

Anwendung von grafischen Validierungsregeln bei der Entwicklung von IT-Integrationsprozessen

Jens Drawehn und Sven Feja

jens.drawehn@iao.fraunhofer.de, svfe@informatik.uni-kiel.de

Abstract: Zunehmend stellt die Modellierung von (Geschäfts-)Prozessmodellen den Ausgangspunkt der Entwicklung von Software bspw. basierend auf Service-orientierten Architekturen dar. Um die syntaktische und semantische Korrektheit der Modelle zu überprüfen, sind sinnvoll einsetzbare Mechanismen notwendig. Die syntaktische Prüfung wird meist direkt von den Modellierungswerkzeugen unterstützt. Hingegen gibt es kaum Ansätze zum Einsatz von Validierungstechniken – wie dem Model Checking – auf Geschäftsprozessebene. Außerdem wird die Wiederverwendung von Validierungsregeln bisher vernachlässigt.

In diesem Beitrag wird die Erweiterung eines auf grafischen Validierungsregeln (EPK-G-CTL) basierenden Verfahrens vorgestellt, mit dem fachliche Prozessmodelle gezielt auf semantische Aspekte überprüft werden können. Im Fokus der Betrachtungen steht die Frage, wie der Validierungsmechanismus in den Entwicklungsprozess eingebunden werden kann, so dass für die Entwickler ein erkennbarer Nutzen entsteht. Dabei sind die Aspekte der einfachen Erstellung und Anwendbarkeit sowie der Wiederverwendbarkeit der Validierungsregeln von Bedeutung.

1 Einleitung

Im BMBF-geförderten Forschungsprojekt OrViA [FKT08] wurde ein durchgängiges Framework für die fachlich getriebene Realisierung von IT-Integrationsprojekten entwickelt, dem sowohl die Prinzipien der modellgetriebenen Softwareentwicklung [SVEH07] als auch das Konzept der Service-orientierten Architekturen (SOA) [MLM⁺06] zugrunde liegen. Dabei erfolgt die Gestaltung der IT-Integrations szenarien primär anhand der Prozesse. Einer vergleichenden Betrachtung verschiedener Vorgehensweisen in SOA-Projekten [HWSD07] zufolge handelt es sich dabei um einen Top-Down-Ansatz.

Eine der wesentlichen Zielstellungen von SOA-Projekten besteht in der Unterstützung der langfristigen Entwicklung und Pflege von Softwarelandschaften. Durch die Wiederverwendung von Services soll eine höhere Flexibilität bei der Umsetzung neuer Anforderungen erreicht werden. Da Services im Sinne des Konzepts der SOA nicht als reine IT-Funktionen zu betrachten sind, sondern vielmehr Geschäftsfunktionen realisieren [TS07], ergeben sich dabei neue Anforderungen an den Softwareentwicklungsprozess. Unter anderem stellt sich bei der Erstellung neuer Services die Frage, wie die semantische Korrektheit der modellierten Szenarien beurteilt werden kann.

Für die fachliche Modellierung von Geschäftsprozessen steht die Werkzeugkategorie der

Geschäftsprozessmodellierungswerkzeuge (Business Process Modeling Tools, BPMT) zur Verfügung. Eine Untersuchung zu den Einsatzmöglichkeiten von BPMT in SOA-Projekten findet sich in [DH08]. Allgemein zeichnet sich der Trend ab, BPMT und die damit abgebildeten Prozess- und sonstigen Modelle zunehmend im Sinne der modellgetriebenen Softwareentwicklung zu nutzen. Dies wirft die Frage auf, wie man die semantische Korrektheit der Prozessmodelle sichern kann. Grundsätzlich bieten BPMTs die Möglichkeit semantische Informationen im Zusammenhang mit den Prozessen zu erfassen. Diese Informationen sind aber nicht geeignet für automatische Modellprüfungsverfahren – wie dem Model Checking. Der Grund ist die nicht formale Hinterlegung der semantischen Informationen. Dies erschwert außerdem die Erreichung des Ziels der Wiederverwendbarkeit, Wartbarkeit und Pflegebarkeit von Geschäftslogik, welche in den oben geschilderten Szenarien gefordert werden. Letztendlich existieren in der Praxis keine etablierten, standardisierten Formate und Verfahren zur semantischen Prüfung von Geschäftsprozessmodellen auf der Modellebene.

Daher wird in diesem Beitrag ein Ansatz zur Formulierung von wiederverwendbaren, grafischen Validierungsregeln vorgestellt. Die grafische Modellierung der Regeln basiert dabei auf [FFS08]. Der Fokus in diesem Beitrag liegt auf der Wiederverwendbarkeit von Validierungsregeln. Diese wird durch die Zuordnung der Regeln in verschiedene inhaltliche Reichweiten erzielt.

2 Hintergrund

Geschäftsprozessmodelle bilden mittlerweile den zentralen Punkt der Unternehmens- und Softwareentwicklung. Daher wird zunehmend versucht durch "explizite Modellierung und Implementierung von Geschäftsregeln" [End04] die Geschäftslogik direkt in die verwendeten Modelle aufzunehmen. Die Geschäftslogik wird in sogenannten Geschäftsregeln formuliert. Der Begriff der Geschäftsregel ist dabei laut [HH00]:

"A business rule is a statement that defines or constrains some aspect of the business."

Dabei wird zwischen verschiedenen Typen unterschieden. In [End04] findet sich ein umfassendes Schemata zur Klassifikation der Geschäftsregeln. Die Klassifikation der Geschäftsregeln unterscheidet sich zwischen den verschiedenen Autoren aufgrund der unterschiedlichen Blickpunkte [Bus08]. Die Business Rule Group unterscheidet zwischen den Blickpunkten *Business Perspective* und der *Information System Perspective*. Die letztere ist die überwiegend in diesem Beitrag betrachtete Perspektive. Diese Regeln setzen den Fokus auf die semantische d. h. fachliche Korrektheit der Geschäftsprozesse.

Geschäftsregeln können grundsätzlich zur Spezifikation von Softwaresystemen genutzt werden. [End04] schlägt die direkte Entwicklung von Geschäftsprozessen aus modellierten Geschäftsregeln vor. Hingegen wird in [Ger04] festgestellt, dass die Wiederverwendbarkeit und Flexibilität von Geschäftsregeln erst durch die getrennte Verwaltung von Regeln und Geschäftsprozessen ermöglicht wird. Dieser Ansatz wird von dem in [FFS08] vorgestellten Verfahren verfolgt. Dabei werden Geschäftsregeln auf der Basis von Modell-

prüfungstechniken (dem Model Checking) als Validierungsregeln definiert. Diese Validierungsregeln sind temporale Regeln, die in der vorgestellten Notation auf der Ebene von Geschäftsprozessen erstellt werden können. Allerdings enthält das in [FFS08] vorgestellte Verfahren bisher nicht die Möglichkeit allgemeingültige, wiederverwendbare Regeln bzw. Anforderungen zu erstellen, da die Validierung nur direkt über den Namen des betroffenen Geschäftsprozesselements stattfindet.

3 Das Szenario "Elektronische Steuererklärung"

In diesem Abschnitt wird die Anwendung des Validierungsmechanismus anhand eines Szenarios aus der Domäne der Steuerverwaltung beispielhaft beschrieben. Ziel der aktuellen Vorhaben ELSTER und EloSt ist die durchgängige, medienbruchfreie elektronische Verarbeitung von Einkommenssteuer-Erklärungen beginnend beim Bürger. Herausforderungen bei der Realisierung dieser Projekte bestehen in organisatorischen Unterschieden, heterogenen IT-Strukturen, unterschiedlich guter IT-Ausstattung der verschiedenen Finanzverwaltungen und Finanzämter und der Berücksichtigung länderspezifischer Besonderheiten. Eine Beschreibung des Szenarios und der Rahmenbedingungen findet sich in [PD08].

Im Fokus der Betrachtungen stehen hier der IT-Integrationsprozess und der Einsatz des Validierungsverfahrens bei dessen Entwicklung. Zielstellung ist die Verfügbarkeit eines Services von Seiten der Finanzbehörde für die Bürger, die damit die Möglichkeit erhalten, ihre Steuererklärungen in elektronischer Form in einem standardisierten Format an ein Finanzamt zu übergeben. Das fachliche Prozessmodell ist in Abbildung 1 dargestellt. Für die Realisierung des Szenarios wurden verschiedene Methoden und Werkzeuge eingesetzt. Für den hier relevanten Ausschnitt der Entwicklung des Integrationsprozesses wurden der ARIS SOA Architect für die Geschäftsprozessmodellierung und der Oracle BPEL Process Manager für die Ausführung der Prozesse verwendet. Die Transformation des fachlichen Modells in das ausführbare Format BPEL leistet der ARIS SOA Architect.

Die Validierungsregeln liegen in ARIS in Form von EPK-G-CTL-Diagrammen vor. Abbildung 2 zeigt drei beispielhafte Validierungsregeln. Die allgemeingültige Regel *Persistenz von Datenobjekten* wurde in einem früher modellierten Szenario entwickelt. Wenn für ein Datenobjekt die Eigenschaft *Persistenz gefordert* zutrifft, verlangt die Regel, dass auf allen gültigen Pfaden des Prozessmodells ein Service aufgerufen wird, an den dieses Datenobjekt übergeben wird und der die Eigenschaft *Persistenz erfüllt* aufweist. Im Modell ist die Information hinterlegt, dass für die ELSTER-Nachricht die Eigenschaft *Persistenz gefordert* zutrifft. Dadurch wird die Regel im modellierten Szenario wirksam.

Die domänenspezifische Regel *Unveränderbarkeit* drückt aus, dass das spezifische Datenobjekt ELSTER-Nachricht im Ablauf eines Prozesses nicht verändert werden darf. Die szenariospezifische Regel *Eingangsstempel* bezieht sich ebenfalls ausschließlich auf die ELSTER-Nachricht. Sie verlangt, dass die ELSTER-Nachricht im ersten Prozessschritt an einen Service übergeben wird, der die Eigenschaft *Persistenz erfüllt* aufweist. Eine genaue Erläuterung der Regeln und die Zuordnung zu den Regeltypen erfolgt im nächsten Abschnitt.

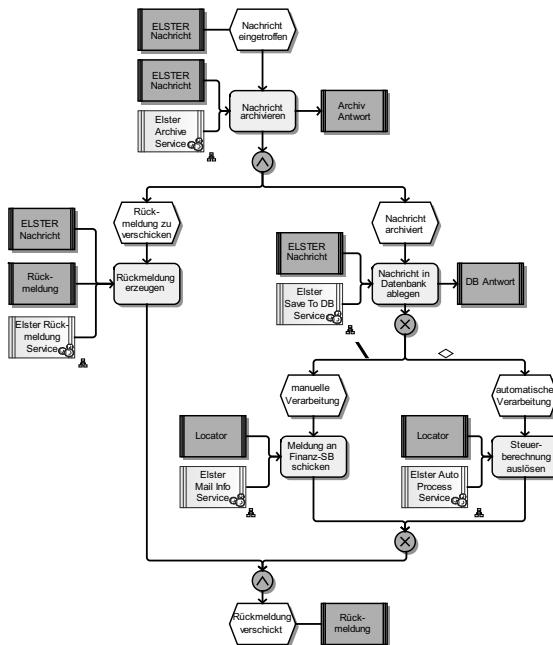


Abbildung 1: EPK des ELSTER-Prozesses

4 Anwendung der Validierungsregeln bei der Prozessmodellierung

Mit dem in [FFS08] beschriebenen Validierungsmechanismus ist es möglich, komplexe Bedingungen für Prozessmodelle zu formulieren und automatisiert zu überprüfen. Entscheidend für die Anwendbarkeit der Validierung ist aber auch, dass für die Anwender (in diesem Fall sind das die Prozessmodellierer) ein erkennbarer Nutzen entsteht. Dem wurde auf zweierlei Weise Rechnung getragen.

Um den Aufwand für die Formulierung der Regeln möglichst gering zu halten, erfolgt die Erfassung der Regeln in grafischer Form direkt im Modellierungswerkzeug. Das hat neben der einheitlichen Oberfläche für die Modellierung von Prozessen und Regeln auch den Vorteil, dass Modellierungsobjekte aus den Prozessen einfach in den Regeln wiederverwendet werden können. Die textuelle Erstellung der Validierungsregel würde das Verfahren der Wiederverwendung von grafischen Prozesselementen nicht ermöglichen. Dennoch stellt die Modellierung der Regeln einen Zusatzaufwand für den Modellierer dar, der sich bei der einmaligen Anwendung der Regel auf das aktuell modellierte Szenario nur in größeren, komplexen Anwendungsfällen rentiert.

Kann die modellierte Regel aber wiederholt für die Modellprüfung eingesetzt werden, steigt der Nutzen für die Modellierer. Da SOA-Projekte die langfristige Pflege und Weiterentwicklung von IT-Landschaften zum Ziel haben, ist zu erwarten, dass die modellierten Szenarien in der Zukunft überarbeitet werden müssen. Bei der Überarbeitung können die

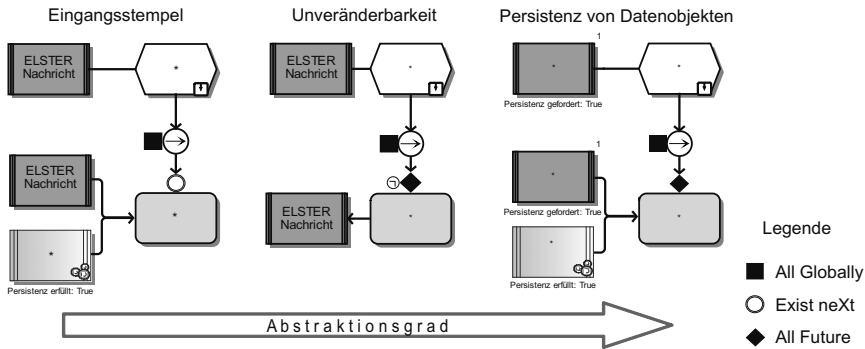


Abbildung 2: Beispielregeln für die drei Regeltypen.

bestehenden Regeln erneut angewandt werden um zu prüfen, ob die geänderten Modelle diesen Regeln entsprechen.

Neben der Regelformulierung direkt im Modellierungswerkzeug und der wiederholten Nutzung von Regeln für geänderte Prozesse, ist auch die szenarioübergreifende Nutzung der Validierungsregeln anzustreben. Dazu ist die Erweiterung des in [FFS08] vorgestellten Regelmechanismus nötig. Die notwendige Erweiterung umfasst zwei Hauptaufgaben.

Zum einen werden die Validierungsregeln in 3 Typen unterteilt. Demnach gibt es *szenariospezifische*, *domänenspezifische* und *allgemeingültige Regeln*, wobei der Abstraktionsgrad, der Umfang der Geltungsdomäne und der Wiederverwendungsgrad der Regel ansteigend sind.

Neue Regeln, die bei der Modellierung eines Szenarios erstellt werden, sind zunächst oft szenariospezifisch. Wenn sich die Wiederverwendbarkeit einer solchen Regel in weiteren Szenarien abzeichnet, ist zu untersuchen, ob sie in verallgemeinerter Form abgebildet werden kann, die dann entweder für alle Szenarien einer Anwendungsdomäne oder für alle zu betrachtenden Szenarien gültig ist. Die Regel *Eingangsstempel* ist szenariospezifisch, d.h. sie gilt nur für den in Abbildung 1 dargestellten Prozess und kann nicht in anderen Szenarien eingesetzt werden. Sie bezieht sich auf ein einzelnes, konkretes Datenobjekt und bildet eine Eigenschaft ab, die in anderen Szenarien nicht zwingend benötigt wird. Würde beispielsweise der dargestellte Prozess das Objekt an einen Service weitergeben, der selbst als Prozess modelliert wurde, dann wird für diesen (Sub-)Prozess die Eingangsstempel-Eigenschaft nicht mehr benötigt. Die Regel *Unveränderbarkeit* ist domänenspezifisch, da sie sich auf ein Datenobjekt der Domäne Steuerverwaltung bezieht. In dieser Domäne soll die mit der Regel abgebildete Eigenschaft für alle modellierten Prozesse gelten. Die Regel *Persistenz von Datenobjekten* ist allgemeingültig, da sie sich auf eine allgemeine Eigenschaft von Datenobjekten und Services bezieht.

Eine Herausforderung bei der Erstellung der Validierungsregeln besteht darin, dass die beiden Zielstellungen der leichten Erstellbarkeit bzw. Verständlichkeit der Regeln und ihre möglichst große Wiederverwendbarkeit nicht immer miteinander vereinbar sind. So sind szenariospezifische Regeln oft leicht verständlich und einfach zu erstellen, aber nur in

geringem Maße wiederverwendbar. Sehr weit abstrahierte, allgemeingültige Regeln sind prinzipiell gut wiederverwendbar, aber oft schwierig zu verstehen. In solchen Fällen ist das Ergebnis der Validierung mit der Aussage, dass ein Prozess einer bestimmten Regel nicht entspricht, für den Modellierer schwierig zu interpretieren. Deshalb ist es nicht trivial, den "richtigen" Abstraktionsgrad einer Validierungsregel und damit auch den Regeltyp zu bestimmen. Allgemeingültige oder domänenspezifische Regeln müssen dabei nicht in jedem Szenario von Bedeutung sein: Wenn z.B. die Persistenz von Datenobjekten in einem modellierten Szenario keine Rolle spielt, sind die für die Regel *Persistenz von Datenobjekten* entscheidenden Attribute nicht befüllt, diese Regel kommt daher nicht zur Ausführung und der Modellierer erhält keine Fehlermeldung. Szenariospezifische Regeln dagegen sollten auf das betreffende Szenario anwendbar sein, ansonsten sind sie entweder überflüssig oder aber es liegt die Vermutung nahe, dass ein Modellierungsfehler vorliegt. Deshalb erhält der Benutzer mit den Ergebnissen der Validierung die Information, welche szenariospezifischen Regeln nicht anwendbar sind. Für die (vermutlich weit größere) Zahl von domänenspezifischen und allgemeingültigen Regeln wird eine solche Information nicht ausgegeben.

Zum anderen muss der Validierungsmechanismus aus [FFS08] auf *Aussagen zu Eigenschaften bzw. Attributen der Modellelemente* erweitert werden. Szenariospezifische Regeln wie die in Abschnitt 3 beschriebene Regel "Eingangsstempel" treffen Aussagen über konkrete Objekte. Da Benennungen von Objekten durch den Modellierer oft frei vergeben werden, sind einer Wiederverwendung solcher Regeln in anderen Szenarien enge Grenzen gesetzt. Wiederverwendbare Regeln können nur über Eigenschaften von Objekten gesteuert werden, die szenarioübergreifend von Bedeutung sind. Bei der in Abschnitt 3 beschriebenen Regel "Persistenz von Datenobjekten" werden bspw. 2 Attribute genutzt, die Aussagen darüber treffen, ob für ein Datenobjekt (bzw. eine Nachricht) Persistenz gefordert wird bzw. ob ein aufgerufener Service die Persistenz herstellt. Dabei ist es aus Sicht des Prozessmodellierers von Vorteil, dass diese Objekte nicht bei der Modellierung eines einzelnen Prozesses erzeugt werden, sondern prozessunabhängig zur Verfügung gestellt werden. Dadurch entsteht für den Modellierer in diesem Fall weder für die Formulierung der (bereits vorhandenen) Regel noch für die Erfassung der Validierungsattribute ein Zusatzaufwand.

5 Technische Realisierung

Der Einsatz von Modellprüfungstechniken erfordert deren Einbindung in den Modellierungsprozess von Geschäftsprozess- bzw. Integrationsmodellen. In Abbildung 3 ist ein mögliches Vorgehensmodell dargestellt. Die Modellierung beginnt dabei mit einem neu erstellten Modell oder (falls vorhanden) basierend auf einem Referenzmodell. Um einem neu erstellten Modell die Zuordnung von existierenden Validierungsregeln zu ermöglichen, besitzt das Modell Attribute, über welche sich die Szenario- oder Domänenzugehörigkeit festlegen lässt. Bei Referenzmodellen sind diese Attribute bereits gesetzt. Durch die Zuordnung der Modelle zu Szenarien und Domänen werden dem Modell aus dem *Datenbestand Validierungsregeln* die entsprechenden Regeln, sowie die allgemeingültigen Regeln,

zugeordnet. Der zugeordnete Regelbestand kann dann zur Validierung des entsprechenden Modells genutzt werden. Darüber hinaus können die Regeln angepasst oder neue Regeln hinzugefügt werden. Dabei ist zu berücksichtigen, dass die neuen Regeln nicht direkt in den Gesamtbestand aufgenommen werden sollten. Vielmehr sind neue bzw. angepasste Regeln in einem gesondert dem Modell zugeordneten Datenbestand abzuspeichern. Diese Datenbestände können dann über Analysen in den Gesamtdatenbestand übergehen.

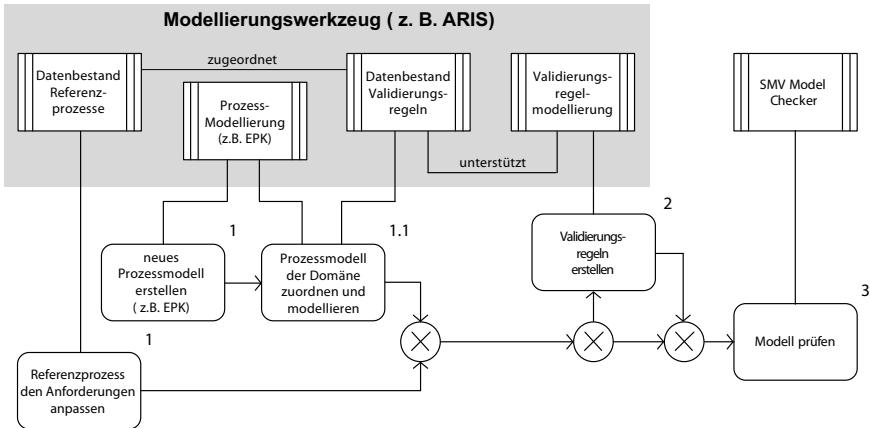


Abbildung 3: Vorgehensmodell der Modellierung und Validierung eines Geschäftsprozessmodells.

Für die prototypische Umsetzung des Verfahrenmodells dient, wie auch in [FFS08], der ARIS SOA Architekt als Modellierungswerkzeuge. Die Grundlagen der technischen Realisierung von Modellprüftechniken im ARIS SOA Architekt werden in [FFS08] vorgestellt. Darauf basierend erfolgte eine Anpassung bzw. Erweiterung für den in diesem Beitrag vorgestellte Validierungsmechanismus. Als Neuerung existiert nur der *Datenbestand Validierungsregeln*. Dieser Datenbestand ist im ARIS über verschiedene Datenbanken verteilt. Die verschiedenen Datenbanken entsprechen den vorhandenen Domänen und enthalten somit jeweils die allgemeingültigen sowie die domänen- und szenariospezifischen Regeln. Die Zuordnung der Regeln zu den Modellen erfolgt im ARIS SOA Architekt somit über die Datenbank bzw. der direkten Zuordnung der szenariospezifischen Regeln.

6 Zusammenfassung und Ausblick

Die Grundlage dieses Beitrags ist ein Verfahren zur grafischen Modellierung von Validierungsregeln und deren Nutzung für automatisierte Modellprüfungsverfahren (Model Checking) zur Sicherung der semantischen Korrektheit von Prozessmodellen wie dem Geschäftsprozessmodell der ereignisgesteuerten Prozesskette (EPK). Das hier vorgestellte Verfahren ermöglicht die Wiederverwendung der grafisch formulierten Validierungsregeln. Dies wird über eine Kategorisierung der Regeln erreicht. Dabei unterteilen sich

die Regeln nach ihrer inhaltlichen Reichweite. Bei einer kontinuierlichen Nutzung des Verfahrens über einen längeren Zeitraum wird eine Regeldatenbank aufgebaut, die allgemeingültige, domänenspezifische und szenariospezifische Regeln enthält.

Die bessere Wiederverwendbarkeit der Regeln und die grafische Darstellungsform vereinfacht, wie oben beschrieben, die Formulierung der Regeln gegenüber einer textbasierten Erstellung. Dennoch ist für die Regelerstellung ein grundsätzliches Verständnis der zugrundeliegenden Mechanismen erforderlich. Für die Zukunft ist geplant, die Regelerstellung durch die Verwendung von Mustern bzw. Regelbausteinen sowie durch Assistenten noch besser zu unterstützen. Der Einsatz von Mustern ermöglicht die effiziente Erstellung komplexer Regeln. Ebenso kann eine angemessene Benutzerführung durch dialogbasierte Assistenten eine wertvolle Unterstützung für den Modellierer darstellen.

Literatur

- [Bus08] Business Rules Group. What is a Business Rule? <http://www.businessrulesgroup.org/defnbrg.shtml>, December 2008.
- [DH08] Jens Drawehn und Oliver Höß. Ausführung von Geschäftsprozessen. In Dieter Spath und Anette Weisbecker, Hrsg., *Business Process Management Tools 2008 - Eine evaluierende Marktstudie zu aktuellen Werkzeugen*. Fraunhofer IRB Verlag, 2008.
- [End04] Rainer Endl. *Regelbasierte Entwicklung betrieblicher Informationssysteme*. Eul, J, 2004.
- [FFS08] Sven Feja, Daniel Fötsch und Sebastian Stein. Grafische Validierungsregeln am Beispiel von EPKs. In *Software Engineering 2008, Fachtagung des GI-Fachbereichs Softwaretechnik, 18.-22.2.2008 in München*, LNI, Seiten 198–204. GI, 2008.
- [FKT08] Klaus-Peter Fähnrich, Stefan Kühne und Maik Thränert, Hrsg. *Model-Driven Integration Engineering*. Universität Leipzig, 2008.
- [Ger04] Rik Gerrits. Business rules, can they be re-used? *Business Rules Journal*, 5, 2004.
- [HH00] David Hay und Keri Anderson Healy. Defining Business Rules What Are They Really? http://www.businessrulesgroup.org/first_paper/BRG-whatBR_3ed.pdf, July 2000.
- [HWS07] Oliver Höß, Anette Weisbecker, Thomas Specht und Jens Drawehn. Migration zu serviceorientierten Architekturen - top-down oder bottom-up? In Knut Hildebrand, Hrsg., *HMD - Praxis der Wirtschaftsinformatik*, Jgg. 257, Seiten 39–46. dpunkt.verlag, 2007.
- [MLM⁺06] C. Matthew MacKenzie, Ken Laskey, Francis McCabe, Peter F Brown und Rebekah Metz. Reference Model for Service Oriented Architecture 1.0, August 2006.
- [PD08] Oliver Pape und Jens Drawehn. Anwendungsszenario Elektronische Steuererklärung. In Klaus-Peter Fähnrich, Stefan Kühne und Maik Thränert, Hrsg., *Model-Driven Integration Engineering*, Seiten 157–170. Universität Leipzig, 2008.
- [SVEH07] Thomas Stahl, Markus Völter, Sven Efftinge und Arno Hase. *Modellgetriebene Softwareentwicklung - Techniken, Engineering, Management*. dpunkt.verlag, 2007.
- [TS07] Stefan Tilkov und Gernot Starke. SOA aus technischer Perspektive. In *SOA-Expertenwissen*, Kapitel 1.3, Seite 16ff. dpunkt.verlag, 2007.