

# Verzahnung von Requirements Engineering und Geschäftsprozessdesign

Matthias Weidlich, Alexander Grosskopf  
Hasso-Plattner-Institut Potsdam  
{matthias.weidlich,alexander.grosskopf}@hpi.uni-potsdam.de

Daniel Lübke, Kurt Schneider, Eric Knauss, Leif Singer  
Leibniz Universität Hannover, FG Software Engineering  
{daniel.luebke,kurt.schneider,eric.knauss,leif.singer}@inf.uni-hannover.de

**Abstract:** Geschäft und IT wachsen insbesondere in SOA-Projekten immer stärker zusammen. Hierbei wird deutlich: prozess- als auch anforderungsbezogene Rollen versuchen dasselbe Problem zu lösen — namentlich die Erhebung von Anforderungen und die Neudefinition der Unternehmung. Dies geschieht jedoch auf verschiedenen Ebenen und mit Hilfe unterschiedlicher Techniken. Da in Projekten bisher keine gegenseitige Unterstützung zwischen diesen beiden Rollen vorgesehen ist, sind redundant ausgeführte Tätigkeiten und Inkonsistenzen in der Spezifikation häufig die Folge. Die Anforderungen für SOA-Projekte könnten jedoch weitaus effizienter erhoben und verwaltet werden: durch eine verbesserte Koordination zwischen den verschiedenen Rollen, die Definition von Abhängigkeiten zwischen ihnen sowie die Fixierung der gegenseitig zu erbringenden Resultate. Eben jene Verzahnung verlangt darüberhinaus eine Reihe unterstützender Techniken. Die schnellere Erhebung sowie die verbesserte Qualität der Anforderungen erhöhen die Agilität der Projekte und stärken so einen der Hauptvorteile von SOA.

## 1 Einführung und Motivation

Die Service-orientierte Architektur (SOA) teilt die Geschäftslogik der Anwendungen einer Organisation auf sogenannte Services auf. So können alle Anwendungen gleichermaßen Funktionalität nutzen, die mit Hilfe standardisierter Technologien plattformunabhängig zur Verfügung gestellt wird.

Da die Anwendungen der Organisation ihre Funktionalität von den Services beziehen, lassen sich so neue Anforderungen an das Geschäft leichter, schneller und kostengünstiger umsetzen. Vorhandene Services können zentral angepasst werden; neue Anwendungen können aus bestehender Funktionalität orchestriert werden. Die IT kann so schnell auf die Erfordernisse des Geschäftes reagieren und notwendige Funktionalität bereitstellen.

Die Fähigkeit zur schnellen Anpassung der Geschäftsprozesse ist entscheidend für die Wettbewerbsfähigkeit eines Unternehmens, dessen Marktposition auf softwaregestützte Verfahrensabläufe begründet ist. Zwei Disziplinen der Softwaretechnik konzentrieren sich im Besonderen auf die Erfassung von Anforderungen für softwaregestützte Abläufe.

Das Requirements Engineering (RE) ist die methodische Erhebung, Dokumentation, Verhandlung, Verwaltung und Verfolgung von Anforderungen in einem Softwareentwicklungsprojekt. Die Rolle des Requirements Engineers führt eine, einige oder alle dieser Tätigkeiten aus. [FHW07].

Das Geschäftsprozessmanagement hingegen umfasst Methoden und Techniken zur Erfassung, Analyse, Optimierung und Automatisierung von Geschäftsabläufen [Wes07]. Die Basis sind explizite Prozessrepräsentationen zur Kommunikation und Prozessautomatisierung.

Projekte, die von beiden Disziplinen adressiert werden, zeichnen sich durch einen hohen Grad an fachlichem Prozesswissen aus. Wiederverwendung bestehender Funktionalität basierend auf Diensten einer SOA geht einher mit der Anpassung bestehender Softwareprodukte. Oft sind solche Projekte mit organisationaler Restrukturierung und Änderungen der Arbeitsabläufe verbunden. Daraus ergibt sich ein Zeit- und Erfolgsdruck. Um schnelle Durchlaufzeiten zu ermöglichen und gute Ergebnisse zu erzielen, sollten Prozessdesigner und Requirements Engineers eng zusammenarbeiten.

In Abschnitt 2 dieses Papiers erläutern wir ein Szenario, bei dem beide Disziplinen entkoppelt an einem Projekt arbeiten, und stellen die Problemtypen heraus. Folgend wird in Abschnitt 3 eine Arbeitsweise der engen Verzahnung beschrieben. In Abschnitt 4 erläutern wir bereits bestehende Techniken, die diese Arbeitsweise unterstützen können. Wir fassen diese in Abschnitt 5 zusammen und diskutieren mögliche weitere Schritte zur engeren Verzahnung.

## **2 Identifikation möglicher Probleme**

Die Entkopplung von Requirements Engineering und Prozessdesign kann zu einer Reihe von Problemen führen, welche wir im Folgenden exemplarisch verdeutlichen. Grundlage ist ein fiktives Szenario eines Call-Centers, das auf Flugreisen spezialisiert ist. Aufgrund einer Ausweitung der angebotenen Buchungsleistungen (Mietwagen, Hotels) wurde ein Projekt zur Umstrukturierung aufgesetzt. Im Zuge der Umstrukturierung wurden bisher nicht über Services angeschlossene Hotelketten und Mietwagenverleiher integriert. Zudem soll das Buchungssystem um die neuen Funktionen erweitert werden.

Die Umsetzung dieses Projekts besteht im Wesentlichen aus den folgenden Schritten. Ein externer Prozessberater nimmt Ist-Prozesse auf und definiert Soll-Prozesse unter Rücksprache mit der IT-Abteilung des Flugbuchungsunternehmens, um eine Abdeckung der Prozessaktivitäten durch Services sicherzustellen. Parallel dazu wird ein Requirements Engineer mit einer Anforderungsanalyse für die Erweiterung des Buchungssystems beauftragt. Nachdem sowohl die detaillierten Soll-Prozesse, als auch die Anforderungsspezifikation von der Geschäftsführung des Unternehmens abgenommen sind, beginnt die Umsetzung. Sie erfolgt einerseits durch ein IT-Systemhaus, welches die Soll-Prozesse und Schnittstellen durch Service-Orchestrierungen umsetzen soll. Andererseits wird der Hersteller der Buchungssoftware mit Erweiterungen auf Basis der Anforderungsspezifikation beauftragt.

Während der Umsetzung des Projektes ergeben sich die folgenden Probleme: Die Flugsuche

des Buchungssystems sollte Allianzen von Fluglinien optional berücksichtigen können, allerdings wird diese Funktionalität nicht von den implementierten Service-Orchestrierungen abgedeckt. Somit wird nachträglich die Implementierung eines weiteren Services notwendig. Bei der Buchung von Hotels hingegen können im Buchungssystem nicht alle angebotenen Rabattarten der Services genutzt werden. Dazu wird nachträglich eine zusätzliche Erweiterung des Buchungssystems spezifiziert. Letztlich stellt sich heraus, dass die Kategorien für Mietwagen, welche vom Service zur Mietwagensuche geliefert wurden, nicht den im Buchungssystem hinterlegten entsprechen. Eine zusätzliche Datentransformation wird notwendig, um Services und das erweiterte Buchungssystem zu koppeln.

Diese Probleme sind exemplarisch für negative Auswirkungen durch die Entkopplung von Anforderungsanalyse und Prozessdesign. Die fehlende Abstimmung zwischen Requirements Engineer und Prozessdesigner führt zu einem Versatz zwischen der Prozessimplementierung und Softwareentwicklung. Die Probleme lassen sich wie folgt kategorisieren:

#### **Fehlende Services für die in Anforderungen spezifizierte Funktionalität**

Funktionale oder nicht-funktionale Anforderungen sind nicht in den Soll-Prozessen adressiert bzw. durch Services abgedeckt.

#### **Fehlende Anbindung der in den Soll-Prozessen spezifizierten Funktionalität**

Funktionalität, welche in den Soll-Prozessen definiert und über Services implementiert ist, wird nicht in Anwendungssysteme eingebunden, da sie nicht in der Anforderungsspezifikation berücksichtigt wurde.

#### **Fehlende Funktionalität**

Notwendige Funktionalität wird weder vom Requirements Engineer, noch vom Prozessdesigner erfaßt. Grund hierfür ist die Annahme, dass die jeweils andere Rolle diese Aufgabe wahrnimmt.

#### **Duplizierte Funktionalität**

Aufgrund fehlender Abstimmung zwischen dem Requirements Engineer und dem Prozessdesigner kommt es zu einer Neudefinition eines Services anstatt zu einer Orchestrierung vorhandener Services und somit zu Redundanzen innerhalb der Service Landschaft.

#### **Inkonsistenzen zwischen spezifizierter und implementierter Funktionalität**

Gewisse Funktionalität wird sowohl in den Anforderungen, als auch in den Soll-Prozessen erfaßt, allerdings gibt es Unterschiede in der Spezifikation. Somit ist eine Anbindung der Anwendungssysteme an die Services nicht ohne weiteres möglich. Gründe dafür sind oft unterschiedliche Begriffsdefinitionen und Datenmodelle.

#### **Überflüssige Interaktionen**

Endnutzer werden mehrfach interviewt: im Rahmen der Ist-Prozess-Analyse, der Soll-Prozess-Definition und der Anforderungsanalyse.

### **3 Verzahnung von Requirements Engineering und Prozessdesign**

Die im obigen Abschnitt beschriebenen Probleme zeigen, dass eine bessere Kopplung von Requirements Engineering und Prozessdesign notwendig ist.

Eine entsprechende Verknüpfung von Prozessdesign und Requirements Engineering wurde bereits von González und Díaz [dIVGD07] beschrieben. Dieses Vorgehensmodell sieht jedoch nur eine unidirektionale Verknüpfung vor. Ein iteratives Vorgehen und kontinuierliche Sicherstellung der Konsistenz zwischen Prozessmodellen und Use-Case-Beschreibungen sind somit nicht gegeben.

In diesem Abschnitt beschreiben wir daher ein Vorgehensmodell, das diese Probleme durch Verzahnung der Rollen und Aktivitäten adressiert. Dazu betrachten wir die beteiligten Rollen, die Kommunikationsstruktur und die gemeinsam benutzten Artefakte. Ziel ist eine enge Zusammenarbeit zwischen Requirements Engineer und Prozessdesigner auf Grundlage einer gemeinsamen Spezifikation. Weil Beschreibungen dieser Rollen in der Literatur sehr uneinheitlich sind (vergl. [FHW07]), zeigen wir in Abbildung 1, wie diese Rollen in den Eingangs erwähnten Projekttypen zusammenarbeiten können.

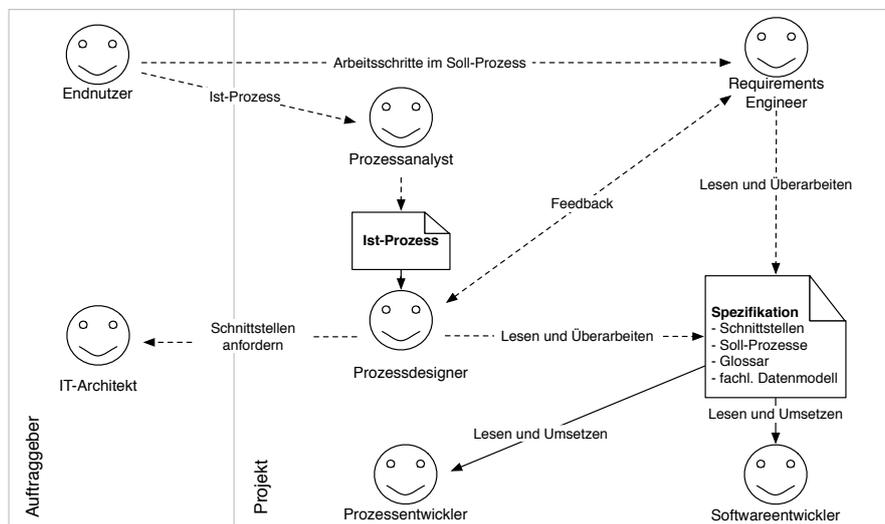


Abbildung 1: Rollen, Artefakte und Interaktionen in gemeinsamen Projekten von RE & BPM

Im ersten Schritt erarbeitet ein Prozessanalyst den aktuellen Prozess mit dem Endnutzer. Dazu gehören Arbeitsabläufe, Zuständigkeiten sowie die erzeugten und verwendeten Informationen. Der Prozessanalyst erstellt eine möglichst genaue Abbildung der aktuellen Situation als Ist-Prozess. Dafür muss er vor allem Interviewtechniken und Prozessmodellierung beherrschen.

Der Ist-Prozess ist die erste Arbeitsgrundlage für den Prozessdesigner. Er kennt die gängige Praxis anderer Unternehmen in der gleichen Domäne und er hat einen Überblick über die IT-Strategie, IT-Landschaft und technischen Schnittstellen des Kunden. Der Prozessdesigner erstellt den ersten Soll-Prozess als Teil der Spezifikation. Diese enthält eine Beschreibung des zukünftigen Arbeitsflusses, die beteiligten Endnutzer und Systeme, sowie eine grobe Beschreibung der Soll-Schritte im Prozess. Der Soll-Prozess ist, anders als vorher, Bestandteil einer gemeinsamen Spezifikation die von Prozessdesigner und Requirements Engineer verwaltet wird.

Der Soll-Prozess kapselt, welche Endnutzerrollen an den einzelnen Arbeitsschritten der Prozesse beteiligt sein werden. Der Requirements Engineer führt gezielt Interviews mit Endnutzern durch, um die Ausgestaltung einzelner Prozessschritte zu erarbeiten. Für jeden Prozessschritt werden Oberflächen, benötigte Funktionalität und nicht-funktionale Eigenschaften wie bspw. Antwortzeiten festgelegt. Die Interviews mit den Endnutzern führen zu Anpassungen der Spezifikation. Zum Beispiel werden vorher nicht betrachtete Daten essentiell für den neuen Prozessablauf. Für diese Informationen müssen Systemschnittstellen zur Verfügung stehen oder geschaffen werden. Der Requirements Engineer bestimmt die Auswirkungen von Anforderungen auf die Spezifikation und passt sie entsprechend an.

Parallel kommuniziert der Prozessdesigner mit dem IT-Architekten beim Kunden, um die Schnittstellen für benötigte Funktionalität zu finden oder auszuhandeln. Der IT-Architekt überwacht die System-Landschaft des Kunden. Er koordiniert verschiedene Projekte und sorgt für einheitliche Standards, z.b. bei der Granularität von Schnittstellen. Der Prozessdesigner kann durch die Abstimmung mit dem IT-Architekten genauer bestimmen, welche Schnittstellen zur Verfügung stehen, modifiziert oder neu geschaffen werden müssen. Diese Informationen werden genutzt, um den Soll-Prozess weiter zu verfeinern.

Der Prozessdesigner kann zu benutzende Schnittstellen für einzelne Prozessschritte hinterlegen, der Requirements Engineer kann die gebotene Funktionalität gegen die Anforderungen der Endnutzer prüfen und ggf. Erweiterungen als Teil der Spezifikation definieren. Die Informationen, die Prozessdesigner und Requirements Engineer in die gemeinsame Spezifikation einpflegen, erfassen die Anforderungen entlang der Soll-Prozesse. Um eine reibungslose Abstimmung zu gewährleisten, arbeiten beide mit einem gemeinsamen Glossar, einem fachlichen Datenmodell und einer einheitlichen Schnittstellenbeschreibungen, die ebenso wie die Soll-Prozesse Teil der Spezifikation sind.

Die Spezifikation repräsentiert das gemeinsam erarbeitete Wissen von Prozessdesigner und Requirements Engineer. Durch die Zusammenführung der vormalig getrennten Dokumente kann man den in Abschnitt 2 beschriebenen Probleme begegnen. So können bspw. fehlende Services für Soll-Prozesse genauso wie duplizierte Funktionalität durch Konsistenzprüfung erkannt werden. Das geteilte Wissen kann auch das Problemverständnis fördern und so überflüssige Interaktionen mit dem Benutzer vermeiden helfen.

Die Spezifikation ist die Grundlage für die Implementierung. Sowohl Prozessentwickler als auch Softwareentwickler bekommen ein Dokument mit einheitlichem Vokabular und einem gemeinsamen Lösungskonzept.

## **4 Unterstützende Techniken**

Um das Zusammenspiel zwischen Prozessdesigner und Requirements Engineer zu verbessern, sind unterstützende Techniken wünschenswert, die die Kluft zwischen den beiden Rollen überbrücken. Im Rahmen verschiedener Forschungsprojekte wurden bereits einige Techniken entwickelt, die sich hierfür nutzen lassen. Einige Möglichkeiten wollen wir in diesem Abschnitt kurz vorstellen.

#### 4.1 Use-Case-Visualisierung

Eine Technik, die zwei vielgenutzte Modelle in der Geschäftsprozess- und der Anforderungswelt zusammenbringt, ist die Use-Case-Visualisierung. Hierbei werden von den Vor- und Nachbedingungen von Use-Cases Geschäftsprozesse abgeleitet und in einer gängigen Notation wie z.B. EPK [Lüb06] oder BPMN [LSW08] dargestellt. Der generierte Prozess ist eine neue Sicht auf die Anforderung. Der Prozessdesigner kann feststellen ob die Softwareanforderungen dem vorgegebenen Prozess entsprechen oder wo Abweichungen vorhanden sind. Somit kann diese Visualisierungstechnik [KL08] genutzt werden um die Kommunikation zwischen Prozessdesignern und den Requirements Engineers zu verbessern. Umgekehrt können auch Use-Case Modelle aus Prozessmodellen generiert werden [DJ02]. Dieses Verfahren erlaubt es Inkonsistenzen zwischen den manuell dokumentierten und durch die Soll-Prozesse implizierten Anforderungen zu identifizieren.

#### 4.2 Gemeinsames Glossar

Eines der herausragenden Ziele des Requirements Engineerings ist es, ein einheitliches Verständnis von Projektzielen und Projektumfang bei allen Projektbeteiligten zu erzeugen. Einheitlicher Sprachgebrauch ist eine wichtige Grundlage, um dieses Ziel zu erreichen. Dabei gibt es zwei grundsätzliche Schwierigkeiten:

1. Haben zwei Personen eine andere Definition eines Begriffs, kann es zu Missverständnissen kommen. Wenn es nicht an einem konkreten Beispiel zur Sprache kommt, *wissen* zwei Personen nicht, dass der Gesprächspartner eine andere Definition zu Grunde legt [Fis00].
2. Aus dem selben Grund kann selbst ein gutes Glossar unzulänglich sein: Bei der Dokumentation von Anforderungen ist dem Requirements Engineer nicht unbedingt klar, dass ein von ihm verwendeter Begriff schon anders im Glossar definiert ist.

Im zweiten Fall kann mit recht einfachen Mitteln ein im Glossar definierter Begriff automatisch im Text hervorgehoben werden [KMS08]. Für die erste Schwierigkeit, zu definierende Begriffe überhaupt erst zu finden, haben wir gute Erfahrungen mit automatisch erzeugten Vorschlagslisten gemacht [KMS08]. Diese Begriffe werden vor allem basierend auf ihrer Häufigkeit im Anforderungsdokument vorgeschlagen. Zusätzlich kommt eine erfahrungsbasierte Komponente zum Einsatz, die Begriffe, die schon in anderen Projekten in das Glossar aufgenommen wurden, höher priorisiert.

Im Gegensatz zu anderen Ansätzen (vgl. [Kof05]) ist der Fokus unseres Ansatzes die Anregung zur Diskussion. Dazu ist direktes Feedback nötig: Der Requirements Engineer soll auf eine möglicherweise anderslautende Definition eines verwendeten Begriff hingewiesen werden, sobald er ihn schreibt. Die Vorschlagsliste soll regelmäßig zur Reflektion anregen, ob ein Begriff nicht doch missverständlich sein könnte. Diese Unterstützung kann Requirements Engineers und Prozessmodellierern die begriffliche Verzahnung erleichtern.

#### 4.3 Rollenzentrierte Geschäftsprozessausschnitte

Um Geschäftsprozesse mit Endnutzer abzustimmen, ist es sinnvoll, ihnen nur die für sie relevanten Teile zu zeigen. Daher werden aus großen Geschäftsprozessmodellen die Teile ausgeschnitten, die zu einer Rolle oder Organisationseinheit gehören.

Einen Ansatz, um EPKs entsprechend zu partitionieren, wurde von Gottschalk et al. vorgestellt [GRvdA05]: Dabei werden die EPK-Funktionen (d.h. modellierte Aktivitäten) aus einem Prozess selektiert, die Bezug zu einer Rolle haben. Die daraus resultierenden Teilprozesse werden mit Prozesswegweisern verbunden. Auf diese Art und Weise können Teilprozesse mit Benutzern besprochen und validiert werden, weil diese Technik eine für den jeweiligen Benutzer angepasste Sicht bereitstellt. Rollenzentrierte Prozesssichten schlagen die Brücke zwischen den rollenübergreifenden Prozessen und der Sicht der Endnutzer. Sie erleichtern somit nicht ausschließlich die Kommunikation mit dem Endnutzer, sondern auch jene zwischen Prozessdesignern und Requirements Engineers, zum Beispiel wenn es um die Einordnung von Nutzeranforderungen in den Prozesskontext geht.

#### **4.4 Modellübergreifendes Tracing**

Wenn Prozessdesigner und Requirements Engineers zusammenarbeiten, sollten nicht nur die Softwareanforderungen nachverfolgbar sein. Interessant ist vielmehr, welche Softwareanforderungen mit welchen Geschäftsprozessen und Prozessteilen zusammenhängen. Daher ist das Tracing, welches im Requirements Engineering bereits vielfach eingesetzt wird, auf die Geschäftsprozessmodellierung auszudehnen. Problematisch ist, dass das Tracing systemübergreifend arbeiten muss, weil verschiedenen Modelltypen in verschiedenen Systemen gespeichert werden.

Daher muss das Tracing so gestaltet werden, dass sich Ressourcen (Modelle, textuelle Anforderungen, ...) frei referenzieren lassen. So wie sich mittels URLs Webseiten beliebig untereinander verlinken lassen, kann man die Semantic-Web-Techniken dazu benutzen, verschiedene Ressourcen computerlesbar miteinander zu verknüpfen. Einen Ansatz für Tracing mittels dem Resource Description Framework (RDF) wurde von Zhu entwickelt und prototypisch umgesetzt [QZ07]. Somit ist es technisch möglich, verschiedenartige Modelle miteinander zu verknüpfen. Bei generierten Modellen (z.B. aus Use-Cases generierte Geschäftsprozesse) können diese Verknüpfungen automatisch erzeugt werden. Welche Verknüpfungen sonst sinnvollerweise einzufügen und noch praktikabel handbar sind, muss weiter untersucht werden.

## **5 Zusammenfassung und Ausblick**

Softwareprojekte mit dem Ziel Geschäftsprozesse zu unterstützen müssen sowohl die Geschäftsprozesse kennen, als auch softwarespezifische Anforderungen sammeln. Geschieht dies unabhängig voneinander, können durch Reibungsverluste viele Probleme, wie inkonsistente, nicht umsetzbare oder fehlende Anforderungen entstehen, die im weiteren Projektverlauf die Entwicklung verzögern und zusätzliche Kosten verursachen.

Wir schlagen daher vor, die beiden klassischen Bereiche Geschäftsprozessmodellierung und Requirements Engineering miteinander zu verzahnen und so Fehler bereits in der Spezifikationsphase zu verhindern. Durch klar definierte Rollen, integrierte Modelle und festgesetzte Feedbackzyklen kann die Qualität von Anforderungen gesteigert werden, so dass die nachfolgende Entwicklung schneller und effizienter durchgeführt werden kann.

Da beide Seiten ihre eigenen Methoden und Notationen haben, ist es nötig, technische

Hilfsmittel anzubieten, die Sichten auf die Spezifikation erzeugen, welche von beiden Seiten verstanden wird. Einige Techniken, die Grundlage für solche Werkzeuge sein könnten, haben wir hier vorgestellt. Diese Techniken müssen aber noch weiter verfeinert und auf diesen neuen Einsatzbereich zugeschnitten werden.

Als nächste Schritte streben wir eine genauere Rollendefinition in den von uns skizzierten Projekten an, die die Grundlage eines Entwicklungsmodells sein soll. Diese Methodik möchten wir in Industrieprojekten einsetzen und mit passenden Sichten, die durch Tools automatisch generiert werden, unterstützen.

## Literatur

- [DJ02] R.M. Dijkman und S.M.M. Joosten. An Algorithm to Derive Use Cases from Business Processes. In *Proceedings of the 6th IASTED International Conference on Software Engineering and Applications (SEA)*, Seiten 679–684, Boston, MA, USA, 2002.
- [dlVGD07] Jose Luis de la Vara González und Juan Sánchez Díaz. Business process-driven requirements engineering: a goal-based approach. In *Proceedings of the 8th Workshop on Business Process Modeling, Development, and Support (BPMDS)*, Trondheim, Norway, June 2007.
- [FHW07] Ralf Fahney, Andrea Herrmann und Rüdiger Weißbach. Bericht des AK Requirements Engineering und Projektmanagement über das Jahr 2007. *Softwaretechnik-Trends*, 28(1), 2007.
- [Fis00] Gerhard Fischer. Symmetry of ignorance, social creativity, and meta-design. *KBS Special Issues C&C99*, 13(78):527537, 2000.
- [GRvdA05] Florian Gottschalk, Michael Rosemann und Wil M.P. van der Aalst. My own process: Providing dedicated views on EPCs. In Markus Nüttgens und Frank J. Rump, Hrsg., *EPK 2005 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten*, Seiten 156–175, 2005.
- [KL08] Eric Knauss und Daniel Lübke. Using the Friction between Business Processes and Use Cases in SOA Requirements. In *Proceedings of REFS'08*, 2008.
- [KMS08] Eric Knauss, Sebastian Meyer und Kurt Schneider. Recommending Terms for Glossaries: A Computer-Based Approach. In *Proceedings of Workshop on Managing Requirements Knowledge at 16th IEEE RE Conference*, Barcelona, Spain, 2008.
- [Kof05] Leonid Kof. Text Analysis for Requirements Engineering. Dissertation, Technische Universität München, 2005.
- [LSW08] Daniel Lübke, Kurt Schneider und Matthias Weidlich. Visualizing Use Case Sets as BPMN Processes. In *Proceedings of REV Workshop, co-located at RE 2008, Barcelona, Spain*, 2008.
- [Lüb06] Daniel Lübke. Transformation of Use Cases to EPC Models. In *Proceedings of the EPK 2006 Workshop, Vienna, Austria*, 2006.
- [QZ07] Qiuyue Zhu. Tracing in SOA Projekten mit Semantic Web Technologien. Diplomarbeit, Gottfried Wilhelm Leibniz Universität Hannover, 2007.
- [Wes07] M. Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer-Verlag Berlin Heidelberg, 2007.