

Qualität von Geschäftsprozessmodellen aus dem Blickwinkel der Anforderungsanalyse

Ralf Laue¹

Abstract: Kommunikation von Anforderungen ist nur einer von zahlreichen Einsatzzwecken von Geschäftsprozessmodellen. Nicht selten stehen zur Anforderungserhebung daher Modelle zur Verfügung, die ursprünglich eine andere Aufgabe hatten (etwa die Prozessdokumentation oder die Unterstützung einer internen Schulung). Das hat zur Folge, dass solche Modelle unter Umständen wesentliche Informationen nicht enthalten, da selten vorkommende Abläufe, Fehlerzustände und Ausnahmen bewusst nicht modelliert wurden. Aufgabe bei der Anforderungsanalyse ist es, solche Modellierungslücken zu erkennen und das Stellen geeigneter Nachfragen zu unterstützen.

In diesem Beitrag wird die Idee eines Werkzeugs vorgestellt, das einige typische Indikatoren für fehlende Informationen im Modell automatisch findet und die erforderlichen Nachfragen formuliert. Dies kann Anforderungsanalytiker bei der Arbeit mit Geschäftsprozessmodellen unterstützen.

1 Einführung

Die Zwecke, zu denen Geschäftsprozessmodelle (GPM) erstellt werden, sind sehr vielfältig: Häufig dienen sie der Prozessdokumentation, der Vorbereitung von Prozessverbesserungen, der Prozessanalyse (z.B. durch Simulation) oder der Bestimmung der Prozessqualität [NK05].

Ebenso können Geschäftsprozessmodelle aber auch bei der Kommunikation von Anforderungen unterstützen oder sogar selber (als Konfiguration einer Workflow-Engine) Bestandteil einer Softwarelösung werden.

In der Literatur finden sich zahlreiche Untersuchungen zur Definition des Qualitätsbegriffs für Geschäftsprozessmodelle [MDN09, Ri10, Ne12, OBS12]. Wird dabei keine Unterscheidung nach dem Zweck eines GPM getroffen, ist eine Qualitätseinschätzung jedoch unbefriedigend, da die Qualität eines Modells immer nur mit Hinblick auf dessen Verwendungszweck beurteilt werden kann. In diesem Beitrag wollen wir uns mit der Nutzung von GPM zur Beschreibung von Anforderungen befassen.

Nicht selten erhält der für die Anforderungserhebung zuständige Analytiker zur Beschreibung eines Prozesses GPM, die zuvor für einen ganz anderen Zweck erstellt wurden. Selbst wenn die Qualität der Modelle für diesen ursprünglichen Zweck hoch war, heißt das noch nicht, dass dies auch für die Modellqualität zum Zwecke der Anforderungserhebung gilt.

So empfiehlt beispielsweise Miers [Mi06] zur Vereinfachung der Kommunikation, in früheren Phasen von Modellierungsprojekten nur die am häufigsten durchlaufenen Pfade ins

¹ Westsächsische Hochschule Zwickau, Fachgruppe Informatik, Dr.-Friedrichs-Ring 2a, 08056 Zwickau, ralf.laue@fh-zwickau.de

Modell aufzunehmen. Es entstehen GPM, die nur den sog. „Sonnenschein-Pfad“ zeigen und mögliche Fehlerzustände und Ausnahmen nicht abdecken. Um einen Überblick über einen Prozess zu bekommen, sind solche GPM durchaus geeignet. Sie verwirren den Leser nicht durch die Betrachtung seltener Fälle. Bei der Erhebung von Anforderungen ist es aber das Ziel, eine *vollständige* Beschreibung des Leistungsumfangs des zu schaffenden Systems zu bekommen.

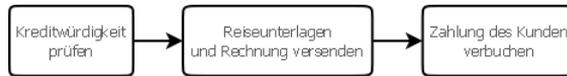


Abb. 1: Fragment aus einem Prozess „Reise verkaufen“

Während also zur Dokumentation eines Geschäftsprozesses für das Management das in Abb. 1 gezeigte Modellfragment möglicherweise völlig ausreichend ist, muss ein Anforderungsanalytiker gerade nach den Ausnahmefällen fragen: Wie soll sich das zu schaffende Softwaresystem verhalten, wenn der Kunde nicht kreditwürdig ist oder nicht zahlt? Die Antwort auf die letztgenannte Nachfrage könnte etwa zu einer bisher nicht gefundenen Menge von Anforderungen zum Forderungsmanagement führen.

Im Beitrag werden typische Situationen in einem GPM gezeigt, die Anforderungsanalytiker zu Nachfragen veranlassen sollten. Ein Werkzeug-Prototyp nutzt Methoden der automatischen Sprachverarbeitung, um diese Nachfragen automatisiert am Modell anzuzeigen.

2 Automatische Erkennung sprachlicher Probleme in Anforderungen

Die in diesem Beitrag vorgestellte Forschung ist inspiriert von Arbeiten, die potentielle Unklarheiten in natürlichsprachigen Anforderungsdokumenten untersuchen [BKK03, BK04]. Einige der dort erkannten Muster können auch bei Beschriftungen von GPM-Elementen zu Problemen führen.

Eine weitere Anregung waren Arbeiten, die Methoden der automatischen Sprachverarbeitung nutzen, um die Qualität natürlichsprachiger Anforderungen zu untersuchen [Fa02]. Entsprechende Ansätze werden bereits in verschiedenen akademischen sowie kommerziellen Werkzeugen verwendet [Ki08]. Sie lassen sich jedoch aus zwei Gründen nicht unmittelbar auf die Analyse von GPM übertragen. Erstens sind Modellelemente in GPM in der Regel nicht mit vollständigen Sätzen, sondern nur mit kurzen Wortgruppen beschriftet. Gängige Werkzeuge etwa zum Part-of-Speech-Tagging setzen aber das Vorliegen vollständiger Sätze voraus [SLG15].

Zum Zweiten müssen die Modellelemente, die den Kontrollfluss in GPM abbilden (also etwa Verzweigungen, Schleifen und Parallelität – in der Sprache BPMN durch Gateways dargestellt), ebenso als Bestandteil der zu analysierenden Sprache betrachtet werden wie die natürlichsprachigen Beschriftungen. Folglich müssen Indikatoren für Unvollständigkeit oder Missverständlichkeit in GPM sowohl den Kontrollfluss als auch die Beschriftungen der Elemente berücksichtigen.

Bisherige Untersuchungen zur Anwendung der automatischen Sprachverarbeitung in GPM zielten nicht auf das Finden potentieller Verständnisprobleme, sondern eher auf das Vorbeugen solcher Probleme durch die Durchsetzung von Namenskonventionen [LSM12, Le13] sowie Vereinheitlichung der in den Beschriftungen verwendeten Terminologie [Ha14, PW10].

Nur wenige Arbeiten beschäftigen sich mit der Analyse von Beschriftungen in GPM, um mögliche Modellierungsprobleme zu erkennen. In [GL11] zeigten wir, wie vermutliche Fehler in GPM aus der Analyse von Elementbeschriftungen erkannt werden können. Ein Beispiel für einen solchen Fehler ist, wenn im Modell die Ereignisse „Antrag angenommen“ und „Antrag abgelehnt“ gleichzeitig eintreten können. Anders als in dieser Arbeit soll es im vorliegenden Beitrag weniger um wirkliche Fehler im GPM als vielmehr um Unvollständigkeit und Unklarheit gehen. Diese müssen im Rahmen der Anforderungsermittlung erkannt und behoben werden.

Um solche Probleme in GPM zu identifizieren, wurde das in [GL11] beschriebene regelbasierte Werkzeug weiterentwickelt. Dieses zerlegt zunächst, ähnlich wie in [LKW07] vorgeschlagen, die Beschriftung einer Aufgabe in ein oder mehrere Aktivitätsverben und Zusatzinformationen, die aus Geschäftsobjekten und näheren Angaben zu Aktivitätsverben und Geschäftsobjekten bestehen. Wie aus dem Beispiel in Abb. 2 zu erkennen ist, werden dabei Stoppwörter (z.B. Artikel) ignoriert und verschiedene Flexionsformen des Verbs in die Grundform umgewandelt. Dies geschieht mittels einer Vollformen-Liste. Sie enthält die Flexionsformen von gegenwärtig etwa 1800 Verben, die typischerweise in GPM genutzt werden. Außerdem wird eine Liste von derzeit etwa 500 Funktionsverbgefügen berücksichtigt, so dass etwa der Wendung „Genehmigung erteilen“ das Verb „genehmigen“ zugeordnet wird.

Ergebnis der Analyse ist es, dass Fragen, die während der Anforderungsermittlung auftauchen, direkt im Modellierungswerkzeug angezeigt werden (siehe Abb. 3). Ebenso ist auch ein Durchsuchen von Modellen in Stapelverarbeitung möglich.

Der kommende Abschnitt stellt die Indikatoren für potentielle Unklarheiten und offene Fragen in GPM vor, die automatisch erkannt werden können.

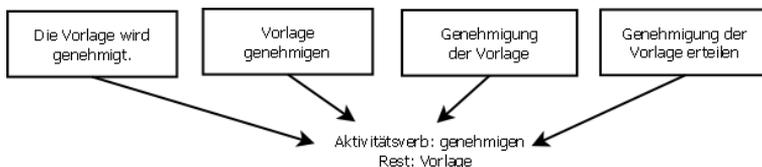


Abb. 2: Aufteilung einer Beschriftung in Aktivitätsverb und Zusatzinformationen

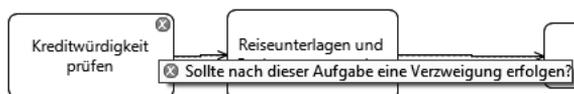


Abb. 3: Mögliches Problem wird im Modellierungswerkzeug angezeigt

3 Indikatoren für mögliche Qualitätsprobleme in Geschäftsprozessmodellen aus Sicht der Anforderungserhebung

3.1 Fehlende Aufgaben

Wenn in einem Modell wie in Abb. 1 nur der „Sonnenschein-Pfad“ modelliert ist, so ist nach einer Aufgabe, nach der üblicherweise eine Verzweigung folgen sollte, nur der „normale“ positive Ausgang modelliert. Möglich ist auch, dass in Erwartung eines positiven Ausganges ein Test, ob dieser auch tatsächlich eintritt, ganz fehlt.

Das Vorliegen des erstgenannten Falls erkennt man daran, dass eine Aufgabe ein Aktivitätsverb wie z.B. „prüfen“ enthält, dieser Aufgabe jedoch kein XOR-Split-Gateway folgt (oder weitere Prüfungen gefolgt von einem XOR-Split). Dies ist in Abb. 1 für die Aufgabe „Kreditwürdigkeit prüfen“ der Fall. Im Kasten in Abb. 4 sind Aktivitätsverben zusammengestellt, nach denen in der Regel eine Verzweigung folgen sollte. Fehlt eine solche, ist zu hinterfragen, ob im Modell nur der positive Ausgang einer Entscheidung modelliert ist und wie in anderen möglichen Fällen zu verfahren ist.

Dass der Test auf mögliche Ausnahmen im Modell ganz fehlt, kann erkannt werden, wenn im Modell Zustände (bzw. Ereignisse) durch Wörter wie „erfolgreich“ modelliert sind, jedoch eine davorliegende Verzweigung zu einem negativen Gegenstück (hier: „erfolglos“, „ohne Erfolg“, „fehlgeschlagen“, etc.) fehlt. In Abb. 5 ist eine Auswahl der Wörter und Wortgruppen zusammengestellt, bei denen vermutet werden kann, dass auch ein möglicherweise negativer Ausgang im Modell berücksichtigt werden sollte. Besonders gut gelingt diese Analyse in Modellen in der Modellierungssprache der ereignisgesteuerten Prozessketten (EPK), da diese - anders als die Sprache BPMN - die Modellierung von Ereignissen zwingend vorschreibt.

entscheiden (Eine Entscheidung kann mehrere Ausgänge haben.), ebenso Verbindungen wie *Entscheidung treffen*

bewerten (Das Ergebnis der Bewertung sollte den weiteren Prozessablauf bestimmen.), ebenso: *abwägen, begutachten, beurteilen, einschätzen, reviewen, rezensieren, vidieren*, dazu weitere Verbindungen wie *Gutachten abgeben*

prüfen (Eine Prüfung hat immer mehr als einen möglichen Ausgang.), ebenso: *abchecken, abprüfen, checken, durchchecken, durchprüfen, durchtesten, evaluieren, kontrollieren, nachprüfen, testen, überprüfen, validieren, vergewissern, verifizieren*, dazu weitere Verbindungen wie *einer Prüfung unterziehen*

suchen (Möglicherweise wird das Gewünschte nicht gefunden.), ebenso: *heraussuchen, lokalisieren*

vergleichen (Das Ergebnis des Vergleichs sollte den weiteren Prozessablauf bestimmen.), ebenso: *abgleichen*
versuchen (Der Versuch kann glücken oder nicht.), ebenso *ausprobieren, erproben, proben, probieren*

Abb. 4: Aktivitäten, denen in der Regel eine Verzweigung folgen sollte.

Es gibt außerdem noch einen weiteren Fall, in dem automatisiert erkannt werden kann, dass Aufgaben im Modell fehlen: Wenn Aufgaben einander folgen, zwischen denen vernünftigerweise mindestens eine weitere Aufgabe liegen sollte. Typische Beispiele sind:

- *Daten abrufen* → *Daten löschen*
- *Dokument empfangen* → *Dokument weiterleiten* Dazwischen sollte eine Entscheidung liegen, ob und an wen das Dokument weiterzuleiten ist, oft verbunden mit einer Prüfungs- und Genehmigungsaktivität

erfolgreich, *bestanden, gültig, in Ordnung, korrekt, ok, mit Erfolg*
vollständig, *ausreichend, ohne Ausnahme*
fehlerfrei, *ohne Fehler, korrekt*
akzeptieren, *anerkennen, approbieren, befürworten, bewilligen, bejahen, billigen, einwilligen, erlauben, gestatten, genehmigen, gewähren, positiv entscheiden, stattgeben, zubilligen, zuerkennen, zugestehen, zulassen, zusprechen, zustimmen, ebenso Funktionsverbgefüge wie Einwilligung aussprechen*
einverstanden
rechtzeitig, *fristgemäß, fristgerecht, form- und fristgerecht, innerhalb der gesetzten Frist*
vorhanden, *verfügbar, auf Lager*
annehmbar, durchführbar, machbar, gefunden, passend

Abb. 5: Wörter und Wortverbindungen, die in der Regel das Vorliegen des „guten Falls“ beschreiben

- *Antrag empfangen* → *Antrag unterschreiben* Auch hier fehlt die Information, unter welchen Voraussetzungen die Unterschrift zu leisten ist.

3.2 Fehlende Informationen zu einer Entscheidung

In [BKK03, BK04] wird darauf hingewiesen, wie problematisch die Verwendung logischer Konjunktionen wie „und“, „oder“, „beziehungsweise“, etc. in Anforderungsdokumenten sein kann. Dies gilt auch für GPM: Auch wenn die gängigen Geschäftsprozessmodellierungssprachen eigene Symbole zur Modellierung von Verzweigungen und paralleler Ausführung mitbringen, finden sich doch auch in den Beschriftungen von Aufgaben in GPM solche Konjunktionen.

Ein *oder* oder *bzw.* zwischen zwei Aktivitätsverben deutet darauf hin, dass im Modell die Entscheidung, welche der beiden Aktionen durchzuführen ist, nicht modelliert ist und somit bei der Anforderungserhebung erfragt werden sollte. Ein Beispiel für eine solche Beschriftung wäre „Ware bestellen oder produzieren“².

Ähnliches gilt für die in Abb. 6 aufgeführten Wörter und Wortverbindungen, die darauf hindeuten, dass eine Aufgabe nur optional auszuführen ist. Auch hier ist zu hinterfragen, auf welche Weise die Entscheidung über die Ausführung zu fällen ist.

falls/wenn erforderlich/nötig/notwendig
optional, eventuell, gegebenenfalls, in der Regel

Abb. 6: Wörter und Wortverbindungen, auf Optionalität hindeuten

Auch wenn Entscheidung und mögliche Ausgänge modelliert sind, kann es vorkommen, dass mögliche Ausgänge im Modell vergessen wurden. Ein typischer Fall wäre ein GPM mit einer Aufgabe „Erfragen des Alters“, gefolgt von zwei möglichen Bedingungen „Anmelder unter 27 Jahre alt“ und „Anmelder über 27 Jahre alt“. Hier ist die Frage zu stellen, wie für den Randwert =27 zu verfahren ist. In [Pa05] ist beschrieben, wie eine Menge von solchen als Wertevergleich formulierten Bedingungen auf Vollständigkeit geprüft werden kann. Da Bedingungen in GPM auch in natürlicher Sprache beschriftet sein können, müssen neben den Vergleichsoperatoren auch Wortgruppen wie *ist größer als*, *beträgt*

² gefunden in einem Modell des BPM Academic Initiative-Repositories (bpmi.org)

mehr als, überschreitet, übertrifft oder *übersteigt* (und andere Flexionsformen dieser Verben) berücksichtigt werden.

3.3 Fehlende Informationen zu einer Aufgabe

Findet sich zwischen zwei Aktivitätsverben ein *und* oder *sowie*, (*A und B*, also z.B. *Handbuch korrekturlesen und übersetzen*), so ist zu hinterfragen, ob implizit eine zeitliche Reihenfolge ausgedrückt werden soll, d.h. A immer vor B stattfinden muss. Ebenso ist die Möglichkeit zu prüfen, ob A und B parallel durchgeführt werden sollen.

Schließlich finden sich in GPM leider häufig Aktivitätsverben, die so vage sind, dass beim Lesen des Modells unklar bleibt, welche Aufgabe denn nun konkret durchzuführen ist. So sagt eine Aufgabenbeschriftung „Arbeitsvorbereitung unterstützen“ nichts Konkretes über die erwarteten Tätigkeiten aus. Eine Zusammenstellung solcher vagen Verben, die sämtlich Aufgabenbeschreibungen in GPM entnommen wurden, findet sich in Abb. 7.

abfertigen, absichern, administrieren, anvisieren, assistieren, aufarbeiten, ausgestalten, beachten, bearbeiten, befassen, befolgen, beitragen, berücksichtigen, beteiligen, bewirtschaften, bewirken, durchsetzen, einbinden, einbeziehen, ermöglichen, gewährleisten, helfen, herbeiführen, koordinieren, kooperieren, managen, mitarbeiten, mitwirken, neuorganisieren, nachgehen, organisieren, reagieren, reorganisieren, sicherstellen, unterstützen, verarbeiten, verhindern, vermeiden, verwalten, verwerten, weiterverarbeiten, zuarbeiten, zusammenarbeiten

Abb. 7: Verben, die zu wenig über die durchzuführende Aufgabe aussagen

Ebenso gibt es Modelle, in denen nicht die Verben, sondern die Geschäftsobjekte unklar beschriftet sind. Beispiele hierfür sind Wörter wie *Daten, Dokument, Maßnahmen, Unterlagen* oder *Informationen*, aber auch Kombinationen wie *zuständige Stellen* oder *erforderliche Aktivitäten*, sofern nicht an anderer Stelle eindeutig dokumentiert ist, was unter den Begriffen in diesem Kontext zu verstehen ist. Auch bei solchen Begriffen muss bei der Anforderungserhebung nachgefragt werden, was jeweils genau gemeint ist.

3.4 Fehlende Informationen zu einem Datenobjekt

Selbst dann, wenn in einem GPM „nur“ der Kontroll- und nicht der Datenfluss modelliert ist, liefern Aktivitätsverben wie beispielsweise „abrufen“, „eintreffen“, „senden“ oder „weiterleiten“ Informationen über Empfang und Weitergabe von Daten und Dokumenten. Trčka et al. katalogisierten in [TvS09] Indikatoren für Datenflussprobleme in GPM, die automatisiert gefunden werden können und Anlass für gezielte Nachfragen bei der Anforderungsermittlung bieten. Ein wichtiger Fall ist, wenn in einer Aufgabe ein Dokument erzeugt wird, das von keiner anderen Aufgabe im Modell benötigt wird. Eine Prüfung für den umgekehrten Fall - ein Dokument wird benutzt, ohne dass klar ist, wann es erzeugt oder empfangen wurde, wird in [HF12] beschrieben.

Auch hier werden wir es in der Regel nicht mit einem wirklich falschen Modell zu tun haben, sondern mit einem, in das einzelne Informationen bewusst oder unbewusst nicht

aufgenommen wurden. Bei der Ermittlung von Anforderungen müssen die fehlenden Informationen über Herkunft bzw. Zweck von Dokumenten bzw. Informationen erfragt werden.

4 Zusammenfassung und Ausblick

Bei der Anforderungserhebung werden andere Qualitätsanforderungen an GPM gestellt werden als bei anderen Einsatzzwecken. Insbesondere sollen die modellierten Informationen über mögliche Ausführungspfade möglichst vollständig sein und nicht nur die häufigsten Fälle abdecken. In diesem Beitrag wurden Ideen für eine automatische Untersuchung von GPM vorgestellt. Diese Untersuchung soll fehlende Informationen in GPM aufspüren und damit Anforderungsanalytikern helfen, zusätzliche Informationen einzufordern. Es ist zu betonen, dass ein solches Werkzeug zwar eine Hilfe bei der Suche nach offenen Fragen zu einem GPM ist, jedoch keinesfalls eine gründliche manuelle Inspektion der Modelle ersetzen kann.

Bei der Entwicklung eines Werkzeug-Prototypen zeigte sich, dass die Vielfalt an Beschriftungsstilen (vgl. Abb. 2) die Aufgabe wesentlich erschwert. In einschlägigen Veröffentlichungen zur Analyse von GPM-Beschriftungen werden nicht selten vereinfachende Annahmen gemacht, etwa dass die Beschriftungen aus genau einem Verb im Infinitiv und genau einem Objekt bestehen. Eine solche Annahme ist zwar bequem und erlaubt eine schnelle Werkzeug-Erstellung. Das Werkzeug ist dann aber nur unter den getroffenen Annahmen einsetzbar (und sonst wenig nützlich).

Unter der Annahme, dass ein Anforderungsanalytiker schon existierende GPM vorgelegt bekommt, kann keine bestimmte Namenskonvention für die Modellbeschriftung vorausgesetzt werden. Wie vielfältig dabei die Regeln zur Bestimmung eines Aktivitätsverbs sein müssen, zeigen die folgenden Beispielbeschriftungen von Aufgaben in einem Geschäftsprozessmodell:

- Student bereitet sich auf Prüfung vor → Als Aktivitätsverb sollte „vorbereiten“ identifiziert werden.
- Prüfung durchführen → Als Aktivitätsverb sollte „prüfen“ identifiziert werden.
- Werkstück einer Prüfung unterziehen → Als Aktivitätsverb sollte „prüfen“ identifiziert werden.

An den Beispielen ist zu erkennen, dass für eine sinnvolle Analyse eine eingehende Beschäftigung mit der sprachlichen Vielfalt von tatsächlich in der Praxis vorkommenden Modellbeschriftungen unerlässlich (wenn auch freilich aufwendig) ist. Eine unkritische Anwendung von Standard-Werkzeugen der Computerlinguistik ohne weitere Anpassung führt nicht zum gewünschten Ziel. In den beiden letztgenannten Beispielen würden dann etwa die Aktivitätsverben „durchführen“ und „unterziehen“ ermittelt werden, was nicht wie in Abb. 3 die Folgerung erlaubt, dass nach einer entsprechend beschrifteten Aufgabe im Modell eine Verzweigung zu erwarten ist.

Der in diesem Beitrag diskutierte Werkzeug-Prototyp zur Identifikation offener Fragen zu einem GPM aus Sicht des Anforderungsanalytikers erkennt alle in Abschnitt 3 genannten Indikatoren für potentielle Unklarheiten im Modell. Darüber hinaus werden noch einige weitere gefunden, etwa fehlende Informationen über Zuständigkeiten und unterstützende EDV-Systeme. Es ist beabsichtigt, das Werkzeug weiter auszubauen und als Add-On für das freie Modellierungswerkzeug *bflow* toolbox* (www.bflow.org) bereitzustellen.

Literaturverzeichnis

- [BK04] Berry, Daniel M.; Kamsties, Erik: Ambiguity in Requirements Specification. In: Perspectives on Software Requirements, Jgg. 753 in The Springer International Series in Engineering and Computer Science, S. 7–44. Springer, 2004.
- [BKK03] Berry, Daniel M.; Kamsties, Erik; Krieger, Michael M.: , From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity, A Handbook, 2003.
- [Fa02] Fantechi, Alessandro; Gnesi, Stefania; Lami, Giuseppe; Maccari, Alessandro: Application of Linguistic Techniques for Use Case Analysis. In: Proceedings of the 10th Anniversary IEEE Joint International Conference on Requirements Engineering. S. 157–164, 2002.
- [GL11] Gruhn, Volker; Laue, Ralf: Detecting Common Errors in Event-Driven Process Chains by Label Analysis. Enterprise Modelling and Information Systems Architectures, 6(1):3–15, 2011.
- [Ha14] Havel, Jean-Marie; Steinhorst, Matthias; Dietrich, Hanns-Alexander; Delfmann, Patrick: Supporting Terminological Standardization in Conceptual Models - a Plugin for a Meta-Modelling Tool. In: 22st European Conference on Information Systems. 2014.
- [HF12] Humm, Bernhard G.; Fengel, Janina: Assessing Semantic Consistency of Business Process Models. International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering, 6(6):94 – 99, 2012.
- [Ki08] Kiyavitskaya, Nadzeya; Zeni, Nicola; Mich, Luisa; Berry, Daniel M.: Requirements for tools for ambiguity fication and measurement in natural language requirements specifications. Requir. Eng., 13(3):207–239, 2008.
- [Le13] Leopold, Henrik; Eid-Sabbagh, Rami-Habib; Mendling, Jan; Azevedo, Leonardo Guerreiro; Baião, Fernanda Araujo: Detection of naming convention violations in process models for different languages. Decision Support Systems, 56(0):310 – 325, 2013.
- [LKW07] Lincoln, Maya; Karni, Reuven; Wasser, Avi: A Framework for Ontological Standardization of Business Process Content. In: International Conference on Enterprise Information Systems 2007. S. 257–263, 2007.
- [LSM12] Leopold, Henrik; Smirnov, Sergey; Mendling, Jan: On the refactoring of activity labels in business process models. Inf. Syst., 37(5):443–459, 2012.
- [MDN09] Mohagheghi, Parastoo; Dehlen, Vegard; Neple, Tor: Definitions and approaches to model quality in model-based software development - A review of literature. Information & Software Technology, 51(12), 2009.
- [Mi06] Miers, Derek: Best Practice BPM. ACM Queue, 4(2):40–48, 2006.

- [Ne12] Nelson, H.James; Poels, Geert; Genero, Marcela; Piattini, Mario: A conceptual modeling quality framework. *Software Quality Journal*, 20(1):201–228, 2012.
- [NK05] Nysetvold, Anna Gunhild; Krogstie, John: Assessing Business Process Modeling Languages Using a Generic Quality Framework. In: *Proceedings of the CAiSE'05 Workshops*. S. 545–556, 2005.
- [OBS12] Overhage, Sven; Birkmeier, Dominik Q.; Schlauderer, Sebastian: Quality Marks, Metrics, and Measurement Procedures for Business Process Models. *Business & Information Systems Engineering*, 4(5):229–246, 2012.
- [Pa05] Pap, Zsigmond; Majzik, Istvn; Pataricza, Andrs; Szegi, Andrs: Methods of checking general safety criteria in UML statechart specifications. *Reliability Engineering & System Safety*, 87(1):89 – 107, 2005.
- [PW10] Peters, Nicolas; Weidlich, Matthias: Using Glossaries to Enhance the Label Quality in Business Process Models. In: *Dienstleistungsmodellierung 2010*. 2010.
- [Ri10] Rittgen, Peter: Quality and perceived usefulness of process models. In: *Proceedings of the 2010 ACM Symposium on Applied Computing*. ACM, New York, NY, USA, S. 65–72, 2010.
- [SLG15] Storch, Arian; Laue, Ralf; Gruhn, Volker: Flexible Evaluation of Textual Labels in Conceptual Models. In: *Enterprise modelling and information systems architectures - EMISA 2015*. 2015.
- [TvS09] Trčka, Nikola; van der Aalst, Wil M. P.; Sidorova, Natalia: Data-Flow Anti-patterns: Discovering Data-Flow Errors in Workflows. In: *Conference on Advanced Information Systems Engineering 2009*. Jgg. 5565 in *Lecture Notes in Computer Science*. Springer, S. 425–439, 2009.