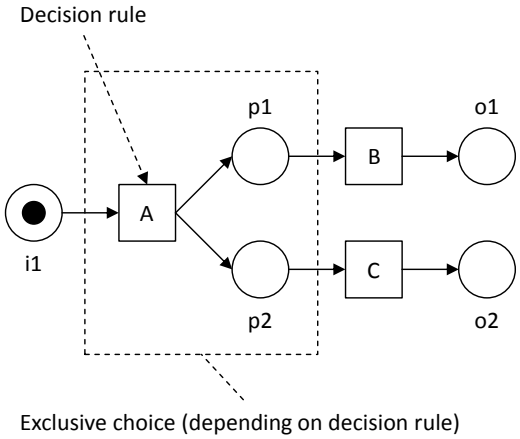

Abbildung 4.12: Kontrollflussmuster 4 (Exklusives Oder)

Bei der einfachen Zusammenführung (simple merge) werden zwei oder mehrere Kanten zu einer einzigen ausgehenden Kante vereinigt. Jede aktivierte eingehende Kante übergibt die Kontrolle unmittelbar an die ausgehende Kante (Abbildung 4.13).

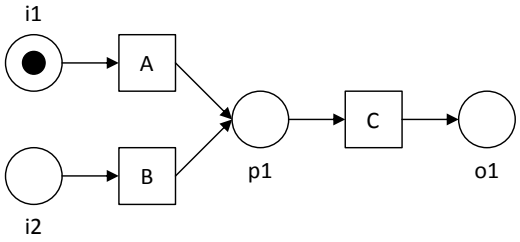

Abbildung 4.13: Kontrollflussmuster 5 (Einfache Zusammenführung)

Statusbasierte Muster bilden Situationen ab, in denen der Status einer Prozessinstanz sowohl mit der Prozessausführung verbundene Daten beinhalten als auch den Status anderer Aktivitäten einbeziehen kann (Russell, ter Hofstede, van der Aalst, & Mulyar, 2006). Beim impliziten Oder (deferred choice) stellen zunächst alle (ausgehenden) Kanten mögliche Ausführungsalternativen dar. Die Entscheidung fällt dabei zugunsten der Kante, in der die erste Aktivität gestartet wird, d.h. es existiert keine explizite Wahlmöglichkeit. Nachdem die Entscheidung für eine Kante gefallen ist, wird den anderen Kanten die Ausführungsmöglichkeit entzogen (Abbildung 4.14) (Russell, ter Hofstede, van der Aalst, & Mulyar, 2006).

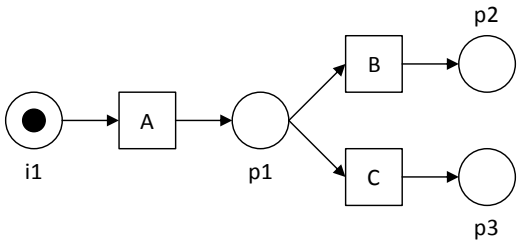

Abbildung 4.14: Kontrollflussmuster 16 (Implizites Oder)

Der ursprüngliche Schwerpunkt der Workflow Patterns lag zunächst auf der Abbildung der Kontrollflussperspektive von Workflow Systemen (Russell, ter Hofstede, van der Aalst, & Mulyar, 2006). Eine umfassende Analyse von Geschäftsprozessen bzw. Workflows erfordert jedoch zusätzlich die Einbeziehung von Ressourcen und Daten (Jablonski & Bussler, 1996). So wurden von der Workflow Patterns Initiative in den vergangenen Jahren neben 43 Ressourcenmustern (resource patterns) 40 Datenmuster (data patterns) veröffentlicht (Russell, ter Hofstede, Edmond, & van der Aalst, 2004a; Russell, ter Hofstede, Edmond, & van der Aalst, 2004b; Russell, ter Hofstede, Edmond, & van der Aalst, 2005; Russell, van der Aalst, ter Hofstede, & Edmond, 2005). Darüber hinaus wurden die Beziehungen zwischen den verschiedenen Perspektiven untersucht und ein musterbasiertes Rahmenwerk für die Ausnahmebehandlung (exception handling patterns) in Workflowsystemen vorgeschlagen (Russell, van der Aalst, & ter Hofstede, 2006a; Russell, van der Aalst, & ter Hofstede, 2006b).

Weitere im Rahmen der Workflow Patterns Initiative veröffentlichte Arbeiten befassen sich mit der Verwendung von Mustern zur Charakterisierung von (Web) Service-Interaktionen sowie der Entwicklung der Workflow Pattern Specification Language (WPSL) (Russell, ter Hofstede, van der Aalst, & Mulyar, 2006). Workflow-Netze werden im Rahmen der Evaluation (Kapitel 8) wieder aufgegriffen.

5 Compliance-Anforderungen und -Regeln für Geschäftsprozesse

In diesem Kapitel wird zunächst der Begriff der Compliance-Anforderung definiert und im Anschluss ein Lebenszyklus für das Compliance Management präsentiert. Darauf aufbauend wird ein musterbasierter Ansatz zur Spezifikation von Compliance-Regeln vorgestellt und anschließend eine Einordnung der Regeln anhand verschiedener Kategorien vorgenommen. Abschließend wird mit dem Business Process Compliance Management Model (BPCMM) ein konzeptuelles Modell entwickelt, das die Basis für eine informationstechnologische Unterstützung im Rahmen des Compliance Managements für Geschäftsprozesse legt.

5.1 Compliance-Anforderungen

Anforderungen spielen bei der Entwicklung von Produkten und Dienstleistungen eine entscheidende Rolle (Hull, Jackson, & Dick, 2011). Sie treffen Aussagen über zu erfüllende Eigenschaften oder zu erbringende Leistungen eines Systems (bzw. Produkts oder einer Dienstleistung), eines Prozesses oder am Prozess beteiligter Menschen (Balzert, 2009; Partsch, 2010).

Definition 5.1: Eine Anforderung (requirement) ist eine „Bedingung oder Fähigkeit, die ein System oder eine Komponente erfüllen bzw. haben muss, um einen Kontrakt, einen Standard, eine Spezifikation oder ein anderes formales Dokument zu erfüllen“ (IEEE, 1990).

Im Rahmen der Anforderungsdefinition bzw. -spezifikation (requirements specification) werden verschiedene Arten von Anforderungen unterschieden, die sich wiederum nach unterschiedlichen Kriterien klassifizieren lassen (Partsch, 2010).

Des Weiteren existieren eine Reihe von Qualitätskriterien, die Anforderungen erfüllen sollten (Balzert, 2009; Partsch, 2010):

- Eine Anforderung ist korrekt (correct), wenn sie die Bedürfnisse der Interessensgruppen (stakeholder) adäquat wiedergibt.
- Eine Anforderung ist eindeutig (unambiguous), wenn sie von allen Interessensgruppen gleich interpretiert wird bzw. nur eine mögliche Interpretation zulässt.

- Eine Anforderung ist vollständig (complete), wenn die geforderte Funktionalität bzw. Eigenschaft vollständig erfasst und beschrieben ist.
- Eine Anforderung ist konsistent (consistent), wenn sie in sich widerspruchsfrei ist.
- Eine Anforderung ist klassifizierbar (ranked), wenn sie z.B. nach ihrer Wichtigkeit, Stabilität oder Verbindlichkeit eingeordnet werden kann.
- Eine Anforderung ist überprüfbar (verifiable), wenn sie nach ihrer Realisierung testbar ist.
- Eine Anforderung ist verfolgbar (traceable), wenn sie eindeutig identifizierbar ist.

Darüber hinaus sollten Anforderungen modifizierbar (modifiable) und erweiterbar (extensible) sein sowie Abhängigkeiten von bzw. Beziehungen zu anderen Anforderungen angeben (Balzert, 2009; Partsch, 2010).

Zur Sicherstellung der Einhaltung der Qualitätskriterien existieren verschiedene Anforderungsattribute, die bei der Anforderungserfassung bzw. -dokumentation verwendet werden können (Balzert, 2009). Hierzu zählen u.a. ein eindeutiger Bezeichner zur Identifikation und Referenzierung der Anforderung, eine Kurzbezeichnung, ein Anforderungstyp, eine informale, semiformale oder formale Beschreibung, eine Einschätzung der Kritikalität bzw. eines damit verbundenen Risikos, eine Quelle, Querbezüge zur Darstellung von Abhängigkeiten sowie Schlüsselwörter zur Filterung der Anforderung (Balzert, 2009; Partsch, 2010; Hull, Jackson, & Dick, 2011). Für jedes Anforderungsattribut sollte zudem ein geeignetes Attributierungsschema festgelegt werden, das neben einem eindeutigen Attributnamen eine Attributsemantik, einen zugehörigen Wertebereich sowie eine Wertesemantik umfasst (Balzert, 2009).

In der Regel werden Anforderungen in natürlicher Sprache verfasst bzw. formuliert. Neben den sich daraus ergebenden Vorteilen (einfach, flexibel, universell) bestehen jedoch eine Reihe von Nachteilen (lexikalische, syntaktische, semantische und referentielle Mehrdeutigkeit) (Balzert, 2009; Partsch, 2010).

Zur Reduktion bzw. Vermeidung der syntaktischen und semantischen Mehrdeutigkeit können sogenannte Anforderungsschablonen (auch Satzschablonen bzw. Satzbaumuster, templates) verwendet werden. Darüber hinaus existieren verschiedene semiformale sowie formale Konzepte, die abhängig von den zu beschreibenden Aspekten eine präzisere Beschreibung von Anforderungen erlauben (Balzert, 2009; Partsch, 2010).

Unabhängig von einer gewählten Vorgehensweise bzw. einem Vorgehensmodell umfasst die Anforderungsdefinition bzw. -spezifikation die folgenden Aktivitäten (Balzert, 2009; Partsch, 2010):

- Im Rahmen der Anforderungsermittlung (requirements elicitiation) sind zunächst das Umfeld zu bestimmen sowie relevante Interessensgruppen zu identifizieren. Darüber hinaus sind zu berücksichtigende Gesetze, Normen und Standards einzubeziehen. Im Anschluss werden die Anforderungen mit Hilfe verschiedener Ermittlungstechniken erhoben.
- Im Anschluss an die Anforderungsermittlung folgen die systematische Spezifikation der Anforderungen sowie deren schrittweise Übertragung in Anforderungsschablonen (requirements specification). Dabei werden in der Regel verschiedene Anforderungsattribute ergänzt. Soweit möglich, können Teile der Anforderungen bereits mit Hilfe semiformaler sowie formaler Konzepte präzisiert werden.
- Im Rahmen der Anforderungsanalyse (requirements analysis) werden die einzelnen Anforderungen anhand der Qualitätskriterien überprüft. Die Überprüfung von in natürlicher Sprache dokumentierten Anforderungen erfolgt in der Regel manuell, während Anforderungen, die semiformal oder formal beschrieben sind, (semi-)automatisiert überprüft werden können.
- Abschließend erfolgt eine formelle Abnahme sowie ggf. Priorisierung der Anforderungen.

Die einzelnen Aktivitäten (auch Phasen) werden in Zusammenarbeit mit den Interessensgruppen durchlaufen und in der Regel iterativ (in Rückkopplungsschleifen) verfeinert (Partsch, 2010; Hull, Jackson, & Dick, 2011).

Die abgenommenen Anforderungen bilden im Anschluss die Grundlage für die Konzeption bzw. Modellierung einer entsprechenden (fachlichen) Lösung. Die Verwaltung, Änderung und Nachverfolgung von Anforderungen wird häufig unter dem Begriff Anforderungsmanagement (requirements management) zusammengefasst (Balzert, 2009; Partsch, 2010).

Aufbauend auf den in Kapitel 2 behandelten Grundlagen von Governance, Risk Management und Compliance sowie den aus dem Requirements Engineering bekannten Konzepten wird eine Compliance-Anforderung wie folgt definiert:

Definition 5.2: Eine Compliance-Anforderung (compliance requirement, auch compliance objective oder compliance concern) ist eine Bedingung oder Fähigkeit, die ein Unternehmen, ein Geschäftsprozess oder ein Informationssystem erfüllen bzw. haben

muss, um ein Gesetz, einen Vertrag, einen Standard, eine Norm oder ein anderes, unternehmensinternes oder -externes Dokument (z.B. eine Richtlinie oder Verfahrensanweisung) zu erfüllen.

Compliance-Anforderungen stellen häufig Einschränkungen oder Behauptungen dar und legen damit die Grenzen des erwarteten Verhaltens fest (COMPAS, 2009b; El Kharbili, 2012). Die Vielzahl unterschiedlicher Gesetze, Verträge, Standards, Normen und Richtlinien erfordert ein strukturiertes Vorgehen bei der Ermittlung der für ein Unternehmen relevanten Compliance-Anforderungen (COMPAS, 2008; COMPAS, 2009a).

Im Folgenden werden drei Beispiele (Auszüge) für Compliance-Anforderungen (Gesetz, Norm, unternehmensinterne Richtlinie) genannt:

- „[…] requiring each annual report […] to contain an internal control report, which shall […] contain an assessment, as of the end of the most recent fiscal year of the issuer, of the effectiveness of the internal control structure and procedures of the issuer for financial reporting“ (SEC, 2002).
- „Die Organisation muss in geplanten Abständen interne Audits durchführen […]. Ein Auditprogramm muss geplant werden […]. Die Auditkriterien, der Auditumfang, die Audithäufigkeit und die Auditmethoden müssen festgelegt werden […]“ (ISO, 2008).
- Im Anschluss an die Fertigung von Bauteilen des Typs [A] muss innerhalb [eines Zeitraums] eine Qualitätskontrolle gemäß [eines internen Prüfplans] durchgeführt werden.

Aufbauend auf den Arbeiten von (Giblin, Liu, Müller, Pfitzmann, & Zhou, 2005) präsentieren (Ramezani, Fahland, van der Werf, & Mattheis, 2011; Ramezani, Fahland, & van der Aalst, 2013) einen Lebenszyklus für das Compliance Management (Abbildung 5.1).

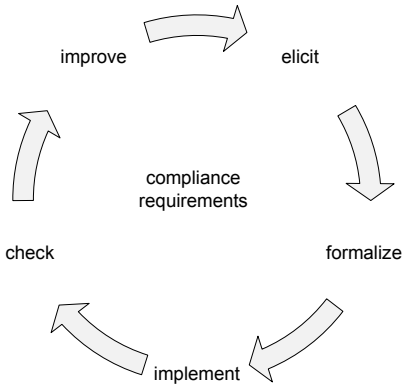

Abbildung 5.1: Lebenszyklus für das Compliance Management

Der Lebenszyklus umfasst die folgenden fünf Phasen:

- Compliance elicitation: Im Rahmen dieser Phase werden die für ein Unternehmen sowie dessen Geschäftsprozesse relevanten Compliance-Anforderungen ermittelt. Hierzu werden u.a. die relevanten Compliance-Quellen identifiziert und deren Umfang sowie potentielle Auswirkungen bestimmt, analysiert und bewertet (Giblin, Liu, Müller, Pfitzmann, & Zhou, 2005).
- Compliance formalization: Die zuvor ermittelten Compliance-Anforderungen besitzen häufig einen informalen und abstrakten Charakter und sind in der Regel in natürlicher Sprache formuliert. In dieser Phase werden die Compliance-Anforderungen mit Hilfe strukturierter, semi-formaler Notationen bzw. Schablonen präzisiert, ggf. formalisiert und schließlich in Form von Compliance-Regeln spezifiziert.
- Compliance implementation: In dieser Phase werden die auf den initial ermittelten Compliance-Anforderungen basierenden und im Anschluss spezifizierten Compliance-Regeln (technisch) implementiert.
- Compliance checking: Die Prüfung der zuvor implementierten Compliance-Regeln kann sowohl manuell als auch (semi-)automatisiert mit Hilfe von Informationssystemen erfolgen.
- Compliance optimization: In der letzten Phase werden die Ergebnisse der Compliance-Prüfung analysiert. In Abhängigkeit von den Compliance-Anforderungen können verschiedene Maßnahmen zur Verbesserung der Compliance-Aktivitäten bzw. der Geschäftsprozesse eines Unternehmens abgeleitet werden (Giblin, Liu, Müller, Pfitzmann, & Zhou, 2005).

Darüber hinaus gewinnt die kontinuierliche Überwachung des Compliance-Zustands eines Unternehmens (compliance monitoring) zunehmend an Bedeutung (Rodríguez, et al., 2013).

5.2 Compliance-Regeln

Die Präzisierung der häufig in natürlicher Sprache formulierten, informalen Compliance-Anforderungen stellt eine der größten Herausforderungen bei der Spezifikation von Compliance-Regeln dar (Ramezani, Fahland, van der Werf, & Mattheis, 2011; Ramezani, Fahland, & van der Aalst, 2013). In dieser Arbeit wird eine Compliance-Regel in Anlehnung an (COMPAS, 2009b) wie folgt definiert:

Definition 5.3: Eine Compliance-Regel (compliance rule, auch compliance constraint) ist eine operationelle Beschreibung einer Compliance-Anforderung.

Compliance-Regeln, die eine ähnliche Semantik besitzen, werden dabei häufig zu einem Compliance-Regelsatz (compliance rule set, auch compliance policy) zusammengefasst (Giblin, Liu, Müller, Pfitzmann, & Zhou, 2005; COMPAS, 2009b).

Zur Spezifikation der Compliance-Regeln kommen neben strukturierten, semi-formalen Notationen bzw. Schablonen auch formale Techniken zum Einsatz, die im Rahmen der Verifikation von Softwaresystemen eingesetzt werden (Ramezani, Fahland, van der Werf, & Mattheis, 2011). Dabei werden die erwarteten Eigenschaften des Systems zunächst mit Hilfe geeigneter, formaler Spezifikationssprachen (z.B. temporale Logiken oder reguläre Ausdrücken) präzisiert (Dwyer, Avrunin, & Corbett, 1998; Dwyer, Avrunin, & Corbett, 1999). Zur Prüfung der formal spezifizierten Eigenschaften kommen anschließend Werkzeuge zur Modellprüfung (model checker) zum Einsatz (Manna & Pnueli, 1992; Clarke, Grumberg, & Peled, 1999).

Die im Rahmen der formalen Spezifikation der erwarteten Eigenschaften auftretende und durch die verschiedenen Formalismen bedingte Komplexität stellt in der Praxis eine besondere Herausforderung dar. Um auch Anwendern ohne eine entsprechend umfassende Kenntnis formaler Spezifikationssprachen eine korrekte Beschreibung der erwarteten Eigenschaften zu ermöglichen, existieren verschiedene musterbasierte Ansätze, die die Erfahrung von Experten in einer für Anwender verständlichen, parametrisierbaren sowie von einem bestimmten Formalismus unabhängigen Form festhalten. Der musterbasierte Ansatz eignet sich dabei besonders zur Lösung von wiederholt auftretenden bzw. wiederkehrenden, domänenspezifischen Problemen bzw. Problemstellungen (Dwyer, Avrunin, & Corbett, 1998; Dwyer, Avrunin, & Corbett, 1999).

Aufbauend auf den Arbeiten von (Gamma, Helm, Johnson, & Vlissides, 1994) entwickeln (Dwyer, Avrunin, & Corbett, 1998) sogenannte Eigenschaftsspezifikationsmuster (property specification patterns), die eine „generalisierte Beschreibung häufig vorkommender Anforderungen an die erlaubten Zustände bzw. Ereignissequenzen eines endlichen Transitionssystems darstellen, somit die grundlegende Struktur bestimmter Aspekte eines Systemverhaltens abbilden und diese mit Hilfe eines allgemeingültigen Formalismus ausdrücken“ (Dwyer, Avrunin, & Corbett, 1999).

Die Eigenschaftsspezifikationsmuster umfassen verschiedene Attribute wie einen Namen, eine präzise Beschreibung der Struktur des erwarteten Verhaltens, eine Abbildung in bzw. Zuordnung zu einer oder mehreren Spezifikationssprachen, Beispiele zur Verwendung sowie Beziehungen zu anderen Mustern. Des Weiteren können zugehörige Anforderungen, verschiedene Parameter, die Quelle bzw. der Autor (des Musters), zugehörige Anwendungsdomänen sowie weitere, informale Notizen festgehalten werden. Darüber hinaus unterscheiden (Dwyer, Avrunin, & Corbett, 1999) verschiedene Geltungsbereiche (scopes) für die Eigenschaftsspezifikationsmuster. Der Umfang der Geltungsbereiche wird durch die Spezifikation eines Start- und eines Endzustands bzw. eines Start- und eines Endereignisses bestimmt.

Abbildung 5.2 veranschaulicht die von (Dwyer, Avrunin, & Corbett, 1999) entwickelten und basierend auf ihrer Semantik hierarchisch angeordneten Eigenschafts- spezifikationsmuster.

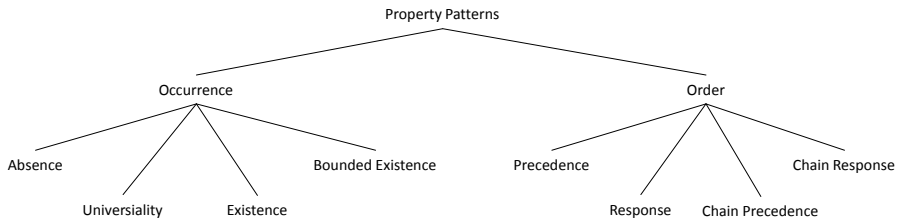

Abbildung 5.2: Hierarchie von Eigenschaftsspezifikationsmustern

Dabei werden die folgenden Eigenschaftsspezifikationsmuster unterschieden (Dwyer, Avrunin, & Corbett, 1999):

- Absence: Ein Zustand bzw. ein Ereignis darf nicht innerhalb eines Geltungsbereichs auftreten.
- Existence: Ein Zustand bzw. ein Ereignis muss innerhalb eines Geltungsbereichs auftreten.

- Bounded existance: Ein Zustand bzw. ein Ereignis muss k-mal innerhalb eines Geltungsbereichs auftreten. Varianten dieses Musters spezifizieren z.B. das mindestens oder maximal k-malige Auftreten eines Zustands bzw. Ereignisses.
- Universality: Ein Zustand bzw. ein Ereignis muss über den gesamten Geltungsbereich hinweg auftreten.
- Precedence: Ein Zustand bzw. ein Ereignis P muss innerhalb eines Geltungsbereichs immer einem Zustand bzw. ein Ereignis Q vorangehen.
- Response: Ein Zustand bzw. Ereignis P muss innerhalb eines Geltungsbereichs immer von einem Zustand bzw. Ereignis Q gefolgt werden.
- Chain Precendence: Einer Sequenz von Zuständen bzw. Ereignissen $P_1, \dots, P_n$ muss immer eine Sequenz von Zuständen bzw. Ereignissen $Q_1, \dots, Q_m$ vorangehen. Dieses Muster stellt eine Verallgemeinerung des Precedence-Musters dar.
- Chain Response: Eine Sequenz von Zuständen bzw. Ereignissen $P_1, \dots, P_n$ muss immer von einer Sequenz von Zuständen bzw. Ereignissen $Q_1, \dots, Q_m$ gefolgt werden. Dieses Muster stellt eine Verallgemeinerung des Response-Musters dar.

Um die zunehmende Zahl von Compliance-Anforderungen beherrschbar zu machen und eine automatisierte Compliance-Prüfung zu ermöglichen, werden in der Literatur verschiedene musterbasierte Ansätze vorgestellt, die im Rahmen der semiformalen und formalen Spezifikation von Compliance-Anforderungen für Geschäftsprozesse in Anlehnung an die Arbeiten von (Dwyer, Avrunin, & Corbett, 1999) häufig wiederkehrende bzw. wiederholt auftretende Muster (compliance patterns oder compliance rule patterns) identifizieren und in Form von parametrisierbaren Compliance-Regeln abbilden (COMPAS, 2008; COMPAS, 2009a; Papazoglou, 2011).

Tabelle 5.1 enthält eine Auswahl häufig zitierter Arbeiten, in denen verschiedene Compliance-Regeln für Geschäftsprozesse spezifiziert werden. Die Compliance-Regeln lassen sich in vier Kategorien einordnen (Becker, Delfmann, Eggert, & Schwittay, 2012; El Kharbili, 2012; Fellmann & Zasada, 2014):

- Kontrollfluss (control-flow): Diese Kategorie umfasst Compliance-Regeln, die sich auf die strukturelle Abfolge (order) sowie das Auftreten (occurrence) von Aktivitäten beziehen.
- Organisation (organizational): Diese Kategorie umfasst Compliance-Regeln, die sich auf organisatorische Rollen und Ressourcen (Personen, Systeme) beziehen.
- Zeit (time): Diese Kategorie umfasst Compliance-Regeln, die sich auf temporale Eigenschaften von Geschäftsprozessen und Aktivitäten beziehen.

- Daten (data bzw. information): Diese Kategorie umfasst Compliance-Regeln, die sich auf die im Rahmen von Geschäftsprozessen auszutauschenden Daten bzw. Informationen beziehen.

Tabelle 5.1: Compliance-Regeln für Geschäftsprozesse in der Literatur

Quelle	Kategorie			
	Kontrollfluss	Organisation	Zeit	Daten
(Awad, Decker, & Weske, 2008)	●	○	○	●
(Awad, Weidlich, & Weske, 2009)	●	○	●	●
(Awad, Goré, Thomson, & Weidlich, 2011)	●	●	○	○
(Crampton, 2004)	○	●	○	○
(Elgammal, Türetken, van den Heuvel, & Papazoglou, 2010a)	●	●	●	●
(Elgammal, Türetken, van den Heuvel, & Papazoglou, 2010b)	●	●	●	○
(Ghose & Koliadis, 2007)	●	●	○	●
(Governatori, Hoffmann, Sadiq, & Weber, 2008)	●	●	○	●
(Hoffmann, Weber, & Governatori, 2012)	●	●	○	●
(Knuplesch, Ly, Rinderle-Ma, Pfeifer, & Dadam, 2010)	●	○	○	●
(Kumar & Liu, 2008)	○	○	○	●
(Küster, Ryndina, & Gall, 2007)	●	○	●	●
(Lanz, Weber, & Reichert, 2014)	○	○	●	○
(Liu, Müller, & Xu, 2007)	●	○	●	●
(Lohmann, 2011)	●	○	○	●
(Lu, Sadiq, & Governatori, 2007)	●	●	○	●
(Ly, et al., 2010)	●	○	○	●
(Ly, Rinderle-Ma, & Dadam, 2010)	●	○	●	●
(Ly, Rinderle-Ma, Göser, & Dadam, 2012)	●	○	○	●
(Ly, Rinderle-Ma, Knuplesch, & Dadam, 2011)	●	○	○	○
(Namiri & Stojanovic, 2007a)	●	○	○	●
(Namiri & Stojanovic, 2007b)	●	○	○	●

(Ramezani, Fahland, van der Werf, & Mattheis, 2011)	●	○	○	○
(Ramezani, Fahland, van Dongen, & van der Aalst, 2013)	●	○	●	●
(Ramezani, Gromov, Fahland, & van der Aalst, 2014)	○	●	○	●
(Reichert & Weber, 2012)	●	○	○	○
(Sackmann & Kähmer, 2008)	●	○	○	●
(Sadiq, Governatori, & Namiri, 2007)	●	●	●	●
(Sadiq & Governatori, 2010)	○	○	○	●
(Schleicher, Anstett, Leymann, & Mietzner, 2009)	●	○	○	○
(Schumm, et al., 2010)	●	○	●	●
(Schumm, Leymann, & Streule, 2010)	●	○	○	○
(Weidlich, Polyvyanyy, Desai, & Mendling, 2010)	●	○	○	○

Im Folgenden werden Compliance-Regeln bzw. -Regelsätze mit Hilfe einer strukturierten, semiformalen Schablone (compliance rule template) spezifiziert. Die hierzu verwendete Schablone ist an die Arbeiten von (Dwyer, Avrunin, & Corbett, 1999) angelehnt (Tabelle 5.2).

Tabelle 5.2: Schablone zur Spezifikation von Compliance-Regeln

Kurzname:	Name:
Kategorie:	
Parameter:	
Beschreibung:	

Dabei werden die in der Literatur (Tabelle 5.1) für die Kategorien Kontrollfluss, Organisation und Zeit vorgestellten Compliance-Regeln konsolidiert, angepasst und neu entwickelte Compliance-Regeln bzw. -Regelsätze ergänzt. Aufgrund der hohen domänen-, unternehmens- sowie anwendungsspezifischen Wiederverwendbarkeit wird mit der Kategorie Kosten eine konkrete Ausprägung der Kategorie Daten betrachtet. In Anlehnung an die Arbeiten von (Wynn, et al., 2013; Wynn, Low, & Nauta, 2013; Wynn, et al., 2013; Wynn, Low, ter Hofstede, & Nauta, 2014) werden für die Kategorie Kosten verschiedene Compliance-Regeln bzw. -Regelsätze entwickelt.

Zur Parametrisierung der Compliance-Regeln können die folgenden Parameter verwendet werden (COMPAS, 2009a; Papazoglou, 2011; Ramezani, Fahland, & van der Aalst, 2012; Ramezani, Fahland, van Dongen, & van der Aalst, 2013):

- Geschäftsprozess P
- Aktivität A
- Rolle RO
- Ressource RE
- Zeitpunkt T
- Zeitdauer D
- Kosten C

Für jede Kategorie werden beispielhaft jeweils drei Regeln ausgewählt und mit Hilfe der in der Schablone festgelegten Attribute spezifiziert. Jede Compliance-Regel erhält dabei eine eindeutige Identifikation (Kurznamen), einen Namen, ihre Kategorie, die zulässigen Parameter sowie eine erläuternde Beschreibung. Die übrigen, gemäß der Schablone spezifizierten Compliance-Regeln, sind im Anhang A dieser Arbeit aufgeführt.

5.2.1 Kategorie Kontrollfluss

In Tabelle 5.3 werden Compliance-Regeln für die Kategorie Kontrollfluss zusammengefasst.

Tabelle 5.3: Compliance-Regeln der Kategorie Kontrollfluss

Kurzname	Compliance-Regel
CR-CF-0001	ActivityExistence
CR-CF-0002	ActivityAbsence
CR-CF-0003	ActivityExactly
CR-CF-0004	ActivityAtLeast
CR-CF-0005	ActivityAtMost
CR-CF-0006	ActivityExactlyRow
CR-CF-0007	ActivityAtLeastRow
CR-CF-0008	ActivityAtMostRow
CR-CF-0009	ActivityCoexistence
CR-CF-0010	ActivitySequenceExactly

CR-CF-0011	ActivitySequenceAtLeast
CR-CF-0012	ActivitySequenceAtMost
CR-CF-0013	ActivityPrecedenceDirect
CR-CF-0014	ActivityPrecedenceIndirect
CR-CF-0015	ActivityPrecedenceDirectMultiple
CR-CF-0016	ActivityPrecedenceIndirectMultiple
CR-CF-0017	ActivityPrecedenceDirectMultipleDifferent
CR-CF-0018	ActivityPrecedenceIndirectMultipleDifferent
CR-CF-0019	ActivityPrecedenceNeverDirect
CR-CF-0020	ActivityPrecedenceNever
CR-CF-0021	ActivitySequencePrecedenceDirect
CR-CF-0022	ActivitySequencePrecedenceIndirect
CR-CF-0023	ActivitySequencePrecedenceNeverDirect
CR-CF-0024	ActivitySequencePrecedenceNever
CR-CF-0025	ActivityResponseDirect
CR-CF-0026	ActivityResponseIndirect
CR-CF-0027	ActivityResponseDirectMultiple
CR-CF-0028	ActivityResponseIndirectMultiple
CR-CF-0029	ActivityResponseDirectMultipleDifferent
CR-CF-0030	ActivityResponseIndirectMultipleDifferent
CR-CF-0031	ActivityResponseNeverDirect
CR-CF-0032	ActivityResponseNever
CR-CF-0033	ActivitySequenceResponseDirect
CR-CF-0034	ActivitySequenceResponseIndirect
CR-CF-0035	ActivitySequenceResponseNeverDirect
CR-CF-0036	ActivitySequenceResponseNever
CR-CF-0037	ActivityExclusive
CR-CF-0038	ActivityMutualExclusive
CR-CF-0039	ActivityInclusive
CR-CF-0040	ActivityPrerequisite
CR-CF-0041	ActivitySubstitute
CR-CF-0042	ActivityCorequisite

In den folgenden Tabellen werden beispielhaft die Compliance-Regeln CR-CF-0001, CR-CF-0002 und CR-CF-0003 spezifiziert.

Tabelle 5.4: Compliance-Regel CR-CF-0001 – ActivityExistence

Kurzname: CR-CF-0001	Name: ActivityExistence
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a	
Beschreibung: Eine Aktivität a muss innerhalb eines Prozesses p (mindestens einmal) vorkommen. Die Compliance-Regel ist verletzt, wenn eine Aktivität a nicht innerhalb eines Prozesses p vorkommt.	

Tabelle 5.5: Compliance-Regel CR-CF-0002 – ActivityAbsence

Kurzname: CR-CF-0002	Name: ActivityAbsence
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a	
Beschreibung: Eine Aktivität a darf innerhalb eines Prozesses p nicht vorkommen. Die Compliance-Regel ist verletzt, wenn eine Aktivität a innerhalb eines Prozesses p vorkommt.	

Tabelle 5.6: Compliance-Regel CR-CF-0003 – ActivityExactly

Kurzname: CR-CF-0003	Name: ActivityExactly
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a, Ganzzahl i	
Beschreibung: Eine Aktivität a muss innerhalb eines Prozesses p genau i-mal vorkommen. Die Compliance-Regel ist verletzt, wenn die Aktivität a weniger oder mehr als i-mal innerhalb eines Prozesses p vorkommt.	

5.2.2 Kategorie Organisation

In Tabelle 5.7 sind Compliance-Regeln für die Kategorie Organisation zusammengefasst.

Tabelle 5.7: Compliance-Regeln der Kategorie Organisation

Kurzname	Compliance-Regel
CR-ORG-0001	ProcessRoleExistence
CR-ORG-0002	ProcessRoleAbsence
CR-ORG-0003	ProcessRoleExactly
CR-ORG-0004	ProcessRoleAtLeast
CR-ORG-0005	ProcessRoleAtMost
CR-ORG-0006	ProcessRoleCoexistence
CR-ORG-0007	ProcessRoleExclusive
CR-ORG-0008	ProcessRoleMutualExclusive
CR-ORG-0009	ProcessRoleInclusive
CR-ORG-0010	ProcessRolePrerequisite
CR-ORG-0011	ProcessRoleSubstitute
CR-ORG-0012	ProcessRoleCorequisite
CR-ORG-0013	ProcessResourceExistence
CR-ORG-0014	ProcessResourceAbsence
CR-ORG-0015	ProcessResourceExactly
CR-ORG-0016	ProcessResourceAtLeast
CR-ORG-0017	ProcessResourceAtMost
CR-ORG-0018	ProcessResourceCoexistence
CR-ORG-0019	ProcessResourceExclusive
CR-ORG-0020	ProcessResourceMutualExclusive
CR-ORG-0021	ProcessResourceInclusive
CR-ORG-0022	ProcessResourcePrerequisite
CR-ORG-0023	ProcessResourceSubstitute
CR-ORG-0024	ProcessResourceCorequisite
CR-ORG-0025	ActivityRoleBonded
CR-ORG-0026	ActivityRoleSegregated
CR-ORG-0027	ActivityResourceBonded
CR-ORG-0028	ActivityResourceSegregated
CR-ORG-0029	ActivityRoleBondedResourceBonded
CR-ORG-0030	ActivityRoleBondedResourceSegregated
CR-ORG-0031	ActivityRoleSegregatedResourceSegregated

In den folgenden Tabellen werden beispielhaft die Compliance-Regeln CR-ORG-0001, CR-ORG-0002 und CR-ORG-0003 spezifiziert.

Tabelle 5.8: Compliance-Regel CR-ORG-0001 – ProcessRoleExistence

Kurzname: CR-ORG-0001	Name: ProcessRoleExistence
Kategorie: Organisation	
Parameter: Prozess p, Rolle ro	
Beschreibung: Eine Rolle ro muss innerhalb eines Prozesses p (mindestens einmal) vorkommen. Die Compliance-Regel ist verletzt, wenn die Rolle ro nicht innerhalb eines Prozesses p vorkommt.	

Tabelle 5.9: Compliance-Regel CR-ORG-0002 – ProcessRoleAbsence

Kurzname: CR-ORG-0002	Name: ProcessRoleAbsence
Kategorie: Organisation	
Parameter: Prozess p, Rolle ro	
Beschreibung: Eine Rolle ro darf innerhalb eines Prozesses p nicht vorkommen. Die Compliance-Regel ist verletzt, wenn eine Rolle ro innerhalb eines Prozesses p vorkommt.	

Tabelle 5.10: Compliance-Regel CR-ORG-0003 – ProcessRoleExactly

Kurzname: CR-ORG-0003	Name: ProcessRoleExactly
Kategorie: Organisation	
Parameter: Prozess p, Rolle ro, Ganzzahl i	
Beschreibung: Eine Rolle ro muss innerhalb eines Prozesses p genau i-mal vorkommen. Die Compliance-Regel ist verletzt, wenn die Rolle ro weniger oder mehr als i-mal innerhalb eines Prozesses p vorkommt.	

5.2.3 Kategorie Zeit

In Tabelle 5.11 sind Compliance-Regeln für die Kategorie Zeit zusammengefasst.

Tabelle 5.11: Compliance-Regeln der Kategorie Zeit

Kurzname	Compliance-Regel
CR-TIME-0001	ProcessLeadTimeEqual
CR-TIME-0002	ProcessLeadTimeLess
CR-TIME-0003	ProcessLeadTimeGreater
CR-TIME-0004	ProcessLeadTimeLessOrEqual
CR-TIME-0005	ProcessLeadTimeGreaterOrEqual
CR-TIME-0006	ActivityStartExactly
CR-TIME-0007	ActivityStartBefore
CR-TIME-0008	ActivityStartAfter
CR-TIME-0009	ActivityFinishExactly
CR-TIME-0010	ActivityFinishBefore
CR-TIME-0011	ActivityFinishAfter
CR-TIME-0012	ActivityStartBeforeFinishBefore
CR-TIME-0013	ActivityStartBeforeFinishAfter
CR-TIME-0014	ActivityStartAfterFinishBefore
CR-TIME-0015	ActivityStartAfterFinishAfter
CR-TIME-0016	ActivityServiceTimeEqual
CR-TIME-0017	ActivityServiceTimeLess
CR-TIME-0018	ActivityServiceTimeGreater
CR-TIME-0019	ActivityServiceTimeLessOrEqual
CR-TIME-0020	ActivitySerivceTimeGreaterOrEqual

In den folgenden Tabellen werden beispielhaft die Compliance-Regeln CR-TIME-0001, CR-TIME-0002 und CR-TIME-0003 spezifiziert.

Tabelle 5.12: Compliance-Regel CR-TIME-0001 – ProcessLeadTimeEqual

Kurzname: CR-TIME-0001	Name: ProcessLeadTimeEqual
Kategorie: Zeit	
Parameter: Prozess p, Zeitdauer d	
Beschreibung: Die Durchlaufzeit eines Prozesses p muss genau d Zeiteinheiten betragen. Die Compliance-Regel ist verletzt, wenn die Durchlaufzeit des Prozesses p weniger oder mehr als d Zeiteinheiten beträgt.	

Tabelle 5.13: Compliance-Regel CR-TIME-0002 – ProcessLeadTimeLess

Kurzname: CR-TIME-0002	Name: ProcessLeadTimeLess
Kategorie: Zeit	
Parameter: Prozess p, Zeitdauer d	
Beschreibung: Die Durchlaufzeit eines Prozesses p muss weniger als d Zeiteinheiten betragen. Die Compliance-Regel ist verletzt, wenn die Durchlaufzeit des Prozesses p genau oder mehr als d Zeiteinheiten beträgt.	

Tabelle 5.14: Compliance-Regel CR-TIME-0003 – ProcessLeadTimeGreater

Kurzname: CR-TIME-0003	Name: ProcessLeadTimeGreater
Kategorie: Zeit	
Parameter: Prozess p, Zeitdauer d	
Beschreibung: Die Durchlaufzeit eines Prozesses p muss mehr als d Zeiteinheiten betragen. Die Compliance-Regel ist verletzt, wenn die Durchlaufzeit des Prozesses p genau oder weniger als d Zeiteinheiten beträgt.	

5.2.4 Kategorie Kosten

In Tabelle 5.15 sind Compliance-Regeln für die Kategorie Kosten zusammengefasst.

Tabelle 5.15: Compliance-Regeln der Kategorie Kosten

Kurzname	Compliance-Regel
CR-COST-0001	ProcessCostEqual
CR-COST-0002	ProcessCostLess
CR-COST-0003	ProcessCostGreater
CR-COST-0004	ProcessCostLessOrEqual
CR-COST-0005	ProcessCostGreaterOrEqual
CR-COST-0006	ProcessResourceCostEqual
CR-COST-0007	ProcessResourceCostLess
CR-COST-0008	ProcessResourceCostGreater
CR-COST-0009	ProcessResourceCostLessOrEqual
CR-COST-0010	ProcessResourceCostGreaterOrEqual
CR-COST-0011	ActivityCostEqual
CR-COST-0012	ActivityCostLess
CR-COST-0013	ActivityCostGreater
CR-COST-0014	ActivityCostLessOrEqual
CR-COST-0015	ActivityCostGreaterOrEqual
CR-COST-0016	ActivityResourceCostEqual
CR-COST-0017	ActivityResourceCostLess
CR-COST-0018	ActivityResourceCostGreater
CR-COST-0019	ActivityResourceCostLessOrEqual
CR-COST-0020	ActivityResourceCostGreaterOrEqual

In den folgenden Tabellen werden beispielhaft die Compliance-Regeln CR-COST-0001, CR-COST-0002 und CR-COST-0003 spezifiziert.

Tabelle 5.16: Compliance-Regel CR-COST-0001 – ProcessCostEqual

Kurzname: CR-COST-0001	Name: ProcessCostEqual
Kategorie: Kosten	
Parameter: Prozess p, Kosten c	
Beschreibung: Die Kosten eines Prozesses p müssen gleich c sein. Die Compliance-Regel ist verletzt, wenn die Kosten eines Prozesses p kleiner oder größer als c sind.	

Tabelle 5.17: Compliance-Regel CR-COST-0002 – ProcessCostLess

Kurzname: CR-COST-0002	Name: ProcessCostLess
Kategorie: Kosten	
Parameter: Prozess p, Kosten c	
Beschreibung: Die Kosten eines Prozesses p müssen kleiner als c sein. Die Compliance-Regel ist verletzt, wenn die Kosten eines Prozesses p gleich oder größer als c sind.	

Tabelle 5.18: Compliance-Regel CR-COST-0003 – ProcessCostGreater

Kurzname: CR-COST-0003	Name: ProcessCostGreater
Kategorie: Kosten	
Parameter: Prozess p, Kosten c	
Beschreibung: Die Kosten eines Prozesses p müssen größer als c sein. Die Compliance-Regel ist verletzt, wenn die Kosten eines Prozesses p gleich oder kleiner als c sind.	

In Abhängigkeit von konkreten Compliance-Anforderungen können auf Basis des präsentierten Ansatzes zusätzliche Kategorien sowie neue Compliance-Regeln oder Varianten bereits bestehender Compliance-Regeln ergänzt werden.

5.3 Konzeptuelles Modell

In der Literatur werden verschiedene Ansätze für konzeptuelle Modelle für GRC bzw. das Compliance Management diskutiert. Dabei werden in der Regel auf Basis empirischer Untersuchungen zunächst die wesentlichen, im Zusammenhang mit dem Compliance Management verwendeten Begriffe identifiziert und konsolidiert. Anschließend werden strukturelle Zusammenhänge abgeleitet und darauf aufbauend verschiedene Modellelemente bzw. Informationsobjekte konstruiert (Vicente & Mira da Silva, 2011; Schultz, 2013; Nissen & Marefika, 2014). Abhängig vom Betrachtungsschwerpunkt (z.B. strategisch, operational, technisch) können die konzeptuellen Modelle weiter in geeignete Teilmodelle untergliedert werden (Nissen & Marefika, 2013; Nissen & Marefika, 2014). Tabelle 5.19 enthält eine Zusammenfassung der auf Basis empirischer Untersuchungen in der Literatur identifizierten Begriffe.

Tabelle 5.19: Wesentliche Begriffe im Compliance Management

Quelle	Compliance-*					
	Quelle	Anforderung	Regel	Kontrolle	Risiko	Audit
(Namiri & Stojanovic, 2007a)	○	●	○	●	●	○
(Namiri & Stojanovic, 2007b)	○	○	○	●	●	○
(Karagiannis, Mylopoulos, & Schwab, 2007)	○	○	○	●	●	○
(Karagiannis, 2008)	○	○	○	●	●	○
(Sadiq & Governatori, 2010)	○	●	○	●	○	○
(Lu, Sadiq, & Governatori, 2007)	○	●	○	●	○	○
(Nissen & Marefika, 2014)	●	●	○	●	●	●
(Papazoglou, 2011)	●	●	●	●	●	○
(Rosemann & zur Muehlen, 2005)	○	○	○	○	●	○
(Schultz, 2013)	●	●	○	●	●	●
(Schumm, et al., 2010)	●	●	●	●	●	○
(Strecker, Heise, & Frank, 2011)	●	●	○	●	●	●
(Türetken, Elgammal, van den Heuvel, & Papazoglou, 2011)	●	●	●	●	●	○
(Türetken, Elgammal, van den Heuvel, & Papazoglou, 2012)	●	●	●	●	●	○
(Vicente & Mira da Silva, 2011)	○	○	○	●	●	●

Aufbauend auf den in Tabelle 5.19 aufgeführten Arbeiten wird in einem ersten Schritt ein konzeptuelles Modell – das Compliance Management Model (CMM) – konstruiert (Abbildung 5.3):

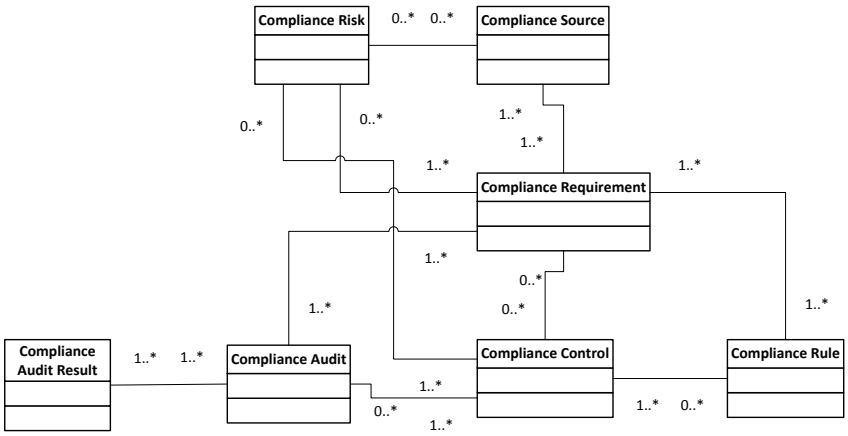

Abbildung 5.3: Compliance Management Model

Compliance-Kontrollen (compliance control) verbinden die zuvor vorgestellten Compliance-Anforderungen und Compliance-Regeln und sollen als Teil der internen Kontrollen eines Unternehmens die Einhaltung der Compliance-Anforderungen sicherstellen (Schultz, 2013). Compliance-Kontrollen können dabei manuell durchgeführt werden oder z.B. mit Hilfe von Informationssystemen eine (semi-)automatisierte Prüfung von Compliance-Regeln ermöglichen (Schultz, 2013). Aus der Verletzung von Compliance-Anforderungen ergeben sich wiederum Compliance-Risiken (compliance risk), die spezielle Risiken eines Unternehmens darstellen. Im Rahmen von Compliance-Audits (compliance audit, auch compliance assessment) werden die Compliance-Kontrollen eines Unternehmens und damit die Erfüllung bzw. Verletzung von Compliance-Anforderungen überprüft. Die Ergebnisse der Compliance-Audits (compliance audit results) werden im Anschluss in Form von Audit-Berichten (compliance audit reports) dokumentiert und (ggf. visuell) aufbereitet (El Kharbili, 2012). Compliance-Audits können sowohl unternehmensintern oder -extern durchgeführt werden als auch prozessbasiert oder -unabhängig erfolgen (Schultz, 2013).

In einem zweiten Schritt werden aus dem Geschäftsprozessmanagement bekannte Elemente in das zuvor entwickelte CMM integriert. Es resultiert ein erweitertes, konzeptuelles Gesamtmodell – das Business Process Compliance Management Model (BPCMM) (Abbildung 5.4).

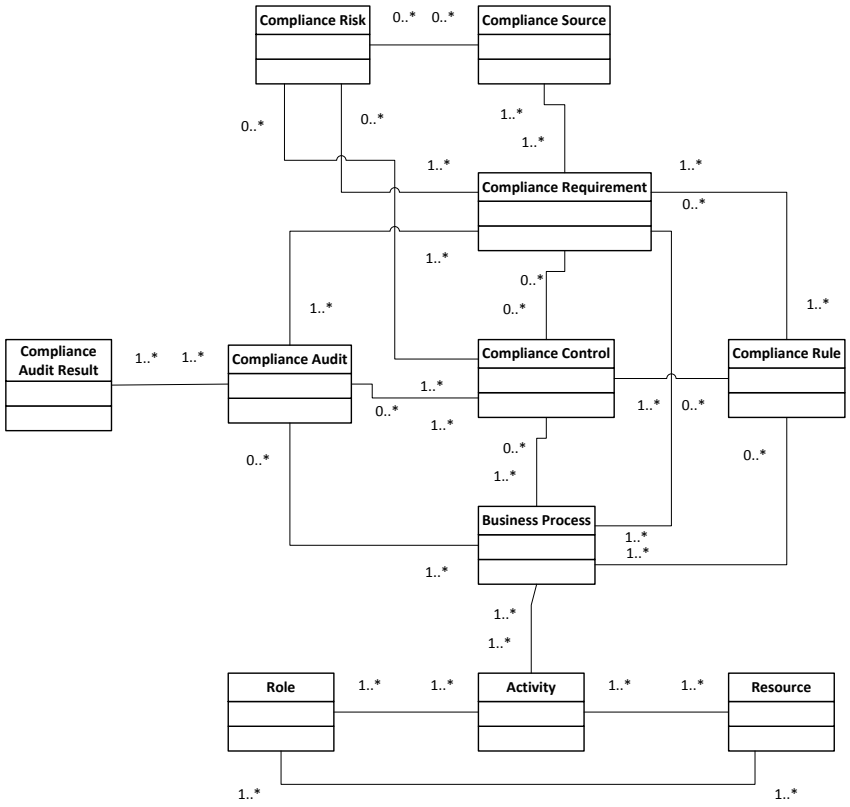

Abbildung 5.4: Business Process Compliance Management Model

Neben einer begrifflichen Einordnung soll das entwickelte BPCMM zu einer Verbesserung der Transparenz im Rahmen einer integrierten Betrachtung des Compliance Managements für Geschäftsprozesse beitragen (Schultz, 2013). Darüber hinaus legt es die Basis für eine Umsetzung bzw. Implementierung einer informationstechnologischen Unterstützung des Compliance Managements in der Praxis (Nissen & Marefika, 2014).

6 Ereignisprotokollbasierte Compliance-Prüfung von Geschäftsprozessen

In diesem Kapitel wird aufbauend auf einer Literaturrechere eine Einordung verschiedener in der Literatur diskutierter Arbeiten für die Compliance-Prüfung von Geschäftsprozessen vorgenommen, und es werden ausgewählte Ansätze präsentiert. Im Anschluss wird auf die Ergeignisprotokollierung im Kontext von Geschäftsprozessen eingegangen, und es wird eine formale Definition wesentlicher Begriffe gegeben. Des Weiteren wird mit dem eXtensible Event Stream ein Standard für die Ereignisprotokollierung vorgestellt. Abschließend wird ein Ansatz zur ereignisprotokollbasierten Compliance-Prüfung mit XQuery hergeleitet, und es wird eine beispielhafte Implementierung der im vorangegangenen Kapitel beschriebenen Compliance-Regeln vorgestellt.

6.1 Verwandte Arbeiten

In der Literatur beschäftigt sich eine Vielzahl von Arbeiten mit der Compliance-Prüfung von Geschäftsprozessen, die in Ansätze zur Vorwärtsprüfung (forward compliance checking, a-priori, by design) sowie Rückwärtsprüfung (backward compliance checking, a-posteriori, ex-post, by detection) unterschieden werden können (Sackmann, 2008; Becker, Delfmann, Eggert, & Schwittay, 2012; Accorsi, Damiani, & van der Aalst, 2013).

Alternativ kann eine Einordnung der Ansätze in die Kategorien Design Time Compliance (zur Entwurfszeit, vor der Ausführung, pre-execution), Run Time Compliance (zur Laufzeit, während der Ausführung, execution) sowie Post-Execution Compliance (nachgelagert, nach der Ausführung, post-execution) vorgenommen werden (El Kharbili, 2012; Fellmann & Zasada, 2014).

Tabelle 6.1 enthält eine Übersicht über verwandte Arbeiten und ordnet diese anhand der genannten Kategorien ein (Becker, Delfmann, Eggert, & Schwittay, 2012; El Kharbili, 2012; Fellmann & Zasada, 2014).

Tabelle 6.1: Business Process Compliance und Conformance – Verwandte Arbeiten

Quelle	Forward Compliance Checking		Backward Compliance Checking
	Design-Time Compliance	Run-Time Compliance	Post-Execution Compliance
(Awad, 2010)	●	○	○
(El Kharbili & Pulvermüller, 2009)	●	○	○
(Elgammal, Türetken, van den Heuvel, & Papazoglou, 2010b)	●	○	○
(Ghose & Koliadis, 2007)	●	○	○
(Giblin, Liu, Müller, Pfitzmann, & Zhou, 2005)	●	○	○
(Goedertier & Vanthienen, 2006b)	●	○	○
(Governatori & Milosevic, 2005)	●	○	○
(Liu, Müller, & Xu, 2007)	●	○	○
(Ly, Rinderle-Ma, Göser, & Dadam, 2012)	●	○	○
(Montali, 2010)	○	○	●
(Mulo, Zdun, & Dustdar, 2013)	○	●	○
(Namiri, 2008)	●	○	○
(Pesic, 2008)	○	●	○
(Ramezani, Fahland, van Dongen, & van der Aalst, 2013)	○	○	●
(van der Aalst, de Beer, & van Dongen, 2005)	○	○	●
(Weber, 2009)	●	○	○

In der ersten Kategorie Forward Compliance Checking/Design Time Compliance werden Ansätze für die Vorwärtsprüfung von Geschäftsprozessen bzw. Geschäftsprozessmodellen zusammengefasst (Becker, Delfmann, Eggert, & Schwittay, 2012; Hashmi & Governatori, 2013).

PENELOPE (Process Entailment from Elicitation of Obligations and Permissions) ist eine die deklarative Sprache, mit der verschiedene aus Geschäftsregeln stammende Beschränkungen festgehalten und auf Basis von BPMN-Modellen zur Entwurfszeit geprüft werden können (Goedertier & Vanthienen, 2006a; Goedertier & Vanthienen, 2006b).

Die Process Compliance Language (PCL) ist eine auf deontischer Logik basierende Sprache zur Modellierung und Konstruktion verschiedener normativer Anforderungen, sogenannter PCL specifications, die zur Annotation von Geschäftsprozessmodellen (insbesondere BPMN-Modellen) eingesetzt werden können (Governatori & Milosevic, 2005). Im Anschluss kann geprüft werden, ob ein Ausführungspfad eines Geschäftsprozessmodells zu den annotierten Spezifikationen konsistent ist (Governatori & Sadiq, 2008; Sadiq & Governatori, 2010; Governatori & Rotolo, 2010a; Governatori & Rotolo, 2010b).

Die Business Process Modelling Notation-Query (BPMN-Q) ermöglicht eine abfragebasierte Prüfung von Compliance-Regeln auf Basis visueller Muster (visual patterns). Die an die BPMN angelehnten, visuellen Muster werden zunächst in Formeln der Linearen Temporalen Logik (LTL) übersetzt und zusammen mit einem zuvor generierten Zustandsraum eines Geschäftsprozessmodells an ein Werkzeug zur Modellprüfung übergeben (Awad, Decker, & Weske, 2008; Awad, Weidlich, & Weske, 2009; Awad & Weidlich, 2009; Awad, 2010; Awad, Goré, Thomson, & Weidlich, 2011; Awad, Weidlich, & Weske, 2011). Ein ähnlicher Ansatz zur strukturbasierten Verifikation von BPMN-Modellen wird von (Müller, 2010) vorgestellt.

Das Software-Werkzeug SeaFlows stellt verschiedene Funktionalitäten für die Verifikation von Geschäftsprozessmodellen auf Basis von Compliance-Regeln bereit. Die Modellierung der Compliance-Regeln erfolgt mit Hilfe eines graphbasierten Formalismus, sogenannter Compliance Rule Graphs (CRG). Das zugrunde liegende Rahmenwerk verfügt darüber hinaus über zwei komplementäre Ansätze zur Compliance-Prüfung (Knuplesch, Ly, Rinderle-Ma, Pfeifer, & Dadam, 2010; Ly, Rinderle-Ma, & Dadam, 2010; Ly, Rinderle-Ma, Knuplesch, & Dadam, 2011; Ly, Rinderle-Ma, Göser, & Dadam, 2012).

Die im Rahmen des EU-Projekts COMPAS entwickelte Compliance Request Language (CRL) ist eine Sprache zur abstrakten, musterbasierten Spezifikation von Compliance-Anforderungen. Die Compliance-Anforderungen werden zunächst unter Verwendung von Compliance-Mustern (compliance patterns) grafisch modelliert. Im Anschluss erfolgt eine Transformation der als CRL-Ausdrücke vorliegenden Compliance-Muster in formale Compliance-Regeln bzw. LTL-Formeln. Abschließend können die Compli-

ance-Regeln zusammen mit einer formalen Geschäftsprozessspezifikation an ein Werkzeug zur Modellprüfung übergeben werden (COMPAS, 2008; COMPAS, 2009a; COMPAS, 2009b; Elgammal, Türetken, van den Heuvel, & Papazoglou, 2010a; Elgammal, Türetken, van den Heuvel, & Papazoglou, 2010b; Papazoglou, 2011).

In der zweiten Kategorie Run Time Compliance werden Ansätze zusammengefasst, die während der Geschäftsprozessausführung, d.h. zur Laufzeit von Geschäftprozessen zum Einsatz kommen. Neben Ansätzen, die eine Ausführung bzw. Prüfung deklarativer Modelle ermöglichen (z.B. Declare, ehemals ConDec oder MoBuCon) (Pesic & van der Aalst, 2006; van der Aalst & Pesic, 2006; Pesic, Schonenberg, Sidorova, & van der Aalst, 2007; Pesic, 2008; Maggi, Montali, Westergaard, & van der Aalst, 2011) werden verschiedene auf das Complex Event Processing (CEP) aufbauende Ansätze diskutiert (Thullner, Rozsnyai, Schiefer, Obweger, & Suntinger, 2011; Weidlich, et al., 2011; Mulo, Zdun, & Dustdar, 2013). Die Verarbeitung der im Rahmen der Prozessausführung auftretenden Ereignisse erfolgt in der Regel mit Hilfe einer Event Processing Engine (z.B. ESPER). Für die Compliance-Prüfung wird häufig eine zugehörige Abfragesprache (z.B. Event Processing Language, EPL) eingesetzt (Mulo, Zdun, & Dustdar, 2013).

In der dritten Kategorie Backward Compliance Checking/Post-Execution Compliance werden Ansätze für eine Rückwärtsprüfung von Geschäftsprozessen zusammengefasst.

Hierbei kommen häufig Verfahren bzw. eine Kombination von Verfahren des Process Mining, insbesondere der Übereinstimmungsprüfung (conformance checking) sowie zugehörige Software-Werkzeuge (z.B. ProM) zum Einsatz (Rozinat & van der Aalst, 2008; IEEE, 2010; van der Aalst, 2011).

In (van der Aalst, de Beer, & van Dongen, 2005) wird eine LTL-basierte Sprache vorgestellt, mit der verschiedene temporale Eigenschaften formuliert und auf Basis von Ereignisprotokollen geprüft werden können. Darüber hinaus wird ein zugehöriges Plugin (LTL Checker) für das Software-Werkzeug ProM zur Verfügung gestellt.

Mit der im Rahmenwerk CLIMB (Computational Logic for the verIfication and Modeling of Business constraints) enthaltenen Sprache SCIFF sowie dem zugehörigen, ebenfalls für das Software-Werkzeug ProM verfügbaren Plugin (SCIFF Checker) exisiert ein weiterer Ansatz zur nachgelagerten Prüfung von Ereignisprotokollen (Montali, 2010; Montali, et al., 2010).

Die genannten Ansätze und zugehörigen Implementierungen bzw. Software-Werkzeuge (LTL Checker, SCIFF Checker, ProM) werden in verschiedenen Arbeiten zur Evalua-

tion sowie im Rahmen von Anwendungsbeispielen eingesetzt (Accorsi & Stocker, 2012; Stocker & Accorsi, 2013; Stocker, Accorsi, & Rother, 2013; Schultz, Müller-Wickop, Werner, & Nüttgens, 2013; Holderer, 2014).

In (Ramezani, Fahland, van Dongen, & van der Aalst, 2013) wird ein auf Petri-Netzen basierender Ansatz zur Compliance-Prüfung vorgestellt. Hierzu werden verschiedene Compliance-Regeln identifiziert und als parametrisierte Petri-Netz-Muster (parameterized Petri-net patterns) formalisiert. Im Anschluss können die in Ereignisprotokollen beschriebene Abläufe mit den zuvor parametrisierten Petri-Netz-Mustern abgeglichen werden. Eine frühe Implementierung des Ansatzes wird in Form zweier Plugins (Paket compliance) für das Software-Werkzeug ProM bereitgestellt. Ähnliche Petri-Netz-basierte Ansätze werden in den Arbeiten von (Accorsi, Lowis, & Sato, 2011; Lohmann, 2011; Accorsi & Lehmann, 2012) präsentiert.

Alle Ansätze der dritten Kategorie haben dabei gemein, dass die zu prüfenden Compliance-Regeln zunächst formal spezifiert (LTL Checker, SCIFF Checker) bzw. als Petri-Netz beschrieben werden müssen, bervor sie im Anschluss mit einem Software-Werkzeug geprüft werden können. Zwar werden einige, häufig verwendete Compliance-Regeln in bereits spezifizierter bzw. beschriebener Form vorgehalten, eine Anpassung vorhandener bzw. die Entwicklung neuer Regeln setzt jedoch die Kenntnis formaler Spezifikationssprachen bzw. des Petri-Netz-basierten Ansatzes voraus. Des Weiteren verfügen die vorgestellten Ansätze häufig nur über rudimentäre Funktionalitäten zur Verwaltung von Compliance-Anforderungen, -Kontrollen und/oder -Regeln; eine Möglichkeit zur Beschreibung von Beziehungen und Abhängigkeiten wird von keinem der vorgestellten Ansätze bzw. Software-Werkzeuge angeboten. Darüber hinaus stellen die Ansätze nur zum Teil Funktionalitäten zur Aufbereitung, Darstellung sowie einer visuellen Analyse der Prüfergebnisse zur Verfügung; hier fehlt eine flexible sowie auf offenen Standards basierende Lösung. Zudem erfolgt bisher keine Integration in ein Geschäftsprozessmanagement und -modellierungswerkzeug.

6.2 Geschäftsprozesse und Ereignisprotokollierung

Prozessgestützte Informationssysteme (process-aware information systems, PAIS) werden zunehmend für die Ausführung und Steuerung operationeller Geschäftsprozesse eingesetzt. Typische Beispiele für prozessgestützte Informationssysteme sind Workflow Management Systeme (WfMS) oder Business Process Management Systeme (BPMS) (van der Aalst, 2009; Reichert & Weber, 2012).

Das Aufzeichnen bzw. Speichern der in diesen Systemen anfallenden Ereignisdaten (event data) erfolgt in der Regel in sogenannten Ereignisprotokollen (auch Ereignislogs, event logs, process logs), die einen zentralen Ausgangspunkt sowie eine Basis für die nachträgliche Analyse der Geschäftsprozesse darstellen. In den meisten Unternehmen kommen zudem weitere Informationssysteme zum Einsatz, die auf klassischen relationalen Datenbanken bzw. Datenbanksystemen oder neuen NoSQL-Technolgien basieren (van der Aalst, 2015). Eine wesentliche Herausforderung besteht dabei in der Konvertierung der häufig in unterschiedlichen Tabellen vorliegenden und über verschiedene Schlüsselbeziehungen miteinander verbundenen Ereignisdaten (van der Aalst, 2015). Um eine Konvertierung der häufig in unterschiedlichen Tabellen vorliegenden und über verschiedene Schlüsselbeziehungen miteinander verbundenen Ereignisdaten zu erleichtern, wird in (Westergaard & van Dongen, 2013) ein generischer Ansatz zur Transformation unterschiedlicher Ausgangsformate in ein standardisiertes Format für Ereignisprotokolle auf der Basis von Schlüssel-Wert-Mengen (KeyValueSets) präsentiert.

Wil van der Aalst präsentiert zwölf grundlegende Leitlinien für die Protokollierung von Ereignisdaten (event logging). In diesem Kontext werden Ereignisse zunächst als „Dinge, die passieren“ definiert, die wiederum mit verschiedenen Referenzen und Attributen beschrieben werden können. Referenzen dienen dabei der Identifizierung bestimmter Objekte (z.B. Aktivitäten, Rollen, Ressourcen), denen mit Hilfe von Attributen ein Wert zugewiesen werden kann (van der Aalst, 2015):

- Referenz- und Attributnamen sollten eine eindeutige Semantik besitzen, d.h. sie sollten für alle involvierten Personen, die Ereignisdaten erzeugen und analysieren, dieselbe Bedeutung haben und von unterschiedlichen Interessensgruppen in gleicher Weise interpretiert werden können.
- Es sollte eine strukturierte und verwaltete Menge von Referenz- und Attributnamen geben, d.h. Referenz- und Attributnamen sollten ähnlich einer Taxonomie oder Ontologie hierarchisch gruppiert bzw. zusammengefasst werden. Darüber hinaus sollte die Möglichkeit bestehen, domänen- oder unternehmensspezifische Erweiterungen vornehmen zu können.
- Referenzen sollten stabil sein, d.h. eindeutige Bezeichner sollten nicht wiederverwendet werden oder kontextabhängig sein.
- Attributwerte sollten so präzise wie möglich sein. Besitzt ein Attributwert nicht die gewünschte Genauigkeit, sollte dies explizit angegeben werden.
- Unsicherheiten bezüglich des Auftretens von Ereignissen sowie von Referenzen oder Attributen sollten ebenfalls angegeben werden.
- Ereignisse sollten entweder explitzit (z.B. in einer Liste) oder implizit anhand eines Attributs (z.B. eines Zeitstempels) geordnet sein.

- Zu den Ereignissen sollten weitere Transaktionsinformationen gespeichert werden, so dass z.B. eine eindeutige Zuordnung von Ereignissen zu Aktivtäten bzw. Aktiviätsinstanzen möglich ist.
- Es sollten regelmäßige Konsistenz- und Korrektheitsprüfungen durchgeführt werden, um z.B. die syntaktische Korrektheit der Ereignisprotokolle zu prüfen sowie fehlende Referenzen und Attribute erkennen, korrigieren oder ergänzen zu können.
- Um eine Vergleichbarkeit von Ereignisprotokollen sicherzustellen, sollten die zugrunde liegenden Protokollierungsprinzipien nicht bzw. nicht ohne eine entsprechende Kenntlichmachung verändert werden.
- Ereignisse sollten in einer möglichst ursprünglichen sowie feingranularen Form gespeichert werden, d.h. es sollten keine aggregierten Ereignisse in einem Ereignisprotokoll gespeichert werden bzw. enhalten sein. Eine Aggregation sollte erst im Rahmen eines anschließenden Analyseprozesses erfolgen.
- Ereignisprotokolle sollten reproduzierbar sein, d.h. es sollten keine Ereignisse (nachträglich) aus einem Ereignisprotokoll entfernt werden.
- Sensible oder private Daten sollten so früh wie möglich (d.h. vor einer Analyse) anonymisiert oder aus dem Ereignisprotokoll entfernt werden. Dabei sollten jedoch keine ggf. noch benötigten Informationen über Beziehungen, Abhängigkeiten oder Korrelationen von Ereignissen verloren gehen.

Ereignisprotokolldaten (event log data) beschreiben die zeitliche Reihenfolge der im Rahmen der Ausführung von Geschäftsprozessen auftretenden Ereignisse. Jedes Ereignis ist einer Aktivität (actvitiy, auch Prozessschritt) und einem spezifischen Fall (case, auch Prozessinstanz) zugeordnet. Darüber hinaus können Ereignisprotokolldaten zusätzliche Informationen zu den Ereignissen enthalten (IEEE, 2010; van der Aalst, 2011; van der Aalst, 2015).

Definition 6.1: Sei $\mathcal{E}$ die Menge aller möglichen Ereignisse. Ereignisse können durch verschiedene Attribute charakterisiert werden. Sei AN eine Menge von Attributnamen. Für jedes Ereignis $e \in \mathcal{E}$ und Attributnamen $n \in AN$ gilt:

1. $\#_n(e)$ ist der Wert des Attributs n für das Ereignis e.
2. Besitzt das Ereignis e kein Attribut mit dem Namen n, dann ist $\#_n(e) = \perp$ (Nullwert) (van der Aalst, 2011).

In der Regel werden Ereignisse anhand des Namens ihres zugeordneten Aktivitätsattributs identifiziert. Wird in einem Ereignisprotokoll jedoch zusätzlich der Transaktionstyp (z.B. start oder complete) von Ereignissen aufgezeichnet bzw. gespeichert, können

mehrere Ereignisse einer Aktivität zugeordnet werden. Dieses Korrelationsproblem kann durch das Hinzufügen zusätzlicher Informationen gelöst werden, z.B. indem ein zusätzliches Attribut für die Aktivitätsinstanz hinzugefügt wird (van der Aalst, 2011).

Definition 6.2: Sei $\mathcal{C}$ die Menge aller möglichen Prozessinstanzen. Prozessinstanzen können wie Ereignisse Attribute besitzen. Für jede Prozessinstanz $c \in \mathcal{C}$ und Attributnamen $n \in AN$ gilt:

1. $\#_n(c)$ ist der Wert des Attributs n für die Prozessinstanz c. Besitzt die Prozessinstanz c kein Attribut mit dem Namen n, dann ist $\#_n(c) = \perp$ (Nullwert).
2. Ein Ablauf ist eine endliche Abfolge von Ereignissen $\sigma \in \mathcal{E}$. Jede Prozessinstanz besitzt ein spezielles, verpflichtendes Attribut $\#_{\text{trace}}(c) \in \mathcal{E}$, das mindestens ein Ereignis enthält: $\#_{\text{trace}}(c) \neq \langle\,\rangle$.
3. Ein Ereignisprotokoll ist wiederum eine Menge von Prozessinstanzen, d.h. $L \in \mathcal{C}$ (van der Aalst, 2011).

Werden in einem Ereignisprotokoll Zeitstempel aufgezeichnet bzw. gespeichert, sollten die in einem Ablauf enthaltenen Ereignisse aufsteigend geordnet sein (van der Aalst, 2011).

6.3 eXtensible Event Stream

eXtensible Event Stream (XES) ist ein im Jahr 2010 von der IEEE Task Force on Process Mining ausgewählter, XML-basierter Standard für Ereignisprotokolle. XES wurde als Nachfolger des ebenfalls XML-basierten Mining XML (MXML) mit dem Ziel entwickelt, ein allgemein anerkanntes und erweiterbares Format für Ereignisprotokolldaten bereitzustellen und damit den Austausch zwischen verschiedenen Anwendungsdomänen und Werkzeugen zu erleichtern. Sowohl dem Entwurf als auch der Entwicklung des XES-Standards lagen die folgenden vier Leitlinien zugrunde (IEEE, 2010; Günther & Verbeek, 2014):

- Einfachheit (simplicity): Sowohl die Repräsentation von Informationen in als auch die Erstellung und Analyse von Ereignisprotokollen soll so einfach wie möglich sein. Ereignisprotokolldaten sollen nicht nur von Maschinen interpretiert und verarbeitet, sondern auch von Menschen gelesen und verstanden werden können.
- Flexibilität (flexiblity): Der Standard soll unabhängig von einer konkreten Anwendungsdomäne oder IT-Unterstützung zur Aufzeichnung bzw. Speicherung und Analyse prozessbasierter Ereignisdaten verwendet werden können.

- Erweiterbarkeit (extensibility): Eine zukünftige Erweiterung des Standards soll so einfach und transparent wie möglich erfolgen können. Zudem soll stets eine Rückwärts- und Vorwärtskompatibilität gewährleistet sein. Darüber hinaus soll es möglich sein, den Standard für spezifische Anwendungsdomänen und Werkzeugimplementierungen anpassen zu können.
- Expressivität (expressivity): Bei der Serialisierung von Ereignisdaten sollen so wenige Informationen wie möglich verloren gehen. Hierzu ist eine strenge Typisierung notwendig. Darüber hinaus soll es eine Methode geben, um (einzelnen) Elementen eine für Menschen lesbare Semantik hinzuzufügen.

Als generisches Austauschformat definiert der XES-Standard nur die Elemente, die in nahezu jeder Umgebung vorkommen bzw. für die entsprechend viele Anwendungsfälle identifiziert werden können. Werden weitere, über den XES-Standard hinausgehende Informationen benötigt, können diese mit Hilfe von Erweiterungen (extensions) als (optionale) Attribute standardisiert werden (Verbeek, Buijs, van Dongen, & van der Aalst, 2010; Günther & Verbeek, 2014).

6.3.1 XES-Metamodell

Im Folgenden werden das XES-Metamodell (Abbildung 6.1) sowie die wesentlichen Komponenten und Konzepte des XES-Standards vorgestellt (Verbeek, Buijs, van Dongen, & van der Aalst, 2010; Günther & Verbeek, 2014).

Auf der obersten Ebene eines XES-Dokuments befindet sich das log-Element, das alle zu einem Prozess gehörenden Ereignisdaten enthält und folgende Pflichtattribute besitzt (Tabelle 6.2) (Verbeek, Buijs, van Dongen, & van der Aalst, 2010; Günther & Verbeek, 2014):

Tabelle 6.2: Attribute des log-Elements

Attribut	Attributtyp	Beschreibung
xes.version	xs:decimal	Das Attribut enthält die Version des XES-Standards, zu der das XES-Dokument konform ist
xes.features	xs:token	Das Attribut enthält eine durch Leerzeichen getrennte Liste optionaler Merkmale, (z.B. „nested-attributes“). Werden keine optionalen Merkmale verwendet, muss das Attribut leer sein

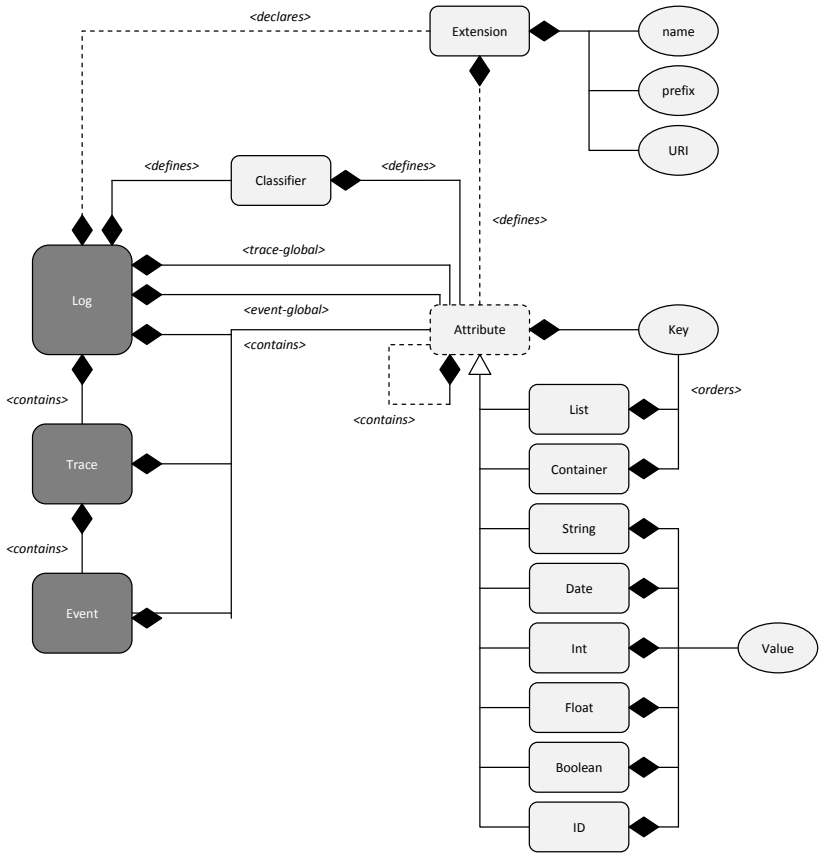

Abbildung 6.1: XES-Metamodell

Das log-Element enthält eine beliebige Anzahl von trace-Elementen, die den Ablauf einer bestimmten Prozessinstanz repräsentieren. Trace-Elemente besitzen keine Attribute (Verbeek, Buijs, van Dongen, & van der Aalst, 2010; Günther & Verbeek, 2014).

Jedes trace-Element enthält wiederum eine beliebige Anzahl von event-Elementen. Diese repräsentieren Ereignisse (Aktivitäten) die während der Prozessausführung auftreten bzw. beobachtet werden können. Event-Elemente besitzen ebenfalls keine Attribute (Verbeek, Buijs, van Dongen, & van der Aalst, 2010; Günther & Verbeek, 2014).

Die log-, trace- und event-Elemente enthalten keine Informationen, sondern definieren lediglich die Struktur eines XES-Dokuments. Die eigentlichen Informationen werden in sogenannten Attributelementen festgehalten. Alle Attributelemente besitzen einen string-

basierten Schlüssel (key), für den der XES-Standard die folgenden Eigenschaften definiert (Verbeek, Buijs, van Dongen, & van der Aalst, 2010; Günther & Verbeek, 2014):

- Schlüssel dürfen keine Zeilenumbrüche oder Tabulatoren enhalten; führende bzw. nachfolgende Leerzeichen sind erlaubt.
- Schlüssel dürfen nur einmal innerhalb ihres umschließenden Containers vorkommen, d.h. es darf nur ein Attribut mit einem bestimmten Schlüssel geben. Einzige Ausnahme stellen dabei die innerhalb einer Liste bzw. eines Listenattributs auftretenden Schlüssel dar.

Die log-, trace- und event-Elemente können eine beliebige Anzahl von Attributelementen enthalten. Der XES-Standard definiert die folgenden sechs Attributelemente (Tabelle 6.3) (Günther & Verbeek, 2014). Neben diesen sechs Attributelementen definiert der XES-Standard zwei zusätzliche spezielle Attributelemente (collections), mit denen verschiedene Attributelemente zusammengefasst werden können (Tabelle 6.4) (Günther & Verbeek, 2014).

Tabelle 6.3: Elementare XES-Attribute

Attribut	Attributtyp	Beschreibung
string	xs:string	String-Attributelemente enthalten untypisierte Informationen beliebiger Länge
date	xs:dateTime	Date-Attributelemente enthalten Informationen über einen spezifischen Zeitpunkt
int	xs:long	Int-Attributelemente enthalten eine Ganzzahl (mit 64-Bit Genauigkeit)
float	xs:double	Float-Attributelemente enthalten eine Fließkommazahl (mit 64-Bit doppelter Genauigkeit)
boolean	xs:boolean	Boolean-Attributelemente enthalten einen Wahrheitswert (true, false)
id	xs:string	Id-Attributelemente enthalten einen Universally Unique Identifier (UUID)

Tabelle 6.4: Spezielle XES-Attribute

Attribut	Attributtyp	Beschreibung
list	*	List-Attributelemente enthalten eine beliebige Anzahl geordneter Kindelemente
container	*	Container-Attributelemente enthalten eine beliebige Anzahl ungeordneter Kindelemente

Um eine maximale Flexibilität bei der Aufzeichnung bzw. Speicherung von Ereignisdaten zu ermöglichen, erlaubt der XES-Standard die Verwendung verschachtelter Attributelemente (nested attributes). Dieses Merkmal wird z.B. für list- und container-Attribute benötigt (Günther & Verbeek, 2014).

Des Weiteren enthält das log-Element zwei Listen globaler Attribute für die trace- und die event-Ebene, mit denen das Vorhandensein sowie die korrekte Definition von Attributelementen für das gesamte XES-Dokument festgelegt werden kann (Verbeek, Buijs, van Dongen, & van der Aalst, 2010; Günther & Verbeek, 2014).

6.3.2 Standard-Erweiterungen

Mit dem Konzept der Erweiterungen verfügt der XES-Standard über eine Möglichkeit, auf jeder Ebene eines XES-Dokuments (log, trace, event und meta für verschachtelte Attribute) verschiedene Attribute festzulegen, die für eine bestimmte Anwendungsdomäne oder zur Unterstützung einer bestimmten Analyseanwendung erforderlich sind bzw. benötigt werden. Diesen Attributen kann darüber hinaus eine grundlegende Semantik zugewiesen werden (Verbeek, Buijs, van Dongen, & van der Aalst, 2010; Günther & Verbeek, 2014).

Erweiterungen werden innerhalb des log-Elements deklariert und enthalten neben dem Namen der Erweiterung einen zugehörigen Präfix, der allen in der Erweiterung definierten Attributen vorangestellt wird, sowie eine zugehörige URI, die auf eine im XML-basierten XESEXT-Format vorliegende Beschreibung der Erweiterung verweist und Informationen über die in der Erweiterung definierten Attributelemente enthält (Günther & Verbeek, 2014).

Der XES-Standard enthält die folgenden Standard-Erweiterungen (Verbeek, Buijs, van Dongen, & van der Aalst, 2010; Günther & Verbeek, 2014):

Die Concept Extension (Präfix: `concept`, URI: `http://www.xes-standard.org/concept.xesext`) definiert zwei Attributelemente: ein name-Attribut, das in Abhängigkeit von der Ebene (log, trace, event) einen Namen oder eine ID eines Prozesses, einer Prozessinstanz oder einer Aktivität enthält, sowie ein instance-Attribut auf event-Ebene, das eine ID für eine Aktivitätsinstanz enthält (Tabelle 6.5) (Verbeek, Buijs, van Dongen, & van der Aalst, 2010; Günther & Verbeek, 2014).

Tabelle 6.5: XES – Concept Extension

Attributebene	Attributschlüssel	Attributtyp	Beschreibung
log, trace, event	name	string	Name oder ID eines Prozesses, einer Prozessinstanz oder einer Aktivität
event	instance	string	ID einer Aktivitätsinstanz

Die Lifecycle Extension (Präfix: `lifecycle`, URI: `http://www.xes-standard.org/lifecycle.xesext`) definiert ebenfalls zwei Attributelemente: ein model-Attribut auf log-Ebene, das sich auf das verwendete Transaktionsmodell bezieht, sowie ein transition-Attribut auf event-Ebene, mit dem der Lebenszyklusübergang einer Aktivität beschrieben wird (Tabelle 6.6) (Verbeek, Buijs, van Dongen, & van der Aalst, 2010; Günther & Verbeek, 2014).

Tabelle 6.6: XES – Lifecycle Extension

Attributebene	Attributschlüssel	Attributtyp	Beschreibung
log	model	string	Verwendetes Transaktionsmodell (z.B. standard)
event	transition	string	Lebenszyklusübergang einer Aktivität (z.B. start oder complete)

Es können beliebige Transaktionsmodelle verwendet werden; der XES-Standard enthält folgendes Standard-Transaktionsmodell (Abbildung 6.2) (Günther & Verbeek, 2014):

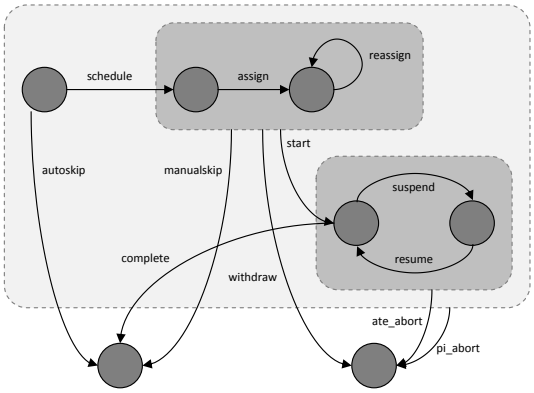

Abbildung 6.2: XES – Standard-Transaktionsmodell

Die Organizational Extension (Präfix: `org`, URI: `http://www.xes-standard.org/org.xesext`) definiert drei Attributelemente auf event-Ebene: die resource-, role- und group-Attribute enthalten sowohl den Namen einer Ressource, die ein Event ausgelöst bzw. eine Aktivität ausgeführt hat, als auch die zugehörige Rolle bzw. Gruppe innerhalb einer organisatorischen Struktur (Tabelle 6.7) (Verbeek, Buijs, van Dongen, & van der Aalst, 2010; Günther & Verbeek, 2014).

Tabelle 6.7: XES – Organizational Extension

Attributebene	Attributschlüssel	Attributtyp	Beschreibung
event	resource	string	Name der Ressource
event	role	string	Rolle der Ressource
event	group	string	Gruppe der Ressource

Die Time Extension (Präfix: `time`, URI: `http://www.xes-standard.org/time.xesext`) definiert ein Attributelement auf event-Ebene: das timestamp-Attribut enthält das exakte Datum bzw. den exakten Zeitpunkt, zu dem das Ereignis aufgetreten ist (Tabelle 6.8) (Verbeek, Buijs, van Dongen, & van der Aalst, 2010; Günther & Verbeek, 2014).

Tabelle 6.8: XES – Time Extension

Attributebene	Attributschlüssel	Attributtyp	Beschreibung
event	timestamp	date	Datum bzw. Zeitpunkt eines Ereignisses

Die Cost Extension (Präfix: `cost`, URI: `http://www.xes-standard.org/cost.xesext`) definiert fünf Attributelemente: das total- und das currency-Attribut auf trace- und event-Ebene, die sowohl Informationen zu den Kosten einer Prozessinstanz als auch den Gesamtkosten eines Ereignisses bzw. einer Aktivität enthalten. Die auf einer meta-Ebene definierten amount-, driver- und type-Attribute werden jeweils innerhalb eines Kostenattributs zusammengefasst und ordnen diesem jeweils einen bestimmten (Teil-) Betrag, einen Kostentreiber sowie einen Kostentyp zu (Tabelle 6.9) (Verbeek, Buijs, van Dongen, & van der Aalst, 2010; Günther & Verbeek, 2014).

Tabelle 6.9: XES – Cost Extension

Attributebene	Attributschlüssel	Attributtyp	Beschreibung
trace, event	total	float	Kosten einer Prozessinstanz; Gesamtkosten eines Ereignisses bzw. einer Aktivität
trace, event	currency	string	Währung bzw. zulässiges Währungsformat
meta	amount	float	(Teil-)Betrag
meta	driver	string	Kostentreiber (z.B. die ID eines zugehörigen Kostenattributs)
meta	type	string	Kostentyp

Darüber hinaus definiert der XES-Standard eine an SA-MXML (Semantically Annotated Mining XML) angelehnte Semantic Extension, die die Referenzierung verschiedener in Form von Ontologien vorliegenden Modellierungskonzepten erlaubt, sowie eine ID Extension, die eine Speicherung von UUIDs ermöglicht (Verbeek, Buijs, van Dongen, & van der Aalst, 2010; Günther & Verbeek, 2014).

6.4 Ereignisprotokollbasierte Compliance-Prüfung mit XQuery

Die meisten Ansätze zur Compliance-Prüfung von Geschäftsprozessen setzen die Kenntnis von formalen Spezifikationssprachen (insbesondere LTL) sowie eine entsprechende Erfahrung bei der Anwendung entsprechender Formalismen voraus (Accorsi, Damiani, & van der Aalst, 2013).

Ausgehend von in natürlicher Sprache vorliegenden und formulierten Compliance-Anforderungen werden zunächst Compliance-Regeln formal spezifiziert. Die formalisierten Compliance-Anforderungen können im Anschluss sowohl für eine Vorwärts- als auch eine Rückwärtsprüfung verwendet werden. Dabei liegen viele für Compliance-Fragestellungen interessante, relevante sowie erforderliche Daten und Informationen frühestens zur Laufzeit bzw. erst nach der Prozessausführung, z.B. in Form von Ereignisprotokollen, vor.

Im Gegensatz zu bereits bestehenden Ansätzen zur Compliance-Prüfung von Geschäftsprozessen abstrahiert der im Rahmen dieser Arbeit entwickelte Ansatz von der Verwendung komplexer formaler Spezifikationssprachen zur Beschreibung von Compliance-

Anforderungen bzw. -Regeln sowie dem anschließenden Einsatz von Software-Werkzeugen zur Modellprüfung.

Grundlage sowie Ausgangspunkt ist eine Arbeit von Sylvain Hallé und Roger Villemaire, in der eine XML-basierte Methode zur Validierung temporaler Eigenschaften auf Basis von Nachrichtenabläufen (message trace) vorgestellt wird (Hallé & Villemaire, 2008b). Hierzu werden zunächst verschiedene temporale Eigenschaften als LTL-Formeln spezifiziert und im Anschluss in XQuery – einer vom W3C-Konsortium standardisierten Abfragesprache für XML-Dokumente (W3C, 2014) – übersetzt bzw. überführt. Im Rahmen einer experimentellen Evaluation am Beispiel von Web Service Choreographien wird abschließend gezeigt, dass nicht nur eine Übersetzung bzw. Überführung von LTL-Formeln in XQuery-Ausdrücke sondern auch eine effiziente Validierung der XQuery-Ausdrücke unter Verwendung eines quelloffenen XQuery-Prozessors (Saxon) möglich ist (Hallé, Villemaire, Cherkaoui, & Ghandour, 2007; Hallé, Villemaire, & Cherkaoui, 2009).

In der Folge werden verschiedene darauf aufbauende Ansätze zur Laufzeit-Überwachung (runtime monitoring) von Web Service Choreographien sowie nachrichtenbasierter Workflows vorgestellt und diskutiert (Hallé & Villemaire, 2008a; Hallé & Villemaire, 2009; Hallé, Bultan, Hughes, Alkhalaf, & Villemaire, 2010; Sebahi & Hacid, 2010; Hallé & Villemaire, 2012). Darüber hinaus wird die Leistungsfähigkeit bzw. Performance verschiedener (relationaler) Datenbanken zur Validierung temporaler Eigenschaften für sogenannte Ereignisabläufe (event traces) untersucht (Vallet, Mrad, Hallé, & Beaudet, 2013).

Der Schwerpunkt des im Rahmen dieser Arbeit entwickelten Ansatzes liegt auf einer nachgelagerten, ereignisprotokollbasierten Compliance-Prüfung von Geschäftsprozessen bzw. Geschäftsprozessinstanzen. Aufbauend auf den Arbeiten von Sylvain Hallé und Roger Villemaire werden die im vorangegangen Kapitel 5 semiformal spezifizierten Compliance-Regeln in XQuery-Ausdrücke zur Abfrage von auf dem XES-Standard basierenden Ereignisprotokollen überführt. Hierbei werden die folgenden Annahmen getroffen bzw. es gelten die folgenden Voraussetzungen:

- Abläufe (d.h. Prozessinstanzen) enthalten eine endliche Menge an Ereignissen (closed world assumption).
- Alle für die Compliance-Prüfung relevanten Daten und Informationen sind vorhanden und werden im Ereignisprotokoll aufgezeichnet bzw. gespeichert.
- Die Compliance-Regeln sind in sich konsistent, d.h. widerspruchsfrei und nicht redundant.

Zur Sicherstellung korrekter Ergebnisse der Compliance-Prüfung werden für die in XQuery-Ausdrücke überführten Compliance-Regeln jeweils positive und negative Testfälle entwickelt und Bedingungsüberdeckungstests durchgeführt. Die positiven und negativen Testfälle werden in Form entsprechender Ereignisprotokolle umgesetzt.

Die folgenden fünf Standard-Erweiterungen werden von dem im Rahmen dieser Arbeit entwickelten Ansatz unterstützt und sind gemäß dem XES-Standard im Ereignisprotokoll zu deklarieren (Tabelle 6.10).

Tabelle 6.10: Deklaration von Standard-Erweiterungen im Ereignisprotokoll

```
...
<extension name="Concept" prefix="concept" uri="http://www.xes-
standard.org/concept.xesext"/>
<extension name="Lifecycle" prefix="lifecycle" uri="http://www.xes-
standard.org/lifecycle.xesext"/>
<extension name="Organizational" prefix="org" uri="http://www.xes-
standard.org/org.xesext"/>
<extension name="Time" prefix="time" uri="http://www.xes-
standard.org/time.xesext"/>
<extension name="Cost" prefix="cost" uri="http://www.xes-
standard.org/cost.xesext"/>
...
```

In Abhängigkeit von der Kategorie einer Compliance-Regel bestehen verschiedene Mindestanforderungen an die auf dem XES-Standard basierenden Ereignisprotokolle bzw. Ereignisprotokolldaten. So wird, um eine korrekte Funktion der XQuery-Ausdrücke gewährleisten zu können, das Vorhandensein verschiedener Attribute vorausgesetzt. Die für die Compliance-Prüfung erforderlichen Attribute werden als globale Attribute am Anfang des Ereignisprotokolls deklariert (Tabelle 6.11).

Tabelle 6.11: Beispielhafte Deklaration globaler Attribute im Ereignisprotokoll

```
...
<global scope="trace">
  <string key="concept:name" value="0"/>
</global>
<global scope="event">
  <string key="concept:instance" value=""/>
  <string key="concept:name" value="__INVALID__"/>
  <string key="lifecycle:transition" value="complete"/>
  ...
</global>
...
```

Analog zu Kapitel 5.2 werden im Folgenden für jede Kategorie beispielhaft drei Compliance-Regeln ausgewählt und eine zugehörige Implementierung mit XQuery vorgestellt. Darüber hinaus werden die an die Compliance-Regeln bzw. die XQuery-Ausdrücke zu übergebenden Parameter aufgeführt.

6.4.1 Kategorie Kontrollfluss

Für Compliance-Regeln der Kategorie Kontrollfluss müssen folgende Attribute bzw. Ereignisdaten vorhanden und gemäß folgender Standard-Erweiterungen im Ereignisprotokoll festgehalten sein:

- Concept Extension
- Lifecycle Extension

Dabei werden im Rahmen des in dieser Arbeit entwickelten Ansatzes nur abgeschlossene Aktivitäten (`lifecycle:transition="complete"`) betrachtet.

In den folgenden Tabellen sind die XQuery-Ausdrücke für die Compliance-Regeln CR-CF-0001, CR-CF-0002 und CR-CF-0003 aufgeführt.

Tabelle 6.12: Implementierung der Compliance-Regel CR-CF-0001 mit XQuery

Kurzname: XQ-CR-CF-0001	Name: ActivityExistence
Kategorie: Kontrollfluss	
Parameter: Event-Log `e` als `document-node()`, Aktivität `a1` als `xs:string`	

XQuery-Ausdruck:
`let $r:=(for $n at $i in $e//trace[position()=(1 to last())]`
`let $ac:=$n/event[string/@key="lifecycle:transition" and string/@value="complete"]`
`return if (exists($ac[string/@key="concept:name" and string/@value=$a1]))`
`  then (<trace id="{$i}" value="true"/>)`
`  else (<trace id="{$i}" value="false"/>) )`
`return <result rule="ActivityExistence">{$r}</result>`	

Tabelle 6.13: Implementierung der Compliance-Regel CR-CF-0002 mit XQuery

Kurzname: XQ-CR-CF-0002	Name: ActivityAbsence
Kategorie: Kontrollfluss	
Parameter: Event-Log `e` als `document-node()`, Aktivität `a1` als `xs:string`	

XQuery-Ausdruck:
`let $r:=(for $n at $i in $e//trace[position()=(1 to last())]`
`let $ac:=$n/event[string/@key="lifecycle:transition" and string/@value="complete"]`
`return if (exists($ac[string/@key="concept:name" and string/@value=$a1]))`
`  then (<trace id="{$i}" value="false"/>)`
`  else (<trace id="{$i}" value="true"/>) )`
`return <result rule="ActivityAbsence">{$r}</result>`	

Tabelle 6.14: Implementierung der Compliance-Regel CR-CF-0003 mit XQuery

Kurzname: XQ-CR-CF-0003	Name: ActivityExactly
Kategorie: Kontrollfluss	
Parameter: Event-Log `e` als `document-node()`, Aktivität `a1` als `xs:string`, Ganzzahl `i1` als `xs:integer`	

XQuery-Ausdruck:
`let $r:=(for $n at $i in $e//trace[position()=(1 to last())]`
`let $ac:=$n/event[string/@key="lifecycle:transition" and string/@value="complete"]`
`return if (count($ac[string/@key="concept:name" and string/@value=$a1])=$i1)`
`then (<trace id="{$i}" value="true"/>)`
`else (<trace id="{$i}" value="false"/>) )`
`return <result rule="ActivityExactly">{$r}</result>`	

6.4.2 Kategorie Organisation

Für Compliance-Regeln der Kategorie Organisation müssen folgende Attribute bzw. Ereignisdaten vorhanden und gemäß folgender Standard-Erweiterungen im Ereignisprotokoll festgehalten sein:

- Concept Extension
- Lifecycle Extension
- Org Extension

Dabei wird im Rahmen des in dieser Arbeit entwickelten Ansatzes die Annahme getroffen, dass nur in abgeschlossenen Aktivitäten (`lifecycle:transition="complete"`) Informationen zu auslösenden bzw. ausführenden Rollen und Ressourcen festgehalten werden.

In den folgenden Tabellen sind die XQuery-Ausdrücke für die Compliance-Regeln CR-ORG-0001, CR-ORG-0002 und CR-ORG-0003 aufgeführt.

Tabelle 6.15: Implementierung der Compliance-Regel CR-ORG-0001 mit XQuery

Kurzname: XQ-CR-ORG-0001	Name: ProcessRoleExistence
Kategorie: Organisation	
Parameter: Event-Log `e` als `document-node()`, Rolle `ro1` als `xs:string`	

XQuery-Ausdruck:
`let $r:=(for $n at $i in $e//trace[position()=(1 to last())]`
`let $ac:=$n/event[string/@key="lifecycle:transition" and string/@value="complete"]`
`return if (exists($ac[string/@key="org:role" and string/@value=$ro1]))`
`then (<trace id="{$i}" value="true"/>)`
`else (<trace id="{$i}" value="false"/>) )`
`return <result rule="ProcessRoleExistence">{$r}</result>`	

Tabelle 6.16: Implementierung der Compliance-Regel CR-ORG-0002 mit XQuery

Kurzname: XQ-CR-ORG-0002	Name: ProcessRoleAbsence
Kategorie: Organisation	
Parameter: Event-Log `e` als `document-node()`, Rolle `ro1` als `xs:string`	

XQuery-Ausdruck:
`let $r:=(for $n at $i in $e//trace[position()=(1 to last())]`
`let $ac:=$n/event[string/@key="lifecycle:transition" and string/@value="complete"]`
`return if (exists($ac[string/@key="org:role" and string/@value=$ro1]))`
`then (<trace id="{$i}" value ="false"/>)`
`else (<trace id="{$i}" value ="true"/>) )`
`return <result rule="ProcessRoleAbsence">{$r}</result>`	

Tabelle 6.17: Implementierung der Compliance-Regel CR-ORG-0003 mit XQuery

Kurzname: XQ-CR-ORG-0003	Name: ProcessRoleExactly
Kategorie: Organisation	
Parameter: Event-Log `e` als `document-node()`, Rolle `ro1` als `xs:string`, Ganzzahl `i1` als `xs:integer`	

XQuery-Ausdruck:
`let $r:=(for $n at $i in $e//trace[position()=(1 to last())]`
`let $ac:=$n/event[string/@key="lifecycle:transition" and string/@value="complete"]`
`return if (count($ac[string/@key="org:role" and string/@value=$ro1])=$i1)`
`then (<trace id="{$i}" value="true"/>)`
`else (<trace id="{$i}" value="false"/>) )`
`return <result rule="ProcessRoleExactly">{$r}</result>`	

6.4.3 Kategorie Zeit

Für Compliance-Regeln der Kategorie Zeit müssen folgende Attribute bzw. Ereignisdaten vorhanden und gemäß folgender Standard-Erweiterungen im Ereignisprotokoll festgehalten sein:

- Concept Extension
- Lifecylce Extension
- Time Extension

Dabei wird im Rahmen des in dieser Arbeit entwickelten Ansatzes die Annahme getroffen, dass mindestens zum Beginn und Ende einer Aktivität (`lifecycle:transition="start" und "complete"`) Zeitstempel festgehalten werden.

In den folgenden Tabellen sind die XQuery-Ausdrücke für die Compliance-Regeln CR-TIME-0001, CR-TIME-0002 und CR-TIME-0003 aufgeführt.

Tabelle 6.18: Implementierung der Compliance-Regel CR-TIME-0001 mit XQuery

Kurzname: XQ-CR-TIME-0001	Name: ProcessLeadTimeEqual
Kategorie: Zeit	
Parameter: Event-Log `e` als `document-node()`, Zeitdauer `d1` als `xs:dayTimeDuration`	

XQuery-Ausdruck:
`let $r:=(for $n at $i in $e//trace[position()=(1 to last())]`
`let $as:=$n/event[string/@key="lifecycle:transition" and @value="start"]`
`let $ac:=$n/event[string/@key="lifecycle:transition" and string/@value="complete"]`
`return if (xs:dateTime($ac[position()=last()]/date[@key="time:timestamp"]/@value/data())-xs:dateTime($as[position()=1]/date[@key="time:timestamp"]/@value/data())=$d1)`
`then (<trace id="{$i}" value="true"/>)`
`else (<trace id="{$i}" value="false"/>) )`
`return <result rule="ProcessLeadTimeEqual">{$r}</result>`	

Tabelle 6.19: Implementierung der Compliance-Regel CR-TIME-0002 mit XQuery

Kurzname: XQ-CR-TIME-0002	Name: ProcessLeadTimeLess
Kategorie: Zeit	
Parameter: Event-Log `e` als `document-node()`, Zeitdauer `d1` als `xs:dayTimeDuration`	

XQuery-Ausdruck:
`let $r:=(for $n at $i in $e//trace[position()=(1 to last())]`
`let $as:=$n/event[string/@key="lifecycle:transition" and @value="start"]`
`let $ac:=$n/event[string/@key="lifecycle:transition" and string/@value="complete"]`
`return if (xs:dateTime($ac[position()=last()]/date[@key="time:timestamp"]/@value/data())-xs:dateTime($as[position()=1]/date[@key="time:timestamp"]/@value/data())<$d1)`
`then (<trace id="{$i}" value="true"/>)`
`else (<trace id="{$i}" value="false"/>) )`
`return <result rule="ProcessLeadTimeLess">{$r}</result>`	

Tabelle 6.20: Implementierung der Compliance-Regel CR-TIME-0003 mit XQuery

Kurzname: XQ-CR-TIME-0003	Name: ProcessLeadTimeGreater
Kategorie: Zeit	
Parameter: Event-Log `e` als `document-node()`, Zeitdauer `d1` als `xs:dayTimeDuration`	

XQuery-Ausdruck:
`let $r:=(for $n at $i in $e//trace[position()=(1 to last())]`
`let $as:=$n/event[string/@key="lifecycle:transition" and @value="start"]`
`let $ac:=$n/event[string/@key="lifecycle:transition" and string/@value="complete"]`
`return if (xs:dateTime($ac[position()=last()]/date[@key="time:timestamp"]/@value/data())-xs:dateTime($as[position()=1]/date[@key="time:timestamp"]/@value/data())>$d1)`
`then (<trace id="{$i}" value="true"/>)`
`else (<trace id="{$i}" value="false"/>) )`
`return <result rule="ProcessLeadTimeGreater">{$r}</result>`	

6.4.4 Kategorie Kosten

Für Compliance-Regeln der Kategorie Kosten müssen folgende Attribute bzw. Ereignisdaten vorhanden und gemäß folgender Standard-Erweiterungen im Ereignisprotokoll festgehalten sein:

- Concept Extension
- Lifecylce Extension
- Org Extension
- Cost Extension

Dabei wird im Rahmen des in dieser Arbeit entwickelten Ansatzes die Annahme getroffen, dass mindestens die initialen Kosten für Prozessinstanzen (`cost:total`) sowie die Kosten (`cost:total`) der Aktivitäten (`lifecycle:transition="complete"`) festgehalten werden.

In den folgenden Tabellen sind die XQuery-Ausdrücke für die Compliance-Regeln CR-COST-0001, CR-COST-0002 und CR-COST-0003 aufgeführt.

Tabelle 6.21: Implementierung der Compliance-Regel CR-COST-0001 mit XQuery

Kurzname: XQ-CR-COST-0001	Name: ProcessCostEqual
Kategorie: Kosten	
Parameter: Event-Log `e` als `document-node()`, Kosten `c1` als `xs:float`	

XQuery-Ausdruck:
`let $r:=(for $n at $i in $e//trace[position()=(1 to last())]`
`let $ac:=$n/event[string/@key="lifecycle:transition" and string/@value="complete"]`
`return if ((sum($ac[position()=(1 to last())]/float[@key="cost:total"]/@value/data())+$n/float[@key="cost:total"]/@value/data())=$c1)`
`then (<trace id="{$i}" value="true"/>)`
`else (<trace id="{$i}" value="false"/>) )`
`return <result rule="ProcessCostEqual">{$r}</result>`	

Tabelle 6.22: Implementierung der Compliance-Regel CR-COST-0002 mit XQuery

Kurzname: XQ-CR-COST-0002	Name: ProcessCostLess
Kategorie: Kosten	
Parameter: Event-Log `e` als `document-node()`, Kosten `c1` als `xs:float`	

XQuery-Ausdruck:
`let $r:=(for $n at $i in $e//trace[position()=(1 to last())]`
`let $ac:=$n/event[string/@key="lifecycle:transition" and string/@value="complete"]`
`return if ((sum($ac[position()=(1 to last())]/float[@key="cost:total"]/@value/data())+$n/float[@key="cost:total"]/@value/data())<$c1)`
`then (<trace id="{$i}" value="true"/>)`
`else (<trace id="{$i}" value="false"/>) )`
`return <result rule="ProcessCostLess">{$r}</result>`	

Tabelle 6.23: Implementierung der Compliance-Regel CR-COST-0003 mit XQuery

Kurzname: XQ-CR-COST-0003	Name: ProcessCostGreater
Kategorie: Kosten	
Parameter: Event-Log `e` als `document-node()`, Kosten `c1` as `xs:float`	

XQuery-Ausdruck:
`let $r:=(for $n at $i in $e//trace[position()=(1 to last())]`
`let $ac:=$n/event[string/@key="lifecycle:transition" and string/@value="complete"]`
`return if ((sum($ac[position()=(1 to last())]/float[@key="cost:total"]/@value/data())+$n/float[@key="cost:total"]/@value/data())>$c1)`
`then (<trace id="{$i}" value="true"/>)`
`else (<trace id="{$i}" value="false"/>) )`
`return <result rule="ProcessCostGreater">{$r}</result>`	

Mit der Übergabe zusätzlicher Parameter (z.B. eine zugehörige Compliance-Anforderung oder -Kontrolle) und deren Übernahme im result-Element der XQuery-Ausdrücke, kann eine detaillierte Aufbereitung, Darstellung und Analyse der Ergebnisse der Compliance-Prüfung ermöglicht werden.

7 Prototypische Umsetzung

Zur informationstechnologischen Unterstützung des Compliance Managements von Geschäftsprozessen werden im Rahmen dieser Arbeit drei prototypische Software-Komponenten konzipiert bzw. umgesetzt:

Mit dem Process Compliance Manager (PCM) wird, aufbauend auf dem in Kapitel 5.3 vorgestellen Business Process Compliance Management Model, eine Erweiterung für das Geschäftsprozessmanagement und -modellierungswerkzeug Horus Business Modeler konzipiert, die verschiedene Funktionalitäten zur Verwaltung von Compliance-Anforderungen, -Kontrollen und -Regeln sowie zur Durchführung von Compliance-Audits zur Verfügung stellt.

Des Weiteren wurde mit der XES Process Compliance Library (XPCL) eine Bibliothek entwickelt, die eine Referenzimplementierung des in Kapitel 6 vorgestellten Ansatzes darstellt und in Verbindung mit dem PCM eine ereignisprotokollbasierte Compliance-Prüfung von Geschäftsprozessen ermöglicht.

Darüber hinaus wurde mit dem Process Compliance Dashboard (PCD) ein webbasiertes Werkzeug entwickelt, das eine anschließende Aufbereitung, Darstellung und Analyse der Auditergebnisse erlaubt.

7.1 Process Compliance Manager

Der Process Compliance Manager ist als Plugin für das Geschäftsprozessmanagement und -modellierungswerkzeug Horus Business Modeler konzipiert und stellt maßgeblich Funktionalitäten zur Verwaltung von Compliance-Anforderungen, -Kontrollen und -Regeln sowie zur Durchführung von Compliance-Audits zur Verfügung. Die Integration in ein Geschäftsprozessmangement und -modellierungs-werkzeug erlaubt eine enge Verzahnung der im Rahmen des Compliance Managements durchzuführenden Aktivitäten mit denen des Geschäftsprozessmangements und ermöglicht so eine engere Kollaboration der in den unterschiedlichen Lebenszyklus-Phasen beteiligten Akteure.

In Anlehnung an die in der Literatur (Tarantino, 2008; IDW, 2010; TÜV Rheinland, 2011) diskutierten sowie aus dem Geschäftsprozessmanagement bekannten Akteure (Kapitel 3) können für das Compliance Management folgende Rollen identifiziert bzw. unterschieden werden:

- Chief Compliance Officer: Der Chief Compliance Officer (CCO) vertritt auf der obersten Unternehmensebene die Compliance-Interessen eines Unternehmens und steht in dieser Funktion zudem einem entsprechenden Gremium (z.B. Compliance Board) vor. Der Chief Compliance Officer ist für die unternehmensweite Umsetzung der Compliance-Anforderungen verantwortlich.
- Compliance Manager: Compliance Manager verantworten einen fachlichen oder organisatorischen Bereich, für den sie die Einhaltung der Compliance-Anforderungen überwachen. Sie sind für die Ermittlung, Spezifikation und Dokumentation sowie Pflege der Compliance-Anforderungen, -Kontrollen und -Regeln verantwortlich. Aufgrund der Vielzahl an Compliance-Anforderungen kann es für unterschiedliche Bereiche bzw. Themen mehrere Compliance Manager geben, die wiederum ein spezielles Teilgebiet verantworten. Sie sind in der Regel die ersten Ansprechpartner der Business Engineers und Prozesseigentümer hinsichtlich möglicher Compliance-Fragestellungen und berichten direkt an den Chief Compliance Officer des Unternehmens.
- Compliance Analyst: Compliance Analysten (auch Compliance Experten) verfügen über ein besonderes domänenspezifisches Wissen und unterstützten den Compliance Manager des Unternehmens bzw. eines Bereichs oder Teilgebiets bei der Durchführung seiner Aufgaben. Hierzu überwachen sie die für ein Unternehmen relevanten Compliance-Quellen und leiten in Abstimmung mit den Prozesseigentümern, Process Designern sowie Systemarchitekten (neue) Compliance-Kontrollen und Compliance-Regeln ab.

In Abhängigkeit von der Größe des Unternehmens bzw. dem Umfang und der Komplexität der Compliance-Fragestellungen können die Rollen Compliance Manager und Compliance Analyst auch von derselben Person ausgefüllt werden.

- Compliance Auditor: Compliance Auditoren können sowohl unternehmensintern benannte als auch von externen Unternehmen gestellte bzw. beauftragte Personen sein, die für die ordnungsgemäße Planung, Durchführung und Dokumentation von Compliance-Audits verantwortlich sind. Darüber hinaus unterstützen sie die Business Engineers, Prozesseigentümer, Compliance Manager und Compliance Analysten bei der Analyse der Auditergebnisse und leiten gemeinsam Verbesserungsvorschläge und -maßnahmen ab.

Ziel des in Anlehnung an das Business Process Compliance Management Model und als Plugin für das Geschäftsprozessmanagement und -modellierungswerkzeug Horus Business Modeler konzipierten Process Compliance Manager ist es, die einzelnen Akteure bei der Durchführung ihrer Aufgaben adäquat zu unterstützen.

Hierzu wird mit den Compliance-Modellen eine zusätzliche, auf Objektmodellen basierende Modellart eingeführt und in den Workspace bzw. Workspace Explorer des Horus Business Modeler integriert. Compliance-Modelle dienen dabei maßgeblich der Verwaltung von Compliance-Anforderungen, -Kontrollen und -Regeln. Über den Workspace Explorer kann ein neues Compliance-Modell hinzugefügt bzw. erstellt werden (Abbildung 7.1). Für die Ableitung und Erstellung der Compliance-Modelle sind der Compliance Manager sowie Compliance Analysten verantwortlich.

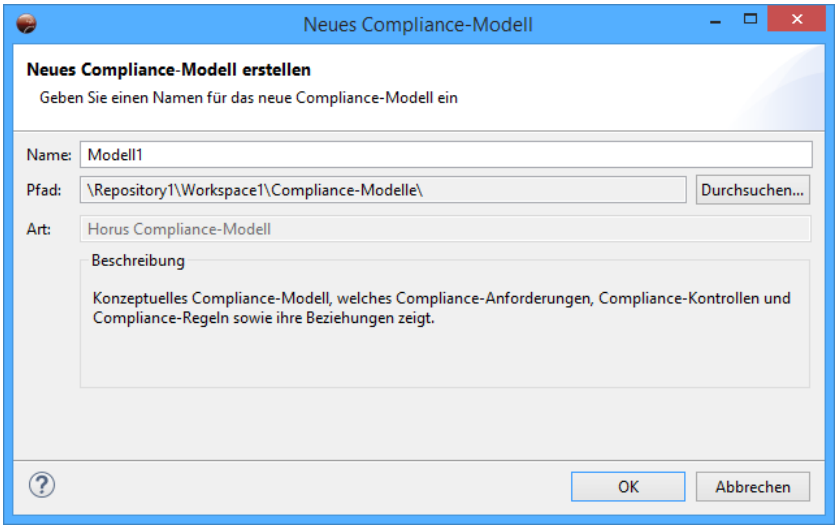

Abbildung 7.1: Neues Compliance-Modell erstellen

Als Notation für die Compliance-Modelle wird eine Kombination von Konzepten aus dem Asset Oriented Modelling (AOM), UML-Klassendiagrammen und Entity-Relationship- bzw. Erweiteren Entitiy-Relationship-Modellen (ER- bzw. EER) verwendet (Schönthaler, Vossen, Oberweis, & Karle, 2011). Die aus dem AOM bekannten assets entsprechen dabei Compliance-Anforderungen, -Kontrollen und Regeln. Mit Hilfe von Kanten (arcs) werden Beziehungen und Abhängigkeiten abgebildet (Abbildung 7.2).

Über eine Palette können die einzelnen Modellelemente (Compliance-Anforderungen, -Kontrollen und -Regeln sowie Beziehungen) ausgewählt und einem Compliance-Modell hinzugefügt werden. Ein Compliance-Modell enthält genau eine Compliance-Anforderung sowie mehrere Compliance-Kontrollen und -Regeln. Eine Compliance-Anforderung kann nur mit Compliance-Kontrollen, Compliance-Kontrollen wiederum nur mit Compliance-Regeln verbunden werden. Eine Compliance-Regel wird genau einer Compliance-Kontrolle zugeordnet. Werden mehrere Modellelemente mit einem anderen Modellelement verbunden, wird dies als logische Und-Bedingung interpretiert (Abbildung 7.2) (Schleicher, et al., 2011).

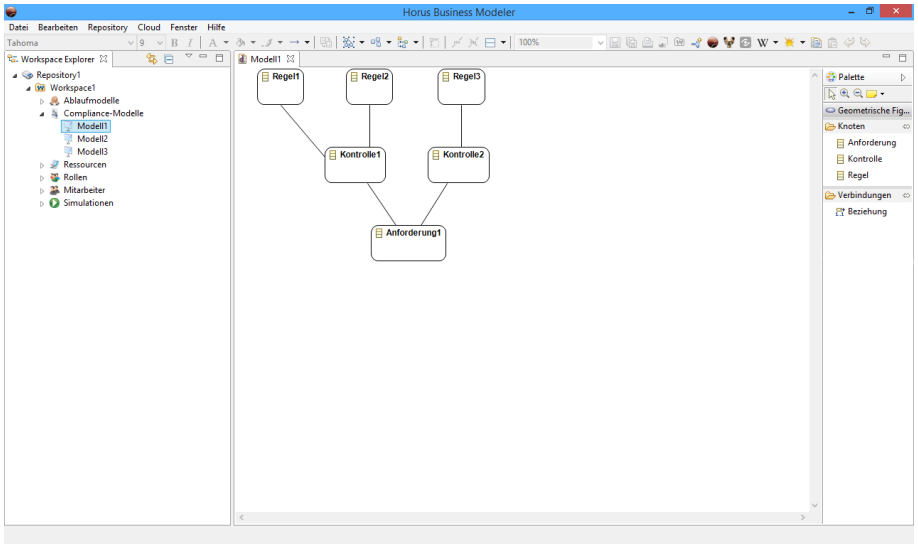

Abbildung 7.2: Beispiel eines Compliance-Modells

In Abbildung 7.3 ist der Eigenschaftsdialog einer Compliance-Anforderung dargestellt. Unter dem Menüpunkt „Allgemein“ können ein Kurzname, ein Name und eine Beschreibung eingetragen werden. Darüber hinaus kann die Compliance-Anforderung einem Ablaufmodell zugewiesen werden. Unter dem Menüpunkt „Erweitert“ können, wie in Kapitel 5.1 beschrieben, zusätzliche Attribute für die Compliance-Anforderung festgelegt werden, z.B. eine Compliance-Quelle oder -Risiken.

Im Eigenschaftsdiaglog einer Compliance-Kontrolle können unter dem Menüpunkt „Allgemein“ ebenfalls ein Kurzname, ein Name und eine Beschreibung eingetragen werden. Darüber hinaus muss für eine Compliance-Kontrolle festgelegt werden, ob es sich um eine manuelle oder eine automatisierte Compliance-Kontrolle handelt.

Für das Attribut „Typ“ ist der Wert „manuell“ voreingestellt. Wird der Wert „automatisiert“ ausgewählt, können einer Compliance-Kontrolle im Anschluss mehrere Compliance-Regeln zugewiesen werden (Abbildung 7.4).

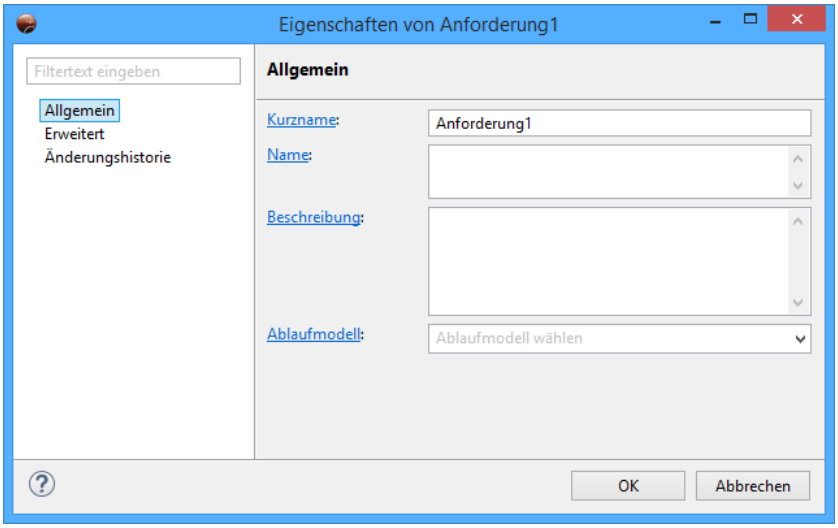

Abbildung 7.3: Eigenschaften einer Compliance-Anforderung

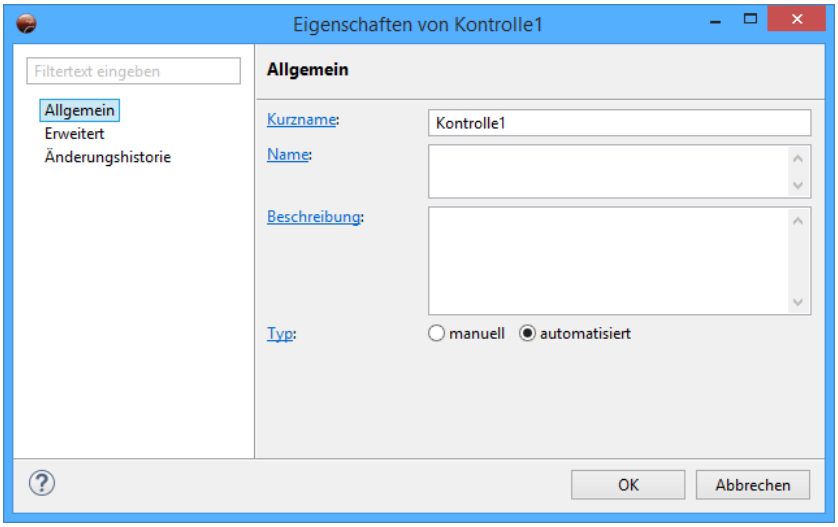

Abbildung 7.4: Eigenschaften einer Compliance-Kontrolle

Im Eigenschaftsdialog einer Compliance-Regel kann eine von der XPCL automatisiert prüfbare Compliance-Regel ausgewählt werden (Abbildung 7.5).

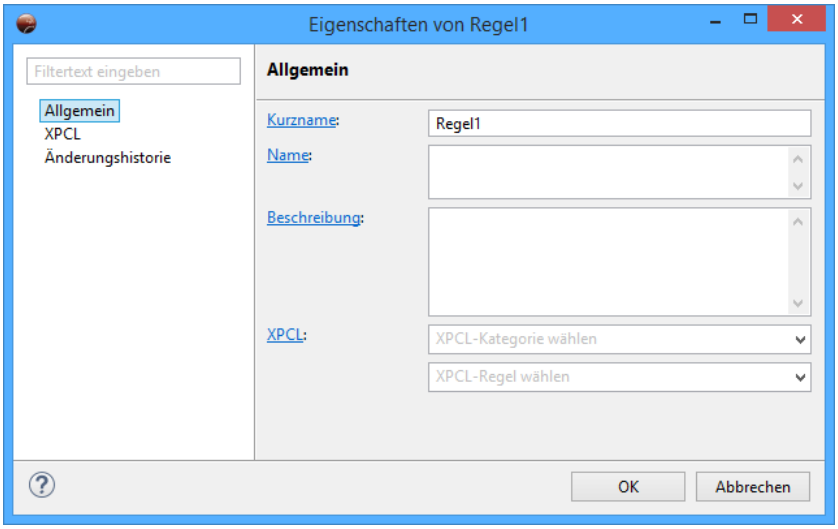

Abbildung 7.5: Eigenschaften einer Compliance-Regel

Unter dem Menüpunkt „XPCL“ können der Compliance-Regel verschiedene, für die Compliance-Prüfung benötigte Parameter zugewiesen werden (Abbildung 7.6). Für die korrekte Zuweisung der für die Compliance-Prüfung benötigten Parameter ist eine enge Abstimmung der Compliance Analysten mit den Prozesseigentümern und Systemarchitekten erforderlich.

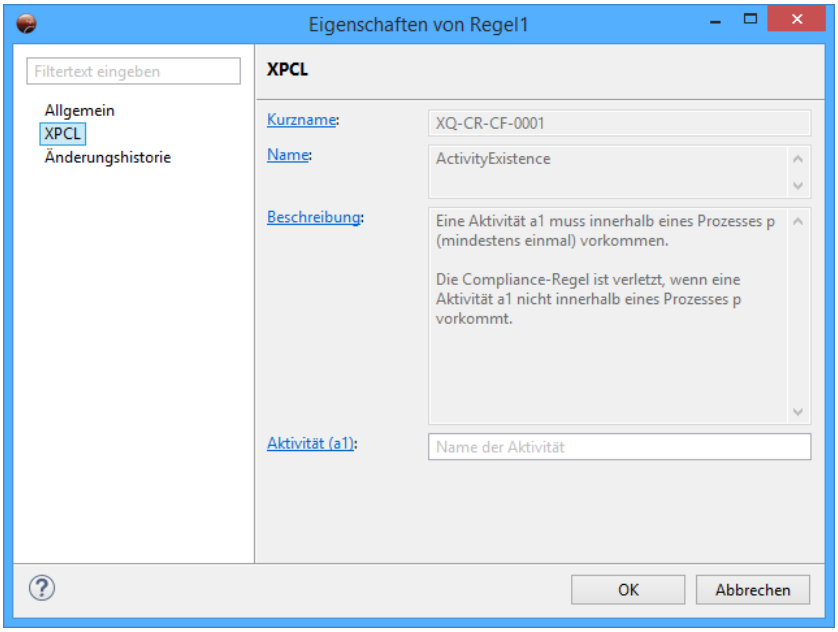

Abbildung 7.6: Parametrisierung einer Compliance-Regel

Nach der Erstellung der Compliance-Modelle kann eine automatisierte Compliance-Prüfung erfolgen. Ein vom Process Compliance Manager zur Verfügung gestellter Dialog unterstützt bei der Durchführung eines Compliance-Audits (Abbildung 7.7). Für die Planung, Durchführung und Dokumentation der Compliance-Audits sind die Compliance Auditoren zuständig.

- Im ersten Schritt wird ein Name für das Compliance-Audit festgelegt sowie ein zu auditierender Geschäftsprozess in Form eines Ablaufmodells ausgewählt. Darüber hinaus erfolgt die Auswahl eines Simulations-Trace oder Ereignisprotokolls (Event-Log).
- Im zweiten Schritt werden die dem Ablaufmodell zugeordneten Compliance-Anforderungen zur Auswahl gestellt. Dabei werden nur diejenigen Compliance-Anforderungen angezeigt, für die zuvor auch automatisierte Compliance-Kontrollen sowie Compliance-Regeln definiert wurden. Einzelne Compliance-Anforderungen können abgewählt und damit von der Compliance-Prüfung ausgenommen werden.
- Im dritten Schritt wird das Compliance-Audit durchgeführt. Hierzu werden die für die Compliance-Prüfung erforderlichen Angaben sowie benötigten Parameter an die XPCL übergeben.

- Nach Abschluss der Compliance-Prüfung können die Auditergebnisse im vierten Schritt über den angegebenen Link im webbasierten Process Compliance Dashboard dargestellt und analysiert werden.

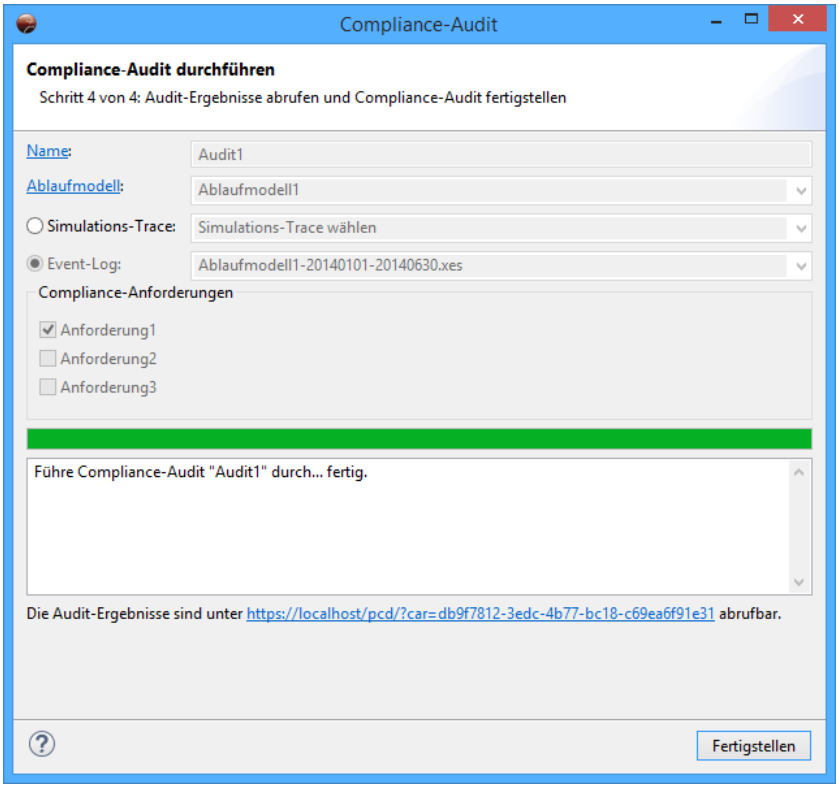

Abbildung 7.7: Compliance-Audit durchführen – Schritt 4 von 4

7.2 XES Process Compliance Library

Die XES Process Compliance Library stellt eine Referenzimplementierung des in Kapitel 6 beschriebenen Ansatzes zur ereignisprotokollbasierten Compliance-Prüfung mit XQuery (W3C, 2014) dar. Mit BaseX[1] unterstützt die in Java umgesetzte Bibliothek eine leichtgewichtige, skalierbare sowie plattformunabhängige Open-Source XML-Datenbank und W3C-konformen XPath/XQuery 3.1-Prozessor. Die Kommunikation mit BaseX erfolgt auf Basis einer Client-Server-Architektur.

[1] `http://basex.org/`

Im Rahmen der prototypischen Umsetzung werden die einzelnen Compliance-Regeln als Klassen umgesetzt, die die parametrisierbaren XQuery-Ausdrücke als vorbereitete Anweisungen (prepared expressions, prepared statements) enthalten. Compliance-Regeln bzw. Klassen einer Kategorie werden in einem Paket (package) zusammengefasst; ein Paket compliance enthält wiederum die Pakete der einzelnen Kategorien (controlflow, organization, time, cost). Abbildung 7.8 veranschaulicht beispielhaft den Aufbau bzw. Inhalt des Pakets controlflow.

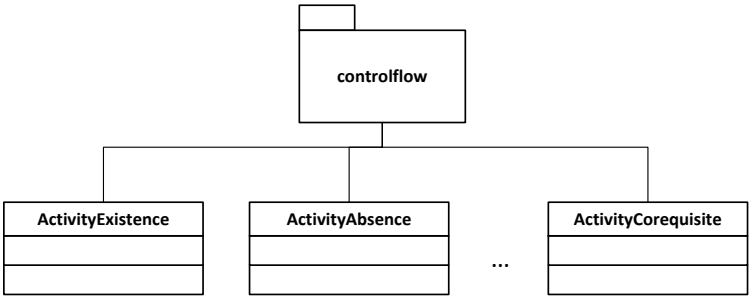

Abbildung 7.8: Paket controlflow und enthaltene Klassen

Darüber hinaus stellt die XES Process Compliance Library verschiedene Funktionalitäten zur Validierung und Transformation von XML-Dokumenten zur Verfügung (Pakete validation und transformationen). Abbildung 7.9 stellt die XPCL als Paketdiagramm dar.

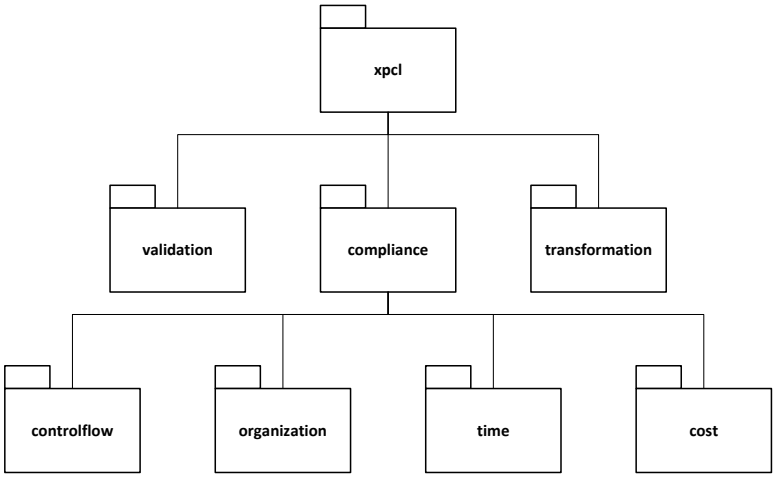

Abbildung 7.9: Paketdiagramm der XPCL

Die Ergebnisse eines Compliance-Audits bzw. der ereignisprotokollbasierten Compliance-Prüfung werden in einem XML-basierten Compliance Audit Result-Dokument gespeichert. Ein Compliance Audit Result-Dokument enthält ein Wurzelelement audit, in dem neben dem Namen sowie dem Zeitpunkt des Audits Informationen zum auditierten Geschäftsprozess bzw. Ereignisprotokoll sowie eine UUID gespeichert werden. Ein audit-Element enthält eine beliebige Anzahl von result-Elementen, in denen für jede Compliance-Regel neben ihrem Namen und ihrer Kategorie die übergebenen Parameter, die zugeordnete Compliance-Anforderung und -Kontrolle, ein Ergebnistyp und eine UUID festgehalten werden. Result-Elemente enthalten wiederum eine beliebige Anzahl von trace-Elementen, in denen für jede Prozessinstanz (id-Attribut) das Ergebnis der Compliance-Prüfung als Boole'scher Wert (value-Attribut) gespeichert wird. Im Folgenden ist ein beispielhafter Ausschnitt eines Compliance Audit Result-Dokuments dargestellt, das wiederum (maschinell) weiterverarbeitet werden kann (Tabelle 7.1).

Tabelle 7.1: Ausschnitt eines Compliance Audit Result-Dokuments

```
<?xml version="1.0" encoding="UTF-8"?>
<audit name="Audit1" process="Ablaufmodell1" eventlog="Ablaufmodell1-20140101-
20140630.xes" time="1404199815" uuid="db9f7812-3edc-4b77-bc18-c69ea6f91e31">
  <result rule="ActivityExistence" parameter="a1:A" control="Kontrolle1"
requirement="Anforderung1" category="CF" type="boolean" uuid="e05bace3-722a-4a0a-
b6bd-d4243b3569b2">
    <trace id="1" value="true"/>
    <trace id="2" value="true"/>
    <trace id="3" value="true"/>
    ...
  </result>
  <result rule="ActivityAbsence" parameter="a1:B" control="Kontrolle1"
requirement="Anforderung1" category="CF" type="boolean" uuid="bbedd8f6-65b7-4f39-
b66f-384b1387d490">
    <trace id="1" value="false"/>
    <trace id="2" value="false"/>
    <trace id="3" value="false"/>
    ...
  </result>
  <result rule="ActivityCoexistence" parameter="a1:C#a2:D" control="Kontrolle2"
requirement="Anforderung1" category="CF" type="boolean" uuid="a689d6e4-62b2-45d0-
b8ad-19f977d0691d">
    <trace id="1" value="false"/>
    <trace id="2" value="true"/>
    <trace id="3" value="false"/>
    ...
  </result>
</audit>
```

Das XSD-Schema eines Compliance Audit Result-Dokuments ist im Anhang B dieser Arbeit aufgeführt.

7.3 Process Compliance Dashboard

Zur Aufbereitung, Darstellung und Analyse der Auditergebnisse wird im Rahmen dieser Arbeit ein webbasiertes Process Compliance Dashboard entwickelt, das insbesondere Compliance Auditoren, aber auch Compliance Manager und -Experten bei ihrer Arbeit unterstützen soll. Die Umsetzung erfolgt dabei maßgeblich mit Hilfe der folgenden offenen Frameworks und Bibliotheken:

- Bootstrap[2] ist ein schlankes und anpassungsfähiges sowie mit einem Schwerpunkt auf mobile Endgeräte ausgerichtetes Framework zur schnellen und einfachen Web-Entwicklung.
- d3.js[3] ist eine JavaScript-Bibliothek zur datengetriebenen Manipulation von Dokumenten, die die Erstellung von dynamischen, webbasierten Grafiken unter Verwendung von HTML, SVG und CSS ermöglicht.
- dc.js[4] ist eine JavaScript-Bibliothek mit nativer Crossfilter[5]-Unterstützung, die eine effiziente Untersuchung von mehrdimensionalen Datensätzen anhand interaktiver Diagramme ermöglicht.

Abbildung 7.10 veranschaulicht das im Rahmen der protoypischen Umsetzung zur Aufbereitung und Darstellung der Auditergebnisse gewählte Vorgehen. Dabei werden die als XML-basierte Compliance Audit Result-Dokumente vorliegenden Auditergebnisse zunächst in JSON bzw. JSON-Objekte transformiert (Data Transformation) und im Anschluss an die JavaScript-Bibliotheken d3.js und dc.js zur Erzeugung interaktiver, dynamischer Diagramme übergeben (Chart Generation). Abschließend können die derart aufbereiteten Auditergebnisse im webbasierten Process Compliance Dashboard dargestellt werden.

[2] http://getbootstrap.com/
[3] http://d3js.org/
[4] http://dc-js.github.io/dc.js/
[5] http://square.github.io/crossfilter/

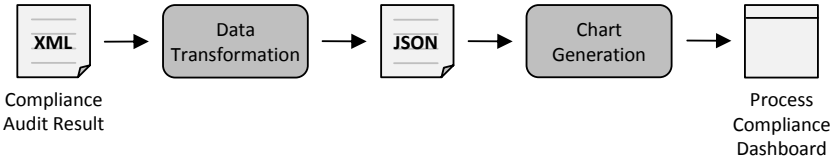

Abbildung 7.10: Vom Compliance Audit Result zum Process Compliance Dashboard

Das Process Compliance Dashboard lässt sich in fünf Darstellungsbereiche (tabs) unterteilen, die über das Navigationsmenü aufgerufen werden können. Auf der Start- bzw. Übersichtsseite werden die wesentlichen zu dem Compliance-Audit gehörenden Informationen zusammengefasst (Abbildung 7.11).

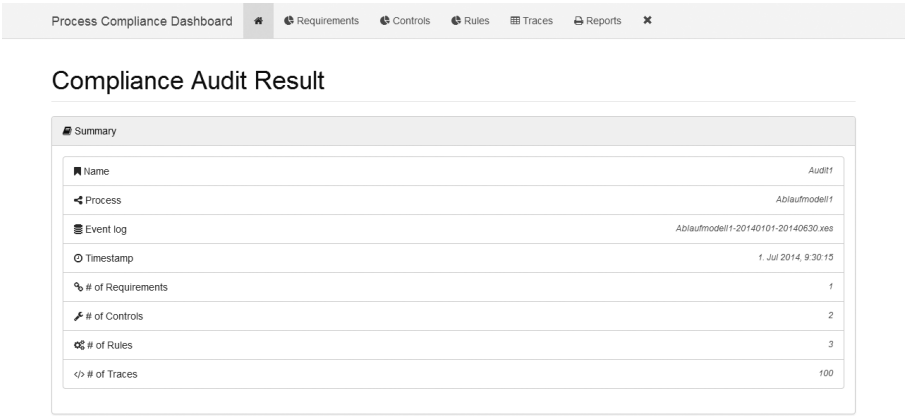

Abbildung 7.11: Startseite des PCD

In den folgenden Darstellungsbereichen (Requirements, Controls, Rules) werden die Auditergebnisse in Abhängigkeit von dem jeweiligen Ergebnis der Compliance-Prüfung sowie der Gesamtzahl der geprüften Abläufe bzw. Prozessinstanzen als Kuchendiagramme dargestellt. Compliance-Regeln werden zudem anhand ihrer Kategorie gruppiert. Über die unterhalb der einzelnen Diagramme angeordneten Schaltflächen können verschiedene Informationen (z.B. Beziehungen und Abhängigkeiten sowie übergebene Parameter) angezeigt werden (Abbildung 7.12).

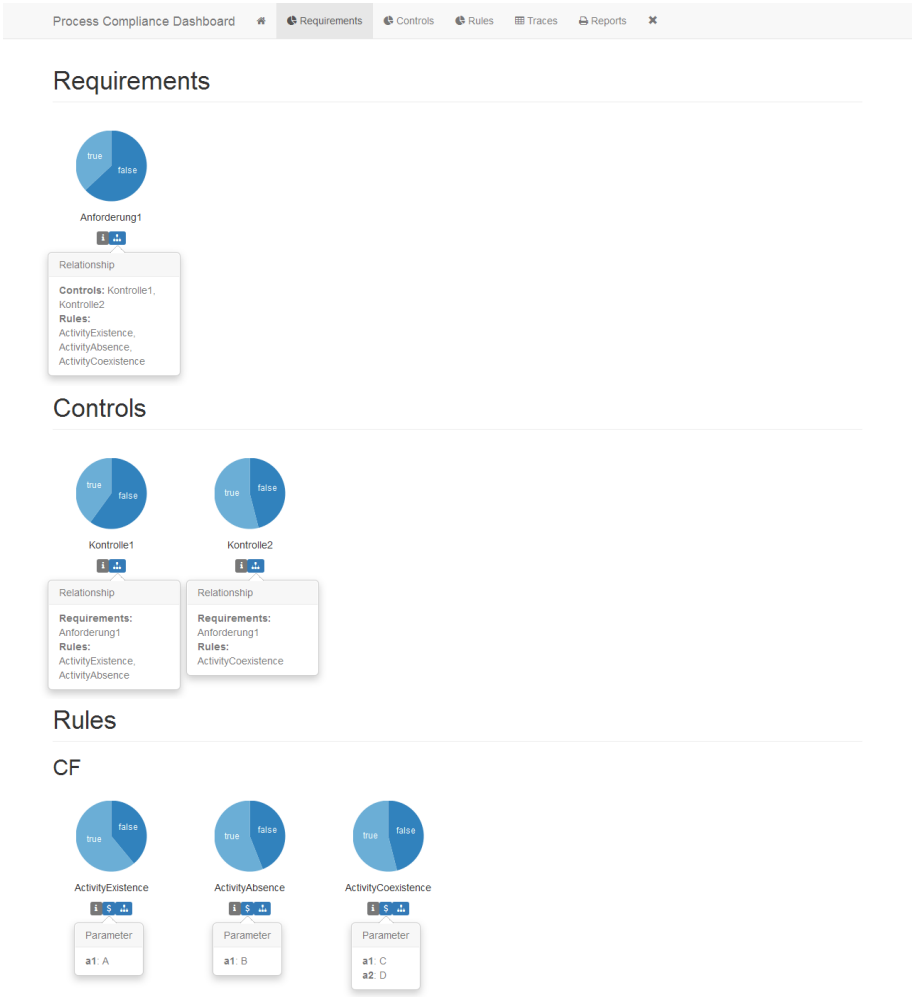

Abbildung 7.12: PCD – Requirements, Controls, Rules

Im vorletzten Darstellungsbereich (Traces) werden die Ergebnisse der Compliance-Prüfung nach Compliance-Regel und Prozessinstanz tabellarisch dargestellt (Abbildung 7.13).

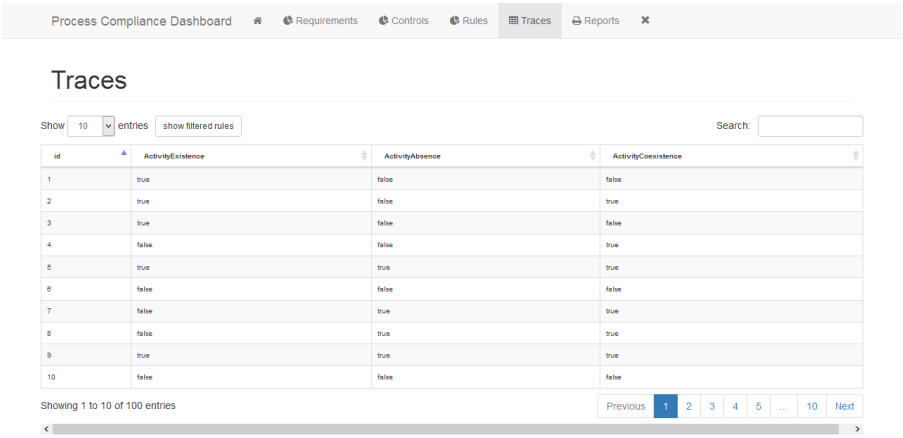

Abbildung 7.13: PCD – Traces

Der letzte Darstellungsbereich (Reports) enthält eine detaillierte Zusammenfassung der Auditergebnisse (Abbildung 7.14).

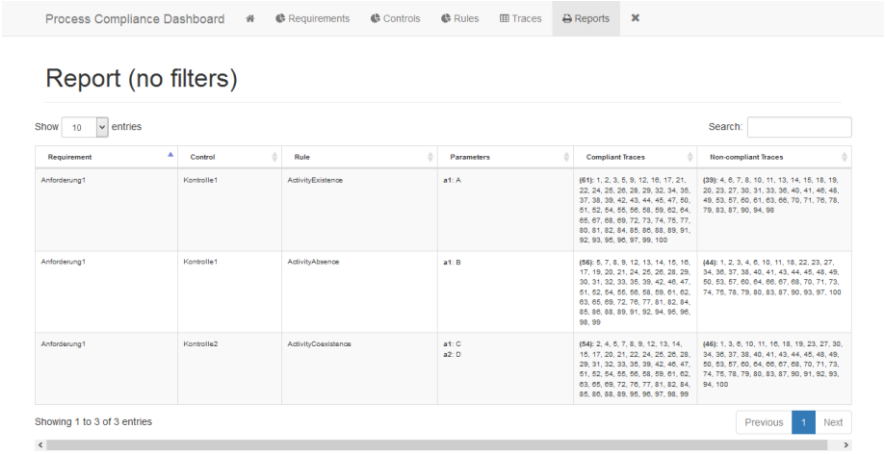

Abbildung 7.14: PCD – Reports

Um eine Ursachenanalyse (root cause analysis) zu ermöglichen, kann durch eine interaktive Auswahl einzelner Diagrammbereiche (true/false) eine Eingrenzung der für die Verletzung von Compliance-Anforderungen, -Kontrollen und -Regeln verantwortlichen Prozessinstanzen vorgenommen werden. Hierzu werden die in den Diagrammen dargestellten sowie in den Datentabellen aufgeführten Prozessinstanzen entsprechend der

getroffenen Auswahl und basierend auf den in den Compliance-Modellen definierten und modellierten Beziehungen und Abhängigkeiten dynamisch gefiltert. Darüber hinaus erfolgt eine farbliche Hervorhebung der Diagramme (Abbildung 7.15).

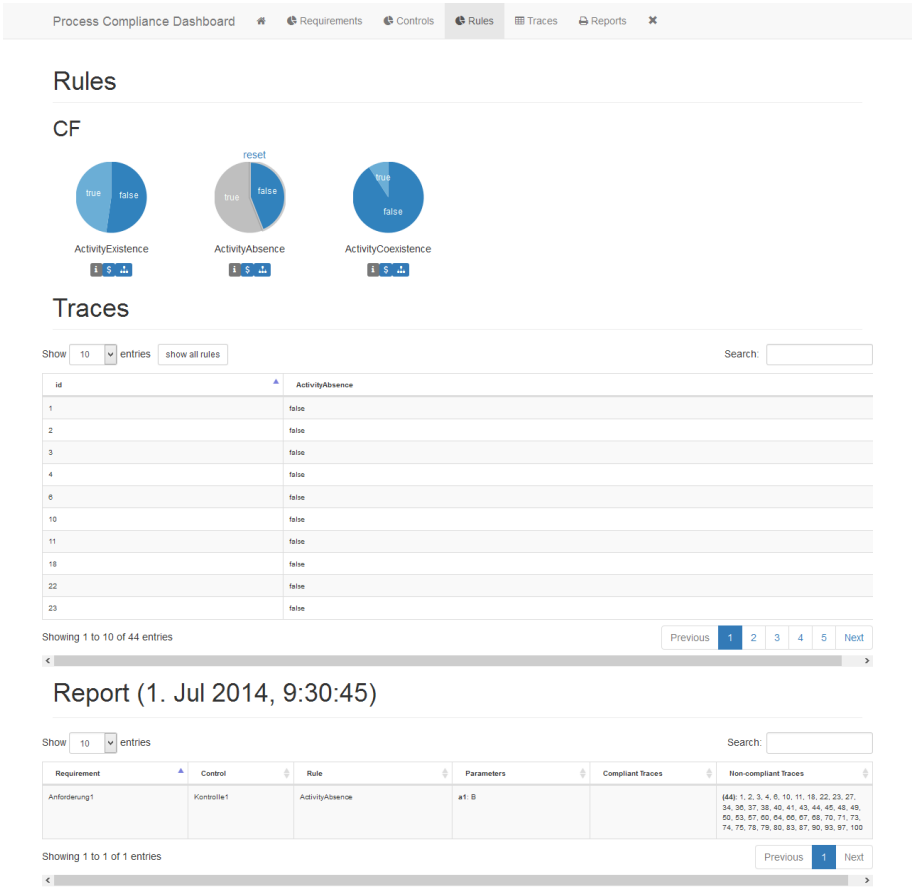

Abbildung 7.15: PCD – Interaktive Auswahl und dynamische Filterung

Über das im Navigationsmenü dargestellte Kreuz kann die getroffene Auswahl wieder zurückgesetzt werden. Die für die prototypische Umsetzung gewählten offenen Frameworks und Bibilotheken erlauben eine flexible Anpassung und Erweiterung des Process Compliance Dashboards um zusätzliche Analyse- und Reporting-Funktionalitäten (z.B. Heatmaps).

8 Evaluation

In diesem Kapitel wird der im Rahmen der vorliegenden Arbeit entwickelte Ansatz zur ereignisprotokollbasierten Compliance-Prüfung von Geschäftsprozessen anhand eines Anwendungsbeispiels evaluiert. Abschließend werden die Grenzen des Ansatzes aufgezeigt und die Ergebnisse der Evaluation kritisch diskutiert.

8.1 Anwendungsbeispiel Laborprozess

Als Anwendungsbeispiel wird ein Geschäftsprozesses gewählt, der an einen im Rahmen eines Industrieprojekts betrachteten Laborprozess angelehnt ist. Das Projekt wurde gemeinsam mit Partnern aus dem Bereich der Laborautomatisierung sowie der privaten und klinischen Diagnostik durchgeführt. Für die Evaluation wird das folgende Vorgehen gewählt:

- In einem ersten Schritt wird der Laborprozess modelliert und (formal) beschrieben. Im Anschluss werden mit Hilfe der computergestützten Simulation XES-konforme Ereignisprotokolle erzeugt.
- In einem zweiten Schritt werden in Anlehnung an die im Rahmen des Industrieprojekts gewonnenen Einblicke beispielhafte Compliance-Anforderungen, -Kontrollen und -Regeln definiert.
- In einem dritten Schritt wird ein Compliance-Audit bzw. eine ereignisprotokollbasierte Compliance-Prüfung auf Basis des zuvor modellierten Laborprozesses sowie der beispielhaft definierten Compliance-Anforderungen durchgeführt. Neben der Aufbreitung, Darstellung und Analyse der Auditergebnisse im Process Compliance Dashboard erfolgt darüber hinaus eine Untersuchung der Leistungsfähigkeit (performance) des entwickelten Ansatzes bzw. dessen Referenzimplementierung (XES Process Compliance Library).

8.1.1 Modellierung und Simulation

Die Modellierung des im Rahmen des Anwendungsbeispiels betrachteten Laborprozesses erfolgt mit Hilfe von annotierten Workflow-Netzen.

Definition 8.1: Ein annotiertes Workflow-Netz ist ein 9-Tupel $WN_A = (P, T, F, L, RE, RO, F_T, F_{RE}, F_{RO})$ für das gilt:

- (P, T, F) ist ein Workflow-Netz.
- L ist eine endliche Menge von Beschriftungen (label).
- RE ist eine endliche Menge von (menschlichen und maschinellen) Ressourcen, die an einem Geschäftsprozess beteiligt sind.
- RO ist eine endliche Menge von Rollen.
- $F_T: T \rightarrow L$ ist eine Funktion, die jeder Transition eine Beschriftung zuweist.
- $F_{RE}: RE \rightarrow RO$ ist eine Funktion, die jeder Ressource eine oder mehrere Rolle(n) zuweist.
- $F_{RO}: RO \rightarrow T$ ist eine Funktion, die jeder Rolle eine oder mehrere Transition(en) zuweist.

In Anlehnung an (van der Aalst, 1995; van der Aalst, 1998) wird im Rahmen dieser Arbeit für ein annotiertes Workflow-Netz gefordert, dass es intakt ist und in endlicher Zeit terminiert. Darüber hinaus kann ein annotiertes Workflow-Netz Transitionen enthalten, die nicht beobachtbar sind bzw. sein sollen und ausschließlich zur Steuerung des Kontrollflusses in das annotierte Workflow-Netz eingefügt werden. Nicht beobachtbare Transitionen werden als stille (silent) oder unsichtbare (invisible) Transitionen bezeichnet; ihnen wird die Beschriftung τ (auch leere Beschriftung) zugewiesen (van der Aalst, 2011).

Abbildung 8.1 veranschaulicht das im Horus Business Modeler als Ablaufmodell modellierte annotierte Workflow-Netz WN_A^{LP}:

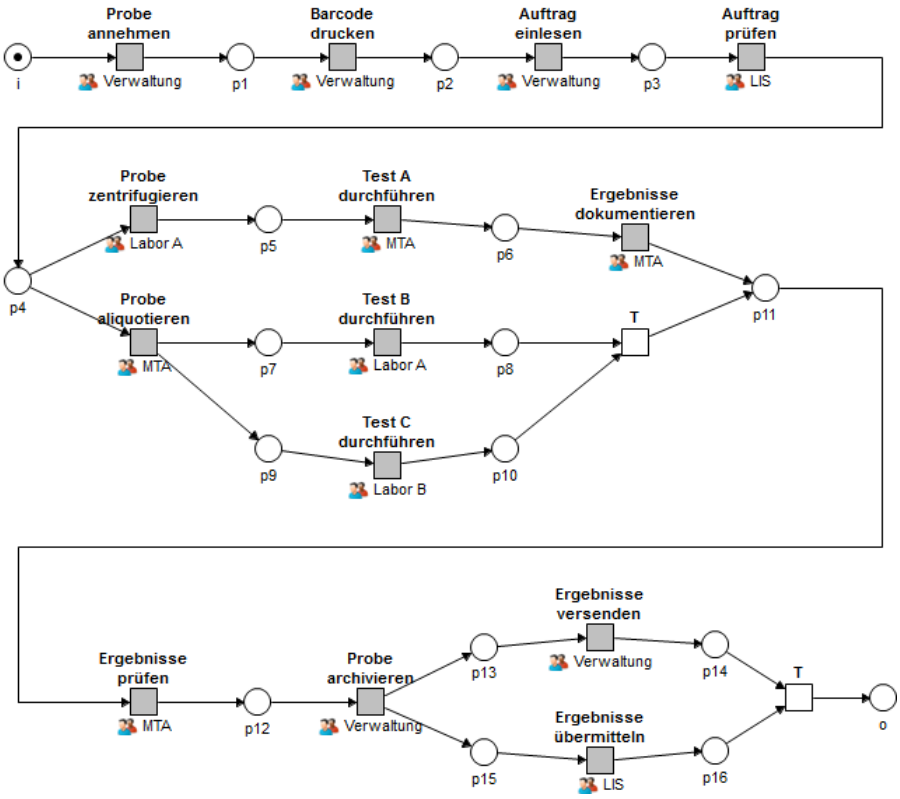

Abbildung 8.1: Anwendungsbeispiel Laborprozess

Der Laborprozess beginnt mit der Annahme einer Probe im Wareneingang (Probe annehmen). Im Anschluss wird ein Barcode gedruckt (Barcode drucken) und an der Probe angebracht. Durch das Scannen des Barcordes wird die Probe im Laborinformationssystem registriert. In der Folge wird der einer Probe beiliegende, schriftliche Untersuchungsauftrag in das Laborinformationssystem eingelesen bzw. eingegeben und der Probe zugeordnet (Auftrag einlesen). Nach Sicherstellung der Plausibilität und Durchführbarkeit der zu einer Probe beauftragten Tests (Auftrag prüfen) wird die Probe zur Untersuchung freigegeben. Anschließend wird die Probe in Abhängigkeit von den beauftragten bzw. durchzuführenden Tests entweder zentrifugiert (Probe zentrifugieren) oder aliquotiert (Probe aliquotieren). Die Durchführung der Tests kann manuell (Test A durchführen) oder automatisiert (Test B und C durchführen) erfolgen. Nach Abschluss von Test A werden die in Schriftform vorliegenden Unterschuchungsergebnisse im Laborinformationssystem dokumentiert (Ergebnisse dokumentieren); die Untersuchungsergebnisse von Test B und C werden von den Laborgeräten direkt an das Labor-

informationssystem übermittelt. Nach der Sicherstellung der Plausibilität der Untersuchungsergebnisse (Ergebnisse prüfen) wird die Probe verschlossen und in einem klimatisierten Raum archiviert (Probe archivieren). Abschließend werden die freigegebenen Untersuchungsergebnisse gedruckt und versendet (Ergebnisse versenden) sowie elektronisch an den Auftraggeber übermittelt (Ergebnisse übermitteln).

Das annotierte Workflow-Netz WN_A^{LP} lässt sich wie folgt formal beschreiben:

- Workflow-Netz (P^{LP}, T^{LP}, F^{LP}):

$$P^{LP} = \begin{Bmatrix} i, p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, \\ p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, o \end{Bmatrix}$$

$$T^{LP} = \begin{Bmatrix} t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, \\ t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16} \end{Bmatrix}$$

$$F^{LP} = \left\{ \begin{array}{c} (i, t_1), (t_1, p_1), (p_1, t_2), (t_2, p_2), (p_2, t_3), (t_3, p_3), \\ (p_3, t_4), (t_4, p_4), (p_4, t_5), (t_5, p_5), (p_5, t_6), (t_6, p_6), \\ (p_6, t_7), (t_7, p_{11}), (p_4, t_8), (t_8, p_7), (p_7, t_9), (t_9, p_8), \\ (p_8, t_{11}), (t_8, p_9), (p_9, t_9), (t_9, p_{10}), (p_{10}, t_{11}), (t_{10}, p_{11}), \\ (p_{11}, t_{12}), (t_{12}, p_{12}), (p_{12}, t_{13}), (t_{13}, p_{13}), (p_{13}, t_{14}), (t_{14}, p_{14}), \\ (p_{14}, t_{16}), (t_{13}, p_{15}), (p_{15}, t_{15}), (t_{15}, p_{16}), (p_{16}, t_{16}), (t_{16}, o), \end{array} \right\}$$

$$M_0^{LP} = \{1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\}$$

- Beschriftungen L^{LP}:

$$L^{LP} = \left\{ \begin{array}{c} \text{Probe annehmen, Barcode drucken, Auftrag einlesen,} \\ \text{Auftrag prüfen, Probe zentrifugieren, Test A durchführen,} \\ \text{Ergebnisse dokumentieren, Probe aliquotieren,} \\ \text{Test B durchführen, Test C durchführen, Ergebnisse prüfen,} \\ \text{Probe archivieren, Ergebnisse versenden,} \\ \text{Ergebnisse übermitteln} \end{array} \right\}$$

- Ressourcen RE^{LP}:

$$RE^{LP} = \begin{Bmatrix} \text{Eva, Laborgerät 1, Laborgerät 2,} \\ \text{Laborinformationssystem, Max, Moritz} \end{Bmatrix}$$

- Rollen RO^{LP}:

$$RO^{LP} = \{\text{Labor A, Labor B, LIS, MTA, Verwaltung}\}$$

- Funktionen $F_T^{LP}, F_{RE}^{LP}, F_{RO}^{LP}$:

$$
\begin{aligned}
&F_T^{LP}(t_1) = \text{Probe annehmen},\\
&F_T^{LP}(t_2) = \text{Barcode drucken},\\
&F_T^{LP}(t_3) = \text{Auftrag einlesen},\\
&F_T^{LP}(t_4) = \text{Auftrag prüfen},\\
&F_T^{LP}(t_5) = \text{Probe zentrifugieren},\\
&F_T^{LP}(t_6) = \text{Test A durchführen},\\
&F_T^{LP}(t_7) = \text{Ergebnisse dokumentieren},\\
&F_T^{LP}(t_8) = \text{Probe aliquotieren},\\
&F_T^{LP}(t_9) = \text{Test B durchführen},\\
&F_T^{LP}(t_{10}) = \text{Test C durchführen},\\
&F_T^{LP}(t_{11}) = \tau,\\
&F_T^{LP}(t_{12}) = \text{Ergebnisse prüfen},\\
&F_T^{LP}(t_{13}) = \text{Probe archivieren},\\
&F_T^{LP}(t_{14}) = \text{Ergebnisse versenden},\\
&F_T^{LP}(t_{15}) = \text{Ergebnisse übermitteln},\\
&F_T^{LP}(t_{16}) = \tau
\end{aligned}
$$

$$
\begin{aligned}
&F_{RE}^{LP}(\text{Moritz}) = \{\text{MTA, Verwaltung}\},\\
&F_{RE}^{LP}(\text{Eva}) = \{\text{MTA}\},\\
&F_{RE}^{LP}(\text{Max}) = \{\text{MTA}\},\\
&F_{RE}^{LP}(\text{Laborgerät 1}) = \{\text{Labor A}\},\\
&F_{RE}^{LP}(\text{Laborgerät 2}) = \{\text{Labor B}\},\\
&F_{RE}^{LP}(\text{Laborinformationssystem}) = \{\text{LIS}\}
\end{aligned}
$$

$$
\begin{aligned}
&F_{RO}^{LP}(\text{Verwaltung}) = \{t_1, t_2, t_3, t_{13}, t_{14}\},\\
&F_{RO}^{LP}(\text{MTA}) = \{t_6, t_7, t_8, t_{12}\},\\
&F_{RO}^{LP}(\text{Labor A}) = \{t_5, t_9\},\\
&F_{RO}^{LP}(\text{Labor B}) = \{t_{10}\},\\
&F_{RO}^{LP}(\text{Laborinformationssystem}) = \{t_4, t_{15}\}
\end{aligned}
$$

Im Anschluss werden die für die Evaluation benötigten Ereignisdaten bzw. Ereignisprotokolle mit Hilfe der computergestützten Simulation erzeugt.

Definition 8.2: Die Simulation ist „das Nachbilden eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind. […] Im weiteren Sinne wird unter Simulation das Vorbereiten, Durchführen und Auswerten gezielter Experimente mit einem Simulationsmodell verstanden“ (VDI, 1993).

Definition 8.3: Ein Simulationsexperiment ist „die zielgerichtete Untersuchung des Verhaltens eines Simulationsmodells durch wiederholte Ausführung von Simulationen mit systematischen Parameter- und/oder Strukturvariationen“ (VDI, 1993).

Erfolgt ein Simulationsexperiment mit Hilfe eines Rechners bzw. unter Verwendung von Software-Werkzeugen, wird von einer computergestützten bzw. softwaregestützten Simulation (auch Computersimulation oder Rechnersimulation) gesprochen. Die Durchführung einer computergestützten Simulation kann entweder interaktiv oder automatisiert erfolgen (VDI, 1993; Oberweis, 1996; van der Aalst, 2010).

Neben den im annotierten Workflow-Netz WN_A^{LP} abgebildeten Perspektiven (Kontrollfluss, Ressourcen und Rollen/Organisation) können den im Horus Business Modeler modellierten Ablaufmodellen weitere Eigenschaften, wie Bearbeitungszeiten und -kosten (z.B. für Aktivitäten) sowie Entscheidungsregeln (z.B. in Form von Wahrscheinlichkeiten für Verbindungen bzw. Ablaufalternativen) zugewiesen werden (van der Aalst, 2010; Schönthaler, Vossen, Oberweis, & Karle, 2011). Im Folgenden wird das im Horus Business Modeler als Ablaufmodell modellierte, annotierte Workflow-Netz WN_A^{LP} als Simulationsmodell verwendet.

Die Durchführung der computergestützten Simulation erfolgt mit Hilfe der Simulationskomponente des Horus Business Modeler. Über einen zugehörigen Dialog kann das Simulationsexperiment konfiguriert bzw. parametrisiert sowie die Anzahl der zu erzeugenden Instanzen festgelegt werden (hier: 100). Neben einer Wahrscheinlichkeitsverteilung (hier: Exponentialverteilung) kann eine Ankunftsrate für die von der Simulationskomponente zu erzeugenden Instanzen angegeben werden (hier: 10 Minuten). Die Ankunftsrate bestimmt, in welchem Zeitintervall neue Instanzen des gewählten Ablaufmodells erzeugt werden (Abbildung 8.2). Über einen weiteren Dialog (Reiter: Optionen) kann zudem ein Start- und ein Endzeitpunkt für den Simulationszeitraum festgelegt werden.

In der Folge werden die Simulationsexperimente so konfiguriert, dass nur die im Ablaufmodell angegebenen, festen Werte (z.B. für Bearbeitungszeiten und -kosten) angenommen und vorhandene Nullwerte durch den Wert „0“ ersetzt werden.

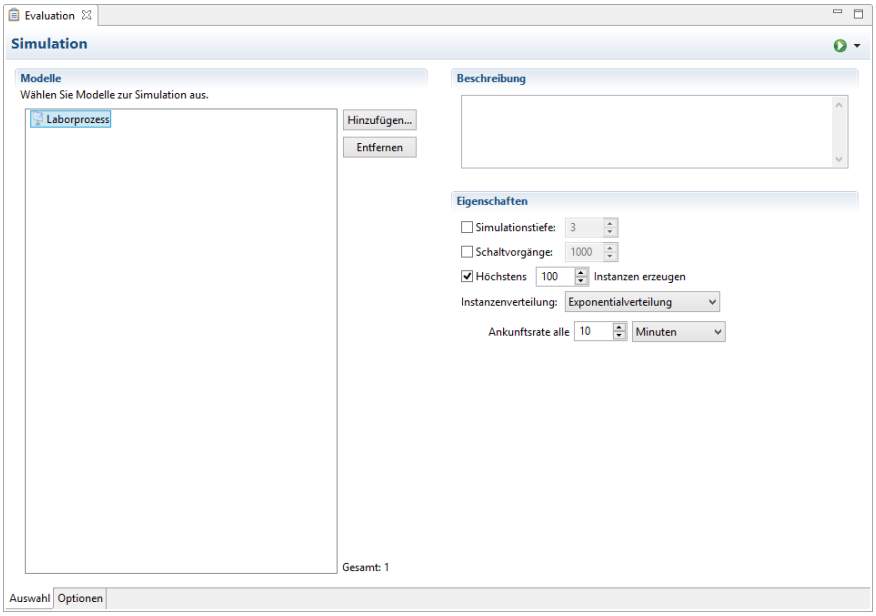

Abbildung 8.2: Konfiguration eines Simulationsexperiments im Horus Business Modeler

Das Ergebnis eines Simulationsexperiments wird in Form eines XML-basierten Simulations-Trace ausgegeben (Tabelle 8.1). Das trace-Element enthält neben den für das Ablaufmodell verfügbaren Ressourcen (inklusive ihrer Rollen) eine Protokollierung der in den einzelnen Instanzen des Ablaufmodells ausgeführten Schaltvorgänge (firingEvent-Element, processInstance-Attribut). Des Weiteren enthält das firingEvent-Element Informationen zum Startzeitpunkt (arrivalTime-Attribut), der Bearbeitungszeit (processingTime-Attribut), den verursachten Kosten (costs-Attribut) sowie einen eindeutigen Bezeichner der Transition, die geschaltet hat (transitionId-Attribut). Aus letzterem lässt sich wiederum die Beschriftung der Transition (d.h. der Name der Aktivität) ableiten bzw. ermitteln. Das resource-Element enthält einen Verweis auf die Ressource (id-Attribut), die die Aktivität „ausgeführt“ (processrole-Attribut) hat.

Tabelle 8.1: Ausschnitt eines Simulations-Trace

```
<?xml version="1.0" encoding="UTF-8"?>
<p:trace ...>
  ...
  <p:resources>
    <p:humanresource id="1" name="Moritz" resourcedbid="0"
resourceuri=".../Moritz.employee_def" roledbid="0"
roleuri=".../Verwaltung.role_def"/>
    <p:humanresource id="2" name="Moritz" resourcedbid="0"
resourceuri=".../Moritz.employee_def" roledbid="0" roleuri=".../MTA.role_def"/>
  ...
  </p:resources>
  <p:firingEvent arrivalTime="1388559600000" costs="1.25" diagramid="0"
processInstance="1" processingTime="300000" transitionId="t12345678" ...>
    <p:resource id="1" processrole="operator" workload="100.0"/>
    <p:preplace placeid="p12345678" tchange="1" tokens="1"/>
    <p:postplace placeid="p23456789" tchange="1" tokens="0"/>
  </p:firingEvent>
  ...
</p:trace>
```

Mit Hilfe der von der XES Process Compliance Library zur Verfügung gestellten Funktionalitäten kann das Simulations-Trace im Anschluss in ein XES-konformes Ereignisprotokoll transformiert werden (Tabelle 8.2).

Tabelle 8.2: Ausschnitt eines XES-konformen Ereignisprotokolls

```
<?xml version="1.0" encoding="UTF-8"?>
<log xes.version="1.0" xes.features="nested-attributes" xmlns="http://www.xes-
standard.org/">
  ...
  <trace>
    <string key="concept:name" value="1"/>
    <float key="cost:total" value="0"/>
    <event>
      <string key="concept:instance" value="1"/>
      <string key="concept:name" value="Probe annehmen"/>
      <string key="lifecycle:transition" value="start"/>
      <date key="time:timestamp" value="2014-01-01T07:00:00+01:00"/>
    </event>
    <event>
      <string key="concept:instance" value="1"/>
      <string key="concept:name" value="Probe annehmen"/>
      <string key="lifecycle:transition" value="complete"/>
      <date key="time:timestamp" value="2014-01-01T07:05:00+01:00"/>
      <string key="org:resource" value="Moritz"/>
      <string key="org:role" value="Verwaltung"/>
      <float key="cost:total" value="1.25"/>
    </event>
  ...
  </trace>
  ...
</log>
```

8.1.2 Compliance-Anforderungen und -Regeln

Im Folgenden werden in Anlehnung an die im Rahmen des Industrieprojekts gewonnenen Einblicke beispielhafte Compliance-Anforderungen, -Kontrollen und -Regeln für den Laborprozess definiert. Die Beziehungen und Abhängigkeiten können mit Hilfe von Compliance-Modellen im Process Compliance Manager abgebildet werden:

- Die erste Anforderung legt fest, dass die Durchlaufzeit einer Prozessinstanz nicht mehr als eine Stunde betragen soll. Für die Anforderung wird eine automatisierte Kontrolle definiert. Im Rahmen der Kontrolle wird geprüft, ob die Durchlaufzeit einer Prozessinstanz kleiner oder gleich einer Stunde gewesen ist.

- Die zweite Anforderung legt fest, dass die Kosten für die Untersuchung einer Probe 20 Euro nicht überschreiten sollten. Für die Anforderung wird eine automatisierte Kontrolle definiert. Im Rahmen der Kontrolle wird geprüft, ob die Kosten einer Prozessinstanz kleiner oder gleich 20 Euro gewesen sind.
- Die dritte Compliance-Anforderung legt fest, dass für jede Probe so früh wie möglich ein Barcode gedruckt und an der Probe angebracht werden muss, um diese in der Folge eindeutig identifizieren zu können. Für die Anforderung wird eine automatisierte Kontrolle definiert. Im Rahmen der Kontrolle wird geprüft, ob direkt nach der Aktivtiät „Probe annehmen“ die Aktivität „Barcode drucken“ ausgeführt wurde.
- Die vierte Anforderung legt fest, dass eine Probe vor der Durchführung von Test A zunächst zentrifugiert werden muss. Des Weiteren darf der Test aufgrund der für die Durchführung benötigten spezifischen Qualifikation nur von bestimmten Mitarbeitern durchgeführt werden. Des Weiteren sollte die Bearbeitungszeit 20 Minuten nicht überschreiten. Für die Anforderung werden drei automatisierte Kontrollen definiert. Im Rahmen der ersten Kontrolle wird geprüft, ob die Aktivität „Probe zentrifugieren“ vor der Aktivität „Test A durchführen“ ausgeführt wurde. Darüber hinaus wird geprüft, ob die Aktivität „Test A durchführen“ von der Mitarbeiterin „Eva“ ausgeführt wurde und ob die Bearbeitungszeit kleiner oder gleich 20 Minuten gewesen ist.
- Die fünfte Anforderung legt fest, dass eine Probe vor der Durchführung der Tests B und C zunächst aliquotiert werden muss. Für die Anforderung wird eine automatisierte Kontrolle definiert. Im Rahmen der Kontrolle wird geprüft, ob die Aktivität „Probe aliquotieren“ sowohl vor der Aktivität „Test A durchführen“ als auch der Aktivität „Test B durchführen“ ausgeführt wurde.
- Die sechste Anforderung legt fest, dass die Untersuchungsergebnisse erst nach einer entsprechenden Überprüfung versendet bzw. elektronisch übermittelt werden dürfen. Für die Anforderung wird eine automatisierte Kontrolle definiert. Im Rahmen der Kontrolle wird geprüft, ob die Aktivität „Ergebnisse prüfen“ sowohl vor der Aktivität „Ergebnisse versenden“ als auch der Aktivität „Ergebnisse übermitteln“ ausgeführt wurde.

In Tabelle 8.3 werden die einzelnen Compliance-Anforderungen, -Kontrollen und parametrisierten Compliance-Regeln sowie deren Beziehungen und Abhängigkeiten dargestellt.

Tabelle 8.3: Compliance-Anforderungen, -Kontrollen und -Regeln

Compliance-Anforderung	Compliance-Kontrolle	Compliance-Regel
Anforderung 1	Kontrolle 1	Regel 1: ProcessLeadTimeLessEqual(„PT1H“)
Anforderung 2	Kontrolle 2	Regel 2: ProcessCostLessEqual(20.0)
Anforderung 3	Kontrolle 3	Regel 3: ActivityResponseDirect(„Probe annehmen“, „Barcode drucken“)
Anforderung 4	Kontrolle 4	Regel 4: ActivityPrecedenceIndirect(„Test A durchführen“, „Probe zentrifugieren“)
	Kontrolle 5	Regel 5: ActivityResourceBonded(„Test A durchführen“, „Eva“)
	Kontrolle 6	Regel 6: ActivityServiceTimeLessEqual(„Test A durchführen“, „PT20M“)
Anforderung 5	Kontrolle 7	Regel 7: ActivityPrecedenceIndirect(„Test B durchführen“, „Probe aliquotieren“)
		Regel 8: ActivityPrecedenceIndirect(„Test C durchführen“, „Probe aliquotieren“)
Anforderung 6	Kontrolle 8	Regel 9: ActivityPrecedenceIndirect(„Ergebnisse versenden“, „Ergebnisse prüfen“)
		Regel 10: ActivityPrecedenceIndirect(„Ergebnisse übermitteln“, „Ergebnisse prüfen“)

8.1.3 Ereignisprotokollbasierte Compliance-Prüfung

Ausgehend von den für den Laborprozess beispielhaft definierten Compliance-Anforderungen, -Kontrollen und -Regeln wird im Folgenden die Durchführung eines Compliance-Audits bzw. einer ereignisprotokollbasierten Compliance-Prüfung beschrieben.

Hierfür werden zunächst für die im Laborprozess modellierten Ablaufalternativen (erster Fall: Test A durchführen, zweiter Fall: Test B und Test C durchführen) Simulationsexperimente mit je 100 Instanzen durchgeführt sowie XES-konforme Ereignisprotokolle erzeugt. Jede Instanz enthält dabei zwischen 20 und 30 Ereignisse (d.h. zwischen 10 und 15 Aktivitäten). Im Anschluss werden verschiedene Compliance-Verletzungen in die Ereignisprotokolle eingefügt, um die Funktionsfähigkeit bzw. Korrektheit des im Rahmen dieser Arbeit entwickelten Ansatzes bzw. der Referenzimplementierung zu evaluieren. In den folgenden Tabellen werden die Ergebnisse der durchgeführten Compliance-Audits bzw. der ereignisprotokollbasierten Compliance-Prüfungen aufgeführt (Tabelle 8.4 und Tabelle 8.5).

Tabelle 8.4: Ergebnisse des Compliance-Audits – Erster Fall

Compliance-Anforderung	Compliance-Kontrolle	Compliance-Regel	Anzahl der Prozessinstanzen, die die Regel erfüllen bzw. verletzen (true/false)
Anforderung 1	Kontrolle 1	Regel 1	90/10
Anforderung 2	Kontrolle 2	Regel 2	85/15
Anforderung 3	Kontrolle 3	Regel 3	90/10
Anforderung 4	Kontrolle 4	Regel 4	95/5
	Kontrolle 5	Regel 5	90/10
	Kontrolle 6	Regel 6	85/15
Anforderung 6	Kontrolle 8	Regel 9	95/5
		Regel 10	95/5

Aufgrund der zuvor definierten Beziehungen und Abhängigkeiten ergibt sich für den ersten Fall (Test A durchführen) das folgende Auditergebnis:

- Compliance-Anforderung 1 wird von 90 Prozessinstanzen erfüllt, 10 Prozessinstanzen verletzen die Anforderung.
- Compliance-Anforderung 2 wird von 85 Prozessinstanzen erfüllt, 15 Prozessinstanzen verletzen die Anforderung.
- Compliance-Anforderung 3 wird von 95 Prozessinstanzen erfüllt, 5 Prozessinstanzen verletzen die Anforderung.
- Compliance-Anforderung 4 wird von 72 Prozessinstanzen erfüllt, 28 Prozessinstanzen verletzen die Anforderung.
- Compliance-Anforderung 6 wird von 91 Prozessinstanzen erfüllt, 9 Prozessinstanzen verletzen die Anforderung.

Der Erfüllungsgrad (compliance degree) der Compliance-Anforderungen beträgt für den ersten Fall 43%, der Verletzungsgrad (violation degree) entspricht 57%.

Tabelle 8.5: Ergebnisse des Compliance-Audits – Zweiter Fall

Compliance-Anforderung	Compliance-Kontrolle	Compliance-Regel	Anzahl der Prozessinstanzen, die die Regel erfüllen bzw. verletzen (true/false)
Anforderung 1	Kontrolle 1	Regel 1	95/5
Anforderung 2	Kontrolle 2	Regel 2	90/10
Anforderung 3	Kontrolle 3	Regel 3	95/5
Anforderung 5	Kontrolle 7	Regel 7	90/10
		Regel 8	95/5
Anforderung 6	Kontrolle 8	Regel 9	90/10
		Regel 10	90/10

Für den zweiten Fall (Test B und Test C durchführen) ergibt sich das folgende Auditergebnis:

- Compliance-Anforderung 1 wird von 95 Prozessinstanzen erfüllt, 5 Prozessinstanzen verletzen die Anforderung.
- Compliance-Anforderung 2 wird von 90 Prozessinstanzen erfüllt, 10 Prozessinstanzen verletzen die Anforderung.
- Compliance-Anforderung 3 wird von 95 Prozessinstanzen erfüllt, 5 Prozessinstanzen verletzen die Anforderung.
- Compliance-Anforderung 5 wird von 84 Prozessinstanzen erfüllt, 16 Prozessinstanzen verletzen die Anforderung.
- Compliance-Anforderung 6 wird von 82 Prozessinstanzen erfüllt, 18 Prozessinstanzen verletzen die Anforderung.

Der Erfüllungsgrad der Compliance-Anforderungen beträgt für den zweiten Fall 54%, der Verletzungsgrad entspricht 46%.

Für den Laborprozess ergibt sich somit ein Gesamterfüllungsgrad (overall compliance degree) von 48,5% bzw. ein Gesamtverletzungsgrad (overall violation degree) von 51,5%; d.h. 48,5% der Prozessinstanzen erfüllen die für den Laborprozess definierten Compliance-Anforderungen. In Abhängigkeit von dem für ein Unternehmen bzw. einen Geschäftsprozess definierten Toleranzbereich können durch die verantwortlichen

Akteure (u.a. Chief Compliance Officer, Compliance Manager, Compliance Auditor) sowohl eine Bewertung der Audit- bzw. Analyseergebnisse vorgenommen als auch konkrete Maßnahmen zur Verbesserung der Compliance-Aktivitäten abgeleitet werden.

Eine im Anschluss mit Hilfe des Process Compliance Dashboard durchgeführte Analyse der Auditgergebnisse zeigt, dass alle in die Ereignisprotokolle eingefügten Compliance-Verletzungen korrekt erkannt werden.

Um die Leistungsfähigkeit des im Rahmen dieser Arbeit entwickelten Ansatzes bzw. der Referenzimplementierung zu evaluieren, werden für beide Ablaufalternativen bzw. Fälle zusätzliche Simulationsexperimente mit 1.000 und 10.000 Instanzen durchgeführt sowie XES-konforme Ereignisprotokolle erzeugt. Im Anschluss werden die Compliance-Audits für die verschiedenen Ereignisprotokollgrößen wiederholt und die Laufzeiten der ereignisprotokollbasierten Compliance-Prüfungen (d.h. die Einzellaufzeiten der Compliance-Regeln sowie die Gesamtlaufzeiten) ermittelt. Die Leistungsmessung erfolgt single-threaded; zur Durchführung der Leistungstests kommt ein Workstation-PC mit Intel Core i7-3770-Prozessor und 16 GB Arbeitsspeicher zum Einsatz. In Tabelle 8.6 werden die Ergebnisse der Leistungsmessung bzw. der Leistungstests aufgeführt.

Tabelle 8.6: Laufzeiten der ereignisprotokollbasierten Compliance-Prüfung

Compliance-Regel	Laborprozess – Erster Fall			Laborprozess – Zweiter Fall		
	100 Traces	1.000 Traces	10.000 Traces	100 Traces	1.000 Traces	10.000 Traces
Regel 1	0,04s	0,19s	2,59s	0,04s	0,19s	2,51s
Regel 2	0,03s	0,15s	2,18s	0,03s	0,15s	2,21s
Regel 3	0,05s	0,32s	3,90s	0,05s	0,29s	3,61s
Regel 4	0,06s	0,30s	3,69s	-	-	-
Regel 5	0,05s	0,28s	3,43s	-	-	-
Regel 6	0,07s	0,41s	4,65s	-	-	-
Regel 7	-	-	-	0,05s	0,29s	3,62s
Regel 8	-	-	-	0,05s	0,31s	3,67s
Regel 9	0,06s	0,34s	4,10s	0,05s	0,30s	3,67s
Regel 10	0,06s	0,35s	4,08s	0,05s	0,29s	3,68s
Gesamtlaufzeit	0,42s	2,34s	28,62s	0,32s	1,82s	22,97s

Darüber hinaus wird eine vergleichende Untersuchung mit dem im Software-Werkzeug ProM zur Verfügung stehenden Plugin LTL Checker vorgenommen, um die individuelle Leistungsfähigkeit beider Ansätze bzw. ihrer Implementierungen zu ermitteln. Hierzu werden einige in dem Plugin LTL Checker enthaltenen Regeln ausgewählt und eine erneute Compliance-Prüfung des Ereignisprotokolls (Laborprozess – Erster Fall) durchgeführt. In Tabelle 8.7 werden die Ergebnisse der vergleichenden Untersuchung dargestellt.

Tabelle 8.7: Laufzeiten der vergleichenden Untersuchung mit ProM

Compliance-Regel	Laborprozess – Erster Fall			
	1.000 Traces		10.000 Traces	
	ProM LTL Checker	XPCL	ProM LTL Checker	XPCL
„Eventually A“	0,9s	0,1s	4,8s	2,1s
„A next B“	0,9s	0,3s	5,0s	4,0s
„A next B next C“	1,0s	0,3s	5,3s	4,3s
„A then B“	0,9s	0,3s	5,1s	4,0s
„A then B then C“	1,0s	0,6s	5,3s	6,6s
„First A“	0,9s	0,2s	4,5s	2,1s
„Last A“	1,0s	0,2s	5,3s	3,1s
„P does A“	1,0s	0,4s	4,9s	4,9s

Die Ergebnisse der Leistungsmessungen zeigen, dass der in dieser Arbeit entwickelte Ansatz bzw. die Referenzimplementierung für die im Rahmen des Anwendungsbeispiels und der vergleichenden Untersuchung betrachteten Compliance-Anforderungen bzw. -Regeln eine effiziente Compliance-Prüfung ermöglicht.

8.2 Grenzen und kritische Diskussion

Im Folgenden werden die Grenzen der in dieser Arbeit entwickelten Lösung aufgezeigt und die Ergebnisse der Evaluation kritisch diskutiert.

Aufbauend auf einer Literaturrecherche werden in Kapitel 5 ein Ansatz zur Spezifikation semiformaler Compliance-Regeln vorgestellt sowie verschiedene generische Regeln beschrieben. Dabei stellen die in dieser Arbeit identifizierten, musterbasierten

Compliance-Regeln eine initiale Regelmenge dar; eine Überprüfung auf Vollständigkeit der Compliance-Regeln wird nicht vorgenommen. Grundsätzlich erlaubt der entwickelte Ansatz in Abhängigkeit von konkreten Compliance-Anforderungen die Definition und Umsetzung weiterer Compliance-Regeln. So können neben verschiedenen Sicherheitseigenschaften (u.a. Binding of Duty, BoD bzw. Separation of Duty, SoD) weitere, z.B. im Rechnungswesen zum Einsatz kommende Best Practices (u.a. Two-Way-, Three-Way- oder Four-Way-Match) abgebildet werden (Crampton, 2004; Schefer, Strembeck, & Mendling, 2011; Schefer, Strembeck, Mendling, & Baumgrass, 2011; Accorsi & Stocker, 2012; Stocker & Accorsi, 2013). Hierfür sind ggf. zusätzliche Ereignisdaten bzw. Informationen in den Ereignis- protokollen standardisiert aufzuzeichnen bzw. zu speichern. Dabei erlaubt XQuery als Abfragesprache für XML-Dokumente grundsätzlich den Zugriff auf alle in einem XML-Dokument gespeicherten Daten. Bei zunehmender Länge aufgezeichneter Prozessinstanzen sowie einer hohen Anzahl insbesondere rekursiver Knotentests kann es u.U. zu Leistungseinbußen kommen; d.h. es besteht die Möglichkeit, derart komplexe Compliance-Regeln zu spezifizieren, für die nur eine ineffiziente Implementierung erfolgen kann. Darüber hinaus erfolgt keine prozessinstanzübergreifende Compliance-Prüfung.

Folgende Annahmen werden im Rahmen dieser Arbeit getroffen:

- Es wird keine Überprüfung auf Redundanz- und/oder Konfliktfreiheit hinsichtlich der zu prüfenden Compliance-Regeln vorgenommen, d.h. es obliegt den im Rahmen des Compliance Managements beteiligten bzw. verantwortlichen Akteuren sicherzustellen, dass keine redundanten und/oder konfliktären Compliance-Anforderungen, -Kontrollen und -Regeln definiert werden. Eine (automatisierte) Redundanz- und/oder Konfliktbehandlug sollte bereits so früh wie möglich, d.h. bereits auf Ebene der Compliance-Modelle bzw. im Rahmen der Parametrisierung der Compliance-Regeln erfolgen (Awad, 2010; El Kharbili, 2012).
- Die im Rahmen eines Compliance-Audits zu prüfenden Abläufe (d.h. Prozessinstanzen) sowie Ereignisprotokolle enthalten eine endliche Menge an Ereignissen und liegen im standardisierten XES-Format vor. Vor der Durchführung einer ereignisprotokollbasierten Compliance-Prüfung ist sicherzustellen, dass es sich um ein zu dem Geschäftsprozess korrespondierendes Ereignisprotokoll handelt. Des Weiteren ist sicherzustellen, dass die im Ereignisprotokoll aufgezeichneten bzw. gespeicherten Aktivitäts-, Rollen- oder Ressourcennamen oder -bezeichnungen den in den Compliance-Regeln verwendeten Parametern entsprechen. Eine derartige Überprüfung kann einerseits beim Einlesen bzw. Import der Ereignisprotokolle, andererseits vor der Durchführung eines Compli-

ance-Audits bzw. einer ereignisprotokollbasierten Compliance-Prüfung vorgenommen werden.

- Ein Geschäftsprozess bzw. -modell kann verschiedene Ablaufalternativen in Form von exklusiven Verzweigungen enthalten. Werden die Instanzen der verschiedenen Abläufe in einem gemeinsamen Ereignisprotokoll aufgezeichnet bzw. gespeichert, sind zusätzliche Ereignisdaten bzw. Informationen (z.B. mit Hilfe der Concept Extension des XES-Standards) festzuhalten, um eine entsprechende Filterung des Ereignisprotokolls zu ermöglichen. Ohne eine Unterscheidungsmöglichkeit kann es zu unerwünschten Effekten und damit fehlerhaften Auditergebnissen kommen. Aus diesem Grund werden für beide im Anwendungsbeispiel modellierten Ablaufalternativen je ein Simulationsexperiment sowie separate Compliance-Audits durchgeführt.

Im Rahmen der Evaluation wird gezeigt, dass der entwickelte Ansatz bzw. die XES Process Compliance Library alle in die Ereignisprotokolle eingefügten Compliance-Verletzungen korrekt erkennt sowie eine (semi-)automatisierte Durchführung von Compliance-Audits bzw. ereignisprotokollbasierter Compliance-Prüfungen ermöglicht. Dabei bestätigen die Leistungsmessungen bzw. Leistungstests die Ergebnisse der von Sylvain Hallé und Roger Villemaire im Rahmen ihrer Arbeiten durchgeführten, experimentellen Untersuchungen (Hallé & Villemaire, 2008b; Hallé, Villemaire, & Cherkaoui, 2009; Vallet, Mrad, Hallé, & Beaudet, 2013).

9 Zusammenfassung und Ausblick

In diesem Kapitel werden die Ergebnisse der Arbeit zusammengefasst, und es wird ein Ausblick auf zukünftige Forschungsarbeiten gegeben.

Einleitend wurde in Kapitel 1 anhand einer Beschreibung der Ausgangslage auf relevante Problemstellungen eingegangen und das Thema der Arbeit motiviert. Nachfolgend wurde auf die Zielsetzung der Arbeit eingegangen. Gesamtziel der vorliegenden Arbeit war es, eine Methode und Werkzeuge für eine geeignete informationstechnologische Unterstützung des Compliance Managements für Geschäftsprozesse zu entwickeln.

In Kapitel 2 wurde ein Überblick über den Stand der Forschung sowie eine Definition des Begriffes Governance, Risk Management und Compliance (GRC) gegeben. Im Anschluss wurden die einzelnen Disziplinen separat vorgestellt und wesentliche Begriffe eingeordnet. Abschließend wurde auf GRC im Kontext von Informationstechnologien eingegangen.

In Kapitel 3 und 4 wurden Grundlagen des Geschäftsprozessmanagements und der Geschäftsprozessmodellierung behandelt und die wesentlichen Begriffe definiert. In diesem Zusammenhang wurde auf ausgewählte Modellierungssprachen und -notationen sowie auf das Konzept der Workflow Patterns eingegangen. Einzelne in Kapitel 3 und 4 behandelte Konzepte wurden in den nachfolgenden Kapiteln wieder aufgegriffen.

In Kapitel 5 wurde, aufbauend auf den Grundlagen von GRC sowie in Anlehnung an das Requirements Engineering, der Begriff der Compliance-Anforderung definiert sowie ein Lebenszyklus für das Compliance Management vorgestellt. Nachfolgend wurde ein musterbasierter Ansatz zur semiformalen Spezifikation von Compliance-Regeln vorgestellt sowie verschiedene generische Regeln identifiziert und eine Einordnung anhand verschiedener Kategorien vorgenommen. Darüber hinaus wurde mit dem Business Process Compliance Management Model ein konzeptuelles Modell entwickelt, das in der Folge als Basis für die Konzeption und Umsetzung einer informationstechnologischen Unterstützung verwendet wurde.

In Kapitel 6 wurden aufbauend auf einer Literaturrecherche eine Einordnung verschiedener in der Literatur diskutierter Ansätze für die Compliance-Prüfung von Geschäftsprozessen vorgenommen und ausgewählte Arbeiten vorgestellt. Im Anschluss wurde auf die Ereignisprotokollierung im Kontext von Geschäftsprozessen eingegangen, und es wurde eine formale Definition wesentlicher Begriffe gegeben. Mit dem eXtensible

Event Stream (XES) wurde ein XML-basierter Standard zur Ereignisprotokollierung vorgestellt. Im Anschluss wurde ein Ansatz zur ereignisprotokollierten Compliance-Prüfung hergeleitet und eine beispielhafte Implementierung der in Kapitel 5 beschriebenen Regeln präsentiert. Der Ansatz erlaubt durch eine Überführung der zuvor spezifizierten, semiformalen Compliance-Regeln in XQuery-Ausdrücke die Prüfung von auf dem XES-Standard basierenden Ereignisprotokollen.

In Kapitel 7 wurden drei prototypische Software-Komponenten konzipiert bzw. umgesetzt. Mit dem Process Compliance Manager wurde eine Erweiterung für ein Geschäftsprozessmanagement und -modellierungswerkzeug konzipiert, die verschiedene Funktionalitäten zur Verwaltung von Compliance-Anforderungen, -Kontrollen und -Regeln sowie zur Durchführung von Compliance-Audits zur Verfügung stellt. Mit der XES Process Compliance Library wurde eine Referenzimplementierung des in Kapitel 6 vorgestellten Ansatzes entwickelt, die in Verbindung mit dem Process Compliance Manager eine Compliance-Prüfung XES-konformer Ereignisprotokolle ermöglicht. Darüber hinaus wurde mit dem Process Compliance Dashboard ein webbasiertes Werkzeug entwickelt, das eine anschließende Aufbereitung, Darstellung und Analyse der Auditergebnisse erlaubt. Basierend auf den Audit- bzw. Analyseergebnissen können im Anschluss Maßnahmen zur Verbesserung der Compliance-Aktivitäten sowie der Geschäftsprozesse eines Unternehmens abgeleitet werden.

Im abschließenden Kapitel 8 wurde die präsentierte Lösung anhand eines Anwendungsbeispiels evaluiert. Hierzu wurde ein Geschäftsprozess gewählt, der an einen im Rahmen eines Industrieprojekts betrachteten Laborprozess angelehnt ist. Im Anschluss wurde der Laborprozess mit annotierten Workflow-Netzen modelliert und (formal) beschrieben. Mit Hilfe der computergestützten Simulation wurden anschließend XES-konforme Ereignisprotokolle erzeugt. Aufbauend auf den im Rahmen des Industrieprojekts gewonnenen Einblicken wurden beispielhafte Compliance-Anforderungen, -Kontrollen und -Regeln definiert. Mit Hilfe der in Kapitel 7 vorgestellten, prototypischen Umsetzungen wurde im Anschluss ein Compliance-Audit durchgeführt. Es wurde gezeigt, dass der in dieser Arbeit entwickelte Ansatz eine effiziente (semi-)automatisierte Compliance-Prüfung ermöglicht. Abschließend wurden Grenzen aufgezeigt und die Ergebnisse der Evaluation kritisch diskutiert.

Folgende, zukünftige Forschungsarbeiten sind geplant:

- Um ein umfassenderes Anforderungsmanagement zu ermöglichen, soll der Process Compliance Manager um eine Repository-Funktionalität erweitert werden, die neben einer Kategorisierung der einzelnen Compliance-Anforderungen (z.B. nach zugehörigen Compliance-Quellen oder -Risiken) das Durch-

suchen des Repositories ermöglicht. Darüber hinaus soll eine Funktionalität entwickelt werden, die ein Vorschlagssystem (recommender system) für Compliance-Anforderungen (z.B. für bestimmte Anwendungsdomänen) realisiert.

- Die mit Hilfe des Horus Business Modeler (Schönthaler, Vossen, Oberweis, & Karle, 2011) erzeugten Simulations-Traces werden derzeit mit Hilfe der von der Referenzimplementierung XES Process Compliance Library zur Verfügung gestellten Funktionalitäten in XES-konforme Ereignisprotokolle transformiert. Zukünftig soll hierfür die Referenzimplementierung des XES-Standards (OpenXES) verwendet und in das Geschäftsprozessmanagement und -modellierungswerkzeug Horus Business Modeler integriert werden. Darüber hinaus sollen die durch die XES Process Compliance Library bereitgestellten Funktionalitäten als Web Service zur Verfügung gestellt werden, um eine Einbindung in andere Software-Werkzeuge zu erleichtern.
- Zukünftig sollen die einzelnen Rechenaufgaben der Compliance-Prüfung (d.h. die Prüfung der einzelnen Compliance-Regeln) auf mehrere (unabhängige) Rechenknoten verteilt werden. Die einzelnen Prüfergebnisse sollen im Anschluss wieder in einem Compliance Audit Result-Dokument zusammengeführt werden können. Darüber hinaus ist geplant, die Leistungsmessung bzw. die Leistungstests für den im Rahmen dieser Arbeit entwickelten Ansatz mit andernen XML-Datenbanken (z.B. eXist, MarkLogic, Sedna) und/oder XQuery-Prozessoren (z.B. Saxon, Zorba) zu wiederholen.
- Grundsätzlich eignet sich der in dieser Arbeit präsentierte, musterbasierte Ansatz für eine Analyse der Ereignisprotokolle im Rahmen der (Business) Process Intelligence (van der Aalst, 2011). Anhand häufig verwendeter Kennzahlen (Key Perfomance Indicator, KPI) können wiederum entsprechende Muster abgeleitet, in XQuery-Ausdrücke überführt, implementiert und in Form einer separaten Bibliothek bereitgestellt werden.

Anhang A

Nachfolgend sind die ausgefüllten Schablonen der Compliance-Regeln der Kategorien Kontrollfluss, Organisation, Zeit und Kosten aufgeführt.

Kategorie Kontrollfluss

Kurzname: CR-CF-0004	Name: ActivityAtLeast
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a, Ganzzahl i	
Beschreibung: Eine Aktivität a muss innerhalb eines Prozesses p mindestens i-mal vorkommen. Die Compliance-Regel ist verletzt, wenn die Aktivität a weniger als i-mal innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-CF-0005	Name: ActivityAtMost
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a, Ganzzahl i	
Beschreibung: Eine Aktivität a darf innerhalb eines Prozesses p höchstens i-mal vorkommen. Die Compliance-Regel ist verletzt, wenn die Aktivität a mehr als i-mal innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-CF-0006	Name: ActivityExactlyRow
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a, Ganzzahl i	
Beschreibung: Eine Aktivität a muss innerhalb eines Prozesses p genau i-mal hintereinander vorkommen. Die Compliance-Regel ist verletzt, wenn die Aktivität a weniger oder mehr als i-mal hintereinander innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-CF-0007	Name: ActivityAtLeastRow
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a, Ganzzahl i	
Beschreibung: Eine Aktivität a muss innerhalb eines Prozesses p mindestens i-mal hintereinander vorkommen. Die Compliance-Regel ist verletzt, wenn die Aktivität a weniger als i-mal hintereinander innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-CF-0008	Name: ActivityAtMostRow
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a, Ganzzahl i	
Beschreibung: Eine Aktivität a darf innerhalb eines Prozesses p höchstens i-mal hintereinander vorkommen. Die Compliance-Regel ist verletzt, wenn die Aktivität a mehr als i-mal hintereinander innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-CF-0009	Name: ActivityCoexistence
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Innerhalb eines Prozesses p muss ausgehend von der Anzahl der Aktivität a_1 die gleiche Anzahl der Aktivität a_2 vorkommen. Die Compliance-Regel ist verletzt, wenn ausgehend von der Anzahl der Aktivität a_1 nicht die gleiche Anzahl der Aktivität a_2 innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-CF-0010	Name: ActivitySequenceExactly
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Ganzzahl i	
Beschreibung: Die Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) muss innerhalb eines Prozesses p genau i-mal vorkommen. Die Compliance-Regel ist verletzt, wenn die Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ nicht genau i-mal innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-CF-0011	Name: ActivitySequenceAtLeast
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Ganzzahl i	
Beschreibung: Die Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) muss innerhalb eines Prozesses p mindestens i-mal vorkommen. Die Compliance-Regel ist verletzt, wenn die Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ weniger als i-mal innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-CF-0012	Name: ActivitySequenceAtMost
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Ganzzahl i	
Beschreibung: Die Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) darf innerhalb eines Prozesses p höchstens i-mal vorkommen. Die Compliance-Regel ist verletzt, wenn die Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ mehr als i-mal innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-CF-0013	Name: ActivityPrecedenceDirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Innerhalb eines Prozesses p muss jeder Aktivität a_1 eine Aktivität a_2 direkt vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Aktivität a_1 eine Aktivität a_2 nicht direkt vorangeht.	

Kurzname: CR-CF-0014	Name: ActivityPrecedenceIndirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Innerhalb eines Prozesses p muss jeder Aktivität a_1 eine Aktivität a_2 direkt oder indirekt vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Aktivität a_1 keine Aktivität a_2 vorangeht.	

Kurzname: CR-CF-0015	Name: ActivityPrecedenceDirectMultiple
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Innerhalb eines Prozesses p muss jeder Aktivität a_1 entweder eine Aktivität a_1 oder eine Aktivität a_2 direkt vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Aktivität a_1 weder eine Aktivität a_1 noch eine Aktivität a_2 direkt vorangeht.	

Kurzname: CR-CF-0016	Name: ActivityPrecedenceIndirectMultiple
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Innerhalb eines Prozesses p muss jeder Aktivität a_1 entweder eine Aktivität a_1 oder eine Aktivität a_2 direkt oder indirekt vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Aktivität a_1 weder eine Aktivität a_1 noch eine Aktivität a_2 vorangeht.	

Kurzname: CR-CF-0017	Name: ActivityPrecedenceDirectMultipleDifferent
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p muss jeder Aktivität a_1 entweder eine Aktivität a_2 oder eine Aktivität a_3 direkt vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Aktivität a_1 weder eine Aktivität a_2 noch eine Aktivität a_3 direkt vorangeht.	

Kurzname: CR-CF-0018	Name: ActivityPrecedenceIndirectMultipleDifferent
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p muss jeder Aktivität a_1 entweder eine Aktivität a_2 oder eine Aktivität a_3 direkt oder indirekt vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Aktivität a_1 weder eine Aktivität a_2 noch eine Aktivität a_3 vorangeht.	

Kurzname: CR-CF-0019	Name: ActivityPrecedenceNeverDirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Innerhalb eines Prozesses p darf einer Aktivität a_1 keine Aktivität a_2 direkt vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Aktivität a_1 eine Aktivität a_2 direkt vorangeht.	

Kurzname: CR-CF-0020	Name: ActivityPrecedenceNever
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Innerhalb eines Prozesses p darf einer Aktivität a_1 keine Aktivität a_2 vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Aktivität a_1 eine Aktivität a_2 vorangeht.	

Kurzname: CR-CF-0021a	Name: ActivitySequencePrecedenceDirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p muss jeder Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) eine Aktivität a_3 direkt vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ keine Aktivität a_3 direkt vorangeht.	

Kurzname: CR-CF-0021b	Name: ActivitySequencePrecedenceDirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p muss jeder Aktivität a_1 eine Sequenz der Aktivitäten $\langle a_2, a_3 \rangle$ (d.h. a_2 direkt gefolgt von a_3) direkt vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Aktivität a_1 keine Sequenz der Aktivitäten $\langle a_2, a_3 \rangle$ direkt vorangeht.	

Kurzname: CR-CF-0021c	Name: ActivitySequencePrecedenceDirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3, Aktivität a_4	
Beschreibung: Innerhalb eines Prozesses p muss jeder Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) eine Sequenz der Aktivitäten $\langle a_3, a_4 \rangle$ (d.h. a_3 direkt gefolgt von a_4) direkt vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ keine Sequenz der Aktivitäten $\langle a_3, a_4 \rangle$ direkt vorangeht.	

Kurzname: CR-CF-0022a	Name: ActivitySequencePrecedenceIndirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p muss jeder Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) eine Aktivität a_3 direkt oder indirekt vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ keine Aktivität a_3 vorangeht.	

Kurzname: CR-CF-0022b	Name: ActivitySequencePrecedenceIndirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p muss jeder Aktivität a_1 eine Sequenz der Aktivitäten $\langle a_2, a_3 \rangle$ (d.h. a_2 direkt gefolgt von a_3) direkt oder indirekt vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Aktivität a_1 keine Sequenz der Aktivitäten $\langle a_2, a_3 \rangle$ vorangeht.	

Kurzname: CR-CF-0022c	Name: ActivitySequencePrecedenceIndirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3, Aktivität a_4	
Beschreibung: Innerhalb eines Prozesses p muss jeder Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) eine Sequenz der Aktivitäten $\langle a_3, a_4 \rangle$ (d.h. a_3 direkt gefolgt von a_4) direkt oder indirekt vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ keine Sequenz der Aktivitäten $\langle a_3, a_4 \rangle$ vorangeht.	

Kurzname: CR-CF-0023a	Name: ActivitySequencePrecedenceNeverDirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p darf jeder Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) keine Aktivität a_3 direkt vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ eine Aktivität a_3 direkt vorangeht.	

Kurzname: CR-CF-0023b	Name: ActivitySequencePrecedenceNeverDirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p darf jeder Aktivität a_1 keine Sequenz der Aktivitäten $\langle a_2, a_3 \rangle$ (d.h. a_2 direkt gefolgt von a_3) direkt vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Aktivität a_1 eine Sequenz der Aktivitäten $\langle a_2, a_3 \rangle$ direkt vorangeht.	

Kurzname: CR-CF-0023c	Name: ActivitySequencePrecedenceNeverDirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3, Aktivität a_4	
Beschreibung: Innerhalb eines Prozesses p darf jeder Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) keine Sequenz der Aktivitäten $\langle a_3, a_4 \rangle$ (d.h. a_3 direkt gefolgt von a_4) direkt vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ eine Sequenz der Aktivitäten $\langle a_3, a_4 \rangle$ direkt vorangeht.	

Kurzname: CR-CF-0024a	Name: ActivitySequencePrecedenceNever
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p darf jeder Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) keine Aktivität a_3 vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ eine Aktivität a_3 vorangeht.	

Kurzname: CR-CF-0024b	Name: ActivitySequencePrecedenceNever
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p darf jeder Aktivität a_1 keine Sequenz der Aktivitäten $\langle a_2, a_3 \rangle$ (d.h. a_2 direkt gefolgt von a_3) keine vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Aktivität a_1 eine Sequenz der Aktivitäten $\langle a_2, a_3 \rangle$ vorangeht.	

Kurzname: CR-CF-0024c	Name: ActivitySequencePrecedenceNever
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3, Aktivität a_4	
Beschreibung: Innerhalb eines Prozesses p darf jeder Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) keine Sequenz der Aktivitäten $\langle a_3, a_4 \rangle$ (d.h. a_3 direkt gefolgt von a_4) vorangehen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p einer Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ eine Sequenz der Aktivitäten $\langle a_3, a_4 \rangle$ vorangeht.	

Kurzname: CR-CF-0025	Name: ActivityResponseDirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Innerhalb eines Prozesses p muss jede Aktivität a_1 von einer Aktivität a_2 direkt gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a_1 nicht direkt von einer Aktivität a_2 gefolgt wird.	

Kurzname: CR-CF-0026	Name: ActivityResponseIndirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Innerhalb eines Prozesses p muss jede Aktivität a_1 von einer Aktivität a_2 direkt oder indirekt gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a_1 nicht von einer Aktivität a_2 gefolgt wird.	

Kurzname: CR-CF-0027	Name: ActivityResponseDirectMultiple
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Innerhalb eines Prozesses p muss jede Aktivität a_1 entweder von einer Aktivität a_1 oder einer Aktivität a_2 direkt gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a_1 weder von einer Aktivität a_1 noch von einer Aktivität a_2 direkt gefolgt wird.	

Kurzname: CR-CF-0028	Name: ActivityResponseIndirectMultiple
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Innerhalb eines Prozesses p muss jede Aktivität a_1 entweder von einer Aktivität a_1 oder einer Aktivität a_2 direkt oder indirekt gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a_1 weder von einer Aktivität a_1 noch einer Aktivität a_2 gefolgt wird.	

Kurzname: CR-CF-0029	Name: ActivityResponseDirectMultipleDifferent
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p muss jede Aktivität a_1 entweder von einer Aktivität a_2 oder einer Aktivität a_3 direkt gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a_1 weder von einer Aktivität a_2 noch von einer Aktivität a_3 direkt gefolgt wird.	

Kurzname: CR-CF-0030	Name: ActivityResponseIndirectMultipleDifferent
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p muss jede Aktivität a_1 entweder von einer Aktivität a_2 oder einer Aktivität a_3 direkt oder indirekt gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a_1 weder von einer Aktivität a_2 noch von einer Aktivität a_3 gefolgt wird.	

Kurzname: CR-CF-0031	Name: ActivityResponseNeverDirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Innerhalb eines Prozesses p darf eine Aktivität a_1 nie von einer Aktivität a_2 direkt gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a_1 von einer Aktivität a_2 direkt gefolgt wird.	

Kurzname: CR-CF-0032	Name: ActivityResponseNever
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Innerhalb eines Prozesses p darf eine Aktivität a_1 nie von einer Aktivität a_2 gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a_1 von einer Aktivität a_2 gefolgt wird.	

Kurzname: CR-CF-0033a	Name: ActivitySequenceResponseDirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p muss jede Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) von einer Aktivität a_3 direkt gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ nicht von einer Aktivität a_3 direkt gefolgt wird.	

Kurzname: CR-CF-0033b	Name: ActivitySequenceResponseDirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p muss jede Aktivität a_1 von einer Sequenz der Aktivitäten $\langle a_2, a_3 \rangle$ (d.h. a_2 direkt gefolgt von a_3) direkt gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a_1 nicht von einer Sequenz der Aktivitäten $\langle a_2, a_3 \rangle$ direkt gefolgt wird.	

Kurzname: CR-CF-0033c	Name: ActivitySequenceResponseDirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3, Aktivität a_4	
Beschreibung: Innerhalb eines Prozesses p muss jede Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) einer Sequenz der Aktivitäten $\langle a_3, a_4 \rangle$ (d.h. a_3 direkt gefolgt von a_4) direkt gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ nicht von einer Sequenz der Aktivitäten $\langle a_3, a_4 \rangle$ direkt gefolgt wird.	

Kurzname: CR-CF-0034a	Name: ActivitySequenceResponseIndirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p muss jede Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) von einer Aktivität a_3 direkt oder indirekt gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ nicht von einer Aktivität a_3 gefolgt wird.	

Kurzname: CR-CF-0034b	Name: ActivitySequenceResponseIndirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p muss jede Aktivität a_1 von einer Sequenz der Aktivitäten $\langle a_2, a_3 \rangle$ (d.h. a_2 direkt gefolgt von a_3) direkt oder indirekt gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a_1 nicht von einer Sequenz der Aktivitäten $\langle a_2, a_3 \rangle$ gefolgt wird.	

Kurzname: CR-CF-0034c	Name: ActivitySequenceResponseIndirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3, Aktivität a_4	
Beschreibung: Innerhalb eines Prozesses p muss jede Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) von einer Sequenz der Aktivitäten $\langle a_3, a_4 \rangle$ (d.h. a_3 direkt gefolgt von a_4) direkt oder indirekt gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ nicht von einer Sequenz der Aktivitäten $\langle a_3, a_4 \rangle$ gefolgt wird.	

Kurzname: CR-CF-0035a	Name: ActivitySequenceResponseNeverDirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p darf jede Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) nicht von einer Aktivität a_3 direkt gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ von einer Aktivität a_3 direkt gefolgt wird.	

Kurzname: CR-CF-0035b	Name: ActivitySequenceResponseNeverDirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p darf jede Aktivität a_1 nicht von einer Sequenz der Aktivitäten $\langle a_2, a_3 \rangle$ (d.h. a_2 direkt gefolgt von a_3) direkt gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a_1 von einer Sequenz der Aktivitäten $\langle a_2, a_3 \rangle$ direkt gefolgt wird.	

Kurzname: CR-CF-0035c	Name: ActivitySequenceResponseNeverDirect
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3, Aktivität a_4	
Beschreibung: Innerhalb eines Prozesses p darf jede Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) von einer Sequenz der Aktivitäten $\langle a_3, a_4 \rangle$ (d.h. a_3 direkt gefolgt von a_4) direkt gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ von einer Sequenz der Aktivitäten $\langle a_3, a_4 \rangle$ direkt gefolgt wird.	

Kurzname: CR-CF-0036a	Name: ActivitySequenceResponseNever
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p darf jede Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) von von einer Aktivität a_3 gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ von einer Aktivität a_3 gefolgt wird.	

Kurzname: CR-CF-0036b	Name: ActivitySequenceResponseNever
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3	
Beschreibung: Innerhalb eines Prozesses p darf jede Aktivität a_1 nicht von einer Sequenz der Aktivitäten $\langle a_2, a_3 \rangle$ (d.h. a_2 direkt gefolgt von a_3) gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a_1 von einer Sequenz der Aktivitäten $\langle a_2, a_3 \rangle$ gefolgt wird.	

Kurzname: CR-CF-0036c	Name: ActivitySequenceResponseNever
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2, Aktivität a_3, Aktivität a_4	
Beschreibung: Innerhalb eines Prozesses p darf jede Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ (d.h. a_1 direkt gefolgt von a_2) nicht von einer Sequenz der Aktivitäten $\langle a_3, a_4 \rangle$ (d.h. a_3 direkt gefolgt von a_4) gefolgt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Sequenz der Aktivitäten $\langle a_1, a_2 \rangle$ von einer Sequenz der Aktivitäten $\langle a_3, a_4 \rangle$ gefolgt wird.	

Kurzname: CR-CF-0037	Name: ActivityExclusive
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Kommt eine Aktivität a_1 (mindestens einmal) innerhalb eines Prozesses p vor, so darf eine Aktivität a_2 nicht vorkommen. Die Compliance-Regel ist verletzt, wenn sowohl eine Aktivität a_1 als auch eine Aktivität a_2 innerhalb eines Prozesses p vorkommen.	

Kurzname: CR-CF-0038	Name: ActivityMutualExclusive
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Innerhalb eines Prozesses p muss entweder eine Aktivität a_1 oder eine Aktivität a_2 (mindestens einmal) vorkommen. Die Compliance-Regel ist verletzt, wenn beide Aktivitäten innerhalb eines Prozesses p vorkommen oder nicht vorkommen.	

Kurzname: CR-CF-0039	Name: ActivityInclusive
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Kommt eine Aktivität a_1 (mindestens einmal) innerhalb eines Prozesses p vor, so muss eine Aktivität a_2 ebenfalls (mindestens einmal) vorkommen. Die Compliance-Regel ist verletzt, wenn eine Aktivität a_1 innerhalb eines Prozesses p vorkommt, ohne dass auch eine Aktivität a_2 vorkommt.	

Kurzname: CR-CF-0040	Name: ActivityPrerequsisite
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Kommt eine Aktivität a_1 innerhalb eines Prozesses p nicht vor, so darf eine Aktivität a_2 ebenfalls nicht vorkommen. Die Compliance-Regel ist verletzt, wenn eine Aktivität a_2 (mindestens einmal) innerhalb eines Prozesses p vorkommt, ohne dass auch eine Aktivität a_1 (mindestens einmal) vorkommt.	

Kurzname: CR-CF-0041	Name: ActivitySubsitute
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Kommt eine Aktivität a_1 innerhalb eines Prozesses p nicht vor, so muss eine Aktivität a_2 (mindestens einmal) vorkommen (und vice versa). Die Compliance-Regel ist verletzt, wenn keine der beiden Aktivitäten innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-CF-0042	Name: ActivityCorequisite
Kategorie: Kontrollfluss	
Parameter: Prozess p, Aktivität a_1, Aktivität a_2	
Beschreibung: Die Aktivitäten a_1 und a_2 müssen innerhalb eines Prozesses p entweder beide vorkommen oder sie dürfen nicht vorkommen. Die Compliance-Regel ist verletzt, wenn nur eine der beiden Aktiviäten innerhalb eines Prozesses p vorkommt.	

Kategorie Organisation

Kurzname: CR-ORG-0004	Name: ProcessRoleAtLeast
Kategorie: Organisation	
Parameter: Prozess p, Rolle ro, Ganzzahl i	
Beschreibung: Eine Rolle ro muss innerhalb eines Prozesses p mindestens i-mal vorkommen. Die Compliance-Regel ist verletzt, wenn die Rolle ro weniger als i-mal innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-ORG-0005	Name: ProcessRoleAtMost
Kategorie: Organisation	
Parameter: Prozess p, Rolle ro, Ganzzahl i	
Beschreibung: Eine Rolle ro darf innerhalb eines Prozesses p höchstens i-mal vorkommen. Die Compliance-Regel ist verletzt, wenn die Rolle ro mehr als i-mal innerhalb eines p Prozesses vorkommt.	

Kurzname: CR-ORG-0006	Name: ProcessRoleCoexistance
Kategorie: Organisation	
Parameter: Prozess p, Rolle ro_1, Rolle ro_2	
Beschreibung: Die Rollen ro_1 und ro_2 müssen innerhalb eines Prozesses p gleich häufig vorkommen. Die Compliance-Regel ist verletzt, wenn die Rollen ro_1 und ro_2 innerhalb eines Prozesses p nicht gleich häufig vorkommen.	

Kurzname: CR-ORG-0007	Name: ProcessRoleExclusive
Kategorie: Organisation	
Parameter: Prozess p, Rolle ro_1, Rolle ro_2	
Beschreibung: Kommt eine Rolle ro_1 innerhalb eines Prozesses p (mindestens einmal) vor, so darf eine Rolle ro_2 nicht vorkommen. Die Compliance-Regel ist verletzt, wenn sowohl eine Rolle ro_1 als auch eine Rolle ro_2 innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-ORG-0008	Name: ProcessRoleMutualExclusive
Kategorie: Organisation	
Parameter: Prozess p, Rolle ro_1, Rolle ro_2	
Beschreibung: Innerhalb eines Prozesses p muss entweder eine Rolle ro_1 oder eine Rolle ro_2 (mindestens einmal) vorkommen. Die Compliance-Regel ist verletzt, wenn beide Rollen innerhalb eines Prozesses p vorkommen oder nicht vorkommen.	

Kurzname: CR-ORG-0009	Name: ProcessRoleInclusive
Kategorie: Organisation	
Parameter: Prozess p, Rolle ro_1, Rolle ro_2	
Beschreibung: Kommt eine Rolle ro_1 (mindestens einmal) innerhalb eines Prozesses p vor, so muss eine Rolle ro_2 ebenfalls (mindestens einmal) vorkommen. Die Compliance-Regel ist verletzt, wenn eine Rolle ro_1 innerhalb eines Prozesses p vorkommt, ohne dass auch eine Rolle ro_2 vorkommt.	

Kurzname: CR-ORG-0010	Name: ProcessRolePrerequisite
Kategorie: Organisation	
Parameter: Prozess p, Rolle ro_1, Rolle ro_2	
Beschreibung: Kommt eine Rolle ro_1 innerhalb eines Prozesses p nicht vor, so darf eine Rolle ro_2 ebenfalls nicht vorkommen. Die Compliance-Regel ist verletzt, wenn eine Rolle ro_2 (mindestens einmal) innerhalb eines Prozesses p vorkommt, ohne dass auch eine Rolle ro_1 (mindestens einmal) vorkommt.	

Kurzname: CR-ORG-0011	Name: ProcessRoleSubstitue
Kategorie: Organisation	
Parameter: Prozess p, Rolle ro_1, Rolle ro_2	
Beschreibung: Kommt eine Rolle ro_1 innerhalb eines Prozesses p nicht vor, so muss eine Rolle ro_2 (mindestens einmal) vorkommen (und vice versa). Die Compliance-Regel ist verletzt, wenn keine der beiden Rollen innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-ORG-0012	Name: ProcessRoleCorequisite
Kategorie: Organisation	
Parameter: Prozess p, Rolle ro_1, Rolle ro_2	
Beschreibung: Innerhalb eines Prozesses p müssen die Rollen ro_1 und ro_2 entweder beide vorkommen oder beide dürfen nicht vorkommen. Die Compliance-Regel ist verletzt, wenn nur eine der beiden Rollen innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-ORG-0013	Name: ProcessResourceExistence
Kategorie: Organisation	
Parameter: Prozess p, Ressource re	
Beschreibung: Eine Ressource re muss innerhalb eines Prozesses p (mindestens einmal) vorkommen. Die Compliance-Regel ist verletzt, wenn die Ressource re nicht innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-ORG-0014	Name: ProcessResourceAbsence
Kategorie: Organisation	
Parameter: Prozess p, Ressource re	
Beschreibung: Eine Ressource re darf innerhalb eines Prozesses p nicht vorkommen. Die Compliance-Regel ist verletzt, wenn die Ressource re innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-ORG-0015	Name: ProcessResourceExactly
Kategorie: Organisation	
Parameter: Prozess p, Ressource re, Ganzzahl i	
Beschreibung: Eine Ressource re muss innerhalb eines Prozesses p genau i-mal vorkommen. Die Compliance-Regel ist verletzt, wenn die Ressource re weniger oder mehr als i-mal innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-ORG-0016	Name: ProcessResourceAtLeast
Kategorie: Organisation	
Parameter: Prozess p, Ressource re, Ganzzahl i	
Beschreibung: Eine Ressource re muss innerhalb eines Prozesses p mindestens i-mal vorkommen. Die Compliance-Regel ist verletzt, wenn die Ressource re weniger als i-mal innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-ORG-0017	Name: ProcessResourceAtMost
Kategorie: Organisation	
Parameter: Prozess p, Ressource re, Ganzzahl i	
Beschreibung: Eine Ressource re darf innerhalb eines Prozesses p höchstens i-mal vorkommen. Die Compliance-Regel ist verletzt, wenn die Ressource re mehr als i-mal innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-ORG-0018	Name: ProcessResourceCoexistance
Kategorie: Organisation	
Parameter: Prozess p, Ressource re_1, Ressource re_2	
Beschreibung: Die Ressourcen re_1 und re_2 müssen innerhalb eines Prozesses p gleich häufig vorkommen. Die Compliance-Regel ist verletzt, wenn die Ressourcen re_1 und re_2 innerhalb eines Prozesses p nicht gleich häufig vorkommen.	

Kurzname: CR-ORG-0019	Name: ProcessResourceExclusive
Kategorie: Organisation	
Parameter: Prozess p, Ressource re_1, Ressource re_2	
Beschreibung: Kommt eine Ressource re_1 innerhalb eines Prozesses p (mindestens einmal) vor, so darf eine Ressource re_2 nicht vorkommen. Die Compliance-Regel ist verletzt, wenn sowohl eine Ressource re_1 als auch eine Ressource re_2 innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-ORG-0020	Name: ProcessResourceMutualExclusive
Kategorie: Organisation	
Parameter: Prozess p, Ressource re_1, Ressource re_2	
Beschreibung: Innerhalb eines Prozesses p muss entweder eine Ressource re_1 oder eine Ressource re_2 (mindestens einmal) vorkommen. Die Compliance-Regel ist verletzt, wenn beide Ressourcen innerhalb eines Prozesses p vorkommen oder nicht vorkommen.	

Kurzname: CR-ORG-0021	Name: ProcessResourceInclusive
Kategorie: Organisation	
Parameter: Prozess p, Ressource re_1, Ressource re_2	
Beschreibung: Kommt eine Ressource re_1 (mindestens einmal) innerhalb eines Prozesses p vor, so muss eine Ressource re_2 ebenfalls (mindestens einmal) vorkommen. Die Compliance-Regel ist verletzt, wenn eine Ressource re_1 innerhalb eines Prozesses p vorkommt, ohne dass auch eine Ressource re_2 vorkommt.	

Kurzname: CR-ORG-0022	Name: ProcessResourcePrerequisite
Kategorie: Organisation	
Parameter: Prozess p, Ressource re_1, Ressource re_2	
Beschreibung: Kommt eine Ressource re_1 innerhalb eines Prozesses p nicht vor, so darf eine Ressource re_2 ebenfalls nicht vorkommen. Die Compliance-Regel ist verletzt, wenn eine Ressource re_2 (mindestens einmal) innerhalb eines Prozesses p vorkommt, ohne dass auch eine Ressource re_1 (mindestens einmal) vorkommt.	

Kurzname: CR-ORG-0023	Name: ProcessResourceSubstitute
Kategorie: Organisation	
Parameter: Prozess p, Ressource re_1, Ressource re_2	
Beschreibung: Kommt eine Ressource re_1 innerhalb eines Prozesses p nicht vor, so muss eine Ressource re_2 mindestens einmal) vorkommen (und vice versa). Die Compliance-Regel ist verletzt, wenn keine der beiden Ressourcen innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-ORG-0024	Name: ProcessResourceCorequisite
Kategorie: Organisation	
Parameter: Prozess p, Ressource re_1, Ressource re_2	
Beschreibung: Innerhalb eines Prozesses p müssen die Ressource re_1 und re_2 entweder beide vorkommen oder beide dürfen nicht vorkommen. Die Compliance-Regel ist verletzt, wenn nur eine der beiden Rollen innerhalb eines Prozesses p vorkommt.	

Kurzname: CR-ORG-0025	Name: ActivityRoleBonded
Kategorie: Organisation	
Parameter: Prozess p, Aktivität a, Rolle ro	
Beschreibung: Innerhalb eines Prozesses p muss eine Aktivität a von einer Ressource mit der Rolle ro ausgeführt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a nicht von einer Ressource mit der Rolle ro ausgeführt wird.	

Kurzname: CR-ORG-0026	Name: ActivityRoleSegregated
Kategorie: Organisation	
Parameter: Prozess p, Aktivität a, Rolle ro	
Beschreibung: Innerhalb eines Prozesses p darf eine Aktivität a nicht von einer Ressource mit der Rolle ro ausgeführt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a von einer Ressource mit der Rolle ro ausgeführt wird.	

Kurzname: CR-ORG-0027	Name: ActivityResourceBonded
Kategorie: Organisation	
Parameter: Prozess p, Aktivität a, Ressource re	
Beschreibung: Innerhalb eines Prozesses p muss eine Aktivität a von einer Ressource re ausgeführt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a nicht von einer Ressource re ausgeführt wird.	

Kurzname: CR-ORG-0028	Name: ActivityResourceSegregated
Kategorie: Organisation	
Parameter: Prozess p, Aktivität a, Ressource re	
Beschreibung: Innerhalb eines Prozesses p darf eine Aktivität a nicht von einer Ressource re ausgeführt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a von einer Ressource re ausgeführt wird.	

Kurzname: CR-ORG-0029	Name: ActivityRoleBondedResourceBonded
Kategorie: Organisation	
Parameter: Prozess p, Aktivität a, Rolle ro, Ressource re	
Beschreibung: Innerhalb eines Prozesses p muss eine Aktivität a von einer Ressource re mit der Rolle ro ausgeführt werden. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a nicht von einer Ressource re mit der Rolle ro ausgeführt wird.	

Kurzname: CR-ORG-0030	Name: ActivityRoleBondedResourceSegregated
Kategorie: Organisation	
Parameter: Prozess p, Aktivität a, Rolle ro, Ressource re	
Beschreibung: Innerhalb eines Prozesses p muss eine Aktivität a von einer Ressource mit der Rolle ro ausgeführt werden; dabei darf es sich nicht um die Ressource re handeln. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a von einer Ressource re ausgeführt wird oder der Ressource nicht die Rolle ro zugewiesen ist.	

Kurzname: CR-ORG-0031	Name: ActivityRoleSegregatedResourceBonded
Kategorie: Organisation	
Parameter: Prozess p, Aktivität a, Rolle ro, Ressource re	
Beschreibung: Innerhalb eines Prozesses p muss eine Aktivität a von einer Ressource re ausgeführt werden; dabei darf der Ressource re nicht die Rolle ro zugewiesen sein. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a nicht von einer Ressource re ausgeführt wird oder der Ressource re die Rolle ro zugewiesen ist.	

Kategorie Zeit

Kurzname: CR-TIME-0004	Name: ProcessLeadTimeLessOrEqual
Kategorie: Zeit	
Parameter: Prozess p, Zeitdauer d	
Beschreibung: Die Durchlaufzeit eines Prozesses p muss kleiner als oder gleich d Zeiteinheiten sein. Die Compliance-Regel ist verletzt, wenn die Durchlaufzeit des Prozesses p größer als d Zeiteinheiten ist.	

Kurzname: CR-TIME-0005	Name: ProcessLeadTimeGreaterOrEqual
Kategorie: Zeit	
Parameter: Prozess p, Zeitdauer d	
Beschreibung: Die Durchlaufzeit eines Prozesses p muss gleich oder größer als d Zeiteinheiten sein. Die Compliance-Regel ist verletzt, wenn die Durchlaufzeit des Prozesses p kleiner als d Zeiteinheiten ist.	

Kurzname: CR-TIME-0006	Name: ActivityStartExactly
Kategorie: Zeit	
Parameter: Prozess p, Aktivität a, Zeitpunkt t	
Beschreibung: Innerhalb eines Prozesses p muss der Start einer Aktivität a genau zum Zeitpunkt t erfolgen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p der Start einer Aktivität a nicht genau zum Zeitpunkt t erfolgt.	

Kurzname: CR-TIME-0007	Name: ActivityStartBefore
Kategorie: Zeit	
Parameter: Prozess p, Aktivität a, Zeitpunkt t	
Beschreibung: Innerhalb eines Prozesses p muss der Start einer Aktivität a vor dem Zeitpunkt t erfolgen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p der Start einer Aktivität a genau zum oder nach dem Zeitpunkt t erfolgt.	

Kurzname: CR-TIME-0008	Name: ActivityStartAfter
Kategorie: Zeit	
Parameter: Prozess p, Aktivität a, Zeitpunkt t	
Beschreibung: Innerhalb eines Prozesses p muss der Start einer Aktivität a nach dem Zeitpunkt t erfolgen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p der Start einer Aktivität a vor dem oder genau zum Zeitpunkt t erfolgt.	

Kurzname: CR-TIME-0009	Name: ActivityFinishExactly
Kategorie: Zeit	
Parameter: Prozess p, Aktivität a, Zeitpunkt t	
Beschreibung: Innerhalb eines Prozesses p muss eine Aktivität a genau zum Zeitpunkt t abgeschlossen sein. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a nicht genau zum Zeitpunkt t abgeschlossen ist.	

Kurzname: CR-TIME-0010	Name: ActivityFinishBefore
Kategorie: Zeit	
Parameter: Prozess p, Aktivität a, Zeitpunkt t	
Beschreibung: Innerhalb eines Prozesses p muss eine Aktivität a vor dem Zeitpunkt t abgeschlossen sein. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a genau zum oder nach dem Zeitpunkt t abgeschlossen ist.	

Kurzname: CR-TIME-0011	Name: ActivityFinishAfter
Kategorie: Zeit	
Parameter: Prozess p, Aktivität a, Zeitpunkt t	
Beschreibung: Innerhalb eines Prozesses p muss eine Aktivität a nach dem Zeitpunkt t abgeschlossen sein. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p eine Aktivität a vor dem oder genau zum Zeitpunkt t abgeschlossen ist.	

Kurzname: CR-TIME-0012	Name: ActivityStartBeforeFinishBefore
Kategorie: Zeit	
Parameter: Prozess p, Aktivität a, Zeitpunkt t_1, Zeitpunkt t_2	
Beschreibung: Innerhalb eines Prozesses p muss der Start einer Aktivität a vor dem Zeitpunkt t_1 erfolgen; die Aktivität a muss vor dem Zeitpunkt t_2 abgeschlossen sein. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p der Start einer Aktivität a genau zum oder nach dem Zeitpunkt t_1 erfolgt und/oder die Aktivität a genau zum oder nach dem Zeitpunkt t_2 abgeschlossen ist.	

Kurzname: CR-TIME-0013	Name: ActivityStartBeforeFinishAfter
Kategorie: Zeit	
Parameter: Prozess p, Aktivität a, Zeitpunkt t_1, Zeitpunkt t_2	
Beschreibung: Innerhalb eines Prozesses p muss der Start einer Aktivität a vor dem Zeitpunkt t_1 erfolgen; die Aktivität a muss nach dem Zeitpunkt t_2 abgeschlossen sein. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p der Start einer Aktivität a genau zum oder nach dem Zeitpunkt t_1 erfolgt und/oder die Aktivität a vor dem oder genau zum Zeitpunkt t_2 abgeschlossen ist.	

Kurzname: CR-TIME-0014	Name: ActivityStartAfterFinishBefore
Kategorie: Zeit	
Parameter: Prozess p, Aktivität a, Zeitpunkt t_1, Zeitpunkt t_2	
Beschreibung: Innerhalb eines Prozesses p muss der Start einer Aktivität a nach dem Zeitpunkt t_1 erfolgen; die Aktivität a muss vor dem Zeitpunkt t_2 abgeschlossen sein. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p der Start einer Aktivität a vor dem oder genau zum Zeitpunkt t_1 erfolgt und/oder die Aktivität a genau zum oder nach dem Zeitpunkt t_2 abgeschlossen ist.	

Kurzname: CR-TIME-0015	Name: ActivityStartAfterFinishAfter
Kategorie: Zeit	
Parameter: Prozess p, Aktivität a, Zeitpunkt t_1, Zeitpunkt t_2	
Beschreibung: Innerhalb eines Prozesses p muss der Start einer Aktivität a nach dem Zeitpunkt t_1 erfolgen; die Aktivität a muss nach dem Zeitpunkt t_2 abgeschlossen sein. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p der Start einer Aktivität a vor dem oder genau zum Zeitpunkt t_1 erfolgt und/oder die Aktivität a vor dem oder genau zum Zeitpunkt t_2 abgeschlossen ist.	

Kurzname: CR-TIME-0016	Name: ActivityServiceTimeEqual
Kategorie: Zeit	
Parameter: Prozess p, Aktivität a, Zeitdauer d	
Beschreibung: Innerhalb eines Prozesses p muss die Bearbeitungszeit einer Aktivität a genau d Zeiteinheiten betragen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p die Bearbeitungszeit einer Aktivität weniger oder mehr als d Zeiteinheiten beträgt.	

Kurzname: CR-TIME-0017	Name: ActivityServiceTimeLess
Kategorie: Zeit	
Parameter: Prozess p, Aktivität a, Zeitdauer d	
Beschreibung: Innerhalb eines Prozesses p muss die Bearbeitungszeit einer Aktivität a weniger als d Zeiteinheiten betragen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p die Bearbeitungszeit einer Aktivität a genau oder mehr als d Zeiteinheiten beträgt.	

Kurzname: CR-TIME-0018	Name: ActivityServiceTimeGreater
Kategorie: Zeit	
Parameter: Prozess p, Aktivität a, Zeitdauer d	
Beschreibung: Innerhalb eines Prozesses p muss die Bearbeitungszeit einer Aktivität a mehr als d Zeiteinheiten betragen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p die Bearbeitungszeit einer Aktivität a genau oder weniger als d Zeiteinheiten beträgt.	

Kurzname: CR-TIME-0019	Name: ActivityServiceTimeLessOrEqual
Kategorie: Zeit	
Parameter: Prozess p, Aktivität a, Zeitdauer d	
Beschreibung: Innerhalb eines Prozesses p muss die Bearbeitungszeit einer Aktivität a genau oder weniger als d Zeiteinheiten betragen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p die Bearbeitungszeit einer Aktivität a mehr als d Zeiteinheiten beträgt.	

Kurzname: CR-TIME-0020	Name: ActivityServiceTimeGreaterOrEqual
Kategorie: Zeit	
Parameter: Prozess p, Aktivität a, Zeitdauer d	
Beschreibung: Innerhalb eines Prozesses p muss die Bearbeitungszeit einer Aktivität a genau oder mehr als d Zeiteinheiten betragen. Die Compliance-Regel ist verletzt, wenn innerhalb eines Prozesses p die Bearbeitungszeit einer Aktivität a weniger als d Zeiteinheiten beträgt.	

Kategorie Kosten

Kurzname: CR-COST-0004	Name: ProcessCostLessOrEqual
Kategorie: Kosten	
Parameter: Prozess p, Kosten c	
Beschreibung: Die Kosten eines Prozesses p müssen kleiner oder gleich c sein. Die Compliance-Regel ist verletzt, wenn die Kosten eines Prozesses p größer als c sind.	

Kurzname: CR-COST-0005	Name: ProcessCostGreaterOrEqual
Kategorie: Kosten	
Parameter: Prozess p, Kosten c	
Beschreibung: Die Kosten eines Prozesses p müssen gleich oder größer als c sein. Die Compliance-Regel ist verletzt, wenn die Kosten eines Prozesses p kleiner als c sind.	

Kurzname: CR-COST-0006	Name: ProcessResourceCostEqual
Kategorie: Kosten	
Parameter: Prozess p, Ressource re, Kosten c	
Beschreibung: Die innerhalb eines Prozesses p durch eine Ressource re verursachten Kosten müssen gleich c sein. Die Compliance-Regel ist verletzt, wenn die innerhalb eines Prozesses p durch eine Ressource re verursachten Kosten kleiner oder größer als c sind.	

Kurzname: CR-COST-0007	Name: ProcessResourceCostLess
Kategorie: Kosten	
Parameter: Prozess p, Ressource re, Kosten c	
Beschreibung: Die innerhalb eines Prozesses p durch eine Ressource re verursachten Kosten müssen kleiner als c sein. Die Compliance-Regel ist verletzt, wenn die innerhalb eines Prozesses p durch eine Ressource re verursachten Kosten gleich oder größer als c sind.	

Kurzname: CR-COST-0008	Name: ProcessResourceCostGreater
Kategorie: Kosten	
Parameter: Prozess p, Ressource re, Kosten c	
Beschreibung: Die innerhalb eines Prozesses p durch eine Ressource re verursachten Kosten müssen größer als c sein. Die Compliance-Regel ist verletzt, wenn die innerhalb eines Prozesses p durch eine Ressource re verursachten Kosten gleich oder kleiner als c sind.	

Kurzname: CR-COST-0009	Name: ProcessResourceCostLessOrEqual
Kategorie: Kosten	
Parameter: Prozess p, Ressource re, Kosten c	
Beschreibung: Die innerhalb eines Prozesses p durch eine Ressource re verursachten Kosten müssen gleich oder kleiner als c sein. Die Compliance-Regel ist verletzt, wenn die innerhalb eines Prozesses p durch eine Ressource re verursachten Kosten größer als c sind.	

Kurzname: CR-COST-0010	Name: ProcessResourceCostGreaterOrEqual
Kategorie: Kosten	
Parameter: Prozess p, Ressource re, Kosten c	
Beschreibung: Die innerhalb eines Prozesses p durch eine Ressource re verursachten Kosten müssen gleich oder größer als c sein. Die Compliance-Regel ist verletzt, wenn die innerhalb eines Prozesses p durch eine Ressource re verursachten Kosten kleiner als c sind.	

Kurzname: CR-COST-0011	Name: ActivityCostEqual
Kategorie: Kosten	
Parameter: Prozess p, Aktivität a, Kosten c	
Beschreibung: Die innerhalb eines Prozesses p durch eine Aktivität a verursachten Kosten müssen jeweils gleich c sein. Die Compliance-Regel ist verletzt, wenn die innerhalb eines Prozesses p durch eine Aktivität a verursachten Kosten jeweils kleiner oder größer als c sind.	

Kurzname: CR-COST-0012	Name: ActivityCostLess
Kategorie: Kosten	
Parameter: Prozess p, Aktivität a, Kosten c	
Beschreibung: Die innerhalb eines Prozesses p durch eine Aktivität a verursachten Kosten müssen jeweils kleiner als c sein. Die Compliance-Regel ist verletzt, wenn die innerhalb eines Prozesses p durch eine Aktivität a verursachten Kosten gleich oder größer als c sind.	

Kurzname: CR-COST-0013	Name: ActivityCostGreater
Kategorie: Kosten	
Parameter: Prozess p, Aktivität a, Kosten c	
Beschreibung: Die innerhalb eines Prozesses p durch eine Aktivität a verursachten Kosten müssen jeweils größer als c sein. Die Compliance-Regel ist verletzt, wenn die innerhalb eines Prozesses p durch eine Aktivität a verursachten Kosten jeweils gleich oder kleiner als c sind.	

Kurzname: CR-COST-0014	Name: ActivityCostLessOrEqual
Kategorie: Kosten	
Parameter: Prozess p, Aktivität a, Kosten c	
Beschreibung: Die innerhalb eines Prozesses p durch eine Aktivität a verursachten Kosten müssen jeweils gleich oder kleiner als c sein. Die Compliance-Regel ist verletzt, wenn die innerhalb eines Prozesses p durch eine Aktivität a verursachten Kosten jeweils größer als c sind.	

Kurzname: CR-COST-0015	Name: ActivityCostGreaterOrEqual
Kategorie: Kosten	
Parameter: Prozess p, Aktivität a, Kosten c	
Beschreibung: Die innerhalb eines Prozesses p durch eine Aktivität a verursachten Kosten müssen jeweils gleich oder größer als c sein. Die Compliance-Regel ist verletzt, wenn die innerhalb eines Prozesses p durch eine Aktivität a verursachten Kosten jeweils kleiner als c sind.	

Kurzname: CR-COST-0016	Name: ActivityResourceCostEqual
Kategorie: Kosten	
Parameter: Prozess p, Aktivität a, Ressource re, Kosten c	
Beschreibung: Die innerhalb eines Prozesses p durch eine Ressource re bei Ausführung einer Aktivität a verursachten Kosten müssen jeweils gleich c sein. Die Compliance-Regel ist verletzt, wenn die innerhalb eines Prozesses p durch eine Ressource re bei Ausführung einer Aktivität a verursachten Kosten jeweils kleiner oder größer als c sind.	

Kurzname: CR-COST-0017	Name: ActivityResourceCostLess
Kategorie: Kosten	
Parameter: Prozess p, Aktivität a, Ressource re, Kosten c	
Beschreibung: Die innerhalb eines Prozesses p durch eine Ressource re bei Ausführung einer Aktivität a verursachten Kosten müssen jeweils kleiner als c sein. Die Compliance-Regel ist verletzt, wenn die innerhalb eines Prozesses p durch eine Ressource re bei Ausführung einer Aktivität a verursachten Kosten jeweils gleich oder größer als c sind.	

Kurzname: CR-COST-0018	Name: ActivityResourceCostGreater
Kategorie: Kosten	
Parameter: Prozess p, Aktivität a, Ressource re, Kosten c	
Beschreibung: Die innerhalb eines Prozesses p durch eine Ressource re bei Ausführung einer Aktivität a verursachten Kosten müssen jeweils größer als c sein. Die Compliance-Regel ist verletzt, wenn die innerhalb eines Prozesses p durch eine Ressource re bei Ausführung einer Aktivität a verursachten Kosten jeweils gleich oder kleiner als c sind.	

Kurzname: CR-COST-0019	Name: ActivityResourceCostLessOrEqual
Kategorie: Kosten	
Parameter: Prozess p, Aktivität a, Ressource re, Kosten c	
Beschreibung: Die innerhalb eines Prozesses p durch eine Ressource re bei Ausführung einer Aktivität a verursachten Kosten müssen jeweils gleich oder kleiner als c sein. Die Compliance-Regel ist verletzt, wenn die innerhalb eines Prozesses p durch eine Ressource re bei Ausführung einer Aktivität a verursachten Kosten jeweils größer als c sind.	

Kurzname: CR-COST-0020	Name: ActivityResourceCostGreaterOrEqual
Kategorie: Kosten	
Parameter: Prozess p, Aktivität a, Ressource re, Kosten c	
Beschreibung: Die innerhalb eines Prozesses p durch eine Ressource re bei Ausführung einer Aktivität a verursachten Kosten müssen jeweils gleich oder größer als c sein. Die Compliance-Regel ist verletzt, wenn die innerhalb eines Prozesses p durch eine Ressource re bei Ausführung einer Aktivität a verursachten Kosten jeweils kleiner als c sind.	

Anhang B

Nachfolgend ist das XSD-Schema eines Compliance Audit Result-Dokuments aufgeführt.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="audit">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="result"/>
      </xs:sequence>
      <xs:attribute name="eventlog" use="required" type="xs:string"/>
      <xs:attribute name="name" use="required" type="xs:string"/>
      <xs:attribute name="process" use="required" type="xs:string"/>
      <xs:attribute name="time" use="required" type="xs:integer"/>
      <xs:attribute name="uuid" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-
[0-9a-fA-F]{4}-[0-9a-fA-F]{12}"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>
  <xs:element name="result">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="trace"/>
      </xs:sequence>
      <xs:attribute name="category" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="CF"/>
            <xs:enumeration value="ORG"/>
            <xs:enumeration value="TIME"/>
            <xs:enumeration value="COST"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
```

```
      <xs:attribute name="control" use="required" type="xs:string"/>
      <xs:attribute name="parameter" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="[a-z]{1,2}\d\:(\w|\d|\s)+
(#[a-z]{1,2}\d\:(\w|\d|\s)+)*"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="requirement" use="required" type="xs:string"/>
      <xs:attribute name="rule" use="required" type="xs:string"/>
      <xs:attribute name="type" fixed="boolean" type="xs:string"/>
      <xs:attribute name="uuid" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-
[0-9a-fA-F]{4}-[0-9a-fA-F]{12}"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>
  <xs:element name="trace">
    <xs:complexType>
      <xs:attribute name="id" use="required" type="xs:integer"/>
      <xs:attribute name="value" use="required" type="xs:boolean"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Literaturverzeichnis

Accorsi, R., & Lehmann, A. (2012). Automatic Information Flow Analysis of Business Process Models. *Proceedings of the 10th International Conference on Business Process Management, BPM 2012* (S. 172-187). Tallinn, Estonia: Springer.

Accorsi, R., & Stocker, T. (2012). On the exploitation of process mining for security audits: the conformance checking case. *Proceedings of the ACM Symposium on Applied Computing, SAC 2012* (S. 1709-1716). Riva, Trento, Italy: ACM.

Accorsi, R., Damiani, E., & van der Aalst, W. (2013). Unleashing Operational Process Mining (Dagstuhl Seminar 13481). *Dagstuhl Reports 3(11)*, S. 154-192.

Accorsi, R., Lowis, L., & Sato, Y. (2011). Automated Certification for Compliant Cloud-based Business Processes. *Business & Information Systems Engineering 3(3)*, S. 145-154.

Adam, D. (1996). *Planung und Entscheidung: Modelle – Ziele – Methoden.* Gabler Verlag.

Alexander, C., Ishikawa, S., & Silverstein, M. (1978). *A Pattern Language: Towns, Buildings, Construction.* Oxford University Press.

Awad, A. (2010). *A Compliance Management Framework for Business Process Models.* Dissertation, Hasso-Plattner-Institut Potsdam.

Awad, A., & Weidlich, M. (2009). Visualization of Compliance Violation in Business Process Models. *Revised Papers of the BPM 2009 International Workshops on Business Process Management, BPM Workshops 2009* (S. 182-193). Ulm, Germany: Springer.

Awad, A., Decker, G., & Weske, M. (2008). Efficient Compliance Checking Using BPMN-Q and Temporal Logic. *Proceedings of the 6th International Conference on Business Process Management, BPM 2008* (S. 326-341). Milan, Italy: Springer.

Awad, A., Goré, R., Thomson, J., & Weidlich, M. (2011). An Iterative Approach for Business Process Template Synthesis from Compliance Rules. *Proceedings of the 23rd International Conference on Advanced Information Systems Engineering, CAiSE 2011* (S. 406-421). London, UK: Springer.

Awad, A., Weidlich, M., & Weske, M. (2009). Specification, Verification and Explanation of Violation for Data Aware Compliance Rules. *Proceedings of the 7th International Joint Conference on Service-Oriented Computing, ICSOC-ServiceWave 2009* (S. 500-515). Stockholm, Sweden: Springer.

Awad, A., Weidlich, M., & Weske, M. (2011). Visually specifying compliance rules and explaining their violations for business processes. *J. Vis. Lang. Comput. 22(1)*, S. 30-55.

Balzert, H. (2009). *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering.* Springer Spektrum.

Basel Committee on Banking Supervision. (2006). *International Convergence of Capital Measurement and Capital Standards.* Basel Committee on Banking Supervision.

Becker, J. (1998). *Die Grundsätze ordnungsmäßiger Modellierung und ihre Einbettung in ein Vorgehensmodell zur Erstellung betrieblicher Informationsmodelle.* Rundbrief, Westfälische Wilhelms-Universität Münster.

Becker, J., Delfmann, P., Eggert, M., & Schwittay, S. (2012). Generalizability and Applicability of Model-Based Business Process Compliance-Checking Approaches – A State-of-The-Art Analysis and Research Roadmap. *BuR – Business Research, Volume 5, Issue 2*, S. 221-247.

Becker, J., Probandt, W., & Vering, O. (2012). *Grundsätze ordnungsmäßiger Modellierung: Konzeption und Praxisbeispiel für ein effizientes Prozessmanagement.* Berlin, Heidelberg: Springer-Verlag.

Becker, J., Rosemann, M., & Schütte, R. (1995). Grundsätze ordnungsmäßiger Modellierung. *Wirtschaftsinformatik 37(5)*, S. 435-445.

Becker, J., Rosemann, M., & von Uthmann, C. (2000). Guidelines of Business Process Modeling. *Business Process Management, Models, Techniques, and Empirical Studies* (S. 30-49). Springer.

Böhm, M. (2008). IT-Compliance als Triebkraft von Leistungssteigerung und Wertbeitrag der IT. In K. Hildebrand, & S. Meinhardt, *HMD – Praxis Wirtschaftsinform. 263.* dpunkt.verlag.

Caldwell, F., & Wheeler, J. (2013). *Magic Quadrant for Enterprise Governance, Risk and Compliance Platforms.* Gartner, Inc.

Caldwell, F., Eid, T., & Casper, C. (2009). *Magic Quadrant for Enterprise Governance, Risk and Compliance Platforms.* Gartner, Inc.

Caldwell, F., Scholtz, T., & Hagerty, J. (2011). *Magic Quadrant for Enterprise Governance, Risk and Compliance Platforms.* Gartner, Inc.

Chatterjee, A., & Milam, D. (2008). Gaining Competitive Advantage from Compliance and Risk Management. In D. Pantaleo, & N. Pal, *From Strategy to Execution: Turning Accelerated Global Change into Opportunity* (S. 167-183). Berlin Heidelberg: Springer.

Clarke, E., Grumberg, O., & Peled, D. (1999). *Model Checking.* The MIT Press.

COMPAS. (2008). *D2.1 State-of-the-art in the field of compliance languages.* FP7 Project Consortium.

COMPAS. (2009a). *D2.2 Initial specification of compliance language constructs and operators.* FP7 Project Consortium.

COMPAS. (2009b). *D5.3 Final goal-oriented data model.* FP7 Project Consortium.

COSO. (1992). *Internal Control – Integrated Framework.* Committee of Sponsoring Organizations of the Treadway Commission.

COSO. (2004). *Enterprise Risk Management – Integrated Framework: Executive Summary.* Committee of Sponsoring Organizations of the Treadway Commision.

COSO. (2013). *Internal Control – Integrated Framework: Executive Summary.* Committee of Sponsoring Organizations of the Treadway Commission.

Crampton, J. (2004). *On the Satisfiability of Constraints in Workflow Systems.* Technical Report RHUL-MA-2004-1, Royal Holloway University of London.

Davenport, T. (1992). *Process Innovation: Reengineering Work Through Information Technology.* Harvard Business Review Press.

DCGK. (2013). *Deutscher Corporate Governance Kodex.* Regierungskommision Deutscher Corporate Governance Kodex.

De Haes, S., & van Grembergen, W. (2005). IT Governance Structures, Processes and Relational Mechanisms: Achieving IT/Business Alignment in a Major Belgian Financial Group. *Proceedings of the 38th Hawaii International Conference on System Sciences, HICSS 2005.* Big Island, HI, USA: IEEE Computer Society.

Desel, J. (2002). Tutorium: Validierung und Verifikation von Prozessmodellen. *Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen, Promise 2002* (S. 78-80). Potsdam, Germany: GI.

Dijkstra, E. (1982). EWD 447: On the role of scientific thought. *Selected writings on Computing: A Personal Perspective* (S. 60-66). New York: Springer-Verlag.

Draheim, D. (2010). *Business Process Technology: A Unified View on Business Processes, Workflows and Enterprise Applications.* Berlin Heidelberg: Springer-Verlag.

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. (2013). *Fundamentals of Business Process Management.* Berlin Heidelberg: Springer-Verlag.

Dwyer, M., Avrunin, G., & Corbett, J. (1998). Property specification patterns for finite-state verification. *Proceedings of the Second Workshop on Formal Methods in Software Practice, FMSP 1998* (S. 7-15). Clearwater Beach, Florida, USA: ACM.

Dwyer, M., Avrunin, G., & Corbett, J. (1999). Patterns in Property Specifications for Finite-State Verification. *Proceedings of the 1999 International Conference on Software Engineering, ICSE 1999* (S. 411-420). Los Angeles, CA, USA: ACM.

economiesuisse. (2007). *Swiss Code of Best Practice for Corporate Governance*. economiesuisse – Verband der Schweizer Unternehmen.

El Kharbili, M. (2012). Business Process Regulatory Compliance Management Solution Frameworks: A Comparative Evaluation. *Proceedings of the Eighth Asia-Pacic Conference on Conceptual Modelling, APCCM 2012* (S. 23-32). Melbourne, Australia: Australian Computer Society.

El Kharbili, M., & Pulvermüller, E. (2009). A Semantic Framework for Compliance Management in Business Process Management. *Business Process, Services Computing and Intelligent Service Management, BPSC 2009* (S. 60-80). Leipzig, Germany: GI.

Elgammal, A., Türetken, O., van den Heuvel, W.-J., & Papazoglou, M. (2010a). On the Formal Specification of Regulatory Compliance: A Comparative Analysis. *Revised Selected Papers of the ICSOC 2010 International Workshops on Service-Oriented Computing, ICSOC Workshops 2010* (S. 27-38). San Francisco, CA, USA: Springer.

Elgammal, A., Türetken, O., van den Heuvel, W.-J., & Papazoglou, M. (2010b). Root-Cause Analysis of Design-Time Compliance Violations on the Basis of Property Patterns. *Proceedings of the 8th International Conference on Service-Oriented Computing, ICSOC 2010* (S. 17-31). San Francisco, CA, USA: Springer.

Fellmann, M., & Zasada, A. (2014). State-of-the-Art of Business Process Compliance Approaches. *Proceedings of the 22st European Conference on Information Systems, ECIS 2014.* Tel Aviv, Israel: AIS Electronic Library (AISeL).

Fiege, S. (2006). *Risikomanagement- und Überwachungssystem nach KonTraG: Prozess, Instrumente, Träger.* Deutscher Universitätsverlag.

Fisher, J. (2007). Compliance in the Performance Management Context. *Bank Accounting & Finance, Volume 20, Issue 6*, S. 41.

Frigo, M., & Anderson, R. (2009). A Strategic Framework for Governance, Risk, and Compliance. *Strategic Finance*.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software.* Prentice Hall.

Gericke, A., Fill, H.-G., Karagiannis, D., & Winter, R. (2009). Situational method engineering for governance, risk and compliance information systems. *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, DESRIST 2009.* Philadelphia, Pennsylvania, USA: ACM.

Ghose, A., & Koliadis, G. (2007). Auditing Business Process Compliance. *Proceedings of the Fifth International Conference on Service-Oriented Computing, ICSOC 2007* (S. 169-180). Vienna, Austria: Springer.

Giblin, C., Liu, A., Müller, S., Pfitzmann, B., & Zhou, X. (2005). Regulations Expressed As Logical Models (REALM). *Proceedings of the Eighteenth Annual Conference on Legal Knowledge and Information Systems, JURIX 2005* (S. 37-48). Brussels, Belgium: IOS Press.

Gill, S., & Purushottam, U. (2008). Integrated GRC – Is your Organization Ready to Move? *SETLabs Briefings: Governance, Risk and Compliance*, S. 37-46.

Goedertier, S., & Vanthienen, J. (2006a). Business Rules for Compliant Business Process Models. *Proceedings of the 9th International Conference on Business Information Systems, BIS 2006* (S. 558-572). Klagenfurt, Austria: GI.

Goedertier, S., & Vanthienen, J. (2006b). Designing Compliant Business Processes with Obligations and Permissions. *Proceedings of the BPM 2006 International Workshops on Business Process Management, BPM Workshops 2006* (S. 5-14). Vienna, Austria: Springer.

Governatori, G., & Milosevic, Z. (2005). Dealing with contract violations: formalism and domain specific language. *Proceedings of the Ninth IEEE International Enterprise Distributed Object Computing Conference, EDOC 2005* (S. 46-57). Enschede, The Netherlands: IEEE Computer Society.

Governatori, G., & Rotolo, A. (2010a). A conceptually rich model of business process compliance. *Proceedings of the Seventh Asia-Pacific Conference on Conceptual Modelling, APCCM 2010* (S. 3-12). Brisbane, Australia: Australian Computer Society.

Governatori, G., & Rotolo, A. (2010b). Norm Compliance in Business Process Modeling. *Proceedings of the International Symposium on Semantic Web Rules, RuleML 2010* (S. 194-209). Washington, DC, USA: Springer.

Governatori, G., & Sadiq, S. (2008). *The Journey to Business Process Compliance.* Technical Report, The University of Queensland.

Governatori, G., Hoffmann, J., Sadiq, S., & Weber, I. (2008). Detecting Regulatory Compliance for Business Process Models through Semantic Annotations. *Revised Papers of the BPM 2008 International Workshops on Business Process Management, BPM Workshops 2008* (S. 5-17). Milano, Italy: Springer.

Günther, C., & Verbeek, E. (2014). *XES Standard Definition 2.0.* IEEE Task Force on Process Mining.

Hagerty, J., Gaughan, D., & Verma, K. (2008). *The Governance, Risk Management, and Compliance (GRC) Landscape, Part 2: Software's Integral Role in GRC Automation.* Gartner, Inc.

Hallé, S., & Villemaire, R. (2008a). Runtime Monitoring of Message-Based Workflows with Data. *Proceedings of th 12th International IEEE Enterprise Distributed Object Computing Conference, EDOC 2008* (S. 63-72). Munich, Germany: IEEE Computer Society.

Hallé, S., & Villemaire, R. (2008b). XML Methods for Validation of Temporal Properties on Message Traces with Data. *Proceedings of the OTM 2008 Confederated International Conferences: On the Move to Meaningful Internet Systems, OTM Conferences (1) 2008* (S. 337-353). Monterrey, Mexico: Springer.

Hallé, S., & Villemaire, R. (2009). Runtime monitoring of web service choreographies using streaming XML. *Proceedings of the 2009 ACM Symposium on Applied Computing, SAC 2009* (S. 2118-2125). Honolulu, Hawaii, USA: ACM.

Hallé, S., & Villemaire, R. (2012). Runtime Enforcement of Web Service Message Contracts with Data. *IEEE Trans. Services Computing 5(2)*, S. 192-206.

Hallé, S., Bultan, T., Hughes, G., Alkhalaf, M., & Villemaire, R. (2010). Runtime Verification of Web Service Interface Contracts. *IEEE Computer 43(3)*, S. 59-66.

Hallé, S., Villemaire, R., & Cherkaoui, O. (2009). Specifying and Validating Data-Aware Temporal Web Service Properties. *IEEE Trans. Software Eng. 35(5)*, S. 669-683.

Hallé, S., Villemaire, R., Cherkaoui, O., & Ghandour, B. (2007). Model Checking Data-Aware Workflow Properties with CTL-FO+. *Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2007* (S. 267-278). Annapolis, Maryland, USA: IEEE Computer Society.

Hammer, M., & Champy, J. (1993). *Reengineering the Corporation: A Manifesto for Business Revolution.* Harper Business.

Harrington, J. (1991). *Business Process Improvement: The Breakthrough Strategy for Total Quality, Productivity, and Competitiveness.* McGraw-Hill Education.

Hashmi, M., & Governatori, G. (2013). A Methodological Evaluation of Business Process Compliance Management Frameworks. *Selected Papers of the First Asia Pacific Conference on Business Process Management, AP-BPM 2013* (S. 106-115). Beijing, China: Springer.

Hoffmann, J., Weber, I., & Governatori, G. (2012). On compliance checking for clausal constraints in annotated process models. *Information Systems Frontiers 14(2)*, S. 155-177.

Holderer, J. (2014). cLog: Complex Event Log Synthesis Tool. *Multikonferenz Wirtschaftsinformatik 2014*. Paderborn.

Hull, E., Jackson, K., & Dick, J. (2011). *Requirements Engineering.* London: Springer-Verlag.

IDW. (2010). *IDW Prüfungsstandard: Grundsätze ordnungsgemäßer Prüfung von Compliance Management Systemen (IDW EPS 980).* Institut der Wirtschaftsprüfer in Deutschland e.V.

IEEE. (1990). *Glossary of Software Engineering Terminology (610.12-1990).* IEEE Standard.

IEEE. (2010). *Process Mining Manifesto.* IEEE Task Force on Process Mining.

ISACA. (2008). *COBIT Mapping: Mapping of ITIL V3 With COBIT 4.1.* Information Systems Audit and Control Association.

ISACA. (2011). *Relating the COSO Internal Control – Integrated Framework and COBIT.* Information Systems Audit and Control Association.

ISO. (2008). *DIN EN ISO 9001:2008-12 – Qualitätsmanagementsysteme – Anforderungen.* International Organization for Standardization.

ISO. (2008). *ISO/IEC 38500:2008 – Corporate governance of IT.* International Organization for Standardization.

Jablonski, S., & Bussler, C. (1996). *Workflow management – modeling concepts, architecture and implementation.* International Thomson.

Karagiannis, D. (2008). A Business Process-Based Modelling Extension for Regulatory Compliance. *Multikonferenz Wirtschaftsinformatik 2008.* München: GITO-Verlag.

Karagiannis, D., Mylopoulos, J., & Schwab, M. (2007). Business Process-Based Regulation Compliance: The Case of the Sarbanes-Oxley Act. *Proceedings of the 15th IEEE International Requirements Engineering Conference, RE 2007* (S. 315-321). New Delhi, India: IEEE Computer Society.

Klotz, M., & Dorn, D.-W. (2008). IT-Compliance – Begriff, Umfang und relevante Regelwerke. In K. Hildebrand, & S. Meinhardt, *HMD – Praxis Wirtschaftsinform. 263.* dpunkt.verlag.

Knolmayer, G., & Loosli, G. (2006). IT Governance. In R. Zaugg, *Handbuch Kompetenzmanagement* (S. 449-457). Haupt Verlag.

Knuplesch, D., Ly, L., Rinderle-Ma, S., Pfeifer, H., & Dadam, P. (2010). On Enabling Data-Aware Compliance Checking of Business Process Models. *Proceedings of the 29th International Conference on Conceptual Modeling, ER 2010* (S. 332-346). Vancouver, BC, Canada: Springer.

Kosiol, E. (1962). *Organisation der Unternehmung.* Gabler Verlag.

Kumar, A., & Liu, R. (2008). A Rule-Based Framework Using Role Patterns for Business Process Compliance. *Proceedings of the International Symposium on Rule Representation, Interchange and Reasoning on the Web, RuleML 2008* (S. 58-72). Orlando, FL, USA: Springer.

Küster, J., Ryndina, K., & Gall, H. (2007). Generation of Business Process Models for Object Life Cycle Compliance. *Proceedings of the 5th International Conference on Business Process Management, BPM 2007* (S. 165-181). Brisbane, Australia: Springer.

Lanz, A., Weber, B., & Reichert, M. (2014). Time patterns for process-aware information systems. *Requir. Eng. 19(2)*, S. 113-141.

Lewis, E., & Millar, G. (2010). The Viable Governance Model: A Theoretical Model for the Corporate Governance of IT. *IJITBAG 1(3)*, S. 19-35.

Lindland, O., Sindre, G., & Solvberg, A. (1994). Understanding Quality in Conceptual Modeling. *IEEE Software 11(2)*, S. 42-49.

Liu, Y., Müller, S., & Xu, K. (2007). A static compliance-checking framework for business process models. *IBM Systems Journal, Volume 46, Issue 2*, S. 335-361.

Lohmann, N. (2011). Compliance by Design for Artifact-Centric Business Processes. *Proceedings of the 9th International Conference on Business Process Management, BPM 2011* (S. 99-115). Clermont-Ferrand, France: Springer.

Lu, R., Sadiq, S., & Governatori, G. (2007). Compliance Aware Business Process Design. *Revised Selected Papers of the BPM 2007 International Workshops on Business Process Management, BPM Workshops 2007* (S. 120-131). Brisbane, Australia: Springer.

Ly, L., Knuplesch, D., Rinderle-Ma, S., Göser, K., Pfeifer, H., Reichert, M., & Dadam, P. (2010). SeaFlows Toolset – Compliance Verification Made Easy for Process-Aware Information Systems. *Selected Extended Papers of the Information Systems Evolution, CAiSE Forum 2010* (S. 76-91). Hammamet, Tunisia: Springer.

Ly, L., Rinderle-Ma, S., & Dadam, P. (2010). Design and Verification of Instantiable Compliance Rule Graphs in Process-Aware Information Systems. *Proceedings of the 22nd International Conference on Advanced Information Systems Engineering, CAiSE 2010* (S. 9-23). Hammamet, Tunisia: Springer.

Ly, L., Rinderle-Ma, S., Göser, K., & Dadam, P. (2012). On enabling integrated process compliance with semantic constraints in process management systems – Requirements, challenges, solutions. *Information Systems Frontiers 14(2)*, S. 195-219.

Ly, L., Rinderle-Ma, S., Knuplesch, D., & Dadam, P. (2011). Monitoring Business Process Compliance Using Compliance Rule Graphs. *Proceedings of the OTM 2011 Confederated International Conferences: On the Move to Meaningful Internet Systems, OTM Conferences (1) 2011* (S. 82-99). Hersonissos, Crete, Greece: Springer.

Maggi, F., Montali, M., Westergaard, M., & van der Aalst, W. (2011). Monitoring Business Constraints with Linear Temporal Logic: An Approach Based on Colored Automata. *Proceedings of the 9th International Conference on Business Process Management, BPM 2011* (S. 132-147). Clermont-Ferrand, France: Springer.

Manna, Z., & Pnueli, A. (1992). *The Temporal Logic of Reactive and Concurrent Systems – Specification.* New York: Springer-Verlag.

Marefika, W., & Nissen, V. (2009). *Strategisches GRC-Management – Grundzüge eines konzeptionellen Bezugsrahmens.* Forschungsbericht Nr. 2009-02, Technische Universität Ilmenau.

McClean, C. (2009). *The Forrester Wave™: Enterprise Governance, Risk, And Compliance Platforms, Q3 2009.* Forrester Research, Inc.

McClean, C. (2011). *The Forrester Wave™: Enterprise Governance, Risk, And Compliance Platforms, Q4 2011.* Forrester Research, Inc.

McClean, C. (2014). *The Forrester Wave™: Governance, Risk, And Compliance Platforms, Q1 2014.* Forrester Research, Inc.

McClean, C., & Rasmussen, M. (2007). *The Forrester Wave™: Enterprise Governance, Risk, And Compliance Platforms, Q4 2007.* Forrester Research, Inc.

McClean, C., McNabb, K., & Dill, A. (2009). *The GRC Technology Puzzle: Getting All The Pieces To Fit.* Forrester Research, Inc.

McNally, S. (2013). *The 2013 COSO Framework & SOX Compliance.* The Committee of Sponsoring Organizations of the Treadway Commission (COSO).

Mendling, J., Reijers, H., & van der Aalst, W. (2010). Seven process modeling guidelines (7PMG). *Information & Software Technology 52(2)*, S. 127-136.

Menzies, C. (2006). *Sarbanes-Oxley und Corporate Compliance – Nachhaltigkeit, Optimierung, Integration.* Stuttgart: Schäffer-Poeschel.

Mitchell, S. (2007). GRC360: A framework to help organisations drive principled performance. *International Journal of Disclosure and Governance (2007) 4*, S. 279-296.

Montali, M. (2010). *Specification and Verification of Declarative Open Interaction Models – A Logic-Based Approach.* Springer.

Montali, M., Pesic, M., van der Aalst, W., Chesani, F., Mello, P., & Storari, S. (2010). Declarative specification and verification of service choreographies. *TWEB 4(1)*.

Müller, J. (2010). *Strukturbasierte Verifikation von BPMN-Modellen.* Dissertation, Eberhard Karls Universität Tübingen.

Mulo, E., Zdun, U., & Dustdar, S. (2013). Domain-specific language for event-based compliance monitoring in process-driven SOAs. *Service Oriented Computing and Applications 7(1)*, S. 59-73.

Namiri, K. (2008). *Model-Driven Management of Internal Controls for Business Process Compliance.* Dissertation, Universität Karlsruhe (TH).

Namiri, K., & Stojanovic, N. (2007a). Pattern-Based Design and Validation of Business Process Compliance. *Proceedings of the OTM 2007 Confederated International Conferences: On the Move to Meaningful Internet Systems, OTM Conferences (1) 2007* (S. 59-76). Vilamoura, Portugal: Springer.

Namiri, K., & Stojanovic, N. (2007b). Using Control Patterns in Business Processes Compliance. *Proceedings of the WISE 2007 International Workshops on Web Information Systems Engineering, WISE Workshops 2007* (S. 178-190). Nancy, France: Springer.

Nissen, V., & Marefika, W. (2013). Towards a Research Agenda for Strategic Governance, Risk and Compliance (GRC) Management. *Proceedings of the IEEE 15th Conference on Business Informatics, CBI 2013* (S. 1-6). Vienna, Austria: IEEE.

Nissen, V., & Marefika, W. (2014). The Development of a Data-Centred Conceptual Reference Model for Strategic GRC-Management. *Journal of Service Science and Management (2014) 7*, S. 63-76.

Nordsieck, F. (1932). *Die schaubildliche Erfassung und Untersuchung der Betriebsorganisation.* Stuttgart: C. E. Poeschel.

Oberweis, A. (1996). *Modellierung und Ausführung von Workflows mit Petri-Netzen.* Teubner.

OCEG. (2009). *GRC Capability Model "Red Book" 2.0.* Open Compliance and Ethics Group.

OCEG. (2012). *GRC Capability Model "Red Book" 2.1.* Open Compliance and Ethics Group.

OCEG. (2014). *Open Compliance and Ethics Group.* Von http://www.oceg.org/ abgerufen

Ohki, E., Harada, Y., Kawaguchi, S., Shiozaki, T., & Kagaua, T. (2009). Information security governance framework. *Proceedings of the 1st ACM Workshop on Information Security Governance, WISG 2009* (S. 1-6). Chicago, IL, USA: ACM.

OMG. (2013). *BPMN 2.0.2*. Von http://www.omg.org/spec/BPMN/2.0.2/ abgerufen

Overhage, S., Birkmeier, D., & Schlauderer, S. (2012). Quality Marks, Metrics, and Measurement Procedures for Business Process Models – The 3QM-Framework. *Business & Information Systems Engineering 4(5)*, S. 229-246.

Papazoglou, M. (2011). Making Business Processes Compliant to Standards and Regulations. *Proceedings of the 15th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2011* (S. 3-13). Helsinki, Finland: IEEE Computer Society.

Parnas, D. (1972). On the Criteria To Be Used in Decomposing Systems into Modules. *Commun. ACM 15(12)*, S. 1053-1058.

Partsch, H. (2010). *Requirements-Engineering systematisch – Modellbildung für softwaregestützte Systeme.* Berlin Heidelberg: Springer-Verlag.

Paulus, S. (2009). *Overview Report: A GRC Reference Architecture.* KuppingerCole.

PCAOB. (2007). *Auditing Standard No. 5 – An Audit of Internal Control Over Financial Reporting That Is Integrated with An Audit of Financial Statements (PCAOB Release No. 2007-005A).* Public Company Accounting Oversight Board.

Pesic, M. (2008). *Constraint-Based Workflow Management Systems: Shifting Control to Users.* PhD thesis, Eindhoven University of Technology.

Pesic, M., & van der Aalst, W. (2006). A Declarative Approach for Flexible Business Processes Management. *Proceedings of the BPM 2006 International Workshops on Business Process Management, BPM Workshops 2006* (S. 169-180). Vienna, Austria: Springer.

Pesic, M., Schonenberg, H., Sidorova, N., & van der Aalst, W. (2007). Constraint-Based Workflow Models: Change Made Easy. *Proceedings of the OTM 2007 Confederated International Conferences: On the Move to Meaningful Internet Systems, OTM Conferences (1) 2007* (S. 77-94). Vilamoura, Portugal: Springer.

Petri, C. A. (1962). *Kommunikation mit Automaten.* Dissertation, Universität Hamburg.

Porter, M. (1998). *Competitive Advantage: Creating and Sustaining Superior Performance.* Free Press.

Proctor, P., Caldwell, F., & Eid, T. (2008). *A Comparison Model for the GRC Marketplace, 2008 to 2010.* Gartner, Inc.

Protiviti. (2007). *Guide to the Sarbance-Oxley Act: Internal Control Reporting Requirements – Frequently Asked Questions Regarding Section 404.* Protiviti, Inc.

PwC. (2004). *8th annual global CEO survey.* PricewaterhouseCoopers.

Racz, N., Weippl, E., & Seufert, A. (2010a). A Frame of Reference for Research of Integrated Governance, Risk and Compliance (GRC). *Proceedings of the 11th IFIP TC 6/TC International Conference on Communications and Multimedia Security, CMS 2010* (S. 106-117). Linz, Austria: Springer.

Racz, N., Weippl, E., & Seufert, A. (2010b). A process model for integrated IT governance, risk, and compliance management. *Proceedings of the Ninth International Baltic Conference on Databases and Information Systems, DB&IS 2010* (S. 155-170). Riga, Latvia: University of Latvia Press.

Racz, N., Weippl, E., & Seufert, A. (2010c). Questioning the Need for Separate IT Risk Management Frameworks. *Informatik 2010: Beiträge der 40. Jahrestagung der Gesellschaft für Informatik e.V., GI Jahrestagung (2) 2010* (S. 245-252). Leipzig, Germany: GI.

Ramezani, E., Fahland, D., & van der Aalst, W. (2012). Where Did I Misbehave? Diagnostic Information in Compliance Checking. *Proceedings of the 10th International Conference on Business Process Management, BPM 2012* (S. 262-278). Tallinn, Estonia: Springer.

Ramezani, E., Fahland, D., & van der Aalst, W. (2013). Supporting Domain Experts to Select and Configure Precise Compliance Rules. *Revised Papers of the BPM 2013 International Workshops on Business Process Management, BPM Workshops 2013* (S. 498-512). Beijing, China: Springer.

Ramezani, E., Fahland, D., van der Werf, J., & Mattheis, P. (2011). Separating Compliance Management and Business Process Management. *Revised Selected Papers of the BPM 2011 International Workshops on Business Process Management, BPM Workshops (2) 2011* (S. 459-464). Clermont-Ferrand, France: Springer.

Ramezani, E., Fahland, D., van Dongen, B., & van der Aalst, W. (2013). Diagnostic Information for Compliance Checking of Temporal Compliance Requirements. *Proceedings of the 25th International Conference on Advanced Information Systems Engineering, CAiSE 2013* (S. 304-320). Valencia, Spain: Springer.

Ramezani, E., Gromov, V., Fahland, D., & van der Aalst, W. (2014). Compliance Checking of Data-Aware and Resource-Aware Compliance Requirements. *Proceedings of the OTM 2014 Confederated International Conferences: On the Move to Meaningful Internet Systems, OTM Conferences 2014* (S. 237-257). Amantea, Italy: Springer.

Reichert, M., & Weber, B. (2012). *Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies.* Berlin Heidelberg: Springer-Verlag.

Rinderle-Ma, S., Ly, L., & Dadam, P. (2008). Business Process Compliance (Aktuelles Schlagwort). *EMISA Forum*, (S. 24-29).

Rodríguez, C., Schleicher, D., Daniel, F., Casati, F., Leymann, F., & Wagner, S. (2013). SOA-enabled compliance management: instrumenting, assessing, and analyzing service-based business processes. *Service Oriented Computing and Applications 7(4)*, S. 275-292.

Rosemann, M., & zur Muehlen, M. (2005). Integrating Risks in Business Process Models. *Proceedings of the 16th Australasian Conference on Information Systems, ACIS 2005.* Sydney, Australia: AIS Electronic Library (AISeL).

Rozinat, A., & van der Aalst, W. (2008). Conformance checking of processes based on monitoring real behavior. *Inf. Syst. 33(1)*, S. 64-95.

Russell, N., ter Hofstede, A., Edmond, D., & van der Aalst, W. (2004a). *Workflow Data Patterns.* Technical Report FIT-TR-2004-01, Queensland University of Technology.

Russell, N., ter Hofstede, A., Edmond, D., & van der Aalst, W. (2004b). *Workflow Resource Patterns.* BETA Working Paper Series WP 127, Eindhoven University of Technology.

Russell, N., ter Hofstede, A., Edmond, D., & van der Aalst, W. (2005). Workflow Data Patterns: Identification, Representation and Tool Support. *Proceedings of the 24th International Conference on Conceptual Modeling, ER 2005* (S. 353-368). Klagenfurt, Austria: Springer.

Russell, N., ter Hofstede, A., van der Aalst, W., & Mulyar, N. (2006). *Workflow Control-Flow Patterns: A Revised View.* BPM Center Report BPM-06-22, BPMcenter.org.

Russell, N., van der Aalst, W., & ter Hofstede, A. (2006a). *Exception Handling Patterns in Process-Aware Information Systems.* BPM Center Report BPM-06-04, BPMcenter.org.

Russell, N., van der Aalst, W., & ter Hofstede, A. (2006b). Workflow Exception Patterns. *Proceedings of the 18th International Conference on Advanced Information Systems Engineering, CAiSE 2006* (S. 288-302). Luxembourg, Luxembourg: Springer.

Russell, N., van der Aalst, W., ter Hofstede, A., & Edmond, D. (2005). Workflow Resource Patterns: Identification, Representation and Tool Support. *Proceedings of the 17th International Conference on Advanced Information Systems Engineering, CAiSE 2005* (S. 216-232). Porto, Portugal: Springer.

Sackmann, S. (2008). Automatisierung von Compliance. In K. Hildebrand, & S. Meinhardt, *HMD – Praxis Wirtschaftsinform. 263.* dpunkt.verlag.

Sackmann, S., & Kähmer, M. (2008). ExPDT: Ein Policy-basierter Ansatz zur Automatisierung von Compliance. *Wirtschaftsinformatik 50(5)*, S. 366-374.

Sadiq, S., & Governatori, G. (2010). Managing Regulatory Compliance in Business Processes. In J. vom Brocke, & M. Rosemann, *Handbook on Business Process Management 2: Strategic Alignment, Governance, People and Culture* (S. 159-175). Berlin Heidelberg: Springer-Verlag.

Sadiq, S., Governatori, G., & Namiri, K. (2007). Modeling Control Objectives for Business Process Compliance. *Proceedings of the 5th International Conference on Business Process Management, BPM 2007* (S. 149-164). Brisbane, Australia: Springer.

Scheer, A.-W. (1992). *Architektur integrierter Informationssyteme: Grundlagen der Unternehmensmodellierung.* Berlin Heidelberg: Springer-Verlag.

Scheer, A.-W., Nüttgens, M., & Zimmermann, V. (1995). Rahmenkonzept für ein integriertes Geschäftsprozeßmanagement. *Wirtschaftsinformatik 37(5)*, S. 426-434.

Schefer, S., Strembeck, M., & Mendling, J. (2011). Checking Satisfiability Aspects of Binding Constraints in a Business Process Context. *Revised Selected Papers of the BPM 2011 International Workshops on Business Process Management, BPM Workshops (2) 2011* (S. 465-470). Clermont-Ferrand, France: Springer.

Schefer, S., Strembeck, M., Mendling, J., & Baumgrass, A. (2011). Detecting and Resolving Conflicts of Mutual-Exclusion and Binding Constraints in a Business Process Context. *Proceedings of the OTM 2011 Confederated International Conferences: On the Move to Meaningful Internet Systems, OTM Conferences (1) 2011* (S. 329-346). Hersonissos, Crete, Greece: Springer.

Schleicher, D., Anstett, T., Leymann, F., & Mietzner, R. (2009). Maintaining Compliance in Customizable Process Models. *Proceedings of the OTM 2009 Confederated International Conferences: On the Move to Meaningful Internet Systems, OTM Conferences (1) 2009* (S. 60-75). Vilamoura, Portugal: Springer.

Schleicher, D., Grohe, S., Leymann, F., Schneider, P., Schumm, D., & Wolf, T. (2011). An approach to combine data-related and control-flow-related compliance rules. *Proceedings of the 2011 IEEE International Conference on Service-Oriented Computing and Applications, SOCA 2011* (S. 1-8). Irvine, CA, USA: IEEE Computer Society.

Schönthaler, F., Vossen, G., Oberweis, A., & Karle, T. (2011). *Geschäftsprozesse für Business Communities: Modellierungssprachen, Methoden, Werkzeuge.* München: Oldenbourg Verlag.

Schultz, M. (2013). Towards an Empirically Grounded Conceptual Model for Business Process Compliance. *Proceedings of the 32th International Conference on Conceptual Modeling, ER 2013* (S. 138-145). Hong-Kong, China: Springer.

Schultz, M., Müller-Wickop, N., Werner, M., & Nüttgens, M. (2013). Geschäftsprozessorientierte Prüfung von IT-Systemen. In M. Knoll, *HMD – Praxis Wirtschaftsinform. 289*. dpunkt.verlag.

Schumm, D., Leymann, F., & Streule, A. (2010). Process Views to Support Compliance Management in Business Processes. *Proceedings of the 11th International Conference on E-Commerce and Web Technologies, EC-Web 2010* (S. 131-142). Bilbao, Spain: Springer.

Schumm, D., Türetken, O., Kokash, N., Elgammal, A., Leymann, F., & van den Heuvel, W.-J. (2010). Business Process Compliance through Reusable Units of Compliant Processes. *Revised Selected Papers of the ICWE 2010 Workshops on Web Engineering, ICWE Workshops 2010* (S. 325-337). Vienna, Austria: Springer.

Sebahi, S., & Hacid, M.-S. (2010). Business Process Monitoring with BPath – (Short Paper). *Proceedings of the OTM 2010 Confederated International Conferences: On the Move to Meaningful Internet Systems, OTM Conferences (1) 2010* (S. 446-453). Hersonissos, Crete, Greece: Springer.

SEC. (2002). *Sarbanes-Oxley Act of 2002 (Sec. 404 – Management Assessment of Internal Controls)*. Securities and Exchange Commssion, Public Law 107-204, 107th Congress.

Spanaki, K., & Papazafeiropoulou, A. (2013). Analysing The Governance, Risk And Compliance (Grc) Implementation Process: Primary Insights. *Proceedings of the 21st European Conference on Information Systems, ECIS 2013* (S. 58). Utrecht, The Netherlands: AIS Electronic Library (AISeL).

Standards Australia. (2006). *Australian Standard – Compliance programs (AS 3806-2006)*. Standards Australia.

Stemper, H. (2008). Kompliziert compliant? In K. Hildebrand, & S. Meinhardt, *HMD – Praxis Wirtschaftsinform. 263*. dpunkt.verlag.

Stocker, T., & Accorsi, R. (2013). SecSy: Synthesizing Process Event Logs. *Proceedings of the 5th International Workshop on Enterprise Modelling and Information Systems Architectures, EMISA 2013* (S. 71-84). St. Gallen, Switzerland: GI.

Stocker, T., Accorsi, R., & Rother, T. (2013). Computergestützte Prozessauditierung mit Process Mining. In S. Strahringer, *HMD – Praxis Wirtschaftsinform. 292*. dpunkt.verlag.

Strecker, S., Heise, D., & Frank, U. (2011). Prolegomena of a modelling method in support of audit risk assessment – Outline of a domain-specific modelling language for internal controls and internal control systems. *Enterprise Modelling and Information Systems Architectures 6(3)*, S. 5-24.

Sylvester, D. (April 2011). ISO 38500 – Why Another Standard? *COBIT Focus: Using COBIT, Val IT, Risk IT, BMIS and ITAF, Volume 2*, S. 1-3.

Tapscott, D. (2006). *Trust and Competitive Advantage: An Integrated Approach.* New Paradigm Learning Corporation.

Tarantino, A. (2008). *Governance, Risk and Compliance Handbook.* John Wiley & Sons, Inc.

Thullner, R., Rozsnyai, S., Schiefer, J., Obweger, H., & Suntinger, M. (2011). Proactive Business Process Compliance Monitoring with Event-Based Systems. *Workshops Proceedings of the 15th IEEE International Enterprise Distributed Object Computing Conference, EDOCW 2011* (S. 429-437). Helsinki, Finland: IEEE Computer Society.

Türetken, O., Elgammal, A., van den Heuvel, W.-J., & Papazoglou, M. (2011). Enforcing compliance on business processes through the use of patterns. *Proceedings of the 19th European Conference on Information Systems, ECIS 2011.* Helsinki, Finland: AIS Electronic Library (AISeL).

Türetken, O., Elgammal, A., van den Heuvel, W.-J., & Papazoglou, M. (2012). Capturing Compliance Requirements: A Pattern-Based Approach. *IEEE Software 29(3)*, S. 28-36.

TÜV Rheinland. (2011). *TR CMS 101:2011 – Standard für Compliance Management Systeme (CMS).* TÜV Rheinland Köln.

Vallet, J., Mrad, A., Hallé, S., & Beaudet, É. (2013). The Relational Database Engine: An Efficient Validator of Temporal Properties on Event Traces. *Proceedings of the 17th IEEE International Enterprise Distributed Object Computing Conference Workshops, EDOC Workshops 2013* (S. 275-284). Vancouver, BC, Canada: IEEE Computer Society.

van der Aalst, W. (1995). *A class of Petri nets for modeling and analyzing business processes.* Computing Science Report 95/26, Eindhoven University of Technology.

van der Aalst, W. (1998). The Application of Petri Nets to Workflow Management. *Journal of Circuits, Systems, and Computers 8(1)*, S. 21-66.

van der Aalst, W. (2009). Process-Aware Information Systems: Lessons to Be Learned from Process Mining. *Trans. Petri Nets and Other Models of Concurrency 2*, S. 1-26.

van der Aalst, W. (2010). Business Process Simulation Revisited. *Selected Papers of the 6th International Workshop on Enterprise and Organizational Modeling and Simulation, EOMAS 2010* (S. 1-14). Hammamet, Tunisia: Springer.

van der Aalst, W. (2011). *Process Mining: Discovery, Conformance and Enhancement of Business Processes.* Berlin Heidelberg: Springer.

van der Aalst, W. (2015). Extracting Event Data from Databases to Unleash Process Mining. In J. vom Brocke, & T. Schmiedel, *BPM – Driving Innovation in a Digital World* (S. 105-128). Springer International Publishing.

van der Aalst, W., & Pesic, M. (2006). DecSerFlow: Towards a Truly Declarative Service Flow Language. *Proceedings of the Third International Workshop on Web Services and Formal Methods, WS-FM 2006* (S. 1-23). Vienna, Austria: Springer.

van der Aalst, W., & Stahl, C. (2011). *Modeling Business Processes – A Petri Net-Oriented Approach.* The MIT Press.

van der Aalst, W., Barros, A., ter Hofstede, A., & Kiepuszewski, B. (2000). Advanced Workflow Patterns. *Proceedings of the 7th International Conference on Cooperative Information Systems, CoopIS 2000* (S. 18-29). Eilat, Israel: Springer.

van der Aalst, W., de Beer, H., & van Dongen, D. (2005). Process Mining and Verification of Properties: An Approach Based on Temporal Logic. *Proceedings of the OTM 2005 Confederated International Conferences: On the Move to Meaningful Internet Systems, OTM Conferences (1) 2005* (S. 130-147). Agia Napa, Cyprus: Springer.

van der Aalst, W., ter Hofstede, A., Kiepuszewski, B., & Barros, A. (2003). Workflow Patterns. *Distributed and Parallel Databases 14(1)*, S. 5-51.

VDI. (1993). *VDI-Richtlinie: Simulation von Logistik-, Materialfluss- und Produktionssystemen – Grundlagen (VDI 3633 Blatt 1).* VDI Verein Deutscher Ingenieure e.V.

Verbeek, E., Buijs, J., van Dongen, B., & van der Aalst, W. (2010). XES, XESame, and ProM 6. *Selected Extended Papers of the Information Systems Evolution, CAiSE Forum 2010* (S. 60-75). Hammamet, Tunisia: Springer.

Vicente, P., & Mira da Silva, M. (2011). A Conceptual Model for Integrated Governance, Risk and Compliance. *Proceedings of the 23rd International Conference on Advanced Information Systems Engineering, CAiSE 2011* (S. 199-213). London, UK: Springer.

Volonino, L., Gessner, G., & Kermis, G. (2004). Holistic Compliance with Sarbanes-Oxley. *Communications of the Association for Information Systems, Volume 14, Article 11*.

W3C. (2014). *XQuery 3.0: An XML Query Language*. Von https://www.w3.org/TR/xquery-30/ abgerufen

Weber, I. (2009). *Semantic methods for execution level business process modeling: modeling support through process verification and service composition.* Dissertation, Karlsruher Institut für Technologie (KIT).

Weidlich, M., Polyvyanyy, A., Desai, N., & Mendling, J. (2010). Process Compliance Measurement Based on Behavioural Profiles. *Proceedings of the 22nd International Conference on Advanced Information Systems Engineering, CAiSE 2010* (S. 499-514). Hammamet, Tunisia: Springer.

Weidlich, M., Ziekow, H., Mendling, J., Günther, O., Weske, M., & Desai, N. (2011). Event-Based Monitoring of Process Execution Violations. *Proceedings of the 9th International Conference on Business Process Management, BPM 2011* (S. 182-198). Clermont-Ferrand, France: Springer.

Weill, P., & Ross, J. (2004). *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results.* Harvard Business Review Press.

Weske, M. (2012). *Business Process Management: Concepts, Languages, Architectures.* Springer.

Westergaard, M., & van Dongen, B. (2013). *KeyValueSets: Event Logs Revisited.* BPM Center Report BPM-13-25, BPMcenter.org.

WFMC. (1999). *Terminology & Glossary (WFMC-TC-1011).* Workflow Management Coalition.

Wiesche, M., Schermann, M., & Krcmar, H. (2011). Exploring the contribution of information technology to governance, risk management, and compliance (GRC) initiatives. *Proceedings of the 19th European Conference on Information Systems, ECIS 2011*. Helsinki, Finland: AIS Electronic Library (AISeL).

Wynn, M., De Weerdt, J., ter Hofstede, A., van der Aalst, W., Reijers, H., Adams, M., . . ., Low, W. (2013). Cost-Aware Business Process Management: A Research Agenda. *Proceedings of the 24th Australasian Conference on Information Systems, ACIS 2013*. Melbourne, Australia: Australian Computer Society.

Wynn, M., Low, W., & Nauta, W. (2013). A Framework for Cost-Aware Process Management: Generation of Accurate and Timely Management Accounting Cost Report. *Proceedings of the Ninth Asia-Pacific Conference on Conceptual Modelling, APCCM 2013* (S. 79-88). Adelaide, Australia: Australian Computer Society.

Wynn, M., Low, W., ter Hofstede, A., & Nauta, W. (2014). A Framework for Cost-Aware Process Management: Cost Reporting and Cost Prediction. *J. UCS 20(3)*, S. 406-430.

Wynn, M., Reijers, H., Adams, M., Ouyang, C., ter Hofstede, A., van der Aalst, W., . . . Hoque, Z. (2013). Cost-Informed Operational Process Support. *Proceedings of the 32th International Conference on Conceptual Modeling, ER 2013* (S. 174-181). Hong-Kong, China: Springer.

Die Einhaltung regulatorischer Anforderungen und interner Richtlinien ist zunehmend kritisch für den Unternehmenserfolg geworden. Aufgrund der Vielzahl von Anforderungen und Richtlinien werden Methoden, Verfahren und Werkzeuge benötigt, um Unternehmen und ihre Mitarbeiter bei der Kontrolle, Bewertung und Verbesserung ihrer Geschäftsprozesse geeignet zu unterstützen. In dieser Arbeit wird ein Ansatz vorgestellt, der eine effiziente Compliance-Prüfung von Geschäftsprozessen basierend auf den in standardisierten Ereignisprotokollen aufgezeichneten Ereignisdaten ermöglicht. Neben einer Referenzimplementierung des Ansatzes werden eine Erweiterung für ein Geschäftsprozessmanagement und -modellierungswerkzeug sowie ein webbasiertes Dashboard für die interaktive Analyse von Auditergebnissen entwickelt. Anhand eines Anwendungsbeispiels wird der Ansatz evaluiert.

ISBN 978-3-7315-0507-5