February 2005

Entwicklungsmethodiken zur Integration von Anwendungssystemen in überbetrieblichen Geschäftsprozessen - ein Überblick über ausgewählte Ansätze

Martin Schissler
Otto-Friedrich-Universität Bamberg

Stephan Mantel
Otto-Friedrich-Universität Bamberg

Sven Eckert
Otto-Friedrich-Universität Bamberg

Otto K. Ferstl
Otto-Friedrich-Universität Bamberg

Elmar J. Sinz
Otto-Friedrich-Universität Bamberg

In: Ferstl, Otto K, u.a. (Hg) 2005. *Wirtschaftsinformatik 2005: eEconomy, eGovernment, eSociety*; 7. Internationale Tagung Wirtschaftsinformatik 2005. Heidelberg: Physica-Verlag

ISBN: 3-7908-1574-8

Entwicklungsmethodiken zur Integration von Anwendungssystemen in überbetrieblichen Geschäftsprozessen – ein Überblick über ausgewählte Ansätze

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

Otto-Friedrich-Universität Bamberg

Zusammenfassung: Die überbetriebliche Integration von Anwendungssystemen ist ein wichtiger Erfolgsfaktor für eine durchgehende Automatisierung überbetrieblicher Geschäftsprozesse. Eine solche Integration erfordert eine Kopplung der beteiligten Anwendungssysteme durch Kopplungssysteme. Aufgrund der in der Regel hohen Komplexität der Kopplungen werden umfassende Ansätze benötigt, die v. a. die Aspekte Modellierung und Vorgehen im Rahmen einer Entwicklungsmethodik berücksichtigen. Der vorliegende Beitrag stellt zunächst eine Systematik zur Erfassung wesentlicher Aspekte solcher Ansätze vor. Anschließend werden vier ausgewählte Ansätze unter Verwendung dieser Systematik beschrieben.

Schlüsselworte: Anwendungssystem, Integration, Kopplungssystem, Entwicklungsmethodik, Modellierung, Vorgehensmodell

1 Einleitung

Ein überbetrieblicher Geschäftsprozess verbindet Aktivitäten mehrerer Unternehmen und stimmt diese ganzheitlich auf eine gemeinsame Zielsetzung hin ab. Beispiele hierfür bieten das Supply-Chain-Management oder Virtuelle Unternehmen. Ein wesentlicher Erfolgsfaktor bei der Gestaltung überbetrieblicher Geschäftsprozesse ist die Integration der zugehörigen Anwendungssysteme (AwS) durch geeignete Kopplungssysteme. Produkte und Plattformen zur Realisierung von Kopplungen werden am Markt in großer Breite angeboten, wie z. B. Microsoft® BizTalk™ oder IBM® WebSphere® MQ Integrator®. Angesichts der in der Regel hohen Komplexität der Kopplungen werden darüber hinaus umfassende methodische Ansätze benötigt, die auch die Aspekte Modellierung und Vorgehen bei der Entwicklung von AwS-Kopplungen berücksichtigen. Solche Ansätze sind in Teilen bereits verfügbar. Bei Durchführung eines Integrationsprojekts ist aus der Menge verfügbarer Ansätze ein geeigneter Ansatz auszuwählen, wobei die Selektion u. a.

durch unterschiedliche Blickwinkel, Begriffsysteme und Formen der Beschreibung erschwert wird. Der vorliegende Beitrag gibt einen Überblick über vier ausgewählte Entwicklungsmethodiken zur überbetrieblichen Kopplung von AwS. Er ist ein Ergebnis des im Rahmen des Bayerischen Forschungsverbundes Wirtschaftsinformatik (FORWIN) durchgeführten Projekts „Offene Anwendungssystem-Architekturen in überbetrieblichen Wertschöpfungsketten“ (OASYS). Zur Darstellung der Ansätze wird eine einheitliche Beschreibungssystematik verwendet. Eine solche Darstellung bietet einen guten Ausgangspunkt für die nachfolgende, im Rahmen dieses Beitrags nicht betrachtete, Auswahl eines Ansatzes.

In Kapitel 2 werden zunächst die Gestaltung überbetrieblicher Geschäftsprozesse und ihre Unterstützung durch gekoppelte AwS beleuchtet. Kapitel 3 gibt einen Überblick über die vier betrachteten Ansätze. Hierzu wird zunächst die erwähnte Beschreibungssystematik eingeführt und anschließend auf die Ansätze angewendet. In Kapitel 4 erfolgt eine Zusammenfassung der Ergebnisse.

2 Unterstützung überbetrieblicher Geschäftsprozesse durch gekoppelte Anwendungssysteme

Geschäftsprozesse innerhalb eines Unternehmens sind auf dessen Unternehmensziele ausgerichtet. Sie interagieren an den Unternehmensgrenzen mit Geschäftsprozessen benachbarter Unternehmen. Eine Zusammenfassung der interagierenden Geschäftsprozesse zu einem überbetrieblichen Geschäftsprozess mit abgestimmter Zielsetzung ist Gegenstand betriebswirtschaftlicher Konzepte, wie z. B. Supply-Chain-Management und Virtuelle Unternehmen [Chri98, S. 15ff.; Pico^{+}01, S. 2ff.]. Um die Potenziale überbetrieblicher Geschäftsprozesse auszuschöpfen, sind maschinelle Aufgabenträger in Form von AwS einzusetzen und unternehmensübergreifend zu koppeln. Damit können z. B. Potenziale wie die Verkürzung der Durchlaufzeiten von Aufträgen, die Senkung von Lagerbeständen und höhere Flexibilität bei der Auftragsabwicklung genutzt werden [Chri98, S. 31ff.; Pico^{+}01, S. 9ff.].

Die überbetriebliche Kopplung von AwS erfolgt mittels eines *Kopplungssystems*, das alle kopplungsrelevanten Elemente der AwS enthält. Bei jedem der beteiligten AwS wird zwischen dem Kern und dem Kopplungs-Teilsystem unterschieden [Mant^{+}04, S. 22f.]. Der *AwS-Kern* enthält die zu koppelnden Elemente sowie weitere, nicht kopplungsrelevante Elemente. Das *Kopplungs-Teilsystem* enthält diejenigen Elemente des AwS, die ausschließlich der Kopplung mit anderen AwS dienen. Es wird auf der Basis von *Kopplungsmechanismen* realisiert, die Dienste und Kommunikationsprotokolle für die Kopplung bereitstellen. Die Architektur eines Kopplungssystems wird in Form einer *Kopplungsarchitektur* spezifiziert.

3 Entwicklungsmethodiken zur Kopplung von Anwendungssystemen

Im folgenden Kapitel werden vier Ansätze zur Kopplung von AwS vorgestellt. Den wesentlichen Bestandteil dieser Ansätze bildet die jeweilige Entwicklungsmethodik. Bei der Auswahl der vier Ansätze wurde versucht, einen repräsentativen Überblick über aktuelle Entwicklungsmethodiken im überbetrieblichen Umfeld zu geben. Es handelt sich dabei um zwei Ansätze aus einer Kooperation von Wissenschaft und Praxis (MOVE, Juric et al.), einen Ansatz einer Standardisierungsinitiative (ebXML) sowie einen Ansatz aus einem Forschungsprojekt (OASYS).

Zur strukturierten und einheitlichen Darstellung der Ansätze wird eine Beschreibungssystematik eingeführt. Diese berücksichtigt das zugrunde liegende Untersuchungsproblem, die Entwicklungsmethodik und die Verfügbarkeit wiederverwendbarer Modellbausteine (Abbildung 1).

Das *Untersuchungsproblem* besteht aus den Untersuchungsobjekten und den Untersuchungszielen der Methodik [Fers79, S. 43]. Die *Untersuchungsobjekte* der Ansätze werden danach differenziert, ob der betrachtete Ansatz nur die überbetriebliche Kopplung von AwS oder zusätzlich auch die überbetrieblichen Geschäftsprozesse erfasst. Hinsichtlich der *Untersuchungsziele* wird zwischen Analyse und Gestaltung des jeweiligen Untersuchungsobjekts unterschieden.

Die *Entwicklungsmethodik* wird gemäß [Fers+01, S. 180] anhand des Vorgehens und der dabei unterstützten Modellebenen charakterisiert. Hinsichtlich der *Modellebenen* [Sinz02] werden unterschieden: (1) die *Anwendungsmodellebene,* deren Gegenstand ein fachlich orientiertes Modell der Diskurswelt ist, (2) die *Softwareentwurfsebene,* bestehend aus der Spezifikation der Kopplungsarchitektur in Form von Softwarekomponenten und deren Beziehungen, sowie (3) die *Implementierungsebene* für die Umsetzung der Spezifikation in das Kopplungssystem. Wesentliche Aspekte sind hierbei die verfügbaren Modellierungskonstrukte auf den Ebenen, Beziehungen zwischen den Ebenen und die methodische Unterstützung bei der Wiederverwendung von Modellbausteinen.

Hinsichtlich des *Vorgehensmodells* wird geprüft, ob die verwendeten Ansätze ein iteratives oder ein inkrementelles Vorgehen unterstützen. Ein *iterativer* Ansatz geht auf Basis einer verrichtungsorientierten Zerlegung der Gestaltungsaufgabe schrittweise vor und betrachtet in jedem Schritt das gesamte Aufgabenobjekt. Ein mehrfach zyklisches Durchlaufen der Schrittfolge ist dabei möglich, aber nicht zwingend. Ein *inkrementeller* Ansatz zerlegt im Gegensatz dazu das Aufgabenobjekt der Gestaltungsaufgabe in Teilobjekte. Die Teilobjekte werden dann unabhängig voneinander oder aufeinander abgestimmt gemäß den Gestaltungszielen behandelt.

Zusätzlich zu den Modellebenen und zum Vorgehen wird untersucht, ob die Methodik den Entwickler bei der Spezifikation *nicht-funktionaler Anforderungen*, wie z. B. Skalierbarkeit oder Anpassbarkeit, unterstützt, die bei der Gestaltung von Kopplungssystemen zu berücksichtigen sind. Funktionale Anforderungen an das Kopplungssystem werden typischerweise in der Anwendungsmodellebene erfasst. Funktionale und nicht-funktionale Anforderungen konkretisieren folglich die Ziele bez. der Gestaltung des Kopplungssystems. Die Erfassung der nicht-funktionalen Anforderungen kann unstrukturiert oder in Form eines systematischen Kataloges erfolgen.

			MOVE	**ebXML**	**Juric et al.**	**OASYS**
U-Problem	Untersuchungs-objekt	Überbetriebl. GP	+	+	o	+
		Überbetriebl. KS	+	+	o	+
	Untersuchungs-ziel	Analyse GP	+	+	+	+
		Gestaltung GP	+	+	-	+
		Analyse KS	-	-	+	+
		Gestaltung KS	+	+	+	+
Methodik	Modellebenen	Anwendungsmodell	+	+	+	+
		Softwareentwurf	-	o	+	+
		Implementierung	+	+	+	-
	Vorgehen	Iterativ	+	+	+	+
		Inkrementell	o	+	+	+
	nicht-f. Anford.		-	o	o	+
Wiederv.	Verfügbarkeit wiederverwend-barer Bausteine	Anwendungsmodell	+	+	-	+
		Softwareentwurf	-	o	o	-
		Implementierung	+	+	o	-

GP: Geschäftsprozess KS: Kopplungssystem nicht-f. Anford.: nicht-funktionale Anforderungen

\+ trifft voll zu o trifft teilweise zu - trifft nicht zu

Abbildung 1: Einordnung der vier Ansätze in die Beschreibungssystematik

Abschließend wird untersucht, ob die betrachteten Ansätze eine *Wiederverwendung von Modellbausteinen* auf den einzelnen Modellebenen, z. B. in Form von Bausteinbibliotheken, unterstützen.

Zur Erläuterung der vier Ansätze werden in den jeweiligen Abschnitten Abbildungen verwendet, in denen die Vorgehensschritte des Ansatzes den einzelnen Modellebenen zugeordnet werden. Ein Vorgehensschritt wird einer Ebene zugeordnet, wenn er Modelle dieser Ebene erzeugt. Die Kanten in den Abbildungen stellen Reihenfolgebeziehungen zwischen den Schritten dar.

3.1 MOVE

Das Projekt MOVE - Modellierung einer verteilten Architektur für die Entwicklung unternehmensübergreifender Informationssysteme und ihre Validierung im Handelsbereich - ist ein Verbundvorhaben zwischen dem Beratungs- und Softwarehaus BFK GmbH, dem Hersteller von Kassen- und Warenwirtschaftssystemen ICL Retail Systems GmbH, dem Europäischen Handelsinstitut und dem Schwerpunkt Wirtschaftsinformatik 1 der Universität-GH Paderborn [Fisc+98].

Untersuchungsproblem

Das Untersuchungsobjekt des Projektes bilden überbetriebliche Geschäftsprozesse im Rahmen des Supply-Chain-Management und ihre Unterstützung durch Interorganisationssysteme (IOS). Ein IOS realisiert hierbei eine Menge von Kopplungs-Teilsystemen und kann autonom gegenüber den AwS-Kernen agieren [Fisc+99, S. 17f.]. Untersuchungsziel des Projektes ist zum einen die Analyse und Gestaltung überbetrieblicher Geschäftsprozesse mit besonderer Betonung der überbetrieblichen Informationsflüsse. Ein weiteres Ziel ist die Entwicklung eines auf die Geschäftsprozesse abgestimmten IOS auf der Basis einer im Rahmen des MOVE-Projektes entwickelten Architektur, die den Austausch von Geschäftsdokumenten ermöglicht [Fisc+98; Fisc+99, S. 24]. Die AwS-Kerne werden als nicht direkt veränderbar betrachtet [Fisc+99, S. 17].

Methodik

Im MOVE-Ansatz erfolgt keine explizite Differenzierung von Modellebenen [Fisc+99, S. 10f.]. Es können jedoch die Anwendungsmodellebene und die Implementierungsebene identifiziert werden.

Auf der *Anwendungsmodellebene* wird ein Wertschöpfungsfluss auf Basis eines objektorientierten Metamodells modelliert. Das Metamodell umfasst die vier Entwurfselemente Klient, Interaktion, Objekt und Kanal. *Klienten*, z. B. Unternehmen oder Haushalte, tauschen Leistungen, Zahlungen und Informationen in Form von *Interaktionen* aus. Diese werden durch das *Objekt* als Gegenstand der Interaktion und den zu ihrer Realisierung notwendigen *Kanal* näher bestimmt. Jedes Entwurfselement kann durch Attribute, Methoden und Rollen beschrieben werden [Fisc+99, S. 11ff., S. 159]. Sämtliche Modellelemente müssen aus einer von allen beteiligten Unternehmen akzeptierten Referenzbibliothek stammen [Fisc+99, S. 162].

Der MOVE-Ansatz unterscheidet auf der Anwendungsmodellebene drei Sichten. (1) Im *Prozessübersichtsmodell* werden alle am jeweiligen Wertschöpfungsfluss beteiligten Klienten und ihre Interaktionen modelliert [Fisc+99, S. 169f.]. (2) *Interaktionsmodelle* beschreiben detailliert die Interaktionen von genau zwei in einer Geschäftsbeziehung stehenden Klienten. Hierbei werden für jede Interaktion

das übertragene Objekt und der verwendete Kanal angegeben [Fisc+99, S. 171]. (3) *Strukturmodelle der Informationsobjekte* beschreiben die Attribute und Methoden eines im Rahmen einer Informations-Interaktion übertragenen Objektes [Fisc+99, S. 171ff.].

Die *Implementierungsebene* umfasst die Implementierung eines IOS auf der Basis eines Java™-Framework. Alternativ ist auch eine Implementierung auf der Basis von Komponententechnologien, wie DCOM oder CORBA®, möglich, über die jedoch keine weiteren Aussagen getroffen werden [Fisc+99, S. 16]. Daher wird auf diese Alternative im Folgenden nicht weiter eingegangen.

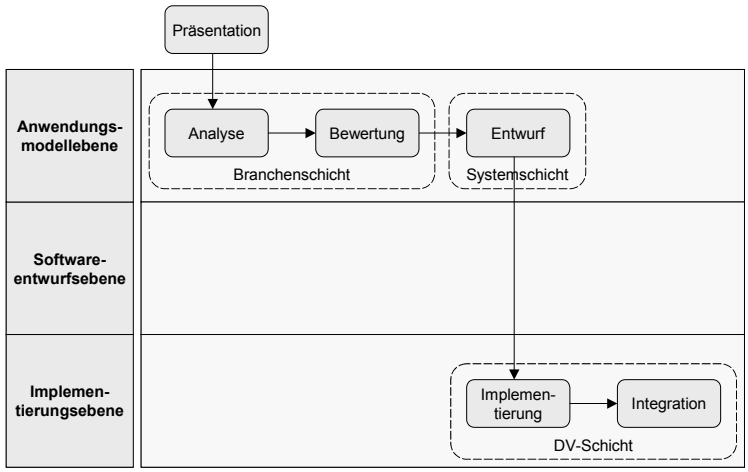

Abbildung 2: Modellebenen und Phasen des MOVE-Ansatzes

Im MOVE-Ansatz erfolgt aus Entwicklersicht ein direkter Übergang von der Anwendungsmodell- auf die Implementierungsebene. Die Softwareentwurfsebene wird durch das MOVE-Java™-Framework vorgegeben. Der Entwickler erstellt keinen Softwareentwurf eines IOS, sondern kann aus den Modellen der Anwendungsmodellebene Software-Artefakte in Form von Java™-Klassen erzeugen und in das Framework einbinden [Fisc+99, S. 184].

MOVE gliedert das Vorgehen bei der Entwicklung eines IOS in die folgenden Vorgehensschritte, die als Phasen bezeichnet werden (Abbildung 2):

- In der Phase *„Präsentation“* können sich interessierte Unternehmen über die Anwendung des MOVE-Ansatzes informieren [Fisc+99, S. 228f.].
- In der *„Analyse“* wird der überbetriebliche Geschäftsprozess untersucht und das Nutzenpotenzial einer MOVE-Implementierung identifiziert. Im Rahmen einer Nutzwertanalyse wird hierzu die Wirkung einer solchen Implementierung auf Kosten und Umsatz bestimmt [Fisc+99, S. 93ff., S. 229].

- Durch Simulation verschiedener Geschäftsprozessalternativen wird in der Phase *„Bewertung“* versucht, eine geeignete Gestaltung des überbetrieblichen Geschäftsprozesses zu ermitteln [Fisc+99, S. 121ff., S. 229].
- In der Phase *„Entwurf“* wird ein Modell der gewählten Alternative des überbetrieblichen Geschäftsprozesses erstellt. Ziel ist die detaillierte Beschreibung der Informationsobjekte und ihrer Beziehungen [Fisc+99, S. 153, S. 229].
- Die *„Implementierung“* umfasst die Generierung von Java™-Klassen aus den im Entwurf beschriebenen Entwurfselementen und deren Einbindung in das Java™-Framework [Fisc+99, S. 183, S. 230].
- In der Phase *„Integration“* erfolgt die Einbettung der implementierten Klassen in die Systemumgebung des Unternehmens [Fisc+99, S. 230].

Zur Strukturierung des Ansatzes werden von MOVE die Schichten Branchen-, System- und DV-Schicht unterschieden. Diese können als grobe Phasen bei der Entwicklung eines IOS verstanden werden. Hierbei umfasst die Branchenschicht die Phasen Analyse und Bewertung, die Systemschicht die Phase Entwurf und die DV-Schicht die Phasen Implementierung und Integration.

Das Vorgehensmodell von MOVE ermöglicht den Rücksprung zu früheren Phasen und ist daher als zyklisch iterativ zu bezeichnen [Fisc+99, S. 19]. Des Weiteren erlaubt es ein inkrementelles Vorgehen, indem Teilprojekte abgegrenzt und unabhängig bearbeitet werden („Tintenklecksansatz“). Bei der Zusammenführung der Inkremente müssen diese daher eventuell angepasst werden. Wie dies erfolgen soll, wird nicht näher erläutert [Fisc+99, S. 232].

Der MOVE-Ansatz erfasst keine nicht-funktionalen Anforderungen bez. der Gestaltung eines IOS, da das Design und die Implementierung des IOS durch das Java™-Framework vorgegeben sind. Somit können in einem konkreten Integrationsprojekt keine spezifischen diesbezüglichen Anforderungen berücksichtigt werden.

Verfügbarkeit wiederverwendbarer Modellbausteine

Auf der Anwendungsmodellebene stellt MOVE eine Referenzbibliothek zur Verfügung. In dieser Bibliothek sind Referenzklassen für die Entwurfselemente Klient, Objekt, Interaktion und Kanal in Form von Klassendiagrammen zusammengefasst. Darüber hinaus enthält sie einen Rollenkatalog für Klienten und Objekte, sowie Informationsbausteine, aus denen sich die Attribute der zu modellierenden Informationsobjekte zusammensetzen [Fisc+99, S. 162]. Das Java™-Framework von MOVE stellt ein Wiederverwendungsangebot auf Implementierungsebene dar.

3.2 ebXML

Die *Electronic Business Extensible Markup Language* (ebXML) ist Gegenstand einer internationalen Initiative, die durch das „United Nations Centre for Trade Facilitation and Electronic Business" (UN/CEFACT) und die „Organization for the Advancement of Structured Information Standards" (OASIS) getragen wird. Das Kernziel dieser Initiative ist die Realisierung eines XML-basierten, offenen Framework, das eine einheitliche und konsistente Basis für den sicheren Austausch von Informationen im Kontext überbetrieblicher Integration schafft. ebXML sieht modular gegliederte Spezifikationen vor, durch deren Umsetzung eine Einführung des Framework realisiert wird [ebXM01a, S. 7].

Untersuchungsproblem

Untersuchungsobjekt von ebXML sind überbetriebliche Geschäftsprozesse und Kopplungssysteme. Die Untersuchungsziele beziehen sich auf die Analyse und Gestaltung der Geschäftsprozesse sowie die Entwicklung von Kopplungssystemen unter Berücksichtigung von Eigenschaften der Geschäftsprozesse [Busi01, S. 5f.]. Anpassungen bestehender AwS-Kerne sind nicht Gegenstand von ebXML.

Methodik

Das ebXML-Framework beinhaltet keine Methodik zur Entwicklung von AwS-Kopplungen. Es wird jedoch die Verwendung der UN/CEFACT Modeling Methodology (UMM) empfohlen, die auf der Unified Modeling Language (UML™) basiert und bei der es sich um eine Spezialisierung des Rational Unified Process® (RUP®) handelt [ebXM01b, S. 9]. Die UMM nimmt keine explizite Differenzierung von Modellebenen vor. Es kann jedoch eine der Beschreibungssystematik entsprechende Ebeneneinteilung identifiziert werden (Abbildung 3).

Zur Erfassung und Modellierung der Geschäftsprozesse auf der *Anwendungsmodellebene* werden gemäß UMM Geschäftsprozesse als Menge von Kollaborationen erfasst, die die Interaktionen zwischen Rollen im Geschäftsprozess als Abfolge von Geschäftstransaktionen beschreiben. Einer Geschäftstransaktion ist eine anfragende und eine antwortende Aktivität zugeordnet. Sie dient dem Austausch von Geschäftsdokumenten. Des Weiteren sieht die UMM auf der Anwendungsmodellebene eine initiale Analyse der Geschäftsdokumente vor, die unter Verwendung wiederverwendbarer, kontextabhängiger Bausteine, der ebXML Core Components (ebCC), definiert werden können. Zur Modellierung der Geschäftsprozesse werden u. a. Use-Case-, Activity- und Sequence-Diagrams, zur Spezifikation der Geschäftsdokumente Class-Diagrams eingesetzt [UNCE01, S. 2-1ff., S. 9-1ff.].

Die *Softwareentwurfsebene* besitzt im Rahmen der UMM eine untergeordnete Bedeutung. Das Kopplungssystem wird in Form von Kollaborationen zwischen

Netzwerkkomponenten beschrieben, die Dienste bereitstellen. Außerdem wird ein Informationsmodell erstellt, das die Geschäftsdokumente detailliert spezifiziert. Auf dieser Ebene werden Sequence-, Collaboration- und Class-Diagrams der UML™ verwendet [UNCE01, S. 5-1ff., S. 9-41ff.].

Auf der *Implementierungsebene* erfolgt die Konfiguration eines der *Message Service Specification* entsprechenden ebXML Message Service (ebMS) [OASI02a]. Diese legt technische Aspekte des Austausches von ebXML-Nachrichten fest, beschreibt den Aufbau der Nachrichten und definiert Dienste wie Routing oder Packaging. Eine ebMS Implementierung stellt einen Kopplungsmechanismus zur Realisierung eines ebXML-konformen Kopplungs-Teilsystems dar. Sie beinhaltet das Message Service Interface (MSI) als Schnittstelle zu bestehenden AwS-Kernen, sowie einen Message Service Handler (MSH), der die Funktionalität des ebMS kapselt [OASI02a, S. 8ff.].

Die Konfiguration des ebMS erfolgt auf der Basis verschiedener XML-Dokumente: (1) ebXML Business Processes Specification (ebBPS), (2) Collaboration Protocol Profile (CPP) und (3) Collaboration Protocol Agreement (CPA). Eine ebBPS enthält die Informationen über einen Geschäftsprozess, die für einen ebMS relevant sind, basierend auf dem *ebXML Business Process Specification Schema* (ebBPSS) als Metamodell. ebBPSS ist eine Untermenge des UMM-Metamodells, wodurch eine automatisierte Transformation der mit UML™ modellierten Geschäftsprozesse in XML-basierte ebBPS ermöglicht wird [Busi01]. Die *Collaboration Protocol Profile and Agreement Specification* definiert ein Regelwerk für die Erstellung von CPP und CPA [OASI02b]. Das CPP dient der Festlegung eines unternehmensspezifischen Profils auf Basis einer ebBPS. Es erfolgt eine Zuordnung des Unternehmens zu relevanten Rollen und Aktivitäten sowie eine Erweiterung des fachlich modellierten Geschäftsprozesses um technische Parameter, wie z. B. Verschlüsselungsanforderungen. Ein CPA, das die CPP zweier Unternehmen abgleicht, beschreibt fachliche und technische Protokolle, die für die Unternehmen bei der Durchführung der Interaktion bindend sind [Busi01, S. 9f.].

Bei Verwendung der UMM kommt das Vorgehensmodell des RUP® mit UMM- und ebXML-spezifischen Anpassungen zum Einsatz [Chap+01, S. 53ff.; ebXM01b, S. 7ff.; Kruc00; UNCE01]. Es werden folgende als Workflows bezeichnete Vorgehensschritte unterschieden:

- In den Workflows „*Geschäftsmodellierung*“, „*Erfassung Anforderungen*“ und „*Analyse*“ wird das Geschäftsprozessmodell erfasst und sukzessive verfeinert.
- „*Entwurf*“ umfasst die Abbildung der Kollaborationen des Geschäftsprozessmodells auf konkrete Dienste und die Verfeinerung der Geschäftsdokumente.
- Im Workflow „*Implementierung*“ werden eine ebBPS und zugehörige XML-basierte Geschäftsdokument-Spezifikationen auf der Grundlage der UML™-Modelle erzeugt. Basierend auf der ebBPS werden unternehmensspezifische Profile im XML-Format erstellt. Durch den Abgleich zweier solcher CPP wird

anschließend ein CPA generiert. Zuletzt erfolgt die Konfiguration des ebMS auf der Grundlage dieses CPA [Busi01, S. 5f.].

- Die beiden letzten Workflows umfassen das „*Testen*“ und die „*Einführung*“ des Kopplungssystems.

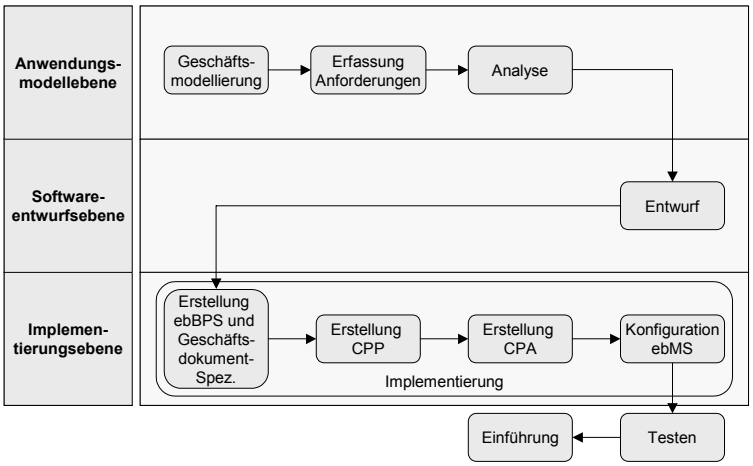

Abbildung 3: Modellebenen und Workflows des ebXML-Framework

Der RUP® erlaubt ein zyklisch iteratives und teilweise abgestimmtes inkrementelles Vorgehen. Das Vorgehensmodell unterscheidet, zusätzlich zu den genannten Workflows, die vier von einem Softwareprojekt zu durchlaufenden Phasen Initial-, Ausarbeitungs-, Konstruktions- und Übergangsphase. In jeder dieser Phasen werden null, ein oder mehrere Inkremente erstellt. Im Rahmen der Erstellung eines Inkrements werden die genannten Workflows durchgeführt. Je nach zugehöriger Phase werden die Workflows hierbei unterschiedlich stark betont [Kruc00, S. 53ff.].

Die Definition nicht-funktionaler Anforderungen wird nicht als expliziter Vorgehensschritt berücksichtigt. Im Rahmen der Erstellung der UMM-Modelle sowie von ebBPS und CPP können jedoch verschiedene Anforderungen erfasst werden, z. B. Sicherheits-, Verbindlichkeits- und Zuverlässigkeitsanforderungen bez. der Transaktionen. Ein resultierendes CPA beinhaltet sämtliche funktionale und nichtfunktionale Anforderungen, die für die Realisierung einer unternehmensübergreifenden Integration bindend sind [OASI02b, S. 11].

Verfügbarkeit wiederverwendbarer Modellbausteine

Auf Anwendungsmodellebene stehen wiederverwendbare Modellbausteine in Form von ebCC zur Verfügung [UNOA01]. Außerdem beinhaltet die UMM sechs Analysemuster, die bei der Erfassung des Geschäftsprozesses eingesetzt werden

können [UNCE01, S. 8-3ff.]. Auf der Softwareentwurfsebene stellt die UMM fünf korrespondierende Designmuster zur Modellierung der Interaktionen zwischen Diensten bereit [UNCE01, S. 8-32ff.]. Bez. der Implementierungsebene kann auf verschiedene, einen ebMS implementierende Integrationsprodukte zurückgegriffen werden, wie z. B. BEA Weblogic Integration. Außerdem ist zu erwarten, dass zukünftig über öffentliche ebXML Registries auf weitere Bausteine zugegriffen werden kann [Chap+01, S. 257].

3.3 Juric et al.

In [Juri+01] beschreiben Matjaz B. Juric, S. Jeelani Basha, Rick Leander und Ramesh Nagappan wie mithilfe der von J2EE™ bereitgestellten Mechanismen AwS gekoppelt werden können.

Untersuchungsproblem

Der Ansatz beschäftigt sich im Schwerpunkt mit der Gestaltung von Kopplungssystemen zur Integration der AwS eines Unternehmens. Das wesentliche Ziel ist dabei die Entwicklung eines *Composite-Information-System* [Juri+01, S. 33f., S. 75]. Hierbei handelt es sich um ein neues AwS, das auf den bestehenden AwS aufbaut und diese, ausgehend von einer Analyse ihrer Eigenschaften, durch ein entsprechendes Kopplungssystem integriert. Die bestehenden AwS-Kerne werden bei diesem Vorgehen durch *virtuelle Komponenten* gekapselt, die Bestandteile des Kopplungs-Teilsystems darstellen [Juri+01, S. 77ff.]. Eine Anpassung bestehender AwS-Kerne wird nicht betrachtet. Die Geschäftsprozesse werden hinsichtlich ihrer für die Entwicklung des Kopplungssystems relevanten Eigenschaften analysiert. Die Gestaltung der Geschäftsprozesse steht nicht im Fokus des Ansatzes.

Die überbetriebliche Integration von AwS stellt eher ein Randthema des Ansatzes dar. Allerdings werden weite Teile der hierzu notwendigen überbetrieblichen Kopplungsarchitektur bereits durch die Architektur des Composite-Information-System abgedeckt. Zusätzlich wird beschrieben, wie dessen Architektur zu erweitern ist, um eine solche Kopplung zu realisieren [Juri+01, S. 102ff., S. 851ff.].

Es werden vier Stufen der Integration unterschieden. (1) Die *Data-Level-Integration* beschäftigt sich mit der gemeinsamen Nutzung von Daten durch mehrere AwS durch gegenseitigen Zugriff auf deren Datenbanken [Juri+01, S. 80ff.]. (2) Die *Application-Interface-Level-Integration* beschäftigt sich mit der gemeinsamen Nutzung von Funktionen und Daten durch mehrere AwS unter Verwendung vorhandener bzw. neu zu implementierender API (Application-Programming-Interface) der AwS [Juri+01, S. 85ff.]. (3) Die gleiche Zielsetzung wird durch die *Business-Method-Level-Integration* verfolgt [Juri+01, S. 91ff.]. Im Gegensatz zur Application-Interface-Level-Integration, bei der die virtuellen Komponenten nur eine technische Abstraktion der AwS-Kerne bereitstellen, erfolgt

hier zusätzlich eine fachliche Abstraktion. Die virtuellen Komponenten werden entsprechend den Anforderungen des angestrebten Composite-Information-System fachlich abgegrenzt. (4) Mithilfe der *Presentation-Level-Integration* wird den Nutzern eine einheitliche Oberfläche angeboten, die den Zugriff auf die Funktionen und Daten der verschiedenen AwS erlaubt [Juri+01, S. 97ff.].

Methodik

Eine explizite Unterscheidung von Modellebenen wird durch den Ansatz nicht vorgenommen. Die verwendeten Modelle lassen sich jedoch den in Abbildung 1 genannten drei Modellebenen zuordnen. Zur Darstellung der Modelle nutzt der Ansatz in weiten Teilen UML™-Diagramme [OMG03]. Teilweise werden auch tabellarische Darstellungen verwendet. Zu bemängeln ist die teilweise geringe Durchgängigkeit.

Auf der *Anwendungsmodellebene* kommt den Use-Case-Diagrams der UML™ eine zentrale Bedeutung zu [Juri+01, S. 167ff.]. Im Rahmen der Business-Method-Level-Integration werden zusätzlich fachliche Class-Diagrams und zugehörige Sequence-Diagrams eingesetzt [Juri+01, S. 402ff.].

Auf der *Ebene des Softwareentwurfs* werden Modelle zur Erfassung von Eigenschaften der bestehenden AwS erstellt, sowie Modelle, die beschreiben wie die formulierten Anforderungen durch eine Kopplungsarchitektur umgesetzt werden können. Hierfür wird vor allem auf die durch die UML™ angebotenen Component-, Deployment- und Sequence-Diagrams zurückgegriffen [Juri+01, S. 415ff.]. Komponenten dienen dabei zur Abbildung der bestehenden AwS und zur Beschreibung der zu entwickelnden virtuellen Komponenten. Im Rahmen der Data-Level-Integration werden zum Zwecke der Datenmodellierung in Abhängigkeit vom verwendeten Datenbankmodell verschiedene Ansätze vorgeschlagen, z. B. das Entity Relationship Model [Juri+01, S. 215ff.].

Aufgrund der Konzentration auf J2EE™ werden auf der *Implementierungsebene* vor allem die Sprachelemente von Java™ sowie die verschiedenen von J2EE™ zusätzlich angebotenen Mechanismen verwendet, z. B. Java Database Connectivity (JDBC™), Java Message Service (JMS), Java Connector Architecture (JCA).

Juric et al. vermeiden die Vorgabe eines detaillierten Prozesses [Juri+01, S. 148]. Sie orientieren sich jedoch teilweise an verbreiteten Softwareentwicklungsprozessen, z. B. dem RUP® [Kruc00]. Wie in Abbildung 4 dargestellt, wird das Vorgehen bei der Entwicklung des Composite-Information-System anhand der von Juric et al. unterschiedenen vier Stufen der Integration in vier entsprechende *Integrationsphasen* gegliedert. Die Reihenfolge der Integrationsphasen ist dabei so gewählt, dass jeweils auf den Ergebnissen der vorherigen Phasen aufgebaut werden kann. In den Integrationsphasen sind verschiedene, z. T. sehr detailliert beschriebene *Aktivitäten* durchzuführen [Juri+01, S. 155ff.]:

- *„Erfassung der Anforderungen“* zielt auf die Ermittlung von funktionalen und nicht-funktionalen Anforderungen an das Composite-Information-System.
- In der Aktivität *„Analyse der existierenden AwS“* werden die relevanten Eigenschaften der bestehenden AwS erfasst.
- Die Aktivität *„Auswahl der Integrationsinfrastruktur“* beschäftigt sich mit der Auswahl der zu verwendenden Kopplungsmechanismen.
- In der Aktivität *„Analyse der Problemdomäne“* werden dann verschiedene fachliche Modelle des angestrebten Kopplungssystems erstellt.
- Die technische Umsetzung dieser Modelle wird anschließend in der Aktivität *„Entwurf“* erarbeitet.
- In den drei letzten Aktivitäten erfolgen schließlich *„Implementierung“*, *„Testen“* und *„Einführung“* des Kopplungssystems.

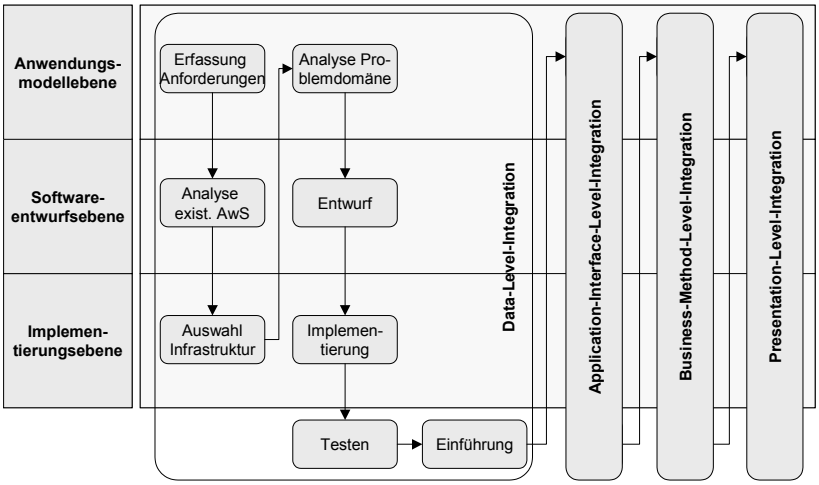

Abbildung 4: Modellebenen, Phasen und Aktivitäten des Ansatzes von Juric et al.

Der Ansatz unterstützt ein zyklisch iteratives und teilweise abgestimmtes inkrementelles Vorgehen [Juri+01, S. 154]. Er orientiert sich hierbei an den entsprechenden Vorschlägen des RUP® [Kruc00, S. 53ff.]. Die Abgrenzung der Inkremente erfolgt anhand der in der Aktivität „*Erfassung der Anforderungen*“ identifizierten Use-Cases.

Nicht-funktionale Anforderungen an das Kopplungssystem werden in [Juri+01] an verschiedenen Stellen berücksichtigt. Eine systematische Behandlung findet sich aber nicht.

Verfügbarkeit wiederverwendbarer Modellbausteine

Weite Teile von [Juri+01] beschreiben detailliert anhand von Programmierbeispielen wie auf der Grundlage von J2EE™ Kopplungsarchitekturen implementiert werden können. Diese Beispiele können grundsätzlich als Vorlagen bei der Implementierung von eigenen Kopplungsarchitekturen genutzt werden.

Der Softwareentwurf wird durch die Beschreibung verschiedener Muster unterstützt. Eine zentrale Rolle nimmt dabei das Architekturmuster ein, das den Aufbau eines Composite-Information-System beschreibt [Juri+01, S. 74ff., S. 96, S. 100]. Dessen Architektur gliedert sich in die drei Schichten User-Interface, Business-Logic und Data-Persistence [Juri+01, S. 76]. Die bestehenden AwS-Kerne finden sich in der Data-Persistence-Schicht wieder. Die Business-Logic-Schicht wird durch die bereits angesprochenen virtuellen Komponenten gebildet, die über einen Integration-Broker kommunizieren. Die im Rahmen der Presentation-Level-Integration entwickelten Nutzerschnittstellen bilden die User-Interface-Schicht. Zur Realisierung einer überbetrieblichen Kopplungsarchitektur wird die beschriebene Architektur durch Einfügen einer Web-Services-Schicht erweitert. Diese beinhaltet Web Services, die auf den virtuellen Komponenten der Business-Logic-Schicht aufbauen [Juri+01, S. 102ff.].

Allgemein lässt sich festhalten, dass Modellbausteine zwar zum Teil angeboten werden, dies erfolgt jedoch nicht in einer systematischen und strukturierten Form. Die Auswahl der Bausteine wird teilweise durch Beschreibung der Konsequenzen hinsichtlich der nicht-funktionalen Anforderungen unterstützt.

3.4 OASYS

Das Projekt „Offene Anwendungssystem-Architekturen in überbetrieblichen Wertschöpfungsketten“ (OASYS) wird an der Universität Bamberg im Rahmen des Bayerischen Forschungsverbundes Wirtschaftsinformatik (FORWIN) durchgeführt. Ziel des noch nicht abgeschlossenen Projektes ist die Entwicklung und Erprobung von Kopplungssystemen zur Integration heterogener AwS mehrerer Unternehmen. Dabei werden insbesondere Methoden und Werkzeuge zur Unterstützung der Integration entwickelt.

Untersuchungsproblem

Untersuchungsobjekt des OASYS-Ansatzes sind überbetriebliche Geschäftsprozesse im Rahmen von unternehmensübergreifenden Wertschöpfungsketten sowie die zugehörigen Kopplungssysteme. Ziel ist die Analyse und Gestaltung der überbetrieblichen Geschäftsprozesse, sowie die Entwicklung von Kopplungssystemen unter Berücksichtigung relevanter Eigenschaften der Geschäftsprozesse und der

bestehenden AwS-Kerne. Eine Anpassung bestehender AwS-Kerne wird nur hinsichtlich ihrer Schnittstellen zum Kopplungs-Teilsystem verfolgt.

Es werden drei Arten von Kopplungsarchitekturen unterschieden, die sich anhand des jeweils verfolgten Ziels unterscheiden [Mant+04, S. 23; Schi+02, S. 460ff.]. (1) *Ereignisorientierte Kopplungsarchitekturen* zielen auf die Übertragung von Ereignissen und zugehörigen Daten zwischen AwS durch den Austausch von Nachrichten in Form einer losen Kopplung [Fers+01, S. 225]. (2) *Datenorientierte Kopplungsarchitekturen* dienen der Manipulation gemeinsamer Daten mehrerer AwS in Form einer engen Kopplung der auf den Daten operierenden Funktionen [Fers+01, S. 225]. (3) *Funktionsorientierte Kopplungsarchitekturen* ermöglichen die gemeinsame Nutzung von Funktionen und ggf. zugehörigen Daten durch mehrere AwS.

Methodik

Der OASYS-Ansatz unterscheidet sechs Modellebenen [Mant+04], die den Ebenen aus der Beschreibungssystematik zugeordnet werden können (Abbildung 5).

Die *Ebene des überbetrieblichen Geschäftsprozesses* beschreibt diesen gemäß dem Modellierungsansatz des Semantischen Objektmodells (SOM) [Fers+01, S. 179ff.]. Ein Geschäftsprozess wird dabei als ein verteiltes System aus autonomen und lose gekoppelten betrieblichen Objekten modelliert, die sich mittels Transaktionen koordinieren. Die Struktur des Geschäftsprozesses wird im Interaktionsschema (IAS) in Form von betrieblichen Objekten und Transaktionen spezifiziert. Ein betriebliches Objekt umfasst eine Menge von Aufgaben, die zusammengehörige Ziele verfolgen und auf einem gemeinsamen Aufgabenobjekt durchgeführt werden. Das Verhalten wird im Vorgangs-Ereignis-Schema (VES) anhand von Vorgangstypen und Ereignisbeziehungen erfasst [Fers+01, S. 181f.]. Hierbei spezifiziert ein Vorgangstyp die Durchführung einer Aufgabe eines betrieblichen Objekts [Fers+01, S. 91]. Es werden domänenunabhängige und domänenabhängige Strukturmuster unterstützt [Fers+01, S. 189ff.; Fers+98]. Integrationsrelevante Teilbereiche des Geschäftsprozesses werden unter Verwendung von kategorisierten Aufgabenintegrations-Mustern (AIM) abgegrenzt. Diese stellen funktionale Anforderungen an das Kopplungssystem dar und spezifizieren zusammen mit den zu erfassenden nicht-funktionalen Anforderungen die zu lösenden Integrationsprobleme. Im Rahmen des Projekts wurde ein Konzept zur strukturierten Beschreibung von Integrationsproblemen in Form von wiederverwendbaren Mustern erarbeitet.

Die *AwS-Kern-Ebene* beschreibt die maschinellen Aufgabenträger der Aufgaben des Geschäftsprozesses in Form von zu koppelnden AwS-Kernen.

Die folgenden drei Ebenen, die Funktionen-, Subsystem- und Prozess-Ebene, spezifizieren die Architektur eines Kopplungssystems zur Kopplung von zwei oder mehreren AwS-Kernen aus verschiedenen Blickwinkeln. Die *Funktionen-Ebene*

beschreibt die Funktionalität des Kopplungssystems in Form von objektorientierten Funktionskomponenten, unabhängig von Kopplungsmechanismen, die zur Realisierung des Kopplungssystems verwendet werden können. Funktionskomponenten kapseln eine Menge von Operationen und verfügen über einen Speicher. Operationen tauschen Informationen über Kommunikationsbeziehungen aus.

Auf der *Subsystem-Ebene* des Kopplungssystems wird beschrieben, wie die Funktionskomponenten auf der Basis verfügbarer Kopplungsmechanismen durch eine Menge interagierender Subsysteme realisiert werden. Zentrales Modellierungselement ist das Subsystem, das über Schnittstellen eine Menge von Operationen anbietet und einen Speicher besitzt. Über Benutzt-Beziehungen werden Nachrichten zwischen Subsystemen ausgetauscht. Die Modellierungselemente dieser Ebene orientieren sich z. T. an der UML™ [OMG03].

Auf der *Prozess-Ebene* wird das Kopplungssystem als System parallel ablaufender Prozesse, z. B. Betriebssystemprozesse, dargestellt. Prozesse interagieren über Konnektoren, mit denen sie über Ports verbunden sind (vgl. z. B. [Clem+03, S. 103ff., S. 142ff.]). Nachrichten zwischen Prozessen werden über Konnektoren ausgetauscht. Weiterhin werden Prozesse auf Prozessoren verteilt.

Auf der *Ebene der Implementierung* erfolgt die Realisierung des Kopplungssystems anhand der entworfenen Kopplungsarchitektur. Hierbei kann z. B. auf bestehende Integrationsprodukte, wie Microsoft® BizTalk™ oder IBM® WebSphere® MQ Integrator® zurückgegriffen werden. Eine detaillierte Unterstützung dieser Ebene wird im Rahmen der OASYS-Methodik nicht angestrebt.

Das Vorgehen des Ansatzes gliedert sich in neun Schritte (Abbildung 5):

- Der Schritt *„Modellierung des überbetrieblichen Geschäftsprozesses“* umfasst die Analyse und Gestaltung des überbetrieblichen Geschäftsprozesses anhand der SOM-Methodik.
- Die Automatisierungsgrade der Aufgaben und Transaktionen des Geschäftsprozesses werden im Schritt *„Kartierung der AwS-Kerne“* bestimmt. Weiterhin erfolgt die Zuordnung der AwS-Kerne zu den von ihnen unterstützten Aufgaben.
- Im Schritt *„Identifikation von AIM“* werden im Rahmen einer Analyse des Geschäftsprozessmodells integrationsrelevante Teilbereiche in Form von AIM identifiziert. Ein solcher Ausschnitt eines Geschäftsprozessmodells spezifiziert die funktionalen Anforderungen an die Integration der entsprechenden AwS.
- Die *„Spezifikation von nicht-funktionalen Anforderungen an die AwS-Integration“* erfasst die nicht-funktionalen Anforderungen eines durch ein AIM abgegrenzten Integrationsbereiches.
- Im Schritt *„Erstellung des Funktionskomponentenmodells“* wird ausgehend von den spezifizierten Anforderungen die Funktionalität des Kopplungssystems in Form von Funktionskomponenten beschrieben.

- Bei der *„Auswahl geeigneter Kopplungsmechanismen“* werden Kopplungsmechanismen ausgewählt, die in der Lage sind, die im vorigen Schritt modellierten Funktionskomponenten zu unterstützen.
- Im Schritt *„Erstellung des Subsystemmodells“* wird ein Subsystemmodell entworfen, das geeignet ist, die in Form von Funktionskomponenten beschriebene Funktionalität des Kopplungssystems zu realisieren.
- Bei der *„Erstellung des Prozessmodells“* wird das Subsystemmodell in ein entsprechendes Prozessmodell umgesetzt. Hierzu werden die Subsysteme, unter Berücksichtigung von Anforderungen an eine parallele Ausführung, Prozessen zugeordnet.
- Im letzten Schritt erfolgt die *„Implementierung“* des Kopplungssystems auf der Basis der spezifizierten Kopplungsarchitektur.

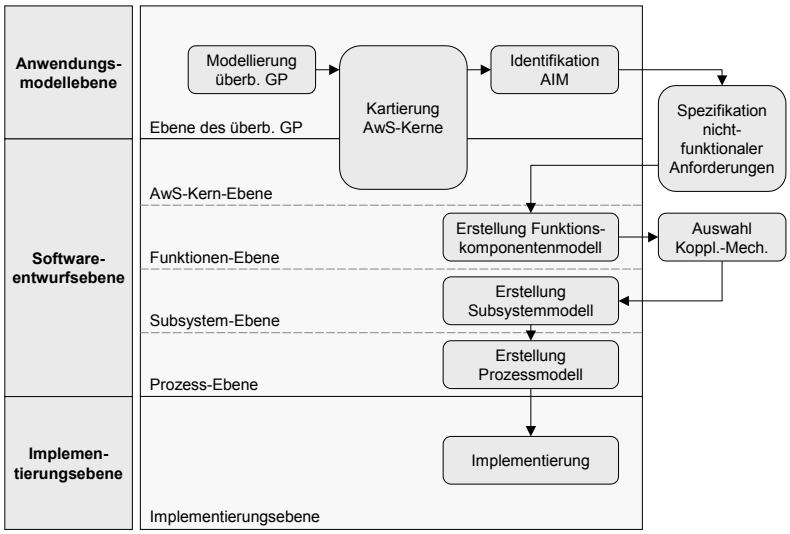

Abbildung 5: Modellebenen und Schritte des OASYS-Ansatzes

Der OASYS-Ansatz verwendet ein zyklisch iteratives Vorgehen, d. h. es erfolgt ein sequentieller Durchlauf durch die beschriebenen Schritte, wobei Rücksprünge zu früheren Schritten möglich sind. Bezüglich der einzelnen AIM ist ein inkrementelles Vorgehen mit abgestimmter Entwicklung der zugehörigen Kopplungssysteme vorgesehen.

Die Erfassung und Auswertung nicht-funktionaler Anforderungen an die Kopplung der AwS wird explizit unterstützt. Sie werden in Form eines strukturierten Kataloges angeboten, der die vier Kategorien Korrektheit, Echtzeitverhalten, Integration und Flexibilität unterscheidet [Fers92, S. 11]. Die den einzelnen Katego-

rien zugeordneten Anforderungen orientieren sich u. a. an den in der ISO-Norm 9126-1 beschriebenen Merkmalen [ISO01].

Verfügbarkeit wiederverwendbarer Modellbausteine

Auf der Ebene des überbetrieblichen Geschäftsprozesses stellt der Ansatz Referenzmodelle, z. B. für das Supply-Chain-Management-Konzept Vendor-Managed-Inventory, bereit. Es können im Rahmen des SOM-Ansatzes entwickelte domänenabhängige Strukturmuster sowie domänenunabhängige Strukturmuster genutzt werden [Fers+98; Fers+01, S. 189ff.]. Darüber hinaus erfolgte im Rahmen des Projekts OASYS u. a. eine Erfassung von Integrationsproblemen Elektronischer Marktplätze in Form von wiederverwendbaren Mustern [Wies04].

4 Zusammenfassung

Im vorliegenden Beitrag wurden vier Ansätze beschrieben, die zur Entwicklung von überbetrieblichen Kopplungssystemen eingesetzt werden können. MOVE, ebXML und OASYS unterstützen neben der Gestaltung von Kopplungssystemen zusätzlich die Gestaltung der überbetrieblichen Geschäftsprozesse. Alle vier Ansätze bieten eine Methodik und wiederverwendbare Modellbausteine. Unterschiede finden sich vor allem in der Betonung der in der Beschreibungssystematik eingeführten drei Modellebenen.

Im MOVE-Ansatz bewegt sich ein Entwickler vor allem auf der Anwendungsmodellebene und nur nachrangig auf der Implementierungsebene. Aus diesem Grund benötigt er kein ausgeprägtes technisches Fachwissen. Der Entwickler wird bei der Implementierung durch eine automatisierte Generierung von Java™-Klassen unterstützt. Diese Klassen werden in ein vorgegebenes Framework eingebunden.

Bei ebXML stehen die Anwendungsmodellebene und die Implementierungsebene im Vordergrund. Die Softwareentwurfsebene hat eine untergeordnete Bedeutung. Auf der Implementierungsebene wird ein ebXML-konformer Kopplungsmechanismus mittels spezifischer XML-Dateien konfiguriert. Der Entwickler benötigt aus diesem Grund, wie bei MOVE, nur relativ geringe technische Kenntnisse. Im Gegensatz zu MOVE ist hier aber eine stärkere Anpassung des Kopplungssystems an die konkreten nicht-funktionalen Anforderungen, z. B. Skalierbarkeit, möglich.

Der Ansatz von Juric et al. beschäftigt sich im Schwerpunkt mit der innerbetrieblichen Integration von AwS, leistet aber auch wesentliche Beiträge für die überbetriebliche Integration. Er betont alle drei Modellebenen in gleicher Weise. Die Implementierung der Kopplungsarchitektur erfolgt auf der Basis von J2EE™. Zur Anwendung des Ansatzes ist somit umfangreiches technisches Wissen insbesondere im Bereich der Implementierung notwendig.

Der OASYS-Ansatz konzentriert sich auf die Anwendungsmodellebene und auf die Softwareentwurfsebene. Der Ansatz versucht eine möglichst breite Abdeckung bez. der behandelten Integrationsprobleme zu erzielen. Das Integrationsproblem wird durch die Identifikation von AIM in Geschäftsprozessmodellen und die Erfassung zugehöriger nicht-funktionaler Anforderungen beschrieben. Anschließend wird eine Integrationslösung in Form einer Kopplungsarchitektur aus verschiedenen Blickwinkeln auf vier Entwurfsebenen spezifiziert, wodurch die Komplexität typischer Integrationsprojekte bewältigt werden kann.

Literatur

[Busi01] Business Process Project Team: ebXML Business Process Specification Schema Version 1.01. http://www.ebxml.org/specs/ebBPSS.pdf, 2001, Abruf am 2004-07-13.

[Chap+01] Chappel, D. A.; Chopra, V.; Dubray, J.-J.; Evans, C.; Harvey, B.; McGrath, T.; Nickull, D.; Noordzij, M.; Peat, B.; van der Eijk, P.; Vegt, J.: Professional ebXML Foundations, Wrox: Birmingham, 2001.

[Chri98] Christopher, M.: Logistics and Supply Chain Management - Strategies for Reducing Cost and Improving Service. 2. Auflage, Financial Times Prentice Hall: London, 1998.

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

[ebXM01a] ebXML Requirements Team: ebXML Requirements Specification Version 1.06. http://www.ebxml.org/specs/ebREQ.pdf, 2001, Abruf am 2004-07-13.

[ebXM01b] ebXML Technical Architecture Project Team: ebXML Technical Architecture Specification Version 1.0.4. http://www.ebxml.org/specs/ebTA.pdf, 2001, Abruf am 2004-07-13.

[Fers79] Ferstl, O. K.: Konstruktion und Analyse von Simulationsmodellen. Hain: Königstein/Ts., 1979.

[Fers92] Ferstl, O. K.: Integrationskonzepte Betrieblicher Anwendungssysteme. Fachberichte Informatik der Universität Koblenz-Landau, Nr.1/1992.

[Fers+98] Ferstl, O. K.; Sinz, E. J.; Hammel, Ch.; Schlitt, M.; Wolf, S.; Popp, K.; Kehlenbeck, R.; Pfister, A.; Kniep, H.; Nielsen, N.; Seitz, A.: WEGA - Wiederverwendbare und erweiterbare Geschäftsprozeß- und Anwendungssystemarchitekturen. Abschlussbericht, Walldorf, 1998.

[Fers+01] Ferstl, O. K.; Sinz, E. J.: Grundlagen der Wirtschaftsinformatik. 4. Auflage, Oldenbourg: München, 2001.

[Fisc+98] Fischer, J.; Hammer, G.; Kern, U.; Rulle, A.; Städler, M.; Steffen, T.: Verbundprojekt MOVE - Modellierung einer verteilten Architektur für die Entwicklung unter-

nehmensübergreifender Informationssysteme und ihre Validierung im Handelsbereich. In: Projektträger Informationstechnik des BMBF beim DLR e.V.: Statusseminar des BMBF Softwaretechnologie, Bonn, 23-24. März 1998.

[Fisc+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.

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

[Juri+01] Juric, M. B.; Basha, S. J.; Leander, R.; Nagappan, R.: Professional J2EE EAI. Wrox: Birmingham, 2001.

[Kruc00] Kruchten, P.: The Rational Unified Process - An Introduction. 2. Auflage, Addison Wesley: Boston, 2000.

[Mant+04] Mantel, S.; Eckert, S.; Schissler, M.; Schäffner, C.; Ferstl, O. K.; Sinz, E. J.: Eine Entwicklungsmethodik für die überbetriebliche Integration von Anwendungssystemen. In: Bartmann, D. et al. (Hrsg.): Überbetriebliche Integration von Anwendungssystemen - FORWIN-Tagung 2004. Shaker: Aachen, 2004, S. 21-39.

[OASI02a] OASIS ebXML Messaging Services Technical Committee: Message Service Specification. http://www.ebxml.org/specs/ebMS2.pdf, 2002, Abruf am 2004-07-13.

[OASI02b] OASIS ebXML Collaboration Protocol Profile and Agreement Technical Committee: Collaboration-Protocol Profile and Agreement Specification Version 2.0. http://www.ebxml.org/specs/ebcpp-2.0.pdf, 2002, Abruf am 2004-07-13.

[OMG03] OMG: Unified Modelling Language Specification Version 1.5. http://www.omg.org/docs/formal/03-03-01.pdf, 2003, Abruf am 2004-07-09.

[Pico+01] Picot, A.; Reichwald, R.; Wigand, R. T.: Die grenzenlose Unternehmung - Information, Organisation und Management. 4. Auflage, Gabler: Wiesbaden, 2001.

[Schi+02] 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.

[Sinz02] Sinz, E. J.: Architektur von Informationssystemen. In: Rechenberg, P.; Pomberger, G. (Hrsg.): Informatik-Handbuch. 3. Auflage, Hanser: München, 2002, S. 1055-1068.

[UNCE01] UN/CEFACT Techniques and Methodologies Working Group (TMWG): UN/CEFACT Modelling Methodology. http://www.ebxml.org/project_teams/jdt/resources/TMWG_N090R9.zip, 2001, Abruf am 2004-10-04.

[UNOA01] UN/CEFACT & OASIS: Core Components Dictionary Version 1.04. http://www.ebxml.org/specs/ccDICT.pdf, 2001, Abruf am 2004-10-12.

[Wies04] Wiese, S.: Integrationsprobleme bei elektronischen Marktplätzen - eine strukturierte Erfassung. Diplomarbeit am Lehrstuhl für Wirtschaftsinformatik, insb. Systementwicklung und Datenbankanwendung, Otto-Friedrich-Universität, Bamberg, 2004.