Ein modellgetriebener Ansatz zur Nutzung von WS-BPEL für Scientific Workflows

G. Scherp[1] und W. Hasselbring[2]
[1]OFFIS Institut für Informatik, Escherweg 2, 26121 Oldenburg
[2]Universität Kiel, Institut für Informatik, 24118 Kiel

Abstract: Scientific Workflows haben sich zunehmend in diversen wissenschaftlichen Disziplinen etabliert, um die Modellierung komplexer Prozessierungsketten und deren Ausführung in verteilten Infrastrukturen wie Grids zu unterstützen. Mit der Einführung von Service-orientierten Architekturen im betrieblichen Kontext und deren Adoption in Grid-Infrastrukturen, haben sich Scientific Workflows und Business Workflows auf der technischen Ausführebene stark angenähert. Daher ist eine in diesem Bereich etablierte und standardisierte Workflow-Sprache wie WS-BPEL technisch gesehen auch für die Ausführung von Scientific Workflows geeignet, allerdings weniger zur fachlichen Modellierung. In diesem Dokument beschreiben wir einen modellgetriebenen Ansatz in Form eines Transformationsframeworks, um aus einer Scientific Workflow-Beschreibung auf der fachlichen Ebene automatisiert einen ausführbaren Workflow auf Basis einer Business Workflow-Sprache wie WS-BPEL auf der technischen Ebene zu generieren.

1 Einleitung

Scientific Workflows haben sich zunehmend zur Beschreibung und Ausführung komplexer Prozessketten in unterschiedlichen wissenschaftlichen Bereichen wie Chemie, Biologie und Physik etabliert. Grid-Infrastrukturen sind auf Grund der üblicherweise großen Anzahl an Ressourcen geeignete Ausführumgebungen für Scientific Workflows. Dazu wurden zahlreiche Workflow-Sprachen und -Engines sowie Werkzeuge entwickelt. In der betrieblichen Informationsverarbeitung werden ebenfalls Workflows, im Folgenden als Business Workflows bezeichnet, eingesetzt. Durch die Etablierung von Service-orientierten Architekturen (SOA) sind in einem Business Workflow genutzte Anwendungen wie beispielsweise Datenbanken durch Dienste gekapselt. Letztlich wird die Ausführung eines Business Workflow zu einer so genannten Diensteorchestrierung (engl. *service orchestration*). Die XML-basierte *Web Services Business Process Execution Language* (WS-BPEL)[1] wurde speziell für die Orchestrierung von Diensten bzw. Web Services entwickelt und hat sich als OASIS-Standard im betrieblichen Umfeld etabliert. Ausgehend von der Definition der *Open Grid Service Architecture* (OGSA)[2] wurde das SOA-Konzept in Grid-Technologien übernommen. Sämtliche Zugriffe auf Ressourcen (Rechner, Speicher) sind in einer solchen Grid-Infrastruktur durch Dienste gekapselt. Dazu wurden sowohl eigene Standards

[1] http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html
[2] http://www.globus.org/ogsa/

wie das *Web Service Resource Framework* (WSRF)[3] entwickelt, das zustandsbehaftete Web Services ermöglicht, als auch bereits bestehende Web Services-Technologien wie WS-Addressing[4] verwendet. Der WSRF-Standard wurde in Grid-Middlewares wie Globus Toolkit 4[5] und UNICORE 6[6] implementiert. Somit sind technisch gesehen sowohl Business Workflows als auch Scientific Workflows Diensteorchestierungen und WS-BPEL demnach zur Ausführung von Scientific Workflows geeignet.

2 WS-BPEL für Scientific Workflows

Der Begriff Business Workflow bzw. Workflow stammt ursprünglich aus dem betrieblichen Kontext und bezeichnet die vollständige oder teilweise technische Abbildung eines so genannten Geschäftsprozesses eines Unternehmens Für Business Workflows gibt es anerkannte Definitionen und ein Referenzmodell von der *Workflow Management Coalition*(WfMC)[7] und es wurden entsprechende Standards entwickelt und etabliert. Für Scientific Workflows hingegeben haben sich bisher weder eine einheitliche Definition noch Standards durchgesetzt. Eine zutreffende Beschreibung der Ziele von Scientific Workflows ist in [LWMB09] gegeben.

> "The main goals of scientific workflows, then, are (i) to save 'human cycles' by enabling scientists to focus on domain-specific (science) aspects of their work, rather than dealing with complex data management and software issues; and (ii) to save machine cycles by optimizing workflow execution on available resources."

Scientific Workflows werden dabei auf zwei Ebenen betrachtet. Erstens auf einer fachlichen Ebene, die die fachliche Modellierung eines Scientific Workflow durch einen Wissenschaftler fokussiert, und von technischen Details abstrahiert. Zweitens auf einer technischen Ebene, auf der ein ausführbarer Workflow vorliegt und auf den verfügbaren Ressourcen ausgeführt wird. Diese Trennung von fachlicher und technischer Ebene ist bei Business Workflows üblich. Beispielsweise kann für ein Business Workflow die fachliche Ebene durch die *Business Process Modelling Notation* (BPMN)[8] abgebildet werden und die technische Ebene durch WS-BPEL. Unterschiedliche Gründe motivieren diese Trennung. Sie ermöglicht die saubere Trennung von Fachlichkeit und IT und somit auch von unterschiedlichen Rollen innerhalb eines Unternehmens (Management und IT-Abteilung). Sie ermöglicht unterschiedliche Absktrationsebenen für den jeweiligen Zweck. Das Management will sich beispielsweise nicht mit IT-technischen Details beschäftigen, sondern betrachtet einen Business Workflow auf einer fachlichen Ebene mit höherer Abstraktion, der von der IT-Abteilung auf der technischen Ebenen mit niedrigerer Abstraktion um-

[3] http://www.oasis-open.org/committees/wsrf/
[4] http://www.w3.org/Submission/ws-addressing/
[5] http://www.globus.org/toolkit/
[6] http://www.unicore.eu
[7] http://www.wfmc.org
[8] http://www.bpmn.org/

gesetzt wird. Üblicherweise sind produktiv eingesetzte Business Workflows sehr stabil, ändern sich selten und werden häufig ausgeführt [LR99]. Die zwangsläufig entstehende Lücke zwischen der fachlichen und technischen Ebene soll beispielsweise durch den Einsatz einer SOA verringert werden, damit Änderungen auf der fachlichen Ebene flexibler und schneller durch die technische Ebene umgesetzt werden können. Diese Trennung der beiden Ebenen macht auch für Scientific Workflows Sinn. Dabei muss allerdings beachtet werden, dass es mit dem Wissenschaftler nur eine Rolle gibt. Dieser erstellt einen Scientific Workflow, der auf "Knopfdruck" ausgeführt werden soll. Scientific Workflows ändern sich häufig, teilweise nach jeder Ausführung. Eine klare Trennung motiviert durch ein Rollenmodell analog zu Business Workflows gibt es bei Scientific Workflows daher nicht. Allerdings spielt der Abstraktionsgrad eine wichtige Rolle. Ein Wissenschaftler hat in der Regel wenig bis gar keine IT-Kenntnisse und benötigt wie in [LWMB09] motiviert eine Domänen-spezifische Abstraktion eines Scientific Workflow, quasi Daten und Prozessoren als grafische Boxen dargestellt und mit Linien verknüpft. Die technische Ausführung erfolgt automatisch im Hintergrund. Die Trennung der fachlichen und technischen Ebene macht von daher Sinn, dass auf jeder Ebene Technologien eingesetzt werden können, die den jeweiligen Zweck erfüllen. Dies setzt allerdings voraus, dass es für Scientific Workflows eine automatisierte Abbildung der fachlichen zur technischen Ebene gibt. Ansätze in diese Richtung gibt es sowohl für Business Workflows als auch für Scientific Workflows und werden kurz in den verwandten Arbeiten in Abschnitt 4 behandelt.

In der Einleitung wurde bereits motiviert, dass WS-BPEL technisch gesehen für die Ausführung von Scientific Workflows geeignet ist, also der technischen Ebene zugeordnet werden kann. Da WS-BPEL ein technischer Standard ist, eignet es sich nicht direkt für die Modellierung eines Scientific Workflows auf der fachliche Ebenen. Generell enthält WS-BPEL keine direkte Fachlichkeit, es implementiert eine technische Service-Orchestrierung. Neben der technischen Eignung gibt es weitere Argumente für die Nutzung von WS-BPEL für Scientific Workflows. WS-BPEL ist ein etablierter OASIS-Standard und erfährt breite Unterstützung seitens der Industrie. Dadurch wird WS-BPEL durch zahlreiche existierende Workflow-Engines unterstützt. Des weiteren wurde WS-BPEL wurde für den Einsatz in SOA konzipiert. Infrastrukturen für Scientific Workflows wie Grids orientieren sich zunehmend am SOA-Konzept. Besteht somit die Ausführung eines Scientific Workflows nur noch aus Service-Aufrufen, so bietet sich für die Service-Orchestrierung die Nutzung einer etablierten und für diesen Zweck konzipierten Sprache an, statt evtl. bestehende Workflow-Engines aus dem Scientific Workflow-Bereich umzubauen. Zusammenfassend bietet sich mit WS-BPEL (oder anderen Business Workflow-Sprachen) eine technische Basis an, die sowohl für die Ausführung von Business Workflows als auch für die Ausführung von Scientific Workflows geeignet ist.

Zur Nutzung von WS-BPEL für die Ausführung von Scientific Workflows sind zwei technische Aspekte zu beachten. Erstens die Unterstützung des WSRF-Standards und zweitens die Unterstützung von Sicherheitsmechanismen in Grids. Die WSRF-Unterstützung betrifft zum Einen den Aufruf von WSRF-konformen Web Services. Da es sich dabei um eine bestimmte Reihenfolge von Web Service-Aufrufen handelt (Ressource erzeugen, nutzen und löschen), ist dies mit WS-BPEL umsetzbar. Wir haben dies mit den Grid-Middlewares Globus Toolkit 4 und UNICORE 6 getestet [SHG+10]. Zum Anderen

muss sich ein WS-BPEL–Prozess nach außen WSRF-konform verhalten, was prinzipiell möglich ist, aber die Komplexität eines WS-BPEL–Prozesses erheblich vergrößern würde [Ley06]. Sicherheitsaspekte sind im WS-BPEL–Standard nicht vorgesehen und müssen durch eine Workflow-Engine im Rahmen des Deployments unterstützt werden. Im Rahmen des D-Grid[9]–Projektes BIS-Grid[10][11] wird die so genannte BIS-Grid Workflow Engine[12] entwickelt, die ein "Grid-Wrapper" für eine beliebige WS-BPEL–Engine im Hintergrund darstellt. Dabei übernimmt die BIS-Grid Workflow Engine Sicherheitsaspekte des Grid sowie die WSRF-konforme Darstellung eines WS-BPEL–Prozesses. In diesem Szenario muss ein WS-BPEL–Prozess somit lediglich den Aufruf von WSRF-Web Services berücksichtigen. Im folgenden Abschnitt wird ein Transformationsframework vorgestellt, mit dem ein entsprechender WS-BPEL–Prozess generiert werden kann.

3 Transformationsframework

Unser Ansatz basiert darauf, dass ein Wissenschaftler einen Scientific Workflow auf der fachlichen Ebene modelliert, und dieser dann über ein Transformationsframework automatisch in eine technische und ausführbare Form übertragen wird, siehe Abbildung 1. Dabei soll die fachlichen Ebene mit bestehenden Scientific Workflow-Technologien und die technische Ebenen mit bestehenden Business Workflow-Technologien umgesetzt werden. Die Verbindung dazwischen soll das durch Transformationsframework abgebildet werden. Prinzipiell soll das Transformationsframework so weit wie möglich unabhängig von der verwendeten Workflow-Technologie sein. Um das Transformationsframework zu testen haben wir uns auf WS-BPEL und die BIS-Grid Workflow Engine auf der technischen Seite festgelegt, die Workflow-Technologie auf der fachlichen Ebene ist derzeit noch offen. Letztlich sind mit dem Transformationsframework folgende Ziele verbunden:

- Verknüpfung von bestehenden Scientific Workflow-Technologien zur Modellierung von Scientific Workflows (fachliche Ebene) und bestehenden Business Workflows-Technologien zur Ausführung von Scientific Workflows (technische Ebene).
- Erweiterbarkeit um weitere Workflow-Sprachen sowohl auf der fachlichen als auch auf der technischen Ebene.
- Generierung eines ausführbaren Scientific Workflow auf Basis einer Business Workflow-Sprache. Komplexe Sprachkonstrukte werden automatisch erzeugt, deren manuelle Erstellung oft mühsam und fehleranfällig ist. Dazu gehören beispielsweise auch Konstrukte zur Fehlerbehandlung, die in der Regel auf der fachlichen Ebene nicht modelliert werden (eher durch Konfiguration einstellbar).

[9] http://www.d-grid.de

[10] http://www.bisgrid.de

[11] Dieses Projekt wird mit Mitteln des Bundesministeriums für Bildung und Forschung (BMBF) unter dem Förderkennzeichen 01IG07005 gefördert und ist Teil der D-Grid–Initiative.

[12] http://bis-grid.sourceforge.net

- Nutzung von etablierten Technologien aus der modellgetriebenen Softwareentwicklung.

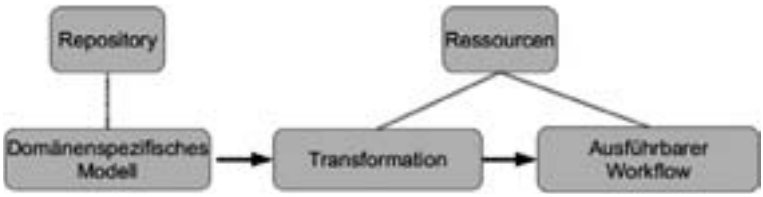

Abbildung 1: Allgemeiner Ansatz

Um eine automatische Abbildung von der fachlichen zur technischen Ebene zu gewährleisten, gehen wir davon aus, dass auf der fachlichen Ebene zumindest grundlegende technische Informationen vorhanden sind, allerdings vor dem Wissenschaftler versteckt. Dazu verwendet der Wissenschaftler zur Modellierung eines Scientific Workflows ein Repository, indem beispielsweise verfügbare Datenquellen, Datensenken und Prozessoren hinterlegt sind. Diese Elemente können über eine einfache grafische Notation verknüpft werden. Hinter jedem Element verbergen sich aber technische Informationen, beispielsweise über Daten und Dienste, die zur konkreten Abbildung auf bestimmte Ressourcen genutzt werden können. Jede technische Information zu einem Element kann dabei konfigurierbare Bereiche enthalten, die als Parameter vom Wissenschaftler modifiziert werden können, beispielsweise um gewünschte Daten zur Prozessierung einzugrenzen. Das Transformationsframework erzeugt somit Quelltext, der zum Einen die generelle Workflow-Ausführung abbildet und zum Anderen das so genannte Binden (engl. *binding*) an konkrete Ressourcen beinhaltet. Das Binden an eine Ressource bzw. einen Dienst ist ein Kernkonzept von SOA und kann statisch schon während des Transformationsprozesses für eine bestimmte Ressource durchgeführt werden, oder es wird Quelltext generiert, der eine dynamische Bindung zur Laufzeit vorsieht. In beiden Fällen muss ein Verzeichnis über verfügbare Ressourcen vorhanden sein. Bei der Auswahl einer Workflow-Technologie für die fachliche Ebene im Rahmen unserer geplanten Tests spielt die Unterstützung eines entsprechenden Repositories eine zentrale Rolle.

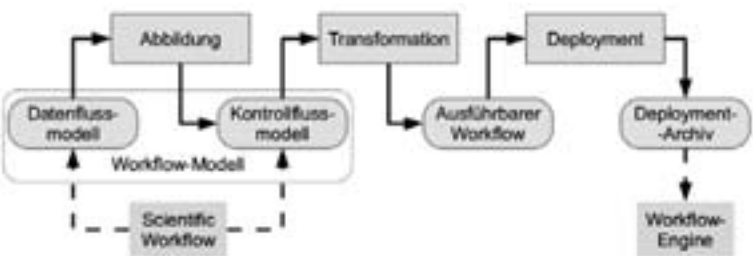

Abbildung 2: Architektur des Transformationsframeworks

Der Transformationsprozess des Transformationsframeworks ist in Abbildung 2 dargestellt. In einem ersten Schritt wird ein Scientific Workflow, der auf Basis einer bestehenden

Scientific Workflow-Sprache vorliegt, in ein internes Workflow-Modell abgebildet. Diese interne Workflow-Modell wird benötigt, um unabhängig von einer bestimmten Scientific Workflow-Sprache zu sein und kapselt sowohl den Datenfluss als auch den Kontrollfluss eines Scientific Workflows. Da Scientific Workflows in der Regel Datenfluss-orientiert sind und in geringem Umfang Kontrollfluss-Elemente zulassen, wird bei der Abbildung in das interne Workflow-Modell der Kontrollfluss nicht vollständig sein. Wenn beispielsweise ein Prozessor Eingangsdaten aus zwei Datenquellen benötigt, so ist in der Regel nicht festgelegt, in welcher Reihenfolge diese Datentransfers durchgeführt sollen oder parallel. Daher muss in dem Kontrollflussmodell die Aufrufreihenfolge von Datentransferdiensten in Form eigenständiger Aktivitäten berücksichtigt werden. Welche Aufrufreihenfolge nachher gewählt wird, kann u. a. von Performanzgründen abhängig sein, was hier nicht weiter behandelt wird. Um diese Erweiterung in dem Kontrollflussmodell auf Basis des Datenflussmodells vorzunehmen, ist der Schritt "Abbildung", siehe Abbildung 2 vorgesehen. Es ist derzeit noch offen, ob das Workflow-Modell aus zwei getrennten Modellen für Kontroll- und Datenfluss besteht, oder ein gesamte hybrides Modell verwendet wird. Des weiteren ist derzeit noch offen, welche Kontroll- und Datenflusselemente unterstützt werden.

Im nächsten Schritt wird das Workflow-Modell in einen ausführbaren Workflow transformiert, siehe "Transformation" in Abbildung 2. Da als Zielsprache Business Workflow-Sprachen genutzt werden, die üblicherweise Kontrollfluss-orientiert sind, wird der Transformationsprozess auf Basis des Kontrollflussmodells bzw. der Kontrollfluss-orientierten Sicht des Workflow-Modells durchgeführt. Dieser Schritt besteht aus zwei Unterschritten. In einem ersten Unterschritt wird der Kontrollfluss erweitert, in dem typische Aktivitäten eines Scientific Workflows um entsprechende Aktivitätsmuster erweitert werden. Die Aktivität zur Einreichung eines Jobs wird beispielsweise durch die Aktivitäten *Prepare*, *Submit*, *Wait*, und *Cleanup* erweitert. Diese Erweiterungen sind unabhängig von der Zielsprache und werden solange fortgeführt, bis keine Aktivitäten mehr erweitert werden können. Im zweiten Unterschritt wird der so erweiterte Kontrollfluss in die Zielsprache übertragen. Im Prinzip wird dabei jede Aktivität durch einen entsprechenden Quelltext-Abschnitt passend zur Ausführumgebung ersetzt. Beispielsweise kann für die *Wait*-Aktivität bei der Job-Einreichung Quelltext für ein *Push*-Modell (Benachrichtungsdienst nutzen; Globus Toolkit 4) oder *Pull*-Modell (Periodische Statusabfrage; UNICORE 6) generiert werden. Am Ende des zweiten Unterschritts liegt der Scientific Workflow in einer ausführbaren Form vor. Sämtliche im Transformationsprozess angewendeten Transformationsregeln oder Templates sind nach unterschiedlichen Aspekten von Scientific Workflows strukturiert. Dazu gehören allgemeine Aspekte wie Fehlerbehandlung und Überwachung einer Workflow-Ausführung, sowie spezifische Aspekte wie die Einreichung eines Jobs oder die Ausführung eines Datentransfers. Dadurch können Anpassungen und Erweiterungen am Transformationsprozess gezielter und einfacher vorgenommen werden.

Im letzten Schritt "Deployment", siehe Abbildung 2, wird der ausführbare Workflow in einem Deployment-Archiv verpackt. Der Aufbau eines Deployment-Archivs ist abhängig vom Hersteller einer Workflow-Engine. Anschließend wird der ausführbare Workflow mittels des Deployment-Archives über die entsprechende Workflow-Engine bereitgestellt und der Scientific Workflow kann ausgeführt werden.

Für die prototypische Implementierung Transformationsframeworks wird das Eclipse Modeling Framework (EMF)[13] verwendet. Das Workflow-Modell und der ausführbare Workflow werden durch Ecore-Modelle repräsentiert. Eine Modell-zu-Modell–Transformation (M2M) beschreibt die Transformation von einem Ecore-Modell zu einem Ecore-Modell, z. B. mit der Query View Transformation (QVT)[14] or ATLAS Transformation Language (ATL)[15]. Eine Modell-zu-Text–Transformation beschreibt die Transformation eines Ecore-Modells zu Quelltext, z. B. mit Xpand[16]. Der Schritt "Abbildung" ist somit eine M2M-Transformation, im Schritt "Transformation" ist der erste Unterschritt ebenfalls eine M2M-Transformation und der zweite Unterschritt eine M2T-Transformation.

4 Verwandte Arbeiten

Es gibt zahlreiche bestehende Lösungen wie GWES, Kepler, Taverna, Triana usw., die Scientific Workflows bereits erfolgreich umsetzen[17], aber hier nicht umfassend betrachtet werden können. Unser Ansatz soll kein Ersatz dieser Lösungen darstellen, sondern eine Ergänzung im besonderen Fokus auf der Verwendung von Business Workflow-Technologien zur Ausführung von Scientific Workflows in Service-orientieren Grids.

Die Eignung von WS-BPEL (bzw. dessen Vorgänger BPEL4WS) zur Ausführung von Scientific Workflows in Service-orientierten Grids wird durch zahlreiche Veröffentlichungen belegt, beispielsweise [WHH05, WEB+06, DFH+07]. In einigen Ansätzen wie [WHH05, DFH+07] werden dazu Erweiterungen für WS-BPEL vorgeschlagen. Damit wird ein Sprachdialekt erzeugt, wodurch eine eigene Workflow-Engine entwickelt werden und zukünftige WS-BPEL–Versionen selbst übernommen werden müssen. In unserem Ansatz wird daher ausschließlich Standard-WS-BPEL verwendet. Teilweise sind auf WS-BPEL basierende Werkzeuge entwickelt worden [WEB+06, DFH+07], die zur Modellierung bzw. Implementierung von Scientific Workflows eingesetzt werden. Diese Werkzeuge sind für Wissenschaftler ohne IT-Kenntnisse nicht geeignet. Daher setzen wir in unserem Ansatz auf eine fachliche Abstraktion mit bestehenden Scientific Workflow-Technologien.

Ansätze zur Abbildung der fachlichen zur technischen Ebene auf Basis von modellgetriebenen Technologien gibt es im Scientific Workflow-Bereich bisher kaum. In [DBG+03] wird eine eigene Code-Generierung verwendet, allerdings mit Fokus auf Workflow-Optimierung. Das in [WEB+06] beschriebene Werkzeug bietet mit so genannten *Plugins* und *Macros* eine Art Template-Mechanismus zur Code-Generierung an. Im Business Workflow-Bereich wird diese Abbildung stärker fokussiert, z. B. in [ODtHvdA06, YZZ+07]. Allerdings handelt es sich dabei um einen Top-Down-Softwarentwicklungsprozess, der schwer automatisierbar ist. Die hier betrachteten Scientific Workflows hingegen stellen auf der fachlichen Ebene eine Komposition aus definierten Elementen eines Repositories dar, woraus ein ausführbarer Workflow generiert wird.

[13] http://www.eclipse.org/modeling/emf/
[14] http://www.omg.org/spec/QVT/1.0/
[15] http://www.eclipse.org/m2m/atl/
[16] http://wiki.eclipse.org/Xpand
[17] Siehe auch http://www.gridworkflow.org/.

5 Fazit

In diesem Dokument wurde die Trennung von fachlicher und technischer Ebene bei Scientific Workflows sowie der Einsatz von etablierten und standardisierten Business Workflow-Technologien wie WS-BPEL auf der technischen Ebene motiviert. Zur Verbindung der fachlichen und technischen Ebenen wurde ein erweiterbares und auf modellgetriebenen Technologien aufbauendes Transformationsframework vorgestellt, das eine Scientific Workflow-Beschreibung auf der fachlichen Ebene in einen ausführbaren Workflow auf der technischen Ebene transformiert. Ein Prototyp ist auf Basis von EMF geplant.

Literatur

[DBG+03] Ewa Deelman, James Blythe, Yolanda Gil, Carl Kesselman, Gaurang Mehta, Karan Vahi, Kent Blackburn, Albert Lazzarini, Adam Arbree, Richard Cavanaugh und Scott Koranda. Mapping Abstract Complex Workflows onto Grid Environments. *Journal of Grid Computing*, 1(1):25–39, March 2003.

[DFH+07] Tim Dörnemann, Thomas Friese, Sergej Herdt, Ernst Juhnke und Bernd Freisleben. Grid Workflow Modelling Using Grid-Specific BPEL Extensions. 2007.

[Ley06] Frank Leymann. Choreography for the Grid: towards fitting BPEL to the resource framework. *Concurr. Comput. : Pract. Exper.*, 18(10):1201–1217, 2006.

[LR99] Frank Leymann und Dieter Roller. *Production Workflow: Concepts and Techniques*. Prentice Hall PTR, September 1999.

[LWMB09] Bertram Ludäscher, Mathias Weske, Timothy McPhillips und Shawn Bowers. Scientific Workflows: Business as Usual? In Umeshwar Dayal, Johann Eder, Jana Koehler und Hajo Reijers, Hrsg., *7th Intl. Conf. on Business Process Management (BPM)*, LNCS 5701, Ulm, Germany, 2009.

[ODtHvdA06] Chun Ouyang, Marlon Dumas, Arthur H. M. ter Hofstede und Wil M. P. van der Aalst. From BPMN Process Models to BPEL Web Services. In *ICWS '06: Proceedings of the IEEE International Conference on Web Services*, Seiten 285–292, Washington, DC, USA, 2006. IEEE Computer Society.

[SHG+10] Guido Scherp, André Höing, Stefan Gudenkauf, Wilhelm Hasselbring und Odej Kao. Using UNICORE and WS-BPEL for Scientific Workflow Execution in Grid Environments. In *Euro-Par 2009 Workshops - Parallel Processing*, Jgg. Lecture Notes in Computer Science of *Lecture Notes in Computer Science*. Springer, 2010.

[WEB+06] Bruno Wassermann, Wolfgang Emmerich, Ben Butchart, Nick Cameron, Liang chen und Jignesh Patel. *Sedna: A BPEL-Based Environment for Visual Scientific Workflow Modeling*, Kapitel 0, Seiten 427–448. Springer, 2006.

[WHH05] Yong Wang, Chunming Hu und Jinpeng Huai. A New Grid Workflow Description Language. In *SCC '05: Proceedings of the 2005 IEEE International Conference on Services Computing*, Seiten 257–260, Washington, DC, USA, 2005. IEEE.

[YZZ+07] Xiaofeng Yu, Yan Zhang, Tian Zhang, Linzhang Wang, Jianhua Zhao, Guoliang Zheng und Xuandong Li. Towards a Model Driven Approach to Automatic BPEL Generation. In *ECMDA-FA*, Seiten 204–218, 2007.