

Berechnung von Modelldifferenzen als Basis für die Evolution von Prozessmodellen

Stanley Hillner, Heiko Kern, Stefan Kühne
Universität Leipzig, Institut für Informatik, Betriebliche Informationssysteme
Johannissgasse 26, 04103 Leipzig
{hillner, kern, kuehne}@informatik.uni-leipzig.de

Abstract: In diesem Beitrag wird die Berechnung von Differenzen zwischen Prozessmodellen betrachtet. Hierzu werden verschiedene Ansätze und Werkzeuge zur Berechnung von Differenzen beschrieben. Ausgehend von einem konkreten Anwendungsfall – einer EPK-zu-BPEL-Transformation – werden verschiedene Testkriterien aufgestellt, die anschließend zur Evaluierung von zwei Differenzbildungswerkzeugen dienen. Abschließend werden die Ergebnisse der Untersuchung entsprechend dargestellt.

1 Einleitung

Einen Schwerpunkt der modellgetriebenen serviceorientierten Integration bildet die (teil-) automatisierte Abbildung von fachlichen Geschäftsprozessmodellen auf technische Orchestrierungsmodelle [TK08]. Mögliche Anforderungen für eine solche Abbildung wurden in Kühne u. a. [KSI08] untersucht und darauf aufbauend ein Rahmenwerk zur Transformation von Geschäftsprozessmodellen definiert. Eine Evaluierung dieses Rahmenwerks wurde am Beispiel der Transformation von Ereignisgesteuerten Prozessketten (EPK) [KNS92] in die Business Process Execution Language for Web Services (BPEL) durchgeführt, wobei u. a. auf das Problem von Modelländerungen in den unterschiedlichen Abstraktionsebenen hingewiesen wurde [SKD⁺08]. Die Änderung der Geschäftsprozessmodelle wird meistens durch Weiterentwicklung der Geschäftsprozesse und die Änderung der Orchestrierungsmodelle wird durch das Hinzufügen von technischen Verfeinerungen bzw. Informationen hervorgerufen. Damit eine erneute Transformation der weiterentwickelten Geschäftsprozessmodelle möglich ist, ohne die Änderungen in den Orchestrierungsmodellen zu verlieren, wird in diesem Artikel eine Lösung vorgeschlagen, die auf der Differenzberechnung zwischen Prozessmodellen basiert. Hierzu werden Ansätze und Werkzeuge zur Berechnung von Modelldifferenzen beschrieben (vgl. Abschnitt 2), an einem konkreten Beispiel – einer EPK-zu-BPEL-Transformation – angewendet und anhand bestimmter Testkriterien evaluiert (vgl. Abschnitt 3).

2 Modelldifferenzen

2.1 Berechnung von Modelldifferenzen

In diesem Abschnitt werden zwei Möglichkeiten zur Berechnung von Modelldifferenzen aufgezeigt. Dabei bilden jeweils die beiden Modelle M_{orig} (Originalmodell) und M_{mod} (modifiziertes Modell) die Grundlage der nachfolgenden Betrachtungen.

Eine Möglichkeit zur Differenzberechnung ist die Aufzeichnung sämtlicher Änderungen an M_{orig} . Das Ergebnis dieses Ansatzes ist ein Protokoll, welches alle Schritte aufzeigt, die nötig sind, um M_{mod} aus M_{orig} zu erzeugen. Dieser Ansatz zeichnet sich vor allem dadurch aus, dass die Änderungen zu 100% erfasst werden. Negativ ist anzumerken, dass sich dieses Vorgehen der Differenzermittlung bspw. schwer für verteilte Modellierung umsetzen lässt.

Ein anderer verbreiteter Ansatz zur Ermittlung von Differenzen besteht darin, einen Vergleich zwischen beiden Modellen vorzunehmen und aufbauend auf den ermittelten Vergleichswerten eine Modelldifferenz zu berechnen. Für diesen Ansatz wurden im Wesentlichen drei Phasen identifiziert, welche sich in der Praxis häufig nur in kleineren Details unterscheiden. Diese Phasen werden nachfolgend kurz betrachtet.

Matching ist die Berechnung von Ähnlichkeiten zwischen Elementen aus M_{orig} und M_{mod} . Hier existieren verschiedene Vorgehensweisen, welche bspw. anhand von UUIDs (Universally Unique Identifier) die Gleichheit von Modellelementen feststellen, oder die Ähnlichkeit anhand anderer Vergleichskriterien wie dem Namen, dem Typ des Modellelements oder den referenzierten Modellelemente berechnen. Bei letzterem Vorgehen ist es durchaus üblich, den Vergleich der Modelle durch gewisse Bedingungen einzuschränken, welche die Komplexität des Vergleichs stark reduzieren können. Die durch den Vergleich ermittelten Matching-Werte stellen Ähnlichkeitsmetriken der Elemente aus M_{orig} und M_{mod} dar.

Mapping bezeichnet die Ermittlung von korrespondierenden Modellelementen. Elemente korrespondieren, falls sie dasselbe konzeptionelle Artefakt beschreiben. Ziel des Mappings ist es, ausgehend vom zuvor berechneten Matching der Modelle eine Abbildung zwischen M_{orig} und M_{mod} zu ermitteln. Im Falle der zuvor angesprochenen Verwendung von UUIDs ergeben sich beim Matching der Modelle lediglich Übereinstimmungen von 0% oder 100%, wodurch das Mapping der Elemente offensichtlich ist. In allen anderen Fällen resultieren Vergleichswerte zwischen 0% und 100%, aus welchen in dieser Phase die optimale Kombination der Elemente berechnet werden muss.

Diff-Bildung ist die Berechnung und Darstellung der Differenzen beider Modelle. In dieser letzten Phase ist es erforderlich, einen erneuten Vergleich der im Mapping ermittelten, korrespondierenden Modellelemente vorzunehmen. Dadurch werden Unterschiede der Elementeigenschaften wie Attribute oder Referenzen erkannt. Die Unterschiede auf Modell- und Elementebene werden anschließend zu einem Diffe-

renzmodell zusammengefasst, welches sämtliche Unterschiede zwischen M_{orig} und M_{mod} enthält.

2.2 Werkzeugunterstützung

Für die Berechnung von Modelldifferenzen steht eine Vielzahl an Werkzeugen zur Verfügung [ADMR05, RB01, Sch07], wobei in diesem Beitrag eine eingeschränkte Auswahl untersucht wurde. Die Betrachtungen beschränken sich in diesem Artikel auf Werkzeuge, welche Modelle des Eclipse Modeling Framework (EMF) verarbeiten können, da die Modelle, welche dem Anwendungsfall zugrunde liegen, im EMF-Format vorliegen. Nachfolgend werden drei Werkzeuge vorgestellt.

EMF Compare¹ ist Teil des Eclipse Modeling Framework Technology (EMFT)²-Projekts. Es berechnet die Differenz zwischen beliebigen EMF-Modellen, indem es einen Vergleichsalgorithmus verwendet, welcher die Elemente beider Modelle anhand spezieller Kriterien gegenüberstellt. Die verwendeten Vergleichskriterien sind die Metatypen, Namen, Attribute und Referenzen sowie die Position der Elemente im Modellbaum.

Der Vergleich von Attributen wird bspw. auf Basis von String-Werten unter Zuhilfenahme der Editierdistanz von Levenstein berechnet. Die Differenzen werden in einem EMF-Modell serialisiert und können somit in EMF weiter verarbeitet werden.

Der ModelMatcher [Hil08] ist ein Projekt der Universität Leipzig und setzt ebenfalls den Vergleichsansatz auf Basis verschiedener Vergleichskriterien um. Dieses Werkzeug verwendet ähnliche Eigenschaften der Modellelemente für den Vergleich, wie das zuvor genannte EMF Compare. Die Elemente werden hinsichtlich ihrer Metatypen, ihren Containment-Beziehungen sowie sämtlicher Attribute und Referenzen gegenüber gestellt. Im Gegensatz zu EMF Compare werden Ähnlichkeiten von Attributen mit Hilfe ihrer nativen Vergleichsoperationen ermittelt. Des Weiteren wird die Position der Elemente im Modellbaum i. d. R. nicht vorgenommen. Diese dient lediglich während des Mappings als Hilfestellung bei der Verarbeitung von grenzwertigen Ergebnissen. Auch hier wird die ermittelte Differenz beider Modelle als EMF-Modell serialisiert.

Das letzte hier genannte Werkzeug ist der bereits im EMF enthaltene Change Recorder. Der Change Recorder stellt eine Realisierung des Protokollierungsansatzes dar und ist somit potentiell fehlerunanfälliger als die beiden zuvor genannten Werkzeuge. Er erstellt eine Beschreibung der vorgenommenen Änderungen am Modell und stellt diese als eigenständiges EMF-Modell zur Verfügung.

¹<http://www.eclipse.org/modeling/emft/?project=compare>

²<http://www.eclipse.org/modeling/emft>

3 Anwendungsfall: EPK-zu-BPEL-Transformation

3.1 Beschreibung des Anwendungsfalls

Die zuvor vorgestellten Werkzeuge sollen anhand eines Anwendungsfalls, der EPK-zu-BPEL-Transformation, hinsichtlich bestimmter Testkriterien untersucht werden. Der betrachtete Anwendungsfall in Abbildung 1 zeigt zwei unterschiedliche Verwendungsmöglichkeiten von Modelldifferenzen bei der Prozessevolution auf.

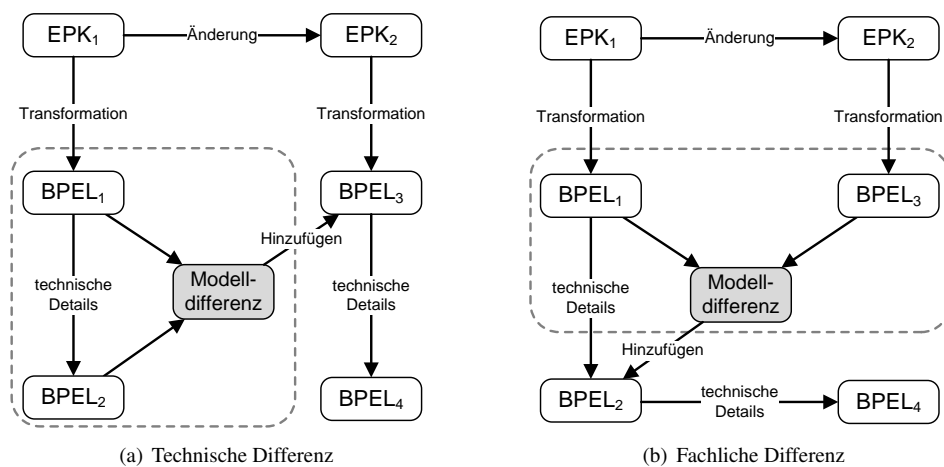


Abbildung 1: Möglichkeiten der Differenzbildung

In der Regel werden Geschäftsprozesse auf einer rein fachlichen Ebene modelliert. Hier bieten sich bspw. EPKs an, da diese für den Fachanwender leicht zu erlernen sind und die abgebildeten Prozesse strukturiert und übersichtlich darstellen. Aus dem initial erstellten, fachlichen Modell (EPK₁) wird im Verlauf der Entwicklung des Systems mittels Modell-zu-Modell-Transformation (M2M-Transformation) ein technisches Orchestrierungsmodell (BPEL₁) erzeugt, welches anschließend manuell um weitere technische Details angereichert wird, wodurch ein BPEL₂-Modell entsteht. Änderungen am System werden, sofern sie nicht technischer Natur sind, auf fachlicher Ebene, also an EPK₁ vorgenommen. Das so entstandene Modell EPK₂ wird erneut in ein technisches Modell (BPEL₃) transformiert, welches jedoch die zuvor hinzugefügten technischen Informationen aus dem BPEL₂-Modell nicht enthält. An diesem Punkt gibt es zwei Möglichkeiten, durch Einsatz von Modelldifferenzen ein vollständiges Modell BPEL₄ zu erzeugen.

Die erste Möglichkeit (siehe Abbildung 1(a)) besteht darin, die Differenz zwischen BPEL₁ und BPEL₂ zu berechnen, was der zuvor durchgeführten technischen Anpassung entspricht. Diese Differenz kann anschließend zum BPEL₃-Modell hinzugefügt werden, wodurch das gewünschte BPEL₄-Modell entsteht. Hierbei müssen evtl. weitere technische Details hinzugefügt werden.

Die zweite Möglichkeit (siehe Abbildung 1(b)) ist durch die Berechnung der Differenz zwischen den Modellen $BPEL_1$ und $BPEL_3$ gegeben, was der Änderung auf fachlicher Ebene entspricht. Diese Differenz wird anschließend zum Modell $BPEL_2$ hinzugefügt und um weitere technische Details erweitert, wodurch ebenfalls das gewünschte $BPEL_4$ -Modell entsteht.

3.2 Werkzeugtests

Aufbauend auf dem zuvor beschriebenen Anwendungsfall sollen die beiden Werkzeuge EMF Compare und ModelMatcher (siehe Kapitel 2.2) für die Berechnung der Modell-differenzen zwischen den BPEL-Modellen getestet werden. Der Change Recorder wird hierbei nicht in die Untersuchung einbezogen, da dieser durch den Protokollierungsansatz immer eine korrekte Differenz ermittelt. Deshalb würde eine Gegenüberstellung mit Werkzeugen, denen der Vergleichsansatz zu Grunde liegt, keinen Sinn machen. Des Weiteren ist eine Protokollierung der fachlichen Änderungen zwischen den technischen BPEL-Modellen durch die vorgenommene M2M-Transformation nicht durchführbar.

Der Werkzeugtest untersucht, wie unterschiedliche Arten von Modelländerungen erkannt und dargestellt werden. Dies soll zum einen die Verwendung der angesprochenen Vergleichskriterien evaluieren und zum anderen eventuelle, daraus resultierende Stärken und Schwächen einzelner Ansätze aufzeigen. Nachfolgend werden drei Kategorien von Änderungsoperationen abgegrenzt.

Kategorie 1: Diese erste Kategorie umfasst Änderungen auf Modellebene wie das Hinzufügen oder Löschen von einzelnen Modellelementen. Bezogen auf den Anwendungsfall treten derartige Änderungen auf, wenn bspw. ein neuer Web-Service in einen Geschäftsprozess eingegliedert wird, oder aber ein nicht mehr benötigter Service entfernt wird.

Kategorie 2: In dieser Kategorie werden Veränderungen auf Elementebene betrachtet. Es werden Elementeigenschaften wie Attribute oder auch Referenzen auf andere Elemente modifiziert. Derartige Änderungen treten bspw. bei der Anpassung von Services eines Prozesses auf. Hier kann es vorkommen, dass die Adresse oder auch der Name eines verwendeten Dienstes modifiziert werden muss. Diese Änderungen erzeugen beim Matching der Modelle niedrige Vergleichswerte, aus denen beim späteren Mapping potenzielle Schwierigkeiten resultieren.

Kategorie 3: Abschließend werden erneut Änderungen auf Modellebene betrachtet. Hier sollen die Auswirkungen von Elementverschiebungen ermittelt werden. Elemente werden als verschoben betrachtet, wenn sie ihre Position im EMF-Modellbaum ändern. In diesem Kontext sind zwei Arten von Verschiebung zu betrachten: Elementverschiebungen, die die Semantik des Modells nicht beeinflussen und Elementverschiebungen, die die Semantik des Modells ändern.

3.3 Auswertung der Tests

Nachfolgend werden die von den Werkzeugen berechneten Differenzen betrachtet und Rückschlüsse auf die Verwendung bestimmter Vergleichskriterien gezogen sowie Problemfelder des Matchings und Mappings aufgezeigt.

Kategorie 1: Während der Tests dieser Kategorie wurde beobachtet, dass die hinzugefügten und gelöschten Modellelemente von beiden Werkzeugen in jedem Fall korrekt identifiziert wurden. Weiterhin wurde festgestellt, dass für die Ermittlung von hinzugefügten bzw. entfernten Elementen nur wenige Vergleichskriterien nötig waren, sofern die übrigen Modellelemente keine gravierenden Veränderungen aufwiesen. Ein Problemfeld, welches sich während der Tests offenbarte, ist die Darstellung der berechneten Differenz. Anforderungen an das Differenzmodell in dieser Kategorie sind zum einen, dass es mindestens die hinzugefügten und gelöschten Elemente ausweist und zum anderen, dass keine redundante Erfassung der Änderungen vorliegt. Es wurde beobachtet, dass beide Werkzeuge zwar die Änderungen identifizieren konnten, allerdings unterschiedlich darstellen. Der ModelMatcher weist die Änderungen redundant aus (durch Aufzeigen der Änderungen an EMF-internen Referenzen), was bei einer automatischen Weiterverarbeitung des Differenzmodells lediglich die Performanz beeinträchtigen würde. Im Gegensatz dazu sind die Differenzen von EMF Compare minimaler als erforderlich. Hier werden bspw. hinzugefügte Elemente, welche Kinder von ebenfalls hinzugefügten Elementen sind, nicht explizit aufgeführt. Diese müssen bei einer Verarbeitung des Differenzmodells aus M_{orig} bzw. M_{mod} erneut separat ermittelt werden.

Kategorie 2: Diese Kategorie zeigte sowohl Probleme bei den Vergleichskriterien beider Werkzeuge, als auch beim Mapping der Werkzeuge auf. Beide Werkzeuge waren nicht in der Lage, alle Differenzen korrekt zu ermitteln. Problematisch zeigten sich die von EMF Compare verwendeten String-basierten Attributvergleiche, bei denen die Levenstheins Editierdistanz genutzt wird und die Verwendung von Baumpositionen während des Matchings der Modelle. Werden diese Kriterien für den Vergleich genutzt, kann es vorkommen, dass die ermittelten Vergleichswerte ungerechtfertigt negativ oder positiv beeinflusst werden. Dies ist bspw. der Fall, wenn die geänderten Elemente nur wenige Attribute besitzen, welche zum Vergleich herangezogen werden können. In diesem Fall kann es schwierig sein, ein geeignetes Mapping zu bilden. Der ModelMatcher verwendet hingegen die Datentyp-spezifischen Vergleichsoperatoren für Attributvergleiche und berücksichtigt die Positionen der Elemente nur, falls nach dem Matching mehrere mögliche Kombinationen für ein Element existieren, um eine eindeutige Abbildung zu finden. Dies führte während der Tests oft zu besseren Ergebnissen. Ein weiteres Problemfeld, welches identifiziert wurde, ist die Verwendung von Schwellenwerten für das Mapping der Modellelemente. Das Problem besteht darin, einen geeigneten Wert zu finden, welcher die Untergrenze für die Abbildung der Elemente festlegt. Bei EMF Compare konnte beobachtet werden, dass diese Schwellenwerte teilweise nicht überschritten wurden, obwohl korrespondierende Elemente existierten. Der ModelMatcher verzichtet hingegen auf die

Verwendung von Schwellenwerten und prüft für jeden maximalen Vergleichswert, ob dieser auch in entgegengesetzter Richtung maximal ist. Dies führt während des Mappings oft zu besseren Ergebnissen, da es möglich ist, korrespondierende Elemente auch mit niedrigen Vergleichswerten zu ermitteln.

Kategorie 3: Die Tests dieser Kategorie wurden vom ModelMatcher ohne Probleme bewältigt, EMF Compare berechnete hingegen teilweise fehlerhafte Differenzen. Wurde bspw. der Ablauf des Prozesses geändert, mussten die einzelnen Aktivitäten des Prozesses verschoben werden. Diese Verschiebung wirkte sich jedoch auf verschiedene Referenzen aus, wodurch diese Änderungen durch beide Werkzeuge korrekt erkannt wurden. Wurde jedoch ein Element verschoben, ohne den Container zu wechseln, wurden von EMF Compare durch die Verwendung der Baumpositionen sowie durch Verwendung der Editierdistanz in einigen Fällen Elemente als gelöscht und neu hinzugefügt betrachtet. Dies ist vor allem dann der Fall, wenn die verschobenen Modellelemente nur wenige Attribute besitzen und Elemente mit ähnlichen Attributwerten existieren. Eine mögliche Lösung des Problems setzt die Erkennung der Art der vorliegenden Verschiebungen voraus. Fraglich ist jedoch, wie Elementbewegungen, die die Semantik des Modells ändern von denen abgegrenzt werden können, die die Semantik des Modells nicht ändern. Eine denkbare Lösung ist eine Konfiguration der Vergleichsalgorithmen, um dem Vergleich Informationen über das verwendete Metamodell bereitzustellen.

4 Zusammenfassung

In diesem Artikel wurden Ansätze zur Ermittlung von Modelldifferenzen sowie eine Auswahl an differenzberechnenden Werkzeugen vorgestellt. Referenzen zu weiteren Werkzeugen wurden ebenfalls in Kapitel 2.2 genannt. Weiter existieren zahlreiche Arbeiten, welche Differenzen aus verschiedenen Blickwinkeln betrachten. So werden bspw. neben der Berechnung von Differenzen im Kontext der modellgetriebenen Softwareentwicklung [AP03, KPP06, Tou06, XiS05] Differenzen von XML-Modellen [WDC03] oder auch Datenbankschemata und Ontologien [ADMR05, RB01, HB08] untersucht.

Im Speziellen wurde der Einsatz von Modelldifferenzen bei der Modellierung von Geschäftsprozessen als Möglichkeit vorgestellt, notwendige manuelle Eingriffe bei der Evolution von Prozessmodellen möglichst minimal zu halten. Für die vorgestellten Verfahren, die Differenzen zweier Modelle zu ermitteln, ergaben sich verschiedene Einsatzfelder. So stellt bspw. die Protokollierung der Änderungen an Modellen die mit Abstand genaueste Methode dar, welche jedoch nicht in jedem Fall verwendet werden kann. In diesen Fällen ist es wichtig, auf alternative Methoden zur Differenzberechnung zurückgreifen zu können. Zur Evaluierung zweier Berechnungsansätze wurde abschließend eine Reihe von Tests mit den zugehörigen Werkzeugen EMF Compare und ModelMatcher durchgeführt. In diesem Abschnitt wurden Problemfelder der Differenzberechnung aufgezeigt.

Literatur

- [ADMR05] David Aumüller, Hong-Hai Do, Sabine Massmann und Erhard Rahm. Schema and ontology matching with COMA++. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, Seiten 906–908, New York, NY, USA, 2005. ACM.
- [AP03] Marcus Alanen und Ivan Porres. Difference and Union of Models. In *„UML“ 2003 – The Unified Modeling Language, Modeling Languages and Applications*, Seiten 2–17. 2003.
- [FKT08] Klaus-Peter Fähnrich, Stefan Kühne und Maik Thränert. *Model-Driven Integration Engineering – Modellierung, Validierung und Transformation zur Integration betrieblicher Anwendungssysteme*. Leipziger Beiträge zur Informatik, Band XI. Eigenverlag Leipziger Informatik-Verbund (LIV), Leipzig, September 2008.
- [HB08] Peter Hänsgen und Stefan Butz. Modellgetriebenes Verfahren zur Migration relationaler Datenbanken. In Fähnrich et al. [FKT08], Seiten 189–203.
- [Hil08] Stanley Hillner. Berechnung und Anwendung von Modelldifferenzen im Geschäftsprozessmanagement. Bachelorarbeit, September 2008.
- [KNS92] G. Keller, M. Nüttgens und August-Wilhelm Scheer. Semantische Prozeßmodellierung auf der Grundlage „Ereignisgesteuerter Prozeßketten (EPK)“. Arbeitsbericht Heft 89, Institut für Wirtschaftsinformatik Universität Saarbrücken, 1992.
- [KPP06] Dimitrios S. Kolovos, Richard F. Paige und Fiona A.C. Polack. Model comparison: a foundation for model composition and model transformation testing. In *GAMMA '06: Proceedings of the 2006 international workshop on Global integrated model management*, Seiten 13–20, New York, NY, USA, 2006. ACM.
- [KSI08] Stefan Kühne, Sebastian Stein und Konstantin Ivanov. Abbildung fachlicher Prozessmodelle auf BPEL-basierte Laufzeitumgebungen. In Fähnrich et al. [FKT08], Seiten 93–110.
- [RB01] Erhard Rahm und Philip A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.
- [Sch07] Maik Schmidt. SiDiff: generische, auf Ähnlichkeiten basierende Berechnung von Modelldifferenzen. In Ernst-Erich Doberkat und Udo Kelter, Hrsg., *Softwaretechnik-Trends*, Jgg. 27, 2007.
- [SKD⁺08] Sebastian Stein, Stefan Kühne, Jens Drawehn, Sven Feja und Werner Rotzoll. Evaluation of OrViA Framework for Model-Driven SOA Implementations: An Industrial Case Study. In Marlon Dumas, Manfred Reichert und Ming-Chien Shan, Hrsg., *Business Process Management*, Jgg. 5240 of LNCS, Seiten 310–325, Milan, Italy, September 2–4 2008. Springer.
- [TK08] Maik Thränert und Stefan Kühne. Model-Driven Integration Engineering zur Integration betrieblicher Anwendungssysteme. In Fähnrich et al. [FKT08], Seiten 17–33.
- [Tou06] Antoine Toulmé. Presentation of EMF Compare Utility. Eclipse Summit Europe 2006, November 2006.
- [WDC03] Y. Wang, D. Dewitt und J. Cai. X-Diff: A Fast Change Detection Algorithm for XML Documents, 2003.
- [XiS05] *UMLDiff: an algorithm for object-oriented design differencing*, New York, NY, USA, 2005. ACM.