

# Konzeption und Implementierung einer konfigurierbaren Komponente zum Modelldatenaustausch am Beispiel der bflow\* Toolbox

Lucas Wirmer, Nick Gehrke, Markus Nüttgens, Martina Peris

Fakultät Wirtschafts- und Sozialwissenschaften, Wirtschaftsinformatik  
Universität Hamburg  
Max-Brauer-Allee 60  
22765 Hamburg  
[vorname].[nachname]@wiso.uni-hamburg.de

**Abstract:** Der vorliegende Beitrag entwickelt ein Konzept zur interaktiven Transformation von Formaten zur Abbildung von Geschäftsprozessmodellen. Die Validierung des Konzepts erfolgt in der Form einer prototypischen Implementierung. Hierdurch werden exemplarisch konfigurierbare Modelltransformationen zwischen den Formaten der bflow-Toolbox (einer Open Source Modellierungssoftware) und marktgängigen Formaten wie der ARIS Markup Language (AML) und der Microsoft Visio Formate (VDX) unterstützt. Die Relevanz des Konzepts und der prototypischen Entwicklung zeigt sich im Wesentlichen in zwei Aspekten. Einerseits wird dem Nutzer ermöglicht, in den Transformationsprozess einzugreifen und Modellmanipulationen vorzunehmen. Diese Eingriffsmöglichkeit soll ein erhöhtes Nutzenpotenzial für den Endnutzer erschließen. Andererseits kann eine Erweiterung um Transformationsvorschriften und Formate ohne Eingriff in den originären Quellcode erfolgen. Diese Flexibilität und Konfigurierbarkeit soll die Produktivität und Akzeptanz in der praktischen Anwendung maßgeblich erhöhen.

## 1 Einleitung

Seit der Entstehung in den 80er Jahren hat das Leitbild der (Geschäfts-) Prozessorientierung und damit auch der Markt für sog. Business Process Management (BPM)-Werkzeuge an Bedeutung gewonnen [BMW09, S.2]. Im Laufe der Zeit wurden zahlreiche Methoden zur Modellierung von Geschäftsprozessen propagiert und in verschiedenen Softwaresystemen implementiert. Nutzerbefragungen belegen jedoch immer wieder die Bedeutung und bislang ungelöste Problematik des Austausches von Modelldaten zwischen verschiedenen BPM-Werkzeugen [De05]. Bis heute konnte sich trotz erster Initiativen kein neutrales Standardformat am Markt etablieren [MNN05]. Dem praktischen Problem der Interoperabilität im BPM-Kontext liegt das theoretische Konzept der Modelltransformation zugrunde [CH06]. Ein Geschäftsprozess in einem bestimmten Format kann abstrakt als ein Modell aufgefasst werden, das in eine andere Form transformiert werden soll.

Gegenstand dieses Beitrages ist die Konzeption und prototypische Implementierung einer konfigurierbaren Komponente für die Transformation von BPM-Modellen. Die

Eigenschaft der Konfigurierbarkeit stellt das Alleinstellungsmerkmal des hier entwickelten Ansatzes dar und soll den Anwender in die Lage versetzen, die Transformation an individuelle Anforderungen anzupassen. Eine Validierung des Konzeptes erfolgt in Form einer prototypischen Implementierung als Plugin für die bflow\*Toolbox (einer Open Source Modellierungssoftware).

## 2 Kontext und Motivation

Ein *Metamodell* ist ein Modell, welches zur Beschreibung anderer Modelle benutzt werden kann [AK03, Be06]. Unter einer *Modelltransformation* versteht man die „Überführung von einem Quellmodell in ein Zielmodell in Abhängigkeit von einer Transformationsdefinition“ [KKF06]. Es existiert eine Vielzahl von Ansätzen zur Durchführung von Modelltransformationen und deren Evaluation [CH06, Mv06]. Dennoch ist keine allgemeingültige Aussage über den besten Ansatz möglich, dies hängt stark vom Einsatzkontext ab.

Gleichmaßen existieren zahlreiche Methoden zur Geschäftsprozessmodellierung [vgl. Re09]. Ebenso fragmentiert zeigt sich der Markt für Software zur Geschäftsprozessmodellierung [vgl. B007]. Hinzu kommt das wachsende Interesse der Unternehmen Open Source Software einzusetzen [Re06]. Vor dem Kontext der Geschäftsprozessmodellierung wurden zahlreiche Intermediärformate zum Modelldatenaustausch entwickelt und miteinander verglichen [MNN05]. Bisher konnte sich jedoch kein Format durchsetzen. Zur eigentlichen Durchführung der Transformationen werden in der Literatur zwei verschiedene Ansätze verfolgt und diskutiert. Der eher als pragmatisch einzuordnende Ansatz ist die Nutzung von Transformationssprachen, die auf der Ebene der Serialisierung ansetzen, wie z.B. die XSLT [MN04, VZS05]. Zwar lässt sich hierdurch relativ schnell eine funktionierende Transformation erreichen, allerdings wird die mangelnde Eignung für komplexe Transformationen bemängelt. Oft genannte Schwachpunkte sind die schlechte Lesbarkeit und Wartbarkeit des Codes sowie die Performanz bei großen Transformationen [CH03, SK03, KKF06]. Für den vorliegenden Beitrag wurde daher der Schwerpunkt auf eine höhere Abstraktionsstufe gelegt, bei der die Transformationen auf Ebene der Metamodelle beschrieben und damit zunächst unabhängig vom Serialisierungsformat vorgenommen wird [Gr06, MK07].

Das Ausgangsproblem für diesen Beitrag ist die in der Praxis bestehende Herausforderung, Modelldaten aus verschiedenen Werkzeugen und in teilweise unterschiedlichen Formaten zu transformieren. Derzeit existiert keine Lösung, um eine zufriedenstellende Interoperabilität im Kontext der Geschäftsprozessmodellierung zu erreichen. Im Rahmen dieses Beitrags soll daher ein Konzept zum Austausch von Modelldaten zwischen verschiedenen BPM-Werkzeugen entwickelt werden. Eines der Hauptziele dabei ist die Sicherstellung einer größtmöglichen Flexibilität bezüglich der unterstützten Notationen. Dazu wurde die Definition eines intermediären Formats auf Ebene eines Metamodells als sinnvollster Ansatz identifiziert. Es soll daher ein Metamodell entworfen werden, das über die nötige Ausdrucksmächtigkeit zur Abbildung einer Vielzahl von im Kontext der Geschäftsprozessmodellierung verwendeten Konstrukten verfügt. Als wesentliches Merkmal der Abgrenzung und Alleinstellung soll eine Konfigurierbarkeit der Transfor-

mationskomponente dienen. In der Praxis sind zahlreiche Szenarien denkbar, bei denen durch eine solche Konfigurierbarkeit der vom Anwender empfundene Nutzen der Transformationskomponente stark gesteigert werden könnte. Mit Hilfe eines zu entwickelnden Prototyps soll die grundsätzliche Umsetzbarkeit des Konzeptes demonstriert werden.

### 3 Konzept

Jegliche im BPM-Kontext auftauchenden Modelle können als Form eines Graphen aufgefasst werden können. Hieraus kann gefolgert werden, dass das Metamodell die allgemeine Form eines Graphen repräsentieren, also im Grunde nur die Elemente Knoten und Kanten enthalten sollte. Damit bleibt die Semantik der Modellelemente zunächst unbestimmt. In Abbildung 1 ist der Aufbau des *generischen Metamodells* in Form eines UML-Klassendiagramms dargestellt. Es handelt sich um das Kompositum Pattern [Ga09]. Knoten und Kanten werden durch die Klassen *Node* bzw. *Edge* repräsentiert. Hinzu kommt die Klasse *Model*, die eine Menge von Knoten und Kanten als eine Art Container zusammenfasst und als Basiselement dient. Jede weitere Spezialisierung des intermediären Modells würde zwingend die Ausdrucksmächtigkeit und Flexibilität einschränken und ist daher nicht wünschenswert. Insbesondere ein Vergleich mit dem integrierten Metamodell von Murzek & Kramler [MK07] zeigt, dass weitere Elemente im hier propagierten Modell durch Attributierung auch ohne explizite Erweiterung des Metamodells hinzugefügt werden können.

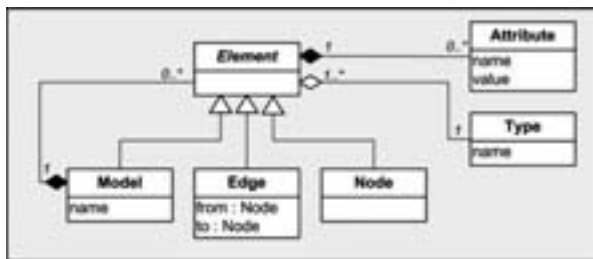


Abbildung 1: Generisches Metamodell

In Abbildung 2 werden die einzelnen Schritte einer Transformation dargestellt. Aus der Anforderung der Konfigurierbarkeit resultiert die Konzeption zweier Modi, dem sogenannten Standard- und dem Expertenmodus. Im Expertenmodus kann an zwei Stellen in die Transformation eingegriffen werden. Im Rahmen der Vorverarbeitung kann der Nutzer ausgewählte Elemente aus dem eingelesenen Modell löschen bzw. bearbeiten. Die zweite Form der Konfiguration ergibt sich bei der Prüfung der Constraints. Dabei wird dem Nutzer bei jeder fehlgeschlagenen Constraint-Prüfung eine vorab definierte Auswahl von Handlungsalternativen präsentiert, wodurch eine individuelle Gestaltung der Transformation ermöglicht wird. Die beiden Schritte des Expertenmodus können unabhängig voneinander gewählt werden. Zur Realisierung sieht das Konzept eine modulartige Architektur der Komponente vor, die entsprechende Eingriffe ermöglicht.



Abbildung 2: Ablauf der Transformation aus konzeptioneller Sicht

## 4 Prototypische Implementierung und Validierung

Eine wesentliche Rahmenbedingung für die Technologieauswahl ist die geplante Integration des Prototypen in die bflow\* Toolbox. Dies führte technologisch dazu, dass die Komponente als ein Plugin für die Eclipse-Plattform entwickelt wurde. Zur technischen Realisierung der Modelltransformationen stehen verschiedene Ansätze und Frameworks zur Wahl. Den Vorzug erhielt *openArchitectureWare* (oAW) [Ha07].

Abbildung 3 zeigt die Kernelemente der Softwarekomponente mit ihren wichtigsten Methoden und Assoziationen. Zur Erhöhung der Verständlichkeit werden ausgewählte relevante Inhalte dargestellt. Das Package *converter.internal* enthält die eigentliche Funktionalität. Als zentrale Konfigurationsdatei dient die *registry.xml*, über die sich sämtliche implementierten Modellformate bei der Anwendung registrieren. In der aktuellen Version unterstützt der Prototyp die Transformation von EPK-Modellen in den Formaten der Werkzeuge ARIS Toolset, bflow\* Toolbox und Microsoft Visio. Dabei wurden Abbildungen für ca. 15 der wichtigsten Elementtypen aus den einzelnen Metamodellen definiert. Zudem werden bei der Transformation als Attribute die Beschriftungen und wesentliche Grafikinformatoren übernommen, was den Wiedererkennungswert der transformierten Prozessmodelle durch den Anwender deutlich erhöht.

Der entwickelte Prototyp dient der Demonstration der Umsetzbarkeit des hier entwickelten Konzepts. Trotz der erfolgreich umgesetzten Grundfunktionalität, existieren Optimierungspotenziale in den Bereichen Ausnahmebehandlung durch die Benutzeroberfläche und Behebung verschiedener identifizierter Fehler. Auch wenn sich rein subjektiv keine nennenswerten Wartezeiten bei der Durchführung der Transformationen ergaben, sollte die Performanz vor einer Weiterentwicklung der Komponente näher untersucht werden. Ansatzpunkte für eine Weiterentwicklung bieten die Formatunterstützung, der Transformationsumfang sowie die Modell-Vorverarbeitung.

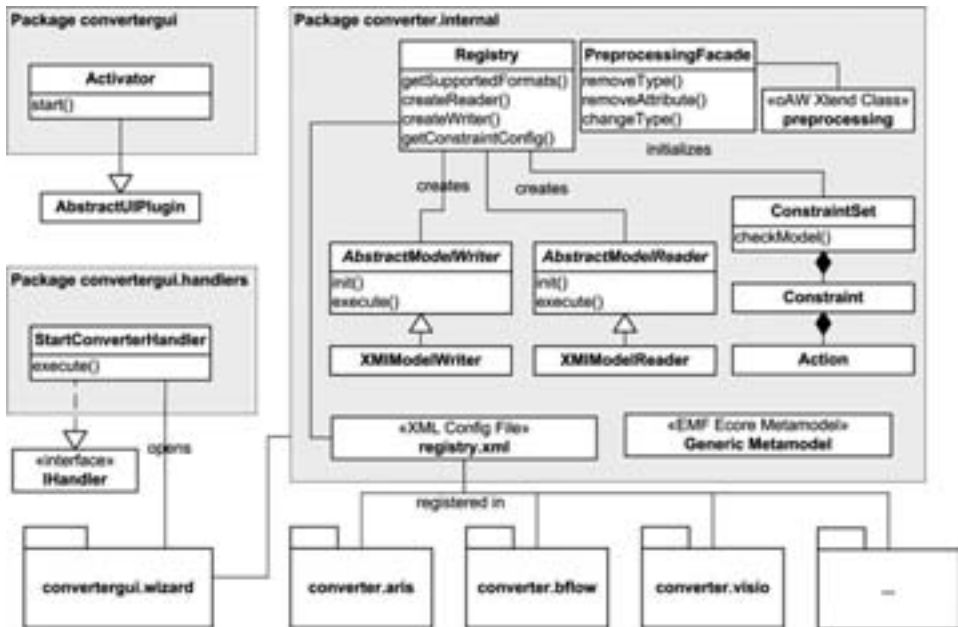


Abbildung 3: Architektur der prototypischen Umsetzung

## 5 Zusammenfassung

Im Rahmen dieses Beitrags wurde der Grundstein für eine konfigurierbare Komponente zur Durchführung von Modelltransformationen im BPM gelegt. Dabei wurde der Schwerpunkt auf eine höhere Abstraktionsstufe gelegt, bei der die Transformationen auf Ebene der Metamodelle beschrieben und mit Hilfe spezieller Frameworks durchgeführt werden. Das entwickelte Konzept zeichnet sich durch die Verwendung eines generischen Metamodells aus, das Prozessmodelle in einer mit Attributen angereicherten Form eines allgemeinen Graphen repräsentiert. Eine Konfigurierbarkeit des Transformationsprozesses ist an zwei Stellen vorgesehen (Experten-Modus).

Bedingt durch das Ziel einer prototypischen Entwicklung existieren noch verschiedene Optimierungspotenziale, die einem Praxiseinsatz im Wege stehen. Diese beziehen sich vornehmlich auf die Behandlung von Ausnahmezuständen und die Benutzbarkeit der grafischen Oberfläche. Die Software könnte dennoch als unabhängiger Vermittler zwischen der Vielzahl existierender Modellierungsformate im BPM-Kontext dienen. Gewisse Einschränkungen des Konzepts, wie der hohe Abstraktionsgrad des generischen Metamodells führen zu einer aufwendigen Plausibilitätsprüfung von Modellinstanzen und damit zu einer großen Zahl benötigter Constraints. Es existieren verschiedene Ansatzpunkte für eine Weiterentwicklung des Prototyps. In erster Linie bietet sich eine Implementierung der Unterstützung weiterer Formate an, verbunden mit einer Erhöhung des Umfangs der transformierten Modellelemente. Zusätzlich könnte die Übertragung des Konzepts auf andere Anwendungsdomänen Inhalt weiterer Arbeiten sein.

## Literaturverzeichnis

- [AK03] Atkinson, C.; Kühne, T.: Model-Driven Development: A Metamodeling Foundation. In *IEEE Software*, 2003, 20; S. 36–41.
- [Be06] Berio, G.: UEML 1.0 and UEML 2.0: Benefits, Problems and Comparison. In (Bussler, C.; Haller, A. Hrsg.): *Business process management workshops. BPM 2005 international workshops BPI, BPD, ENEI, BPRM, WSCOBPM, BPS*, Nancy, France, September 5, 2005; revised selected papers. Springer, Berlin, 2006; S. 245–256.
- [BMW09] Becker, J.; Mathas, C.; Winkelmann, A.: *Geschäftsprozessmanagement*. Springer, Berlin, 2009.
- [Bö07] Böhn, M. et al.: *Prozessmodellierungswerkzeuge. 14 Systeme für Dokumentation, Entwurf, Simulation und Analyse im Vergleich; eine Studie des Business Application Research Centers*. Oxygon-Verl., München, 2007.
- [CH03] Czarnecki, K.; Helsen, S.: Classification of Model Transformation Approaches. [http://www.swen.uwaterloo.ca/~kczarneck/ECE750T7/czarnecki\\_helsen.pdf](http://www.swen.uwaterloo.ca/~kczarneck/ECE750T7/czarnecki_helsen.pdf), 15.09.2009.
- [CH06] Czarnecki, K.; Helsen, S.: Feature-based survey of model transformation approaches. In *IBM Systems Journal*, 2006, 45; S. 621–645.
- [De05] *BPM 2005. Market Milestone Report. A Delphi Group White Paper*, Boston, Mass., 2005.
- [Ga09] Gamma, E. et al.: *Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software*. Addison Wesley, München, 2009.
- [Gr06] Grangel, R. et al.: Exchange of Business Process Models Using the POP\* Meta-model. In (Bussler, C.; Haller, A. Hrsg.): *Business process management workshops. BPM 2005 international workshops BPI, BPD, ENEI, BPRM, WSCOBPM, BPS*, Nancy, France, September 5, 2005; revised selected papers. Springer, Berlin, 2006; S. 233–244.
- [Ha07] Haase, A. et al.: Introduction to openArchitectureWare 4.1.2. <http://www.dsmforum.org/events/MDD-TIF07/oAW.pdf>, 15.09.2009.
- [KKF06] Kern, H.; Kühne, S.; Fötsch, D.: Merkmale und Werkzeugunterstützung für Modelltransformationen im Kontext modellgetriebener Softwareentwicklung. In (Fähnrich, K.-P.; Kühne, S.; Speck, A.; Wagner, J. Hrsg.): *Integration betrieblicher Informationssysteme. Problemanalysen und Lösungsansätze des Model-Driven-Integration-Engineering*. LIV, Leipzig, 2006.
- [MK07] Murzek, M.; Kramler, G.: *The Model Morphing Approach - Horizontal Transformations between Business Process Models*, Tampere, Finland, 2007.
- [MN04] Mendling, J.; Nüttgens, M.: Transformation of ARIS Markup Language to EPML. In (Nüttgens, M.; Rump, F. J. Hrsg.): *Proceedings of the 3rd GI Workshop on Event-Driven Process Chains (EPK 2004)*, Luxembourg, October 2004.
- [MNN05] Mendling, J.; Neumann, G.; Nüttgens, M.: A Comparison of XML Interchange Formats for Business Process Modelling. In (Fischer, L. Hrsg.): *Workflow Handbook 2005. Future Strategies Inc, Lighthouse Point, Florida*, 2005; S. 185–198.
- [Mv06] Mens, T.; van Gorp, P.: A Taxonomy of Model Transformation. *Proceedings of the International Workshop on Graph and Model Transformation (GraMoT 2005)*. In *Electronic Notes in Theoretical Computer Science*, 2006, 152; S. 125–142.
- [Re06] Renner, T. et al.: *Open Source Software: Einsatzpotenziale und Wirtschaftlichkeit*. Fraunhofer IAO, Stuttgart, 2006.
- [Re09] Recker, J. et al.: *Business Process Modelling - A Comparative Analysis*. In *Journal of the Association for Information Systems*, 2009, 10; S. 333–363.
- [SK03] Shane Sendall und Wojtek Kozaczynski. *Model Transformation: The Heart and Soul of Model-Driven Software Development*. *IEEE Software*, 20(5):42–45, 2003.
- [VZS05] Vanderhaeghen, D.; Zang, S.; Scheer, A.-W.: *Interorganisationales Geschäftsprozessmanagement durch Modelltransformation*, Saarbrücken, Februar 2005.

