

# Dynamische Prozesse und deren Abbildung auf ausführbare Workflows

Andreas Hoheisel<sup>1</sup>, Michael Fellmann<sup>2</sup>, Thorsten Dollmann<sup>3</sup>,

<sup>1</sup> Fraunhofer-Institut für Rechnerarchitektur und Softwaretechnik (FIRST)  
Kekuléstraße 7, 12489 Berlin  
andreas.hoheisel@first.fraunhofer.de

<sup>2</sup> Universität Osnabrück  
Institut für Informationsmanagement und Unternehmensführung (IMU)  
Lehrstuhl für Informationsmanagement und Wirtschaftsinformatik  
Katharinenstraße 3, 49069 Osnabrück  
michael.fellmann@uos.de

<sup>3</sup> Institut für Wirtschaftsinformatik (IW<sub>i</sub>) im DFKI  
Stuhlsatzenhausweg 3, 66123 Saarbrücken  
thorsten.dollmann@iwi.dfki.de

**Zusammenfassung:** Der Beitrag beschreibt Überlegungen und neue Ansätze zur Unterstützung der Dynamik von Prozessen in IT-gestützten Geschäftsprozessmanagementsystemen für SOA, Grid- und Cloud-Umgebungen. Hierbei werden häufige Änderungen von Prozessen sowohl auf der fachlichen Ebene als auch auf der Ebene der Implementierung und Ausführung berücksichtigt. Der dargestellte Ansatz basiert auf einer automatischen Transformation von fachlichen in ausführbare Prozessmodelle sowie deren Ausführung und Beobachtung mit dem *Grid Workflow Execution Service* (GWES).

## 1 Einleitung

Aktuelle Systeme zum IT-gestützten Geschäftsprozessmanagement (*Business Process Management – BPM*) gliedern den Ablauf des Prozessmanagements in der Regel in drei getrennte, aufeinanderfolgende Phasen: Prozessmodellierung, Prozessautomatisierung und Prozessausführung [Pe09]. In der Prozessmodellierung werden fachliche Modelle der Geschäftsprozesse erstellt und analysiert. In der Prozessautomatisierung werden aus fachlichen Modellen technische Prozesse abgeleitet, welche als Grundlage für die IT-gestützte Automatisierung dienen. Der Begriff „Automatisierung“ bezieht sich hierbei auf das Prozessmanagement; ein Prozess gilt somit auch dann als automatisiert, wenn einzelne Aktivitäten durch Personen ausgeführt werden. In der Prozessausführung schließlich erfolgt die Ausführung der Prozesse, zum Beispiel in einer SOA, Grid- oder Cloud-Umgebung. Für die drei Phasen der Modellierung, Automatisierung und Ausführung kommen meist jeweils eigenständige Methoden und Werkzeuge zum Einsatz, welche die Durchlässigkeit zwischen den Phasen behindern. Ein etablierter Ansatz ist dabei

die Verwendung der *Business Process Modeling Notation* (BPML) [Ob09] zur Prozessmodellierung und die anschließende Überführung in die *Web Services Business Process Execution Language* (WS-BPEL) [Oa07] anhand der im Anhang A des BPMN-Standards enthaltenen Transformationsregeln. Allerdings bringt dieser Ansatz im Kontext dynamischer und verteilter Prozesse zwei grundlegende Probleme mit sich. Zum einen gestaltet sich eine Transformation von BPMN-Modellen in WS-BPEL-Skripte aufgrund der umfangreichen BPMN-Konstrukte komplex und fehleranfällig. Zum anderen wird eine Virtualisierung bei der Prozessautomatisierung im Kontext von WS-BPEL nicht explizit berücksichtigt.

Der vorliegende Artikel beschreibt ein Vorgehen, das die Dynamik von Prozessen durch eine engere Verzahnung der drei Phasen des Geschäftsprozessmanagements besser unterstützen soll. Basis hierfür ist die Verwendung eines über weite Teile einheitlichen Prozessbeschreibungsformalismus, der eine einfache Abbildung zwischen den Phasen ermöglicht und somit eine direkte Interaktion zwischen Modellierung, Automatisierung und Ausführung zulässt. Dadurch soll eine bessere Abstimmung zwischen fachlichen und technischen Prozessen realisiert werden, was insbesondere für dynamische Prozesse zweckdienlich ist. Ad-hoc-Änderungen in fachlichen Prozessen (z. B. „Unterauftragnehmer insolvent“) können dadurch einfacher an die technische Ebene durchgereicht werden. Umgekehrt ist eine schnelle Reaktion auf Änderungen in der technischen Ebene (z. B. „Datenspeicher voll“) eine wichtige Voraussetzung für die effektive Anwendbarkeit von Geschäftsprozessmanagement.

Dieser Artikel ist eine Weiterführung der Arbeiten aus [Ho09], in welchen der Fokus auf den technischen Aspekten der Abbildungsmethodik zur Überführung von ereignisgesteuerten Prozessketten in Petrinetze lag. Der Schwerpunkt des vorliegenden Artikels liegt darüber hinaus in allgemeinen Überlegungen zur Unterstützung der Dynamik von Prozessen. Diese Arbeit wurde gefördert durch die BMBF-Projekte der D-Grid-Initiative BauVOGrid (Förderkennzeichen: 01IG07001) und PneumoGRID (Förderkennzeichen: 01IG09002D).

## **2 Dynamische Prozesse und Ressourcen**

Die Dynamik von Prozessen und Ressourcen lässt sich anhand der Variabilität von Umwelt und Prozess in vier Kategorien unterteilen:

1. Konstanter Prozess in konstanter Umwelt. Ein Beispiel hierfür sind Unternehmen, die in stabilen Märkten mit einem konstanten Prozess immer gleichartige Leistungen anbieten.
2. Konstanter Prozess in dynamischer Umwelt. Ein Beispiel hierfür ist ein Industrieunternehmen, dessen Fertigungsprozess tolerant ausgelegt sein muss gegenüber dem Ausfall einer Ressource wie einer Maschine oder eines IT-Dienstes.
3. Dynamischer Prozess in konstanter Umwelt. Ein Beispiel hierfür ist ein Unternehmen, dessen Produkte aufgrund einer Unternehmensübernahme ggf. mit anderen (zeitlich konstanten) Ressourcen erstellt werden müssen.

4. Dynamischer Prozess in dynamischer Umwelt. Ein Beispiel hierfür ist ein Unternehmen, dessen Geschäftsprozess ständig an sich ändernde Märkte und Ressourcen angepasst werden muss.

Die für unsere Methodik relevanten Anwendungsszenarien stammen aus den Projekten BauVOGrid und PneumoGRID, in denen sowohl variable wissenschaftliche als auch kommerziell nutzbare IT-Prozesse auf unregelmäßig verfügbaren und fehleranfälligen Grid-Knoten ausgeführt werden sollen. Schwerpunkt unseres Ansatzes ist daher die Berücksichtigung von Prozess- und Ressourcendynamik gemäß Kategorie 4. Bezüglich der drei Phasen Modellierung, Automatisierung und Beobachtung ergeben sich dabei folgende Lösungsansätze:

*Phase 1 – Prozessmodellierung:* Die Auswirkung von Änderungen in fachlichen Prozessen werden direkt während der Modellierung analysiert und simuliert. Falls umgekehrt ein fachlicher Prozess auf Grund der variablen Umwelt modifiziert werden muss – z. B. wegen eines Ressourcenengpasses – so werden die hierfür notwendigen Informationen von der technischen Ebene zurück auf die Ebene der fachlichen Prozessmodellierung abgebildet und dienen dort als Entscheidungshilfe.

*Phase 2 – Prozessautomatisierung:* Änderungen von fachlichen Prozessmodellen werden handhabbar durch eine möglichst automatische Abbildung auf technische, ausführbare Prozessmodelle sowie eine späte Bindung der Fachfunktionalitäten an Ausführungskomponenten. Durch Einführung einer Virtualisierungsschicht, die es ermöglicht, ausführbare Prozesse unabhängig von der verfügbaren Infrastruktur zu formulieren, wird zusätzliche Unabhängigkeit gegenüber einer variablen Umwelt gewonnen.

*Phase 3 – Prozessausführung und -beobachtung:* Variable Ressourcen und Komponenten werden handhabbar durch Mechanismen der Lastverteilung und dynamischen Ressourcenverteilung sowie durch zeitnahe Ressourcenüberwachung und fehlertolerante Ausführung. Unterstützt wird dies durch eine flexible, z. B. Constraint-basierte Ressourcenplanung, die ein dynamisches und automatisches Nachplanen bei Änderungen der Prozesse oder der Umwelt ermöglicht.

Die folgenden Abschnitte stellen unsere bislang realisierten Arbeiten bezüglich Modellierung (Abschnitt 2.1), Automatisierung (Abschnitt 2.2) sowie Ausführung und Beobachtung (Abschnitt 2.3) von dynamischen Prozessen kurz vor.

## **2.1 Modellierung fachlicher und technischer Prozesse**

Zur Modellierung **fachlicher Prozesse** existieren im Wesentlichen zwei etablierte Modellierungsnotationen: Die *Business Process Modeling Notation* (BPMN) [Ob09] sowie die *Ereignisgesteuerten Prozessketten* (EPK) [KS92]. Aufgrund der einfacheren Struktur sowie der semantischen Nähe zu Petrinetzen – welche in dieser Arbeit der Prozessautomatisierung zugrunde liegen – basiert die Geschäftsprozessmodellierung in der vorliegenden Arbeit auf EPK. Zum Austausch von Prozessmodellen, die mit Hilfe einer Ereignisgesteuerten Prozesskette beschrieben werden, hat sich die *Event-driven Process Mar-*

*kup Language (EPML)* durchgesetzt [MN04]. Die XML-basierte Notation gestattet eine plattform- und sprachunabhängige Repräsentation von EPK.

Die Modellierung **technischer Prozesse** basiert auf High-Level-Petrinetzen gemäß ISO/IEC 15909-1 [Is04] und dem entsprechenden XML-Austauschformat *GWorkflowDL* [Ho09, Al06, Al06b, HA06], welches neben der Modellierung und Analyse von technischen Prozessen auch eine direkte Automatisierung und Überwachung der Prozesse unterstützt. Für eine formale Spezifikation der *GWorkflowDL* sei auf das XML-Schema der *GWorkflowDL* [Ho09] verwiesen.

## 2.2 Überführung von fachlichen Geschäftsprozessen in ausführbare IT-Prozesse

Abbildung 1 zeigt den gewählten Lösungsansatz für die schrittweise und zum großen Teil automatisierte Abbildung von fachlichen Prozessbeschreibungen auf die jeweils geeigneten und verfügbaren Ressourcen. Die Geschäftsprozesse werden zunächst annotiert und dann auf abstrakte technische Prozesse (gelb) abgebildet, in diesem Fall durch die Abbildung von EPK auf *GWorkflowDL*-Dokumente [Ho09]. Jeder Aktivität werden dann zunächst mit Hilfe des ResourceMatcher-Dienstes Service-Kandidaten (blau) zugeordnet, welche die entsprechende Funktionalität bereitstellen. Ein Scheduler wählt einen der Kandidaten aus (grün) und führt die Aktivität auf den entsprechenden Ressourcen aus. Ein Beispiel einer Überführung eines EPK-Modells in einen ausführbaren *GWorkflowDL*-Prozess ist in Abbildung 2 dargestellt.

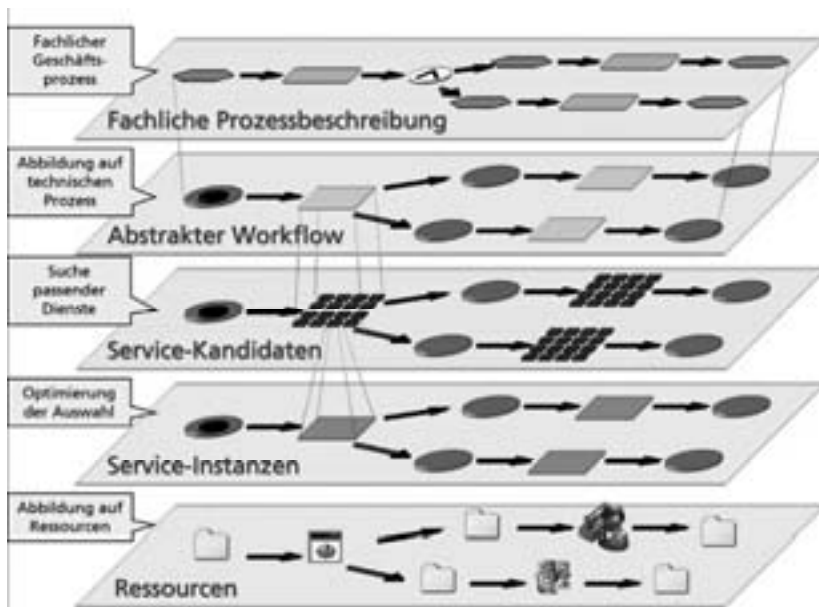


Abbildung 1: Überführung von fachlichen Prozessbeschreibungen in automatisierte Prozesse

### 2.3 Automatisierung und Beobachtung der Prozesse

Der *Grid Workflow Execution Service (GWES)* ermöglicht die Automatisierung und das interaktive Management von komplexen und dynamischen Prozessabläufen in Service-orientierten Architekturen oder Grid-Umgebungen [Ho08, Ho06, GH07, Vo08]. Durch die Marken des Petrinetzes wird nicht nur ein Prozessmuster, sondern auch der Zustand einer jeden Prozessinstanz beschrieben. Dies ermöglicht die direkte Überwachung der Prozesse sowie eine einfache Realisierung von Fehlertoleranzmechanismen.

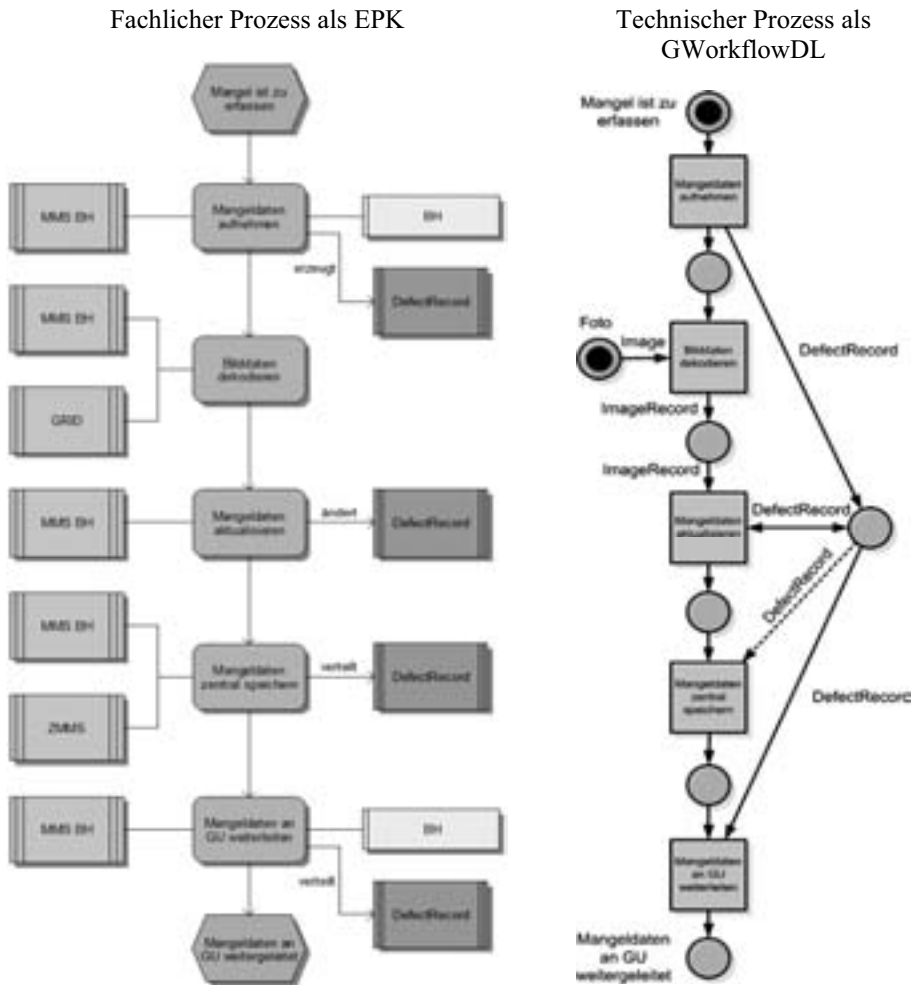


Abbildung 2: Beispiel einer Überführung eines einfachen Geschäftsprozesses aus dem Bereich Mängelmanagement im Bauwesen in einen ausführbaren Grid-Workflow

### 3 Schlussfolgerungen und Ausblick

Der Beitrag hat am Beispiel der Sprachen EPK und GWorkflowDL gezeigt, wie ein durchgängiger Ansatz von der fachlichen Beschreibung von Prozessen bis hin zu deren Ausführung unter Berücksichtigung von Dynamik, Lastverteilung und Ausfallsicherheit erreicht werden kann. Dieser Ansatz ist besonders für sich häufig ändernde Prozesse sowie für verteilte Umgebungen im Bereich des Grid- und Cloud-Computing geeignet, in denen nur partiell eine Kontrolle über die zur Ausführung notwendigen Ressourcen möglich ist. Zukünftiger Forschungsbedarf besteht insbesondere hinsichtlich einer verbesserten Werkzeugunterstützung.

### 4 Literaturverzeichnis

- [Al06] Alt M., Gorlatch S., Hoheisel A., Pohl H.-W.: Using High-Level Petri Nets for Hierarchical Grid Workflows. 2nd IEEE International Conference on e-Science and Grid Computing (e-Science 2006), Amsterdam, Netherlands, IEEE, 2006.
- [Al06b] Alt M., Hoheisel A., Pohl H.-W., Gorlatch S.: A Grid Workflow Language Using High-Level Petri Nets. PPAM05, LNCS 3911, Springer, 2006; S. 715-722
- [GH07] Gubala T.; Hoheisel A.: Highly Dynamic Workflow Orchestration for Scientific Applications. CoreGRID Technical Report Number TR-0101, CoreGRID, 2007.
- [HA06] Hoheisel A.; Alt M.: Petri Nets. In (Taylor I.J., Gannon D., Deelman E., Shields M.S. Hrsg.): Workflows for e-Science – Scientific Workflows for Grids, Springer, 2006.
- [Ho06] Hoheisel, A.: User Tools and Languages for Graph-based Grid Workflows. In: Special Issue of Concurrency and Computation: Practice and Experience, Wiley, 2006.
- [Ho08] Hoheisel, A.: Grid-Workflow-Management. In (Weisbecker, A.; Pfreundt, F.-J.; Linden, J.; Unger, S. Hrsg.): Fraunhofer Enterprise Grids – Software. Fraunhofer IRB Verlag, Stuttgart, 2008. ISBN: 978-3-8167-7804-2
- [Ho09] Hoheisel, A.: XML-Schema der Grid Workflow Description Language (GWorkflowDL) – Version 2.0. Fraunhofer FIRST, 2009; [http://www.gridworkflow.org/kwfggrid/src/xsd/-gworkflowdl\\_2\\_0.xsd](http://www.gridworkflow.org/kwfggrid/src/xsd/-gworkflowdl_2_0.xsd) [Zugriffsdatum 20.09.2009]
- [Ho09b] Hoheisel, A.; Dollmann, T.; Fellmann, M.: Überführung von EPK-Modellen in ausführbare Grid- und Cloud-Prozesse. In: Tagungsband der EPK2009, 2009
- [Is04] ISO/IEC 15909-1: High-level Petri nets – Part 1: Concepts, definitions and graphical notation. 2004.
- [KS92] Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage Ereignisgesteuerter Prozeßketten (EPK). In: Scheer A-W (Hrsg): Veröffentlichungen des Inst. für Wirtschaftsinformatik, Nr. 89, Universität des Saarlandes. 1992
- [MN04] Mendling, J.; Nüttgens, M.: Exchanging EPC Business Process Models with EPML. In (Nüttgens, M.; Mendling, J. Hrsg.) Proc. of the 1st GI Workshop XML4BPM - XML Interchange Formats for Business Process Management; Modellierung 2004, Marburg Germany, 2004, S. 61-79.
- [Oa07] OASIS: Web Services Business Process Execution Language Version 2.0. OASIS, 2007; <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html> [Zugriffsdatum 20.09.2009]
- [Ob09] Object Management Group: Business Process Modeling Notation (BPMN) – Version 1.2., 2009; <http://www.omg.org/spec/BPMN/1.2/> [Zugriffsdatum 20.09.2009]
- [Pe09] Peisl, R.: Business Process Management mit IBM WebSphere. IBM, 2009.
- [Vo08] Vossberg, M.; Hoheisel, A.; Tolxdorff, T.; Krefting, D.: A Workflow-based Approach for Fault-tolerant Medical Image Transfer in Health Grids. In: Future Generation Computer Systems, Elsevier, 2008.