

# Modellierung und Abwicklung von Datenflüssen in unternehmensübergreifenden Prozessen

Markus Bon  
AG DBIS  
Fachbereich Informatik  
Universität Kaiserslautern  
Postfach 3049  
D-67653 Kaiserslautern  
bon@informatik.uni-kl.de

Norbert Ritter  
Arbeitsbereich VSIS  
Fachbereich Informatik  
Universität Hamburg  
Vogt-Kölln-Strasse 30  
D-22527 Hamburg  
ritter@informatik.uni-hamburg.de

Hans-Peter Steiert  
Research & Technology  
RIC/ED  
DaimlerChrysler AG  
Postfach 2360  
D-89013 Ulm  
hans-peter.steiert@daimlerchrysler.com

**Zusammenfassung:** Workflow-Management ist ein mittlerweile wohl erforschtes Gebiet, soweit es um die Möglichkeiten einzelner Workflow-Management-Systeme (WfMS) geht. Eine automatisierte Kontrolle von Abhängigkeiten zwischen von heterogenen WfMS gesteuerten Prozessen ist dagegen weitgehend unerforscht. Eine solche Kontrolle ist jedoch hinsichtlich der Steuerung von firmenübergreifenden Prozessen als äußerst wünschenswert zu betrachten, da eine beträchtliche Aufwandsreduzierung zu erwarten ist. Dieser Artikel leistet einen Beitrag zur Erschließung dieses Gebiets durch Beschreibung eines Ansatzes zur automatisierten Abwicklung von Datenflüssen zwischen heterogenen Workflows. Es werden die wesentlichen, für die Entwicklung einer entsprechenden Integrationskomponente relevanten Aspekte aufgezeigt und ein auf EAI-Technologie basierender Architekturansatz vorgestellt.

## 1 Einleitung

In großen Unternehmen existieren eine Vielzahl verschiedener, oftmals sehr komplexer Prozesse. Aufgrund der Forderungen nach immer kürzeren Entwicklungszeiten wird die Entwicklung nicht mehr allein im Unternehmen selbst durchgeführt, sondern es wird immer öfter auf das Fachwissen spezialisierter Firmen zugegriffen. Gerade zur Steuerung solcher komplexer, unternehmensübergreifender Prozesse bietet Workflow-Management-Technologie ein großes Nutzenpotential.

Im Projekt COW (Cross-Organizational-Workflows [KRS01]) wurde bereits untersucht, inwieweit sich inselübergreifende Prozesse beschreiben lassen, und wie ein globaler Kontrollfluss umgesetzt werden kann. Der Begriff der „Insel“ bezieht sich dabei auf die Gesamtheit der in einer Firma vorhandenen Systeme (Workflow-Management-Systeme, Produktdaten-Management-Systeme, etc.), Prozess- und Workflow-Typen sowie Ressourcen (Applikationen und/oder organisatorische Einheiten). Ein in diesem Zusammenhang äußerst wichtiger Aspekt ist der inselübergreifende Datenfluss [BHR01, BRH02].

Für eine sinnvolle Zusammenarbeit ist es notwendig, dass Daten zwischen den beteiligten Gruppen fließen. Beispielsweise müssen an einer Konstruktion beteiligte Ingenieure CAD-Geometrien austauschen, um die Passgenauigkeit der Einzelteile überprüfen zu können. Die automatisierte Abwicklung von solchen inselübergreifenden Datenflüssen ist unser Ziel.

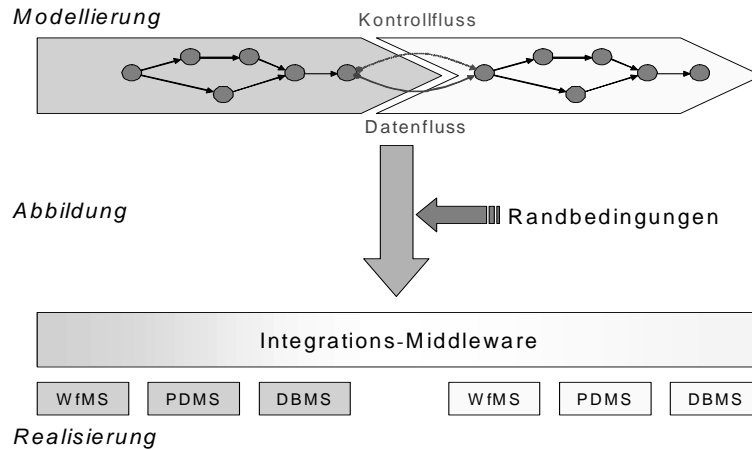


Abbildung 1: Modellierung und Realisierung von Datenflüssen

Um dieses Ziel zu erreichen, sind die folgenden Probleme zu lösen:

- Die Modellierung (siehe Abbildung 1, oberer Teil) unternehmensübergreifender Prozesse muss angemessen unterstützt werden, so dass Datenflussabhängigkeiten (DfA) zwischen den Inseln und die daran geknüpften Eigenschaften hinreichend beschrieben werden können. Hierzu sind Kategorien von Datenflüssen zu identifizieren.
- Für jede Kategorie ergeben sich verschiedene Realisierungsmöglichkeiten. Um zu einer gegebenen Systemlandschaft und organisatorischen Randbedingungen die passende Realisierung bestimmen zu können, ist es notwendig, Randbedingungen zu beschreiben und Implementierungsrichtlinien zu entwickeln (siehe Abbildung 1, mittlerer Teil).
- Letztlich wird eine Middleware benötigt, die eine integrierte Sicht auf unternehmensübergreifende Workflows erlaubt. In unserem Fall bedeutet dies, dass der Datenfluss im Rahmen von unternehmensübergreifenden Prozessen automatisiert abgewickelt wird (siehe Abbildung 1, unterer Teil). Dabei halten wir es für eine wichtige Anforderung, dass die Realisierung möglichst ohne größere Eingriffe in die beteiligten Infrastrukturen erfolgen soll.

In diesem Papier diskutieren wir die wesentlichen Aspekte des inselübergreifenden Datenflusses und leiten daraus einen Satz von sinnvollen Kategorien (Integrationsmuster) ab. Anschließend erläutern wir, dass zu einem Integrationsmuster unterschiedliche Möglichkeiten der Realisierung ins Auge gefasst werden müssen, da sich die Realisierung an den tatsächlichen Gegebenheiten der beteiligten Infrastrukturen orientieren muss. Weiterhin werden wir aufzeigen, wie sich heutige WfMS- und EAI-Werkzeuge zusammen mit ergänzenden Komponenten verwenden lassen, um die benötigte Middleware bereitzustellen, auf deren Basis sich die Datenflüsse realisieren lassen. Wir schließen das Papier mit einer Zusammenfassung und einem Ausblick.

## 2 Identifikation der Klassifikationskriterien

Um einen inselübergreifenden Fluss von Kooperationsdaten definieren zu können, benötigen wir einen Satz von Kriterien, mit denen sich die Eigenschaften des Datenflusses beschreiben lassen. Dazu sind im Einzelfall folgende Fragestellungen zu betrachten, die in den Abschnitten 2.1 bis 2.4 detailliert weiterverfolgt werden.

- In welcher Form werden die Daten vor dem eigentlichen Datenfluss verwaltet?
- Welche Wirkung hat die Durchführung des Datenflusses sowohl auf der Quell- als auch auf der Zielseite?
- Welche Auswirkungen hat der Datenfluss auf die Besitz- und Eigentumsverhältnisse?
- In welcher Form liegen die Daten vor und wie können sie der Zielseite bereitgestellt werden?

Wir gehen für unsere Betrachtungen zunächst von einem vereinfachten Szenario aus. Wir nehmen dabei an, dass lediglich zwei Insel-Systeme miteinander über eine DfA verknüpft werden sollen. Auf beiden Inseln befindet sich jeweils ein WfMS zur automatisierten Unterstützung lokaler Prozesse. Insbesondere bearbeiten diese Systeme auch diejenigen Workflows, die als Quell- bzw. Ziel-Workflows globaler DfAs zu betrachten sind. Weiterhin befinden sich auf beiden Inseln Applikationen, mit deren Hilfe prozessrelevante Tätigkeiten durchgeführt werden sollen. Die zu verarbeitenden Daten werden in der Regel persistent gespeichert. Daher befindet sich auf jeder der Inseln auch ein Datenhaltungssystem ( $DS_Q$  und  $DS_Z$ ). Als weitere Komponente ist die übergeordnete Schicht zur Insel-übergreifenden Workflow-Integration (Wfi) zu nennen. Diese repräsentiert alle Mechanismen, die benötigt werden, um definierte DfAs zu überwachen, aufzulösen und für das Bereitstellen der Kooperationsdaten in der gewünschten Form zu sorgen. Sie stellt dazu die Funktionalität zur Verfügung, die zusätzlich zu den auf den einzelnen Inseln bereits zur Verfügung stehenden Möglichkeiten benötigt wird.

Die Diskussionen in den nachfolgenden Abschnitten beziehen sich auf dieses Szenario und wir gehen davon aus, dass ein Prozessschritt, der durch  $WfMS_Q$  angestoßen und durch  $A_Q$  bearbeitet wurde, Daten produziert, die aufgrund einer definierten DfA von der Insel 1 zur Insel 2 'fließen' müssen.

### 2.1 Verwaltung der Daten

Wir sprechen von *DS-verwalteten Daten*, wenn die Daten nach ihrer Erzeugung im lokalen Datenhaltungssystem  $DS_Q$  gespeichert werden. In diesem Fall wird die Durchführung des globalen Datenflusses von Insel 1 nach Insel 2 einen Zugriff auf  $DS_Q$  erfordern.

Sollte es sich bei den Kooperationsdaten nicht, wie im ersten angesprochenen Fall, um Applikationsdaten sondern um Workflow-relevante Daten handeln, die, wie es die meisten gängigen WfMS unterstützen, von der Applikation an das WfMS zur weiteren Verwaltung übergeben werden, so sprechen wir von *WfMS-verwalteten Daten*.

## 2.2 Wirkung des Datenflusses

Nachdem wir die Verwaltung der Daten auf der Quell-Insel betrachtet haben, wollen wir nun die Abwicklung globaler Datenflüsse näher untersuchen. Wir unterscheiden den Datenfluss anhand seiner Wirkung auf Insel 1 zwei Varianten:

- *quellerhaltend*: die Daten stehen nach der Datenübertragung immer noch auf der Quellseite bereit.
- *quellverbrauchend*: die Daten werden auf der Quellseite entfernt.

Betrachten wir nun die möglichen Wirkungen des Datenflusses als Kombinationen der Verwalter auf beiden Inseln und der Wirkung auf Insel 1, so ergeben sich für das Übertragen der Kooperationsdaten folgende Varianten.

1. *Replizieren*: quellerhaltend von  $DS_Q$  nach  $DS_Z$
2. *Kopieren*: quellerhaltend von  $DS_Q$  nach  $WfMS_Z$
3. *Abgeben*: quellverbrauchend von  $DS_Q$  nach  $DS_Z$
4. *Übergeben*: quellverbrauchendes von  $DS_Q$  nach  $WfMS_Z$
5. *Verteilen*: quellerhaltend von  $WfMS_Q$  nach  $WfMS_Z$
6. *Eintragen*: quellerhaltend von  $WfMS_Q$  nach  $DS_Z$
7. *Reisen*: quellverbrauchend von  $WfMS_Q$  nach  $WfMS_Z$
8. *Ablegen*: quellverbrauchend von  $WfMS_Q$  nach  $DS_Z$

## 2.3 Eigentümer und Besitzer

Ein weiteres Unterscheidungsmerkmal ist die Frage nach Eigentümer und Besitzer der Daten. *Besitzer* von Daten ist grundsätzlich jeder, der auf diese in irgendeiner Form zugreifen kann und darf. *Eigentümer* von Daten zu sein bedeutet, die Kontrolle darüber zu haben, was mit diesen Daten geschieht, bzw. welche Version der Daten als gültig anzusehen ist. Der Eigentümer ist auch für die Gültigkeit und Konsistenz der von ihm angebotenen Daten verantwortlich. Ein Weitergeben von Daten muss dabei nicht den Verlust des Eigentums bedeuten. Es kann durchaus die Situation entstehen, dass der Eigentümer seine Daten zeitweise überhaupt nicht in Besitz hat. In Tabelle 1 haben wir eine Übersicht über die unserer Meinung nach relevanten Fälle zusammengestellt. Der Buchstabe E markiert den Eigentümer, B markiert den Besitzer.

## 2.4 Bereitstellungsmodus

Für die Bereitstellung der Kooperationsdaten auf der Zielinsel kommen zwei verschiedene Bereitstellungsmodi in Frage. Der Modus *materialisiert* beschreibt den Fall, dass die Kooperationsdaten physisch auf der Zielseite zum Zugriff bereitgestellt werden. Natürlich ist dabei zu beachten, dass gerade im CAX-Bereich sehr große Datenmengen anfallen können, deren physische Übertragung sich sehr aufwändig gestalten kann. Daher macht die materialisierte Bereitstellung nur dann Sinn, wenn

sicher ist, dass die Daten auf der Zielseite in vollem Umfang benötigt und zugegriffen werden.

Der Modus *referenziert* hingegen sieht zunächst lediglich die Übergabe einer Referenz vor, so dass die Daten dann, wenn sie tatsächlich gebraucht werden, anhand dieser Referenz angefordert werden können. Hierbei besteht zusätzlich die Möglichkeit einer Art Parametrisierung, so dass exakt die Datenmenge angefordert und bereitgestellt werden kann, die tatsächlich benötigt wird. Ein solches Konzept ist offenbar in dem Fall sinnvoll, in dem die tatsächlich benötigten Daten in ihrem Umfang a priori nicht vollständig spezifiziert werden können.

	Vorher		Nachher		Beschreibung
	Insel 1	Insel 2	Insel 1	Insel 2	
1	<i>E, B</i>	-	<i>E, B</i>	<i>B</i>	Kopie (K)
2	<i>E, B</i>	-	<i>B</i>	<i>E, B</i>	Kopie, Abgabe des Eigentumsrechts (KAE)
3	<i>E, B</i>	-	<i>E, B</i>	<i>E, B</i>	Kopie, Gewähr des Eigentumsrechts (KGE)
4	<i>E, B</i>	-	<i>E</i>	<i>B</i>	Übergabe (Ü)
5	<i>E, B</i>	-	-	<i>E, B</i>	Übergabe, Abgabe des Eigentumsrechts (ÜAE)
6	<i>B</i>	<i>E</i>	-	<i>E, B</i>	Rückgabe (R)
7	<i>B</i>	<i>E</i>	<i>B</i>	<i>E, B</i>	Rückgabe, Beibehaltung der Kopie (RBK)

Tabelle 1: Übersicht über Eigentümer/Besitzer-Konstellationen

## 2.5 Modellierung unternehmensübergreifender Datenflüsse

Nachdem wir in den Abschnitten 2.1 bis 2.4 wesentliche Merkmale von Inselübergreifenden Datenflüssen analysiert haben, führen wir diese nun zusammen und beschreiben in diesem Abschnitt die sinnvoll zu bildenden Kombinationen. Daraus ergibt sich, welche Kategorien von Datenflüssen eine Middleware unterstützen muss, die zur Realisierung herangezogen werden soll.

Aus den bisher beschriebenen Elementen lassen sich eine Vielzahl verschiedener Arten von Datenflüssen herleiten. Betrachten wir das Zusammenspiel zwischen Verwaltung auf der Quellseite, Bereitstellungsmodus, Wirkung der Datenübertragung und der Besitz-/Eigentumsrechte, so sind nicht alle theoretisch bildbaren Kombinationen auch sinnvoll möglich. Wenn die Wirkung des Datenflusses im Transfer einer Kopie besteht, dann muss die Quellinsel auch Besitzer der Daten bleiben; Besteht die Wirkung des Datenflusses in der Übergabe der Daten, dann darf die Quellinsel nicht Besitzer der Daten bleiben. Eine referenzierte Übertragung von auf der Quellseite WfMS-verwalteten Daten ist nicht sinnvoll, da WfMS in der Regel nur eine lokale Verwendung von Workflow-relevanten Daten unterstützen. Tabelle 2 gibt eine Übersicht über mögliche Integrationsmuster, die sich als sinnvolle Kombination hinsichtlich der in den Abschnitten 2.1 bis 2.4 beschriebenen Aspekte ergeben.

Verwaltung Quelle	Modus	Wirkung	E/B-Konstellation
DS	materialisiert, referenziert	Replizieren, Kopieren	K, KAE, KGE, RBK
		Abgeben, Übergeben	Ü, ÜAE, R
WfMS	materialisiert	Verteilen, Eintragen	K, KAE, KGE, RBK
		Ablegen, Reisen	Ü, ÜAE, R

Tabelle 2: Integrationsmuster

### 3 Realisierungsaspekte

Eine Diskussion der wesentlichen Einflussfaktoren eines Insel-übergreifenden Datenflusses und ihrer Kombinationsmöglichkeiten, wie sie im vorangegangenen Kapitel vorgenommen wurde, ist natürlich notwendig, um eine Middleware zu entwickeln, die solche Datenflüsse automatisiert abarbeiten kann.

Bei der Abwicklung eines gewünschten Datenflusses sind jedoch auf den beteiligten Inseln vorherrschende technische und organisatorische Randbedingungen zu berücksichtigen. Diese bestimmen die zur Abwicklung des Datenflusses tatsächlich notwendigen Aktionen. Daraus folgt wiederum, dass eine geeignete Middleware in der Lage sein muss, diese Randbedingungen in flexibler Weise zu berücksichtigen.

#### **Anbindung der WfI an WfMS<sub>1</sub>**

Aktuelle WfMS unterstützen lokale Datenflüsse [LR00]. Soll die lokale Umgebung jedoch verlassen werden, wird eine Schnittstelle nach außen benötigt. Hierzu gibt es im wesentlichen zwei Möglichkeiten: Die lokalen Workflow-Typen werden um Aktivitäten erweitert, die die Zusammenarbeit mit der WfI übernehmen, oder die WfI bekommt die Möglichkeit, die Workflow-Abarbeitung durch das WfMS<sub>1</sub> zu überwachen. Im ersten Fall werden in die lokalen Workflow-Typen an den Stellen spezielle, neue Aktivitäten eingefügt, an denen Datenflüsse zu anderen Inseln abgewickelt werden müssen. Diese Aktivitäten rufen die WfI auf, übergeben die Referenz auf die Daten und lösen so den Datenfluss aus. Im zweiten Fall überwacht die WfI beispielsweise den Zustand relevanter Workflow-Instanzen. Wird das Ende einer Aktivität erkannt, deren Ergebnis übertragen werden soll, so liest die WfI die Datenreferenz aus dem WfMS aus und überträgt die Daten.

#### **Datenzugriff**

Für den Zugriff auf die Kooperationsdaten in DS<sub>Q</sub> bestehen im wesentlichen zwei Alternativen: Zum einen kann eine API von DS<sub>Q</sub> zum direkten lesenden Zugriff genutzt werden; zum anderen unterstützen viele Systeme eine Art Check-Out-Mechanismus. Dieser kann genutzt werden, um die Daten zunächst in ein handhabbares Format, beispielsweise XML, zu extrahieren und anschließend zu übertragen.

## **Eigentümer- und Besitzerverwaltung**

Die Verwaltung der Eigentums- und Besitzerrechte muss die WfI als globale Instanz übernehmen. Dies allein reicht aber nicht aus. Wenn andere beteiligte Komponenten passende Unterstützung anbieten, dann sollte diese genutzt werden. So bieten die meisten PDMS auch Mechanismen für Computer-gestützte Gruppenarbeit an. Wird ein PDMS für die Datenhaltung verwendet, dann muss die WfI die internen Eigentums- und Besitzerrechte auf die unterliegenden Mechanismen abbilden. Dies kann von System zu System unterschiedlich sein.

## **4 Abbildung auf existierende Workflow- und EAI-Technologie**

Wie bereits deutlich wurde, findet die Integration unternehmensübergreifender Abläufe nicht „auf der grünen Wiese“ statt, sondern muss die Systemlandschaften auf den beteiligten Inseln berücksichtigen. Daher muss die WfI mit den vorhandenen Systemen verträglich sein. Weiterhin sollten nur dort Eigenentwicklungen stattfinden, wo keine passenden Komponenten am Markt erhältlich sind. In unserem Fall soll vorhandene EAI-Technologie eingesetzt werden, um notwendige neue Komponenten und vorhandene Systeme zusammen zu schalten. Wir streben die in Abbildung 2 gezeigte Grobarchitektur an.

Auf beiden Inseln arbeiten lokale WfMS Workflows nach den Vorgaben entsprechender Workflow-Typen ab. Wir gehen hier davon aus, dass die Anbindung an die WfI über speziell eingefügte Aktivitäten erfolgt. Diese werden im folgenden auch Datenflussaktivitäten genannt.

Die WfI besteht aus kommerziellen EAI-Brokern, wie beispielsweise IBM's CrossWorlds [IBM02] oder Microsoft's BizTalk [Ms01], und neu entwickelten Bausteinen, mit denen fehlende Funktionalität ergänzt wird. Im Bild ist ein Baustein für die Eigentümer-/Benutzerverwaltung vorgesehen.

Der EAI Broker kommuniziert mit den anderen Komponenten (Datenquellen, WfI-Komponenten, WfMS) über sogenannte Konnektoren. Diese haben drei Aufgaben:

- Wenn der EAI-Broker einen Dienst von einer integrierten Komponente anfordert, wird diese Anforderung vom Konnektor auf die Schnittstelle des entsprechenden Systems abgebildet.
- Wenn in einer angebundenen Komponente ein für die Integration relevantes Ereignis auftritt, wird dieses erkannt und an den EAI-Broker weitergeleitet. Die Ereigniserkennung erfordert entweder ein regelmäßiges Abfragen des Zustandes oder die Mithilfe der Komponente, beispielsweise über einen Callback-Mechanismus, einen direkten Aufruf oder, im Fall eines relationalen Datenbanksystems, durch einen geeignet definierten Trigger.
- Da EAI-Broker intern bevorzugt mit einem neutralen Datenformat arbeiten, bzw. unterschiedliche Komponenten mit unterschiedlichen Datenformaten arbeiten, führen die Konnektoren auch Datentransformationen durch.

Die Art und Weise, wie ein Integrationsmuster abgearbeitet werden soll, wird in so genannten *Kollaborationen* bestimmt. Wird über den Konnektor, der das WfMS anbindet, ein entsprechendes Ereignis signalisiert, wird im Broker der Start der entsprechenden Kollaboration ausgelöst.

Die beschriebene Architektur erlaubt uns, die in Abschnitt 2 eingeführten Kategorien von Datenflüssen zunächst unabhängig von der Technologie als Kollaborationen zu definieren und zu realisieren. Es ist Aufgabe der Konnektoren, die Abbildung auf konkrete Systeme durchzuführen. Zusätzlich benötigte Funktionalität kann ebenfalls leicht über neue Systemkomponenten integriert werden.

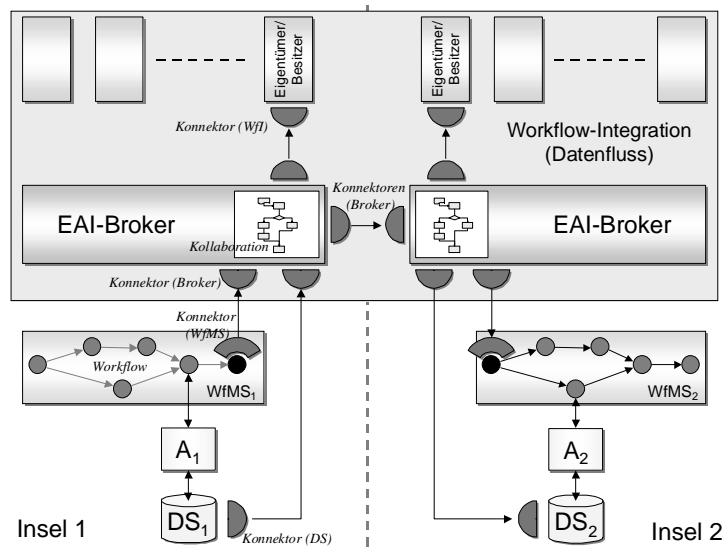


Abbildung 2: Integrationsarchitektur für Insel-übergreifende Datenflüsse

## 5 Verwandte Arbeiten

Wie in diesem Artikel deutlich geworden ist, streben wir eine Prozessintegration durch Bereitstellung einer Middleware-Komponente an, die auf Workflow-Technologie aufsetzt. Zur Realisierung dieser zentralen Komponente soll kein globales Workflow-Modell herangezogen werden, sondern es sollen die in der Realität auftretenden grundlegenden Abhängigkeiten zwischen Workflow-Umgebungen untersucht und eine Integrationskomponente entwickelt werden, welche die verschiedenen Formen dieser Abhängigkeiten spezifisch unterstützt. Unseres Wissens gibt es keine anderen Ansätze, die auch einen solchen Anspruch verfolgen.

Es gibt jedoch sehr wohl andere Ansätze, die sich mit der Thematik des firmenübergreifenden Austauschs von Produktdaten zum gemeinsamen Entwickeln von Produkten beschäftigen. Im Projekt ANICA wird versucht, über einen Integrationsbus den Zugriff auf verteilte CAX-Datenquellen zu ermöglichen. Dabei soll nicht nur der Ort der Daten transparent gehalten werden, sondern durch interne Konvertierung auch



(soweit möglich) das Format [AJSK98]. In dem in [AGL98] beschriebenen Projekt liegt der Fokus ebenfalls auf den Möglichkeiten der direkten Kopplung von heterogenen PDMS. Ein weiterer Ansatz, der Datenaustausch zwischen heterogenen PDMS ermöglichen soll, ist die „Product Data Markup Language“, die für das amerikanische Verteidigungsministerium (Department of Defense, DoD) entwickelt wurde [Bur99]. Hier wird versucht, zu Integrationszwecken ein neutrales Integrationsschema zu nutzen. Somit gehören die vorgenannten Projekte zu einer Klasse von Forschungsansätzen, die das Gesamtproblem eher von der Seite der Datenintegration als von der Seite der Prozessintegration lösen wollen.

## **6 Zusammenfassung und Ausblick**

Dieser Artikel leistet einen Beitrag zur Erforschung des Gebiets der unternehmensübergreifenden Workflows, die sich im wesentlichen über Abhängigkeiten zwischen heterogenen Workflow-Umgebungen (Inseln) definieren. Wir gehen dabei von einem Ansatz aus, der einerseits eine automatisierte Kontrolle solcher Abhängigkeiten ermöglicht, andererseits aber die betroffenen Systemumgebungen weitestgehend unbeeinflusst lässt.

In diesem Rahmen besteht der eigentliche Beitrag dieses Artikels in der Betrachtung von Datenflussabhängigkeiten, die im Anwendungsfall selbstverständlich zunächst erkannt und modelliert werden müssen, um sie dann zur Laufzeit automatisiert abwickeln zu können und damit einen reibungslosen Gesamtprozess, d.h. eine reibungslose ‚Kooperation‘ heterogener Workflows, gewährleisten zu können. Als wesentliche Bestimmungsfaktoren haben wir die Art der Verwaltung der Kooperationsdaten auf der Quellseite, die Wirkung der Ausführung des eigentlichen Datenflusses auf sowohl die Quell- als auch die Zielinsel, den Modus der Bereitstellung der Kooperationsdaten auf der Zielinsel sowie die geeignete Regelung der Eigentums- und Besitzrechte angesichts der durch den eigentlichen Datenfluss sich verändernden Situation identifiziert.

Aus diesen Aspekten und den möglichen Kombinationen ihrer Ausprägungen ergeben sich eine Vielzahl von Integrationsmustern. Jedes dieser Muster beschreibt einen möglicherweise auftretenden Fall, so dass die Systemunterstützung bereits angesichts der Vielzahl dieser Muster sehr flexibel ausgelegt werden muss. Die integrierende Middleware muss auch hinsichtlich der heterogenen Systemlandschaften ausreichend flexibel einsetzbar sein, um die spezifischen Fähigkeiten der durch eine zu kontrollierende Datenflussabhängigkeit verbundenen Systeme (insbesondere Workflow-Management-Systeme) gezielt und gewinnbringend bei der automatischen Abwicklung zu berücksichtigen. Diese Anforderung macht es zunächst sehr schwer, die Systemlösung vorzuschlagen. Daher haben wir unseren Architekturvorschlag ausgehend von einem beispielhaften Integrationsmuster motiviert.

Die dargestellte Architektur, die auf den Einsatz existierender EAI-Technologie setzt, muss zwar noch verfeinert werden, um alle möglichen Fälle zu unterstützen, zeigt aber, wie von uns identifizierte Integrationsmuster als Kollaborationen allgemein anwendbar gemacht werden können. Systemspezifische Eigenschaften werden in Konnektoren gekapselt, welche die Systeme befähigen, an der Integration teilzunehmen. Fehlende Funktionalität wird zusätzlich durch ergänzenden Systemkomponenten integriert.

Weitere Untersuchungen und prototypische Implementierungen werden zeigen müssen, ob aktuelle EAI-Technologie geeignet ist, auch weitere Aspekte zu berücksichtigen. Offen ist auch, wie sich die EAI-Lösungen im Fehlerfall verhalten und inwieweit sich unterschiedliche EAI-Broker zur Zusammenarbeit bewegen lassen.

## 7 Literatur

- [AGL98] Abramovici, M., Gerhard, D., Langenberg, L.: Supporting Distributed Product Development Processes with PDM, in: Krause, Heimann, Raupach.(Hrsg), New Tools and Workflows for Product Development, Proc. CIRP Seminar STC Design, Mai 1998, Berlin, Fraunhofer IRB Verlag, 1998, 1-11
- [AJSK98] Arnold F., Janocha A. T., Swieczek B., Kilb T.: Die CAX-Integrationsarchitektur ANICA und ihre erste Umsetzung in die Praxis, in: Proc. Workshop Integration heterogener Softwaresysteme (IHS '98), 28. GI-Jahrestagung Informatik'98 - Informatik zwischen Bild und Sprache, Magdeburg, September 1998, 43-54
- [BRZ00] Bon, M., Ritter, N., Zimmermann, J.: Interoperabilität heterogener Workflows, Proc. GI-Workshop Grundlagen von Datenbanken, 2000, 11-15
- [BHR01] Bon, M., Härder, T., Ritter, N.: Produktdaten-Verwaltung in heterogenen Workflow-Umgebungen, Interner Bericht, Dezember 2001
- [BRH02] Bon, M., Ritter, N., Härder, T.: Sharing Product Data among Heterogeneous Workflow Environments, in Proc. Int. Conf. CAD 2002 - Corporate Engineering Research, Dresden, März 2002, 139-149
- [Bur99] Burkett, W.: PDML - Product Data Markup Language - A New Paradigm for Product Data Exchange and Integration, 30.04.1999, [www.pdml.org/whitepap.pdf](http://www.pdml.org/whitepap.pdf)
- [IBM02] Technical Introduction to IBM CrossWorlds, IBM Corporation 2002
- [KRS01] Kulendik, O., Rothermel, K., Siebert, R.: Cross-organizational workflow management - General Approaches and their Suitability for Engineering Processes. in: Schmid, B., Stanoevska-Slabeva, K., Tschammer, V. (Hrsg.): Proc. First IFIP-Conference on E-Commerce, E-Business, E-Government: I3E 2001, Zürich, Schweiz, Oktober 2001
- [LR00] Leymann, F., Roller, D.: Production Workflow: Concepts and Techniques, Prentice Hall PTR (ECS Professional), 2000, ISBN 0-13-021753-0
- [Ms01] Microsoft BizTalk Server 2000: Documented, Microsoft Press, 2001