

February 2005

Strategie-, Prozess- und IT-Management: Ein Pattern-orientierter Integrationsansatz

Harald Kühn
BOC Information Systems GmbH

Dimitris Karagiannis
Universität Wien

Follow this and additional works at: <http://aisel.aisnet.org/wi2005>

Recommended Citation

Kühn, Harald and Karagiannis, Dimitris, "Strategie-, Prozess- und IT-Management: Ein Pattern-orientierter Integrationsansatz" (2005). *Wirtschaftsinformatik Proceedings 2005*. 78.
<http://aisel.aisnet.org/wi2005/78>

This material is brought to you by the Wirtschaftsinformatik at AIS Electronic Library (AISeL). It has been accepted for inclusion in Wirtschaftsinformatik Proceedings 2005 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

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

© Physica-Verlag Heidelberg 2005

Strategie-, Prozess- und IT-Management: Ein Pattern-orientierter Integrationsansatz

Harald Kühn

BOC Information Systems GmbH

Dimitris Karagiannis

Universität Wien

Zusammenfassung: Das Augenmerk von Enterprise Application Integration (EAI) liegt auf technischen Lösungen zur Integration und Interoperabilität heterogener Komponenten und Workflows für durchgehende, interorganisationelle Geschäftsanwendungen. Eine ganzheitliche Unternehmensintegration muss jedoch bereits auch auf der Strategie-, Geschäfts- und konzeptionellen Ebene beginnen. Aufgrund der Heterogenität von Modellierungssprachen und Methoden in diesen Bereichen gibt es keine einheitliche Modellierungssprache bzw. Methode in der modellgestützten Unternehmensintegration. Vielmehr wird ein unternehmensspezifisches Bündel von situations- und problemspezifischen Methoden und Methodenfragmenten benötigt, die durch Kombination und Integration die geforderte Hilfestellung für den Strategen, Architekten, Modellierer und Entwickler liefern. Im vorliegenden Beitrag werden Integrations-Patterns des Enterprise Model Integration Ansatzes (EMI) und deren Anwendung vorgestellt.

Schlüsselworte: Enterprise Model Integration, Integrations-Patterns, Methodenintegration, Metamodellierung, IT-gestützte Managementansätze

1 Einführung

Die Integration und Interoperabilität von Anwendungen zur Unterstützung unternehmensübergreifender Geschäftsprozesse stellt sowohl einen aktuellen Forschungsgegenstand als auch einen starken Interessenschwerpunkt in der Entwicklung betriebswirtschaftlicher Anwendungen dar [AiSc04, HaRe04]. Für die technische Integration zeigen sich Ansätze hilfreich wie Daten-orientierte, Benutzerschnittstellen-orientierte, Methoden-orientierte, Anwendungsschnittstellen-orientierte, Service-orientierte oder Prozess-orientierte Integration [Joha⁺00, Lint00, Bole⁺04].

Projekterfahrungen haben jedoch gezeigt, dass Anwendungsintegration bereits auf der Strategie-, Geschäfts- und konzeptionellen Ebene beginnen muss [Kühn⁺03].

Heutige Zielplattformen wie Workflow-Management-Systeme (WMS), Integrations-Server, ERP-Systeme, J2EE-Plattformen etc. nutzen größtenteils proprietäre Modellierungssprachen und Modellrepräsentationen. Eine ähnliche Vielfalt findet sich für die Beschreibung strategischer Aspekte, der Geschäftsprozesse und der organisatorischen Umgebung. Nahezu das einzige – jedoch recht prominente – Beispiel für eine Standardisierung auf Modellierungsebene ist die Unified Modelling Language (UML). Mit der Meta Object Facility (MOF) und der Model Driven Architecture (MDA) bestehen darüber hinaus Bestrebungen zur Nutzung von Objektorientierung, Meta-Konzepten und Modellierung zur Produktivitätssteigerung in der Softwareentwicklung und Anwendungsintegration [OMG02, OMG03].

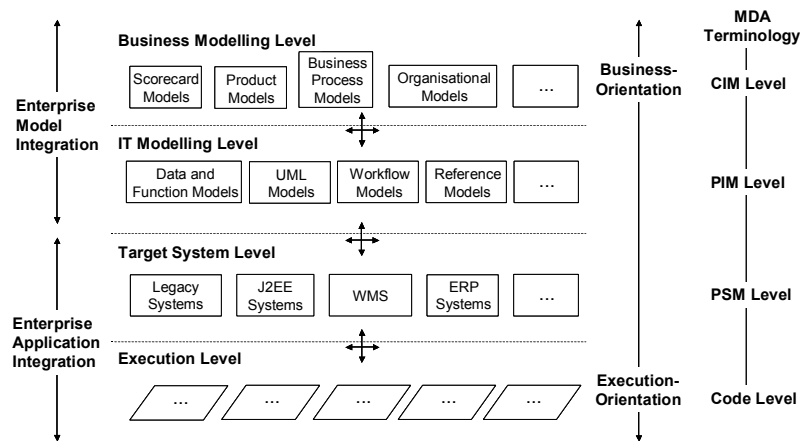


Abbildung 1: Modellierung in der Unternehmensintegration

Aufgrund der Heterogenität von Modellierungssprachen und Methoden im Bereich der Unternehmensmodellierung und EAI wird ein unternehmensspezifisches Bündel von Methoden und Methodenfragmenten benötigt. Diese liefern durch Kombination und Integration die geforderte Hilfestellung für den Strategen, Architekten, Modellierer und Entwickler. Im vorliegenden Beitrag werden Integrations-Patterns des *Enterprise Model Integration* Ansatzes vorgestellt.

In Kapitel 2 wird der Enterprise Model Integration Ansatz und der dort enthaltene Ansatz zur Metamodellintegration skizziert. Kapitel 3 gibt eine Einführung in Patterns für die Methodenintegration und stellt drei konkrete Patterns vor. In Kapitel 4 wird der Nutzung von Patterns bei der Methodenintegration anhand eines Anwendungsbeispiels demonstriert. Im abschließenden Kapitel 5 wird eine Zusammenfassung und ein Ausblick auf zukünftige Forschungs- und Entwicklungsthemen gegeben.

2 Enterprise Model Integration

Die Metamodellierung nimmt eine zentrale Rolle in der flexiblen Definition von inner- und zwischenbetrieblichen Aspekten ein [Kühn⁺03]. Davon ausgehend werden in Abschnitt 2.1 wichtige Konzepte der Methodendefinition und -integration vorgestellt. Abschnitt 2.2 gibt einen Kurzüberblick zum EMI Ansatz.

2.1 Methodendefinition und -integration

Der in diesem Beitrag verwendete Methodenbegriff besteht aus drei Elementen: den einzusetzenden Modellierungstechniken, den Mechanismen und Algorithmen zur Nutzung der Modellierungstechniken und dem Vorgehensmodell [Kühn⁺03]. Eine Modellierungstechnik beschreibt "Was bzw. welche Modelle werden mit einer Methode abgebildet?". Die Mechanismen und Algorithmen einer Methode beschreiben "Womit werden die Modelle erstellt und genutzt?". Das Vorgehensmodell beschreibt: "Wann und wie werden Modelle erstellt?".

Zur Wiederverwendung existierender Methodenelemente und deren Integration zu neuen Methoden bzw. Methodenvarianten, wird das Konzept des Methodenfragments genutzt. Methodenfragmente werden in einem Fragmentkatalog verwaltet. Ein Methodenfragment besteht aus einem Interface und einem Package. Das Interface wiederum besteht aus einer Fragmentspezifikation und einem oder mehreren Adaptoren. Die Fragmentspezifikation ist eine nicht formale Beschreibung eines Methodenfragments anhand seines Einsatzbereichs, Einsatzkontexts etc.

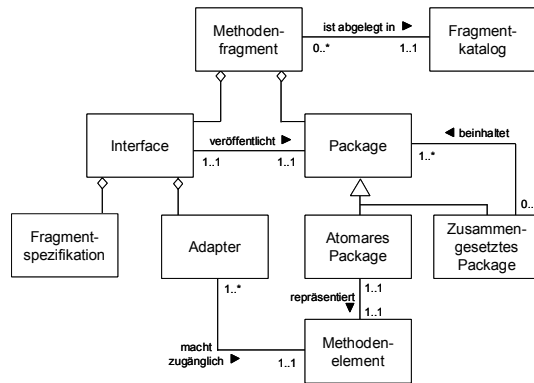


Abbildung 2: Metamodell eines Methodenfragments

Über die Adaptern werden die Elemente eines Package für die Integration zugänglich gemacht. Die Funktionalität eines Methodenfragments wird durch sein Package zur Verfügung gestellt. Ein atomares Package beschreibt eine einzelne, logisch-zusammengehörende und wiederverwendbare Einheit genau eines Methodenelements. In einem zusammengesetzten Package werden zwei oder mehrere

Packages zu einer logischen Einheit höherer Ordnung zusammengefasst. Die in einem zusammengesetzten Package enthaltenen (Sub-)Packages können sowohl atomare als auch zusammengesetzte Packages sein. In einem zusammengesetzten Package können auch Packages unterschiedlicher Methodenelemente gemischt werden. Die Verbindung der Packages innerhalb eines zusammengesetzten Package erfolgt durch Nutzung der Adaptern.

Innerhalb eines Package wird in Strukturelemente und Inhaltselemente unterschieden. Die Strukturelemente beschreiben die in einem Methodenelement enthaltenen Konzepte. Die Inhaltselemente repräsentieren die Eigenschaften der Strukturelemente und werden nicht direkt über die Adaptern zugänglich gemacht, sondern implizit über die assoziierten Strukturelemente. Für weiterführende Ausführungen zur Nutzung von Methodenfragmenten in der Methodendefinition und -integration vgl. [Kühn⁺03, Kühn04].

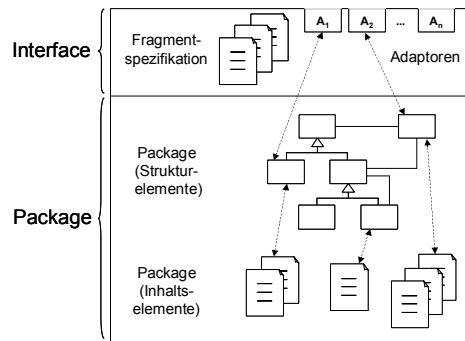


Abbildung 3: Interface und Package eines Methodenfragments

2.2 EMI: Übersicht und Komponenten

Die Ziele und Rahmenbedingungen des Einsatzes einer Methode beeinflussen die Auswahl und Konfiguration der Methodenelemente und deren Interdependenzen untereinander. Entsprechende Hilfestellung bei der Methodenintegration liefern die drei im Folgenden skizzierten Teilbereiche des EMI Ansatzes. Detaillierte Ausführungen hierzu finden sich in [Kühn⁺03, Kühn04].

- *Metamodellierung und Metamodellintegration*: Die Elemente einer Methode werden jeweils durch ihr Metamodell beschrieben. Zur Definition und Integration der Metamodelle werden u.a. das im Abschnitt 2.1 vorgestellte Fragmentkonzept und die in Kapitel 3 ausgeführten Integrations-Patterns genutzt.
- *Modelltransformation*: Wird eine Methode verändert, ändert sich dadurch das ihr zugrundeliegende Metamodell. Dabei stellt sich die Aufgabenstellung, bereits existierende Modelle eines Metamodells A auf Modelle basierend auf ei-

nem Metamodell B zu überführen. In EMI wird der BMT Ansatz für die regelbasierte Modelltransformation genutzt [Kühn⁺04].

- *Vorgehensmodell*: Die Anwendung einer Methode und die damit verbundene Ergebniserstellung wird durch Vorgehensmodelle beschrieben. EMI bietet ein Vorgehensmodell für das Vorgehen im Kontext der Methodenintegration.

3 Integrations-Patterns

Für die Definition und Integration von Methoden ist sowohl ein intensives Kennen der mit einer Methode verfolgten Ziele als auch die Rahmenbedingungen des Methodeneinsatzes sehr wichtig. Darüber hinaus trägt das Wissen von konkreten Vorgehensweisen und Basiskonzepten der Methodenintegration wesentlich zur effizienten und qualitätsgesicherten Gestaltung einer praxistauglichen Methode bei. Um dieses Wissen wiederverwendbar zu gestalten, wird im vorliegenden Beitrag ein Pattern-System zur Methodenintegration vorgestellt. Darauf aufbauend werden anschließend konkrete Integrations-Patterns erläutert.

3.1 Pattern-System

Mittels des Pattern-Systems können Integrations-Patterns in Form eines einheitlichen Templates beschrieben, klassifiziert und die Abhängigkeiten der Patterns untereinander verdeutlicht werden.

3.1.1 Pattern-Template

Die Wiederverwendung von Expertenwissen und von Erfahrungen ist in allen Bereichen menschlicher Arbeit von großem Interesse. Die Hauptvorteile sind dabei eine oft bedeutende Einsparung an Zeit und Kosten und die Erhöhung der Qualität der für eine Problemstellung erarbeiteten Lösung. Um das für die Methodenintegration wichtige Expertenwissen explizit und den damit verbundenen Erfahrungsschatz einem breiteren Publikum zugänglich zu machen, wird im vorliegenden Beitrag ein pattern-orientierter Ansatz eingesetzt [Gamm⁺94, ErPe00]. Zur Beschreibung eines Pattern wird dabei ein schlanker Ansatz verwendet, d.h. ein Template mit wenigen Beschreibungselementen¹:

¹ Gamma et al. beschreiben ein Pattern unter Nutzung eines Template mit 13 Merkmalen (vgl. [Gamm⁺94], S. 6f), Eriksson und Penker nutzen ein Pattern-Template mit 10 Merkmalen (vgl. [ErPe00], S. 175f).

Name: Der Name ist der eindeutige und aussagekräftige Bezeichner des Pattern. Durch den Namen müssen auch ungeübte Anwender intuitiv auf den Anwendungszweck des Integrations-Pattern hingeführt werden.

Kontext: Der Kontext beschreibt die Situation und die Rahmenbedingungen, unter denen ein konkretes Problem bei der Methodenintegration auftritt. Aus der Kontextbeschreibung muss für den Pattern-Nutzer klar erkennbar sein, wann der Einsatz eines bestimmten Pattern in Frage kommt.

Problem: Die Problembeschreibung spezifiziert das Integrationsproblem, welches durch die Anwendung des Pattern gelöst werden soll. Hierdurch weiß der Pattern-Nutzer, wofür er das Pattern einsetzen kann.

Lösung: Der Lösungsansatz beschreibt unter Zuhilfenahme von Richtlinien, Anweisungen und Regeln, wie das Integrations-Pattern in einem bestimmten Kontext zur Lösung eines bestimmten Problems eingesetzt werden kann. Die Struktur eines Pattern wird durch ein UML Klassendiagramm beschrieben.

Klassifikation: Die Klassifikation eines Pattern beschreibt dessen allgemeine Eigenschaften. Hierfür wird ein morphologischer Kasten verwendet. Die Klassifikation dient zusammen mit dem Kontext und der Problembeschreibung als Hilfestellung für die Auswahl eines Integrations-Pattern.

Beispiel: Mit einem oder mehreren Beispielen wird das Anwendungsszenario eines Pattern verdeutlicht. Dabei soll ein Beispiel ein vollständiges Szenario enthalten, d.h. die Anwendung der Lösung auf ein gegebenes Problem in einem spezifischen Kontext.

3.1.2 Pattern-Klassifikation

Die Klassifikation von Integrations-Patterns wird anhand von fünf Merkmalen vorgenommen. Diese Merkmale, inklusive ihrer Ausprägungen, sind:

Methodenelement: Ein Integrations-Pattern kann auf ein spezifisches Methodenelement fokussieren oder übergreifend für mehrere Methodenelemente angewendet werden. Es werden die Ausprägungen "Modellierungstechnik", "Mechanismen & Algorithmen", "Vorgehensmodell" und "Übergreifend" unterschieden.

Patternkategorie: In der Methodenintegration können zwei Patternkategorien unterschieden werden. Auf einer Makro-Ebene werden generelle Abhängigkeiten zwischen Methoden beschrieben. Dagegen beschreiben Patterns, die sich auf einzelne Fragmente einer Methode beziehen, die innere Sichtweise einer Methode. Es werden die Ausprägungen "Methode" und "Fragment" unterschieden.

Integrationsansatz: Bei der Integration von Methoden muss die Abstraktionsstufe der jeweils zu integrierenden Methodenelemente und deren Fragmente berücksichtigt werden. Dies spiegelt sich in unterschiedlichen Integrationsansätzen wider. Es werden die Ausprägungen "vertikal", "horizontal" und "hybrid" unterschieden.

Integrationsdichte: Die Integrationsdichte beschreibt die Abhängigkeit der zu integrierenden Methodenelemente bzw. deren Metamodelle. Sind die Metamodelle vollständig komplementär spricht man von schwacher Integration. Bei teilweiser oder sogar vollständiger Abhängigkeit spricht man von mittlerer und starker Integrationsdichte. Es werden die Ausprägungen "schwach", "mittel" und "stark" unterschieden.

Redundanz: Bei der Integration der Metamodelle der Methodenelemente können Redundanzen entstehen. Bspw. kann das aus einer Integration resultierende Metamodell zweier Organisationsmodellierungssprachen Elemente für Stellen, Positionen und Arbeitsplätze enthalten, die alle den selben Sachverhalt beschreiben. Ziel einer Methodenintegration sollte grundsätzlich die Vermeidung von Redundanzen im Zielmetamodell sein. Es werden die Ausprägungen "keine", "wenige" und "viele" unterschieden.

Merkmal	Ausprägung			
	Modellierungs- technik	Mechanismen & Algorithmen	Vorgehens- modell	Übergreifend
Methodenelement				
Patternkategorie	Methode		Fragment	
Integrationsansatz	vertikal	horizontal	hybrid	
Integrationsdichte	schwach	mittel	stark	
Redundanz	keine	wenige	viele	

Abbildung 4: Klassifikationsschema für Integrations-Patterns

3.1.3 Zusammenhänge der Integrations-Patterns

Bei der Nutzung von Integrations-Patterns ist es hilfreich, Abhängigkeiten zwischen den Patterns zu kennen [Kühn⁺03]. Deshalb gibt Abb. 5 eine Übersicht der wesentlichen Beziehungen zwischen den Patterns. Auf eine Strukturierung der Patterns anhand ihrer Klassifikation wurde in Abb. 5 aufgrund besserer Übersichtlichkeit verzichtet. Eine detaillierte Beschreibung aller Patterns findet sich in [Kühn04]. Dort wird eine Strukturierung der Integrations-Patterns in folgende Kategorien vorgeschlagen:

- Modellierungstechnik-Patterns,
- Mechanismen & Algorithmen-Patterns,
- Vorgehensmodell-Patterns und
- Übergreifende Patterns.

Im Folgenden werden drei Integrations-Patterns näher erläutert, die anschließend im Anwendungsbeispiel eingesetzt werden.

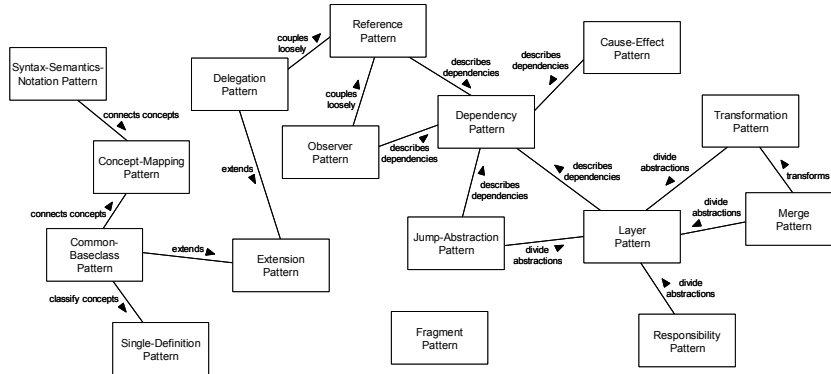


Abbildung 5: Pattern-System von Integrations-Patterns

3.2 Das Fragment-Pattern

Kontext: Bei der Methodenintegration werden die Methodenelemente nicht ausschließlich "als Ganzes" integriert, sondern es können auch nur Teile der Methodenelemente verwendet werden, um diese in eine neue oder bestehende Methode zu integrieren.

Problem: Bei der Integration unterschiedlicher Methodenelemente in eine neue oder bestehende Methode kann es Inkompatibilitäten zwischen den Methodenelementen bzw. den Bestandteilen der Methodenelemente kommen.

Lösung: Ein Methodenelement besteht aus mindestens einem Methodenfragment, ein Methodenfragment kann in keinem, einem oder mehreren Methodenelementen genutzt werden. Ein Methodenelement wird durch ein oder mehrere Interfaces beschrieben, ein Interface ist jedoch nur für genau ein Methodenelement gültig.

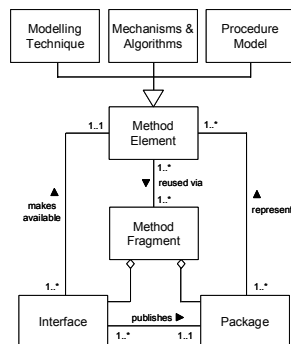


Abbildung 6: Lösungsstruktur des Fragment-Pattern

Die Funktionalität eines Methodenfragments wird durch ein oder mehrere Packages beschrieben. Der Zugang zu einem Package erfolgt über ein oder mehrere Interfaces, ein Interface stellt jedoch genau ein Package zur Verfügung. Hierdurch können die genannten Inkompatibilitäten umgangen werden, da ein Package, je nach Verwendungszweck, durch unterschiedliche Interfaces (vgl. Abschnitt 2.1) zugänglich gemacht werden kann (vgl. Abb. 6).

Klassifikation: Abb. 7 zeigt die Merkmalsausprägungen des Fragment-Pattern.

Merkmal	Ausprägung			
	Modellierungs- technik	Mechanismen & Algorithmen	Vorgehens- modell	Übergreifend
Methodenelement				
Patternkategorie	Methode		Fragment	
Integrationsansatz	vertikal	horizontal	hybrid	
Integrationsdichte	schwach	mittel	stark	
Redundanz	keine	wenige	viele	

Abbildung 7: Klassifikation des Fragment-Pattern

Beispiel: Es steht ein Mechanismenfragment mit einem Generatormechanismus zur Verfügung, um Modelle einer Modellierungstechnik A in eine XML-Datei zu generieren. Um diesen Generator auch für eine andere Modellierungstechnik B zu nutzen, wird das Generator-Package des Mechanismenfragments mittels eines für Modellierungstechnik B kompatiblen Interfaces zugänglich gemacht.

3.3 Das Concept-Mapping-Pattern

Kontext: Ein Ursprungsmetamodell enthält Konzepte, die in einem Zielmetamodell ebenfalls sinnvoll wären, jedoch momentan im Zielmetamodell nicht vorhanden sind bzw. die beiden Metamodelle enthalten ähnliche, jedoch nicht identische Konzepte, die aufeinander abgestimmt werden müssen. Dabei können die Metamodelle sowohl komplementär als auch konzeptionell sehr ähnlich sein.

Problem: (i) Ein Zielmetamodell soll um neue Konzepte erweitert werden. Diese Konzepte sind bereits in einem Ursprungsmetamodell enthalten. Um die Konzepte nicht vollständig neu definieren zu müssen, sollen die existierenden Konzepte aus dem Ursprungsmetamodell in das Zielmetamodell abgebildet werden. (ii) Das Ursprungs- und Zielmetamodell enthält jeweils ähnliche Konzepte. Diese müssen aufeinander abgebildet werden, um eine konsistente Modellierung zu ermöglichen.

Lösung: Um die Konzepte wiederzuverwenden bzw. aufeinander abzubilden, wird eine Abbildungsvorschrift (Concept-Mapping) eingeführt. Das Concept-Mapping kann ein oder mehrere Ursprungskonzepte auf ein oder mehrere Zielkonzepte abbilden. Sowohl ein Ursprungskonzept als auch ein Zielkonzept kann über ein oder mehrere Concept-Mappings verbunden werden (vgl. Abb. 8).

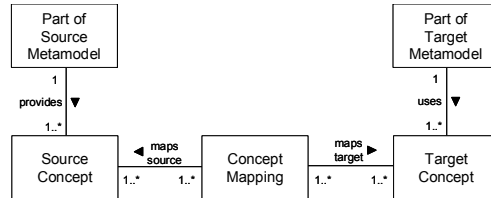


Abbildung 8: Lösungsstruktur des Concept-Mapping-Pattern

Klassifikation: Abb. 9 zeigt die Merkmalsausprägungen des Concept-Mapping-Pattern.

Merkmal	Ausprägung			
	Modellierungs- technik	Mechanismen & Algorithmen	Vorgehens- modell	Übergreifend
Patternkategorie	Methode		Fragment	
Integrationsansatz	vertikal	horizontal	hybrid	
Integrationsdichte	schwach	mittel	stark	
Redundanz	keine	wenige	viele	

Abbildung 9: Klassifikation des Concept-Mapping-Pattern

Beispiel: In einem Metamodell für die Organisationsmodellierung stehen detaillierte Konzepte zur Organisationsbeschreibung wie "Stelle", "Rolle", "Bearbeiter", "Organisationseinheit" etc. zur Verfügung. Diese werden auf das Konzept des Akteurs im Geschäftsprozessmetamodell abgebildet.

3.4 Das Merge-Pattern

Kontext: Zwei oder mehrere Ursprungsmetamodelle werden gemeinsam verwendet. Diese können nahezu gleiche Modellierungssprachen oder auch orthogonale Modellierungssprachen repräsentieren. Um den Pflegeaufwand der Metamodelle zu reduzieren bzw. um die Ausdruckskraft in einer integrierten Modellierungssprache zu nutzen, sollen die relevanten Metamodellausschnitte in ein gemeinsames Metamodell vereinigt werden.

Problem: Bei der Integration der relevanten Teile der Ursprungsmetamodelle in das Zielmetamodell darf die ursprüngliche Syntax, Semantik und Notation nicht verloren gehen bzw. fehlinterpretiert werden. Die Ursprungsmetamodelle dürfen nicht verändert werden.

Lösung: Das Merge-Pattern ist eine Spezialisierung des Transformation-Pattern. Eine Vereinigungsregel (Merge-Rule) generiert genau einen Teil des Zielmetamodells von zwei oder mehreren Ursprungsmetamodellen. Die Vereinigungsregel besteht aus mindestens einer Vereinigungsaktion und keiner, einer oder mehreren Vereinigungsbeschränkungen. Eine Vereinigungsbeschränkung bestimmt das

Verhalten der Vereinigungsaktion. Es werden drei Arten von Vereinigungsbeschränkungen unterschieden: syntaktische, semantische und notationelle Beschränkungen (vgl. Abb. 10).

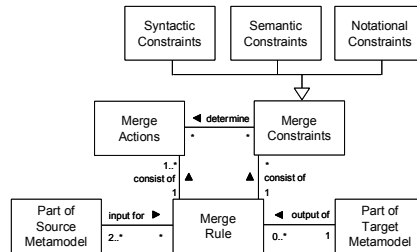


Abbildung 10: Lösungsstruktur des Merge-Pattern

Klassifikation: Abb. 11 zeigt die Merkmalsausprägungen des Merge-Pattern.

Merkmal	Ausprägung			
	Modellierungs- technik	Mechanismen & Algorithmen	Vorgehens- modell	Übergreifend
Methodenelement				
Patternkategorie	Methode		Fragment	
Integrationsansatz	vertikal	horizontal	hybrid	
Integrationsdichte	schwach	mittel	stark	
Redundanz	keine	wenige	viele	

Abbildung 11: Klassifikation des Merge-Pattern

Beispiel: Eine Datenmodellierungssprache und eine Geschäftsprozessmodellierungssprache sollen vereinigt werden, um Datenflussmodellierung zu ermöglichen. Die aus der Vereinigung resultierende Modellierungssprache soll weiterhin die Eigenschaften der Daten- und Geschäftsprozessmodellierung unterstützen.

4 Anwendungsbeispiel: Methodenintegration IT-gestützter Management-Ansätze

Erfolgreiche Unternehmen zeichnen sich u.a. dadurch aus, dass die lang- und mittelfristige IT-Strategie klar an der Unternehmensstrategie ausgerichtet und gewinnbringend in operatives Handeln umgesetzt wird. Um daran beteiligten Personen entsprechende Hilfsmittel an die Hand zu geben, wurden unterschiedliche Managementansätze entwickelt, die in Form unterschiedlicher Methoden etabliert wurden, vgl. bspw. [Kara95, KaNo97, KrSe03]. Im vorliegenden Beitrag werden im Kontext des Architekturmanagements folgende Managementansätze als essenziell betrachtet:

- das *Strategiemanagement* für die Langfristplanung, die IT-Strategiekommunikation und das Ziel- und Performance-Management,
- das *Prozessmanagement* für eine effiziente Gestaltung der Geschäftsprozesse und der dort eingebundenen Ressourcen zur Umsetzung der Unternehmens- und IT-Strategie und
- das *IT-Management* im Sinne eines IT-Servicemanagements und IT-Architekturmanagements für die adäquate informationstechnische Unterstützung der Geschäftsprozesse und der darin arbeitenden Mitarbeiter.

In den folgenden Abschnitten 4.1 bis 4.3 werden drei Methoden für die oben beschriebenen Managementansätze anhand ihres Methodelements "Modellierungstechnik" skizziert. Dabei werden jeweils Anforderungen an die Modellierungstechnik und ein Metamodell für eine entsprechende Modellierungssprache vorgestellt. Auf die Methodelemente "Mechanismen" und "Vorgehensmodell" wird im vorliegenden Beitrag nicht eingegangen. In Abschnitt 4.4 erfolgt die Integration der Modellierungstechniken der drei Methoden.

4.1 Methodendefinition für das Strategiemanagement

Wesentliche Anforderungen an eine Modellierungstechnik für das Strategiemanagement sind:

- Abbildung der Unternehmens- und IT-Strategie und der hierfür wichtigen Erfolgsfaktoren,
- Operationalisierung der Strategie durch Formulierung entsprechender Ziele,
- Beschreibung der Umsetzung der Ziele durch Maßnahmen und Projekte,
- Bewertung der Strategie, der Ziele und deren Umsetzung durch Kennzahlen und
- Definition von Verantwortlichkeiten als Voraussetzung eines adäquaten Strategie-Controllings.

Abb. 12 zeigt das Metamodell einer Modellierungssprache für das Strategiemanagement. Die Modellierungssprache berücksichtigt die oben gestellten Anforderungen.

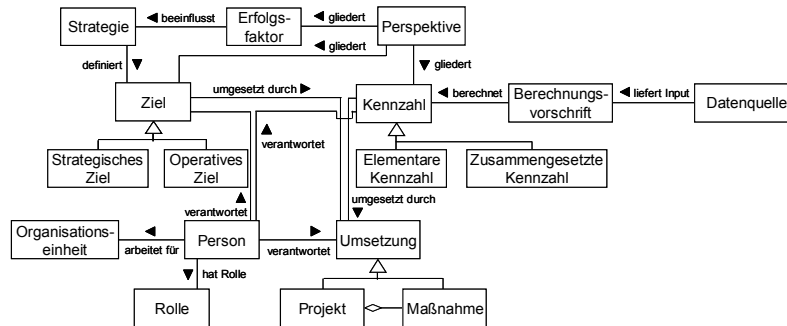


Abbildung 12: Modellierungssprache für Strategiemangement

Die Unternehmensstrategie wird durch von der Unternehmensleitung zu formulierende Erfolgsfaktoren beeinflusst. Hieraus können für das Zielmanagement strategische und operative Ziele abgeleitet werden. Die IT-Strategie wird aus der Unternehmensstrategie abgeleitet. Zur Zielerreichung werden ein oder mehrere Maßnahmen definiert. Die Maßnahmen können zu Projekten gebündelt werden. Für eine quantitative und qualitative Bewertung der Ziele und der damit verbundenen Umsetzungsaspekte werden Kennzahlen eingesetzt. Elementare Kennzahlen stellen atomare Bewertungen dar, zusammengesetzte Kennzahlen werden über Formeln aus elementaren Kennzahlen berechnet. Für beide Arten von Kennzahlen müssen Berechnungsvorschriften definiert werden. Das Datenmaterial für den Input der Berechnungen wird über Datenquellen geliefert. Um eine Kontrolle und Steuerung im Strategiemangement durchführen zu können, werden Verantwortlichkeiten für Ziele, deren Umsetzung und der Bewertung festgelegt. Dies erfolgt durch Zuordnung von Personen zu Zielen, Kennzahlen, Maßnahmen und Projekten.

4.2 Methodendefinition für das Prozessmanagement

Wesentliche Anforderungen an eine Modellierungstechnik für das Prozessmanagement sind:

- Abbildung einer Übersicht der Management-, Kern- und Unterstützungsprozesse eines Unternehmens,
- transparente Definition der Geschäftsprozesse anhand des Kontroll-, Informations- und ggf. Materialflusses,
- Beschreibung der in einem Geschäftsprozess bearbeiteten Artefakte und der hierfür eingesetzten Ressourcen,
- Hinterlegung von statistischen bzw. quantitativen Informationen für eine Modellnutzung im Rahmen quantitativer Auswertungen und

- Definition von Verantwortlichkeiten als Voraussetzung eines adäquaten Prozess-Controllings.

Abb. 13 zeigt das Metamodell einer Modellierungssprache für das Prozessmanagement. Die Modellierungssprache berücksichtigt die oben gestellten Anforderungen.

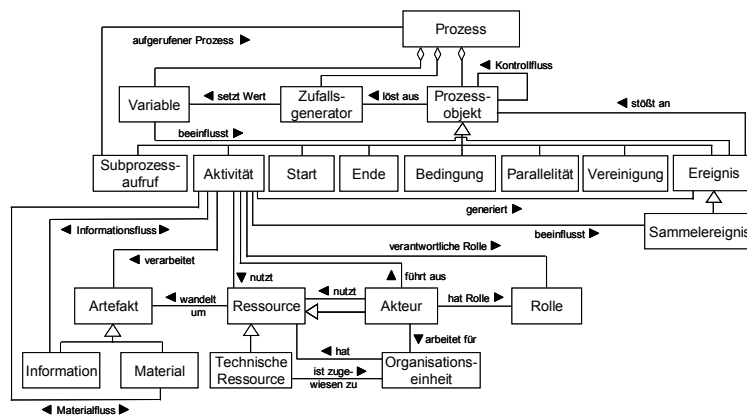


Abbildung 13: Modellierungssprache für Prozessmanagement

Durch den Aufbau von Prozesslandkarten und deren Dekomposition zu Geschäftsprozessmodellen können Übersichten der Prozesse und Prozesskategorien erstellt werden. Das zentrale Element eines Geschäftsprozessmodells bildet die Aktivität. Aktivitäten werden durch Nutzung von Steuerungsobjekten wie "Start", "Ende", "Bedingung", "Parallelität" und "Vereinigung" zu komplexeren Aktivitätsflüssen zusammengefasst. Jede Aktivität wird durch einen Akteur ausgeführt. Dieser nutzt bei der Durchführung von Aktivitäten entsprechende Ressourcen, um die in einem Prozess zu transformierenden Artefakte bearbeiten zu können. Die Akteure können bspw. nach ihren Rollen und zugeordneten Organisationseinheiten strukturiert werden. Die Definition von Zufallsgeneratoren, die zur Zufallszahlengenerierung von den unterschiedlichen in einem Prozess enthaltenen Ablaufobjekte ausgelöst werden können, und die Beschreibung weiterer quantitativer Informationen, liefern die Ausgangsbasis für die quantitative Prozessbewertung.

4.3 Methodendefinition für das IT-Management

Wesentliche Anforderungen an eine Modellierungstechnik für das IT-Management sind:

- Abbildung der IT-Strategie und der damit verbundenen Ziele und Projekte,
- Definition von IT-Servicekatalogen als Basis für die von der IT durchzuführenden Projekte und der zu erbringenden Dienstleistungen,

- Beschreibung der in einem IT-Service gebündelten Elemente wie Serviceprozesse, Anwendungen, Rechner und Netzwerke,
- Bewertung der IT-Services und ihrer Elemente durch Service Level Agreements (SLA) und entsprechender Kennzahlen und
- Definition von Verantwortlichkeiten als Voraussetzung eines adäquaten IT-Controllings.

Abb. 14 zeigt das Metamodell einer Modellierungssprache für das IT-Management. Die Modellierungssprache berücksichtigt die oben gestellten Anforderungen.

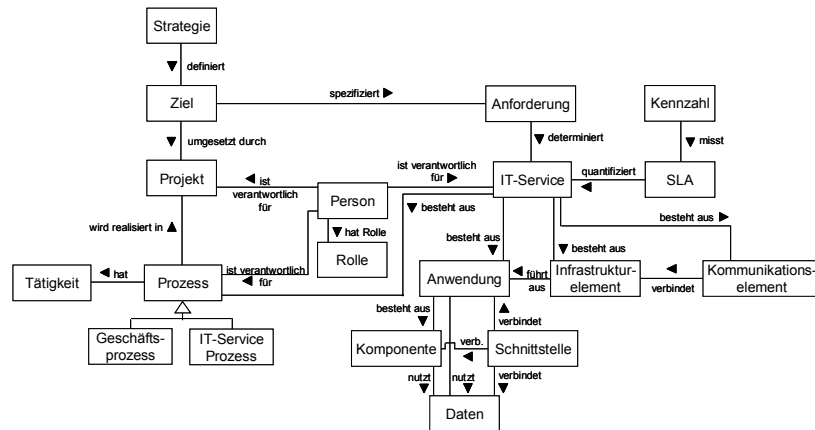


Abbildung 14: Modellierungssprache für IT-Management

Die IT-Strategie bildet die Ausgangsbasis für die Formulierung entsprechender Ziele und der Ableitung notwendiger Projekte, um die Ziele zu erreichen. Weiterhin leiten sich aus den Zielen die Anforderungen an die von der IT zu erbringenden Dienstleistungen (IT-Services) ab. Die Beschreibung eines IT-Service erfolgt durch seine Elemente "Prozess", "Anwendung", "Infrastrukturelement" und "Kommunikationselement". Die Definition der Verantwortung für Projekte, IT-Services und den in den IT-Services enthaltenen Elementen erfolgt durch Zuweisung von Personen. Ein IT-Service wird durch SLAs quantifiziert und diese durch Kennzahlen gemessen. Eine Anwendung besteht aus einer oder mehrerer Komponenten. Anwendungen und Komponenten kommunizieren über Schnittstellen.

4.4 Identifikation der Integrationsstellen

Beim Vergleich der Metamodelle können 6 Integrationsstellen gefunden werden (vgl. Abb. 15). Dabei werden die in Kapitel 3 vorgestellten Integrations-Patterns

eingesetzt. Die folgende Nummerierung korrespondiert zur Nummerierung in Abb. 15.

1. *Strategie-Ziel-Projekt*: Diese Modellierungskonstrukte finden sich sowohl in den Metamodellen zum Strategie- als auch IT-Management wieder. Durch Nutzung des Fragment-Patterns (vgl. Abschnitt 3.2) werden die Konstrukte aus den Metamodellen extrahiert und in ein eigenes Methodenfragment abgebildet. Die Integration erfolgt dann durch die gemeinsame Nutzung dieses Methodenfragments (vgl. Abb. 16).
2. *Aktivität-Tätigkeit*: Im Metamodell zum Prozessmanagement bildet die Aktivität das zentrale Element. Im Metamodell zum IT-Management werden Prozesse durch ihre Tätigkeiten beschrieben. Die beiden korrespondierenden Konstrukte "Aktivität" und "Tätigkeit" werden durch das Concept-Mapping-Pattern (vgl. Abschnitt 3.3) miteinander verbunden und aufeinander abgebildet (vgl. Abb. 16).
3. *Prozess-Prozess*: Sowohl im Metamodell zum Prozessmanagement als auch im Metamodell zum IT-Management steht das Modellierungskonzept "Prozess" zur Verfügung. Durch Nutzung des Common-Baseclass-Pattern (vgl. Abschnitt 3.1.3) können die beiden Modellierungsklassen von einer gemeinsamen Basisklasse abgeleitet und hierdurch eine Integration der Metamodelle erreicht werden.
4. *Person-Rolle-Organisationseinheit*: In allen drei Metamodellen finden sich Modellierungskonstrukte zur Organisationsbeschreibung, bspw. "Rolle", "Person", "Akteur" und "Organisationseinheit". Die Konstrukte zur Organisationsmodellierung werden unter Nutzung des Fragment-Pattern (vgl. Abschnitt 3.2) aus den Metamodellen extrahiert und in ein eigenes Methodenfragment abgebildet. Die Integration erfolgt dann durch die gemeinsame Nutzung des Methodenfragments.
5. *Information-Daten-Datenquelle*: In den drei Metamodellen stehen die äquivalenten Modellierungskonstrukte "Information", "Daten" und "Datenquelle" zur Verfügung. Diese werden mittels Concept-Mapping-Pattern (vgl. Abschnitt 3.3) einander zugeordnet.
6. *Kennzahl-Kennzahl*: Sowohl im Metamodell zum Strategiemangement als auch im Metamodell zum IT-Management steht das Konstrukt "Kennzahl" zur Verfügung. Die Integration erfolgt durch Vereinigung dieses Konstrukts aus beiden Metamodellen (vgl. Abb. 16). Hierzu findet das Merge-Pattern (vgl. Abschnitt 3.4) Anwendung.

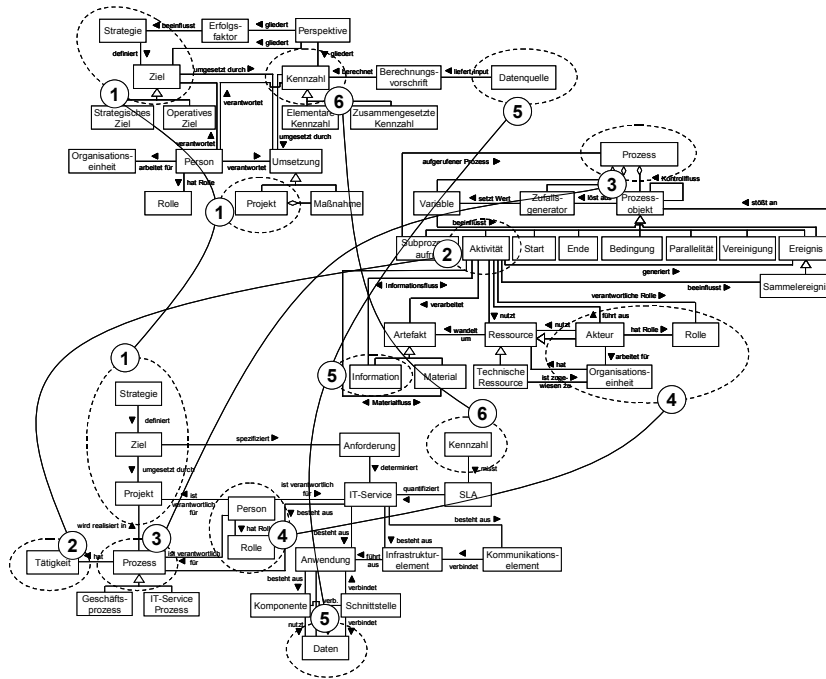


Abbildung 15: Metamodellbasierte Identifikation der Integrationsstellen

4.5 Integration der Methoden

Basierend auf den in Abb. 15 identifizierten Integrationsstellen, wird die Integration der Metamodelle der Modellierungssprachen durchgeführt. Das Ergebnis der Integration zeigt Abb. 16. Aus Gründen der Übersichtlichkeit werden nur drei der sechs durchgeführten Integrationen gezeigt. Konkret sind dies:

- Nr. 1 - Integration mittels Fragment-Pattern,
- Nr. 2 - Integration mittels Concept-Mapping-Pattern und
- Nr. 6 - Integration mittels Merge-Pattern.

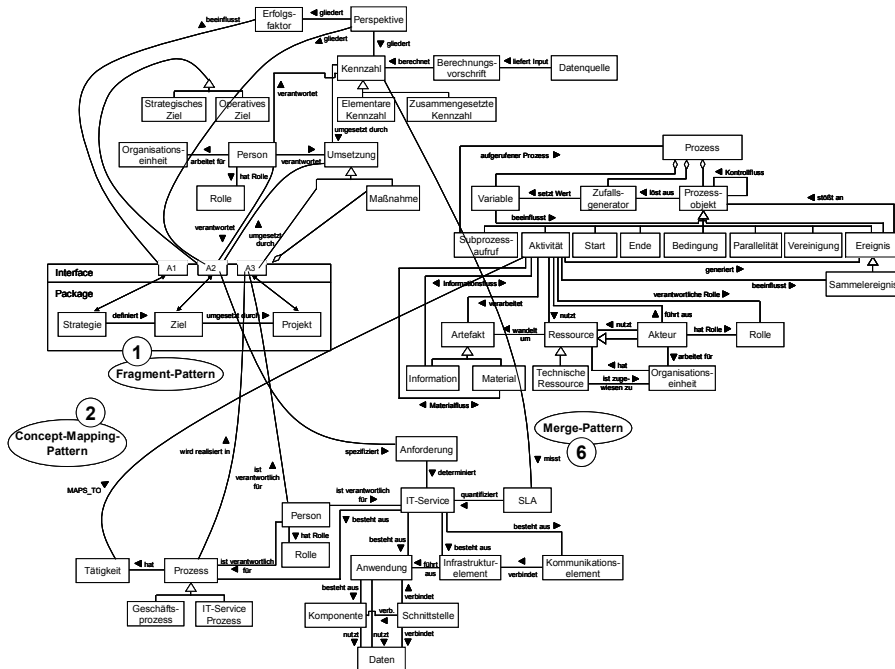


Abbildung 16: Integration der Metamodelle der Modellierungssprachen

5 Zusammenfassung und Ausblick

Dem vorliegenden Beitrag liegt die Überzeugung zugrunde, dass es nicht *die* Methode in der Unternehmensmodellierung gibt und auch nicht geben wird. Vielmehr wird ein unternehmensspezifisches Bündel von situations- und problemspezifischen Methoden und Methodenfragmenten benötigt, die durch Kombination und Integration die geforderte Hilfestellung für den Anwender liefern. Betrachtet man die gegenwärtige Situation und die in den Medien kommunizierten strategischen Akzente der "Global Player" in der Software- und IT-Beratungsbranche, so lässt sich feststellen, dass das Thema "Modellierung" und die Beherrschung modell- und methodengestützter Ansätze mittlerweile als ein Wettbewerbsvorteil gesehen wird und nicht mehr nur als ein "Nice-to-Have". Alle "Großen" haben bereits oder haben zumindest angekündigt, modell-basierte Komponenten in ihre Umgebungen zur Entwicklung und Integration von Geschäftsanwendungen zu nutzen, bspw. IBM's WebSphere, Microsoft's BizTalk Server [BOC04] oder SAP's NetWeaver.

Abschließend werden drei den Autoren zentral erscheinende Tendenzen im Kontext der Definition, Integration und Nutzung von Methoden skizziert.

Enterprise Model Integration: Auf der Ausführungs- und Anwendungsebene haben sich Lösungsansätze im Rahmen des EAI etabliert [Joha⁺00]. Integrationsprojekte müssen aber auch auf einer fachlich-konzeptionellen Ebene begleitet werden. Ein sich abzeichnender Trend ist hier zum einen die verstärkte Standardisierung von Modellierungssprachen für bestimmte Einsatzbereiche [OMG01, BPMI04]. Zum anderen werden Ansätze für die Integration (der Metamodelle) proprietärer Modellierungssprachen benötigt, da bisher der Großteil der Hersteller von Business-Software eigene Modellierungssprachen einsetzt.

Metamodellierung wird Basistechnologie: Im vorliegenden Beitrag wurde die Metamodellierung aufgrund der damit verbundenen Flexibilität als konzeptionelle Basis gewählt. Die Metamodellierung hält als Basistechnologie jedoch mittlerweile auch in verschiedenen anderen Bereichen Einzug, bspw. durch die Meta Object Facility [OMG02], in der Sprachdefinition der UML Version 2.0 oder im Software Engineering-Ansatz der Produktfamilien. Zukünftig wird eine verstärkte Nutzung von Metamodellierung in weiteren Anwendungsgebieten erwartet, bspw. im Workflow-Management [Kara95], im IT-Architekturmanagement und im Wissensmanagement [Kral⁺02].

Integration Business-orientierter und IT-orientierter Methoden: In Kapitel 4 wurde ein Fallbeispiel zur Integration der Managementansätze "Strategiemanagement", "Prozessmanagement" und "IT-Management" vorgestellt. In diesem Kontext wird direkt aus der Praxis kommender, ergänzender Forschungsbedarf gesehen. Ein vielversprechender Ansatz, der sich jedoch noch in einem relativ frühen Entwicklungsstadium befindet, ist die Model Driven Architecture (MDA). Die MDA erhebt den Anspruch, die Fachwelt mit der IT-Welt über Modelltransformationen adäquat zu verbinden. Hier wird Forschungsbedarf v.a. im Kontext verlustfreier Transformationen und der Behandlung und Integration unterschiedlicher Semantiken erwartet.

Literatur

- [AiSc04] Aier, S.; Schönherr, M. (Hrsg.): Enterprise Application Integration – Flexibilisierung komplexer Unternehmensarchitekturen. GITO-Verlag, 2004.
- [BOC04] BOC GmbH: ADONIS-BizTalk Server 2004-Integration, <http://www.boc.eu.com/html/biztalk.html>, Abruf am 2004-10-14
- [Bole⁺04] Boles, C.; Friebe, J.; Luhmann, T.: Typische Integrationsszenarien und deren Unterstützung durch Web Services und andere Technologien. In: [HaRe04], S. 57-67.
- [BPMI04] BPMI.org: Spezifikationen der BPML und BPMN - jeweils Version 1.0. <http://www.bpmi.org>, Abruf am 2004-10-14.
- [ErPe00] Eriksson, H.-E.; Penker, M.: Business Modeling with UML - Business Patterns at Work. John Wiley & Sons, 2000.

- [Gamm⁺94] Gamma, E.; Helm, R.; Johnson, G.; Vlissides, J.: Design Patterns – Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994.
- [HaRe04] Hasselbring, W.; Reichert, M. (Hrsg.): EAI 2004 – Enterprise Application Integration. Tagungsband des GI/GMDS-Workshops EAI'04, OFFIS, Oldenburg, Februar 2004.
- [Joha⁺00] Johannesson, P.; Wangler, B.; Jayaweera, P.: Application and Process Integration – Concepts, Issues, and Research Directions. In: Brinkkemper, S.; Lindencrona, E.; Solvberg, A. (Eds.): Information Systems Engineering Symposium CAiSE 2000, Springer-Verlag, 2000.
- [KaNo97] Kaplan, R. S.; Norton, D. P.: Die Balanced Scorecard: Strategien Erfolgreich Umsetzen. Verlag Schäffer-Poeschel, Handelsblatt-Reihe, Stuttgart, 1997.
- [Kara95] Karagiannis, D.: BPMS: Business Process Management Systems. In: ACM SIGOIS Bulletin, Vol. 16, Nr. 1, August 1995, S. 10-13.
- [Kral⁺02] Krallmann, H.; Schönherr, M.; Aier, S.; Dietrich, J.: Transformation einer industriell geprägten Unternehmensstruktur zu einer service- und wissensbasierten Organisation. 2002.
- [KrSe03] Krüger, S.; Seelmann-Eggelbert, J.: IT-Architektur-Engineering. Galileo Press GmbH, Bonn, 2003.
- [Kühn⁺03] Kühn, H.; Bayer, F.; Junginger, S.; Karagiannis, D.: Enterprise Model Integration. In: Bauknecht, K.; Tjoa, A. M.; Quirchmayr, G. (Hrsg.): Proceedings of the 4th International Conference EC-Web 2003 - Dexa 2003, Prag, Tschechien, September 2003, LNCS 2738, Springer-Verlag, S. 379-392.
- [Kühn04] Kühn, H.: Methodenintegration im Business Engineering. Dissertation, Universität Wien, April 2004.
- [Kühn⁺04] Kühn, H.; Murzek, M.; Bayer, F.: Horizontal Business Process Model Interoperability using Model Transformation. In: Workshop on Interoperability of Enterprise Systems (INTEREST'2004) at ECOOP2004, Oslo, Norwegen, Juni 2004.
- [Lint00] Linticum, D. S.: Enterprise Application Integration. Addison-Wesley, 2000.
- [OMG 01] Object Management Group: OMG Unified Modeling Language Specification, Version 1.4, September 2001. <http://www.omg.org/cgi-bin/doc?formal/01-09-67.pdf>, Abruf am 2004-10-14.
- [OMG02] Object Management Group: Meta Object Facility (MOF) Specification, Version 1.4, April 2002. <http://www.omg.org/cgi-bin/doc?formal/02-04-03.pdf>, Abruf am 2004-10-14.
- [OMG03] Object Management Group: MDA Guide, Version 1.0.1, 12. Juni 2003. <http://www.omg.org/cgi-bin/apps/doc?omg/03-06-01.pdf>, Abruf am 2004-10-14.