

D-Grid



InGrid

Innovative Grid Entwicklungen für ingenieurwissenschaftliche Anwendungen

AP 3.2: Unterstützung ingenieurspezifischer Workflows

D 3.2.1

Anforderungen, Workflow-Spezifikation,
Modellierungsmechanismus

Projekt finanziert durch das BMBF unter Vertrags Nr.	
D-Grid intern	
IN	Internal

Autoren

Thomas Friese
Christian Schridde
Matthew Smith
Thomas Barth
Tim Dörneman

Organisation

Siemens
Universität Marburg
Universität Marburg
Universität Siegen
Universität Marburg

Internal Reviewer**Version history:**

Version	Date
0.1	10.05.2006
0.2	29.05.2006
0.3	02.08.2006
0.4	07.08.2006

Table of Contents

Übersicht	Fehler! Textmarke nicht definiert.
1. Workflow Spezifikationen	4
1.1 Workflow Design	4
1.2 Workflow Scheduling	5
1.3 Data Movement	6
1.4 Taxonomy for Grid Workflow Systems	7
2. Anforderungen	8
2.1 Das GridForEach Konstrukt	9
2.2 Workflow Editor	10
3. Modellierungsmechanismus	11

1. Workflow Spezifikationen

Es wurde eine Taxonomy für Charakteristiken eines Workflows erstellt. Hierbei wurde besonders darauf Wert gelegt, die ingenieursspezifischen Eigenschaften herauszuarbeiten. Durch die Arbeitsgruppe an der Universität Siegen wurden die ingenieurspezifischen Anforderungen an Workflows zusammengefasst. Diese Anforderungen wurden anhand analysierter Prozesse in den Bereichen Gießtechnik (→ AP 2.2) und Umformtechnik (→ AP 2.4) ermittelt. Weitere Erkenntnisse konnten aus der Analyse von Prozessen im AP 3.1 in diese Anforderungen einfließen. Die Erarbeitung ergab im Wesentlichen drei Hauptmerkmale um einen Workflow zu charakterisieren

- *Workflow Design*
- *Workflow Scheduling*
- *Data Movement*

Im Folgenden werden die drei Punkte näher beschrieben, worauf im Anschluss eine Taxonomy folgt, die die existierenden Workflow Systeme anhand der genannten Punkte klassifiziert.

1.1 Workflow Design

Workflow Design bestimmt das Aussehen eines Workflows und die Art und Weise wie einzelne Komponenten zusammengefügt werden. Ein Workflow symbolisiert die Reihenfolge einzelner Teilschritte durch gerichtete Verbindungslinien. Durch zyklische Abhängigkeiten können so Iterationen simuliert werden, wodurch sich die Struktur eines Workflows in die zwei Klassen, DAG (directed acyclic graph) and Non-DAG einordnen lässt.

Des Weiteren kann die Spezifikation eines Workflows Abstract oder Concret erfolgen, je nachdem wie allgemein gültig der entstehende Ablauf kreiert werden soll. Abbildung 1 zeigt den entstandenen Taxonomy-Baum zu diesem Punkt.

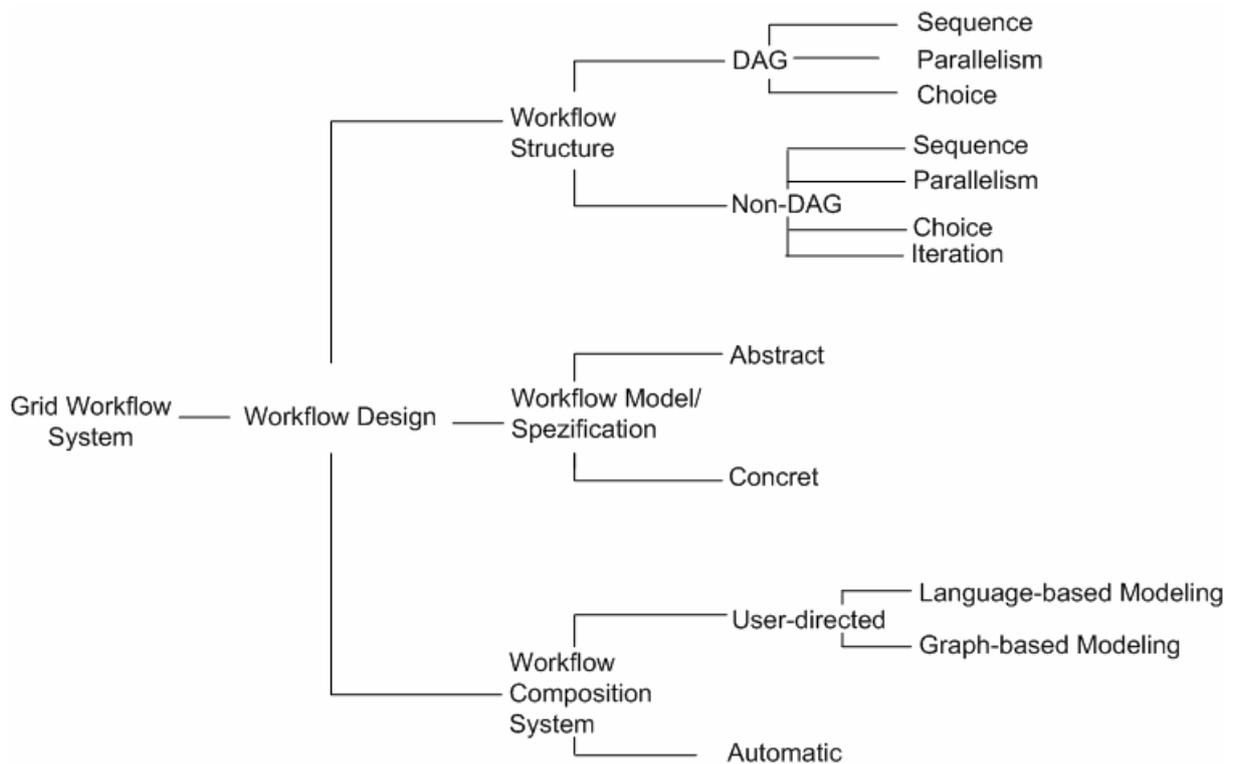


Abbildung 1 – Workflow Design

1.2 Workflow Scheduling

Workflow Scheduling bestimmt und verwaltet die Ausführung von Workflow-Prozessen auf verteilten Ressourcen. Die Architektur der Scheduling-Infrastruktur lässt sich kategorisieren in Central, Hierarchical and Decentralized, je nachdem welches Scheduling-System zum Einsatz kommt. *Decision Making* bestimmt die Art und Weise nach der die Workflow Prozesse auf Ressourcen umgesetzt werden. Hierbei kann man nach lokalen Kriterien oder den Workflow als Ganzes betrachten, um die optimale Zuteilung zu finden. *Planning Scheme* bestimmt auf welche Art ein abstrakter Workflow in konkrete Prozesse umgesetzt wird. Es können entweder vorher statische Regeln definiert werden oder dynamisch, während der Laufzeit, entsprechende Regeln befolgt werden. Ein wichtiger Punkt ist *Scheduling Strategies*. Hier gibt es viele heuristische Ansätze die versuchen verschiedene QoS-Anforderungen zu erfüllen. Ist ein Workflow Performance getrieben, so versucht er die gesamte Ausführungszeit gering zu halten und den anfallenden Overhead zu minimieren. Market getriebene Workflows hingegen, sind darauf programmiert „kostengünstig“ zu arbeiten. In diesem Szenario muss jede allozierte Ressource elektronisch erworben werden und

Rechenzeit wird abgerechnet. Zum Dritten gibt es Workflows die ihre Ressourcen nach Vertrauensmetriken auswählen. Durch numerische Werte beschrieben, werden die vertrauenswürdigsten Anbieter/Ressourcen für eine Berechnung gewählt. Abbildung 2 zeigt die entsprechende Taxonomy.

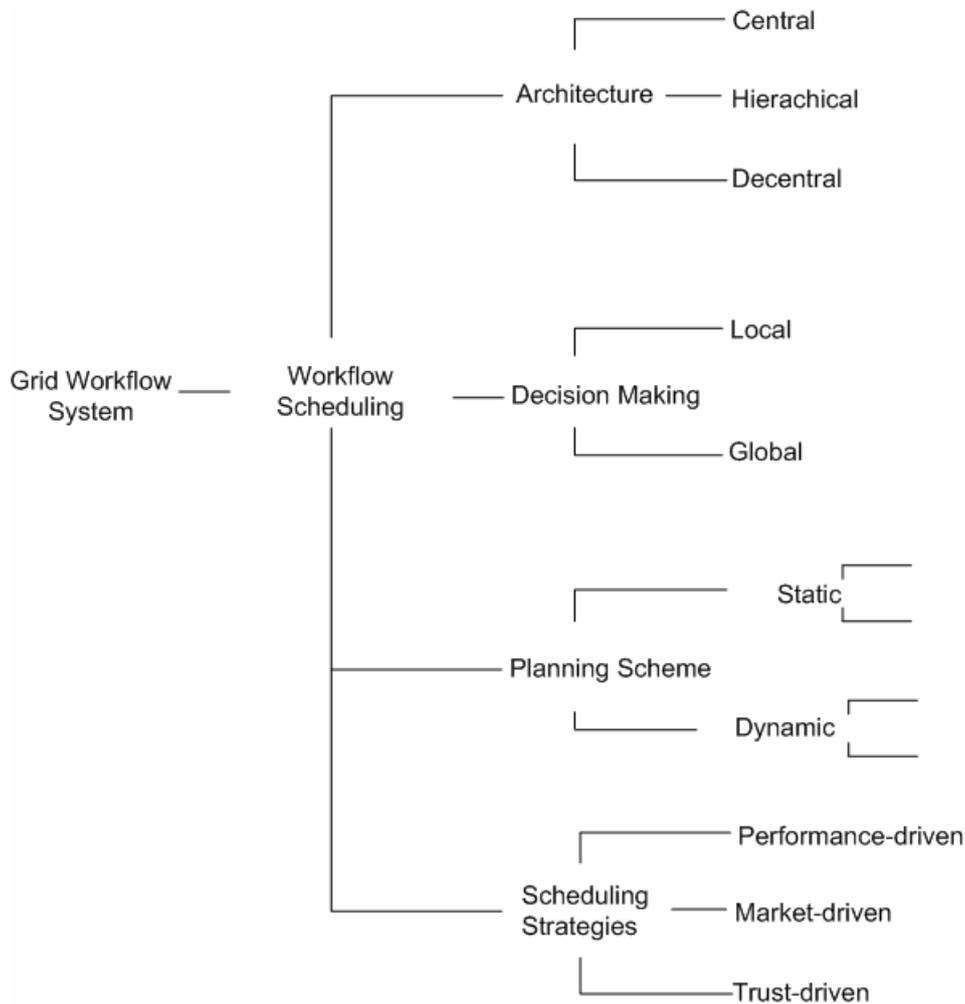


Abbildung 2 – Workflow Scheduling

1.3 Data Movement

Data Movement bestimmt die Art und Weise wie einzelne Workflow Prozesse ihre Daten austauschen und übergeben. Hierbei kann Feedback vom Benutzer erwünscht oder notwendig worunter *user-directed* Workflows fallen. In vielen Fällen lenkt jedoch das System selbstständig die Daten von Prozess zu Prozess. Dabei kann man die Datenübertragungsstruktur wie gewohnt in Centralized, Mediated und Peer-to-Peer einteilen. Abbildung 3 symbolisiert die Taxonomy.

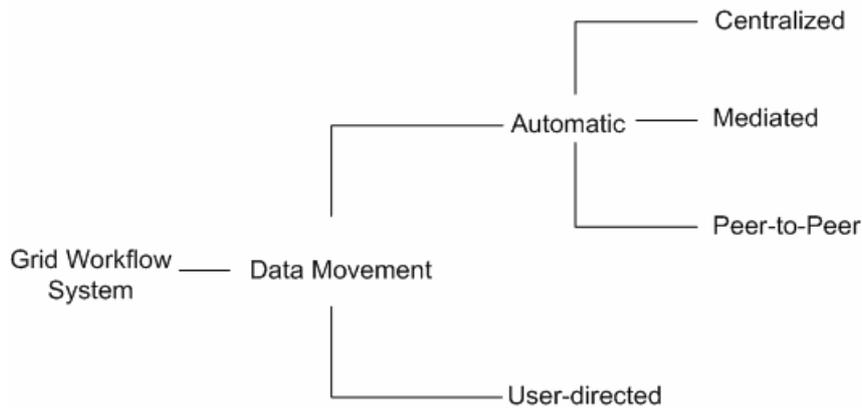


Abbildung 3 – Data Movement

1.4 Taxonomy for Grid Workflow Systems

Um eine Entscheidung bzgl. des zu verwendenden Workflow Systems zu fällen, wurden existierende Grid Workflow Systeme untersucht und eine Taxonomy erstellt (Tabelle 1). Dabei wurde Bezug genommen auf die Arbeiten von [1] und den vorher beschriebenen Klassifikationsmerkmalen.

Tabelle 1 - Taxonomy mapping to Grid Workflow System. Source: A Taxonomy of Scientific Workflow Systems for Grid Computing. Jia Yu and Rajkumar Buyya. University of Melbourne.

Project	Workflow Design		Workflow Scheduling	
	Structure	Model	Architecture	Strategies
DAGMan	DAG	Abstract	Centralized	Performance
Pegasus	DAG	Abstract	Centralized	Performance
ICENI	Non-DAG	Abstract	Centralized	Performance Market
Taverna	DAG	Abstract/Concret	Centralized	Performance
GrADS	DAG	Abstract	Centralized	Performance
GridFlow	DAG	Abstract	Hierarchical	Performance
UNICORE	Non-DAG	Concret	Centralized	User
Gridbus	DAG	Abstract/Concret	Hierarchical	Market
Askalon	Non-DAG	Abstract	Decentralized	Performance Market
OmniBPEL	Non-DAG	Abstract	Centralized	Performance

2. Anforderungen

Im Bereich *Anforderungen* sollte evaluiert werden, in wie weit es ingenieursspezifische Eigenschaften gibt, die bei der Planung/Wahl eines Workflows-Systems berücksichtigt werden müssen. Durch Befragungen der Projektpartner und Anwender ergaben sich folgende Forderungen:

1. Die Struktur von Ingenieurs-Workflows lässt sich nicht mit einem DAG ausdrücken, da Schleifen für Optimierungsschritte gebraucht werden.
2. Die Modellierung der Workflows muss auf einer abstrakten Ebene geschehen, um zum einen die leichte Nutzung für Ingenieure zu gewährleisten und zum anderen, um flexible auf die dynamische Grid Umgebung reagieren zu können.
3. Der entstehende Workflow sollte möglichst standardkonform sein und mit bestehenden Workflow-Systemen aus der Wirtschaft interoperable sein.
4. Der Eingriff des Kontrolleurs während einer Berechnung, bspw. das Hinzufügen von Parametern, soll nicht zu einem Neustart der Berechnung führen und dennoch in den Workflow dynamisch eingegliedert werden können.
5. Um Expertenwissen aus verschiedenen Domänen in eine Workflow zu integrieren wird ein Werkzeug zum kollaborativem Erstellen eines Workflows benötigt.
6. Das Scheduling der Workflows sollte hierarchisch geschehen, damit die verschiedenen Backendschedulingssysteme die in InGrid genutzt werden nahtlos in den Workflow eingebunden werden können.
7. Als Schedulingstrategien sollten neben Performancegetriebene auch Marktplatz- und Securitybasierte Strategien angeboten werden.

Keines der bisherigen Grid Workflow Systeme (siehe Tabelle 1) genügt den Ingenieursanforderungen. Die DAG basierten Workflow Systeme fallen wegen Forderung 1 weg. ICENI benutzt eine eigene, nicht standardkonforme Workflow Sprache sowie eine eigene Grid Middleware die nicht vom DGI unterstützt wird. Askalon nutzt die eigene Workflow Sprache AGWL und ist somit ebenfalls nicht

standardkonform. Das Unicore Workflow System modelliert nicht abstrakte sondern konkrete Workflows und ist nicht interoperable mit Standards aus dem Wirtschaftssektor wie BPEL. OmiBPEL nutzt BPEL als Workflow Beschreibungssprache. Allerdings ist es fest mit einer eigenen Grid Middleware verbunden und generiert nur statische Workflows. Somit genügt keines der untersuchten Workflow Systeme in seiner jetzigen Form den ingenieursspezifischen Anforderungen. Als Results dieser Untersuchung steht fest, dass ein existierendes System um die genannten Anforderungen erweitert werden muss.

Als Basis für die Erweiterungen wurde die BPEL-Engine von Active Endpoints gewählt, da dieser unter der GPL Lizenz veröffentlicht wird. Sie erfüllt Forderungen 1-3. Für die Forderungen 4 bis 7 werden dort standardkonforme Erweiterungen integriert. Erste Prototypische Lösungen für Forderungen 4 und 5 sind schon gemacht und werden im Folgenden beschrieben. Lösungen für 6 und 7 werden später in der Projektlaufzeit angegangen.

2.1 Das GridForEach Konstrukt

(Forderung 4).

Der Eingriff des Kontrolleurs während einer Berechnung, bspw. das Hinzufügen von Parametern, soll nicht zu einem Neustart der Berechnung führen und dennoch in den Workflow dynamisch eingegliedert werden können.

In der gewählten GridWorkflow Engine (BPEL Version 1.1) ist es derzeit nicht möglich, gleichzeitig auf einer beliebig großen Menge von, zur Designzeit des Prozesses nicht bekannten, Endpoint-Services zuzugreifen. Um Punkt 4 der Anforderungen zu erfüllen, wurde in der Arbeitsgruppe von Marburg eine Ergänzung der BPEL-Syntax vorgenommen. Durch die Erweiterung von BPEL durch das "GridForEach"-Konstrukt, ist es nun möglich, während der Laufzeit anfallende Parameter Änderungen direkt mit in die parallele Berechnung einzubauen. Für die Einordnung von GridForEach in die Taxonomy, siehe Abbildung 5.

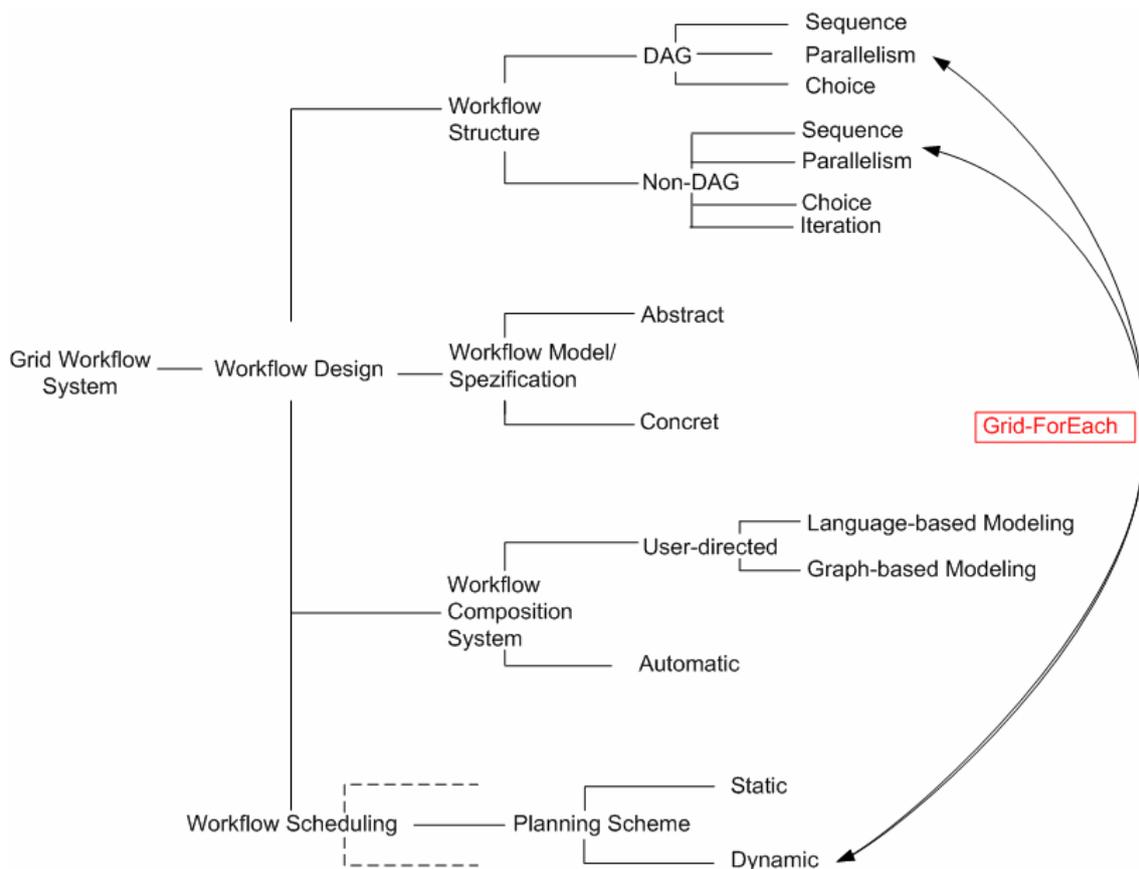


Abbildung 3 - Ein Teil der erstellten Taxonomy, zusammen mit der Eingliederung von der Erweiterung „GridForEach“.

2.2 Workflow Editor

(Forderung 5).

Um Expertenwissen aus verschiedenen Domänen in eine Workflow zu integrieren wird ein Werkzeug zum kollaborativem Erstellen eines Workflows benötigt.

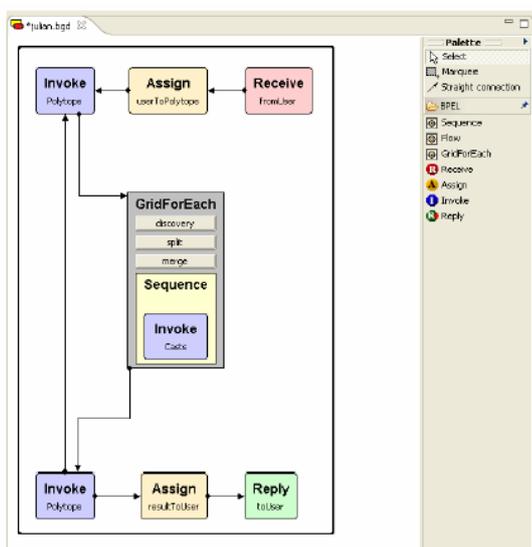


Abbildung 4 – Ein Beispiel Workflow für eine Cast Simulation.

Die Virtualisierung des Workflows in einem kollaborativen Editor trägt dazu bei, die Lücke zwischen Grid-Applikation Entwicklern und Domänen Experten zu schließen. Es wird den Teilnehmern damit die Möglichkeit gegeben, die Applikation interaktive zu entwickeln. Weiterhin stellt der Editor eine Hilfe dar, um Domänen Experten, ohne Informatik-Fachwissen, ein Möglichkeit zu geben, Domänen spezifische Lösungen zu entwickeln, die die gegebene Grid Infrastruktur ausnutzen. Die neben stehende

Abbildung 5 zeigt einen Beispiel Workflow in dem existierenden Editor Prototypen. Dieser wurde auch bereits auf dem InGrid Meeting am 22.06.2006 in Bonn-Oberkassel vorgeführt.

3. Modellierungsmechanismus

In AP 3.1.1 wurde durch die Arbeitsgruppe der Universität Siegen der Prozess der Angebotserstellung sowie das in diesem Prozess erzeugte und verwendete Wissen analysiert und modelliert. Dabei wurde der Prozess mit den Werkzeugen eEPK (erweiterte ereignisgesteuerte Prozesskette) und PROMOTE (Prozess-orientierte Methoden und Werkzeuge für Wissensmanagement) modelliert. Unter dem Aspekt einer späteren Portierung der Ergebnisse zu Web-Services (AP 3.1.5) wurde das Modell ebenfalls mit BPMN (Business Process Modeling Notation) erstellt. Diese Modellierungsmethoden wurden aus folgenden Gründen gewählt:

- eEPK gelten als Quasi-Standard in der Modellierung und sollen auf ihre Potential bzgl. der Umsetzung des Ziels der Wissensmodellierung in einem einheitlichen Modell im Kontext dieses Arbeitspaketes untersucht werden.
- PROMOTE ist auf die Modellierung sowohl von Prozessen als auch von Wissen ausgerichtet. Im Kontext dieses Arbeitspaketes ist eine Validierung notwendig.
- BPMN ist die Modellierungsnotation für Webservices. Da im Rahmen dieses Arbeitspaketes die Implementierung der Ergebnisse als Webservices geplant ist, ist eine direkte Modellierung in der entsprechenden Notation anzustreben

Die Modellierung mit eEPK zeigte einige systemimmanente Probleme dieser Methode. So gibt es keine Möglichkeit, nach einem Split alle Teilstränge zu synchronisieren, bevor der Prozess fortschreitet. Die Modellierung des Wissens ist mittels eEPK nur schwierig und unvollständig durchführbar, da diese Methode originär zur Prozessmodellierung entwickelt wurde und deshalb kaum Visualisierungsmöglichkeiten hinsichtlich der Erfassung von Wissen bietet. Dies ist wiederum eine der Stärken von PROMOTE. Diese Methode legt den Schwerpunkt auf die Modellierung von Wissensprozessen in Verbindung mit Geschäftsprozessen. Aus diesem Grund ist die Darstellung von Geschäftsprozessen weniger formal als dies bei eEPK der Fall ist, es lassen sich aber mit den verschiedenen Methoden unterschiedliche Aspekte des Wissensflusses und der Wissensverarbeitung darstellen.

BPMN ist analog eEPK für die Modellierung von Geschäftsprozessen gedacht, mit dem Aspekt der Generierung von BPEL4WS, einer Ausführungssprache für

Webservices. BPMN bietet eine Vielzahl von Modellierungsmöglichkeiten für Geschäftsprozesse, während das Wissen im Prozess nur in Form von Dokumenten bzw. Data Objects dargestellt werden kann. Durch die Arbeitsgruppe der Universität Marburg wurden daher ebenfalls BPMN-Modelle erstellt, um die Integration in die Aktivitäten zum Thema Workflow (→ AP 3.2) bereits zu diesem Zeitpunkt möglichst effizient zu gewährleisten. Die Ergebnisse der Analyse der Prozesse sowie Notationen in diesem AP flossen ebenfalls in die Arbeiten in AP 3.2 ein.

Auf Basis dieser Modelle wurde in AP 3.1.2 ein Datenmodell in UML (Unified Modeling Language)-Notation mit den relevanten Daten des Angebotsprozesses sowie einfließenden

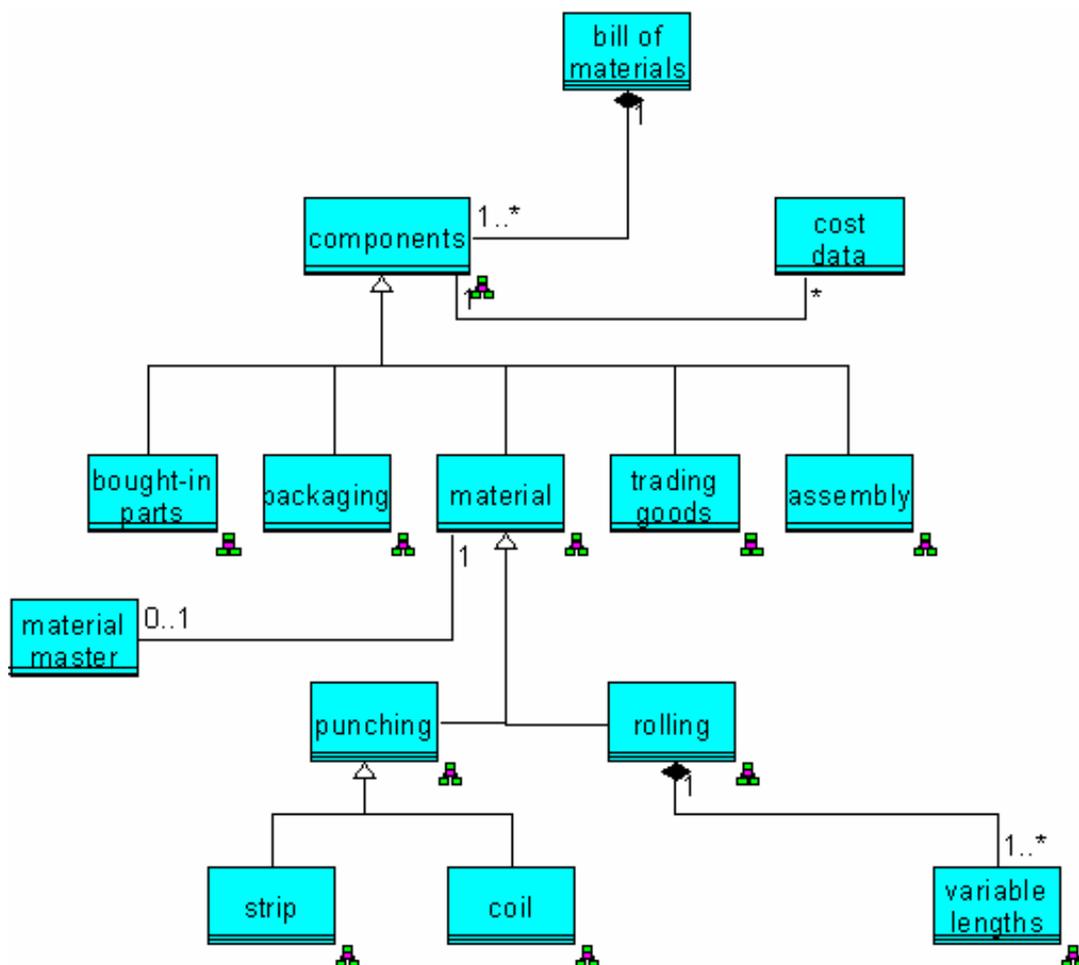


Abbildung 5 - Ausschnitt des Datenmodells (in UML) für die Repräsentation des Wissen in Produkt- und Prozessdaten am Beispiel der Umformtechnik

Stammdaten erstellt (s. Abb. 3). Durch intensive Mitarbeiterinterviews wurden die Attribute identifiziert, deren Ausprägung durch einfließendes Wissen bestimmt wird (Wissensattribute). Diese stehen im Gegensatz zu den Attributen, deren Ausprägung

durch Routinetätigkeiten bestimmt wird. Diese Wissensattribute sind die Basis für das weitere Vorgehen, da in ihnen das relevante Wissen abgelegt ist, das zur Wiederverwendung extrahiert werden muss.

Literatur

[1] : *A Taxonomy of Scientific Workflow Systems for Grid Computing*, Jia Yu and Rajkumar Buyya, *SIGMOD Rec.*, Volume 34, 44-49, 2005. University of Melbourne