

Durchgängige Modellierung von Geschäftsprozessen in einer Service-orientierten Architektur

Stephan Buchwald¹, Thomas Bauer² und Manfred Reichert³

Abstract: Häufig genannte Ziele einer Service-orientierten Architektur (SOA) sind die bessere Unterstützung und Anpassbarkeit von Geschäftsprozessen sowie das Business-IT-Alignment. Diese werden heute nicht erreicht, da die bei der Implementierung eines Fachprozesses notwendigen komplexen Transformationen in einen ausführbaren Workflow schwer nachvollziehbar sind. Dadurch gehen fachliche Anforderungen verloren und es entsteht ein hoher Aufwand bei späteren Prozessanpassungen. Im vorgestellten Ansatz wird ein sog. Abbildungsmodell eingeführt, das Zugehörigkeiten von Aktivitäten des Fachmodells zu denen des Systemmodells (d.h. technische Spezifikation des Informationssystems) explizit dokumentiert. Dadurch werden im Software-Entwicklungsprozess automatisierte Konsistenzprüfungen zwischen den Modellebenen möglich. Werden später Prozessanpassungen erforderlich, so lassen sich die zu einer fachlichen Aktivität gehörenden technischen Aktivitäten direkt erkennen, was die Durchführung der Anpassung erleichtert. Ein wesentlicher Vorteil unseres Ansatzes besteht darin, dass die Erstellung des Abbildungsmodells nur einen minimalen Aufwand verursacht, da keine komplexen Regeln, sondern nur einfache Beziehungen definiert werden müssen.

1 Motivation

Workflow-Technologie alleine reicht nicht aus, um die Anforderung einer *Service-orientierten Architektur (SOA)* an *Flexibilität* zu erfüllen. Gegenüber der heutigen Praxis ist insbesondere eine Verbesserung des Zusammenspiels zwischen Fachbereichen und IT-Abteilungen bei der Software-Entwicklung erforderlich. Dieser Aspekt wird auch als *Business-IT-Alignment* bezeichnet: Informationssysteme sollen die fachlichen Anforderungen exakter treffen als bisher. Hierzu müssen Informationsverlust und Verfälschungen bei der Entwicklung des (prozessorientierten) Informationssystems reduziert werden. Falls Änderungen an fachlichen Anforderungen erfolgen, sollen diese korrekt, unverfälscht und zeitnah in die Implementierung des Informationssystems einfließen. Außerdem kann es in einer SOA zu Änderungen in der Umgebung kommen, z.B. wenn Services wegfallen (d.h. abgeschaltet werden) oder zu einer neuen Version migrieren.

Um solchen Szenarien flexibel zu begegnen, ist eine zusätzliche Modellebene notwendig, welche die Beziehungen zwischen fachlichen Anforderungen und entsprechender IT-Implementierung nachvollziehbar dokumentiert. Wir verwenden hierfür ein Zwischenmodell, welches im Folgenden *Systemmodell* genannt wird. In dieses Systemmodell müssen fachliche Anforderungen an einen Prozess einfließen. Nur dann werden sie in der tatsächlichen Implementierung des Informationssystems umgesetzt. Hierzu ist ein

¹ Abteilung für Daten- und Prozessmanagement, Daimler AG, stephan.buchwald@daimler.com

² Abteilung für Daten- und Prozessmanagement, Daimler AG, thomas.tb.bauer@daimler.com

³ Institut für Datenbanken und Informationssysteme, Universität Ulm, manfred.reichert@uni-ulm.de

Konzept erforderlich, bei dem die Beziehung zwischen fachlichen Anforderungen und ihrer Implementierung im geplanten Informationssystem nachvollziehbar ist. So muss für jede Aktivität des fachlichen Modells (und damit für deren Eigenschaften und Anforderungen) ableitbar sein, in welche Aktivität(en) des Systemmodells sie eingeflossen ist (vgl. Abb. 1). Dies ist normalerweise nicht ohne weiteres erkennbar, da für IT-Implementierungen meist Vorgaben für Aktivitätenbezeichnungen bestehen, die nicht mit denen der Fachabteilungen übereinstimmen.

Ein Beispiel einer einfachen Transformation zeigt Abb. 1 mit der Umbenennung der fachlichen Aktivität [Änderungsantrag stellen] in [HT...Request-Change] im Systemmodell. Während solche Namensunterschiede noch einfach handhabbar sind, sind bei der Abbildung eines Fachmodells auf ein ausführbares Modell meist größere Umstrukturierungen notwendig. Grund dafür ist, dass ein Fachbereich seine Geschäftsprozesse im Regelfall auf einer anderen Abstraktionsebene beschreibt als sie für eine IT-Spezifikation benötigt wird. So werden in einem Fachprozessmodell zahlreiche manuelle Einzelaktivitäten beschrieben, die nicht automatisiert werden sollen, die aber für Verfahrenshandbücher oder Prozesskostenrechnungen relevant sind. Diese werden in einer IT-Spezifikation nicht 1:1 übernommen, sondern zusammengefasst oder ganz weggelassen. Ein Beispiel ist die Aktivität [Änderung umsetzen] in Abb. 1. IT-unterstützte Prozessschritte hingegen sind im Fachprozess häufig nur grob beschrieben, so dass sie verfeinert oder sogar zusätzlich hinzugefügt werden müssen. Andere fachliche Aktivitäten werden in mehrere IT-gestützte Aktivitäten (Benutzerinteraktionen, Service-Aufrufe und Dateninformationen) aufgespalten. So wird in unserem Beispiel die Aktivität [betroffene Bauteile angeben] in eine Benutzerinteraktion (Human Task) und einen Service-Aufruf aufgespalten. Schließlich ist zu beobachten, dass Fehler- und Ausnahmefälle (z.B. Rücksprünge bei unvollständigen Daten) in Fachprozessmodellen meist nicht abgebildet werden, um die Komplexität in dieser frühen Phase gering zu halten.

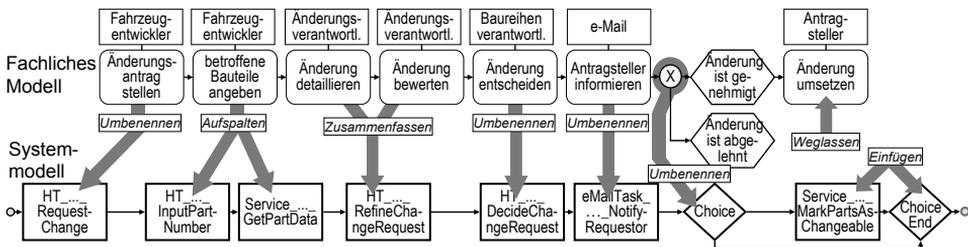


Abb. 1: Transformationen zwischen fachlichem Modell und Systemmodell

Unter Berücksichtigung solcher Transformationen ermöglicht unser Ansatz *Enhanced Process Management by Service Orientation (ENPROSO)* die Nachvollziehbarkeit der Beziehungen zwischen einer fachlichen Aktivität und ihrer IT-Implementierung. Ebenso wird Nachvollziehbarkeit in umgekehrter Richtung unterstützt: So ist es für die robuste Ausführung einer SOA-Anwendung wichtig, die von Umgebungsänderungen betroffenen Prozesse und Aktivitäten zu identifizieren. Nur so wird es möglich, entsprechend schnell auf anstehende Änderungen wie “Service-Abschaltungen“ reagieren zu können.

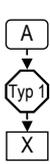
In Kapitel 2 stellen wir die Anforderungen an das zu entwickelnde Konzept vor, insbesondere hinsichtlich der zu unterstützenden Prozesstransformationen. Kapitel 3 skizziert das Abbildungsmodell unseres ENPROSO-Ansatz und beschreibt die dazu notwendigen Prozesstransformationen. Der Ansatz wird in Kapitel 4 detailliert. Kapitel 5 diskutiert verwandte Arbeiten, bevor mit einer Zusammenfassung geschlossen wird.

2 Rahmenbedingungen und Anforderungen

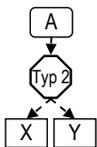
Die Modellierung von Geschäftsprozessen aus fachlicher Sicht dient, neben der Prozessanalyse und -optimierung, vor allem der Dokumentation fachlicher Anforderungen seitens der Fachabteilungen und Prozessverantwortlichen. Da Letztere meist wenig oder keinen IT-Hintergrund haben, werden die Inhalte eines *fachlichen Modells* nicht formal beschrieben. Stattdessen werden einfache graphische Notationen und textuelle Beschreibungen verwendet, wie sie von Geschäftsprozess-Modellierungswerkzeugen angeboten werden (z.B. erweiterte Ereignisgesteuerte Prozess-Ketten (eEPK) in ARIS). Wie erwähnt, sind in dieser frühen Phase noch nicht alle Aspekte im Detail bekannt bzw. sollen aus Komplexitätsgründen noch nicht modelliert werden, so dass das fachliche Prozessmodell (kurz: Fachprozess) an bestimmten Stellen bewusst vage und offen gehalten wird. Diese Unvollständigkeit betrifft die Prozessstruktur (d.h. den Kontrollfluss) selbst, aber auch andere Aspekte (z.B. erfolgt im Fachprozess noch keine detaillierte Festlegung von Datenstrukturen). Hingegen muss ein implementierter Workflow von einer Workflow-Engine ausgeführt werden, weshalb die entsprechende technische Prozessbeschreibung (*ausführbares Modell*) vollständig und formal sein muss. Außerdem muss sie den Vorgaben des von der Workflow-Engine verwendeten Metamodells genügen (z.B. [Rei00]).

Bei der *Transformation* eines Fachprozesses in einen Systemprozess ist es wichtig, dass das Systemmodell geeignet angepasst (d.h. umstrukturiert) wird. Da die durchgängige Realisierung solcher Umstrukturierungen den Schwerpunkt dieses Beitrags bildet, werden die verschiedenen Typen von Strukturänderungen entlang des in Abb. 1 dargestellten (stark vereinfachten) Antragsprozesses für Produktänderungen in der Automobilindustrie motiviert. Dieser stellt sicher, dass Änderungsvorhaben an Bauteilen vor ihrer eigentlichen Umsetzung bewertet, genehmigt und entsprechend dokumentiert werden. Ein Änderungsvorhaben wird angelegt, indem die Änderung in Form eines Antrages beschrieben wird. Da sich Änderungen meist auf mehrere Bauteile auswirken, müssen Informationen über die von ihnen betroffenen Bauteile eingeholt werden. Anschließend wird die Änderung detailliert und bewertet. Abhängig von dieser Bewertung wird entschieden, ob das Änderungsvorhaben umgesetzt wird oder nicht.

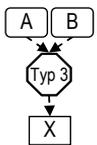
Im Folgenden werden häufig vorkommende Transformationstypen beschrieben.



Typ 1 (Übernahme einer Aktivität inkl. Umbenennung): Im einfachsten Fall wird eine Aktivität des Fachprozesses auf genau eine Aktivität des Systemmodells abgebildet. So kann eine Benutzerinteraktion zum Ausfüllen eines Formulars z.B. als *Human Task* [Agr07] in einem BPEL-Prozess realisiert werden. Da für Aktivitäten auf IT-Ebene aber häufig Namenskonventionen existieren (z.B. um die zugehörige Applikation im IT-Betrieb erkennen zu können), ergeben sich meist unterschiedliche Namen für Aktivitäten und Daten im fachlichen Modell und im Systemmodell. Zwecks Nachvollziehbarkeit müssen wir solche Anpassungen explizit verwalten. Beispielsweise wird die fachliche Aktivität [Änderungsantrag stellen] durch die Human Task [HT ..._ChangeAppl_ProductDevelopment_RequestChange] im Systemmodell realisiert (vgl. Abb. 1).



Typ 2 (Aufspalten einer Aktivität): Service-orientierte Workflow-Engines erfordern eine strikte Unterscheidung von Aktivitäten mit Benutzerinteraktion (*Human Tasks*) und Service-Aufrufen (Invoke-Aktivität in BPEL). Diese Aufspaltung ist neu: Klassische Workflow-Engines haben Aktivitäten als größere Einheiten betrachtet, die sowohl mit dem Benutzer interagieren als auch Daten mit Backend-Systemen austauschen. Da solche Einheiten aber kaum wiederverwendbar sind, entsprechen sie nicht der Grundphilosophie einer SOA. Im Beispiel aus Abb. 1 ist es deshalb notwendig, die fachlich zusammengehörige Aktivität [betroffene Bauteile angeben] in eine Benutzerinteraktion zur Eingabe der Bauteilnummern und einen Service-Aufruf zur Ermittlung der restlichen Bauteildaten aus einem Produktdaten-Management (PDM)-System aufzuspalten



Typ 3 (Zusammenfassen von Aktivitäten): Bei der fachlichen Analyse von Geschäftsprozessen werden logisch zusammengehörige Aufgaben ermittelt, die mittels separaten Aktivitäten modelliert werden. Wenn solche Aktivitäten z.B. unmittelbar aufeinander folgen und stets von derselben Person erledigt werden, kann es Sinn machen, diese im Systemmodell zu einer Aktivität zusammenzufassen, um sie dann durch eine Sequenz von Bildschirmmasken zu bearbeiten. Im dargestellten Beispiel wird es dem Änderungsverantwortlichen dadurch erspart, nach Beendigung der Detaillierung, dieselbe Workflow-Instanz in seiner Arbeitsliste erneut zu suchen, um anschließend die Bewertung der Änderung durchzuführen.



Typ 4 (Einfügen zusätzlicher Aktivitäten in das Systemmodell): Nachdem der Baureihenverantwortliche eine Entscheidung über die Änderung getroffen hat und der Antragsteller entsprechend informiert ist, kann die Änderung durchgeführt werden. Damit dies im PDM-System möglich ist, müssen dort die entsprechenden Bauteile in den Zustand *Veränderbar* versetzt werden. Dies erfolgt durch den Serviceaufruf [*Service_MarkPartsAsChangable*], der zusätzlich in das Systemmodell eingefügt wird. Oftmals sind auch zusätzliche Aktivitäten zur Protokollierung relevanter Aktionen oder eingetretener Fehler auf Workflow-Ebene erforderlich.



Typ 5 (Weglassen von Aktivitäten aus dem fachlichen Modell): In Fachprozessen können sich häufig Aktivitäten befinden, deren Ausführung nicht vom Workflow-System gesteuert bzw. überwacht werden soll. So erfolgt in unserem Beispiel die Umsetzung einer Änderung selbstständig durch den Antragssteller, nachdem dieser informiert wurde, dass sein Antrag genehmigt ist. Die Modellierung dieser Aktivität im Fachprozess ist zwar sinnvoll, da sie nicht unerheblich zu den Prozesskosten und -durchlaufzeiten beiträgt (sie fließt z.B. bei einer Prozess-Simulation ein), ist aber für die Workflow-Ebene nicht relevant.

Ferner können weitere Transformationstypen definiert werden (z.B. n zu m Aktivitäten).

3 Konzept für die Prozesstransformation

Wir entwickeln nun ein Konzept, um ausgehend von einem fachlichen Prozessmodell die erwähnten Transformationen durchzuführen. Hierzu wird eine mehrstufige Vorgehensweise vorgestellt: Zuerst wird der Fachprozess in einen korrespondierenden Prozess auf Systemmodellebene (Systemprozess) transformiert. Aus diesem wird später das ausführbare Prozessmodell abgeleitet. Außerdem wird aufgezeigt, wie Beziehungen zwischen diesen Ebenen in Prozessmodellierungsumgebungen explizit dokumentiert werden können. Dieser Aspekt wird in Kapitel 4 detailliert.

Ebenen der Prozessmodellierung: Der Fachprozess dient der Dokumentation von fachlichen Anforderungen an das zu realisierende Informationssystem. Fachliche Anforderungen werden meist durch Befragung von Endanwendern und Prozessverantwortlichen bzw. -eignern ermittelt. Häufig wird diesen Personen der Fachprozess graphisch dargestellt, so dass sie ihre eigenen Prozessschritte in den Fachprozess einordnen oder Fehler erkennen können. Deshalb ist die wichtigste Anforderung an ein fachliches Modell, dass es leicht verständlich ist. Zudem liegt die Verantwortung für die Erstellung des *Fachmodells* meist bei den jeweiligen Fachbereichen, auch wenn die operative Umsetzung dieser Aufgabe oftmals durch (externe) Berater erfolgt. Für die Inhalte sind die Fachbereiche verantwortlich, da nur diese das entsprechende Fachwissen haben. Bei der Modellierung werden primär die Struktur des Prozessablaufs (Kontrollfluss) mit seinen Aktivitäten, deren Ein- und Ausgabedaten sowie zuständigen Bearbeitern festgelegt.

Die Verantwortung für die Erstellung des *Systemmodells* liegt beim IT-Bereich. Durchgeführte Änderungen am Systemmodell sollten vom zuständigen Fachbereich bestätigt werden. Deshalb muss die Darstellung des Systemmodells (z.B. graphische Notation, Struktur) so gewählt werden, dass sie für Fachanwender verständlich ist. Die zugehörigen Inhalte sind dieselben wie im fachlichen Modell. Allerdings müssen diese ausreichend detailliert, vollständig und formal sein, um die Basis einer plattformunabhängigen IT-Spezifikation bilden zu können. Das bedeutet, vage textuelle Beschreibungen und offene Aspekte aus dem Fachmodell müssen ersetzt werden.

Die Software-Realisierung (*ausführbares Modell*) hingegen wird von IT-Implementieren erstellt. Diese treffen keine (fachlich relevanten) Entscheidungen bei der Erstellung des

werden (und kein Kontrollfluss). Dies ist einfach möglich, da keine Überführung des Fachprozesses in ein anderes Metamodell notwendig ist.

Es lassen sich alle Transformationstypen dokumentieren. Der Ansatz ist erweiterbar, indem zusätzliche Transformationstypen für evtl. zukünftig benötigte Transformationen definiert werden. Alle Veränderungen sind direkt erkennbar und die Beziehungen von Aktivitäten des Fach- und Systemmodells sind bidirektional nachvollziehbar.

4 Gestaltung des Abbildungsmodells

Ein Abbildungsmodell stellt das Verbindungsglied zwischen Fachprozess und Systemprozess dar. Es wird von heutigen Prozessmodellierungswerkzeugen nicht direkt unterstützt. Wir zeigen im Folgenden, wie ein Abbildungsmodell prinzipiell gestaltet werden muss, damit die Beziehung zwischen Fach- und Systemprozess nachvollziehbar ist.

Erstellung und Speicherung: Das Abbildungsmodell wird während der Erstellung des Prozesses auf Systemmodell-Ebene (vgl. Abb. 2) erzeugt. Da beide Modelle in der Verantwortung des IT-Bereichs liegen, sollten dasselbe Modellierungswerkzeug und möglichst dieselbe Notation verwendet werden. Allgemein betrachtet definiert das Abbildungsmodell eine Menge von Relationen, die Aktivitäten des Fachprozesses auf Aktivitäten des Systemmodells abbilden. Jede dieser Relationen entspricht genau einem Transformationstyp. In der Modellierungsumgebung sollte eine Relation durch jeweils ein Objekt realisiert werden, dem außer dem Transformationstyp auch Attribute wie Name, Beschreibung oder der Verantwortliche aus dem Fachmodell zugeordnet sind.

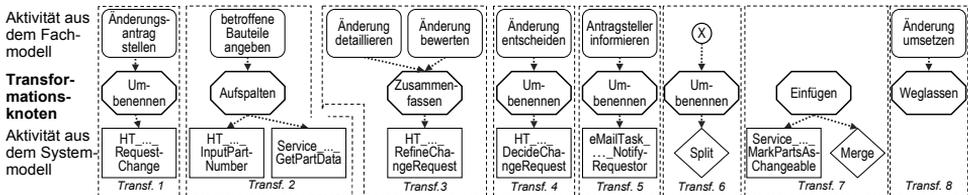


Abb. 3: Abbildungsmodell für das in Abb. 1 vorgestellte Beispielszenario

Da heutige Modellierungswerkzeuge das Konzept des Abbildungsmodells nicht unterstützen, muss bei ihrer Verwendung für diesen Zweck ein existierender Modelltyp verwendet werden (z.B. eEPK, BPMN oder UML Activity Diagram). Dieser muss Aktivitäten aus dem Fach- und Systemprozess sowie Relationen zwischen diesen darstellen können. Je nach Notation und Werkzeug kann hiervon ein Modelltyp für Abbildungsmodelle abgeleitet werden. In diesem abgeleiteten Modell können dann spezielle Knotentypen für Transformationen (Transformationsknoten) sowie Kanten für Transformationskanten verwendet werden. Abb. 3 zeigt exemplarisch das zu dem in Abb. 1 eingeführten Beispiel gehörende Abbildungsmodell in einer „neutralen“ Notation. Hier werden die Transformationsknoten durch Achtecke repräsentiert, um sie leicht von Aktivitäten unterscheiden zu können; Transformationskanten werden gestrichelt dargestellt.

Verwendung des Abbildungsmodells: Durch Konsistenzanalysen wird es nun möglich, eine schnelle Zuordnung (und damit Umsetzung) von geänderten fachlichen Anforderungen zu Aktivitäten des ausführbaren Modells zu erhalten:

1. Nach Änderung des Fachprozesses werden dessen Aktivitäten in das Abbildungsmodell importiert. Die Konsistenzanalyse vergleicht anschließend die im Abbildungsmodell aktualisierte Menge fachlicher Aktivitäten mit der bereits vorhandenen. Sind bestimmte Aktivitäten nicht mehr enthalten, wurden diese aus dem Fachprozess gelöscht (bzw. zusätzliche wurden erzeugt).
2. Verwendet das Werkzeug zur Fachprozessmodellierung zugreifbare Zeitstempel für Aktivitäten-Objekte, sind Veränderungen einzelner fachlicher Aktivitäten erkennbar. Die Konsistenzanalyse vergleicht nicht nur die aktualisierte mit der bereits vorhandenen Aktivitätsmenge, sondern zusätzlich die Zeitstempel einzelner Aktivitäts-Objekte.
3. Analog können bei Veränderungen in der Umgebung der Prozessapplikation, etwa die Abschaltung eines Services (vgl. Abb. 1, *Service_GetPartData*), mit diesem Ansatz leicht betroffene fachliche Aktivitäten (*betroffene Bauteile* angeben) ermittelt werden.

5 Verwandte Arbeiten

MDA-basierte Ansätze wenden Patterns auf fachliche Modelle an [All07, Bau08, FKT08]. Allerdings sind solche Ansätze nicht immer realisierbar. Dies trifft auch auf unser Szenario zu, bei dem eine *freie* Modellierbarkeit des Systemmodells gefordert ist: So ist für Geschäftsprozesse, die fachlich meist grob, sehr abstrakt und vage beschrieben sind, eine strukturelle Adaption auf Systemmodell-Ebene notwendig. Eine solche Überarbeitung mittels automatisch anwendbaren Patterns zu realisieren, wäre zu aufwendig, da umfangreiche Transformationen fachlicher Objekte auf Objekte im Systemmodell formal spezifiziert werden müssten. Zudem sind Transformationen, wie das Einfügen von zusätzlichen Aktivitäten, schwer realisierbar, da für diese keine entsprechende Aktivität im fachlichen Modell existiert. In dem von uns betrachteten Anwendungsfall ist eine Wiederverwendung vordefinierter Patterns in unterschiedlichen Geschäftsprozessen nicht realistisch. Deshalb muss für jeden Geschäftsprozess individuell entschieden werden, wie eine entsprechende technische Repräsentation im Systemmodell aussieht.

6 Zusammenfassung

Von Fachbereichen erstellte Prozessmodelle müssen strukturell adaptiert werden, bevor sie mittels Prozess-Management-Systemen implementiert werden können. Hierbei ist es das Ziel, möglichst schnell und nachvollziehbar von fachlichen Anforderungen zur IT-Implementierung zu gelangen (Business-IT-Alignment). Um dies zu erreichen, haben wir zwischen dem fachlichen und dem ausführbaren Prozessmodell die Ebene des Sys-

temmodells eingeführt, das in Verantwortung des IT-Bereichs liegt und als Spezifikation für die IT-Implementierung dient. Um bei der Modellierung des Systemmodells die durchgeführten Prozesstransformationen nachvollziehbar zu gestalten, haben wir das Konzept des Abbildungsmodells entwickelt. Es ermöglicht weiterhin, die bei der Erstellung des Systemmodells benötigte „freie Geschäftsprozess-Modellierung“, aber erzielt dennoch einen hohen Grad an Nachvollziehbarkeit. Diese Nachvollziehbarkeit wird genutzt, um Prozessimplementierungen schneller erstellen bzw. anpassen zu können. Sie gewährleistet zudem die Konsistenz zwischen den verschiedenen Modellebenen. Entwurfsalternativen und eine prototypische Umsetzung sind in [BBR09] dargestellt.

Literaturverzeichnis

- [Agr07] A. Agrawl et al.: WS-BPEL Extension for People Specification. Technical Report, Active Endpoints, Adobe, BEA, IBM, Oracle, SAP AG, 2007
- [All07] T. Allweyer: Erzeugung detaillierter und ausführbarer Geschäftsprozessmodelle durch Modell-zu-Modell-Transformationen; Proc.: 6. Workshop Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, St. Augustin, Germany, S. 23-38, 2007
- [Bau08] P. Bauler et.al.: Usage of Model Driven Engineering in the context of Business Process Management; In: Proc. 4th GI Workshop XML Integration and Transformation for Business Process Management, S. 1963 – 1974, 2008
- [BBP09] S. Buchwald, T. Bauer, R. Pryss: IT-Infrastrukturen für flexible, service-orientierte Anwendungen; In: Proc. BTW 2009, Münster, 2009
- [BBR09] S. Buchwald, T. Bauer, M. Reichert: Durchgängige Modellierung von Geschäftsprozessen durch Einführung eines Abbildungsmodells: Ansätze, Konzepte, Notationen. Technical Report UIB-2009-12, University of Ulm, 2009
- [FKT08] K.P. Fähnrich; S. Kühne; M. Thränert: Model-Driven Integration Engineering; Eigenverlag Leipziger Informatik-Verbund (LIV), 2008
- [Rei00] M. Reichert: Dynamische Ablaufänderungen in Workflow-Management-Systemen. Dissertation, Universität Ulm, Fakultät für Informatik; 2000

