

Zweitveröffentlichung



Mantel, Stephan; Eckert, Sven; Schissler, Martin; Ferstl, Otto K.; Sinz, Elmar J.

Entwicklungsmethodik für überbetriebliche Kopplungsarchitekturen von Anwendungssystemen

Datum der Zweitveröffentlichung: 19.09.2024

Verlagsversion (Version of Record), Konferenzveröffentlichung

Persistenter Identifikator: urn:nbn:de:bvb:473-irb-981756

Erstveröffentlichung

Mantel, Stephan; Eckert, Sven; Schissler, Martin; Ferstl, Otto K.; Sinz, Elmar J. (2002):

„Entwicklungsmethodik für überbetriebliche Kopplungsarchitekturen von Anwendungssystemen“. In: Dieter Bartmann (Hrsg.), Kopplung von Anwendungssystemen : FORWIN-Tagung 2002, Aachen: Shaker, S. 183–202.

Rechtehinweis

Dieses Werk ist durch das Urheberrecht und/oder die Angabe einer Lizenz geschützt. Es steht Ihnen frei, dieses Werk auf jede Art und Weise zu nutzen, die durch die für Sie geltende Gesetzgebung zum Urheberrecht und/oder durch die Lizenz erlaubt ist. Für andere Verwendungszwecke müssen Sie die Erlaubnis der Rechteinhaberinnen und Rechteinhaber einholen.

Für dieses Dokument gilt eine Creative-Commons-Lizenz.



Die Lizenzinformationen sind online verfügbar:

<https://creativecommons.org/licenses/by/4.0/legalcode>

Entwicklungsmethodik für überbetriebliche Kopplungsarchitekturen von Anwendungssystemen

Stephan Mantel, Universität Bamberg

Sven Eckert, Universität Bamberg

Martin Schissler, Universität Bamberg

Otto K. Ferstl, Universität Bamberg

Elmar J. Sinz, Universität Bamberg

Zusammenfassung: Die überbetriebliche Integration von Anwendungssystemen (AwS) erfolgt durch deren Kopplung auf der Grundlage einer geeigneten Kopplungsarchitektur. Der vorliegende Beitrag stellt zunächst drei Klassen von Kopplungsarchitekturen vor, die sich anhand der jeweils verfolgten Kopplungsziele unterscheiden: ereignisorientierte, datenorientierte und funktionsorientierte Kopplungsarchitekturen. Anschließend wird eine Entwicklungsmethodik zur Gestaltung von AwS-Kopplungsarchitekturen für überbetriebliche Geschäftsprozesse beschrieben. Die Methodik unterscheidet drei Modellebenen: die Ebene des überbetrieblichen Geschäftsprozesses, die AwS-Ebene und die Kopplungsarchitektur-Ebene. Zur Veranschaulichung der Methodik wird diese auf den Bereich des Supply-Chain-Managements angewendet. Dabei werden am Beispiel des überbetrieblichen betriebswirtschaftlichen Konzepts Vendor-Managed-Inventory (VMI) die einzelnen Schritte der Entwicklungsmethodik detailliert dargestellt.

Stichwörter: überbetriebliche Geschäftsprozesse, Integration von Anwendungssystemen, Kopplungsarchitekturen, Supply-Chain-Management

1 Einleitung

Der vorliegende Beitrag beschreibt eine Methodik für die Entwicklung von Kopplungsarchitekturen zur Integration von Anwendungssystemen (AwS). Dabei steht die unternehmensübergreifende Kopplung von AwS zur Unterstützung überbetrieblicher Geschäftsprozesse im Vordergrund. Ein überbetrieblicher Geschäftsprozess umfasst Aufgaben, die unterschiedlichen Unternehmen zugeordnet sind.

Zunächst werden in Kapitel 2 mögliche Arten von Kopplungsarchitekturen zur Kopplung von AwS beschrieben. Anschließend wird in Kapitel 3 das Konzept einer Entwicklungsmethodik für die Gestaltung überbetrieblicher Kopplungsarchitekturen vorgestellt. In Kapitel 4 wird die Entwicklungsmethodik im fachlichen Kontext des Supply-Chain-Managements am Beispiel des Konzepts Vendor-Managed-Inventory (VMI) angewendet. Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick auf zukünftige Forschungsaktivitäten in Kapitel 5.

2 Überbetriebliche Integration von Anwendungssystemen mittels Kopplungsarchitekturen

Eine inner- oder überbetriebliche Integration von AwS erfordert deren Kopplung gemäß einer Kopplungsarchitektur. Unter einer **Kopplungsarchitektur** wird eine Projektion auf die AwS-Architekturen von zwei oder mehreren integrierten AwS verstanden. Sie beschreibt in Form eines Bauplans alle für die Integration relevanten Elemente sowie die Beziehungen zwischen diesen Elementen [SMFS01]. Bei der Beschreibung von Kopplungsarchitekturen wird im Folgenden zwischen dem Kern und dem Kopplungs-Subsystem eines AwS unterschieden. Das Kopplungs-Subsystem enthält diejenigen Elemente eines AwS, die ausschließlich der Kopplung mit anderen AwS dienen. Der AwS-Kern enthält Elemente, die für die Kopplung der AwS nicht relevant sind oder nicht ausschließlich im Kopplungskontext genutzt werden. Die Kopplungsarchitektur eines AwS spezifiziert dessen Kopplungs-Subsystem sowie die Elemente des AwS-Kerns, die im Kopplungskontext genutzt werden.

Beispiel:

Ein AwS benötigt Zugriff auf eine Datenbanktabelle eines R/3-Systems. Das Kopplungs-Subsystem des R/3-Systems stellt die hierzu erforderlichen Zugriffsmechanismen in Form eines BAPI bereit [SAP02]. Die genutzte Datenbanktabelle selbst ist Teil des R/3-AwS-Kerns. Die Kopplungsarchitektur des R/3-Systems umfasst hier die Zugriffsmechanismen als Teil des Kopplungs-Subsystems und die Tabelle als Teil des AwS-Kerns.

Anhand der mit einer Kopplungsarchitektur verfolgten Ziele werden drei Klassen von Architekturen unterschieden: die ereignisorientierten, die datenorientierten und die funktionsorientierten Kopplungsarchitekturen [SMFS02].

- **Ereignisorientierte Kopplungsarchitekturen** zielen auf die Übertragung von Ereignissen und zugehörigen Daten¹ zwischen AwS durch Austausch von Nachrichten in Form einer

¹ Der Begriff Daten wird in diesem Beitrag im Sinne von Dateninstanzen verwendet.

losen Kopplung [FeSi01, 225]. Konkrete Kopplungsarchitekturen dieser Art unterscheiden sich z. B. hinsichtlich der verwendeten Nachrichtenformate und Kommunikationsprotokolle.

- **Datenorientierte Kopplungsarchitekturen** bezwecken die Manipulation gemeinsamer Daten mehrerer AwS in Form einer engen Kopplung der auf den Daten operierenden Funktionen [FeSi01, 225]. Abhängig von ggf. verwendeten Kopien der gemeinsamen Daten wird zwischen redundanter und nicht-redundanter Datenhaltung unterschieden.
- **Funktionsorientierte Kopplungsarchitekturen** ermöglichen die gemeinsame Nutzung von Funktionen und ggf. zugehörigen Daten durch mehrere AwS. Entsprechend ist bezüglich einer gemeinsamen Funktion zu klären,
 - ob Programmmodule, welche die gemeinsame Funktion realisieren, redundant oder nicht-redundant gehalten werden,
 - ob die gemeinsame Funktion in separaten Prozessen je AwS oder in einem gemeinsamen Prozess ausgeführt wird,
 - wie die gemeinsame Funktion aktiviert wird.

In Bezug auf die zugehörigen Daten sind zwei Szenarien denkbar, die sich darin unterscheiden, ob die Daten bei allen beteiligten AwS übereinstimmen oder nicht. Im ersten Fall treten vergleichbare Probleme wie bei einer datenorientierten Kopplungsarchitektur auf.

In Abbildung 1 sind exemplarisch die wesentlichen Subsysteme einer typischen datenorientierten Kopplungsarchitektur in der Subsystem-Notation der UML dargestellt.

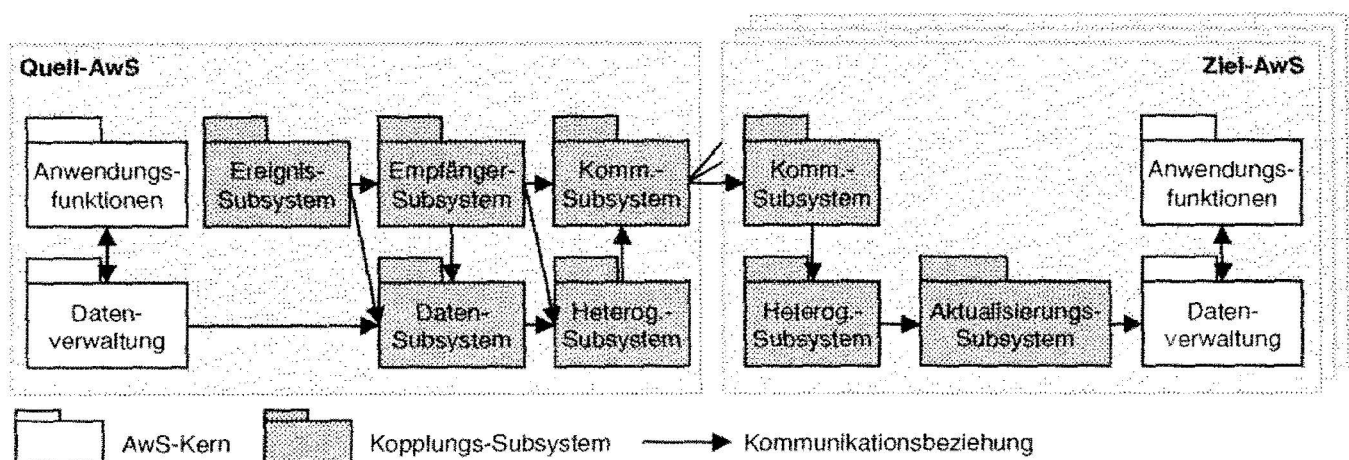


Abb. 1: Datenorientierte Kopplungsarchitektur mit redundanter Datenhaltung [SMFS02]

Wesentliches Merkmal dieser Architektur ist die redundante Speicherung der gemeinsamen Daten mit dem Ziel einer hohen Betriebsunabhängigkeit der AwS. Unter Betriebsunabhängigkeit wird die Unabhängigkeit der zu koppelnden AwS in der Betriebsphase verstanden. Beispiels-

weise sollte eine Nachricht an ein AwS zu einem beliebigen Zeitpunkt übermittelt werden können, auch dann, wenn dieses wegen Wartungsarbeiten nicht verfügbar ist. Es wird davon ausgegangen, dass nur das als Quell-AwS bezeichnete AwS auf die Daten schreibend zugreift. Aus der redundanten Speicherung folgt das Problem der Gewährleistung der Datenkonsistenz (siehe z. B. [Krem99, 107 ff.]). Dieses wird durch entsprechende Mechanismen behandelt.

Die beschriebene Kopplungsarchitektur folgt dabei dem Push-Prinzip, wobei das **Ereignis-Subsystem** des Quell-AwS bestimmt, wann eine Propagation zu erfolgen hat. Das hier dargestellte Ereignis-Subsystem stößt die Propagation zu definierten Zeitpunkten an.

Das **Empfänger-Subsystem** übernimmt die Verteilung der Daten auf die Ziel-AwS. Hier wird davon ausgegangen, dass die zu benachrichtigenden Ziel-AwS allein anhand des Ereignistyps bestimmt werden.

Das **Daten-Subsystem** ermittelt die zu übermittelnden Daten. In einem Propagationslauf werden entweder alle Daten oder nur geänderte Daten übertragen. Im letzteren Fall werden zusätzliche Mechanismen zu deren Bestimmung benötigt.

Das **Heterogenitäts-Subsystem** dient der Überwindung der fachlichen Heterogenität zwischen Quell- und Ziel-AwS. In Abbildung 1 wird davon ausgegangen, dass das Quell-AwS die Daten in ein spezielles, von den AwS-internen Formaten abweichendes Übertragungsformat umsetzt und die Ziel-AwS dieses Übertragungsformat wiederum in die dort benötigten Formate umformen.

Das **Kommunikations-Subsystem** des Quell-AwS übernimmt die Übertragung des Ereignisses und der Daten vom Quell- zu den Ziel-AwS und überwindet gleichzeitig die technische Heterogenität zwischen den AwS.

In den Ziel-AwS werden Daten und Ereignisse durch das jeweilige **Kommunikations-Subsystem** entgegengenommen. Das **Heterogenitäts-Subsystem** setzt die übertragenen Daten in die Formate des Ziel-AwS um.

Das **Aktualisierungs-Subsystem** übernimmt in den Ziel-AwS die Aktualisierung der Daten unter Verwendung der von den Datenverwaltungs-Subsystemen angebotenen Schnittstellen. Dabei muss insbesondere gewährleistet werden, dass die Daten in der Reihenfolge geändert werden, in der sie auch im Quell-AwS geändert wurden. Datenbanktransaktionen im Quell-AwS sind auch in den Ziel-AwS zu berücksichtigen.

3 Konzept einer Entwicklungsmethodik

Im folgenden Kapitel wird das Konzept einer Entwicklungsmethodik zur Gestaltung von AWS-Kopplungsarchitekturen für überbetriebliche Geschäftsprozesse vorgestellt. Die Methodik wird anhand der verwendeten Modellebenen und des Vorgehens bei der Entwicklung von Kopplungsarchitekturen beschrieben.

Modellebenen

Es werden die folgenden Modellebenen unterschieden (Abb. 2):

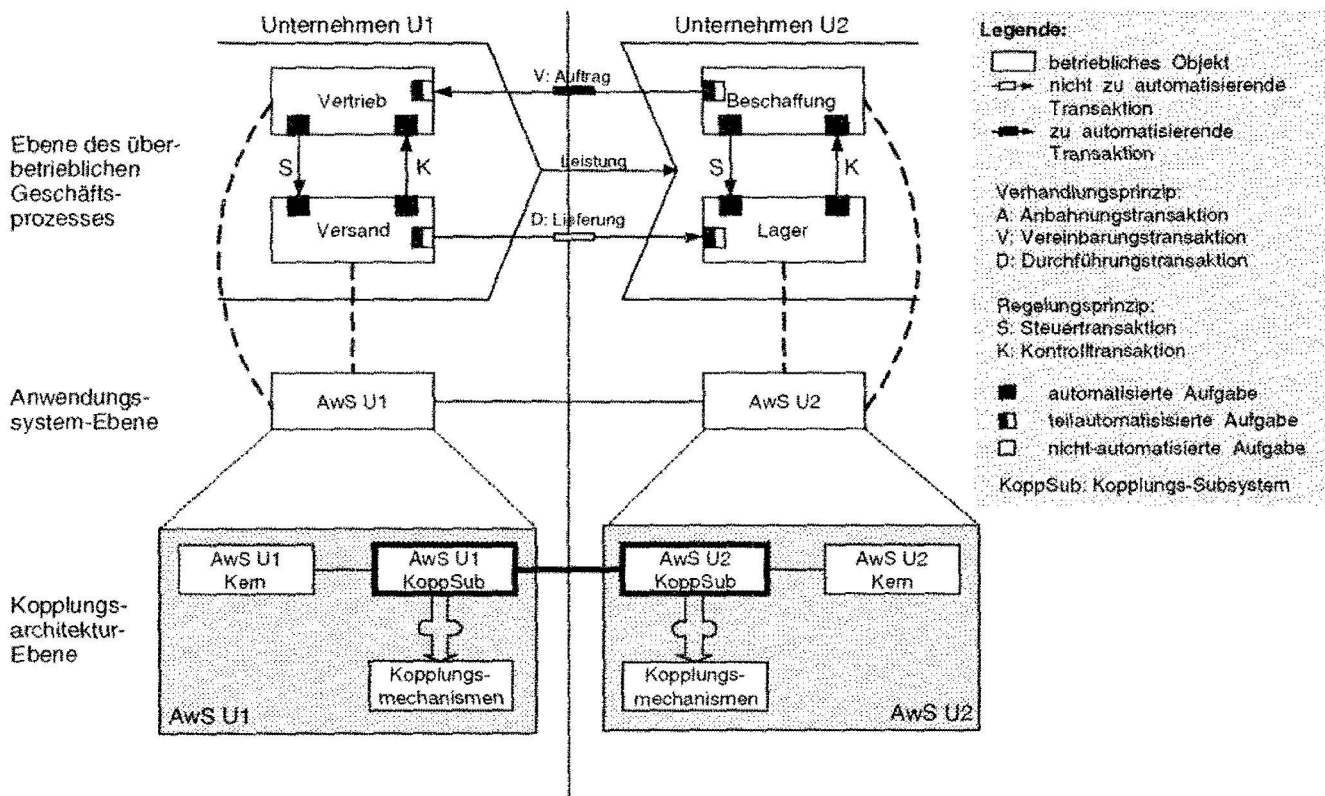


Abb. 2: Modellebenen der Entwicklungsmethodik

- **Ebene des überbetrieblichen Geschäftsprozesses:** Ausgangspunkt der Entwicklung einer Kopplungsarchitektur ist ein überbetrieblicher Geschäftsprozess, der hier anhand des Semantischen Objektmodells (SOM) beschrieben wird [FeSi95; FeSi01, 179 ff.]. Es liegt die Metapher eines verteilten Systems aus autonomen und lose gekoppelten betrieblichen Objekten zugrunde, die sich mittels Transaktionen koordinieren [FeSi01, 181 f.]. Um eine Zuordnung der Aufgaben der betrieblichen Objekte zu Aufgabenträgern vornehmen zu können, werden die Aufgaben als automatisiert, teilautomatisiert oder nicht-automatisiert eingestuft. Die für die Entwicklung von überbetrieblichen Kopplungsarchitekturen relevanten über-

betrieblichen Transaktionen werden entweder als zu automatisierend oder als nicht zu automatisierend eingestuft.

- **Anwendungssystem-Ebene:** Diese Ebene beschreibt die Menge der zu koppelnden AwS. Ein AwS führt automatisierte Aufgaben oder Aufgabenanteile des Geschäftsprozesses aus. Nicht-automatisierte Aufgaben und ihre Zuordnung zu personellen Aufgabenträgern werden, ebenso wie Aufgaben der Leistungserstellung physischer Güter und Dienstleistungen, nicht weiter verfolgt.
- **Kopplungsarchitektur-Ebene:** Auf dieser Ebene wird die Kopplung der AwS in Form von Kopplungsarchitekturen beschrieben. Dabei werden die Kopplungs-Subsysteme einschließlich der dafür verwendeten Kopplungsmechanismen spezifiziert. Kopplungsmechanismen stellen Dienste und Kommunikationsprotokolle für die Realisierung der Kopplungsarchitekturen bereit. Sie nutzen hierfür ihrerseits Dienste von Betriebssystemen, Kommunikationsplattformen, etc. Da Kopplungsmechanismen sowohl systemtechnische als auch fachliche Aspekte beinhalten, geben sie einen engen Rahmen für die Entwicklung einer Kopplungsarchitektur vor [SMFS02].

Vorgehen

Bei der Festlegung der Vorgehensweise zur Entwicklung einer Kopplungsarchitektur ist der zugehörige Entscheidungsraum zu berücksichtigen. Dieser umfasst Entscheidungsebenen bezüglich der Gestaltung des überbetrieblichen Geschäftsprozesses, der Neuentwicklung oder Anpassung von AwS, sowie der Entwicklung der AwS-Kopplung. Je nach Situation sind entweder alle drei, die letzten beiden oder nur die letzte Ebene Teil des Entscheidungsraums. Im Folgenden wird der weitaus häufigste Fall angenommen, dass der zu behandelnde überbetriebliche Geschäftsprozess und die dafür zu verwendenden AwS bereits vorliegen. Die erforderliche Entwicklung beschränkt sich auf die Kopplung bestehender AwS.

Auch bei dieser Beschränkung bleibt die Entwicklung von überbetrieblichen Kopplungsarchitekturen ein komplexes Simultanplanungsproblem. Zur Komplexitätsbewältigung wird das Simultanplanungsproblem in sequentiell zu durchlaufende Schritte zerlegt. Jeder Schritt des Entwicklungsprozesses schränkt durch Festlegung von Kopplungsmerkmalen den Lösungsraum bezüglich geeigneter Kopplungsarchitekturen ein. Falls in einer solchen Schrittfolge keine Lösung mit einem angestrebten Zielerreichungsgrad gefunden werden kann, wird der Lösungsraum durch Backtracking und Neufestlegung verändert. Das Vorgehen zur Entwicklung einer Kopplungsarchitektur unterteilt sich in die folgenden Schritte:

1. **Modellierung des überbetrieblichen Geschäftsprozesses:** Zunächst wird der gegebene überbetriebliche Geschäftsprozess in einem Geschäftsprozessmodell erfasst. Weiterhin wird

der für die Entwicklung von Kopplungsarchitekturen relevante Fokus auf das Geschäftsprozessmodell anhand der überbetrieblichen Transaktionen und der Menge, der an diesen Transaktionen beteiligten und somit potenziell kopplungsrelevanten, betrieblichen Objekte abgegrenzt.

2. **Kartierung der AwS:** Die Aufgaben der potenziell kopplungsrelevanten betrieblichen Objekte werden hinsichtlich ihres Automatisierungsgrades als automatisiert, teilautomatisiert oder nicht-automatisiert eingestuft. Automatisierte und teilautomatisierte Aufgaben werden gegebenen AwS eindeutig zugeordnet. Gleichzeitig werden die erforderlichen Kopplungsbeziehungen zwischen den AwS aus den Informationsbeziehungen zwischen den Aufgaben abgeleitet. Die überbetrieblichen Transaktionen werden als zu automatisierend oder als nicht zu automatisierend gekennzeichnet. Zu automatisierende Transaktionen sind durch eine Kopplungsarchitektur zu unterstützen.
3. **Identifikation von Aufgabenintegrations-Mustern im Geschäftsprozessmodell:** Im überbetrieblichen Geschäftsprozessmodell werden Aufgabenintegrations-Muster identifiziert. Diese grenzen einen Integrationsbereich in Form einer Menge von Transaktionen, einschließlich der zugehörigen Aufgaben, ab und bestimmen die Kopplungsarchitekturklasse, die zur Unterstützung dieses Musters geeignet ist.
4. **Spezifikation von Anforderungen an die AwS-Integration unter Berücksichtigung von Aufgabenintegrations-Mustern:** Für jedes identifizierte Aufgabenintegrations-Muster werden die Anforderungen an die Integration der AwS spezifiziert. Hierbei wird auf die Integrationsziele hinsichtlich der Integrationsmerkmale Redundanz, Verknüpfung, Konsistenz und Zielorientierung Bezug genommen [Fers92].
5. **Beschreibung der AwS:** Die im Rahmen der Kartierung abgegrenzten AwS werden hinsichtlich ihrer relevanten Eigenschaften beschrieben. Relevant sind alle Eigenschaften, die den Prozess des Entwurfs einer Kopplungsarchitektur beeinflussen können. Hierbei handelt es sich um Eigenschaften der zur Verfügung stehenden Kopplungsmechanismen, der AwS-Kern-Schnittstellen und der im Rahmen der Integration relevanten Teile des AwS-Kerns. Eine vollständige Erfassung der relevanten Eigenschaften ist in diesem Schritt aus Gründen des Aufwands nicht immer sinnvoll. Vielmehr können die in diesem Schritt erfassten Eigenschaften im Rahmen des Entwurfs der Kopplungsarchitektur (Schritt 6) bedarfsgerecht vervollständigt werden.
6. **Entwurf der Kopplungsarchitektur:** Für jedes Aufgabenintegrations-Muster wird für das konkrete Kopplungsszenario eine Variante der durch das Muster bestimmten Kopplungsarchitekturklasse anhand der spezifizierten Anforderungen und der kopplungsrelevanten Eigenschaften der zu koppelnden AwS ausgewählt und konkretisiert.

Die einzelnen Schritte werden nachfolgend bei der Anwendung der Entwicklungsmethodik im Supply-Chain-Management detailliert beschrieben.

4 Anwendung der Entwicklungsmethodik im Supply-Chain-Management

Im Folgenden wird die in Kapitel 3 skizzierte Entwicklungsmethodik auf den Bereich des Supply-Chain-Managements angewendet. Hierzu werden die einzelnen Schritte der Entwicklungsmethodik am Beispiel des VMI dargelegt.

Im Rahmen des VMI übernimmt der Lieferant die Bestandsverantwortung für den Lagerbestand seiner Artikel bei ausgewählten Kunden. Um diese Dispositionsdienstleistung erbringen zu können, benötigt der Lieferant Kenntnis des aktuellen Kundenlagerbestandes sowie Informationen über die zukünftigen Bedarfe des Kunden. Wesentlicher Vorteil des Konzepts ist eine Senkung der Gesamtlagerbestände beim Kunden und beim Lieferanten und damit verbunden auch eine Senkung der Kosten der Geschäftspartner [CoGö01, 117; LaCP98, 5; Chris98, 195; KnMZ00, 59].

4.1 Modellierung des überbetrieblichen Geschäftsprozesses

Im ersten Schritt wird der überbetriebliche Geschäftsprozess zwischen zwei Unternehmen (Lieferant, Kunde) gemäß dem Konzept des VMI modelliert (Abb. 3). Ein Geschäftsprozessmodell wird in der SOM-Methodik durch zwei verschiedene Sichten spezifiziert. Das **Interaktionsschema (IAS)** beschreibt die Struktur von Geschäftsprozessen durch die Modellierung betrieblicher Objekte und der sie verbindenden Transaktionen, während das **Vorgangs-Ereignis-Schema (VES)** das zugehörige Verhalten und somit dynamische Aspekte erfasst [FeSi01, 183 f.]. Im Rahmen der Entwicklung von Kopplungsarchitekturen sind primär die strukturellen Beziehungen von Interesse, so dass sich die folgende Beschreibung des überbetrieblichen Geschäftsprozesses auf das IAS beschränkt.

Ausgangspunkt der Modellbildung ist die Abgrenzung der für die Untersuchung relevanten Diskurswelt und die Identifikation des relevanten Teils der Umwelt innerhalb der betrachteten betrieblichen Realität [FeSi01, 5]. Sachziel des VMI ist es, den Beschaffungsbereich des Kunden mit dem Absatzbereich des Lieferanten so zu integrieren, dass der Lieferant die Bestandsverantwortung für den Kunden übernehmen kann. Dazu benötigt er Zugriff sowohl auf aktuelle Lagerbestandsdaten, als auch auf Prognosen des Kunden hinsichtlich seines zukünftigen Bedarfes

[SiKS00, 82, 137; KnMZ00, 59; Chris98, 195]. Somit umfasst die Diskurswelt den Absatzbereich des Lieferanten sowie den Beschaffungsbereich des Kunden. Die daran angrenzenden Teile der betrieblichen Realität, z. B. die Produktion, sind Teil der nicht näher spezifizierten Umwelt.

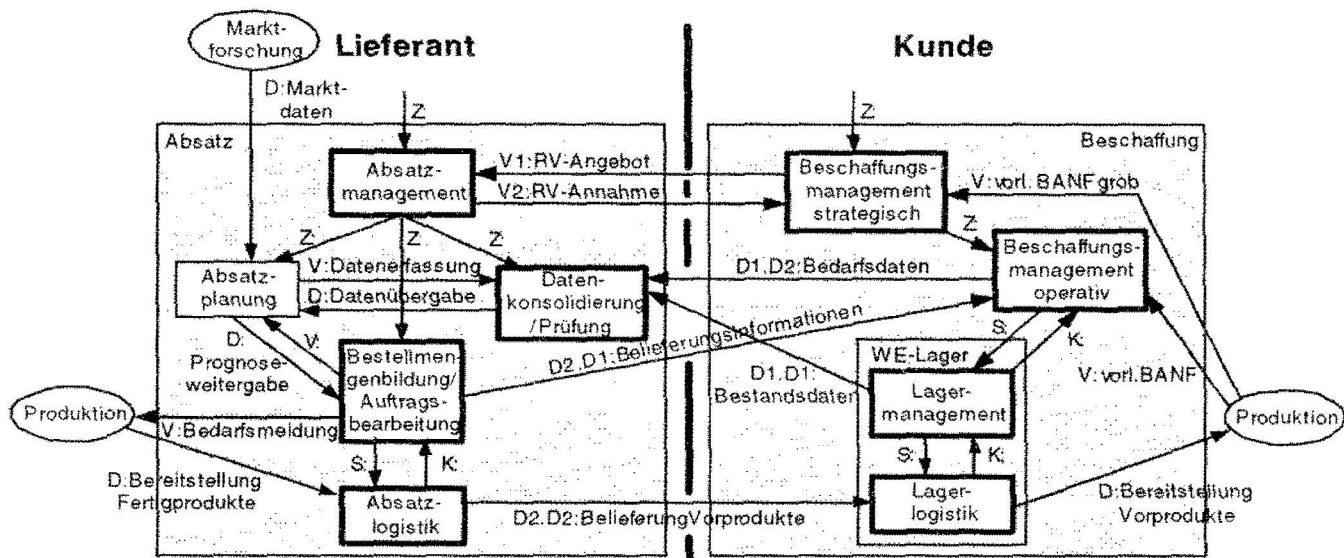


Abb. 3: Interaktionsschema des VMI-Geschäftsprozesses

Grundlegender Bestandteil des VMI ist die Vereinbarung eines Rahmenvertrages, in dem definiert wird, für welche Artikel der Lieferant die Bestandsverantwortung übernehmen soll. Weiterer Bestandteil dieses Vertrages ist die Vereinbarung einzuhaltender Servicelevel, z. B. die minimalen bzw. maximalen Bestandshöhen im Lager des Kunden oder die minimalen / maximalen Nachbelieferungsmengen je Auftrag. [Mau00, 144 f.; Holm98, 129] Dieser Rahmenvertrag wird durch die beiden strategisch ausgerichteten Objekte *Absatzmanagement* und *Beschaffungsmanagement strategisch* über die Transaktionen *V1: RV-Angebot* und *V2: RV-Aannahme* vereinbart. Wesentliche Informationsgrundlage für diese Verhandlung sind auf Seite des Kunden die von der *Produktion* gemeldeten längerfristigen Bedarfe, die als Bedarfsanforderungen (Transaktion *V: vorl. BANF grob*) übermittelt werden. Im Rahmen der Transaktion *V: vorl. BANF* vereinbaren die beiden Objekte *Beschaffungsmanagement operativ* und *Produktion* die rechtzeitige Bereitstellung benötigter Vorprodukte. Diese Informationen bezüglich des zukünftigen Bedarfes werden durch die Transaktion *D1.D2: Bedarfsdaten* dem Objekt *Datenkonsolidierung/Prüfung* des Lieferanten bereitgestellt. Weiterhin werden diesem Objekt die aktuellen Lagerbestandsdaten des Kunden über die Transaktion *D1.D1: Bestandsdaten* bereitgestellt. Aufgabe der *Datenkonsolidierung/Prüfung* ist es diese Daten hinsichtlich ihrer Vollständigkeit und Plausibilität zu überprüfen und der *Absatzplanung* bereitzustellen [BeUV00, 192; KnMZ00, 165]. Auf dieser Grundlage erstellt die *Absatzplanung* unter Berücksichtigung weiterer marktbezogener Daten eine Prognose des zukünftigen Bedarfes des Kunden. Diese Prognose wird

durch das Objekt *Bestellmengenbildung/Auftragsbearbeitung* genutzt, um unter Beachtung der eigenen Bestandssituation, des geplanten eigenen Produktionsprogramms, sowie der mit dem Kunden vereinbarten Servicelevel wirtschaftliche Liefermengen und -zeitpunkte festzulegen [Wald98, 140]. Sobald die Liefermengen und -zeitpunkte festgelegt sind, werden diese Informationen über die Transaktion *D2.D1 Belieferungsinformationen* an das *Beschaffungsmanagement operativ* übergeben. Weiterhin wird die *Absatzlogistik* des liefernden Unternehmens angewiesen die entsprechende Ware an den Kunden zu versenden. Die Übergabe der Ware erfolgt über die Transaktion *D2.D2 Belieferung Vorprodukte*. Aufgabe des Objektes *Beschaffungsmanagement operativ* ist es einen, den Belieferungsinformationen entsprechenden, Auftrag im eigenen AWS anzulegen, sowie das *Lagermanagement* anzuweisen, die Lieferung anzunehmen und zu kontrollieren. Das *Lagermanagement* ist für die Verwaltung der Bestandsdaten sowie die Steuerung der *Lagerlogistik* zuständig, welche u. a. die Aufgaben Annehmen und Zwischenlagern der Ware durchführt.

Tab. 1: überbetriebliche Transaktionen und potenziell kopplungsrelevante Objekte

Überbetriebliche Transaktionen	Potenziell kopplungsrelevante Objekte
V: Vereinbarung Rahmenvertrag	
V1: RV-Angebot	Beschaffungsmanagement strategisch Absatzmanagement
V2: RV-Annahme	Absatzmanagement Beschaffungsmanagement strategisch
D: Erfüllung Rahmenvertrag	
D1: Bestands- und Bedarfsdaten	
D1.D1: Bestandsdaten	Lagermanagement Datenkonsolidierung/Prüfung
D1.D2: Bedarfsdaten	Beschaffungsmanagement operativ Datenkonsolidierung/Prüfung
D2: Belieferung	
D2.D1: Belieferungsinformationen	Bestellmengenbildung/Auftragsbearbeitung Beschaffungsmanagement operativ
D2.D2: Belieferung Vorprodukte	Absatzlogistik Lagerlogistik

Im Anschluss an die Modellierung der Diskurswelt wird der für die Kopplung relevante Teil des Geschäftsprozessmodells abgegrenzt. Dazu werden überbetriebliche Transaktionen identifiziert, an deren Durchführung Objekte verschiedener Unternehmen beteiligt sind. Die beteiligten

Objekte werden als potenziell relevant für die Kopplung von AwS identifiziert. Zur Verdeutlichung sind diese Objekte im Geschäftsprozessmodell durch breitere Rahmen gekennzeichnet. In Tabelle 1 sind die für das vorliegende Beispiel identifizierten Transaktionen und Objekte zusammengefasst.

4.2 Kartierung der Anwendungssysteme

Auf der Grundlage des Geschäftsprozessmodells erfolgt die Kartierung der gegebenen AwS. Unter Kartierung von AwS wird „die Beschreibung einer Anwendungssystemzuordnung zu Geschäftsprozessen durch Einordnung der Anwendungssysteme in Geschäftsprozessmodelle“ [Krum97, 137] verstanden.

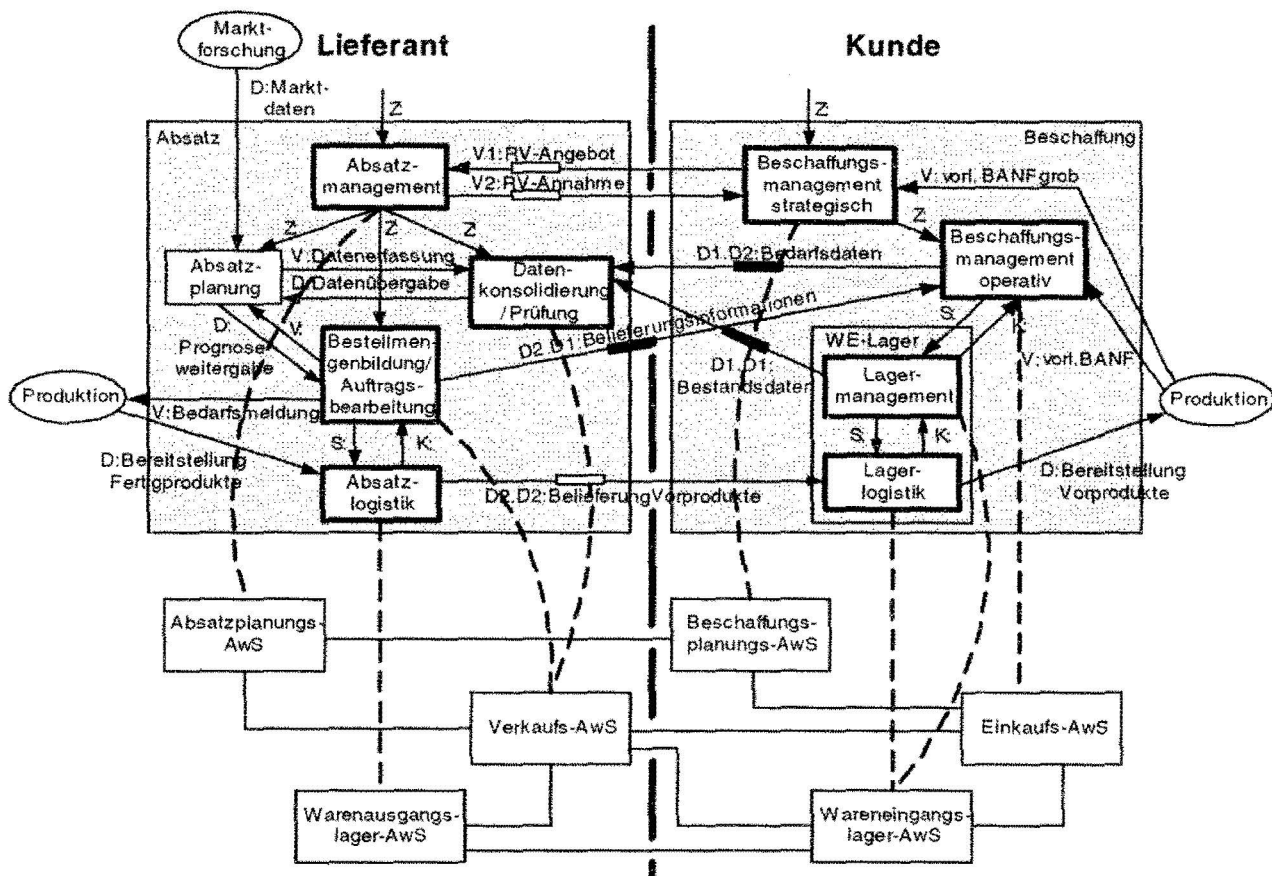


Abb. 4: Kartierung der AwS

Hierzu werden zunächst die Aufgaben der potenziell kopplungsrelevanten Objekte hinsichtlich ihres Automatisierungsgrades als automatisiert, teilautomatisiert oder nicht-automatisiert eingestuft. Anschließend wird jedem Objekt mit voll- oder teilautomatisierten Aufgaben genau ein AwS zugeordnet. Ein AwS führt dabei die automatisierten Aufgabenanteile eines oder

mehrerer Objekte aus. Ist bei einem Objekt eine eindeutige Zuordnung nicht möglich, wird das Objekt entsprechend verfeinert.

Im Rahmen des VMI lassen sich beispielsweise die in Abbildung 4 dargestellten AwS identifizieren. Es wird vereinfachend angenommen, dass sämtliche Aufgaben der betrieblichen Objekte zumindest teilautomatisiert sind.

Im Anschluss werden die überbetrieblichen Transaktionen des Geschäftsprozessmodells als zu automatisierend oder als nicht zu automatisierend eingestuft. Die zu automatisierenden Transaktionen sind durch Kopplungsarchitekturen zu unterstützen. Sie stellen den Fokus der nachfolgenden Betrachtungen dar.

Im vorliegenden VMI-Geschäftsprozessmodell sind die drei Transaktionen *D1.D1: Bestandsdaten*, *D1.D2 Bedarfsdaten* und *D2.D1: Belieferungsinformationen* zu automatisieren (Abb. 4). Es werden nur die an der Übertragung dieser Transaktionen beteiligten AwS weiter betrachtet. Hierbei handelt es sich auf Seite des Lieferanten um das Verkaufs-AwS und auf Seite des Kunden um das Einkaufs-AwS und das Wareneingangslager-AwS.

4.3 Identifikation von Aufgabenintegrations-Mustern im Geschäftsprozessmodell

In diesem Schritt werden im kopplungsrelevanten Teil des Geschäftsprozessmodells - bestehend aus überbetrieblichen Transaktionen und potenziell kopplungsrelevanten, betrieblichen Objekten - Aufgabenintegrations-Muster identifiziert. Aufgabenintegrations-Muster sind Teilstrukturen eines Geschäftsprozessmodells, anhand derer die im jeweiligen Fall geeignete Kopplungsarchitekturklasse bestimmt wird. Ein Aufgabenintegrations-Muster umfasst Aufgaben der betroffenen Objekte und Transaktionen sowie deren Beziehungen und grenzt jeweils einen durch AwS zu integrierenden Bereich innerhalb des Geschäftsprozessmodells ab. Jedem Aufgabenintegrations-Muster ist eine Kopplungsarchitekturklasse zugeordnet. Das Aufgabenintegrations-Muster *Gemeinsame Nutzung von Aufgabenobjekt-Instanzen* wird beispielsweise durch eine datenorientierte, das Aufgabenintegrations-Muster *Reihenfolgebeziehung zwischen Aufgabendurchführungen* durch eine ereignisorientierte Kopplungsarchitektur unterstützt.

Im VMI-Geschäftsprozessmodell lassen sich folgende Aufgabenintegrations-Muster identifizieren:

- **Gemeinsame Nutzung von Aufgabenobjekt-Instanzen**

Die Transaktionen *D1.D1: Bestandsdaten* und *D1.D2: Bedarfsdaten* genügen jeweils dem Aufgabenintegrations-Muster *Gemeinsame Nutzung von Aufgabenobjekt-Instanzen*. Ein

Aufgabenobjekt beschreibt die zu einer Aufgabe gehörigen Attribute und Attributwerte eines betrieblichen Systems. Dabei wird zwischen dem Aufgabenobjekt-Typ (Attribute) und den Aufgabenobjekt-Instanzen (Attributwerte) unterschieden. Das vorliegende Muster ist zum einen dadurch charakterisiert, dass die im Rahmen einer Transaktion übermittelten Zustände der Aufgabenobjekt-Instanzen ggf. auch nach Durchführung der Transaktion weiterhin durch das sendende Objekt manipulierbar bleiben und zum anderen dadurch, dass die Durchführung der Transaktion im empfangenden Objekt keine Verrichtung außer der obligatorischen Eingangsverarbeitung anstößt.

- **Reihenfolgebeziehungen zwischen Aufgabendurchführungen**

Des Weiteren lässt sich ein Aufgabenintegrations-Muster *Reihenfolgebeziehung zwischen Aufgabendurchführungen* für die Transaktion *D2.D1: Belieferungsinformationen* identifizieren. Dieses ist dadurch charakterisiert, dass die im Rahmen der Transaktion übermittelten Zustände der Aufgabenobjekt-Instanzen nach Durchführung der Transaktion durch das sendende Objekt nicht mehr manipulierbar sind und dass die Durchführung der Transaktion im empfangenden Objekt eine Verrichtung anstößt, die über die obligatorische Eingangsverarbeitung hinausgeht.

Das weitere Vorgehen der Entwicklungsmethodik wird am Beispiel des Aufgabenintegrations-Musters *Gemeinsame Nutzung von Aufgabenobjekt-Instanzen* für die Transaktion *D1.D1: Bestandsdaten* aufgezeigt.

4.4 Spezifikation von Anforderungen an die Anwendungssystem-Integration unter Berücksichtigung von Aufgabenintegrations-Mustern

Für jedes identifizierte Aufgabenintegrations-Muster werden die Anforderungen an die Integration der AwS spezifiziert. Hierbei werden auch Anforderungen an die Integration, die sich aus den Aufgabenintegrations-Mustern ableiten lassen, berücksichtigt. Die Anforderungen werden in Form von Integrationszielen festgelegt. Die Integrationsziele nehmen auf ausgewählte Struktur- und Verhaltensmerkmale der fachlichen Ebene von AwS – die Integrationsmerkmale – Bezug und legen Zielausprägungen der Integrationsmerkmale fest [Fers92]. Es werden die Strukturmerkmale Redundanz der Systemkomponenten (differenziert in Daten- und Funktionsredundanz) und Verknüpfung der Komponenten, sowie die Verhaltensmerkmale Konsistenz und Zielorientierung der Teilaufgaben in Bezug auf die Gesamtaufgabe unterschieden [Fers92; FeSi01, 217 ff.].

Für das Aufgabenintegrations-Muster *Gemeinsame Nutzung von Aufgabenobjekt-Instanzen* der Transaktion *D1.D1: Bestandsdaten* sind v. a. die Integrationsmerkmale Datenredundanz, Verknüpfung und Konsistenz anwendbar.

Für die Übermittlung der Bestandsdaten vom Kunden an den Lieferanten im Rahmen des VMI wird eine hohe **Betriebsunabhängigkeit** gefordert. Sie wirkt sich v. a. auf die Integrationsmerkmale Datenredundanz und Verknüpfung aus. Eine hohe Betriebsunabhängigkeit kann durch eine redundante Datenhaltung zwischen den AWS erreicht werden. Hinsichtlich der Verknüpfung kann eine hohe Betriebsunabhängigkeit ein Kommunikationssystem erfordern, das das sendende und das empfangende AWS entkoppelt.

Aktualitätsanforderungen hinsichtlich der Daten wirken sich auf das Integrationsmerkmal Konsistenz aus. Die auf Seite des Lieferanten auf den Bestandsdaten operierende Aufgabe *Absatzplanung* benötigt lediglich tagesaktuelle Daten. Temporäre Inkonsistenzen zwischen den vom Kunden und den vom Lieferanten verwendeten Aufgabenobjekt-Instanzen können somit toleriert werden. Weiterhin gelten die allgemeinen Anforderungen hinsichtlich der **Einhaltung semantischer Integritätsbedingungen** in Bezug auf die gemeinsam genutzten Bestandsdaten [Voss00, 147 ff.]. Die Wahrung dieser Integritätsbedingungen wird vereinfacht aufgrund der Tatsache, dass nur eine der beiden an der Transaktion *D1.D1: Bestandsdaten* beteiligten Aufgaben die gemeinsamen Aufgabenobjekt-Instanzen modifiziert. Die Bestandsdaten werden ausschließlich vom Kunden modifiziert, der Lieferant greift lediglich lesend auf die Bestandsdaten zu.

4.5 Beschreibung der Anwendungssysteme

Die im Rahmen der Kartierung festgelegten AWS werden hinsichtlich ihrer relevanten Eigenschaften beschrieben. Diese Beschreibungen gehen als Restriktionen in den Entwurf der Kopplungsarchitektur ein. Sie definieren den Lösungsraum innerhalb dessen der Entwurf erfolgt. Eine Änderung von Restriktionen ist grundsätzlich möglich, bedeutet aber einen Mehraufwand bei der Entwicklung von Kopplungsarchitekturen durch die erforderliche Wiederholung von Entwicklungsschritten.

Da sich die Diskussion auf die Betrachtung des Aufgabenintegrations-Musters *Gemeinsame Nutzung von Aufgabenobjekt-Instanzen* der Transaktion *D1.D1: Bestandsdaten* beschränkt, werden im Folgenden nur die von dieser Transaktion betroffenen AWS untersucht. Dies ist auf Seite des Kunden das Wareneingangslager-AWS und auf Seite des Lieferanten das Verkaufs-AWS. Die Beschreibung der AWS beschränkt sich auf die Kopplungsmechanismen, da diese für den Entwurf von zentraler Bedeutung sind. Aufgrund des betrachteten Aufgabenintegrations-Musters werden nur Kopplungsmechanismen beschrieben, die zur Unterstützung einer daten-

orientierten Kopplungsarchitektur geeignet sind. Die Beschreibung der Kopplungsmechanismen eines AwS erfolgt anhand der unterstützbaren Subsysteme des Kopplungs-Subsystems. Diese strukturieren die Beschreibung dergestalt, dass für die einzelnen Subsysteme angegeben wird, wie sie durch die Kopplungsmechanismen unterstützt werden.

Das **Wareneingangslager-AwS** bietet einen Kopplungsmechanismus, der zur Realisierung von datenorientierten Kopplungsarchitekturen mit redundanter Datenhaltung geeignet ist. Der Kopplungsmechanismus unterstützt die Subsysteme Ereignis-Subsystem, Empfänger-Subsystem, Daten-Subsystem, Heterogenitäts-Subsystem und Kommunikations-Subsystem. Der Kopplungsmechanismus geht davon aus, dass die gemeinsam genutzten Daten von nur einem der AwS verwaltet werden. Die anderen AwS greifen auf die Daten nur lesend zu. Mechanismen zur Behandlung von parallelen Schreibzugriffen auf Datenkopien werden folglich nicht angeboten (siehe z. B. [Rahm94, 163 ff.]). Die einzelnen Subsysteme werden wie folgt unterstützt:

- Das **Ereignis-Subsystem** wird in der AwS-Kern-sensitiven Variante unterstützt. Es kann somit in Abhängigkeit des Eintretens von Ereignissen im AwS-Kern entschieden werden, wann eine Propagation von Daten durchgeführt werden muss. Zusätzlich wird auch eine Propagation zu definierten Zeitpunkten unterstützt.
- Das **Empfänger-Subsystem** wird nur in der nicht-inhaltssensitiven Variante unterstützt. Es kann somit nicht auf die Inhalte der zu übermittelnden Daten zugreifen, um potenzielle Empfänger zu bestimmen. Die Ermittlung der Empfänger erfolgt nur über den Typ des Ereignisses.
- Zur Realisierung des **Daten-Subsystems** sind Mechanismen vorhanden, die die seit dem letzten Propagationslauf geänderten Daten ermitteln können. Alternativ kann ein Daten-Subsystem realisiert werden, dass alle zu propagierenden Daten aus dem AwS-Kern ermittelt. Der erste Fall wird als Delta-Propagation, der zweite als vollständige Propagation bezeichnet.
- Der Kopplungsmechanismus erlaubt die Realisierung eines **Heterogenitäts-Subsystems**, das die Umsetzung der internen Datenrepräsentation in den RosettaNet-Standard durchführt. Es werden lediglich die im RosettaNet-Standard enthaltenen Nachrichten verwendet. Auf eine Unterstützung der in Form der Partner Interface Processes (PIP) spezifizierten Geschäftsprozesse wird im betrachteten Beispiel verzichtet [Rose02].
- Das **Kommunikations-Subsystem** wird in Form des Kommunikationsprotokolls HTTP und des technischen Nachrichtenformats XML unterstützt.

Auf eine Darstellung der Elemente des Kopplungsmechanismus, die benötigt werden, wenn das Wareneingangslager-AwS als Ziel-AwS auftritt, wird verzichtet. Diese Elemente werden im

betrachteten Fall nicht benötigt, da die Bestandsdaten durch das Wareneingangslager-AwS verwaltet werden.

Das **Verkaufs-AwS** bietet nur einen sehr einfachen Kopplungsmechanismus an. Dieser beschränkt sich auf die Unterstützung von Aspekten der Kommunikation. Er kann daher nur zur Realisierung eines **Kommunikations-Subsystems** eingesetzt werden. Das Kommunikationsprotokoll und das technische Nachrichtenformat orientieren sich hierbei am ONC-RPC-Standard [IETF02].

4.6 Entwurf der Kopplungsarchitektur

Für das Aufgabenintegrations-Muster *Gemeinsame Nutzung von Aufgabenobjekt-Instanzen* der Transaktion *DI.DI: Bestandsdaten* wird nun innerhalb der bereits identifizierten Klasse der datenorientierten Kopplungsarchitekturen eine geeignete Variante ausgewählt und konkretisiert (Abb. 5). Bei der Auswahl einer Variante müssen zum einen die in Abschnitt 4.4 beschriebenen Anforderungen und zum anderen die in Abschnitt 4.5 beschriebenen kopplungsrelevanten Eigenschaften der AwS berücksichtigt werden.

Aufgrund des verfügbaren Kopplungsmechanismus auf Seite des Wareneingangslager-AwS wird für die datenorientierte Kopplungsarchitektur die Variante mit **redundanter Datenhaltung** ausgewählt. Hierfür spricht außerdem, wie in Abschnitt 4.4 bereits ausgeführt, die Forderung nach einer hohen Betriebsunabhängigkeit. Diese lässt sich u. a. durch eine redundante Haltung der Daten erreichen.

Das Kopplungs-Subsystem des **Wareneingangslager-AwS** wird folgendermaßen realisiert:

Wie in Kapitel 2 bereits ausgeführt wird die Propagation von Daten in einer solchen Architektur durch ein **Ereignis-Subsystem** angestoßen. Dabei können die Varianten Propagation zu definierten Zeitpunkten und Propagation beim Eintreten von fachlichen Ereignissen im AwS-Kern unterschieden werden. Der zur Verfügung stehende Kopplungsmechanismus unterstützt grundsätzlich beide Varianten. Aufgrund der relativ geringen Aktualitätsanforderungen und der daraus folgenden Toleranz hinsichtlich temporärer Inkonsistenzen kann auf die erste Variante zurückgegriffen werden. Es wird täglich ein Ereignis des Typs "Propagation VMI-Bestandsdaten" erzeugt.

Durch den verfügbaren Kopplungsmechanismus wird nur ein **Empfänger-Subsystem** unterstützt, das den Empfänger anhand des Ereignistyps ermittelt. Dies ist unproblematisch, da der einzige Empfänger für den Ereignistyp „Propagation VMI-Bestandsdaten“ der betrachtete Lieferant ist.

Der verfügbare Kopplungsmechanismus unterstützt sowohl ein **Daten-Subsystem** zur vollständigen Propagation, als auch zur Delta-Propagation. Das Daten-Subsystem wird in der

Variante einer vollständigen Propagation realisiert. Die Menge der pro Propagationslauf übertragenen Daten ist im betrachteten Beispiel gering. Dies und die nur täglich erfolgende Propagation rechtfertigen nicht die Realisierung eines Daten-Subsystems, das die seit dem letzten Propagationslauf geänderten Daten ermittelt.

Das **Heterogenitäts-Subsystem** setzt die interne Repräsentation der Bestandsdaten in die korrespondierende RosettaNet-Nachricht „Inventory Report Notification“ um. Diese Nachricht dient zur Übermittlung von Bestandsmengen und Statusinformationen von Lagerartikeln. Die Bestandsmengen werden dabei anhand des Artikelstatus differenziert. So können u. a. Gesamtbestände, Bestände bestimmter Lager oder Bestände, die gerade zwischen Lagerorten transportiert werden, ausgewiesen werden [Rose02].

Das **Kommunikations-Subsystem** überträgt die RosettaNet-Nachricht über das Kommunikationsprotokoll HTTP. Grundsätzlich kann die gegebene Anforderung einer hohen Betriebsunabhängigkeit durch ein Kommunikations-Subsystem unterstützt werden, das eine asynchrone, gepufferte Kommunikation erlaubt. Dies erscheint hier jedoch aufgrund der nur einmal täglich stattfindenden Propagation nicht sinnvoll. Außerdem ist eine solche Lösung durch die gegebenen Restriktionen in Form der vorhandenen Kopplungsmechanismen ausgeschlossen.

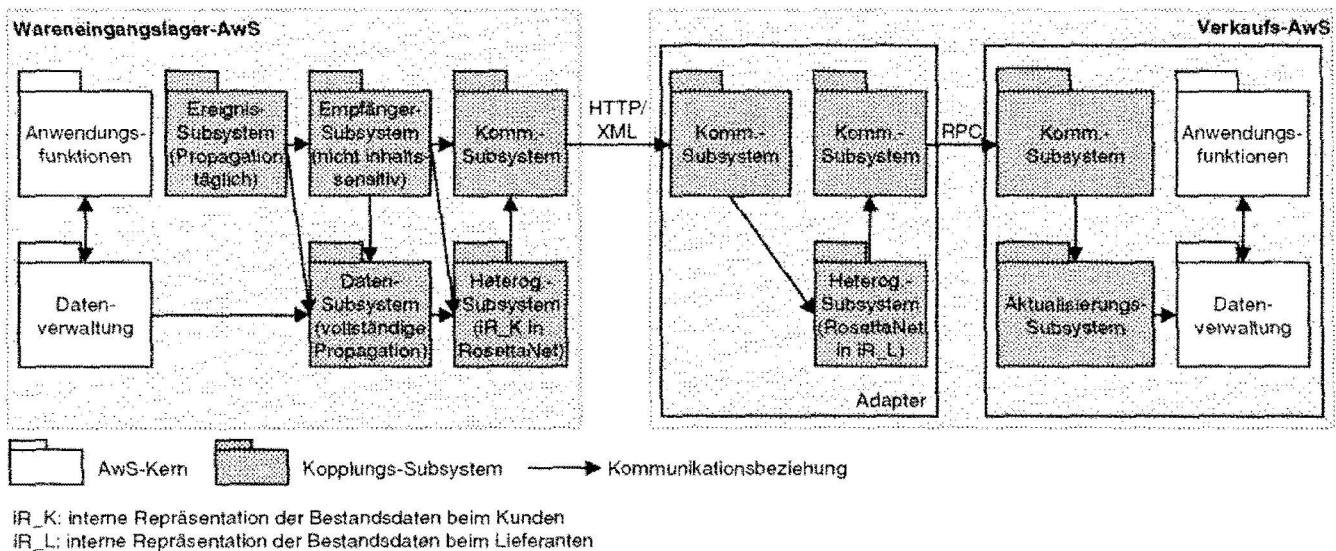


Abb. 5: Kopplungsarchitektur zur gemeinsamen Nutzung von Bestandsdaten im Rahmen des VMI

Das Kopplungs-Subsystem des **Verkaufs-AWS** wird wie folgt realisiert:

Zur Überwindung der gegebenen Heterogenität zwischen Wareneingangslager-AWS und Verkaufs-AWS wird ein Adapter eingeführt. Dieser stellt aus funktionaler Sicht einen Teil des Verkaufs-AWS dar. Er wird jedoch als eigenständige Komponente realisiert. Dies hat den Vorteil, dass der Adapter auch als Teil weiterer AWS des Lieferanten eingesetzt werden kann. Der Adapter besteht aus den folgenden drei Subsystemen:

- Ein **Kommunikations-Subsystem**, das die XML-Nachricht über HTTP entgegennimmt.
- Ein **Heterogenitäts-Subsystem**, das eine Umsetzung des fachlichen Nachrichtenformats der RosettaNet-Nachricht „Inventory Report Notification“ in das proprietäre interne Format des Verkaufs-AwS transformiert. Hierbei kann beispielsweise eine Umsetzung zwischen verschiedenen Datumsformaten erfolgen.
- Ein **Kommunikations-Subsystem**, das die Nachricht in einen entsprechenden RPC-Aufruf umsetzt.

Der RPC-Aufruf wird durch ein weiteres **Kommunikations-Subsystem** entgegengenommen und an ein neu zu implementierendes Aktualisierungs-Subsystem weitergeleitet.

Im Verkaufs-AwS ist die entsprechende Tabelle für die Bestandsdaten bereits vorhanden. Diese wurde bisher nur manuell über die Benutzeroberfläche gepflegt. Fachliche Zugriffsfunktionen, durch die das Kopplungs-Subsystem auf die Daten zugreifen könnte, sind nicht vorhanden. Diese Funktionen werden im **Aktualisierungs-Subsystem** implementiert. Das Aktualisierungs-Subsystem ist außerdem für die Gewährleistung semantischer Integritätsbedingungen verantwortlich. Hierzu muss die Aktualisierung der Daten durch den Einsatz von Datenbanktransaktionen geschützt werden.

5 Zusammenfassung und Ausblick

Kopplungsarchitekturen stellen Realisierungskonzepte für die Integration von AwS dar. Im vorliegenden Beitrag werden überbetriebliche Kopplungsarchitekturen von AwS betrachtet, die zur Unterstützung überbetrieblicher Geschäftsprozesse geeignet sind. Hierzu wird eine Methodik für die Entwicklung überbetrieblicher Kopplungsarchitekturen beschrieben. Die Methodik unterscheidet drei Modellebenen: die Ebene des überbetrieblichen Geschäftsprozesses, die AwS-Ebene und die Kopplungsarchitektur-Ebene. Die Entwicklung von Kopplungsarchitekturen wird im Kontext Supply-Chain-Management am Beispiel des überbetrieblichen Konzepts des VMI erläutert. In diesem Beispiel wird eine datenorientierte Kopplungsarchitektur mit redundanter Datenhaltung entwickelt, die eine gemeinsame Nutzung der Lagerbestandsdaten des Kunden durch je ein AwS des Kunden und des Lieferanten ermöglicht. Beim Entwurf der Kopplungsarchitektur werden der überbetriebliche Geschäftsprozess, die gegebenen Anforderungen an die Integration der zu koppelnden AwS und die kopplungsrelevanten Eigenschaften der AwS berücksichtigt.

Im Rahmen zukünftiger Forschungsaktivitäten sollen u. a. weitere Aufgabenintegrations-Muster identifiziert werden. Hierbei ist v. a. die Bestimmung von Merkmalen interessant, die zur Identifikation von Aufgabenintegrations-Mustern herangezogen werden können. Die Entwicklung von Kopplungsarchitekturen kann aufgrund ihrer Komplexität nur werkzeuggestützt erfolgen. Deshalb soll in weiteren Arbeiten ein Werkzeug zur Unterstützung der Methodik prototypisch implementiert werden.

Literatur

- [BeUV00] Becker, J.; Uhr, W.; Vering, O.: Integrierte Informationssysteme in Handelsunternehmen auf der Basis von SAP-Systemen. Springer, Berlin 2000.
- [Chris98] Christopher, M.: Logistics and Supply Chain Management. 2. Auflage, Financial Times, London 1998.
- [CoGö01] Corsten, H.; Gössinger, R.: Einführung in das Supply Chain Management. Oldenbourg, München 2001.
- [Fers92] Ferstl, O. K.: Integrationskonzepte Betrieblicher Anwendungssysteme. Fachberichte Informatik der Universität Koblenz-Landau Nr. 1/1992.
- [FeSi95] Ferstl, O. K.; Sinz, E. J.: Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen. In: Wirtschaftsinformatik, 37 (1995) 3, S. 209-220.
- [FeSi01] Ferstl, O. K.; Sinz, E. J.: Grundlagen der Wirtschaftsinformatik. 4. Auflage, Oldenbourg, München 2001.
- [Holm98] Holmström, J.: Business process innovation in the supply chain – a case study of implementing vendor managed inventory. In: European Journal of Purchasing & Supply Management, Nr. 4, 1998, S. 127-131.
- [IETF02] Internet Engineering Task Force (IETF), RPC: Remote Procedure Call Protocol Specification Version 2. <http://www.ietf.org/rfc/rfc1831.txt>, Abruf am 2002-04-13.
- [KnMZ00] Knolmayer, G.; Mertens, P.; Zeier, A.: Supply Chain Management auf Basis von SAP-Systemen. Springer, Heidelberg 2000.
- [Krem99] Kremer, R.: Replikatives Informationsmanagement in verteilten Groupware-Umgebungen. Shaker, Aachen 1999.

- [Krum97] Krumbiegel, J.: *Integrale Gestaltung von Geschäftsprozessen und Anwendungssystemen in Dienstleistungsbetrieben*. Dissertation, Deutscher Universitätsverlag, Wiesbaden 1997.
- [LaCP98] Lambert, M. D.; Cooper, M. C.; Pagh, J. D.: *Supply Chain Management: Implementation Issues and Research Opportunities*. In: *The International Journal of Logistics Management*, 9 (1998) 2, S. 1-18.
- [Mau00] Mau, M.: *Supply Chain Management – Realisierung von Wertschöpfungspotentialen durch ECR-Kooperation zwischen mittelständischer Industrie und Handel im Lebensmittelsektor*. Dissertation. Moderne Wirtschaft, Frankfurt/Main 2000.
- [Rahm94] Rahm, E.: *Mehrrechner-Datenbanksysteme: Grundlagen der verteilten und parallelen Datenbankverarbeitung*. Addison-Wesley, Bonn 1994.
- [Rose02] RosettaNet. <http://www.rosettanet.org>, Abruf am 2002-04-16.
- [SAP02] SAP AG: *BAPI-Programmierleitfaden Release 46C*. <https://www012.sap-ag.de/~sapidb/011000358700004269971999d.doc>, Abruf am 2002-04-07. (nur für geschlossene Benutzergruppe zugänglich)
- [SiKS00] Simchi-Levi, D.; Kaminsky, P.; Simchi-Levi, E.: *Designing and managing the Supply Chain*. McGraw-Hill, Boston 2000.
- [SMFS01] Schissler, M.; Mantel, S.; Ferstl, O. K.; Sinz, E. J.: *Unterstützung von Kopplungsarchitekturen durch SAP R/3*. Bayerischer Forschungsverbund Wirtschaftsinformatik, FORWIN-Bericht Nr. FWN-2001-008, Bamberg u. a. 2001.
- [SMFS02] Schissler, M.; Mantel, S.; Ferstl, O. K.; Sinz, E. J.: *Kopplungsarchitekturen zur überbetrieblichen Integration von Anwendungssystemen und ihre Realisierung mit SAP R/3*. Erscheint in: *Wirtschaftsinformatik* 44 (2002) 5.
- [Voss00] Vossen, G.: *Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme*. 4. Auflage, Oldenbourg, Wien 2000.
- [Wald98] Waldmann, J.: *Unternehmensübergreifende logistische Konzepte als Ansatz zur Komplexitätsreduktion*. In: Adam, D. (Hrsg.): *Komplexitätsmanagement, Schriften zur Unternehmensführung*, Band 61, Gabler, Wiesbaden 1998.