

Vom Prozess zur Simulation - ein Transformationsmodell-Ansatz

Oliver Kloos; Volker Nissen
Technische Universität Ilmenau
Fachgebiet Wirtschaftsinformatik für Dienstleistungen
PF 10 05 65
98684 Ilmenau
{oliver.kloos;volker.nissen}@tu-ilmenau.de

1 Einleitung

Simulation stellt ein Werkzeug dar, welches für die Analyse von Prozessen, zur Ermittlung von Schwachstellen, zur Evaluation alternativer Abläufe oder zur Erstellung von Prognosen verwendet werden kann. (s. [NeRS05]). Zur Entscheidungsunterstützung werden Simulationsansätze in verschiedenen Domänen angewendet (s [MTLN95], [DeEr00], [LaLa04], [Bang08]). Die Simulationsmodelle werden auf Basis der in Unternehmen vorhandenen Abläufe erstellt. Diese Abläufe werden außerhalb der Produktion unter anderem mit Notationen zur Modellierung von Geschäftsprozessen dokumentiert (s. [Gada10]). Diese Geschäftsprozessmodelle können ihrerseits als Grundlage für die Erzeugung eines Simulationsmodells verwendet werden.

Der Artikel stellt einen Transformationsmodell-Ansatz vor, welcher für die Vorbereitung von Prozessmodellen auf eine Simulation verwendet werden kann. Mit dem ProSiT (Process to Simulation Transformation) Konzept soll die Vorbereitung der Simulation unabhängig von einer konkreten Simulationsumgebung auf Basis der Prozessmodelle erfolgen. Prozessmodelle werden hierfür in ein Transformationsmodell überführt. Dieses verwendet verschiedene Methoden, um ein Prozessmodell mit zusätzlichen Daten anzureichern, sowie eine Regelbasis, mit der die Komplexität reduziert werden kann.

In Kapitel 2 werden zunächst verwandte Notationen gewürdigt, welche für einen Transformationsmodell-Ansatz verwendet werden können. Kapitel 3 stellt kurz das ProSiT Konzept kurz vor und betrachtet dieses aus Sicht der Model Driven Architec-

ture (MDA). Der Artikel schließt mit einer Zusammenfassung in Kapitel 4 und einem Ausblick auf weiteren Forschungsbedarf.

2 Verwandte Arbeiten

Der Einsatz eines Transformationsmodell-Ansatzes für ein Simulationsvorhaben ähnelt der Model Driven Architecture aus der Domäne der Softwareentwicklung (s. [GrDC06]). Die MDA zielt auf eine Spezifikation einer Software ab, welche unabhängig von der technischen Umsetzung beschrieben wird. Bei der Spezifikation ist jedoch auf die Unterteilung von Plattformunabhängigkeit und Plattformabhängigkeit zu achten. Das ProSiT Konzept setzt jedoch nicht bei der Spezifikation, der Modellierung, von Prozessen an. Der Einsatz des Transformationsmodell-Ansatzes fordert bereits modellierte Prozessmodelle. Der MDA Ansatz bietet jedoch einen Rahmen, welcher für die Klassifikation des ProSiT Konzepts verwendet werden kann (siehe Kapitel 4).

Zur Darstellung des Transformationsmodells – des ProSiT Ablaufdiagramms – im ProSiT Konzept wird eine eigene Notation verwendet (s. [KINP09], [KNPS10]). Kandidaten für mögliche alternative Notationen sind SDL, SysML oder BPMN.

Kuhn et al. (s. [KuGW06]) führen anhand eines Beispiels den Einsatz von SDL im Kontext einer modellgetriebenen Entwicklung auf und Bucci et al. (s. [BuFV03]) sprechen von einer Modellierungssicht und Simulationssicht im Kontext von SDL. Die Verwendung von SDL zur Beschreibung von Softwareprozessen wird von Podnar et al. (s. [PoMA00]) demonstriert. Der Einsatz von SDL ist in der Telekommunikationsbranche und insbeson-

dere im Kontext von Echtzeitsystemen vorzufinden. Durch die Erweiterung von Bucci et al. (s. [BuFV03]) werden Schwächen von SDL mit fehlender Zeitunterstützung und der Annahme von SDL von unbegrenzten Ressourcen behoben. Die simulations-technischen Erweiterungen berücksichtigen jedoch nicht die Verwendung mehrerer Ressourcen für eine Funktion, sowie eine feste Bindung von Ressourcen für mehrere Funktionen. Ebenfalls zielt die Simulationserweiterung auf die Simulation von Echtzeitsystemen ab. Mit Geschäftsprozessmodellen werden jedoch nicht Echtzeitsysteme beschrieben, sondern zeitdiskrete Abläufe.

Bei SysML handelt es sich um eine standardisierte Teilmenge und Erweiterung von UML. SysML ist für die Spezifikation, Analyse und Design von komplexen Systemen entwickelt worden (s. [HoPe08]). Simulation von Abläufen wird nicht zu den Einsatzzwecken von SysML gezählt. Zeitverbrauch von Funktionen, sowie notwendige Ressourcen für deren Ausführung können daher mit den Modellen in SysML nicht definiert werden.

Als eine Notation zur Modellierung von Geschäftsprozessen eignet sich neben der erweiterten Ereignisgesteuerten Prozesskette (eEPK) die Business Process Modeling Notation (BPMN) (s. [Allw08]). BPMN vereinbart laut den Spezifikationen der OMG (s. [OMG09]) die besten Ideen von verschiedenen Notationen zur Modellierung von Prozessabläufen. Die Spezifikationen besagen jedoch, dass Ressourcen und Organisationsmodelle nicht Teil von BPMN sind. In den vorläufigen Spezifikationen von BPMN 2.0 beta 2 wurde jedoch im Vergleich zu Version 1.2 die Nutzung von Ressourcen in den Aktivitäten ausgebaut.

Ebenfalls erlaubt die Spezifikation der Version 2.0 beta 2 mehr Möglichkeiten, ein BPMN Diagramm zu erweitern, beispielsweise um zusätzliche Attribute oder Elemente (s. [OMG10]). Die endgültigen Spezifikationen der BPMN in Version 2.0 stellen eine mögliche Notation dar, die im Rahmen eines Transformationsmodell-Ansatzes verwendet werden könnte.

Wenn das BPMN Metamodellmodell um weitere Elemente wie Warteschlangen oder Ressourcenbindungen erweitert wird und eine Simulationsumgebung vorliegt, welche BPMN Diagramme direkt simulieren kann, dann könnte BPMN als Zielsprache für die Vorbereitung und eigentlichen Simulation von Geschäftsprozessen verwendet werden. Wenn aber das Transformationsmodell lediglich als Grundlage für ein Simulationsmodell in einer Simulationsumgebung dienen soll, dann ist ein BPMN Modell insbesondere bei der Verwendung von Nachrichtenflüssen und verschiedenen Ereignissen problematisch. Für jedes Ereignis müsste eine entsprechende Repräsentation in den Simulationsumgebungen identifiziert und entwickelt werden. Nachrichtenflüsse müssten in Sequenzflüsse umgewandelt werden bei Wahrung des gleichen syntaktischen Kontextes. Durch diese Erweiterbarkeit bietet sich BPMN, in der endgültigen Spezifikation der Version 2.0 als Notation an, wenn die Entwicklung des ProSiT Konzepts abgeschlossen ist.

3 Das ProSiT Konzept

Das ProSiT Konzept definiert ein Regelwerk und ein Transformationsmodell, dessen Einsatzzweck die Vorbereitung von Prozessmodellen auf eine Simulation ist. Ein Prozessmodell ist ein

Modell eines Prozesses, welcher in Anlehnung an Rosemann (s. [Rose96]) als eine zeitlich-logische Abfolge von Arbeitsschritten definiert ist. Die Definition des Begriffs Simulation erfolgt nach der VDI Richtlinie 3633 [VDI95]: „Simulation ist ein Verfahren zur Nachbildung eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind.“

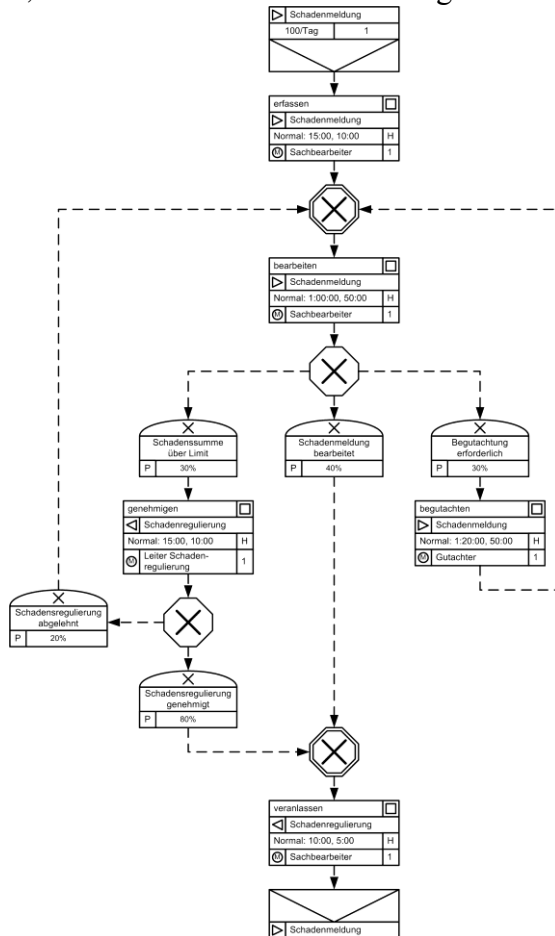


Abb. 1: Beispielprozess in der Notation des ProSiT Ablaufdiagramms

Gegenwärtig befindet sich das ProSiT Konzept noch in der Entwicklung. Die verwendete Notation (s. [KINP09]) ist daher als Arbeitsnotation aufzufassen. Die Arbeitsnotation ist insbesondere dienlich, um die Regelbasis, insbesondere semiautomatische und automatische Normalisierungsregeln, zu entwickeln.

Anwendung findet das ProSiT Konzept, wenn bestehende Prozessmodelle vorliegen. Eine Anwendung ist nicht vorgesehen, wenn keine Prozessmodelle als Grundlage zur Verfügung stehen. In Abbildung 1 ist ein Beispiel für die Arbeitsnotation des ProSiT Ablaufdiagramms dargestellt. Der Prozess beschreibt die Bearbeitung von Schadenmeldungen und wurde aus einer eEPK in [Allw05] in das Ablaufdiagramm überführt.

Das ProSiT Konzept orientiert sich an dem Konzept einer Middleware. Prozessmodelle stellen die Quelle dar, Simulationsmodelle die Ziele. Das Transformationsmodell dient einer einheitlichen Vorbereitung von Prozessmodellen in einer Simulationsumgebung. Abbildung 2 verdeutlicht dieses Konzept.

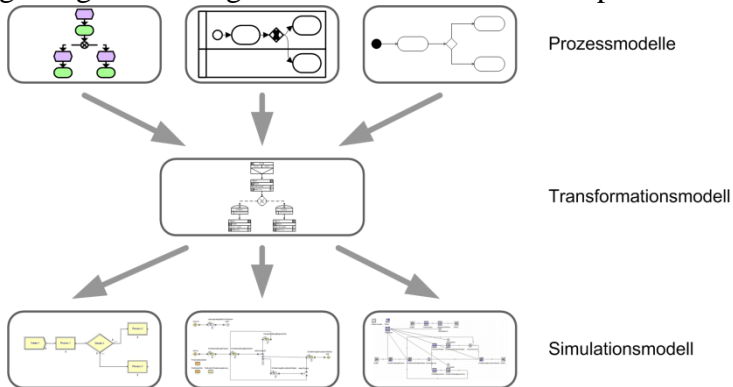


Abb. 2: ProSiT als Konzept einer Middleware

Die Betrachtung des ProSiT Konzepts aus Sicht des Konzepts einer Middleware zeigt auf, dass die Anzahl der Überführungs-

regeln im Vergleich zu einer direkten Transformation der Prozessmodell in Simulationsumgebungen reduziert wird.

Eine andere Sicht, aus der das ProSiT Konzept betrachtet werden kann, ist die Model Driven Architecture. In der MDA sind drei Modelltypen vorzufinden, Computation Independent Model (CIM), Platform Independent Model (PIM) und Platform Specific Model (PSM) (s. [GrDC06]). Das CIM stellt eine Sicht auf das Gesamtsystem dar, unabhängig davon, wie dieses implementiert wird. Das PIM beschreibt formal die Struktur und Funktionalität und ist unabhängig von implementierungstechnischen Details. Das PSM hingegen enthält plattformabhängige Informationen. Diese Modelle werden auch als ausführbare Modelle bezeichnet. In Abbildung 3 ist eine Analogie zwischen der MDA und dem ProSiT Konzept dargestellt.

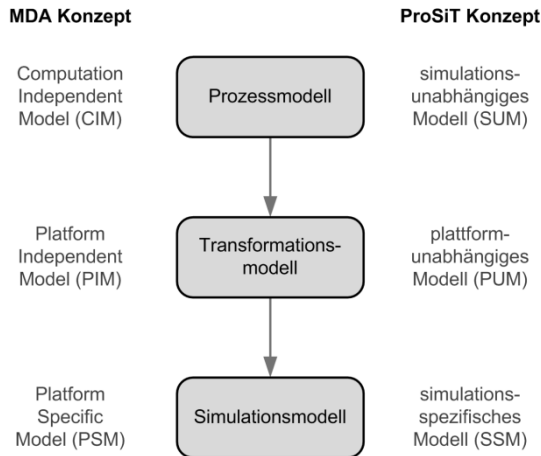


Abb. 3: Analogie zwischen MDA und dem ProSiT Konzept

Als Quellmodelle für das Transformationsmodell dienen primär Prozessmodelle, aber auch andere Modelle können als Quellmodelle verwendet werden, wie beispielsweise Organigramme, oder Funktionsbäume. Das ProSiT Konzept nimmt als Annah-

me, dass diese Modelle ohne Berücksichtigung einer Simulation erstellt wurden. Quellmodelle für das Transformationsmodell können daher in Analogie zum MDA Konzept als simulationsunabhängige Modelle bezeichnet (SUM) werden. Wenn diese Modelle als Ausgangspunkt für ein Simulationsmodell herangezogen werden, dann werden diese zunächst in das Transformationsmodell überführt. Für das Transformationsmodell werden im ProSiT Konzept Normalisierungsschritte entworfen, welche das Modell auf eine Simulation vorbereiten. Die Normalisierung umfasst manuelle Normalisierungsschritte, semiautomatische Normalisierungsregeln und automatische Normalisierungsregeln, sowie Ansätze, mit denen Daten beschafft werden können, welche für ein Simulationsmodell notwendig sind (s. [KNPS10]). Das Transformationsmodell ist dabei unabhängig von der für die Simulation eingesetzten Simulationssoftware und kann somit als plattformunabhängiges Modell (PUM) bezeichnet werden. Wenn die Normalisierung des Transformationsmodells abgeschlossen ist, dann kann dieses in eine Simulationsumgebung überführt werden. Gegenwärtig sind Transformationsregeln in die Simulationsumgebungen AnyLogic, Arena und Plant Simulation formuliert. Aufgrund des unterschiedlichen Funktionsumfangs müssen in der Simulationsumgebung noch letzte Einstellungen vorgenommen werden, welche nicht unabhängig von der Simulationsumgebung getroffen werden können. Die eigentlichen Simulationsmodelle werden als simulationspezifische Modelle (SSM) bezeichnet.

Durch die Betrachtung aus Sicht der MDA können die Regeln, welche im ProSiT Konzept definiert werden, entsprechend klas-

sifiziert werden. In [GrDC06] sind sechs Anwendungsfälle für Transformationen aufgeführt. Bei einer *Verfeinerung* wird das Modell um zusätzliche Informationen angereichert. Eine *Reduktion* auf wesentliche Elemente erfolgt bei der Abstraktion. Die *Migration* ist die Überführung eines Modells in eine andere Plattform. Eine *Refakturierung* erfolgt, wenn sich die innere Struktur eines Modells verändert. Bei der *Optimierung* soll der Ablauf des Modells verbessert werden und bei einer *Darstellungsumformung* ändert sich die innere Struktur oder das äußere Verhalten nicht, lediglich die Repräsentation wird verändert. Eine Klassifizierung der Regelbasis nach den Anwendungsfällen, die in [KNPS10] zum ProSiT Konzept aufgeführt ist, wird in Tabelle 1 dargestellt.

	Verfeinerung	Abstraktion	Migration	Refakturierung	Optimierung	Darstellungsum.
<i>Im Prozessmodell</i>						
Syntaxregeln SR						
Vorbereitungsregeln PR		X				
Transformationsregeln TR			X			(X)
<i>Im Transformationsmodell</i>						
automatische Normalisierungsregeln aNR		X				
semiautomatische Normalisierungsregeln sNR	X	X				
manuelle Normalisierung mN	X	X		X		
<i>Zum Simulationsmodell</i>						
Anwendbarkeitsregel AR						
Transformationsregeln TR			X			(X)

Tab. 1: Klassifikation der Regelbasis des ProSiT Konzepts

Aus der Klassifikation der Regelbasis geht hervor, dass mit dem ProSiT Konzept primär eine Verfeinerung und Abstraktion erfolgt. Transformationsregeln führen ein Modell in eine andere Notation oder Simulationsumgebung und vollziehen daher eine Migration beziehungsweise kann auch als Darstellungsumformung bezeichnet werden. Lediglich bei manuellen Eingriffen wird die innere Struktur des Ablaufdiagramms geändert.

4 Zusammenfassung und Fazit

Der Artikel betrachte den ProSiT Transformationsmodell-Ansatz aus Sicht der Model Driven Architecture und klassifizierte die Regelbasis gemäß den Regeln, wie diese in der MDA auftreten. Erkenntnisse aus der MDA können daher auf andere Domänen außerhalb der Software Entwicklung übertragen werden.

Als eine Notation für ein Transformationsmodell, um Prozessmodelle für eine Simulation vorzubereiten, stellt BPMN einen möglichen Kandidaten für das Ablaufdiagramm dar. Mit entsprechenden Erweiterungen können alle Elemente, die im ProSiT Ablaufdiagramm verwendet werden, in BPMN umgesetzt werden. Die Notation und das ProSiT Konzept befindet sich gegenwärtig jedoch noch in Entwicklung und sollte als Arbeitsnotation aufgefasst werden. Ebenfalls befindet sich BPMN in Version 2.0 gegenwärtig noch im Prozess der Standardisierung. Nach Abschluss der Forschungsarbeit am Transformationsmodell-Ansatz und des Abschlusses der Standardisierung von BPMN, ist BPMN als mögliche Notation für den Transformationsmodell-Ansatz zu prüfen.

Die Regelbasis für die Normalisierung des Transformationsmodells wird allgemein definiert und ist nicht vom Ablaufdiagramm abhängig. Diese kann daher an andere Notationen angepasst werden. Elemente in BPMN müssen lediglich mit entsprechenden Attributen erweitert werden.

Literaturverzeichnis

- [Allw05] Allweyer, Thomas: Geschäftsprozessmanagement - Strategie, Entwurf, Implementierung, Controlling. W3L, Herdecke, 2005.

- [Allw08] Allweyer, Thomas: BPMN Business Process Modeling Notation - Einführung in den Standard für die Geschäftsprozessmodellierung. Books on Demand, Norderstedt, 2008.

- [Bang08] Bangsow, Steffen: Fertigungssimulation mit Plant Simulation. Carl Hanser Verlag, München, 2008.

- [BuFV03] Bucci, Giacomo; Fedeli, Andrea; Vicario, Enrico: Specification and Simulation of Real Time Concurrent Systems Using Standard SDL Tools. In: Reed, Rick; Reed, Jeanne (Hrsg.): 11th International SDL Forum, Stuttgart, 2003, S. 203-217.

- [DeEr00] Desel, Jörg; Erwin, Thomas: Modeling, Simulation and Analysis of Business Processes. In: Aalst, Wil van der; Desel, Jörg; Oberweis, An-

- dreas (Hrsg.): Business Process Management - Models, Techniques, and Empirical Studies. Springer, Berlin, 2000, S. 129-141.
- [Gada10] Gadatsch, Andreas: Grundkurs Geschäftsprozess-Management. 6. Aufl., Vieweg, Wiesbaden, 2010.
- [GrDC06] Gruhn, Volker; Pieper, Daniel; Röttgers, Carsten: MDA : Effektives Software-Engineering mit UML 2 und Eclipse. Springer, Berlin, 2006.
- [HoPe08] Holt, Jon; Perry, Simon: SysML for Systems Engineering. Institution of Engineering and Technology, London, 2008.
- [KINP09] Kloos, Oliver; Nissen, Volker; Petsch, Mathias: From Process to Simulation – A Transformation Model Approach. In: Mendling, Jan; Rinderle-Ma, Stefanie; Esswein Werner (Hrsg.): Proceedings of the 3rd International Workshop EMISA 2009. Gesellschaft für Informatik, Bonn, 2009.
- [KNPS10] Kloos, Oliver; Schorcht, Hagen; Petsch, Mathias;; Nissen, Volker: Dienstleistungsmodellierung als Grundlage für eine Simulation. In: Thomas Oliver; Nüttgens, Markus (Hrsg.): Dienstleistungsmodellierung 2010. Springer, Berlin, 2010, S. 86-106.

- [KuGW06] Kuhn Thomas; Gotzhein, Reinhard; Webel, Christian: Model-Driven Development with SDL - Process, Tools, and Experiences. In: Nierstrasz, Oscar; Whittle, Jon; Harel, David; Reggio, Gianna (Hrsg.): Model Driven Engineering Languages and Systems. Springer, Berlin, 2006, S. 83-97.
- [LaLa04] Lam, Kokin; Lau, R.S.M.: A simulation approach to restructuring call centers. In: Business Process Management Journal 4 (10), S. 481-494.
- [MTLN95] McAleer, W.E.; Turner, J.A.; Lismore, D.; Naqvi, I.A.: Simulation of a hospital's theatre suite. In: Journal of Management in Medicine 5 (9), S. 14-26.
- [NeRS05] Neumann, Stefan; Rosemann, Michael; Schwegmann, Ansgar : Simulation von Geschäftsprozessen. In: Becker, Jörg; Kugeler, Martin; Rosemann, Michael (Hrsg.): Prozessmanagement - Ein Leitfaden zur prozessorientierten Organisationsgestaltung. 5. Aufl., Springer, Berlin, 2005, S. 435-453.
- [PoMA00] Podnar, Ivana; Mikac, Branko; Caric, Antun: SDL Based Approach to Software Process Modeling. In; Conradi, Reidar (Hrsg.) 7th European Workshop, EWSPT 2000, Kaprun, 2000, S. 190-202.

- [OMG09] OMG: Business Process Model and Notation (BPMN) Version 1.2. 2009, <http://www.omg.org/spec/BPMN/1.2/PDF/>. Abruf am 2010-09-09.
- [OMG10] OMG: Business Process Model and Notation (BPMN) FTF Beta 1 for Version 2.0. 2010, <http://www.omg.org/cgi-bin/doc?dtc/10-06-04.pdf>. Abruf am 2010-09-09.
- [Rose96] Rosemann, Michael: Komplexitätsmanagement in Prozeßmodellen: Methodenspezifische Gestaltungsempfehlungen für die Informationsverarbeitung. Gabler, Wiesbaden, 1996.
- [VDI95] VDI: Simulation von Logistik-, Materialfluß- und Produktionssysteme – Begriffsdefinitionen. Verein Deutscher Ingenieure, Berlin, 1995.