Datenmanagementpatterns in Simulationsworkflows

Peter Reimann und Holger Schwarz

Institut für Parallele und Verteilte Systeme, Universität Stuttgart
Vorname.Nachname@ipvs.uni-stuttgart.de

Abstract: Simulationsworkflows müssen oftmals große Datenmengen verarbeiten, die in einer Vielzahl proprietärer Formate vorliegen. Damit diese Daten von den im Workflow eingebundenen Programmen und Diensten verarbeitet werden können, müssen sie in passende Formate transformiert werden. Dies erhöht die Komplexität der Workflowmodellierung, welche i.d.R. durch die Wissenschaftler selbst erfolgt. Dadurch können sich diese weniger auf den Kern der eigentlichen Simulation konzentrieren. Zur Behebung dieses Defizits schlagen wir einen Ansatz vor, mit dem die Aktivitäten zur Datenbereitstellung in Simulationsabläufen abstrakt modelliert werden können. Wissenschaftler sollen keine Implementierungsdetails, sondern lediglich die Kernaspekte der Datenbereitstellung in Form von Patterns beschreiben. Die Spezifikation der Patterns soll dabei möglichst in der Sprache der mathematischen Simulationsmodelle erfolgen, mit denen Wissenschaftler vertraut sind. Eine Erweiterung des Workflowsystems bildet die Patterns automatisch auf ausführbare Workflowfragmente ab, welche die Datenbereitstellung umsetzen. Dies alles reduziert die Komplexität der Modellierung von Simulationsworkflows und erhöht die Produktivität der Wissenschaftler.

1 Einleitung

In den vergangenen Jahren haben sich im Unternehmensumfeld Workflows zur Beschreibung und Ausführung von Geschäftsprozessen durchgesetzt. Immer häufiger wird diese Technologie auch in der Wissenschaft eingesetzt, z.B. um Simulationsabläufe zu beschreiben [Gö11]. Charakteristisch für solche Simulationsabläufe sind komplexe mathematische Berechnungen sowie verschiedene Datenverwaltungs- und Datenbereitstellungsaktivitäten. Oftmals müssen große Datenmengen, die in einer Vielzahl proprietärer Formate vorliegen, aus verschiedenen Quellen verarbeitet werden. Damit diese Daten durch einen Simulationsworkflow und den von ihm eingebundenen Programmen und Diensten verarbeitet werden können, müssen sie in passende Formate transformiert werden. Dies erhöht die Komplexität der Workflowmodellierung, welche i.d.R. durch die an den Simulationsergebnissen interessierten Wissenschaftler selbst erfolgt. Wissenschaftler besitzen aber selten vertiefte Kenntnisse im Bereich der Workflowmodellierung oder der Datenverwaltung. Daher impliziert diese hohe Komplexität der Workflowmodellierung einerseits, dass sich Wissenschaftler weniger auf ihre Kernaufgaben konzentrieren können, d.h. auf die Entwicklung von mathematischen Simulationsmodellen, die Durchführung der Simulationen und die Interpretation der Ergebnisse. Andererseits birgt eine komplexe Datenverwaltung auch die Gefahr einer großen Fehlerrate in sich [Re11].

Um diese Defizite bei der Modellierung von Simulationsworkflows zu beheben, schlagen wir einen Ansatz vor, der es erlaubt, die Aktivitäten zur Datenbereitstellung in Simulationsabläufen abstrakt zu modellieren. Wissenschaftler sollen keine Implementierungsdetails, sondern lediglich die Kernaspekte der Datenbereitstellung in Form von Datenmanagementpatterns direkt beschreiben und wenige relevante Parameter festlegen müssen. Die Parametrisierung der Patterns soll dabei möglichst in der Sprache der jeweiligen Simulationsmodelle und Simulationstechnik erfolgen, mit denen Wissenschaftler besser umgehen können als mit den Sprachen zur Workflow- oder Datenmodellierung. Wenn für eine Simulation beispielsweise Daten von einem Rechner auf einen anderen transferiert werden müssen, so nutzen Wissenschaftler hierfür ein entsprechendes Datentransferpattern. Als Parametrisierung dieses Patterns werden beispielsweise der Pfad einer zu transferierenden Datei sowie das Programm, für welches die Datei bereitgestellt werden soll, festgelegt. Diese Informationen über Patterns und ihre Parametrisierungen sowie zusätzliche Metadaten werden von der Modellierungsumgebung und der Ausführungsumgebung des Simulationsworkflows genutzt, um die Patterns regelbasiert und möglichst automatisch auf ausführbare Workflowfragmente abzubilden, welche die Datenbereitstellung umsetzen. Auf Basis der Umsetzung dieses Abstraktionskonzepts in einem Simulationsworkflowsystem und auf Basis dessen Anwendung auf eine Simulation von Strukturänderungen in Knochen bewerten und diskutieren wir schließlich den vorgestellten Ansatz.

Dieser Beitrag ist wie folgt gegliedert: Zunächst gibt Abschnitt 2 einen Einblick in die Welt der Simulationsworkflows, insbesondere in die wesentlichen Anforderungen an die Datenbereitstellung in solchen Workflows. Die Datenmanagementpatterns sowie die auf ihnen aufbauende Abstraktionsunterstützung stehen im Mittelpunkt von Abschnitt 3. In Abschnitt 4 erläutern wir beispielhaft die regelbasierte Abbildung der Patterns auf ausführbare Workflowfragmente und stellen dabei die Ergebnisse der Bewertung und Diskussion des vorgestellten Ansatzes vor. Verwandte Arbeiten werden in Abschnitt 5 diskutiert, bevor der Beitrag in Abschnitt 6 mit einem Fazit und einem Ausblick abschließt.

2 Datenbereitstellung in Simulationsworkflows

Eine Abstraktionsunterstützung für die Datenbereitstellung in Simulationsworkflows muss eine Reihe spezifischer Anforderungen berücksichtigen. Wir leiten die Diskussion dieser Anforderungen mit einem Beispiel ein, das die wesentlichen Aspekte veranschaulicht.

2.1 Workflows für eine gekoppelte Simulation von Strukturänderungen in Knochen

Die Untersuchung mancher komplexer Probleme erfordert die Kopplung von Simulationsmodellen verschiedener wissenschaftlicher Anwendungsgebiete. Als Beispiel betrachten wir Untersuchungen zu Strukturänderungen in Knochen, die z.B. bei Heilungsprozessen nach Knochenbrüchen relevant sind [Kr11]. Diese Simulation koppelt Simulationsmodelle der Anwendungsgebiete *Biomechanik* und *Systembiologie* und berechnet die Struktur

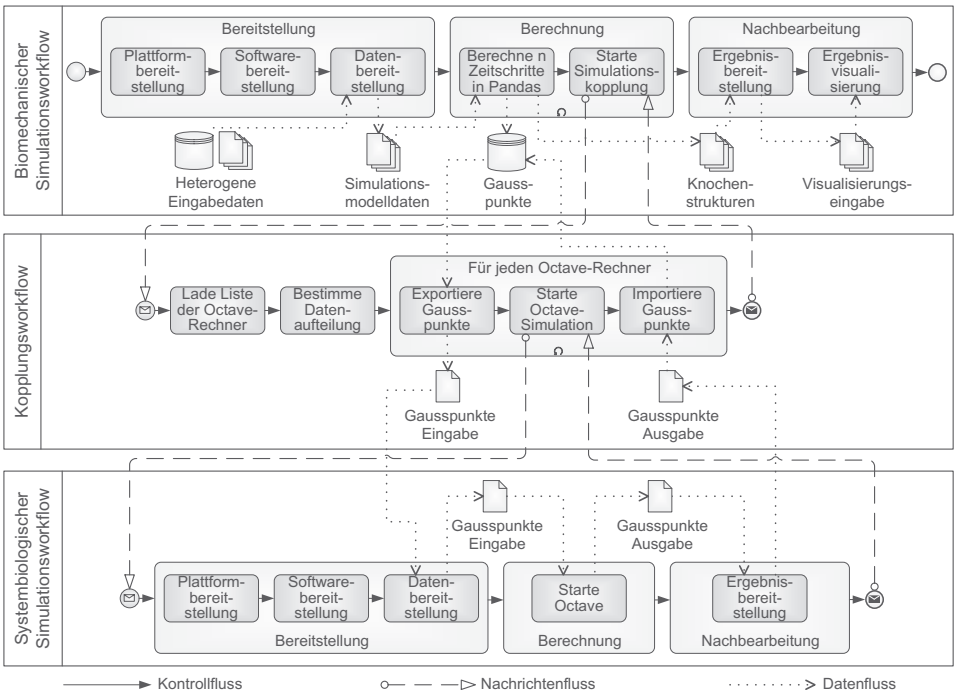

Abbildung 1: Workflows für eine gekoppelte Simulation von Strukturänderungen in Knochen

eines Knochens zeitabhängig unter einer veränderlichen Last. Das biomechanische Simulationsmodell beschreibt das Verhalten von Knochen auf einer makroskopischen Gewebeebene, wobei der Massenaustausch zwischen porösen Festkörpern und darin enthaltenen Flüssigkeiten im Vordergrund steht. Das feingranulare systembiologische Simulationsmodell bestimmt die Bildung oder den Abbau des Knochengewebes auf Basis der Interaktion von Zellen. Zu dem in Abbildung 1 gezeigten gekoppelten Simulationsprozess gehören ein *biomechanischer* und ein *systembiologischer Simulationsworkflow* sowie ein *Kopplungsworkflow*. Die biomechanische Simulation nutzt das auf der Finite-Elemente-Methode (FEM) basierende Pandas-Rahmenwerk[1] und berechnet für mehrere Zeitschritte jeweils die mechanische Belastungsverteilung im Knochengewebe. Die Belastungsverteilung des letzten berechneten Zeitschritts bildet anschließend die Eingabe für die systembiologische Simulation, die mithilfe der Rechenumgebung GNU Octave[2] umgesetzt wird. Deren Ergebnisse werden genutzt, um die Knochenkonfiguration des biomechanischen Modells anzupassen und Belastungsverteilungen für weitere Zeitschritte zu berechnen.

Beide Simulationsworkflows sind in die Phasen *Bereitstellung*, *Berechnung* und *Nachbearbeitung* aufgeteilt. In der Bereitstellungsphase werden zunächst notwendige Plattformen erzeugt, insbesondere Verzeichnisstrukturen auf den jeweiligen Rechnern, in welche die

[1] http://www.mechbau.uni-stuttgart.de/pandas/index.php
[2] http://www.gnu.org/software/octave/

nachfolgenden Aktivitäten Softwarepakete für Pandas bzw. Octave installieren sowie Eingabedaten kopieren können. Im biomechanischen Simulationsworkflow beschreiben diese Eingabedaten das entsprechende Simulationsmodell. Der Workflow muss Daten aus mehreren heterogenen Datenquellen (relationale Datenbanken, strukturierte und unstrukturierte Textdokumente) extrahieren und diese in Dateiformate transformieren, mit denen Pandas arbeiten kann. In der sequentiellen Schleife der Berechnungsphase berechnet Pandas die mechanischen Belastungsverteilungen der ersten n Zeitschritte und speichert sie in einer Datenbank. Die Daten sind in die einzelnen Zeitschritte und in mehrere Tausend oder Millionen Elemente eines FEM-Gitters strukturiert. Jedes Element enthält mehrere Gausspunkte als Stützstellen für die Interpolation der Belastungsverteilung im Knochen. Für jeden Zeitschritt und jeden Gausspunkt speichert Pandas Werte zu zehn Variablen des biomechanischen Simulationsmodells. Die systembiologische Simulation benötigt nur die Werte des letzten berechneten Zeitschritts und nur für zwei der zehn Variablen, was entsprechende Filterungen der Daten nötig macht. Da die systembiologische Simulation feingranularer und somit rechenintensiver ist, wird sie zudem parallelisiert und es findet eine Aufteilung der Daten auf mehrere Rechner und Instanzen von Octave statt.

Der *Kopplungsworkflow* steuert diese Filterung und Aufteilung der Daten. Er lädt dazu eine Liste aller verfügbaren Octave-Rechner aus einem Repository und bestimmt die Aufteilung der Daten auf diese Rechner gemäß der Vorgaben der Wissenschaftler. Die nachfolgende parallele Schleife iteriert über die Liste der Octave-Rechner. In jedem Schleifendurchlauf exportiert der Workflow die passenden Daten aus der Datenbank der Gausspunkte und speichert sie in eine CSV-Datei (Comma-separated Values) auf dem Pandas-Rechner. Anschließend startet er eine neue Instanz des systembiologischen Simulationsworkflows. Dieser stellt die notwendigen Plattformen und Softwarepakete für Octave bereit und kopiert in der Datenbereitstellung die CSV-Datei auf den jeweiligen Octave-Rechner. Anschließend startet er die Software Octave, welche mit der CSV-Datei als Eingabe die geänderten Werte der Gausspunkte in einer weiteren CSV-Datei speichert. Diese wird in der Nachbearbeitungsphase zurück auf den Pandas-Rechner kopiert. Der Kopplungsworkflow importiert die darin enthaltenen Daten in die Datenbank der Gausspunkte, womit die Knochenkonfiguration des biomechanischen Modells angepasst wird.

Der biomechanische Simulationsworkflow wiederholt diesen Prozess, bis alle Zeitschritte der Simulation betrachtet wurden. Zusätzlich zu den mechanischen Belastungsverteilungen speichert der Workflow auch die Knochenstrukturen für alle Zeitschritte in einem Pandas-spezifischen Dateiformat. In der Ergebnisbereitstellung werden diese Daten in Datenformate transformiert, mit denen das von den Wissenschaftlern gewünschte Visualisierungstool arbeiten kann, und bei Bedarf auf den Rechner dieses Tools kopiert.

2.2 Anforderungen an die Datenbereitstellung in Simulationsworkflows

Die Workflows für die Simulation von Strukturänderungen in Knochen enthalten eine Vielzahl an Datenmanagement- und Datenbereitstellungsschritten, welche Daten in vielen heterogenen Datenformaten verarbeiten. Solch eine komplexe Datenlandschaft ist typisch für Simulationen, die über verschiedene Anwendungsbereiche gekoppelt sind, da jeder An-

wendungsbereich eigene Anforderungen wie auch Lösungen für das Datenmanagement besitzt. Wissenschaftler modellieren ihre Simulationsworkflows häufig selbst und müssen dabei auch einen Großteil des Datenmanagements spezifizieren oder implementieren. Sie besitzen zwar eine hohe Expertise in ihrem Anwendungsbereich der Simulationsmodellierung, weisen aber i.d.R. eingeschränkte Fähigkeiten im Bereich der Workflowmodellierung und des Datenmanagements auf. Dies kann eine hohe Fehlerrate implizieren. Zudem verschwenden Wissenschaftler Zeit, die sie eigentlich für ihre Kernaufgaben aufbringen möchten, nämlich die Simulationen selbst. Eine essenzielle Anforderung an die Datenbereitstellung in Simulationsworkflows ist folglich eine geeignete *Abstraktionsunterstützung für die Definition von Datenbereitstellungsschritten*. Diese sollte Wissenschaftler zum Einen davon befreien, Implementierungsdetails der Datenbereitstellung zu spezifizieren. Zum Anderen soll sie Wissenschaftler dazu befähigen, mehr in der Sprache ihrer Simulationsmodelle zu arbeiten, und weniger in den Sprachen der Workflow- oder Datenmodellierung. Es muss also die Brücke zwischen der Welt der Simulationen sowie der Welt der Workflows und Daten geschlagen werden. Die hierfür erforderliche Abstraktionsunterstützung steht im Fokus dieses Beitrags. Bei deren Umsetzung müssen jedoch noch weitere Anforderungen beachtet werden. Wir stellen nachfolgend die wichtigsten drei vor:

- Die erste Anforderung ergibt sich direkt aus dem Wunsch, Simulationen sowie heterogene Daten- und Anwendungslandschaften aus verschiedenen Anwendungsbereichen zu koppeln. Hierfür muss eine Abstraktionsunterstützung hinreichend *generisch und erweiterbar* sein und alle Anwendungsbereiche unterstützen [Re11].
- Die Gesamtgröße der in Simulationsworkflows involvierten Daten kann zwischen wenigen 100 KB und einigen Terabytes liegen sowie sich während des Ablaufs einer Simulation ständig ändern. Dies führt zwangsläufig zu Anforderungen bzgl. der *Effizienz* der Datenverarbeitung und bzgl. der Unterstützung entsprechender Optimierungsmöglichkeiten für diese Datenverarbeitung [Vr07].
- Wissenschaftler führen häufig ad-hoc Änderungen an Workflows während deren Laufzeit durch [SK10]. Dazu muss eine ausreichende Überwachung der Workflowausführungen möglich sein. Ein weiterer wichtiger Aspekt ist die Sicherstellung der Wiederholbarkeit einer Simulation und der Nachvollziehbarkeit ihrer Ergebnisse, was den Begriff Provenance geprägt hat [Fr08]. Diese beiden Aspekte können unter dem Begriff *transparentes Datenmanagement* zusammengefasst werden.

3 Abstraktionsunterstützung durch Datenmanagementpatterns

In diesem Abschnitt stellen wir unseren Ansatz vor, Datenmanagementpatterns für eine Abstraktionsunterstützung der Datenbereitstellung in Simulationsworkflows zu nutzen. Um Datenmanagementpatterns in Simulationsworkflows zu identifizieren, haben wir sowohl eine Reihe von Szenarien aus der Literatur [TDG07, SR09] als auch reale Simulationsprozesse analysiert. Neben der in Abschnitt 2.1 vorgestellten Simulation gehört hierzu insbesondere das in [RK11] betrachtete Beispiel. Im Folgenden erläutern wir zunächst die

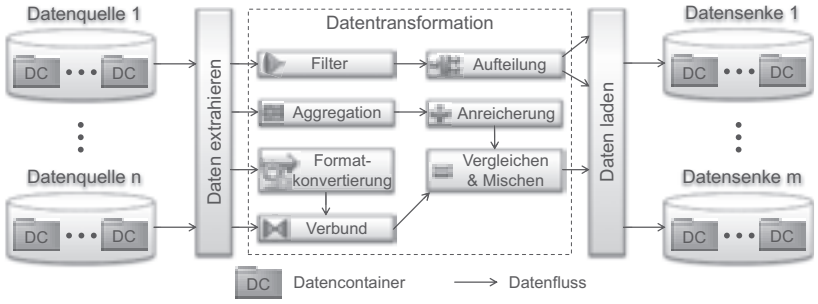

Abbildung 2: Allgemeines Datentransfer- und -transformationspattern

grundlegenden Datenmanagementpatterns. Danach gehen wir auf das zugrundeliegende SIMPL-Rahmenwerk ein (*SimTech - Information Management, Processes, and Languages*) [Re11]. Anschließend wird beschrieben, wie die vorgestellten Patterns in eine umfassendere Patternhierarchie eingegliedert sind und wie diese Hierarchie die Brücke zwischen der Welt der Simulationen und der Welt der Workflows schlagen kann.

3.1 Grundlegende Datenmanagementpatterns in Simulationsworkflows

Die allgemeine Form des *Datentransfer- und -transformationspatterns* (Abbildung 2) beschreibt den Transfer von Daten aus einer oder mehreren Datenquellen in eine oder mehrere Datensenken. Dabei können auf beiden Seiten mehrere Datencontainer angesprochen werden, die jeweils eine identifizierbare Datenmenge umfassen, z.B. eine Datenbanktabelle oder eine Datei. Zu einem solchen Pattern gehören auch ETL-Operationen, mit denen Daten aus den Datenquellen extrahiert, geeignet transformiert und in die Datensenken geladen werden [Re11, TDG07]. In den in Abschnitt 2.1 beschriebenen Workflows kommen z.B. häufig Formatkonvertierungen und Filter als ETL-Operationen zum Einsatz. Weiterhin lassen sich dort drei Varianten des Datentransfer- und -transformationspatterns unterscheiden, je nachdem ob (1) Daten von einem Datencontainer auf einen anderen übertragen werden, ob sie (2) von einem Container auf mehrere Container aufgeteilt werden oder ob sie (3) aus mehreren Container in einen Container zusammengeführt werden. Die erste Variante findet sich in den Bereitstellungs- und Nachbearbeitungsphasen der Simulationsworkflows, während der Kopplungsworkflow Daten aufteilt und wieder zusammenführt.

Das Grundprinzip, dem die *Dateniterationspatterns* folgen, ist die Iteration über einer Datenmenge S und die Ausführung eingebetteter Operationen für einzelne Teilmengen von S. Das *Parallele Dateniterationspattern* (Abbildung 3) umfasst eine Aufteilungs-, eine Operations- und eine Mischphase. Das Ziel ist die parallele Bearbeitung einer Operation auf mehreren Ressourcen. In der *Aufteilungsphase* wird ein Datenaufteilungspattern genutzt, um S in n Teilmengen $T_i \subseteq S$ aufzuteilen und diese auf die Ressourcen zu verteilen. In der *Operationsphase* dienen diese T_i als Eingabe für die anzuwendenden Operationen, die jeweils das zugehörige $T_i{}'$ als Ergebnis liefern. Das anschließende Datenmischpattern

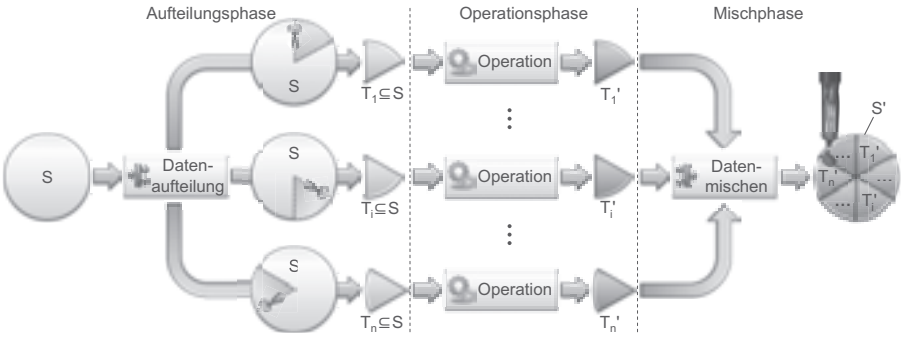

Abbildung 3: Paralleles Dateniterationspattern. S, S', T_i und T_i' entsprechen Datenmengen.

sorgt in der *Mischphase* für die Integration der Teilmengen $T_1{}'$ bis $T_n{}'$, womit sich die Ergebnisdatenmenge S' ergibt. Bei der in Abschnitt 2.1 beschriebenen Simulation ist dieses Pattern im Kopplungsworkflow zu finden. Dabei entsprechen die Daten in der Datenbank zu Gausspunkten den Datenmengen S und S'. Der Aufruf des systembiologischen Simulationsworkflows stellt die Operation dar, während die CSV-Dateien die Rolle der Teilmengen T_i bzw. $T_i{}'$ einnehmen. Die zweite Variante, das *Sequentielle Dateniterationspattern*, umfasst weder eine Parallelverarbeitung noch eine Aufteilung der Datenmenge S, sondern die Iterationsschritte werden nacheinander ausgeführt. Solch ein Pattern kann im Beispiel aus Abschnitt 2.1 sinnvoll sein, um Berechnungen sequentiell durchzuführen, falls für den gewünschten Parallelisierungsgrad zu wenig Octave-Rechner zur Verfügung stehen.

3.2 Das SIMPL-Rahmenwerk für eine Abstraktionsunterstützung

Das SIMPL-Rahmenwerk bietet eine Reihe von Abstraktionsunterstützungen für die Definition der Datenbereitstellung in Simulationsworkflows an [Re11]. Abbildung 4 zeigt, wie es die Architektur eines Simulationsworkflowsystems erweitert. Zur besseren Lesbarkeit lassen wir Komponenten der Gesamtarchitektur aus, die für die Datenbereitstellung weniger relevant sind. Dies betrifft z.B. eine Komponente für das dynamische Binden von Services oder Ressourcen [Gö11]. Die im Rahmen dieses Beitrags relevanten Hauptkomponenten sind die *Workflowmodellierumgebung*, die *Workflowausführungsumgebung*, die *regelbasierte Patterntransformationsumgebung* und der *Service Bus*. Im Folgenden erläutern wir zuerst die Datenzugriffsabstraktion, während die darauf aufbauende Abstraktionsunterstützung mittels Datenmanagementpatterns und deren regelbasierten Transformation auf ausführbare Workflows im nächsten Teilabschnitt diskutiert wird.

Die Datenzugriffsabstraktion basiert auf dem *SIMPL Core*, einer Erweiterung des Service Bus. Diese stellt Wissenschaftlern generische Operationen für den einheitlichen Zugriff auf externe Datenressourcen zur Verfügung. Hierzu gehören (1) *IssueCommand* für das Absenden von Befehlen zur Datenmanipulation oder -definition, (2) *RetrieveData* zum Laden von Daten, (3) *WriteDataBack* für das Zurückschreiben dieser Daten sowie

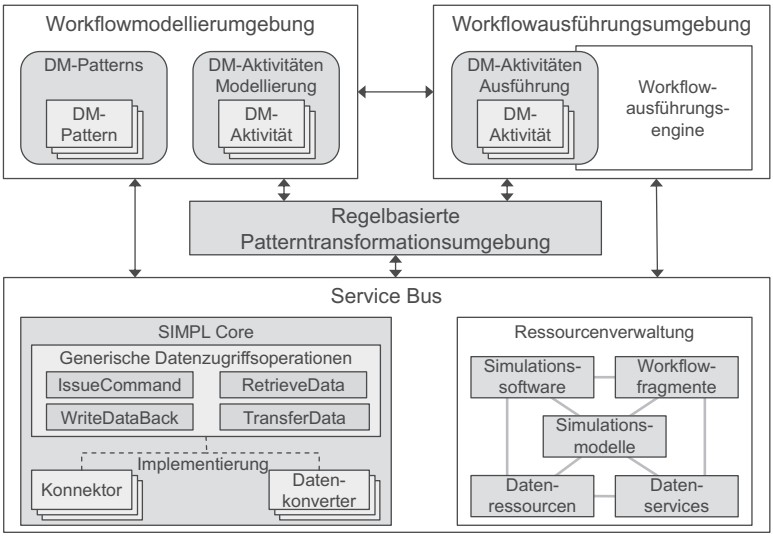

Abbildung 4: Zentrale Komponenten eines durch das SIMPL-Rahmenwerk erweiterten Simulationsworkflowsystems, vgl. [Re11], [Gö11]. Bestandteile des SIMPL-Rahmenwerks sind grau eingefärbt.

(4) *TransferData* für Datentransfers. *Konnektoren* implementieren diese Operationen für bestimmte Datenressourcen und berücksichtigen deren spezifische Zugriffsmechanismen. Für die RetrieveData- und WriteDataBack-Operationen transformieren *Datenkonverter* die Daten vom Format eines Konnektors in das der Client-Anwendung und umgekehrt. Zusätzlich erweitert SIMPL die Ressourcenverwaltung um Metadaten zur expliziten Beschreibung von *Datenressourcen*. Diese Metadaten bilden insbesondere die generischen Zugriffsoperationen auf die konkreten Zugriffsmechanismen einzelner Datenressourcen ab, indem sie u.a. jede Datenressource mit dem passenden Konnektor und Datenkonvertern verknüpfen. Damit die Funktionalität des SIMPL Core auch direkt in Workflowmodellen genutzt werden kann, bieten sowohl die Modellier- als auch die Ausführungsumgebung eine Unterstützung für Datenmanagementaktivitäten (*DM-Aktivitäten*). Die Aktivitäten entsprechen dabei sinngemäß den vier Operationen des SIMPL Core. Sie sind jeweils einer Datenressource zugeordnet, beinhalten einen Befehl in deren Befehlssprache – z.B. in SQL oder XQuery – und senden diesen Befehl bei der Ausführung der Aktivität über den SIMPL Core an die Ressource. Als Alternative können Workflowmodellierer nach wie vor Services für das Datenmanagement verwenden – sog. *Datenservices*.

3.3 Datenmanagementpatterns als Brücke zwischen Simulationen und Workflows

Trotz der erläuterten Datenzugriffsabstraktion mittels DM-Aktivitäten müssen Wissenschaftler in ihren Workflowmodellen viele Details der Datenbereitstellung spezifizieren. Verwenden Wissenschaftler Datenservices, müssen sie passende Services suchen bzw. An-

Abbildung 5: Hierarchie von Datenmanagementpatterns

forderungsbeschreibungen in einer geeigneten Sprache definieren. Bei DM-Aktivitäten müssen sie Datenmanagementoperationen wie z.B. Datentransformationen oder Datenaufteilungen sogar über die Befehlssprachen der involvierten Datenressourcen beschreiben. Da Sprachen für Anforderungsbeschreibungen und vor allem Befehlssprachen von Datenressourcen i.d.R. wenig mit den Sprachen der Simulationsmodelle zu tun haben, fällt Wissenschaftlern dies häufig schwer. Daher erweitert SIMPL die Workflowmodellierumgebung um eine weitere Komponente. Diese ermöglicht die Nutzung der in Abschnitt 3.1 beschriebenen und weiterer Datenmanagementpatterns (*DM-Patterns*) als Bausteine für Datenbereitstellungsschritte in Simulationsworkflows. Durch die Einteilung der Datenmanagementoperationen in einzelne voneinander abgrenzbare Patterns können wir für jedes Pattern die Freiheitsgrade in der Spezifikation dieser Operationen einschränken. Dies reduziert die Komplexität der Spezifikation und ermöglicht eine weiterführende Abstraktion auf Basis der Patterns. Wissenschaftler können die für sie relevanten Patterns auswählen und in ihre Workflowmodelle einfügen. Sie werden dann für jedes Pattern bei der Spezifikation der konkreten Operation unterstützt. Insbesondere müssen sie nur wenige Parameterwerte angeben, anstatt vollständige Implementierungsdetails zu spezifizieren.

Abbildung 5 ordnet Datenmanagementpatterns in einer Hierarchie an. Je höher die Hierarchieebene, desto mehr Informationen bzgl. den zugrundeliegenden Datenmanagementoperationen und Befehlssprachen werden verdichtet. Dementsprechend müssen Wissenschaftler bei der Spezifikation von Operationen über immer weniger Detailwissen verfügen. Mit diesem steigenden Abstraktionsgrad erhöht sich auch der Bezug zwischen den für Patterns anzugebenden Parameterwerten und den Sprachen der jeweiligen Simulationsmodelle, wobei der Bezug zu den Sprachen der Workflow- oder Datenmodellierung entsprechend geringer wird. Umgekehrt müssen die verdichteten Informationen auf dem Weg nach unten durch die Hierarchie wieder angereichert werden, um auf die Ebene *ausführbarer Workflowfragmente* bzw. *Datenservices* zu kommen. Letztgenannte setzen die Patterns in den Ebenen darüber um und beinhalten dabei viele Implementierungsdetails. Die nächsthöhere Ebene umfasst die in Abschnitt 3.1 beschriebenen *grundlegenden Datenmanagementpatterns*. Die Ebene der *zusammengesetzten Datenmanagementpatterns* nutzt die grundlegenden Patterns als Basis und schafft einen höheren Abstraktionsgrad. Hier können z.B. mehrere Datentransfer- und -transformationspatterns, die das gleiche Ziel für den Datentransfer

definieren, in einem *allgemeinen Datenbereitstellungspattern* zusammengefasst werden. Den stärksten Bezug zu Simulationen und damit den für Wissenschaftler höchsten Abstraktionsgrad stellt die Ebene der *simulationsorientierten Patterns* her. Als Beispiel sei ein Pattern für die Kopplung verschiedener Simulationsmodelle genannt. Wissenschaftler können die Kopplung mit diesem Pattern vollständig über Begriffe spezifizieren, die ihnen aus ihren Simulationsmodellen geläufig sind. Im betrachteten Beispiel aus Abschnitt 2.1 geben sie im Wesentlichen die Abhängigkeiten zwischen den verschiedenen Variablen der beiden Simulationsmodelle sowie den relevanten Zeitschritt an. Weiterhin spezifizieren sie abstrakt, dass die Daten gleichmäßig nach Gausspunkten aufgeteilt werden sollen.

Die *regelbasierte Patterntransformationsumgebung* des SIMPL-Rahmenwerks enthält eine erweiterbare Menge von Abbildungsregeln, welche von Wissenschaftlern parametrisierte Patterns Schritt für Schritt nach unten durch die einzelnen Ebenen der Hierarchie abbilden. Hierbei werden Regeln so lange und ggf. rekursiv angewandt, bis alle Patterns in einem Workflow schließlich durch ausführbare Workflowfragmente oder Datenservices ersetzt wurden. Die Regeln nutzen dabei Metadaten bzgl. *Simulationssoftware*, *Workflowfragmenten*, *Simulationsmodellen*, *Datenressourcen* und *Datenservices* sowie Abhängigkeiten zwischen diesen Metadaten, um die für die Abbildung von Patterns notwendige Informationsanreicherung umzusetzen. Auf diese Weise bildet eine Regel z.B. das simulationsorientierte Pattern für die Spezifikation einer Simulationsmodellkopplung auf ein Paralleles Dateniterationspattern ab. Die Parametrisierung dieses Patterns sowie dessen Abbildung auf einen ausführbaren Workflow diskutieren wir in Abschnitt 4.1.

4 Bewertung und Diskussion der Abstraktionsunterstützung

Um die vorgestellte Abstraktionsunterstützung bewerten zu können, entwickelten wir einen Prototypen des SIMPL-Rahmenwerks. Dieser Prototyp nutzt die Workflowsprache Business Process Execution Language (BPEL) [JE07], das Workflowmodelliertool Eclipse BPEL Designer[3] Version 0.8.0 und die Workflowausführungsengine Apache Orchestration Director Engine[4] (ODE) Version 1.3.5. Im nächsten Teilabschnitt illustrieren wir den Einsatz des Parallelen Dateniterationspatterns im Kopplungsworkflow aus Abschnitt 2.1 und wie das resultierende ausführbare Workflowmodell aussieht. Anschließend bewerten wir, inwieweit dieses Pattern eine für Wissenschaftler geeignete Abstraktionsunterstützung zur Definition des Kopplungsworkflows darstellt. Schließlich diskutieren wir unseren Ansatz bzgl. der in Abschnitt 2.2 beschriebenen Anforderungen.

4.1 Umsetzung für die Simulation von Strukturänderungen in Knochen

Das Dateniterationspattern ersetzt im Kopplungsworkflow alle Aktivitäten zwischen dem Eingang der Nachricht vom biomechanischen Simulationsworkflow und dem Rücksenden

[3] http://www.eclipse.org/bpel/
[4] http://ode.apache.org/

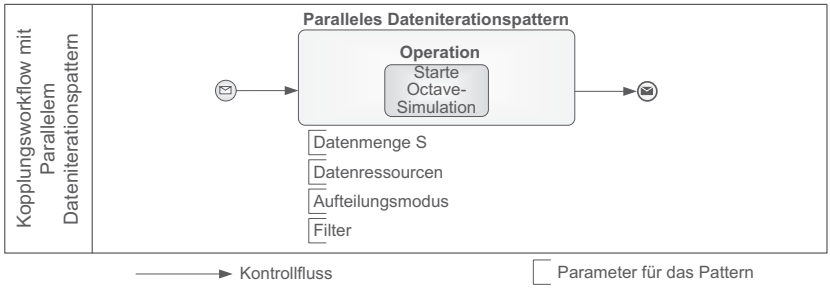

Abbildung 6: Einsatz des Parallelen Dateniterationspatterns im Kopplungsworkflow aus Abbildung 1

der Antwortnachricht (siehe Abbildung 6). Der Aufruf des systembiologischen Simulationsworkflows ist die in das Pattern eingebettete Operation. Mithilfe des Patterns reduziert sich die Anzahl der für Wissenschaftler sichtbaren Aktivitäten und der zu spezifizierenden Parameter, was bei der Definition des Workflowmodells eine erhebliche Erleichterung darstellt. Der Wert des ersten Parameters identifiziert die *Datenmenge* S, über die iteriert werden soll. Der Wissenschaftler gibt hier abstrakt die mechanische Belastungsverteilung im Knochen an, welche bei den Metadaten zum biomechanischen Simulationsmodell als Ausgabedaten registriert ist. Der zweite Parameter bestimmt die *Datenressourcen*, auf welche die Datenmenge S verteilt werden soll. Hier gibt der Wissenschaftler mithilfe von Metadaten zu Services eine Referenz auf einen geeigneten Repositoryservice an, der eine Liste der verfügbaren Octave-Rechner liefert. Mithilfe der Metadaten zum biomechanischen Simulationsmodell kann der Wissenschaftler alle weiteren Parameter des Patterns über Begriffe festlegen, die ihm aus diesem Simulationsmodell geläufig sind. Beim *Aufteilungsmodus* gibt er an, dass die Datenmenge S gleichverteilt nach Gausspunkten aufgeteilt werden soll. Der Parameter *Filter* ermöglicht die Einbindung weiterer, vor der Datenaufteilung durchzuführender Filteroperationen für S. Hier definiert der Wissenschaftler abstrakt die beiden in Abschnitt 2.1 beschriebenen Filter: einen nach dem letzten berechneten Zeitschritt und einen nach den relevanten Variablen des biomechanischen Simulationsmodells.

Über die Anwendung von Abbildungsregeln entsteht das in Abbildung 7 dargestellte ausführbare Workflowmodell. Nachdem der Workflow über den Repositoryservice die Liste der verfügbaren Octave-Rechner geladen hat, holt er sich über eine SIMPL RetrieveData-Aktivität die ID des letzten berechneten Zeitschritts aus der Datenbanktabelle zu Gausspunkten. Dazu setzt er eine SQL SELECT-Anweisung ab, die eine Aggregatfunktion für die maximale Zeitschritt-ID beinhaltet. Da in der Tabelle zu Gausspunkten Daten für mehrere Simulationen gespeichert sein können, ist zusätzlich ein Filterprädikat bzgl. der aktuellen Simulations-ID erforderlich. Die nächste RetrieveData-Aktivität speichert die Anzahl der relevanten Gausspunkte in eine Workflow-Variable. Die SELECT-Anweisung beinhaltet eine entsprechende Aggregatfunktion sowie Filterprädikate nach der Simulations-ID und nach dem Zeitschritt. Anschließend bestimmt ein XPath-Ausdruck in einer BPEL Assign-Aktivität die Anzahl der Gausspunkte pro Octave-Rechner. In der Aufteilungsphase der parallelen Dateniteration realisiert eine IssueCommand-Aktivität für jeden Octave-Rechner den Export der Daten aus der Datenbanktabelle in eine CSV-Datei.

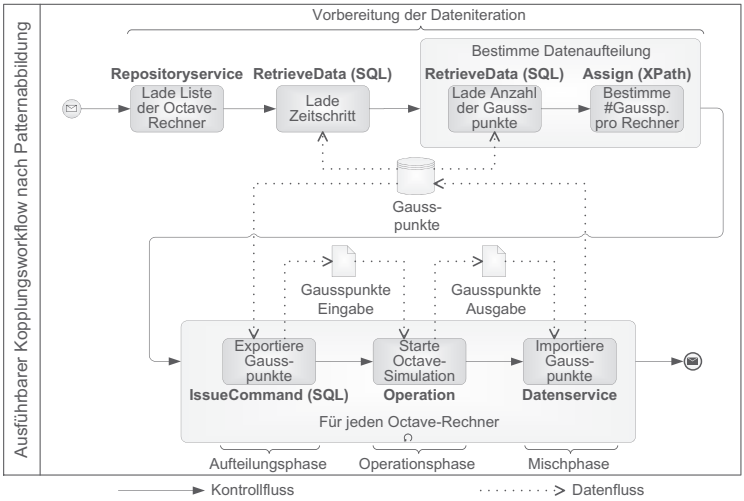

Abbildung 7: Ausführbarer Kopplungsworkflow nach der regelbasierten Abbildung von Patterns

Die eingebettete SQL-Anweisung beinhaltet die Projektionen auf die relevanten Variablen des biomechanischen Simulationsmodells, zwei Filterprädikate nach dem Zeitschritt und der Simulations-ID sowie LIMIT- und OFFSET-Ausdrücke für die Extraktion der richtigen Gausspunkte. Anschließend startet der Kopplungsworkflow den systembiologischen Simulationsworkflow. Sobald dieser beendet ist, nutzt der Kopplungsworkflow einen proprietären Datenservice, um die resultierende CSV-Datei in die Datenbank zu portieren.

4.2 Bewertung der Abstraktionsunterstützung

Modellieren Wissenschaftler direkt ausführbare Workflowmodelle wie den in Abbildung 7 gezeigten Kopplungsworkflow müssen sie auch alle in Abschnitt 4.1 beschriebenen Details der einzelnen Workflowaktivitäten, Serviceaufrufe, SQL-Anweisungen und XPath-Ausdrücke festlegen. Dies würde einen großen, für die Wissenschaftler nicht akzeptablen Aufwand darstellen. Die Modellierung mithilfe des Parallelen Dateniterationspatterns stellt für Wissenschaftler eine erhebliche Vereinfachung dar, da sie solche Implementierungsdetails nicht explizit definieren müssen. Insbesondere müssen sie deutlich weniger Workflowaktivitäten modellieren und können den Großteil des Datenmanagements über Begriffe aus ihren Simulationsmodellen und damit in einem hohen Abstraktionsgrad definieren. Die Hierarchie von Datenmanagementpatterns, die Umformungsregeln und die Metadaten der SIMPL Ressourcenverwaltung schlagen zudem die Brücke zwischen der Welt der Simulationen – dem Pattern – sowie der Welt der Workflows und Daten – dem ausführbaren Workflowmodell. Insgesamt reduziert unser Ansatz die Komplexität in der Workflowmodellierung deutlich. Dadurch sparen Wissenschaftler Zeit und können sich besser auf ihre Kernprobleme konzentrieren, nämlich die eigentlichen Simulationen.

Der auf einer Hierarchie und auf Abbildungsregeln basierende Ansatz ermöglicht zudem die Trennung der Aufgaben entsprechend der Kenntnisse verschiedener Personengruppen. So nutzen Wissenschaftler ihre Kenntnisse im Bereich der Simulationsmodellierung, um Patterns der höchsten Hierarchieebene zu parametrisieren. IT-Experten entwickeln die ausführbaren Workflowfragmente und Services der untersten Ebene und die für diese Ebene genutzten Abbildungsregeln sowie für deren Anwendung benötigte Metadaten. Für die Hierarchieebenen dazwischen können Workflowfragmente, Abbildungsregeln und Metadaten von Experten der Schnittstellen zwischen Simulation und IT entwickelt werden. Dabei dienen die voneinander abgrenzbaren Patterns auch als Mittel, um die jeweils zu erfüllenden Anforderungen zwischen diesen Personengruppen zu kommunizieren.

4.3 Diskussion bezüglich der Anforderungen

Die Anforderung, dass die Abstraktionsunterstützung *generisch in verschiedenen Anwendungsbereichen und Problemfeldern* eingesetzt werden kann, wird in unserem Ansatz im Wesentlichen durch die Wahl der generischen Patterns in der in Abbildung 5 dargestellten Hierarchie unterstützt. Diese Patterns und deren Modellierkonstrukte bzw. Parameter können unabhängig vom Problem oder dem Anwendungsgebiet verwendet werden. Die einzelnen von den Wissenschaftlern definierten Parameterwerte sowie die Abbildungsregeln und die ausführbaren Workflowfragmente bzw. Services berücksichtigen die problem- oder anwendungsgebietspezifischen Aspekte. Zudem ermöglicht die Trennung zwischen Patterns und deren Umsetzung in diesem regelbasierten Ansatz die Erweiterung um weitere problemspezifische Abbildungsregeln und Workflowfragmente bzw. Services.

Der regelbasierte Ansatz zur Abbildung von Patterns auf ausführbare Workflowmodelle ermöglicht die nahtlose Integration entsprechender regelbasierter Optimierungsentscheidungen, wie sie z.B. bei Techniken zur Restrukturierung und Optimierung von Workflowmodellen verwendet werden [Vr07]. Damit kann die *Effizienz der Datenverarbeitung* in Simulationsworkflows erhöht werden. Als Beispiel betrachten wir ein Datentransfer- und -transformationspattern, das auf einen Workflowschritt für eine Datenformatkonvertierung und einen Schritt für den eigentlichen Datentransfer aufgeteilt wird. Reduziert die Datenformatkonvertierung die Datengröße, ist es i.d.R. sinnvoll, sie vor dem Datentransfer auszuführen und umgekehrt. Außerdem können die Parametrisierungen der Patterns um Beschreibungen nichtfunktionaler Anforderungen, z.B. bzgl. der Qualität von Daten [Re12], ergänzt und diese in den Regeln als Optimierungsentscheidungen berücksichtigt werden.

Während unser Ansatz die Anzahl und Komplexität der für Wissenschaftler sichtbaren Workflowaktivitäten reduziert, kann dies zu einem Problem bzgl. des *transparenten Datenmanagements* führen. Die Workflowausführungsumgebung kennt ausschließlich die durch die regelbasierte Abbildung von Patterns entstehenden komplexeren Workflowmodelle. Damit ist die Korrelation für die in der Modellierumgebung sichtbaren Patterns und die in der Ausführungsumgebung gesammelten Audit- bzw. Provenance-Informationen nicht mehr per se gegeben. Damit Wissenschaftler dennoch Workflowausführungen überwachen bzw. Simulationsergebnisse nachvollziehen können, müssen Ausführungsumgebungen erweitert werden und diese Informationen für die Patterns aggregieren.

5 Verwandte Arbeiten

Wir haben in diesem Beitrag eine auf Patterns basierende Abstraktionsunterstützung für die Datenbereitstellung in Simulationsworkflows vorgestellt. Dementsprechend gehen wir in diesem Abschnitt auf verwandte Arbeiten in den Bereichen Workflowsysteme für wissenschaftliche Prozesse und Workflow-Patterns ein. Systeme wie das *Scientific Data Management Center* sowie das dazugehörige Workflowsystem Kepler ermöglichen ebenfalls die Definition und Ausführung wissenschaftlicher Prozesse [Sh07, Lu06]. Die beiden Systeme betrachten aber Prozesse zur Analyse von Daten, die von Simulationen oder Experimenten erzeugt wurden. Im Gegensatz zu unserem Ansatz beschäftigen sie sich nicht mit Simulationsworkflows als Vorstufe solcher Datenanalysen und vor allem nicht mit einer patternbasierten Abstraktionsunterstützung für die Datenbereitstellung in Simulationsworkflows. Das System Microsoft Trident ist hingegen universell für alle Arten von wissenschaftlichen Prozessen und damit auch für Simulationsworkflows einsetzbar [Ba08]. Allerdings fehlt auch hier der Bezug zu einer patternbasierten Abstraktionsunterstützung.

Russel et. al. beschreiben allgemeine Datenpatterns in Workflows [Ru05]. Allerdings betrachten sie in erster Linie Patterns, die typisch für Geschäftsprozesse sind, und nicht für die Datenbereitstellung in Simulationsworkflows. Es handelt sich um sehr feingranulare Patterns, die vor allem bei der Evaluation verschiedener Workflowsprachen und Workflowsysteme als Bewertungsgrundlage dienen, inwieweit diese die Patterns unterstützen. Z.B. werden die Fragen gestellt, ob Workflowaktivitäten bzw. Workflowinstanzen Daten untereinander per Wert oder per Referenz übertragen können. Bezogen auf unseren Ansatz klassifizieren diese feingranularen Patterns eher Implementierungsdetails in Workflowfragmenten auf der untersten Ebene der in Abbildung 5 dargestellten Hierarchie von Datenmangementpatterns. Sie sind also nicht für eine Abstraktionsunterstützung angedacht.

6 Fazit und Ausblick

In diesem Beitrag haben wir einen generischen Ansatz vorgestellt, mit dem Wissenschaftler die Datenbereitstellung in Simulationsworkflows abstrakt modellieren können. Kern dieses Ansatzes bildet eine Hierarchie von Datenmanagementpatterns. Das Workflowsystem bildet Parametrisierungen dieser Patterns über Abbildungsregeln automatisch auf ausführbare Workflowfragmente ab. Über die prototypische Realisierung dieses patternbasierten Ansatzes haben wir gezeigt, dass Wissenschaftler deutlich weniger Workflowschritte wie auch Implementierungsdetails der Datenbereitstellung definieren müssen. Darüber hinaus können sie die Parameterwerte eher in den Sprachen der jeweiligen Simulationsmodelle angeben, mit denen sie besser umgehen können als mit den Sprachen zur Workflow- oder Datenmodellierung. Dies reduziert die Komplexität der Modellierung von Simulationsworkflows, und Wissenschaftler können sich wieder verstärkt auf die eigentliche Simulationsproblematik konzentrieren. Als nächsten Schritt werden wir unseren Ansatz in weiteren Beispielen für Simulationsworkflows einsetzen, um dessen universelle Einsetzbarkeit genauer zu evaluieren. Weiterhin werden wir Integrationsmöglichkeiten von Optimierungsentscheidungen für eine effizientere Datenverarbeitung untersuchen.

Danksagung: Die Autoren danken der Deutschen Forschungsgemeinschaft für die Förderung des Projekts im Rahmen des Exzellenzclusters Simulation Technology. Weiterhin danken wir Michael Reiter und Christoph Stach für ihre hilfreichen Korrekturvorschläge sowie Henrik Pietranek für die Umsetzung des Prototyps im Rahmen seiner Diplomarbeit.

Literatur

[Ba08] R. Barga et al. The Trident Scientific Workflow Workbench. In *Tagungsband der 4. International Conference on e-Science*, Indianapolis, IN, 2008.

[Fr08] J. Freire et al. Provenance for Computational Tasks: A Survey. *Computing in Science and Engineering*, 10(3), 2008.

[Gö11] K. Görlach et al. Conventional Workflow Technology for Scientific Simulation. In *Guide to e-Science*, Kapitel 11. Springer, London, UK, 2011.

[JE07] D. Jordan und J. Evdemon. *Web Services Business Process Execution Language Version 2.0*. Organization for the Advancement of Structured Information Standards, 2007.

[Kr11] R. Krause et al. Bone Remodelling: A Combined Biomechanical and Systems-Biological Challenge. *Applied Mathematics and Mechanics*, 11(1), 2011.

[Lu06] B. Ludäscher et al. Scientific Workflow Management and the Kepler System. *Concurrency and Computation: Practice and Experience*, 18(10), 2006.

[Re11] P. Reimann et al. SIMPL - A Framework for Accessing External Data in Simulation Workflows. In *Gesellschaft für Informatik (Hrsg.): Datenbanksysteme für Business, Technologie und Web*, Kaiserslautern, Deutschland, 2011.

[Re12] M. Reiter et al. Quality-of-Data-Driven Simulation Workflows. In *Tagungsband der 8. International Conference on e-Science*, Chicago, IL, 2012.

[RK11] J. B. Rommel und J. Kästner. The Fragmentation-Recombination Mechanism of the Enzyme Glutamate Mutase Studied by QM/MM Simulations. *Journal of the American Chemical Society*, 26(133), 2011.

[Ru05] N. Russel et al. Workflow Data Patterns: Identification, Representation and Tool Support. In *Tagungsband der 24. International Conference on Conceptual Modeling (ER 2005)*, Klagenfurt, Österreich, 2005.

[Sh07] A. Shoshani et al. SDM Center Technologies for Accelerating Scientific Discoveries. *Journal of Physics: Conference Series (SciDAC 2007)*, 78(1), 2007.

[SK10] M. Sonntag und D. Karastoyanova. Next Generation Interactive Scientific Experimenting Based on the Workflow Technology. In *Tagungsband der 21. International Conference on Modelling and Simulation (MS 2010)*, Prag, Tschechische Republik, 2010.

[SR09] A. Shoshani und D. Rotem. *Scientific Data Management: Challenges, Technology, and Deployment*. Computational Science Series. Chapman & Hall, 2009.

[TDG07] I. Taylor, E. Deelman und D. Gannon. *Workflows for e-Science - Scientific Workflows for Grids*. Springer, London, UK, 2007.

[Vr07] M. Vrhovnik et al. An Approach to Optimize Data Processing in Business Processes. In *Tagungsband der 33. International Conference on Very Large Data Bases (VLDB 2007)*, Wien, Östereich, 2007.