Zweitveröffentlichung

Mantel, Stephan; Eckert, Sven; Schissler, Martin; u. a.

Eine Entwicklungsmethodik für die überbetriebliche Integration von Anwendungssystemen

Datum der Zweitveröffentlichung: 23.10.2024

Verlagsversion (Version of Record), Konferenzveröffentlichung

Persistenter Identifikator: urn:nbn:de:bvb:473-irb-1040703

Erstveröffentlichung

Mantel, Stephan; Eckert, Sven; Schissler, Martin; Schäffner, Claus; Ferstl, Otto K.; Sinz, Elmar J. (2004): Eine Entwicklungsmethodik für die überbetriebliche Integration von Anwendungssystemen, in: Dieter Bartmann, Peter Mertens, und Elmar J. Sinz (Hrsg.), Überbetriebliche Integration von Anwendungssystemen : FORWIN-Tagung 2004, 1. Auflage, Aachen: Shaker, S. 21–39.

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

Eine Entwicklungsmethodik für die überbetriebliche Integration von Anwendungssystemen

Stephan Mantel, Universität Bamberg

Sven Eckert, Universität Bamberg

Martin Schissler, Universität Bamberg

Claus Schäffner, Universität Bamberg

Otto K. Ferstl, Universität Bamberg

Elmar J. Sinz, Universität Bamberg

Zusammenfassung: Eine wichtige Voraussetzung für die Gestaltung und effektive Durchführung von unternehmensübergreifenden Geschäftsprozessen ist eine leistungsfähige und flexible Kopplung der diese Prozesse unterstützenden Anwendungssysteme durch Kopplungssysteme. Trotz einer großen Anzahl an Integrationsprodukten auf dem Markt gibt es nur wenige umfassende Ansätze, die Aspekte der Modellierung und des Vorgehens bei der Entwicklung von Kopplungssystemen berücksichtigen. Der vorliegende Beitrag stellt eine Methodik zur Entwicklung von Kopplungssystemen für die überbetriebliche Integration von Anwendungssystemen anhand der verwendeten Modellebenen und Vorgehensschritte vor.

Stichwörter: Integration von Anwendungssystemen, Kopplungssystem, Entwicklungsmethodik, Modellierung, Vorgehensmodell

1 Einleitung

Die Gestaltung von überbetrieblichen Geschäftsprozessen mit abgestimmter Zielsetzung ist der Kern von betriebswirtschaftlichen Konzepten, wie z. B. Supply-Chain-Management oder Virtuelle Unternehmen. In Zeiten zunehmender Globalisierung nimmt die effektive Durchführung von überbetrieblichen Geschäftsprozessen eine immer wichtigere Rolle ein, um auf umkämpften Märkten erfolgreich agieren zu können. Zentrale Voraussetzung hierzu ist eine flexible und leistungsfähige Kopplung der diese Geschäftsprozesse unterstützenden Anwen-

dungssysteme (AwS). Als Basis für die Entwicklung solcher Kopplungen sind mittlerweile eine große Anzahl von Produkten und Plattformen wie z. B. Microsoft BizTalk™ oder IBM Websphere MQ Integrator® vorhanden. Angesichts der hohen Komplexität typischer Kopplungen werden darüber hinaus umfassende Ansätze benötigt, die zusätzlich die Model-lierung und das Vorgehen bei der Entwicklung von AwS-Kopplungen unterstützen. Zu nennen sind hier z. B. der Ansatz von Juric et al. [JBLN01], die Entwicklungsmethodik des Projekts „Modellierung einer verteilten Architektur für die Entwicklung unternehmensübergreifender Informationssysteme und ihre Validierung im Handelsbereich" (MOVE) [FHK+98; FAS+99] und der Ansatz der Standardisierungsinitiative „Electronic Business Extensible Markup Language" (ebXML) [ebXM01; ebXM03]. Diese Ansätze wurden im Rahmen von [EMR+oJ] untersucht. Die Analyse zeigt Bedarf nach Verbesserungen in den Bereichen Systematik, Ganzheitlichkeit und Durchgängigkeit der Ansätze sowie Bedarf nach einer computergestützten Vorgehensweise bei der Entwicklung von AwS-Kopplungen.

In dieser Arbeit wird die im Rahmen des Projekts „Offene Anwendungssystem-Architekturen in überbetrieblichen Wertschöpfungsketten" (OASYS) im Forschungsverbund FORWIN erarbeitete Methodik zur Entwicklung von Kopplungssystemen vorgestellt. Kapitel 2 beschreibt zunächst die Gestaltung überbetrieblicher Geschäftsprozesse und ihre Unterstützung durch gekoppelte AwS. In Kapitel 3 wird die Methodik anhand der verwendeten Modellebenen und des Vorgehens bei der Entwicklung von Kopplungssystemen näher beschrieben. Anschließend erfolgt in Kapitel 4 eine Zusammenfassung der Ergebnisse.

2 Unterstützung überbetrieblicher Geschäftsprozesse durch die Kopplung von Anwendungssystemen

Zur Unterstützung betriebswirtschaftlicher Konzepte der überbetrieblichen Kooperation, wie z. B. Supply-Chain-Management, ist eine Integration der Geschäftsprozesse der beteiligten Unternehmen zu einem überbetrieblichen Geschäftsprozess mit einer gemeinsamen, abgestimmten Zielsetzung nötig. Um die Potenziale abgestimmter überbetrieblicher Geschäftsprozesse, wie z. B. kürzere Durchlaufzeiten oder geringere Lagerbestände, auszuschöpfen, sind AwS als maschinelle Aufgabenträger einzusetzen und unternehmensübergreifend zu koppeln. Die in dieser Arbeit vorgestellte Entwicklungsmethodik bietet ausgehend von der Gestaltung und Modellierung eines überbetrieblichen Geschäftsprozesses eine durchgehende Unterstützung bei der Entwicklung eines Kopplungssystems zwischen den zu integrierenden AwS. Das **Kopplungssystem** umfasst alle kopplungsrelevanten Elemente der beteiligten AwS. Bei jedem AwS wird zwischen dem eigentlichen Kern und dem Kopplungs-Teilsystem unterschieden [MES+02, 184].

Das **Kopplungs-Teilsystem** enthält diejenigen Elemente der AwS, die ausschließlich der Kopplung mit anderen AwS dienen. Die Realisierung des Kopplungs-Teilsystems basiert auf bestehenden Kopplungsmechanismen, die Dienste und Kommunikationsprotokolle für die Kopplung bereitstellen. Der **AwS-Kern** enthält die zu integrierenden Elemente sowie weitere, nicht kopplungsrelevante Elemente.

Eine Strukturierung der Schnittstellen zwischen AwS-Kern und Kopplungs-Teilsystem ergibt sich anhand der Gliederung des jeweiligen AwS gemäß dem ADK-Modell [FeSi01, 289ff.]. Dieses unterteilt ein AwS in die Teilsysteme Anwendungsfunktionen (A), Datenverwaltung (D) und Kommunikation (K). Letzteres wird weiter in die Teilbereiche Kommunikation mit Personen (K_p) und Kommunikation mit weiteren Maschinen (K_m) zerlegt. A, D und K_p bilden den AwS-Kern, K_m das Kopplungs-Teilsystem. Die vom AwS-Kern angebotenen Schnittstellen können daher in (1) Anwendungsfunktionsschnittstellen, (2) Datenverwaltungsschnittstellen sowie (3) grafische Schnittstellen und Kommandozeilen-Schnittstellen zur Kommunikation mit Personen untergliedert werden. Die letztgenannten können über ihren originären Verwendungszweck hinaus ebenfalls zur Realisierung einer AwS-Kopplung genutzt werden, z. B. durch so genannte Screen-Scraping-Verfahren. Neben diesen drei Arten von angebotenen Schnittstellen kann der AwS-Kern zusätzlich Schnittstellen benötigen, die vom Kopplungs-Teilsystem bereitzustellen sind.

Der Bauplan eines Kopplungssystems wird in Form einer **Kopplungsarchitektur** spezifiziert. Diese beschreibt alle für die Integration relevanten Elemente sowie die Beziehungen zwischen diesen Elementen [SMFS01, 2]. Es werden drei Arten von Kopplungsarchitekturen unterschieden, die anhand der verfolgten Ziele abgegrenzt werden [MES+02, 184f.; SMFS02, 460ff.]:

- **Ereignisorientierte Kopplungsarchitekturen** zielen auf die Übertragung von Ereignissen und Daten zwischen AwS in Form von Nachrichten ab. Typische Vertreter dieser Kategorie sind Architekturen klassischer EDI-Lösungen.
- **Datenorientierte Kopplungsarchitekturen** bezwecken die Manipulation gemeinsamer Daten durch mehrere AwS. Abhängig von ggf. verwendeten Kopien der gemeinsamen Daten wird zwischen redundanter und nicht-redundanter Datenhaltung unterschieden. Im Umfeld von SAP wird diese Art von Kopplungsarchitektur häufig auf der Basis von ALE realisiert.
- **Funktionsorientierte Kopplungsarchitekturen** ermöglichen die Nutzung gemeinsamer Funktionen und ggf. gemeinsamer Daten durch mehrere AwS. Ein mögliches Unterscheidungskriterium ist auch hier die redundante oder nicht-redundante Haltung der Programmmodule, welche die gemeinsamen Funktionen beinhalten. Kopplungsarchitekturen, die die Nutzung gemeinsamer Funktionsserver vorsehen, sind dieser Kategorie zuzuordnen.

3 Konzept der Entwicklungsmethodik

Im folgenden Kapitel wird das Konzept der Entwicklungsmethodik zur Gestaltung überbetrieblicher Kopplungssysteme anhand der verwendeten Modellebenen und des Vorgehensmodells beschrieben. Die einzelnen Modellebenen erfassen unterschiedliche Aspekte des Kopplungsproblems und der Kopplungsarchitektur. Das Vorgehensmodell beschreibt die Durchführung der Entwicklungsmethodik.

Im Weiteren wird davon ausgegangen, dass bereits bestehende AwS zu koppeln sind. Ihre Eigenschaften begrenzen zunächst den Lösungsraum für die Entwicklung des Kopplungssystems, jedoch können während der Entwicklung Anpassungen dieser Eigenschaften erforderlich werden.

3.1 Modellebenen

Die Entwicklungsmethodik unterscheidet sechs Modellebenen (Abb. 1). Aus Aufgaben- bzw. Aufgabenträgersicht [FeSi01, 2ff.] korrespondiert die Ebene des überbetrieblichen Geschäftsprozesses (Ebene 1) mit der Aufgabenebene. Die Ebenen 2 bis 6 dienen gemeinsam der Spezifikation der zugehörigen Aufgabenträger. Diese Spezifikation wird wiederum anhand des Nutzer-/Basismaschinen-Modells [FeSi01, 286ff.] differenziert. Die Nutzermaschine des Kopplungssystems beschreibt gemäß diesem Modell die AwS-Kern-Ebene (Ebene 2) und die Funktionen-Ebene (Ebene 3). Das korrespondierende Programm einschließlich zugehöriger Basismaschinen wird in mehreren Abstraktionen auf der Subsystem-, Prozess- und Implementierungs-Ebene (Ebenen 4 - 6) beschrieben.

Eine weitere Einordnung der Modellebenen kann mit Hilfe der MDA (Model Driven Architecture) durchgeführt werden [MiMu03]. Das auf der Ebene des überbetrieblichen Geschäftsprozesses erstellte Modell entspricht dem Computation Independent Model der MDA [MiMu03, 3-1]. Es bildet den Ausgangspunkt zur Entwicklung eines Kopplungssystems und kann u. a. zur Kommunikation mit fachlichen Experten genutzt werden. Die Modelle der Ebenen 2 und 3 entsprechen dem Platform Independent Model der MDA [MiMu03, 3-1f.]. Sie spezifizieren ein Kopplungssystem weitgehend unabhängig von Kopplungsmechanismen der zu verwendenden Plattform. Vielmehr erfolgt auf ihrer Basis die Auswahl geeigneter Kopplungsmechanismen. Das Platform Specific Model der MDA findet sich in Form der Ebenen 4 und 5 in der Methodik wieder [MiMu03, 3-8]. Hier wird beschrieben, wie das Modell der Ebene 3 unter Verwendung der gewählten Kopplungsmechanismen realisiert wird.

Im Folgenden werden die einzelnen Ebenen der Entwicklungsmethodik detailliert anhand ihrer Metamodelle beschrieben.

		Ebene	Name	Beschreibung
Aufgaben		1	Ebene des *überbetrieblichen* Geschäftsprozesses	Erfassung des überbetrieblichen Geschäftsprozesses anhand des Semantischen Objektmodells
Aufgabenträger	Nutzermaschine	2	AwS-Kern-Ebene	Erfassung der zu integrierenden AwS-Kerne
		3	Funktionen-Ebene des Kopplungssystems	Erfassung der Funktionskomponenten des Kopplungssystems, inklusive Operationen und Speicher, sowie deren Kommunikationsbeziehungen
	Programm + Basismaschine	4	Subsystem-Ebene des Kopplungssystems	Erfassung der Subsysteme mit zugehörigen Schnittstellen sowie deren Benutzt-Beziehungen
		5	Prozess-Ebene des Kopplungssystems	Erfassung von Prozessen und Konnektoren sowie deren Zuordnung zu Prozessoren
		6	Implementierungs-Ebene des Kopplungssystems	Beschreibung des Programms zur Realisierung des Kopplungssystems in ausführbarer Form

Abb. 1: Ebenenübersicht

3.1.1 Ebene des überbetrieblichen Geschäftsprozesses

Ausgangspunkt der Entwicklung eines Kopplungssystems ist ein überbetrieblicher Geschäftsprozess, der gemäß dem Semantischen Objektmodell (SOM) beschrieben wird (Ebene 1) [FeSi95; FeSi01, 179ff.]. Die Spezifikation erfolgt in Form eines verteilten Systems aus autonomen und lose gekoppelten betrieblichen Objekten, die sich mittels Transaktionen koordinieren (Abb. 2) [FeSi01, 181f.].

Die Struktur des Geschäftsprozesses wird im Interaktionsschema (IAS) in Form betrieblicher Objekte, die durch Transaktionen verknüpft sind, modelliert. Betriebliche Objekte sind entweder Umweltobjekte oder Diskursweltobjekte, die jeweils einem der beteiligten Unternehmen zugeordnet sind. Die Koordination von betrieblichen Aufgaben erfolgt gemäß hierarchischer Regelung oder nicht-hierarchischer Verhandlung. Im Falle einer Verhandlung werden Anbahnungs- (A), Vereinbarungs- (V) und Durchführungstransaktionen (D), im Falle einer Regelung Steuer- (S) und Kontrolltransaktionen (K) unterschieden. Jede Transaktion dient der Koordination oder dem Transfer von Güter-, Zahlungs- oder Dienstleistungen [FeSi01, 186]. Den Aufgaben betrieblicher Objekte und den Transaktionen ist jeweils ein Automatisierungsgrad zugeordnet.

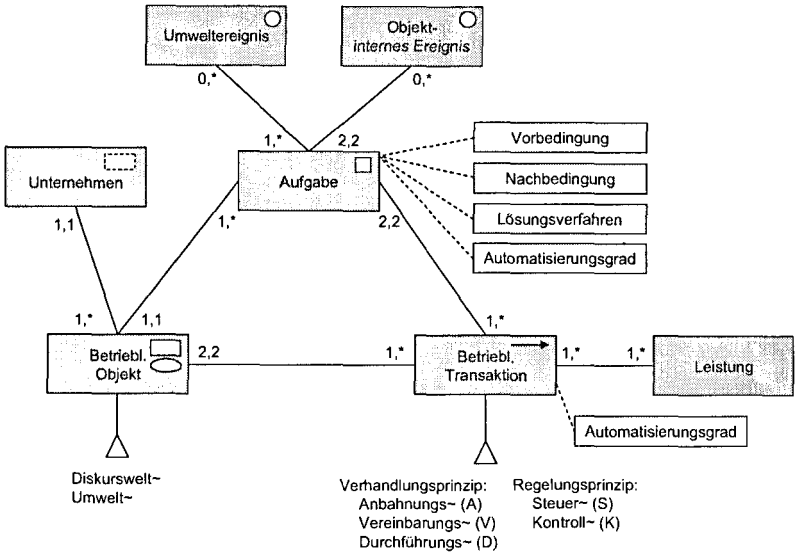

Abb. 2: Metamodell der Ebene des überbetrieblichen Geschäftsprozesses (Ebene 1)

Das Vorgangs-Ereignis-Schema (VES) beschreibt das zugehörige Verhalten eines Geschäftsprozesses in Form von Vorgangstypen/Aufgaben[1] und ihren Ereignisbeziehungen. Eine Transaktion realisiert eine Ereignisbeziehung zwischen zwei Aufgaben, die unterschiedlichen betrieblichen Objekten zugeordnet sind. Beziehungen zwischen Aufgaben eines Objekts werden durch objektinterne Ereignisse hergestellt. Umweltereignisse dienen der Modellierung objektexterner, jedoch nicht in Transaktionen erfasster Ereignisse [FeSi01, 199]. Die Vor- bzw. Nachbedingungen einer Aufgabe geben an, welche Vorereignisse anliegen müssen, damit eine Aufgabe durchgeführt werden kann, bzw. welche Nachereignisse durch die Aufgabendurchführung ausgelöst werden [Popp94, 80f.; Raue96, 144]. Vor- und Nachbedingungen sowie das entsprechende (Teil-)Lösungsverfahren einer Aufgabe werden unter Verwendung der Aussagenlogik spezifiziert.

Die für die Kopplung relevanten Teilbereiche im Geschäftsprozessmodell werden anhand von kategorisierten Aufgabenintegrations-Mustern (AIM) abgegrenzt.

[1] In dieser Arbeit wird nicht zwischen der Außen- und der Innensicht von Aufgaben differenziert, so dass die Begriffe Vorgangstyp und Aufgabe synonym verwendet werden [FeSi01, 183f].

3.1.2 Anwendungssystem-Kern-Ebene

Ebene 2 beschreibt die automatisierten Aufgabenträger der Aufgaben des Geschäftsprozesses in Form von AwS-Kernen, d. h. unabhängig vom Kopplungssystem. AwS-Kerne sind durch Integriert-Beziehungen verbunden und jeweils einem Unternehmen zugeordnet (Abb. 3).

Abb. 3: Metamodell der AwS-Kern-Ebene (Ebene 2)

Die Beziehungen zwischen der AwS-Kern-Ebene und der Ebene des überbetrieblichen Geschäftsprozesses sollen hier nur kurz aufgezeigt werden:

- Ein AwS-Kern führt die automatisierten Aufgabenanteile eines oder mehrerer betrieblicher Objekte aus, umgekehrt wird ein betriebliches Objekt von einem oder mehreren AwS-Kernen unterstützt (n:m-Beziehung) [FeSi01, 202; MES+02]. Soweit möglich, wird der überbetriebliche Geschäftprozess so weit detailliert, dass jedem betrieblichen Objekt nur ein AwS-Kern zugeordnet ist (n:1-Beziehung).
- Einer Integriert-Beziehung sind eine oder mehrere betriebliche Transaktionen zugeordnet.

3.1.3 Funktionen-Ebene des Kopplungssystems

Die dritte Ebene der Modellebenenhierarchie beschreibt die Funktionalität des Kopplungssystems. Die Struktur des Systems wird in Form von objektorientierten Funktionskomponenten modelliert (Abb. 4). Dabei werden Funktionskomponenten der AwS-Kerne und des Kopplungs-Teilsystems unterschieden. Letztere werden spezialisiert u. a. in Ereignis-, Empfänger-, Extraktions-, Heterogenitäts-, Kommunikations- und Aktualisierungs-Funktionskomponenten. Funktionskomponenten sind jeweils genau einem Unternehmen zugeordnet und können aus anderen Funktionskomponenten bestehen. Sie kapseln eine Menge von Operationen und verfügen über einen Speicher für die zeitliche Entkopplung von Erzeugung und Nutzung der Informationen.

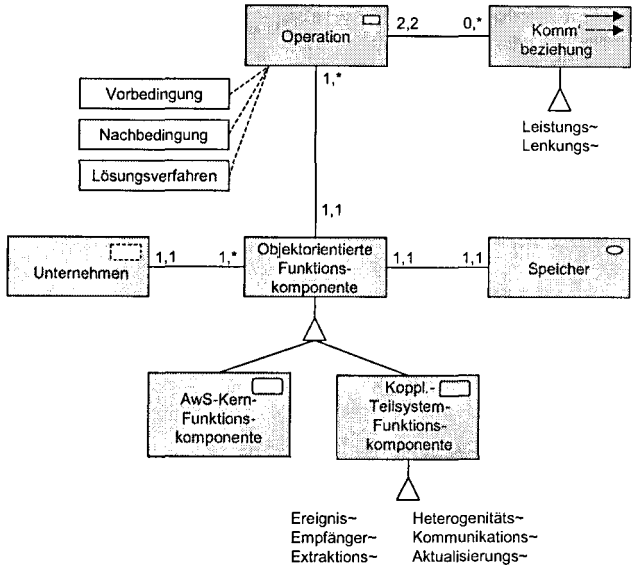

Abb. 4: Metamodell der Funktionen-Ebene (Ebene 3)

Die Operationen verschiedener Funktionskomponenten tauschen Informationen über Kommunikationsbeziehungen aus, die analog zur Geschäftsprozessmodellierung in Leistungs- und Lenkungsbeziehungen differenziert werden [FeSi01, 41ff.]. Leistungen bestehen im Kontext des Kopplungssystems aus Informationen, die eine Funktionskomponente anderen Funktionskomponenten bereitstellt, z. B. Empfänger-Informationen. Der Austausch von Leistungen wird durch Lenkungsbeziehungen gesteuert.

Das Verhalten des Kopplungssystems wird auf dieser Ebene durch die Angabe der Vor- und Nachbedingungen der Operationen modelliert. Hierbei wird den Kommunikationsbeziehungen ein Ereignischarakter zugesprochen. Die Vor- und Nachbedingung einer Operation legen zusammen mit dem Lösungsverfahren fest, welche Nachereignisse durch welche Vorereignisse ausgelöst werden. Sowohl die Vor- und Nachbedingungen als auch das zugehörige (Teil-)Lösungsverfahren werden mittels Aussagenlogik spezifiziert.

Die Kernelemente der Ebene stehen mit Elementen der vorgelagerten AwS-Kern-Ebene (Ebene 2) wie folgt in Beziehung:

- Ein AwS-Kern wird durch eine oder mehrere AwS-Kern-Funktionskomponenten beschrieben.
- Eine Integriert-Beziehung wird auf eine Menge von Kopplungs-Teilsystem-Funktionskomponenten und Kommunikationsbeziehungen abgebildet.

3.1.4 Subsystem-Ebene des Kopplungssystems

Auf der Subsystem-Ebene wird beschrieben, wie die Funktionskomponenten des Kopplungssystems durch eine Menge interagierender Subsysteme realisiert werden (Abb. 5). Die auf dieser Ebene verwendeten Modellierungselemente orientieren sich zum Teil an der UML [OMG03b]. Zentrales Element zur Erfassung der Struktur ist das Subsystem. Ein Subsystem realisiert ein bestimmtes Verhalten. Der Zugriff auf ein Subsystem erfolgt ausschließlich durch Nutzung der von diesem angebotenen Operationen. Operationen werden zu Schnittstellen zusammengefasst. Außerdem kann ein Subsystem einen persistenten oder transienten Speicher besitzen. Subsysteme bestehen ggf. aus weiteren Subsystemen. Es wird zwischen Subsystemen unterschieden, die dem AwS-Kern zugeordnet sind, und solchen, die zum Kopplungs-Teilsystem gehören. Erstere werden nur in der Außensicht betrachtet. Diese Außensicht reicht in vielen Fällen auch für die Subsysteme des Kopplungs-Teilsystems. Die eingesetzten Kopplungsmechanismen bieten häufig wiederverwendbare, parametrisierbare Bausteine an, die nur angepasst werden müssen. Von der Innensicht dieser Subsysteme kann hierbei abstrahiert werden. Ist eine Spezifikation der Innensicht erforderlich, so kann z. B. auf die verschiedenen von der UML bereitgestellten Konstrukte zurückgegriffen werden.

Die Funktionskomponenten eines Kopplungssystems werden durch ein oder mehrere AwS realisiert. Dementsprechend ist eine Zuordnung von Subsystemen zu AwS nötig. Für den Betrieb eines AwS ist ein Unternehmen verantwortlich.

Zur Modellierung des Verhaltens werden die Benutzt-Beziehungen als Interaktionskanäle interpretiert, auf denen Nachrichten ausgetauscht werden. Bei Eintreffen einer Nachricht an einer Schnittstelle wird die korrespondierende Operation aufgerufen. Die Spezifikation der Nachrichten orientiert sich an der Vorgehensweise im Collaboration Diagram der UML [OMG03b, 3-131ff.]. Es können unter anderem ein Zeitindex und die im Rahmen der Nachricht übergebenen Parameter angegeben werden.

Die Beziehungen zwischen Subsystem- und Funktionen-Ebene sollen hier nur kurz skizziert werden:

- Eine Operation einer Funktionskomponente wird durch ein oder mehrere Subsysteme realisiert.
- Der Speicher einer Funktionskomponente wird durch ein oder mehrere Subsysteme realisiert.
- Die Kommunikationsbeziehungen der Ebene 3 werden durch Schnittstellen und Benutzt-Beziehungen und evtl. durch Subsysteme realisiert.

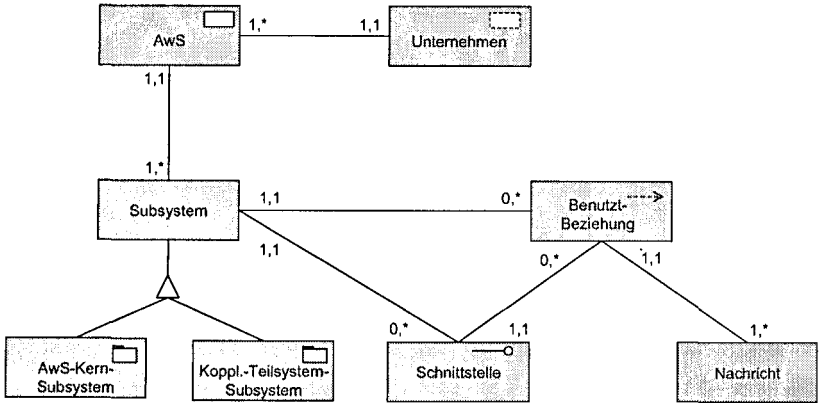

Abb. 5: Metamodell der Subsystem-Ebene (Ebene 4)

3.1.5 Prozess-Ebene des Kopplungssystems

Auf Ebene 5 wird das Kopplungssystem als System parallel ablaufender Prozesse, z. B. Betriebssystemprozesse, dargestellt (Abb. 6). Die Erfassung der strukturellen Aspekte erfolgt primär in Form von Prozessen, die über Konnektoren interagieren (vgl. z. B. [CBB+03, 103ff., 142ff.]). Die Konnektoren sind über Ports mit den Prozessen verbunden. Es werden Angebots-Ports, über die Dienste angeboten werden, und Nachfrage-Ports, über die Dienste genutzt werden, unterschieden. Die Prozesse werden auf Prozessoren ausgeführt, die über Kommunikationsverbindungen verknüpft sind.

Zur Modellierung des Verhaltens werden die Konnektoren als Kanäle für den Austausch von Nachrichten interpretiert. Die Modellierung der Nachrichten erfolgt wiederum unter Rückgriff auf die im Collaboration Diagram der UML eingesetzte Notation. Es können synchrone Nachrichten, bei denen der nachfragende Prozess auf eine Antwort wartet, und asynchrone Nachrichten, bei denen dies nicht der Fall ist, unterschieden werden.

Auch auf die Beziehungen zwischen Subsystem- und Prozess-Ebene kann hier nur knapp eingegangen werden:

- Ein Subsystem ist an der Ausführung eines oder mehrerer Prozesse beteiligt.
- Ein Prozess wird durch ein oder mehrere Subsysteme generiert.
- Wenn zwei durch eine Benutzt-Beziehung verbundene Subsysteme an unterschiedlichen Prozessen beteiligt sind, enthält Ebene 5 einen Konnektor, einen Nachfrage-Port und einen Angebots-Port, der mit der zugehörigen Subsystem-Schnittstelle korrespondiert.

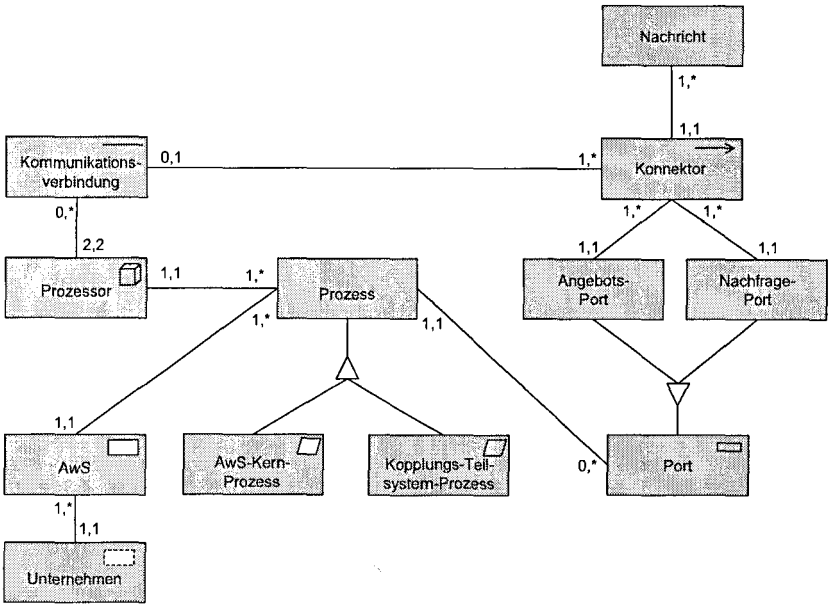

Abb. 6: Metamodell der Prozess-Ebene (Ebene 5)

3.1.6 Implementierungs-Ebene des Kopplungssystems

Die Ebene der Implementierung beschreibt die Realisierung des Kopplungssystems anhand der auf den vorgelagerten Ebenen entworfenen Kopplungsarchitektur. Ein Modell auf dieser Ebene spezifiziert das Programm zur Realisierung des Kopplungssystems in ausführbarer Form. Das Metamodell dieser Ebene ist abhängig von den verwendeten Kopplungsmechanismen. Als Kopplungsmechanismen stehen neben Programmiersprachen, wie Java™, u. a. auch Integrationsprodukte wie z. B. Microsoft BizTalk™ oder IBM WebSphere MQ Integrator® zur Verfügung.

3.2 Vorgehen

Die Entwicklung von überbetrieblichen Kopplungssystemen ist ein komplexes Simultanplanungsproblem. Zur Komplexitätsbewältigung wird das Simultanproblem in sequenziell zu durchlaufende Schritte zerlegt. Jeder Schritt des Entwicklungsprozesses schränkt durch Festlegung von Kopplungsmerkmalen den Lösungsraum bezüglich geeigneter Kopplungssysteme ein. Falls in einer Schrittfolge keine Lösung mit einem angestrebten Zielerreichungsgrad gefunden werden

kann, sind ein Rückschritt und eine Neufestlegung bereits durchgeführter Schritte möglich [MES+02]. Das Vorgehen zur Entwicklung eines Kopplungssystems unterteilt sich in die folgenden Schritte:

Schritt 1: Modellierung des überbetrieblichen Geschäftsprozesses

Im ersten Schritt wird der überbetriebliche Geschäftsprozess (Ebene 1) in Form von IAS und VES modelliert. Durch mehrstufige Verfeinerung des Geschäftsprozessmodells werden Leistungsbeziehungen sukzessive verfeinert und gleichzeitig die Lenkung der Geschäftsprozesse aufgedeckt [FeSi01, 182]. Durch die Zuordnung der betrieblichen Objekte zu Unternehmen lassen sich überbetriebliche Transaktionen identifizieren.

Schritt 2: Kartierung der Anwendungssystem-Kerne

Auf der Grundlage des Geschäftsprozessmodells erfolgt die Kartierung der AwS-Kerne. Hierunter wird „die Beschreibung einer Anwendungssystemzuordnung zu Geschäftsprozessen durch Einordnung der Anwendungssysteme in Geschäftsprozeßmodelle" [Krum97, 137] verstanden. Zunächst werden die Aufgaben der betrieblichen Objekte hinsichtlich ihres Automatisierungsgrades als vollautomatisiert, teilautomatisiert oder nicht-automatisiert eingestuft. Die überbetrieblichen Transaktionen des Geschäftsprozessmodells werden als zu automatisierend oder als nicht zu automatisierend klassifiziert. Die zu automatisierenden Transaktionen sind durch Kopplungssysteme zu unterstützen. Anschließend werden jedem Objekt mit voll- oder teilautomatisierten Aufgaben ein oder mehrere AwS-Kerne als Aufgabenträger zugeordnet und die Integriert-Beziehungen zwischen den AwS-Kernen bestimmt (Ebene 2).

Schritt 3: Identifikation von Aufgabenintegrations-Mustern im Geschäftsprozessmodell

In diesem Schritt werden innerhalb des erstellten Geschäftsprozessmodells AIM identifiziert. Diese umfassen eine Menge von Transaktionen sowie zugehöriger Aufgaben und stellen somit Teilstrukturen eines Geschäftsprozessmodells dar. Sie grenzen einen durch AwS zu integrierenden Bereich ab [MES+02, 194]. Es werden elementare und zusammengesetzte AIM unterschieden. Elementare AIM können nicht weiter zerlegt werden und stehen in typisierter Form zur Verfügung. Jedem AIM-Typ ist eine Kopplungsarchitekturart (vgl. Kapitel 2) zugeordnet, die zur Unterstützung des Musters geeignet ist. Zusammengesetzte AIM bestehen hingegen aus mehreren elementaren AIM und dienen der Visualisierung von fachlichen Zusammenhängen zwischen diesen. Sie können nach Bedarf gebildet werden.

Es lassen sich die folgenden drei Typen von elementaren AIM unterscheiden:

- **Reihenfolgebeziehung zwischen Aufgabendurchführungen**

  Kennzeichnend für ein Muster dieses Typs ist, dass die Durchführung einer im Muster enthaltenen Transaktion im empfangenden Objekt eine Verrichtung anstößt, die über die obligatorische Eingangsverarbeitung hinausgeht. Ein AIM dieses Typs wird durch eine ereignisorientierte Kopplungsarchitektur unterstützt.

- **Gemeinsame Nutzung von Aufgabenobjekt-Instanzen**

  Ein Aufgabenobjekt beschreibt die zu einer Aufgabe gehörenden Attribute (Aufgabenobjekt-Typen) und Attributwerte (Aufgabenobjekt-Instanzen) eines betrieblichen Systems. Ein AIM dieses Typs beschreibt die gemeinsame Nutzung von Aufgabenobjekt-Instanzen durch Aufgaben verschiedener betrieblicher Objekte. Die in dem Muster enthaltenen Transaktionen definieren u. a. semantische Integritätsbedingungen bezüglich der Gleichheit der Zustände der beteiligten Aufgabenobjekt-Instanzen. Unterstützt werden Muster dieses Typs durch datenorientierte Kopplungsarchitekturen.

- **Gemeinsame Nutzung von Lösungsverfahren**

  Ein Lösungsverfahren besteht aus einer Menge von Aktionen, die auf das Aufgabenobjekt einwirken oder dessen Zustände erfassen, sowie einer Aktionensteuerung, die geeignete Aktionen zur Verfolgung der Sach- und Formalziele einer Aufgabe auslöst [Fers92, 6f; FeSi01, 95]. Dieses Muster definiert durch die in ihm enthaltenen Aufgaben eine semantische Integritätsbedingung hinsichtlich der Gleichheit der gemeinsam genutzten Lösungsverfahren sowie der evtl. gemeinsam genutzten Aufgabenobjekt-Instanzen. Ein AIM dieses Typs wird durch funktionsorientierte Kopplungsarchitekturen realisiert.

Schritt 4: *Spezifikation von nicht-funktionalen Anforderungen an die Anwendungssystem-Integration*

Im ersten Schritt der Methodik wurden die funktionalen Anforderungen an die Integration der AwS in Form des Geschäftsprozessmodells spezifiziert. In diesem Schritt erfolgt nun die Erfassung der nicht-funktionalen Anforderungen (vgl. zur Unterscheidung von funktionalen und nicht-funktionalen Anforderungen [Somm01, 109ff.]). Diese werden für jedes identifizierte AIM anhand eines strukturierten Katalogs beschrieben. Je nach Typ des AIM variiert dieser Katalog hinsichtlich seiner Ausprägungen. Der Katalog umfasst die vier Kategorien Flexibilität, Echtzeitverhalten, Integration und Korrektheit [Fers92, 11]. Die unter den jeweiligen Kategorien zusammengefassten Anforderungen orientieren sich u. a. an den in der ISO-Norm 9126-1 aufgeführten Merkmalen [ISO01].

Die erste Kategorie befasst sich mit Anforderungen, die sich auf die **Flexibilität** des Kopplungssystems beziehen. Hierunter fallen u. a. Anforderungen hinsichtlich Skalierbarkeit, Anpassbarkeit

und Generizität. Unter dem Aspekt **Echtzeitverhalten** werden Anforderungen zusammengefasst, die sich insbesondere auf den Grad der Synchronisation zwischen den Vorgängen im Kopplungssystem und denen der abgebildeten Realität beziehen. Verfügbarkeits- [DoC80], Last- und Aktualitätsanforderungen fallen in diese Kategorie. In der Kategorie **Integration** werden die Strukturmerkmale Redundanz und Verknüpfung, die Verhaltensmerkmale Konsistenz und Zielorientierung sowie Aufgabenträgerunabhängigkeit unterschieden [Fers92; FeSi01, 215ff.; MKR+00, 3ff.]. Unter der Kategorie **Korrektheit** werden abschließend Anforderungen erfasst, die sich auf die Eindeutigkeit, Widerspruchsfreiheit und Vollständigkeit der Aufgabendurchführungen im Kopplungssystem beziehen.

Eine ausführliche Beschreibung des Anforderungskatalogs, inklusiv Beispiele, findet sich in [ESMS03].

Schritt 5: Erstellung des Funktionskomponentenmodells

In diesem Schritt wird das Funktionskomponentenmodell (Ebene 3) erstellt. Hierzu werden die Funktionskomponenten der AwS-Kerne und der Kopplungs-Teilsysteme und ihre Operationen spezifiziert sowie die Kommunikationsbeziehungen zwischen den Operationen bestimmt. AwS-Kern-Schnittstellen werden auf dieser Ebene durch AwS-Kern-Funktionskomponenten und den ein- und ausgehenden Kommunikationsbeziehungen ihrer Operationen beschrieben. Diese stellen Restriktionen für den Entwurf der Kopplungsarchitektur dar. Ist jedoch eine AwS-Kern-Schnittstelle nicht zur Realisierung des angestrebten Kopplungssystems geeignet, ergeben sich hieraus Änderungsanforderungen an den AwS-Kern. Das Modell ist hierbei entsprechend anzupassen.

Neben den Leistungsbeziehungen muss auf dieser Ebene auch die Lenkung der Funktionskomponenten spezifiziert werden. Diese erfolgt in Form einer nicht-hierarchischen Koordination.

Schritt 6: Auswahl geeigneter Kopplungsmechanismen

In diesem Schritt werden Kopplungsmechanismen ausgewählt, die geeignet sind, die im vorherigen Schritt modellierten Funktionskomponenten des Kopplungssystems zu unterstützen. Hierbei sind die spezifizierten Anforderungen an die AwS-Integration und die Restriktionen, die sich aus den gegebenen AwS-Kern-Schnittstellen ergeben, zu berücksichtigen.

Schritt 7: Erstellung des Subsystemmodells

Nun wird ein Subsystemmodell (Ebene 4) erstellt, das geeignet ist, die in Form von Funktionskomponenten auf Ebene 3 beschriebene Nutzermaschine des Kopplungssystems zu realisieren. Bei der Erstellung des Modells müssen die durch die AwS-Kern-Schnittstelle und die in Schritt 6 gewählten Basismaschinen gegebenen Restriktionen berücksichtigt werden. Auch hier gilt: Stellt

sich bei der Erstellung des Modells heraus, dass die Restriktionen nicht eingehalten werden können, so muss die Auswahl der Basismechanismen korrigiert bzw. die AwS-Kern-Schnittstelle entsprechend angepasst werden.

Die Erfassung der AwS-Kern-Schnittstelle erfolgt auf dieser Ebene in Form der Subsysteme des AwS-Kerns und deren Schnittstellen sowie in Form von Schnittstellen, die durch Subsysteme des Kopplungs-Teilsystems für den AwS-Kern bereitgestellt werden. Zur detaillierten Beschreibung der syntaktischen Aspekte einer Schnittstelle wie z. B. Datentypen, Signaturen der Operationen, Attribute oder Ausnahmen kann auf die IDL (Interface Definition Language) der OMG zurückgegriffen werden [OMG03a]. Ein großer Nachteil der IDL liegt in der mangelnden Unterstützung von semantischen Informationen, die meist nur in Form von Kommentarfeldern in natürlicher Sprache eingefügt werden können. Zur erweiterten Dokumentation von Schnittstellen bzw. Operationen kann das von Meyer entwickelte und heute im Bereich des Software Engineering weit verbreitete Prinzip des Design by Contract genutzt werden. Dieser Ansatz sieht eine vertragliche Vereinbarung zwischen Nutzer und Anbieter in Form von Vorbedingungen, Nachbedingungen und Invarianten vor [Meye97, 331ff.]. Zur formalen Beschreibung dieser Konstrukte bietet sich die Spezifikationssprache OCL (Object Constraint Language) an, die Teil der offiziellen UML-Spezifikation ist [OMG03b, 617ff.]. Neben diesen syntaktischen und semantischen Aspekten einer Schnittstelle sind ebenfalls qualitative Eigenschaften wie z. B. Performance oder Zuverlässigkeit zu dokumentieren [CBB+03, 228ff.].

Häufig ist es erforderlich, direkt auf ein Datenbankverwaltungssystem des AwS-Kerns zuzugreifen. Im Bereich der am Markt vorherrschenden relationalen Datenbanksysteme hat sich die Sprache SQL als Schnittstellenstandard etabliert. Die Schnittstelle zum AwS-Kern kann in einem solchen Fall grundsätzlich durch Angabe der unterstützten SQL-Version beschrieben werden. Diese sehr allgemeine Beschreibung muss jedoch noch durch die Erfassung der vom Kopplungs-Teilsystem genutzten Tabellen detailliert werden. Hierzu kann der DDL-Anteil (Data Definition Language) von SQL genutzt werden.

Schritt 8: Erstellung des Prozessmodells

In Schritt 8 der Methodik wird das Subsystemmodell in ein entsprechendes Prozessmodell (Ebene 5) umgesetzt. Hierzu werden die Subsysteme, unter Berücksichtigung von Anforderungen an eine parallele Ausführung, Prozessen zugeordnet. Diese werden wiederum auf Prozessoren verteilt. Das Prozessmodell ist vor allem im Kontext der Anforderungen Skalierbarkeit, Verfügbarkeit und Performance hilfreich.

Schritt 9: Implementierung des Kopplungssystems

Im letzten Schritt der Methodik wird das Kopplungssystem entsprechend der spezifizierten Kopplungsarchitektur implementiert. Ziel dieses Schrittes ist die Entwicklung eines lauffähigen Kopplungssystems.

4 Zusammenfassung und Ausblick

Der vorliegende Beitrag stellt eine Methodik zur Entwicklung von Kopplungssystemen vor. Dabei werden auf verschiedenen Abstraktionsebenen Kopplungsarchitekturen entwickelt, die zur Unterstützung überbetrieblicher Geschäftsprozesse geeignet sind. Um die Komplexität typischer Kopplungen beherrschbar zu machen, bietet die Methodik sechs Modellebenen und neun Vorgehensschritte zur Entwicklung von Kopplungssystemen an. Folgende Modellebenen werden dabei unterschieden: die Ebene des überbetrieblichen Geschäftsprozesses, die AwS-Kern-Ebene, die Funktionen-Ebene, die Subsystem-Ebene, die Prozess-Ebene und die Implementierungs-Ebene. Die einzelnen Vorgehensschritte werden sequenziell durchlaufen, wobei aber Rücksprünge möglich sind.

Eine Anwendung der Methodik findet sich im Beitrag „Einsatz einer Entwicklungsmethodik für die überbetriebliche Integration von Anwendungssystemen im Rahmen einer Fallstudie aus der Automobilzulieferindustrie“ in diesem Tagungsband [EcSF04]. In diesem Beitrag wird die Entwicklung von Kopplungssystemen im Kontext des elektronischen Automobilzulieferer-Marktplatzes SupplyOn erläutert, über den Automobilzulieferer Güter bei ihren Lieferanten beschaffen können.

Zur durchgehenden Unterstützung des Entwicklers von der Geschäftsprozessmodellierung bis zur Implementierung des Kopplungssystems entsteht im Rahmen des Projekts ein Werkzeug namens Cobas.

Literatur

[CBB+03] *Clements, P.; Bachmann, F.; Bass, L.; Garlan, D.; Ivers, J.; Little, R.; Nord, R.; Stattford, J.*: Documenting Software Architectures – Views and Beyond. Boston 2003.

[DoC80] *Department of Commerce, National Bureau of Standards*: Guidelines for Security of Computer Application. Federal Information Processing Standards Publication 73, 1980.

[ebXM01] *ebXML Technical Architecture Project Team*: ebXML Technical Architecture Specification Version 1.0.4. http://www.ebxml.org/specs/ebTA.pdf, 2001, Abruf am 2003-01-10.

[ebXM03] *Electronic Business Extensible Markup Language (ebXML).* http://www.ebxml.org, Abruf am 2003-06-05.

[EcSF04] *Eckert, S.; Schissler, M.; Ferstl, O. K.*: Einsatz einer Entwicklungsmethodik für die überbetriebliche Integration von Anwendungssystemen im Rahmen einer Fallstudie aus der Automobilzulieferindustrie. In: *Bartmann, D. et al. (Hrsg.):* Überbetriebliche Integration von Anwendungssystemen – FORWIN-Tagung 2004. Aachen 2004, S. 41-59.

[EMR+oJ] *Eckert, S.; Mantel, S.; Reeg, Th.; Schissler, M.; Ferstl, O. K.; Sinz, E. J.*: Inter-company integration of application systems – a survey of development methodologies. FORWIN-Bericht, Nürnberg o. J. (in Vorbereitung).

[ESMS03] *Eckert, S.; Schissler, M.; Mantel, S.; Schäffner, C.*: Entwicklung von Kopplungsarchitekturen - Evaluierung einer Methodik anhand eines Beispiels aus der Automobilzulieferindustrie. In: *Sinz, E. J.; Plaha, M.; Neckel, P. (Hrsg.):* Modellierung betrieblicher Informationssysteme - MobIS 2003. Bonn 2003, S. 87-107.

[FAS+99] *Fischer, J.; Atzberger, M.; Städler, M.; Kambergs, P.; Hluchy, R.; Hoos, J.; Pauls, M.; Walter, F.; Steffen, T.; Dresing, H.; Rulle, A.; Brentano, F.*: MOVE – Objektorientierte Modelle und Werkzeuge für unternehmensübergreifende Informationssysteme im Rahmen des Electronic Commerce. Sammelband, Paderborn 1999.

[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. Aufl., München 2001.

[FHK+98] *Fischer, J.; Hammer, G.; Kern, U.; Rulle, A.; Städler, M.; Steffen, T.*: Verbundprojekt MOVE - Modellierung einer verteilten Architektur für die Entwicklung unternehmensübergreifender Informationssysteme und ihre Validierung im Handelsbereich. In: *Projektträger Informationstechnik des BMBF beim DLR e.V. (Hrsg.)*: Statusseminar des BMBF Softwaretechnologie, Bonn 1998.

[ISO01] *ISO*: Software engineering – Product quality – Part 1: Quality Model ISO/EIC 9126-1:2001. ISO, 2001.

[JBLN01] *Juric, M. B.; Basha, S. J.; Leander, R.; Nagappan, R.*: Professional J2EE EAI. Birmingham 2001.

[Krum97] *Krumbiegel, J.*: Integrale Gestaltung von Geschäftsprozessen und Anwendungssystemen in Dienstleistungsbetrieben. Wiesbaden 1997, zugl. Bamberg, Univ., Diss., 1997.

[MiMu03] *Miller, J.; Mukerji, J.*: MDA Guide Version 1.0. http://www.omg.org/cgi-bin/apps/doc?omg/03-05-01.pdf, 2003, Abruf am 2003-10-29.

[MES+02] *Mantel, S.; Eckert, S.; Schissler, M.; Ferstl, O. K.; Sinz, E. J.*: Entwicklungsmethodik für überbetriebliche Kopplungsarchitekturen von Anwendungssystemen. In: *Bartmann, D. (Hrsg.)*: Kopplung von Anwendungssystemen – FORWIN-Tagung 2002. Aachen 2002, S. 183-202.

[Meye97] *Meyer, B.*: Object-Oriented Software Construction. 2. Aufl., London 1997.

[MKR+00] *Mantel, S.; Knobloch, B.; Rüffer, T.; Schissler, M.; Schmitz, K.; Ferstl, O. K.; Sinz, E. J.*: Analyse der Integrationspotenziale von Kommunikationsplattformen für verteilte Anwendungssysteme. FORWIN-Bericht Nr. FWN-2000-009, Nürnberg 2000.

[OMG03a] *OMG*: OMG IDL Syntax and Semantics. http://www.omg.org/docs/formal/02-06-07.pdf, 2003, Abruf am 2003-07-21.

[OMG03b] *OMG*: Unified Modelling Language Specification Version 1.5. http://www.omg.org/docs/formal/03-03-01.pdf, 2003, Abruf am 2003-08-04.

[Popp94] *Popp, K. M.*: Spezifikation der fachlichen Klassen-Beziehungs-Struktur objektorientierter Anwendungssysteme auf der Grundlage von Modellen der betrieblichen Diskurswelt. Bamberg, Univ., Diss., 1994.

[Raue96] *Raue, H.*: Wiederverwendbare betriebliche Anwendungssysteme – Grundlagen und Methoden ihrer objektorientierten Entwicklung. Wiesbaden, 1996, zugl. Bamberg, Univ., Diss., 1996.

[SMFS01] *Schissler, M.; Mantel, S.; Ferstl, O. K.; Sinz, E. J.*: Unterstützung von Kopplungsarchitekturen durch SAP R/3. FORWIN-Bericht Nr. FWN-2001-008, Nürnberg 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. In: WIRTSCHAFTSINFORMATIK 44 (2002) 5, S. 459-468.

[Somm01] *Sommerville, I.*: Software Engineering. 6. Aufl., München 2001.