

2 Integrationsarchetypen – Herleitung und Begründung

2.1 Bottom-up-Vorgehen vs. Top-down-Vorgehen der Methodenkonstruktion

Das Thema dieses Buches ist die systematische Methodenunterstützung für Integrationsprojekte. Dafür wurden in Kapitel 1 zunächst die Grundlagen aufbereitet. Darauf aufbauend werden in diesem und dem folgenden Kapitel Grundzüge der Methodik sowie wichtige Methodik-Komponenten vorgestellt. Die Integrationsmanagement-Methodik hat das Ziel, die Planung und die Umsetzung von Integrationsprojekten u. a. durch

- Vorgabe von Aktivitäten und Abläufen,
- Vorgabe von Dokumentstrukturen und Dokumentabhängigkeiten sowie
- Vorgabe von Rollen, Verantwortlichkeiten und Steuerungsmechanismen zu unterstützen.

Die Integrationsmanagement-Methodik umfasst dabei Referenzmodelle, Checklisten, Aktivitätsbeschreibungen usw. und hilft, Projektergebnisse zielgerichtet, konsistent, arbeitsteilig und skalierbar zu erarbeiten, zu dokumentieren und zu kommunizieren.

Integrationsprojekte sind extrem vielgestaltig. Eine systematische Methodenunterstützung müsste an den verschiedenen Aspekten des Integrationsphänomens ansetzen, um problemgetriebene Lösungsbausteine zu entwickeln. So wäre es z. B. denkbar, je nach Reichweite der Integration (bereichsintern, bereichsübergreifend, unternehmensübergreifend) unterschiedliche Governance-Strukturen vorzugeben. Ein solches Vorgehen ist aber wenig realistisch, da die Diskussion in Kapitel 1 zeigt, dass eine konsistente und vollständige Erfassung des Phänomens „Integration“ kaum möglich ist. Deshalb wird im Folgenden ein alternativer Weg beschritten: Methodenunterstützung wird nicht ausgehend von Problemspezifika in Richtung Methodenbausteine entwickelt, sondern ausgehend von elementaren Problemlösungskonzepten in Richtung der zu bearbeitenden Gestaltungsprobleme.

Abbildung 2.1 illustriert dieses Vorgehen anhand einer Analogie zwischen Chemie und Integration. Unterstellt wird für die Chemie das Problem, einen Stoff zu entwickeln, der mittels erwünschter Eigenschaften ein bestimmtes Anwendungsproblem löst (z. B. leitfähig bei hohen Temperaturen und gleichzeitig unempfindlich gegen Oxidation). Wenn alle existierenden und denkbaren Stoffe sowie alle Eigenschaften dieser Stoffe bekannt wären, könnte man das Anwendungsproblem durch Identifikation grundsätzlich geeigneter Stoffe und allenfalls deren zielgerichteter Modifikation lösen. Die Vielfalt existierender und denkbarer Stoffe verunmöglicht aber meist einen solchen „Top-down“-Ansatz. Da jedoch die

Systematik chemischer Reaktionen genauso bekannt ist wie die Systematik der Stoffe und sogar deren Strukturierung aus Elementarteilchen, besteht die Möglichkeit, die gesuchte Problemlösung auch systematisch „Bottom-up“ zu konstruieren. Dazu sind schrittweise die grundsätzlich geeigneten Elemente zu identifizieren und danach die daraus erzeugbaren Stoffe hinsichtlich ihrer „groben“ Passung und schließlich hinsichtlich ihres konkreten Problemlösungsbeitrags zu bewerten.

Bei dem Vorgehen der Weise „von den Lösungsbausteinen zum Problem“ ist stets zu berücksichtigen, dass die Entwicklung der Methodenbausteine ausgehend von den „Elementarteilchen“, also den Archetypen der Integration stattfindet (Bottom-Up), die Anwendung dieser Bausteine jedoch immer ausgehend von der Problemlösung, also, Top-Down erfolgt. Aus diesem Grund sind verbindende Strukturen in beide Richtungen unerlässlich.

Die Chemie-Analogie lässt sich – wenn auch natürlich nur sehr frei – auf Integrationsprojekte übertragen. Für das Integrationsmanagement besteht das Problem darin, ein Vorgehen festzulegen, das mittels erwünschter Ergebnisse ein bestimmtes Integrationsproblem zielgerecht löst (z. B. Ablösung zweier unterschiedlicher CRM-Applikationen durch ein einziges CRM-System ohne signifikante Änderung der organisatorischen Strukturen). Wenn alle existierenden und denkbaren Integrationsprojekte sowie alle Ergebnisse solcher Projekte bekannt wären, könnte man das Integrationsproblem durch Identifikation grundsätzlich „passender“ Projekte und allenfalls deren zielgerichteter Modifikation lösen – der sog. „Best Practice“-Ansatz. Die Vielfalt existierender und denkbarer Integrationsprobleme sowie die oft fehlende terminologische und methodische Vergleichbarkeit verunmöglichen aber meist einen solchen „Top-down“-Ansatz. Außerdem ist nie sicher, ob nur zufällig eine passende Lösung gefunden wurde und wie nahe diese Lösung dem Optimum kommt. Wenn jedoch auf Grundlage einer einheitlichen Terminologie die Grob-Systematik von Integrationssituationen genauso bekannt wäre wie ein System von Integrationsmustern und vielleicht sogar deren Strukturierung aus „Elementarbestandteilen“, bestünde die Möglichkeit, die gesuchte Problemlösung zielgerichtet „Bottom-up“ zu konstruieren. Dazu wären schrittweise zunächst die grundsätzlich geeigneten Integrations-„Elementarbestandteile“ zu identifizieren (falls diese hinsichtlich des Integrationsziels unterschiedliche Beiträge leisten), danach die daraus ableitbaren Aktivitäten hinsichtlich ihrer „groben“ Passung zu bewerten und auszuwählen und zu Methodenfragmenten zusammenzustellen; im letzten Schritt wären diese Methodenfragmente dann situationsbezogen anzupassen, um die konzipierte(n) Lösung(en) schließlich hinsichtlich ihres Problemlösungsbeitrags bewerten zu können.

Für einen systematischen, zielgerichteten Aufbau methodischer Unterstützung für Integrationsprojekte fehlen jedoch wichtige Komponenten. Was sind die „Elementarbestandteile“ von Integration? Wie können sie verbunden und kombiniert werden? Welche Integrations-Situationen sollten unterschieden werden und in welcher Situation sollten welche Muster zum Einsatz kommen? Der Rest des Kapitels 2 sowie Kapitel 3 versuchen auf diese Fragen Antworten zu entwickeln.

Tabelle 2.1. Problemverständnisebenen am Beispiel von Chemie und Integration

	Chemie	Integration
Ebene der Gestaltungsprobleme	Konkreter (Wirk-)Stoff	Konkretes Integrationsprojekt
Ebene der Anwendungsbereiche	Systematik von Reaktionen	Systematik der Integrationssituationen
Ebene der Elemente	Systematik der Elemente	Integrationsmuster
Ebene der Elementarteilchen	Protonen, Elektronen etc.	Integrationsarchetypen

2.2 Situative Unterstützung von Integrationsprojekten

Zunächst wird die „Ebene der Anwendungsbereiche“ in Tabelle 2.1 betrachtet. Ein Schlüssel zur Erarbeitung der Methodenunterstützung ist die Schaffung eines sinnvollen Abstraktionsgrades beim Verständnis von Integrationsprojekten. Es ist nicht sinnvoll, eine generische Methodenunterstützung zu erarbeiten, die den Anspruch erhebt, für alle Arten von Integrationsprojekten effektiv zu sein („one size fits all“): Integrationsprojekte sind im Spannungsfeld von Integrationsstrategie bis zur IT-Umsetzung, von Workflow-Integration bis zur unternehmensübergreifenden Integration, von operativer Datenintegration im Prozess bis zur Kennzahlenkonsolidierung im Konzern so vielgestaltig, dass eine Vorgabe einheitlicher, standardisierter Aktivitäten, Adressaten und Abhängigkeiten über alle Integrationsprojekte hinweg zu abstrakt bleiben müsste und deshalb wenig effektiv sein wird. Aus diesem Grund muss die Methodenunterstützung situativ sein.

2.2.1 Generizitätsgrad

In Analogie zu situativen Ansätzen der Entscheidungstheorie (Fiedler 1964; Simon 1976; Simon 1977) heißt situative Methodenunterstützung, dass aus einer Grundmenge standardisierter, feingranularer Methodenbausteine nur solche Bausteine situationsspezifisch kombiniert werden, die für die Art des jeweiligen Integrationsprojekts und die Art des jeweiligen Integrationskontexts geeignet sind. Bei der Konzeption situativer Methodenunterstützung muss ein wichtiger Generizitäts-Trade-off gelöst werden:

1. Einerseits sollen Methodenbausteine möglichst generisch sein, um oft wiederverwendet werden zu können. Nur dann kann die Menge der notwendigen Methodenbausteine wirtschaftlich entwickelt, verwaltet und hinsichtlich ihrer Problemlösungseignung analysiert werden.
2. Andererseits sollen Methodenbausteine möglichst spezifisch sein, um den jeweiligen Projektzielen und dem jeweiligen Kontext gerecht zu werden.

Abb. 2.2 illustriert den Trade-off zwischen der Effizienz der Methodenkonstruktion und der Effektivität der Methodenanwendung. Die vertikale Dimension repräsentiert die Bandbreite der Generalizität zwischen den Extrema „Problembeschreibung/Methodenbaustein ist komplett projektspezifisch“ (unten) und „Problembeschreibung/Methodenbaustein ist komplett projektunspezifisch“ (oben). Die horizontale Dimension repräsentiert die Bandbreite der unterschiedlichen Integrationsprojekte. Das Dendrogramm visualisiert den Zusammenhang zwischen Problembeschreibungen/Methodenbausteinen auf verschiedenen Generalizitätsgraden. Aus einem großen Spektrum unterschiedlicher, projektspezifischer Problembeschreibungen/Methodenbausteine wird bei sukzessiver Generalisierung eine immer geringere Zahl immer unspezifischerer Problembeschreibungen/Methodenbausteine.

Für die Methodenunterstützung des Integrationsmanagements liegt die Lösung des Tradeoffs tendenziell in der Mitte: Einerseits muss soweit von Problemspezifika abstrahiert werden, dass die Vielfalt der „typischen“ Problembeschreibungen – und damit der Methodenbausteine – stark reduziert wird und handhabbar bleibt. Andererseits darf nicht noch weiter abstrahiert werden, da sonst die Methodenbausteine zu generisch werden und die jeweiligen Integrationsprobleme damit nicht mehr effektiv gelöst werden können. In Abb. 2.1 repräsentiert die unterbrochene Linie eine solche Kompromisslösung: Für den ausgewählten Generalizitätsgrad ergeben sich sechs Situationen. Diese reduzieren die ursprünglich 37 unterschiedlichen Problembeschreibungen erheblich, führen aber zu deutlich höherem Differenzierungspotenzial der Methodenunterstützung als die Verdichtung zu einer einzigen „Durchschnitts“-Problemklasse bei maximaler Generalizität.

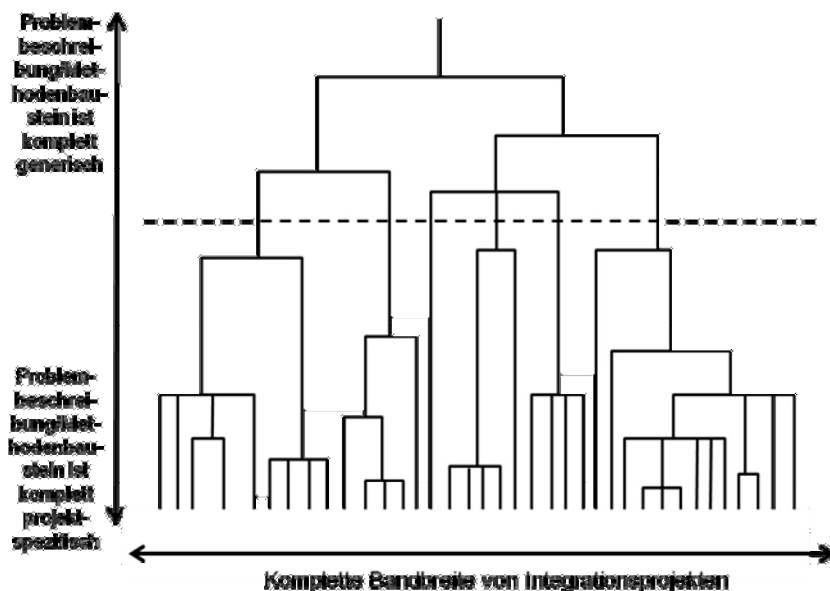


Abb. 2.1. Visualisierung des Tradeoffs zwischen Effizienz der Methodenkonstruktion und Effektivität der Methodenanwendung

2.2.2 Situationsmatrix

Die wichtigste Grundlage für die situative Methodenunterstützung ist zunächst die Typisierung von Integrationsprojekten zu Projekttypen und von Integrationskontexten zu Kontexttypen. Während durch Projekttypen abgebildet wird, welche Zielsetzungen mit dem jeweiligen Integrationsprojekt verfolgt werden (d. h. welche Systemparameter verändert werden), bilden Kontexttypen ab, in welcher Umgebung das jeweilige Integrationsprojekt durchgeführt wird (d. h. welche Systemparameter unverändert bleiben, aber Einfluss auf das Projekt haben) (Bucher et al. 2007). Typische Projekttypen im Integrationsmanagement sind Migrationen, Software-Innovationen oder die Umsetzung organisatorischer Veränderungen (vgl. Kapitel 3). Typische Kontexttypen im Integrationsmanagement sind die Branche oder die Organisationsgröße. Um die für die Konstruktion von Methodenbausteinen relevanten Projekt- und Kontexttypen zu identifizieren, helfen empirische Untersuchungen. Mit statistischen Methoden wie der Faktor- und Clusteranalyse kann ermittelt werden, welches die relevanten Einflussfaktoren sind und in welchen Kombinationen bestimmte Faktorausprägungen besonders häufig zusammen auftreten.

Wenn eine sinnvolle Anzahl an Projekt- und Kontexttypen identifiziert wurde, ist zu untersuchen, welche Kombinationen als Situationstypen relevant sind. Nicht jede Kombination von Projekt- und Kontexttyp existiert. Manchmal ist über mehrere Kontexttypen hinweg der Projekttyp dominant, währenddessen der Kontexttyp ebenso mehrere Projekttypen dominieren kann. In vielen Fällen ist eine bestimmte Kombination von Kontext und Projekt so spezifisch, dass dafür eine Situation definiert werden sollte. Abb. 2.2 illustriert eine solche Situationsmatrix.

	Projekttyp A	Projekttyp B	Projekttyp C	Projekttyp ...
Kontexttyp A	Situation 1	Situation 2		Situation ...
Kontexttyp B		Situation 3		Situation ...
Kontexttyp C		Situation 4		
Kontexttyp ...			Situation ...	

Abb. 2.2. Situationsmatrix für situative Methodenunterstützung (Bucher et al. 2007)

Für die Planung und Durchführung von Integrationsprojekten muss eine solche Integrationsmatrix erst noch entwickelt werden. In Abb. 2.3 ist jedoch zur Verdeutlichung eine konkrete Situationsmatrix für die prozessorientierte Informationslogistikgestaltung dargestellt. Prozessorientierte Informationslogistik zielt auf die Unterstützung und Verbesserung der Prozessausführung durch die Einbettung von analytischen Informationen oder Analysemechanismen in diese Prozessausführung (Bucher u. Dinter 2008b). Aus der Kombination von je vier Kontexttypen und Projekttypen ergeben sich theoretisch 16 unterschiedliche Entwicklungssitua-

tionen der prozessorientierten Informationslogistik. Aufgrund sachlogischer Überlegungen wurden vier dieser Situationskandidaten ausgeschlossen; ihr Auftreten in der Praxis wird als vergleichsweise unwahrscheinlich erachtet. Für die verbleibenden zwölf Entwicklungssituationen werden jeweils spezifische Herausforderungen formuliert, die sich an eine situationsspezifische Methode zur Einführung der prozessorientierten Informationslogistik stellen. In der beispielhaft gezeigten Situationsmatrix wurde auf die – Projekttypen- und/oder Kontexttypen-übergreifende – Zusammenfassung mehrerer Entwicklungssituationen verzichtet; eine Verdichtung wäre jedoch bspw. durch die Zusammenfassung der Projekttypen 2 und 3 möglich.

PTs of Project Types (Common PTs)		Low Excellence in Integration of Analytic Information and Process Execution		High Excellence in Integration of Analytic Information and Process Execution	
Context Types (CTs = CTs)		Low Excellence in Information Supply	High Excellence in Information Supply	Low Excellence in Information Supply	High Excellence in Information Supply
Utilization of Basic Instruments for Information Access and Analysis	Structured Processes	Development Situation 1 Main Challenges: ✓ Improve information supply ✓ Make necessary arrangements for the integration of information and process execution ✓ Employ basic instruments for information access and analysis ✓ (Re-)design structured process	Development Situation 3 Main Challenges: ✓ Maintain high excellence in information supply ✓ Make necessary arrangements for the integration of information and process execution ✓ Employ basic instruments for information access and analysis ✓ (Re-)design structured process	Development Situation 7 Main Challenges: ✓ Improve information supply ✓ Maintain high excellence in integration of information and process execution ✓ Employ basic instruments for information access and analysis ✓ (Re-)design structured process	Context Type 1
	Unstructured Processes		Development Situation 4 Main Challenges: ✓ Maintain high excellence in information supply ✓ Make necessary arrangements for the integration of information and process execution ✓ Employ basic instruments for information access and analysis ✓ (Re-)design unstructured process	Development Situation 8 Main Challenges: ✓ Improve information supply ✓ Maintain high excellence in integration of information and process execution ✓ Employ basic instruments for information access and analysis ✓ (Re-)design unstructured process	
Utilization of Advanced Instruments for Information Access and Analysis	Structured Processes	Development Situation 2 Main Challenges: ✓ Improve information supply ✓ Make necessary arrangements for the integration of information and process execution ✓ Employ advanced instruments for information access and analysis ✓ (Re-)design structured process	Development Situation 5 Main Challenges: ✓ Maintain high excellence in information supply ✓ Make necessary arrangements for the integration of information and process execution ✓ Employ advanced instruments for information access and analysis ✓ (Re-)design structured process	Development Situation 9 Main Challenges: ✓ Improve information supply ✓ Maintain high excellence in integration of information and process execution ✓ Employ advanced instruments for information access and analysis ✓ (Re-)design structured process	Context Type 3
	Unstructured Processes		Development Situation 6 Main Challenges: ✓ Maintain high excellence in information supply ✓ Make necessary arrangements for the integration of information and process execution ✓ Employ advanced instruments for information access and analysis ✓ (Re-)design unstructured process	Development Situation 10 Main Challenges: ✓ Improve information supply ✓ Maintain high excellence in integration of information and process execution ✓ Employ advanced instruments for information access and analysis ✓ (Re-)design unstructured process	
		Project Type 1 (Low Maturity of Process-oriented IL)		Project Type 2 (Medium Maturity of Process-oriented IL)	
				Project Type 3 (Medium Maturity of Process-oriented IL)	
				Project Type 4 (High Maturity of Process-oriented IL)	

Abb. 2.3. Situationsmatrix für prozessorientierte Informationslogistikgestaltung (Bucher u. Dinter 2008a)

Wenn die für den Problembereich relevanten Situationen identifiziert sind, kann die entsprechende situationsspezifische Methodenunterstützung erarbeitet werden. Bisher wurden die anzuwendenden Methodenkomponenten generisch als „Bausteine“ bezeichnet. Eine geeignetere Bezeichnung ist Methodenfragment –

„Fragment“ bringt zum Ausdruck, dass es sich um „kleine“ Methoden mit Rollen-, Aktivitäts-, Vorgehens-, Dokumentations- und Informationsmodell handeln kann, aber auch um Methodenteile wie z. B. eine Analysetechnik, eine Modellierungsvorschrift, ein Dokumentationstemplate oder ein Referenzmodell.

2.2.3 Ablauf der situativen Methodenunterstützung für Integrationsprojekte

Methodenfragmente werden situationsspezifisch kombiniert. In einer bestimmten Situation entsteht die Methodenunterstützung aus den jeweils projekttypspezifischen, kontexttypspezifischen, also den situationsspezifischen, sowie den unspezifischen (d. h. für alle Integrationsprojekte zweckmäßigen) Methodenfragmenten.

So können bspw. die Erarbeitung eines Architekturmodells der zu integrierenden Artefakte, die Modellanalyse, die Ideenfindung für Integrationsalternativen, die Evaluation dieser Integrationsalternativen, gegebenenfalls die prototypische Umsetzung einer oder mehrerer Alternativen und deren Überführung in den operativen Betrieb solche Methodenfragmente darstellen. Für bestimmte Situationen kann sich die Methodenunterstützung durch Kombination (und falls erforderlich Adaption) dieser Methodenfragmente ergeben. Für sehr unterschiedliche Integrationssituationen ist es jedoch auch möglich, dass vollkommen andere Methodenfragmente eingesetzt werden müssen.

Unter Nutzung situativer Methodenunterstützung gestalten sich die Hauptschritte des Integrationsmanagements wie folgt:

1. Zunächst ist das Integrationsproblem zu spezifizieren, d. h. seine Rahmenbedingungen, seine Ziele und die betroffenen fachlichen und IT-Artefakte sind festzulegen.
2. Aufgrund dieser Spezifikation ist es möglich, das betrachtete Integrationsproblem einer Situation zuzuordnen. Dafür müssen lediglich der zum festgestellten Kontext passende Kontexttyp und der zu den definierten Projektzielen passende Projekttyp herausgefunden werden. Danach ist die Situation bekannt, der das betrachtete Integrationsprojekt entspricht.
3. Im nächsten Schritt sind die verfügbaren Integrationsmanagement-Methodenfragmente für die betreffende Situation anzupassen und zu einer Gesamtmethode zu integrieren. Bei der Entwicklung der Methode sollten Anpassungsmechanismen „eingebaut“ worden sein, die diesen Schritt unterstützen.
4. Die damit situativ angepasste Methode wird im Integrationsprojekt angewandt und unterstützt die Erzeugung konkreter Projektergebnisse.

2.3 Archetypen als Elementarteile

Die Konstruktion problemadäquater, situativer Methoden für die Planung und Umsetzung von Integrationsprojekten erfordert eine grundlegende Kenntnis über die Struktur der zu bewältigenden Integrationsaufgaben und der diesen zugrundeliegenden Elementarteilchen, den Archetypen der Integration.

Da eine phänomenologische Betrachtung von Integration immer bereits relevante Kontextfaktoren, wie etwa die Unternehmensgröße (bei Middleware- und EAI-Diskussionen) oder die Unternehmenskultur (bei zwischenbetrieblicher Integration) beinhaltet, können Integrationsarchetypen auf diese Weise nicht sicher bestimmt werden. Darum wird ein deutlich abstrakterer Weg gewählt, indem mögliche Integrationsoperationen auf einem Metamodell der Unternehmensarchitektur analysiert werden. Die Unternehmensarchitektur beschreibt die fundamentale Strukturierung eines Unternehmens und kann beispielsweise in die fünf Ebenen Strategiebene, Organisationsebene, Integrationsebene, Software- und Datenebene sowie IT-Infrastrukturebene mit ihren jeweiligen Teilarchitekturen differenziert werden (Winter u. Fischer 2007). Ein Metamodell der Unternehmensarchitektur beschreibt deren Artefakttypen und die Beziehungen der Artefakttypen untereinander.

Integrationsoperationen auf einem solchen Metamodell können Artefakte des gleichen Artefakttyps oder unterschiedlicher Artefakttypen verbinden oder vereinigen (Rosemann 1999). Die Vereinigung geht dabei immer mit einer Reduktion der Anzahl an Artefakttypen einher und wirkt dadurch komplexitätsreduzierend. Jedoch ist Vereinigung häufig nicht möglich oder nicht sinnvoll. Wenn z. B. Informationssysteme in unterschiedlichen Unternehmen einen unternehmensübergreifenden Geschäftsprozess unterstützen, ist vereinigende Integration aufgrund unterschiedlicher Verantwortlichkeiten und/oder rechtlicher Rahmenbedingungen nicht möglich. Wenn z. B. unterschiedliche Artefakttypen wie fachliche Anforderungen auf der einen Seite und durch IT realisierte Funktionalitäten auf der anderen Seite abgestimmt werden müssen, ist vereinigende Integration ebenfalls nicht möglich. Selbst wenn vereinigende Integration möglich ist, ist sie nicht immer sinnvoll, um z. B. das Entstehen monolithischer, schwer wartbarer Informationssysteme zu vermeiden.

Wenn eine vereinigende Integration nicht möglich oder sinnvoll ist, muss Integration durch Verknüpfung erfolgen. Bei der Integration durch Verknüpfung kann die Verknüpfung von Artefakten gleicher Typen und die Verknüpfung von Artefakten unterschiedlicher Typen differenziert werden. Klassische Beispiele für die Verknüpfung von Artefakten gleicher Typen sind die Integration von Teilprozessen oder Schnittstellen zur Integration von Softwaresystemen.

Werden Artefakte unterschiedlicher Typen verknüpft, kann dies aus zeitlicher Perspektive einen sequentiellen oder einen parallelen Charakter haben. Bei der sequentiellen Verknüpfung werden die Artefakte eines Typs aus den Artefakten eines anderen Typs sequentiell abgeleitet, z. B. die Ablauforganisation aus der Aufbauorganisation. Bei der parallelen Verknüpfung müssen bereits bestehende Arte-

fakte, welche sich unabhängig voneinander entwickeln, aneinander ausgerichtet werden, z. B. sich parallel entwickelnde fachliche und IT-Strukturen.

Aufgrund dieser Überlegungen können die in Abb. 2.4 dargestellten Integrationsarchetypen differenziert werden.

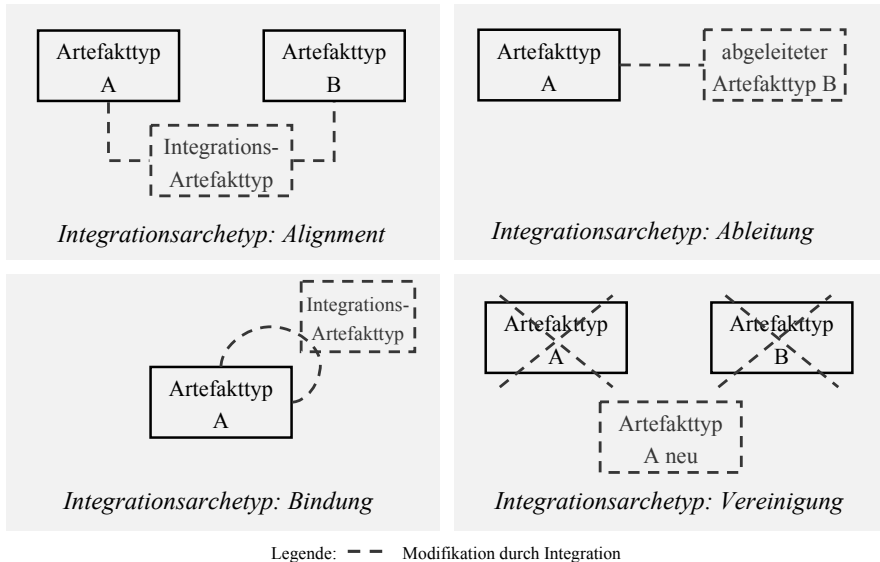


Abb. 2.4. Integrationsarchetypen (Winter 2008a)

Der Integrationsarchetyp *Alignment* beschreibt das gegenseitige aneinander Ausrichten unterschiedlicher und sich unabhängig voneinander verändernder Artefakte. Das klassische Alignment-Beispiel ist das gegenseitige aneinander Ausrichten fachlicher Strukturen, wie der Geschäftsprozessarchitektur und der korrespondierenden Software- und Datenarchitektur. Um eine lose Kopplung zu unterstützen, werden Alignment-Artefakte gebildet. Hierdurch wird es möglich, dass lokale Änderungen nicht unmittelbar Auswirkung auf alle verbundenen Artefakte haben (Glassman 1973; Weick 1976). Ein Beispiel für Alignment-Artefakttypen sind fachliche Services zur Verknüpfung von fachlichen Aktivitäten und IT-Funktionalitäten.

Der zweite Integrationsarchetyp in Abb. 2.4 stellt die Integration als *Ableitung* von Artefakttypen im Metamodell dar. Artefakte eines Typs B werden aus den Artefakten eines Typs A abgeleitet, d. h. ihre Gestaltung beruht zu einem großen Teil auf Vorgaben, welche innerhalb eines anderen Artefakttyps gegeben sind. Die Ableitung der Aufbauorganisation aus der Ablauforganisation (Gaitanides 2007) oder die Ableitung der Softwarearchitektur aus der Datenarchitektur (Stahl u. Völter 2005) sind Beispiele für diesen Integrationsarchetyp.

Der dritte in Abb. 2.4 vorgestellte Integrationsarchetyp wird als *Bindung* bezeichnet. Werden verschiedene Instanzen des gleichen Artefakttyps miteinander verbunden, so entsteht im Metamodell eine rekursive Beziehung. Derartige Rela-

tionen entstehen etwa bei der Verbindung von Softwarekomponenten über Punkt-zu-Punkt-Schnittstellen oder auch bei der Verbindung von Softwaresystemen über ein Integrationsartefakt wie eine Middleware- oder eine EAI-Plattform (Linthicum 2001; Schissler et al. 2002; Rosemann 1999).

Der vierte vorgeschlagene Integrationsarchetyp wird als *Vereinigung* bezeichnet. Integration im Sinne von Vereinigung liegt dann vor, wenn das Metamodell nach der Durchführung von Integrationsprojekten weniger Artefakttypen aufweist als zuvor, z. B. wenn zwei (oder mehr) Artefakttypen zu einem verschmelzen (Rosemann 1999). Dies ist etwa bei Unternehmensübernahmen und Migrationsprojekten der Fall.

Die vier Integrationsarchetypen werden in den folgenden vier Abschnitten weiter beschrieben.

2.3.1 Archetyp 1: Alignment

2.3.1.1 Begriffsbestimmung

Der Integrationsarchetyp *Alignment* beschreibt das gegenseitige zueinander Ausrichten unterschiedlicher und sich unabhängig voneinander verändernder Artefakte und wird in von Aier und Winter (2009) ausführlich am Beispiel der verbindenden Integration zwischen fachlichen Architekturen und IT-Architekturen beschrieben. Dieses „IT/Business Alignment“ stellt seit mehreren Jahren über alle Unternehmens-Größenklassen und über alle Branchen hinweg das Hauptthema der CIOs dar (Luftman 2005).

Da beim Integrationsarchetyp Alignment unterschiedliche Artefakttypen wie fachliche Anforderungen auf der einen Seite und durch IT realisierte Funktionalitäten auf der anderen Seite abgestimmt werden müssen, ist vereinigende Integration nicht möglich. Darum muss hier Integration durch Verbindung erfolgen. Um der (im Gegensatz zum Archetyp Ableitung) unabhängigen Entwicklung der zu verbindenden Artefakte gerecht zu werden, stellen Zuordnungsmechanismen ein zentrales Umsetzungsinstrument dar. Im IT/Business Alignment sind dann z. B. Mechanismen zu spezifizieren, die fachlich-strategische Artefakte (z. B. Leistungen, Ziele) zunächst fachlichen Umsetzungsartefakten (z. B. Prozesse, Organisationsstrukturen) und schließlich auch IT-Umsetzungsartefakten (z. B. Softwarekomponenten, Datenstrukturen) zuordnen.

Es liegt zunächst nahe, die zu verbindenden Artefakte durch direkte Punkt-zu-Punkt-Zuordnungen zu integrieren. Für komplexe Integrationsverhältnisse darf jedoch erwartet werden, dass jedes der beteiligten Artefakte dann an mehreren solcher Zuordnungen beteiligt ist. Änderungen in solchen durch eine Vielzahl von Punkt-zu-Punkt-Zuordnungen verbundenen, komplexen Artefaktstrukturen haben die Tendenz, sich explosionsartig „auszubreiten“. Ein Beispiel für dieses Problem ist die Verknüpfung von Dateistrukturen und Verarbeitungslogik in der Cobol-

Programmierung. Die Änderung der Verarbeitungslogik kann eine Veränderung der „gelesenen“ Dateistruktur erfordern. Als Konsequenz muss nicht nur das Programm angepasst werden, das diese Datei „schreibt“. Auch alle anderen Programme müssen angepasst werden, die diese Datei verarbeiten.

Mit zunehmender Anzahl von Verknüpfungen, zunehmender Komplexität der zu verbindenden Artefaktstrukturen und zunehmender Änderungsdynamik des Gesamtsystems werden direkte Punkt-zu-Punkt-Zuordnungen immer ineffizienter.

Die Lösung dieses Problems sind indirekte Zuordnungen mit Hilfe eines m:n-fähigen, flexiblen Zuordnungsmechanismus („Hub“). Das Konzept „Enterprise Application Integration“ (Linthicum 2000; Kaib 2002) stellt ein gutes Beispiel für einen solchen Zuordnungsmechanismus dar. An die Stelle einer Vielzahl von direkten Punkt-zu-Punkt-Verknüpfungen zwischen allen beteiligten Softwaresystemen tritt ein Zuordnungssystem, zu dem alle beteiligten Softwaresysteme nur eine einzige Verknüpfung (den sog. „Adapter“) haben.

Enterprise Application Integration verbindet jedoch ausschließlich Softwaresysteme. Zur Unterstützung von IT/Business Alignment müssen aber z. B. Aktivitäten aus Prozessen auf der einen Seite mit durch Softwarekomponenten realisierten Funktionalitäten auf der anderen Seite verknüpft werden – mithin also Artefakte unterschiedlicher Typen. Es ist anzustreben, auch diese Form der verbindenden Integration durch indirekte, m:n-fähige, flexible Zuordnungsmechanismen umzusetzen. Ein solcher Zuordnungsmechanismus ist dann architektonisch „zwischen“ fachlichen Strukturen und IT-Strukturen positioniert und spiegelt sich in einer Alignment-Architektur wider. Die Alignment-Architektur beschreibt die Struktur der in ihr enthaltenen Alignment-Artefakte.

Bevor im Folgenden die Herleitung der Anforderungen an die Alignment-Architektur und deren Artefakttypen am Beispiel des IT/Business Alignment aus (Aier u. Winter 2009) nachvollzogen werden, sollen zunächst die allgemeinen Wirkprinzipien an einem Beispiel aus der Systemtheorie betrachtet werden.

2.3.1.2 Beispiele aus der Systemtheorie

Der Ausgangspunkt für die Entkopplung in der Systemtheorie ist das Problem der Komplexität. Die Komplexität der Umwelt ist immer größer als die Komplexität des betrachteten Systems. Das System hat keine ausreichende Kapazität, um für jeden Zustand der Umwelt einen angemessenen eigenen Zustand zu definieren, da es nicht in der Lage ist, für alle Umweltelemente direkte Punkt-zu-Punkt-Beziehungen aufzubauen. Die Möglichkeiten eines Systems, mit diesem Komplexitätsgefälle umzugehen, sind zum einen Ignoranz, d. h. nicht auf Umweltveränderungen zu reagieren, und zum anderen Generalisierung, d. h. auf unterschiedliche Veränderungen der Umwelt in gleicher Art und Weise zu reagieren.

Die Generalisierung unterbricht die direkten Punkt-zu-Punkt-Zuordnungen durch Bündelungen, die im System oder in der Umwelt vorgesehen werden können. Die Bündelung wird in der Systemtheorie unter dem Begriff der Stufentheorie diskutiert (Luhmann 2002). Ashby nennt Systeme, die solche Stufenfunktionen

aufweisen, ultrastabil (Ashby 1981). Ultrastabile Systeme fangen Störungen der Umwelt lokal ab, ohne dass die Störungen auf das gesamte System wirken können. Stufenfunktionen stellen eine Interdependenzunterbrechung dar. Diese Interdependenzunterbrechung wird auch als „loose coupling“ (Glassman 1973; Weick 1976) bezeichnet (Luhmann 2002). Das Gegenteil von „loose coupling“ ist „tight coupling“ und kommt in natürlichen Systemen für gewöhnlich nicht vor.

Je höher der Umfang ist, in welchem ein System „tight coupling“ implementiert, desto gefährdeter ist es, da Störungen, die auf Systemteile wirken, das gesamte System beeinflussen. Dies wurde vor allem in der Diskussion um Hochtechnologie als problematisch charakterisiert, da solche Systeme oft „tight coupling“ implementieren (Luhmann 1991; Perrow 1988). Die Antwort der Systemtheorie ist der Bau einer Schutzhülle, eines Containments, welche selbst wieder auf den Prinzipien des „loose coupling“ beruht (Abb. 2.5) (Luhmann 2002).

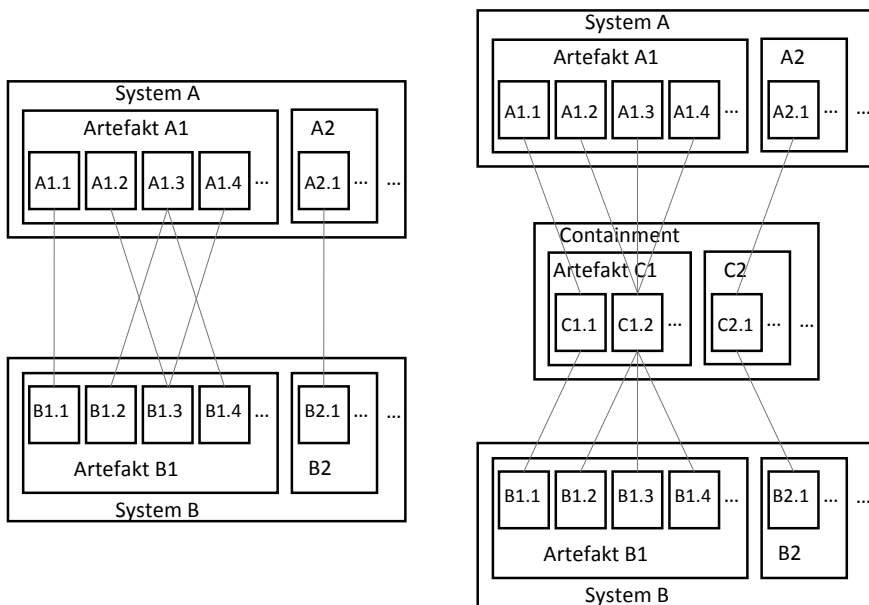


Abb. 2.5. Generischer, entkoppelnder Zuordnungsmechanismus in der Systemtheorie (Aier u. Winter 2009)

Abbildung 2.5 illustriert die Wirkung des Containments. Wenn die elementaren Artefakte des Containments gröber granular sind als die entkoppelten Artefakte, werden direkte m:n-Zuordnungen (linke Seite) durch indirekte m:1- bzw. 1:n-Zuordnungen ersetzt.³ Diese repräsentieren gesamthaft nahezu die gleiche Seman-

³ Die Auflösung der direkten m:n-Zuordnung in m:1- und 1:n-Zuordnungen stellt eine spezielle, stark vergrößernde Lösung dar. Eine allgemeine Lösung für die Vergrößerung ist die Auflösung der direkten m:n-Zuordnung in m:x- und y:n-Zuordnungen mit $m > x$ und $y < n$. Die folgenden Darstellungen illustrieren zur Vereinfachung die spezielle Lösung; Die Aussagen sind jedoch auch für die allgemeine Lösung gültig.

tik der direkten m:n-Zuordnung, verhindern aber gleichzeitig, dass jede Änderung in System A oder System B immer in das jeweils andere System propagiert werden muss.

Mit der Herleitung des Containment-Artefakts beschreibt die Systemtheorie auf einer generischen Ebene den Nutzen und die Wirkung einer entkoppelnden Zuordnung. Durch das Containment können sowohl Artefakte des gleichen Typs wie auch Artefakte unterschiedlicher Typen entkoppelt werden. Das Containment kann sowohl real als auch virtuell umgesetzt werden.

2.3.1.3 Beispiel IT/Business Alignment

Das vorangehend dargestellte Beispiel für Zuordnungsmechanismen liegt zwar nicht im Kontext der in diesem Buch betrachteten Integrationsprojekte, verdeutlicht allerdings sehr gut die grundsätzlichen Wirkmechanismen. Basierend auf diesem Beispiel werden im Folgenden die Anforderungen an die Alignment-Architektur und deren Artefakttypen am Beispiel des IT/Business Alignment hergeleitet.

Das Ziel besteht darin, sich unabhängig entwickelnde Artefakte unterschiedlicher Typen durch Einführung eines Zuordnungsmechanismus so zu entkoppeln, dass Änderungen auf der einen Seite nicht zwangsläufig Änderungen auf der anderen Seite nach sich ziehen und umgekehrt. Angewendet auf das Business/IT-Alignment besteht das Ziel darin, fachliche Architekturen und IT-Architekturen so zu entkoppeln, dass Änderungen auf der Fachseite nicht zwangsläufig Änderungen in der IT nach sich ziehen und umgekehrt. Das konzeptionelle (Struktur-)Modell eines solchen Zuordnungsmechanismus wird als Alignment-Architektur bezeichnet. Zur strukturellen Beschreibung der Alignment-Architektur sind Artefakttypen zu spezifizieren, welche die direkten Beziehungen zwischen den Artefakten der fachlichen und IT-Architekturen durch

- m:1-Beziehungen zwischen den Artefakten der fachlichen Architekturen und den Artefakten der Alignment-Architektur sowie
- 1:n-Beziehungen zwischen den Artefakten der Alignment-Architektur und den Artefakten der IT-Architekturen

auflösen. Durch die m:1- bzw. 1:n-Abbildung muss ein ausreichend großes Komplexitätsgefälle zwischen der Alignment-Architecturebene und den angrenzenden Ebenen erzeugt werden. Nur auf diese Weise kann eine Interdependenzunterbrechung zwischen den Architekturen der angrenzenden Ebenen erreicht werden. Ähnlich wie es z. B. innerhalb der fachlichen und IT-Architekturen Aggregationsebenen zur Vereinfachung gibt, müssen auch innerhalb der Alignment-Architektur Aggregationsebenen vorgesehen werden (Stünzer 1996).

Die Alignment-Architektur und deren Artefakte dienen primär zur „Übersetzung“ der verknüpften Strukturen und sind weder Teil fachlicher Architekturen noch Teil der IT-Architekturen.

Ausgehend von den genannten Anforderungen schlagen (Aier u. Winter 2009) für das IT/Business Alignment die Artefakttypen fachlicher Service, Applikation und Domäne für die Alignment-Architektur vor. Die Artefakttypen stehen in einer hierarchischen Beziehung zueinander (Abb. 2.6).

Fachliche Services repräsentieren betriebswirtschaftlich abgeschlossene Funktionsbündel, die aufgrund der gemeinsamen Unterstützung von Geschäftsprozessen, der gemeinsamen Bewirtschaftung von Informationsobjekten oder gemeinsamer Wiederverwendung zusammenhängen (Schelp u. Winter 2008) und die durch Softwarekomponenten implementiert werden. Da es sich um Funktionsbündel handelt, werden sie durch eine oder mehrere Softwarekomponenten implementiert (1:n-Beziehung) und können in einem oder mehreren Geschäftsprozessen genutzt werden (m:1-Beziehung). Applikationen sind Gruppierungen fachlicher Services, die zusammen bestimmte Geschäftsprozesse unterstützen, zusammen bestimmte Informationsobjekte bewirtschaften oder zusammen wiederverwendet werden (Winter 2003a). Domänen sind Gruppierungen von Applikationen (und damit indirekt von fachlichen Services), welche aufgrund ihrer Nutzung in zusammenhängenden Geschäftsprozessen, ihrer Unterstützung durch zusammenhängende Softwarekomponenten oder ihrer Nutzung zusammenhängender Informationsobjekte einen Integrationsbereich bilden (Hagen 2003).

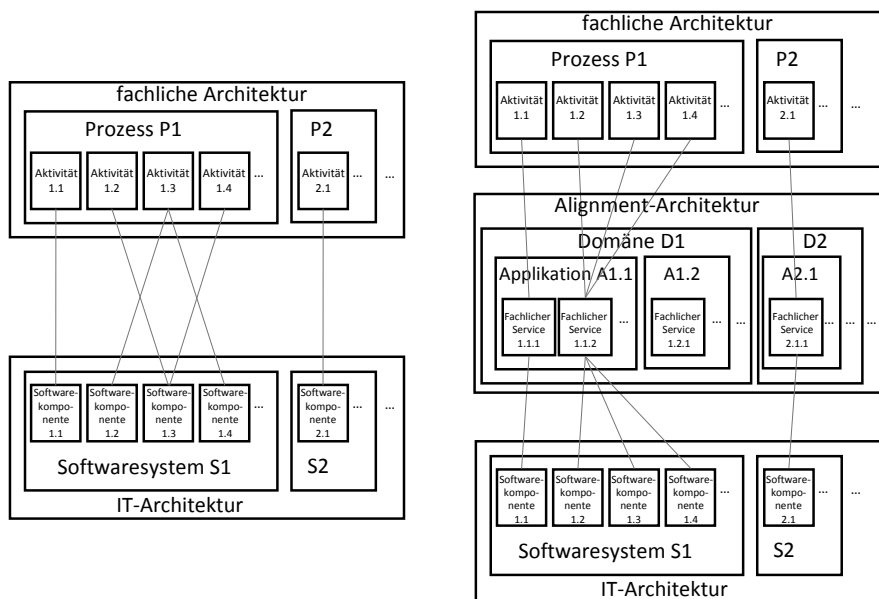


Abb. 2.6. Alignment-Architektur für die entkoppelnde Zuordnung fachlicher und IT-Artefakte (Aier u. Winter 2009)

In Analogie zu Abb. 2.5 zeigt Abb. 2.6 die Alignment-Architektur zur indirekten, m:n-fähigen Zuordnung fachlicher und IT-Artefakte. Wie im Beispiel zur Systemtheorie wird eine neue Architekturebene „zwischen“ fachlichen Architekturen

und IT-Architekturen eingeführt. Die neue Architekturebene dient allein der „Übersetzung“ der verknüpften Strukturen und ist weder Teil fachlicher Architekturen noch von IT-Architekturen. Sie entkoppelt Artefakte verschiedener Typen. Direkte m:n-Zuordnungen werden zu m:1- bzw. 1:n-Zuordnungen aufgelöst. Die Artefakte auf den entkoppelten Ebenen können unabhängig voneinander verändert werden, ohne dass in jedem Fall die jeweils andere Ebene betroffen sein muss. Da mit zunehmender Spezifität der Artefakte der Alignment-Architektur erwartet werden muss, dass der Wertbeitrag der zusätzlichen Architekturebene sinkt, sollten deren elementare Artefakte aggregierter als die entkoppelten Artefakte sein.

2.3.2 Archetyp 2: Ableitung

2.3.2.1 Begriffsbestimmung

Bei der *Ableitung* wird ein Artefakt eines Typs A von einem Artefakt eines Typs B abgeleitet, damit beide Artefakte zueinander konsistent sind. Der Integrationsarchetyp *Ableitung* wird angewendet, wenn ein Artefakttyp in starkem Maße von einem anderen Artefakttypen abhängig ist. Änderungen an dem führenden Artefakttyp ziehen zwangsläufig Änderungen an dem abhängigen Artefakttyp nach sich.

Der Integrationsarchetyp *Ableitung* kommt in der Praxis beispielsweise in folgenden Integrationsaufgaben vor:

- Anwendungssystemarchitektur aus der Datenarchitektur ableiten,
- Aufbauorganisationsarchitektur aus der Ablauforganisation ableiten.

Der Integrationsarchetyp *Ableitung* grenzt sich wie folgt von den Integrationsarchetypen *Alignment* und *Bindung* ab:

Der *Integrationsarchetyp Alignment* bezieht sich – wie der Integrationsarchetyp *Ableitung* – auf Artefakte unterschiedlicher Typen. Allerdings herrscht zwischen den Artefakten keine eindeutige, unmittelbare Abhängigkeit. Daher ist es zweckmäßig, den Archetyp *Alignment* vom Archetyp *Ableitung* zu unterscheiden.

Der *Integrationsarchetyp Bindung* beschreibt, wie sich zwei Artefakte gleichen Typs aneinander binden lassen, z. B. zwei Geschäftsprozesse zweier Organisationen, die eine Kunden-Lieferanten-Beziehung pflegen, oder zwei Datenbanken. Bei einer solchen Integration sind andere Aspekte zu beachten als bei beim Archetypen *Ableitung*, da die Verbindung zwischen zwei Artefakten gleichen Typs im Allgemeinen eine andere Semantik hat als die Verbindung zweier Artefakte unterschiedlichen Typs.

Der Integrationsarchetyp *Ableitung* ist ein Integrationsarchetyp, der bereits seit langem unter den Begriff der Integration fällt. Beispiele für die Verwendung dieses Archetyps sind Begriffe wie „Integrierte Methode“ aus dem Business Engineering oder „Integrierte Anwendungssysteme“, also z. B. Enterprise Resource Plan-

ning-Systeme. Beiden Begriffen ist gemein, dass mit Integration eine innere Konsistenz des jeweiligen Bezugsobjektes gemeint ist. Im Falle des integrierten Anwendungssystems werden verschiedene Funktionen über eine gemeinsame Datenbank integriert, welche die Grundlage für eine innere Konsistenz des Systems bildet. Die Ableitung ist das charakteristische Vorgehen, um die absolute Konsistenz von Artefakten verschiedenen Typs zu gewährleisten. Nur dadurch, dass eines der Artefakte zuerst gestaltet wird und sich der Entwurf des zweiten Artefakts an diesem orientiert, lässt sich die absolute Konsistenz beider Artefakte sicherstellen. Kein anderes Vorgehen als das Ableiten führt zu einer absoluten Konsistenz.⁴ Welches Artefakt von welchem abgeleitet wird, ist dabei nicht immer eindeutig zu bestimmen und hängt in manchen Fällen auch von den Grundannahmen ab, die vor der Gestaltung der Artefakte getroffen worden sind.⁵ Entscheidend für den Artefakttypen Ableitung ist also nicht, welcher Artefakttyp von welchem abgeleitet wird, sondern *dass* ein Artefakt eines Typs von einem Artefakt eines anderen Typs abgeleitet wird, um eine größtmögliche Konsistenz des gesamten Systems sicherzustellen.

Der *Begriff der Ableitung* ist ebenfalls aus der Literatur bekannt. So beschreibt beispielsweise Winter (1998) ausführlich verschiedene Arten der Informationsableitung in Anwendungssystemen. Dabei unterscheidet er unterschiedliche Formen der Informationsableitung: Verdichtung, Verfeinerung, Vererbung, Enumeration und Kombination sowie Simulation. Insbesondere die Verdichtung und Verfeinerung sind auch für die Integration bedeutende Erscheinungsformen der Ableitung. Bei der Verdichtung beispielsweise unterscheidet Winter zwischen Aggregation, Assoziation und Selektion. Bei der Aggregation werden zusammenwirkende Objekte zu einem abstrakten Objekt zusammengefasst, beispielsweise durch Summenbildung. Bei der Assoziation wird eine Gruppe gleichartiger Objekte gebildet. Ein Beispiel für eine Assoziation in der Unternehmensarchitekturmodellierung ist die Zusammenfassung von Applikationen zu Domänen. Im Rahmen der Selektion werden für eine bestimmte Problemstellung irrelevante Objekte aus der weiteren Betrachtung ausgeschlossen. Winter zeigt, dass bereits durch Verknüpfung dieser drei Verdichtungsoperationen komplexe Probleme gelöst werden können. Die Verfeinerung ist ein zur Verdichtung entgegengesetztes Konzept und geht nicht

⁴ Der Artefakttyp *Alignment* grenzt sich durch das Ziel der Konsistenz von dem Artefakttypen *Ableitung* deutlich ab. Aus theoretischen Überlegungen heraus wäre es wünschenswert, die Software-Architektur direkt aus der Prozessarchitektur abzuleiten („IT follows Business“). So könnte eine größtmögliche Konsistenz zwischen Software- und Prozessarchitektur gewährleistet werden, also ein bestmögliches IT/Business-Alignment. Die Erfahrung zeigt aber, dass eine direkte Ableitung der Softwarearchitektur nicht praktikabel ist, da sich die Geschäftsprozesse eines Unternehmens rasch ändern, Änderungen an der Softwarearchitektur allerdings häufig lange dauern (Aier u. Winter 2009). Mit der Lösung, eine entkoppelnde Alignment-Ebene zwischen Organisations- und Softwareebene einzuführen, wird also bewusst auf eine bestmögliche Konsistenz zwischen Software- und Prozessarchitektur verzichtet. Vordergründig stellt sich die Einführung der Alignment-Ebene daher als die zweitbeste Lösung dar, weil die erstbeste Lösung, die Schaffung von Konsistenz, nicht gewählt wird; hintergründig ist die Einführung der Alignment-Ebene allerdings die einzig praktikable Lösung des Problems des IT/Business-Alignments.

⁵ Dieser Zusammenhang wird auch anhand der Beispiele am Ende des Kapitels verdeutlicht.

mit dem Verlust irrelevanter Informationen einher, sondern mit der Anreicherung. Bei Integrationsproblemen werden Elemente aus dem führenden Artefakt häufig selektiert und assoziiert, bevor die Ableitung erfolgen kann. Allerdings wird bei Integrationsproblemen selten eine Verdichtung ausreichen, sondern immer auch eine Anreicherung mit weiteren Informationen (Verfeinerung) notwendig sein.

2.3.2.2 Beispiel Aufbau- und Ablauforganisation

Das erste Beispiel zeigt, wie Ablauf- und Aufbauorganisation konsistent gestaltet werden können. Bei der Untersuchung der Organisation einer Unternehmung haben sich zwei Sichten etabliert: eine aufbauorganisatorische und eine ablauforganisatorische Sicht auf die Organisation. Die aufbauorganisatorische Sicht beschreibt die Aufgabenverteilung, die Ablauforganisation die Arbeitsverteilung (Gaitanides 2007). Beide Aspekte der Organisation einer Unternehmung lassen sich gestalten und sind stark voneinander abhängig.

Das Gestaltungsziel auf aufbauorganisatorischer Ebene besteht hauptsächlich darin, funktionale Synergien zu nutzen. Funktionale Synergien können dann entstehen, wenn gleichartige Aufgaben organisatorisch zusammengefasst werden. Beispiele für funktionale Synergien sind Lerneffekte innerhalb von Organisationseinheiten oder eine bessere Ressourcenausnutzung aufgrund von stochastischen Größenvorteilen. Zeit, Qualität und Kosten der Produktion werden hauptsächlich innerhalb einer Organisationseinheit optimiert mit dem Ziel, auf diese Weise die Zeit, Qualität und Kosten des Ganzen zu optimieren.

Das Gestaltungsziel auf ablauforganisatorischer Ebene ist eine effiziente Gestaltung der Arbeitsabläufe. Ein effizienter Arbeitsablauf zeichnet sich durch eine geringe Durchlaufzeit der Prozesse, eine hohe Qualität der Prozessleistung und geringe Kosten bei der Erstellung der Prozessleistung aus. Im Gegensatz zur funktionalen Effizienz bezieht sich die Prozesseffizienz also nicht auf Prozessschritte, sondern auf den Prozess als Ganzen.

Ein Konzept der strengen Aufgabenteilung wurde z. B. von Fredrick Taylor zu Beginn des 20. Jahrhunderts entwickelt und vor allem infolge seiner Umsetzung durch Henry Ford berühmt, der die Fließbandarbeit in der Automobilproduktion eingeführt hat, was zu großen Produktivitätssteigerungen führte. Bis in die 80er Jahre galt dann das Prinzip der Aufgabenteilung als das wichtigste organisatorische Prinzip. Im Rahmen der Organisationsgestaltung wurde zunächst die Aufbauorganisation gestaltet, um funktionale Synergien zu nutzen. Gaitanides (2007) führt zahlreiche Autoren an, die sich in der Zeit bis zu den 80er Jahren auch mit der Optimierung der Ablauforganisation beschäftigen (z. B. Kosiol 1962), während sie allerdings die Aufbauorganisation als gegeben betrachten und nicht grundlegend verändern, um die Prozesseffizienz zu erhöhen.

Diese Ansicht hat sich durch populäre Beiträge von Hammer (1990) und Hammer u. Champy (1993) geändert. Durch Business Process Reengineering wird das Ziel einer möglichst hohen Prozesseffizienz verfolgt. Im Zuge dessen soll die

Aufbauorganisation so gestaltet werden, dass die Prozesse möglichst effizient ausgeführt werden können.

Für dieses radikale Umdenken, das Hammer und Champy fordern, sprechen vor allem zwei Gründe: erstens veränderte Absatzbedingungen und zweitens veränderte Produktionsbedingungen.

Erstens haben sich die Absatzbedingungen für viele Produkte geändert. Zu Zeiten von Taylor und Ford besaß kaum ein Konsument ein Automobil. So herrschte in diesem Markt – und in vielen anderen Industriegütermärkten – ein Produzentenmarkt. Ziel der Produzenten war es, möglichst viele potentielle Kunden für Produkte zu schaffen. Dies war vor allem durch einen günstigen Absatzpreis möglich. Preisreduktionen ließen sich durch eine Kostenreduktion durch Nutzung funktionaler Synergien erzielen. Primat der Organisationsgestaltung war die Aufbauorganisation. Zu Zeiten von Hammer und Champy hingegen war der Markt mit vielen Konsumgütern gesättigt. Hinzukam, dass die Konkurrenz auf vielen Märkten anstieg, beispielsweise durch die einsetzende Globalisierung. Im Gegensatz zu Zeiten von Taylor und Ford herrschte für viele Güter also ein Konsumentenmarkt. Für viele Produzenten war es nicht mehr ausreichend, die Zahl potentieller Kunden durch Preissenkung zu erhöhen. Um sich von Konkurrenten abzusetzen, mussten viele Produzenten die Qualität ihrer Absatzprodukte erhöhen. In der Automobilindustrie beispielsweise wurden Autos produziert, die im Rahmen einer Massenanfertigung an individuelle Kundenbedürfnisse angepasst wurden („Mass Customisation“). Solche Qualitätsverbesserungen ließen sich nur durch einen starken Fokus auf die Gestaltung der Arbeitsprozesse, also auf die Gestaltung der Ablauforganisation, realisieren.

Zweitens haben sich die Produktionsbedingungen geändert. Während sich Taylor und Ford einer Masse an schlecht ausgebildeten Arbeitnehmern gegenüber sahen, waren die westlichen Gesellschaften zur Zeit von Hammer und Champy infolge eines besseren Bildungs- und Ausbildungssystems mit breit ausgebildeten Arbeitskräften ausgestattet, die zu verschiedenen Tätigkeiten herangezogen werden konnten. Durch die neuen Möglichkeiten der Informationstechnologie, z. B. durch Konfigurationssysteme, war es weiterhin möglich, die Fähigkeiten der Organisation zu erweitern. Hammer u. Champy (1993) beschreiben beispielsweise eine Versicherung, in der verschiedene Expertenteams Schadensmeldungen bearbeiteten. Bei diesem Arbeitsablauf durchlief jede Schadensmeldung mehrere Abteilungen. Zur Bearbeitung vieler Schadensmeldungen war allerdings, dank gut ausgebildeter Mitarbeiter und unterstützender Informationssysteme, kein Expertenwissen vonnöten. Im Rahmen eines Business Process Reengineering ließ sich durch Abschaffung der Schnittstellen zwischen verschiedenen Sachbearbeitern bei vielen der Schadensmeldungsbearbeitungsprozesse die Durchlaufzeit drastisch reduzieren, was zu einer Qualitätssteigerung der angebotenen Versicherungsprodukte führte.

Die Methode PROMET[®]-BPR (IMG 1997a) ist ein Beispiel für eine Methode, in der die Aufbauorganisation aus der Ablauforganisation abgeleitet wird. Das Vorgehensmodell der Methode unterteilt sich in die groben Phasen Vorstudie, Makro-Entwurf, Mikro-Entwurf und Grob-Entwurf. In der Vorstudie werden auf

Basis der Geschäftsstrategie Makro-Prozesse identifiziert und es wird eine Projektorganisation definiert. In der Phase des Makro-Entwurfs steht für jeden Makro-Prozess die Vision eines optimalen Prozesses im Vordergrund. In dieser Phase wird der Makro-Prozess auch in Mikroprozesse zerlegt. Im Mikro-Entwurf werden diese Mikroprozesse im Detail gestaltet. Hier entstehen u. a. Aufgabenkettendiagramme. In der Phase der Umsetzung werden die modellierten Mikro-Prozesse realisiert (IMG 1997a). Im Rahmen des Mikro-Entwurfs wird auch die Aufbauorganisation durch Stellenbeschreibungen definiert. Eine Stellenbeschreibung gibt die Stellenbezeichnung, die Bezeichnung der übergeordneten Stelle, das Ziel der Stelle und die Aufgaben der Stelle an. Durch die Angabe der jeweils übergeordneten Stelle ergibt sich bei einer umfassenden Anwendung der Methode PROMET[®]-BPR die gesamte Aufbauorganisation eines Unternehmens (IMG 1997a). Die Stellenbeschreibung wird auf Grundlage verschiedener Dokumente erstellt, die bei der Anwendung vorhergehender Aktivitäten der Methode entstehen.⁶

Zusammenfassend lässt sich festhalten, dass Ablauforganisation und Aufbauorganisation eng miteinander verknüpft sind. Bis in die 80er Jahre war es üblich, die Ablauforganisation aus der Aufbauorganisation abzuleiten. In den 90er Jahren wurde propagiert, die Aufbauorganisation aus der Ablauforganisation abzuleiten. Beide Vorgehensweisen führen zu einer Konsistenz der beiden Strukturen. Welche der beiden Artefakttypen von welchem abgeleitet werden, hängt von den Produktions- und Absatzbedingungen ab. Stehen Ziele der Prozesseffizienz im Vordergrund, das ist meist auf gesättigten Konsumentenmärkten mit hoher Konkurrenz der Fall, so empfiehlt es sich, die Ablauforganisation zuerst zu gestalten und die Aufbauorganisation aus der Ablauforganisation abzuleiten. Steht das Ziel von Kostenersparnissen durch funktionale Synergien im Vordergrund, das ist vor allem auf nicht gesättigten Produzentenmärkten mit geringer Konkurrenz der Fall, dann sollte die Aufbauorganisation zuerst gestaltet werden und die Ablauforganisation daraus abgeleitet werden. Um eine hohe Prozesseffizienz zu erreichen, werden andere Anforderungen an die Produktionsbedingungen gestellt, als um funktionale Synergien zu erzielen.

2.3.2.3 Beispiel Funktionen und Daten

Ein zweites Beispiel für Ableitungsbeziehungen entstammt dem Software-Engineering und bezieht sich auf die Konsistenz zwischen Funktions- und Datenstrukturen eines Anwendungssystems. Diese hängen eng miteinander zusammen. Daten und Funktionen eines Anwendungssystems sind jeweils eine unterschiedliche Sicht auf dasselbe.

⁶ Diese Dokumente, auf Grundlage derer die Stellenbeschreibung erstellt wird, sind das Aufgabenkettendiagramm Mikro, das Aufgabenverzeichnis Mikro, das Applikationsverzeichnis Mikro, die Kostenanalyse, die Matrix Mikro-Prozess-Aufgabe, die Matrix Stelle-Organisationseinheit, die Matrix Stelle-Standort, die Aufwandsverteilung Mikro und die Durchlaufzeitanalyse (IMG 1997a).

Ähnlich wie im ersten Beispiel bestehen zwei naheliegende Möglichkeiten, die Daten- und Funktionssicht auf ein Anwendungssystem konsistent zu gestalten: Es kann entweder zuerst die Datenstruktur entwickelt werden, aus der sich dann die Funktionsstruktur des Systems ableiten lässt, oder es kann zuerst die Funktionsstruktur des Systems bestimmt werden, aus der sich dann die Datenstruktur ableiten lässt. Beide Vorgehen haben in der Praxis ihre Berechtigung und hängen von den Zielen ab, die mit dem Anwendungssystem verfolgt werden.

Wenn besondere Anforderungen an die Konsistenz der Daten eines Anwendungssystems gestellt werden, sollte zunächst die Datenstruktur gestaltet werden. Ein Beispiel für ein solches Anwendungssystem ist ein Buchhaltungssystem. Für Buchhaltungssysteme schreibt der Gesetzgeber besondere Sorgfalt bei Haltung und Aufbewahrung der Daten vor. Daher empfiehlt es sich hier, die Datenstrukturen des Anwendungssystems konsistent zu gestalten und seine Funktionen aus den Datenstrukturen abzuleiten.

Wenn hingegen besondere Anforderungen an die Funktionalitäten eines Systems gestellt werden, sollten zunächst die Funktionen spezifiziert werden. Im Anschluss daran können dann dazu passende Datenstrukturen entwickelt werden. So können beispielsweise Real-Time-Funktionen eine inkonsistente Datenhaltung erfordern. Ein Beispiel hierfür sind Führungsinformationssysteme. Ihre Anforderungen an die Laufzeit sind häufig kritisch. Daher werden die Daten zu Auswertungszwecken häufig in Datenlagern in aggregierter Form vorgehalten. Es empfiehlt sich, zunächst die Funktionalitäten des Führungsinformationssystems zu definieren und erst dann eine geeignete Datenarchitektur zu entwerfen. So gehen beispielsweise Holten u. Knackstedt (1999) bei der Definition eines Referenzmodells für Führungsinformationssysteme in Handelsunternehmen von den Funktionen aus, die dieses System unterstützen soll, und entwickeln erst im Anschluss daran ein Datenmodell, auf denen diese Funktionen operieren.

Im Zuge der Diskussion um das Paradigma der Serviceorientierung werden verschiedene Vorgehensmodelle zur Servicedefinition diskutiert (für eine Literaturanalyse vgl. z. B. Beverungen et al. 2008). Bei der Servicedefinition wird in vielen Methoden von fachlichen Funktionsstrukturen ausgegangen. Heutschi (2007) beschreibt den Ansatz zur Identifikation von Services bei der Deutsche Post Brief, der stärker auf den Datenstrukturen basiert. Die Deutsche Post Brief entwickelte zunächst eine Applikationsarchitektur, in der fachliche Funktionen und Daten redundanzfrei dargestellt wurden. Grundlage für die Definition von Services waren dann hauptsächlich die Datenstrukturen der Applikationsarchitektur, die so genannten Geschäftsobjekte. Beispiele für solche Geschäftsobjekte sind „Kunde“, „Rechnung“ oder „Vertrag“. Aus diesen Geschäftsobjekten wurden dann Operationen abgeleitet, z. B. „anlegen“, „suchen“ oder „löschen“.

Das Beispiel zusammenfassend lässt sich festhalten, dass Daten und Funktionen eines Anwendungssystems eng zusammenhängen. Wenn besondere Anforderungen an die Datenstruktur gestellt werden, insbesondere an ihre Konsistenz, sollte zuerst die Datenarchitektur gestaltet werden, dann sollten die Funktionsstrukturen daraus abgeleitet werden. Wenn besondere Anforderungen an die Funk-

tionsstrukturen eines Anwendungssystems gestellt werden, sollten diese zunächst beschrieben und dann die Daten des Systems daraus abgeleitet werden.

2.3.2.4 Zusammenfassung

Erscheinungsformen der Ableitung werden bereits in der klassischen Integrationsliteratur diskutiert. Ebenso ist der Begriff der Ableitung in der klassischen Wirtschaftsinformatikliteratur verankert. Der Integrationsarchetyp Ableitung grenzt sich deutlich von anderen Integrationsarchetypen ab, vor allem durch das Merkmal, Konsistenz zweier Artefakttypen zu schaffen. Es kann von den Zielen und Rahmenbedingungen abhängen, welcher Artefakttyp von welchem abgeleitet wird. Ableitungsbeziehungen bestehen z. B. in der Organisationsgestaltung zwischen Ablauf- und Aufbauorganisation und im Software-Engineering zwischen Funktionen und Daten.

2.3.3 Archetyp 3: Bindung

2.3.3.1 Begriffsbestimmung

Beim Integrationsarchetyp *Bindung* werden verschiedene Instanzen des gleichen Artefakttyps miteinander verbunden, so dass im Metamodell eine rekursive Beziehung entsteht. Derartige Relationen entstehen etwa bei der Verbindung von Softwarekomponenten (Informationssystemintegration) (Linthicum 2001; Rosemann 1999; Schissler et al. 2002), bei der Verbindung von Teilprozessen zu einem Ganzen (Geschäftsprozessintegration) oder der Verbindung von Geschäftsprozessen und/oder Softwaresystemen über Unternehmensgrenzen hinweg (unternehmensübergreifende Integration, Business-to-Business-Integration).

Im Gegensatz zum Integrationsarchetyp Alignment werden hier nicht die Artefakte unterschiedlicher Typen integriert, sondern Artefakte gleichen Typs. Darum wäre es prinzipiell auch möglich eine vereinigende Integrationsoperation anzuwenden. Allerdings ist dies oft nicht sinnvoll, da vereinigte, monolithische Artefakte durch die nicht vorhandene Explizierung ihrer Schnittstellen oft eine hohe Änderungsresistenz aufweisen. Diese Änderungsresistenz tritt dabei sowohl bei der Veränderung von Artefakten des gleichen Typs (z. B. Veränderung von Funktionalitäten in Softwaresystemen) als auch bei Veränderungen von Artefakten anderer Typen auf (z. B. Veränderung der IT-Infrastruktur für bestehende Softwaresysteme).

Die Verknüpfung der Artefakte des gleichen Typs kann auf zwei unterschiedliche Arten ausgeführt werden:

1. Die Artefakte können durch eine direkte Punkt-zu-Punkt-Verbindung integriert werden.
2. Die Artefakte können durch eine indirekte Verbindung über einen Hub in einer Hub-and-Spoke-Struktur integriert werden.

Als Punkt-zu-Punkt-Verbindung wird die direkte Integration von zwei gleichberechtigten Artefakten bezeichnet. Werden alle Artefakte einer vernetzten Struktur direkt miteinander verbunden, wächst die Zahl der Verbindung V in Abhängigkeit von der Anzahl der Artefakte n exponentiell an:

$$V = n \cdot \frac{n-1}{2} \quad (2.1)$$

Bei einer hohen Anzahl an zu integrierenden Artefakten erreicht das Gesamtsystem schnell eine Komplexität, die nur schwer beherrschbar und änderbar ist.

2.3.2.2 Beispiel Softwaresystemlandschaft

In dem Beispiel in Abb. 2.7 sind 16 Softwaresysteme und somit nach oben stehender Formel bereits 120 Schnittstellen vorhanden. In der Realität treten meist weniger Schnittstellen auf, da nicht jedes Artefakt immer mit jedem anderen verbunden sein muss. Ebenso sind im gegebenen Beispiel die Systeme einer Systemklasse nicht miteinander verbunden – dennoch ist die Komplexität bereits beachtlich. Wird im Beispiel in Abb. 2.7 ein weiteres Softwaresystem hinzugefügt, müssen im Extremfall bereits zehn neue Schnittstellen geschaffen werden.

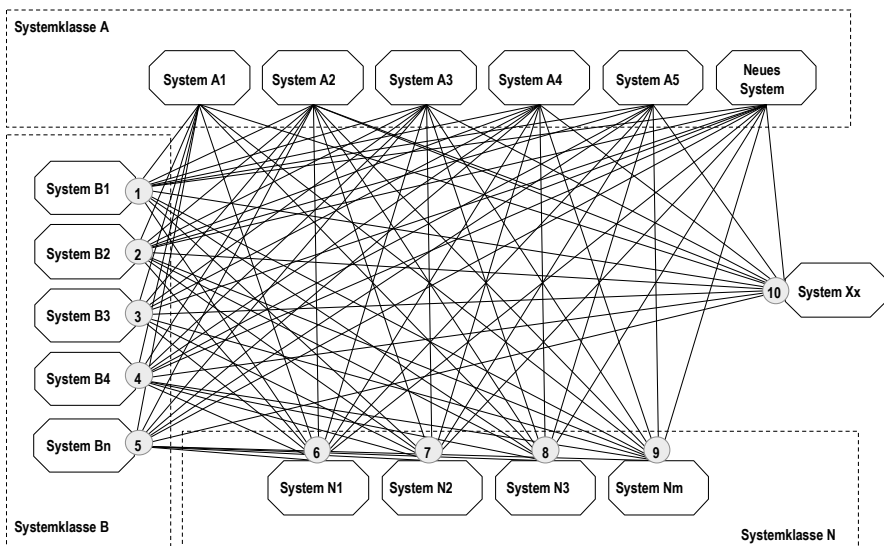


Abb. 2.7. Punkt-zu-Punkt-integrierte Softwaresystemlandschaft (Aier 2007)

Eine solch komplexe Struktur resultiert in realen Systemen häufig aus der Veränderung eines Systems im Verlaufe der Zeit – insbesondere durch die sukzessive Ergänzung weiterer Komponenten ohne eine Neugestaltung der betroffenen Teilarchitektur (Kaib 2002).

Die kurzfristigen Vorteile einer direkten Punkt-zu-Punkt-Verbindung, wie z. B. die schnelle Realisierung notwendiger Verbindungen oder die individuelle Gestaltungsmöglichkeit einzelner Verbindungen, werden auf lange Sicht durch ihre Nachteile kompensiert. Die beiden wesentlichen Nachteile einer direkten Punkt-zu-Punkt-Verbindung sind der hohe Aufwand für die Veränderung der Teilarchitektur sowie bei der Integration zusätzlicher Komponenten (Winkeler et al. 2001).

Die Nachteile der direkten Punkt-zu-Punkt-Verbindung bei einer hohen Anzahl zu verbindender Artefakte können durch die Gestaltung der Verknüpfung in Form eines Hubs, welcher auch von anderen Artefakten genutzt wird, vermieden werden. Typische Beispiele für eine Hub-and-Spoke-Architektur sind (Winter 2003a):

- die Integration von Softwaresystemen durch klassische Middleware, Enterprise Application Integration (EAI, Abb. 2.8) oder große Enterprise Resource Planning (ERP) Systeme (Kaib 2002),
- die Integration von operativen und analytischen Applikationen durch Data Warehouses (Inmon 1996)
- oder die Integration von Produkten und Dienstleistungen über eine Business Collaboration Infrastructure (Alt et al. 2002).

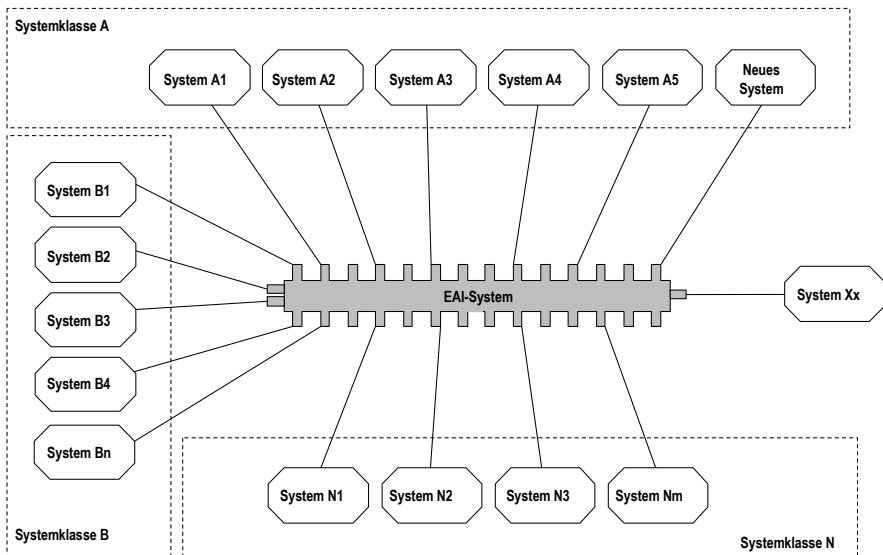


Abb. 2.8. Integration über eine zentrale EAI-Plattform (Aier 2007)

Allen diesen Ansätzen ist gemein, dass durch die Hub-and-Spoke-Integration die Anzahl der Verbindungen stark auf die Anzahl der zu verbindenden Komponenten reduziert werden kann. Um diese Reduktion der Komplexität zu erreichen,

muss der Hub nun anpassbare Funktionalitäten bereitstellen, die sonst von jeder einzelnen Verknüpfung zwar spezifisch, aber redundant implementiert worden sind. Bei diesen Funktionalitäten handelt es sich primär um Übersetzungsfunktionen, die oft auch mit einer Standardisierung des Vokabulars einher gehen.

2.3.4 Archetyp 4: Vereinigung

2.3.4.1 Begriffsbestimmung

Im Rahmen der klassischen, phänomenologischen Analyse des Integrationsbegriffs werden bereits seit langem die Integrationsformen *vereinigen* und *verbinden* voneinander unterschieden (vgl. Literaturanalyse von Rautenstrauch 1993). Der vierte Integrationsarchetyp *Vereinigung* unterscheidet sich von den drei zuvor präsentierten Integrationsarchetypen fundamental, weil er der einzige Integrationsarchetyp ist, der nicht Elemente miteinander verbindet. Der Integrationsarchetyp Vereinigung zeichnet sich dadurch aus, dass ein hinreichend detailliertes Metamodell von dem System, in dem die Integration durch Vereinigung stattfindet, nach Durchführung der Integration weniger Entitäten enthält als davor.

Es können vielfältige Beispiele für eine Integration durch Vereinigung in der betrieblichen Wirklichkeit gefunden werden. Integration durch Vereinigung findet beispielsweise bei Unternehmenszusammenschlüssen oder -übernahmen statt. In einer solchen Situation können Strategien, Produkte, Prozesse und Systeme miteinander vereinigt werden.

In diesem Unterkapitel soll der Integrationsarchetyp Vereinigung näher dargestellt werden. Das Unterkapitel gliedert sich in einen ersten, analytischen und einen zweiten, praktischen Teil: Im ersten Teil wird der hier vertretene metamodellbasierte Integrationsbegriff beschrieben. Damit wird erstens das Ziel verfolgt abzugrenzen, welche Phänomene unter den Integrationsarchetypen Vereinigung fallen und welche nicht; zweitens wird das Ziel verfolgt, ein Verständnis für den metamodellbasierten Ansatz zu schaffen. Im zweiten, praktischen Teil wird die Unternehmensvereinigung auf das Vorkommen des Integrationsarchetypen Vereinigung näher untersucht.

2.3.4.2 Analyse des metamodellbasierten Integrationsbegriffs „Vereinigung“

Basis des diesem Buch zugrundeliegenden Integrationsverständnisses ist die Unterscheidung zwischen Objektmodell und Metamodell (für die Bezeichnung „Objektmodell“ vgl. Schütte 1998). Ein Modell repräsentiert ein Objektsystem für Zwecke eines Subjekts. In diesem Buch sei als Objektmodell ein Modell bezeichnet, das sich unmittelbar auf Objekte in einem realen Unternehmen bezieht. Das Objektmodell befindet sich also auf Ebene der Instanzen. Als Metamodell hinge-

gen sei in diesem Buch ein Modell bezeichnet, das die Modellierungssprache eines Objektmodells beschreibt. Dies sei an dem Beispiel eines Aufbauorganisationsmodells erläutert: Modelle der Aufbauorganisation auf Instanzenebene sind Organigramme, die einzelne Stellen und Organisationseinheiten zueinander in Beziehung setzen. Metamodelle der Aufbauorganisation beschreiben die sprachlichen Elemente eines solchen Modells, d. h. beispielsweise Stellen oder Organisationseinheiten.

Metamodelle können auf unterschiedlichen Detaillierungsebenen erstellt werden. Ein sehr grobgranulares Metamodell eines Organigramms⁷ kann beispielsweise die zwei Typen „Stelle“ und „Organisationseinheit“ voneinander unterscheiden und ihre Beziehungen zueinander darstellen. Ein sehr feingranulares Metamodell kann allerdings auch einzelne Stellentypen voneinander unterscheiden, z. B. den Stellentyp „Finanzbuchhalter“ vom Stellentyp „Controller“. Das feingranularste Metamodell differenziert alle Entitäten, die sich betriebswirtschaftlich voneinander unterscheiden lassen. Zwei Stellen für Finanzbuchhalter, deren Verantwortungs- und Kompetenzbereich sich nicht voneinander unterscheidet, müssen auf Metamodellebene zusammengefasst werden. Nachfolgend gehen wir in diesem Unterkapitel jeweils von möglichst feingranularen Modellen aus.

Eine Vereinigung zeichnet sich dadurch aus, dass nach der Vereinigung weniger Elemente vorhanden sind als vor der Vereinigung. Dabei lässt sich unterscheiden zwischen einer Reduktion der Elementzahl auf Instanzenebene und einer Reduktion der Elementzahl auf Metamodellebene. Bei einer Differenzierung zwischen den zwei Fällen „Reduktion der Elementzahl“ und „keine Reduktion der Elementzahl“ auf beiden Ebenen (Instanzen- und Metamodellebene) ergibt sich logisch eine 2 x 2-Matrix mit folgenden vier Fällen, vgl. Tab. 2.2.

1. Weder die Anzahl der Elemente auf Instanzenebene noch die Anzahl der Elemente auf Metamodellebene reduziert sich.
2. Die Anzahl der Elemente auf Instanzenebene reduziert sich, die Anzahl der Elemente auf Metamodellebene allerdings nicht.
3. Die Anzahl der Elemente auf Metamodellebene reduziert sich, die Anzahl der Elemente auf Instanzenebene allerdings nicht.
4. Sowohl die Anzahl der Elemente auf Metamodellebene als auch die Anzahl der Elemente auf Instanzenebene reduziert sich.

Integrationen durch Vereinigung äußert sich immer entweder durch Fall (3) oder durch Fall (4). Allerdings sind nicht alle Erscheinungsformen des Falls (4) das Ergebnis einer Integration durch Vereinigung. Nachfolgend soll anhand der vier Fälle erläutert werden, unter welchen Bedingungen es sich um eine Integration durch Vereinigung handelt und unter welchen nicht.

⁷ Ein Organigramm auf Instanzenebene beschreibt die Aufbauorganisation eines Unternehmens, d. h. insbesondere die Stellen eines Unternehmens (ggf. mit Attributen wie z. B. Kompetenz- und Verantwortungsbereich) und die hierarchische Gliederung der Stellen.

Tabelle 2.2. Vier Fälle der Elementzahlreduktion in Objekt- und Metamodell

		Instanzenebene	
		Keine Reduktion der Elementzahl	Reduktion der Elementzahl
Meta-modell-ebene	Keine Reduktion der Elementzahl	Fall 1: notwendigerweise keine Integration durch Vereinigung	Fall 2: notwendigerweise keine Integration durch Vereinigung
	Reduktion der Elementzahl	Fall 3: notwendigerweise Integration durch Vereinigung	Fall 4: möglicherweise Integration durch Vereinigung

In Fall (1) reduziert sich die Elementzahl weder auf Instanzen- noch auf Meta-modellebene. Dass in einem solchen Fall, vorausgesetzt die Metamodelle sind hinreichend feingranular, niemals eine Integration durch Vereinigung vorliegen kann, ist auf den ersten Blick ersichtlich und bedarf keiner weiteren Erläuterung.

In Fall (2) reduziert sich die Elementzahl auf Instanzenebene, nicht aber auf Metamodellebene. Auf Metamodellebene werden Entitäten, die aus betriebswirtschaftlicher Perspektive gleich sind, zusammengefasst. Wenn sich die Anzahl solch gleichartiger Elemente auf Instanzenebene reduziert, dann ändert sich die Anzahl der Elemente auf Metamodellebene nicht. Ein solcher Fall fällt allerdings nicht unter den Integrationsbegriff. Dies soll anhand eines Beispiels plausibel gemacht werden: In einer Unterabteilung der Buchhaltung eines Unternehmens sind acht Vollzeitstellen für die Erfassung von Rechnungen vorgesehen. In diesem Beispiel sind in einem Organigramm auf Instanzenebene alle acht Stellen und die dazugehörigen Mitarbeiter aufgeführt. In einem Organigramm auf Metamodellebene allerdings werden die acht Stellen zu einer Stelle zusammengefasst, da sich die einzelnen Stellen aus betriebswirtschaftlicher Sicht nicht voneinander unterscheiden. Infolge einer Untersuchung von einer Unternehmensberatung ist das Management der Meinung, die Arbeit lasse sich auch durch sechs Vollzeitstellen erledigen und streicht die Stellen zweier Mitarbeiter. Auf Ebene des Objektmodells findet eine Reduktion der Elementzahl statt, auf Ebene des Metamodells allerdings nicht. Es ist also der Fall (2) erfüllt. Es findet auch eine Vereinigung statt: Die Aufgaben derjenigen Mitarbeiter, deren Stellen gestrichen worden sind, müssen auf die weiteren Mitarbeiter aufgeteilt werden. Allerdings fällt das Beispiel nicht unter den Integrationsbegriff.

In Fall (3) reduziert sich die Anzahl der Elemente auf Metamodellebene, die Anzahl der Elemente auf Instanzenebene bleibt allerdings gleich. Ein Beispiel

hierfür ist die Integration zweier Organisationseinheiten, beispielsweise des Controllings und der Buchhaltung. Auf Ebene der Instanzen reduziert sich *ceteris paribus* die Zahl der Elemente, das sind die Mitarbeiter, nicht. In einem hinreichend detaillierten Metamodell allerdings werden die Organisationseinheiten „Controlling“ und „Buchhaltung“ vor der Integration durch Vereinigung voneinander unterschieden, nach der Integration allerdings zu einer einzigen Organisationseinheit zusammengefasst. Wenn in einem Metamodell Fall (3) auftritt, so hat immer eine Integration durch Vereinigung stattgefunden.

Bei Fall (4) hingegen reduziert sich die Elementzahl sowohl auf Instanzen- als auch auf Metamodellebene. Die Reduktion auf Instanzenebene kann allerdings zwei Ursachen haben. Eine solche Reduktion der Elementzahl kann Ergebnis einer Integration durch Vereinigung sein, sie muss es aber nicht. Die sei an zwei Beispielen erläutert. Das erste Beispiel sei eine Kombination aus den beiden Beispielen von Fall (2) und Fall (3): In einer Unternehmung werden zwei Organisationseinheiten zusammengeführt, z. B. die Finanzbuchhaltung und das Controlling. Dadurch ergeben sich Synergieeffekte, sodass im Rahmen dieser Reorganisation einige Stellen gestrichen werden können. Es handelt sich hierbei um Fall (4), weil sich sowohl auf Instanzenebene (Mitarbeiter) als auch auf Metamodellebene (Organisationseinheiten) die Elementzahlen reduzieren. Die neu entstehende Organisationseinheit „Rechnungswesen“ nimmt alle betrieblich relevanten Aufgaben wahr, die vorher von den zwei Organisationseinheiten getrennt wahrgenommen wurden. Es handelt sich also um eine Integration durch Vereinigung.

In einem zweiten Beispiel wird eine Organisationseinheit aus dem Unternehmen an ein anderes Unternehmen verkauft. So kann beispielsweise das Call-Center eines Unternehmens an einen Fremdanbieter verkauft werden, der dann die Aufgaben des Call-Centers wahrnimmt. In diesem Fall reduziert sich ebenfalls die Elementzahl auf Instanzen- und auf Metamodellebene. Allerdings geht der Verkauf des Call-Centers mit einem Verlust an Aufgaben einher, die das Unternehmen ausführt. Es ist auf den ersten Blick ersichtlich, dass es sich bei dem Verkauf einer betrieblichen Organisationseinheit weder um eine Integration noch um eine Vereinigung handelt. Der Unterschied zwischen beiden Beispielen ist, dass in dem einen Beispiel alle betriebswirtschaftlich relevanten Eigenschaften des Artefaktes, dessen Anzahl sich auf Instanzen- und Metamodellebene reduziert, erhalten bleiben, in dem anderen aber nicht. Betriebswirtschaftlich relevante Eigenschaften von verschiedenen Artefakttypen sind beispielsweise: Aufgaben von Organisationseinheiten, Funktionen von Anwendungssystemen, Informationen von Datenbanken oder Endprodukte von Prozessen.

Zusammenfassend lässt sich festhalten: Notwendige Bedingung für das Vorhandensein einer Integration durch Vereinigung ist die Reduktion der Anzahl der Elemente eines feingranularen Metamodells. Eine erste hinreichende Bedingung für das Vorhandensein einer Integration durch Vereinigung ist die Konstanz der Zahl der Elemente auf Instanzenebene. Eine zweite hinreichende Bedingung für das Vorhandensein einer Integration durch Vereinigung ist eine Reduktion der Zahl der Elemente auf Instanzenebene, wobei alle betriebswirtschaftlich relevan-

ten Eigenschaften derjenigen Artefakte erhalten bleiben, um die sich das Modell reduziert.

Rosemann (1999) weist darauf hin, dass eine Integration durch Vereinigung gleichartige Elemente oder Systembestandteile voraussetzt.⁸ Tatsächlich setzt eine Integration durch Vereinigung nach dem hier zugrunde liegenden Verständnis zwei fachlich unterschiedliche Artefakttypen voraus, die sich dennoch strukturell ähneln. Einerseits müssen sich die Artefakttypen voneinander unterscheiden; ansonsten ließen sie sich im Metamodell nicht einzeln abbilden, und die Vereinigung könnte nicht im Metamodell nachvollzogen werden. Andererseits bemerkt Rosemann, dass sich Artefakttypen nicht zu stark voneinander unterscheiden dürfen, um vereinigt werden zu können. So werden beispielsweise Datenbanken mit Datenbanken vereinigt. Es sind allerdings auch andere Fälle denkbar, so z. B. die Integration einer Datenbank in ein Anwendungssystem, das die Datenhaltung selbst organisiert.

2.3.4.3 Beispiel Unternehmensvereinigung

Im Folgenden wird anhand eines komplexen Beispiels eine Integration durch Vereinigung illustriert: einer Unternehmensübernahme oder einem Unternehmenszusammenschluss, kurz einer Unternehmensvereinigung (vgl. dazu ausführlich Unterkapitel 3.2).

Bei einer Unternehmensvereinigung lassen sich zwei grundsätzliche Typen voneinander unterscheiden: Der erste Typ umfasst die Vereinigung zweier Unternehmen, die dieselben Produkte anbieten; der zweite Typ die Vereinigung zweier Unternehmen, die unterschiedliche Produkte anbieten (Penzel u. Pietig 2000). Zwei Unternehmen, die unterschiedliche Produkte anbieten, vereinigen sich, wenn sie sich diversifizieren möchten. Aus dem Finanzbereich lassen sich für diesen Typen beispielhaft die Vereinigung der Deutschen Bank mit Morgan Grenfell im Jahre 1997 oder die Vereinigung der Credit Suisse mit der Winterthur im Jahre 1997 anführen. Wenn sich zwei Unternehmen vereinigen, die dieselben Produkte anbieten, dann ist das Ziel in der Regel, Kosteneinsparungen durch Synergieeffekte zu erzielen. Beispiele für derartige Integrationen im Finanzbereich sind die Vereinigung der Deutschen Bank mit der Bankers Trust im Jahre 1998 oder die Vereinigung der Hypo-Bank mit der Vereinsbank im Jahre 1997 (Penzel u. Pietig 2000).

Die beiden Typen unterscheiden sich fundamental voneinander im Hinblick auf die relevanten Integrationsarchetypen. Wenn gleichartige Produkte angeboten werden, so empfiehlt sich in der Regel eine Vereinheitlichung der Produkte. Nur durch eine Vereinheitlichung lassen sich die erhofften Synergien und die damit verbundenen Kostenreduktionen erzielen. Wenn beide Unternehmen allerdings

⁸ Die Integrationsarchetypen *Alignment* und *Ableitung* unterscheiden sich von dem Integrationsarchetypen *Bindung* dadurch, dass unterschiedliche Artefakttypen integriert werden. Eine solche Unterscheidung innerhalb des Integrationsarchetyps Vereinigung ist nicht notwendig.

unterschiedliche Produkte anbieten, so ist eine Vereinigung auf Produktebene nicht möglich. Dennoch lassen sich auch in diesem Fall etliche Vereinigungen auf Organisationsebene durchführen. Viele Unterstützungsprozesse lassen sich vereinheitlichen, so z. B. Buchhaltungsprozesse, Personalprozesse, IT-Management-Prozesse oder Logistikprozesse.

Soll der Integrationsarchetyp Vereinigung verwendet werden, lassen sich auf Ebene der Informationstechnik drei Fälle unterscheiden. Im ersten Fall nutzt das neue Unternehmen die kompletten Systeme eines der ursprünglichen Unternehmen. Im zweiten Fall nutzt das neue Unternehmen die Systeme eines der beiden ursprünglichen Unternehmen, die Systeme werden aber um Funktionalitäten für den anderen Unternehmensteil ergänzt. In einem dritten Fall nutzt das neue Unternehmen keines der Systeme der ursprünglichen Unternehmen; anstelle dessen werden die Systeme komplett erneuert (Penzel u. Pietig 2000). Es ist rasch ersichtlich, dass der erste Fall in der Regel der kostengünstigste sein wird. Hier reduziert sich das Integrationsproblem durch Vereinigung auf ein einfaches Datenmigrationsproblem. Im zweiten Fall muss ebenfalls eine einfache Datenmigration durchgeführt werden; zusätzlich fallen aber auch Aufwendungen für Implementierungsarbeiten an. Im dritten Fall muss eine doppelte Datenmigration durchgeführt werden; zusätzlich fallen die Kosten für den Kauf oder die Entwicklung der neuen Systeme an.

Eine weitreichende Vereinigung auf Systemebene setzt eine weitreichende Vereinigung der Produkte und der Prozesse voraus. In Fällen, in denen eine solche Vereinigung nicht möglich ist, kann es wirtschaftlich sein, auf die Vorteile einer Vereinigung auf Systemebene zu verzichten. Wenn starke Implementierungsarbeiten oder teure Einkäufe von Standardsoftware notwendig sind, damit ein einzelnes neues System alle Anforderungen beider Unternehmensteile abdeckt, dann kann es wirtschaftlich sein, beide Systeme der ursprünglichen Unternehmen weiterzuverwenden und sie, wo nötig, über den Integrationsarchetypen Bindung miteinander zu verknüpfen.

Zusammenfassend lässt sich festhalten, dass eine Integration durch Vereinigung auf Metamodellebene nachvollzogen werden kann. Um eindeutig zu bestimmen, ob es sich um eine Integration durch Vereinigung handelt, müssen allerdings auch die Instanzen betrachtet werden. Im Beispiel einer Unternehmensvereinigung lassen sich Formen der Vereinigung auf allen Ebenen identifizieren, z. B. der Produkte, der Prozesse und der Informationssysteme.

2.3.5 Empirische Überprüfung der Archetypen

Die hier vorgestellten Integrationsarchetypen sind im Gegensatz zu phänomenologischen Ansätzen aufgrund konzeptioneller Überlegungen abseits von Integrationsphänomenen der Realität entstanden. Es ist darum nicht selbstverständlich, dass die Integrationsarchetypen den „richtigen“ Schnitt aufweisen und geeignet sind, Integrationsphänomene der Realität abzubilden.

Aus diesem Grund wurde in (Aier et al. 2009) eine empirische Analyse durchgeführt, die auf Basis einer Umfrage verschiedene Integrationsaufgaben bzgl. eines zugrunde liegenden latenten Zusammenhangs untersucht. Wenn diese latenten Zusammenhänge den definierten Integrationsarchetypen entsprechen, kann davon ausgegangen werden, dass die Integrationsarchetypen geeignet sind, die Integrationsphänomene der Realität abzubilden. Die zugrundeliegende Umfrage sowie ihre Analyse werden im Folgenden kurz dargestellt.

Die empirische Analyse beruht auf einem Datensatz, welcher mittels Fragebogen im Rahmen einer 2008 durchgeführten Fachtagung zum Schwerpunkt Integration und Architektur erhoben wurde. Teilnehmende dieser Veranstaltung waren insbesondere Fach- und Führungskräfte aus den Bereichen Integrations- und Architekturmanagement. Der Fragebogen führt verschiedene Integrationsaufgaben an und fragt deren Relevanz in Bezug auf die Situation *unternehmensübergreifende Integration* ab. Dabei wird die Relevanz anhand einer 5-stufigen Likert-Skala erfasst. Im Vorfeld wurde der Fragebogen durch Experteninterviews sowie einen Pre-Test hinsichtlich seiner Verständlichkeit und Vollständigkeit geprüft. Die Analyse beruht auf 127 vollständig konsistent ausgefüllten Fragebögen. Die Zusammensetzung aus hauptsächlich deutschsprachigen Teilnehmern schränkt die Interpretation insofern nicht ein, als dass in erster Linie Einschätzungen von Großunternehmen (32 % mit mehr als 5.000 Mitarbeitenden, sowie zusätzlich 28 % mit mehr als 1.000 Mitarbeitenden) vorliegen, deren Integrationsprojekte ebenso international ausgerichtet sind. Neben demografischen Angaben zum Unternehmenskontext wurde auch der Kenntnisstand der Teilnehmenden abgefragt. Hierbei gaben 79 % der Befragten an, über mindestens fortgeschrittene Kenntnisse zur Thematik Integration zu verfügen.

Um einen Zusammenhang zwischen verschiedenen Integrationsaufgaben zu identifizieren, wurde eine Faktorenanalyse durchgeführt. Die Faktorenanalyse extrahiert dafür eine geringe Anzahl wechselseitig unabhängiger Faktoren aus einer Vielzahl von Variablen. Sie unterstellt, dass es möglich ist, für eine Vielzahl von Variablen eines Datensatzes wenige relevante Einflussgrößen (Faktoren) zu ermitteln. Basis für die Faktorenanalyse ist im vorliegenden Fall ein reduzierter Datensatz von 15 Variablen. Nicht berücksichtigt wurden Variablen, die im Fragebogen die Funktion von Kontrollfragen erfüllen.

Im Folgenden wird die Zuordnung von Integrationsaufgaben zu den jeweiligen Faktoren erläutert sowie eine Interpretation der extrahierten Faktoren nachvollzogen. Im Anschluss wird aufgezeigt, inwiefern diese Faktoren den dargestellten Integrationsarchetypen entsprechen. Für die im Fragebogen abgefragten Integrationsaufgaben wurden vier grundlegende Faktoren identifiziert.

Insgesamt laden vier Variablen auf Faktor 1. Die zugeordneten Aufgaben beinhalten jeweils die Erstellung eines zusätzlichen Artefakttyps im Rahmen der Integration. So ist z. B. die Integration von fachlichen Funktionsblöcken und IT-Systemen durch die Definition und wechselseitige Verankerung des zusätzlichen konzeptionellen Konstrukts der Applikation möglich. Ebenso kann die Verknüpfung von Geschäftsprozessen und strategischen Zielstellungen einer Organisation mit der Einführung eines weiteren konzeptionellen Konstrukts „Geschäftsfähig-

keit“ umgesetzt werden. Ausschlaggebend ist für diesen Faktor die wechselseitig unabhängige Entwicklung und Bewirtschaftung der zu integrierenden Artefakte in einer Organisation, sodass eine Integration nicht mit Hilfe einer direkten Verknüpfung zwischen den Artefakttypen erfolgen sollte.

Die drei Integrationsaufgaben, die in Faktor 2 gebündelt werden, betonen insbesondere ein gerichtetes Verständnis von Integration. So wird z. B. die Prozessgestaltung derart durchgeführt, dass in erster Linie Vorgaben und Anforderungen aus dem Leistungs- und Zielsystem einer Organisation zu berücksichtigen sind. Analog richtet sich die Gestaltung der Aufbauorganisation maßgeblich nach den Vorgaben, welche die Ablauforganisation definiert. Charakteristisch für Faktor 2 ist somit die einseitige Abhängigkeit der zu integrierenden Artefakttypen: Bei der Integration leiten sich die Eigenschaften eines Artefakttyps direkt aus den Vorgaben des führenden Artefakttyps ab.

Faktor 3 wird durch vier Variablen beschrieben und umfasst Integrationsaufgaben unter dem Gesichtspunkt, den Schnittstellenwildwuchs innerhalb einer Anwendungslandschaft, aber auch einer Prozesslandkarte zu minimieren. Charakteristisch für die Integration ist dabei, dass die Kopplung zwischen z. B. analytischen und operativen Systemen, welche als m:n-Schnittstelle realisiert ist, durch eine n:1-Kopplung der Systeme abgelöst wird. Es wird somit die Komplexität bei der Verknüpfung von Artefakten gleichen Typs adressiert und z. B. durch Einführung von Hub-and-Spoke-Konzepten eine Entflechtung und leichtere Schnittstellenwartung erzielt.

Die vier Integrationsaufgaben, die mit Hilfe von Faktor 4 gemeinsam erklärt werden können, umfassen sämtlich eine Vereinigung von Artefakttypen. Im Zuge eines Integrationsprojektes werden beispielsweise Kundenstämme zusammengelegt oder IT-Landschaften konsolidiert. Faktor 4 umfasst aber auch die Konsolidierung von systemnahen Entitäten, wie z. B. Applikationen oder der gesamten IT-Infrastruktur. Im Gegensatz zu den bereits beschriebenen Faktoren werden bei den Integrationsaufgaben dieses Faktors keine neuen Verbindungen oder Artefakttypen geschaffen, sondern vorhandene Artefakttypen miteinander vereint.

Die in der empirischen Analyse identifizierten Faktoren können unter Berücksichtigung der jeweils zugehörigen Integrationsaufgaben den vorgeschlagenen Integrationsarchetypen zugeordnet werden. Faktor 1 beschreibt genau solche Integrationsaufgaben, die eine gegenseitige Ausrichtung verschiedener Artefakttypen beinhalten, welche autonomen Änderungen unterworfen sind. Auf Basis eines Metamodells empfiehlt sich hierfür die Einführung zusätzlicher Alignment-Artefakte, um die originären Artefakttypen aneinander auszurichten, ohne ihre unabhängige Entwicklung zu beeinflussen. Faktor 1 beschreibt somit ein Alignment unterschiedlicher Metamodellelemente und wird dem Integrationsarchetyp *Alignment* zugeordnet. Charakteristisch für Faktor 2 ist die einseitige Abhängigkeit der zu integrierenden Artefakttypen. Auf Basis des zugrunde gelegten Metamodellverständnisses wird eine solche direkte, gerichtete Verknüpfung zwischen zwei Artefakttypen durch eine neue Verbindung oder Referenz auf einen anderen Artefakttyp realisiert. Eine derartige Modellierungsvorschrift für die Integrationsaufgabe entspricht dem Integrationsarchetyp *Ableitung*. Bezogen auf die Abbildung in ei-

nem Metamodell steht bei Faktor 3 die Verknüpfung zwischen Instanzen eines Artefakttyps im Vordergrund. Dies erfolgt entweder durch eine Punkt-zu-Punkt-Verbindung oder durch eine, bei einer großen Anzahl von Artefakten effizienteren, Hub-and-Spoke Architektur und wird durch eine rekursive Beziehung von Metamodell-Elementen abgebildet. Der Integrationsarchetyp *Bindung* beschreibt diese Abbildungsvorschrift und kann auf diesen Faktor zurückgeführt werden. Mit Faktor 4 werden Integrationsaufgaben gemeinsam erklärt, die eine Reduktion von Artefakttypen im Metamodell bedingen: Durch die Verschmelzung von zwei verschiedenen Artefakttypen entsteht ein neuer Typ. Faktor 4 entspricht somit dem Integrationsarchetyp *Vereinigung*, der diese Metamodelländerungen beschreibt.

2.4 Übersetzung der Archetypen in Integrationsaufgaben

Grundlage für die Kombination von Methodenfragmenten und damit für situationsspezifische Integrationsmethoden ist das Verständnis der „Elementarteilchen“ der Integration sowie ihres Zusammenwirkens. Sind die Elementarteilchen und ihr Verhalten nicht bekannt, bleibt zur Problemlösung nur „Trial-and-Error“. Deshalb wird für das Integrationsmanagement der Versuch unternommen, ausgehend von „Elementarteilchen“, also den identifizierten Archetypen die konzeptionelle Struktur von Integration (im Sinne eines modellhaften Verständnisses) sowie die Dynamik von Integration (im Sinne von Zusammenhängen zwischen Aktivitäten und Ergebnissen) zu analysieren. Es wird eine Verbindung benötigt, welche die konstruktionsorientierte Perspektive (Bottom-up) der Elementarteilchen mit der anwendungsbezogenen Perspektive der Integrationsprojekte (Top-down) in Beziehung setzt. Ohne diese „Zwischenschicht“ kann die Verortung von Integrationsarchetypen in Integrationsprojekten und vice versa nicht aussagekräftig genug sein, um später die situative Selektion von Methodikfragmenten zu unterstützen.

Im Hinblick auf Bereitstellung einer situativen Methode für das Integrationsmanagement kann unterschieden werden zwischen dem Nutzungs- und dem Konstruktionsprozess (in Anlehnung an Fettke u. Loos 2002). Der Konstrukteur hat die Aufgabe, Fragmente zur Verfügung zu stellen, sodass der Anwender daraus eine situativ angepasste Methode erstellen kann. Vor dem Hintergrund einer konkreten Problemstellung im Rahmen eines Integrationsprojektes kombiniert der Nutzer die für ihn in Frage kommenden Methodenfragmente. Abbildung 2.9 visualisiert diese beiden Prozesse und ihre Verbindung.

Aus Konstruktionssicht wird dafür ein exemplarischer Katalog an Integrationsaufgaben vorgeschlagen. Jede Integrationsaufgabe spezifiziert dabei genau einen Integrationsarchetyp, wobei jeder Integrationsarchetyp durch mehrere Aufgaben repräsentiert werden kann (Beziehung 1:n). Abb. 2.10 visualisiert die Verwendung von Integrationsaufgaben als Bindeglied zwischen der Konstruktionssicht und der Nutzersicht. Integrationsaufgaben sowie ihre möglichen Kombinationen in Form von Aktivitäten sind darüber hinaus der Anknüpfungspunkt zur Bildung von Methodenfragmenten für eine situative Integrationsmethode.

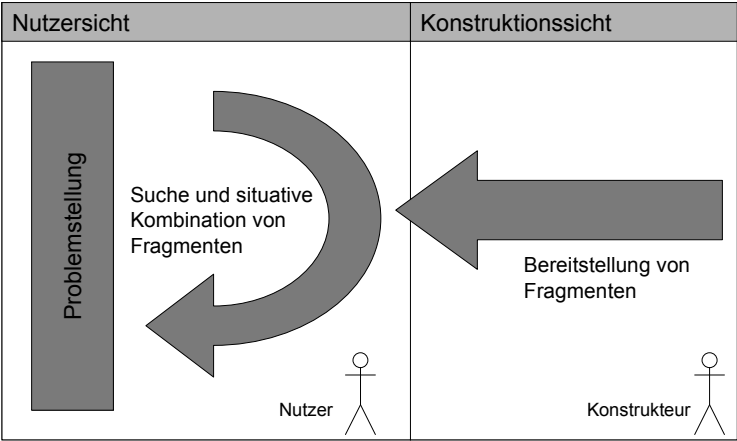


Abb. 2.9. Nutzer- und Konstruktionssicht

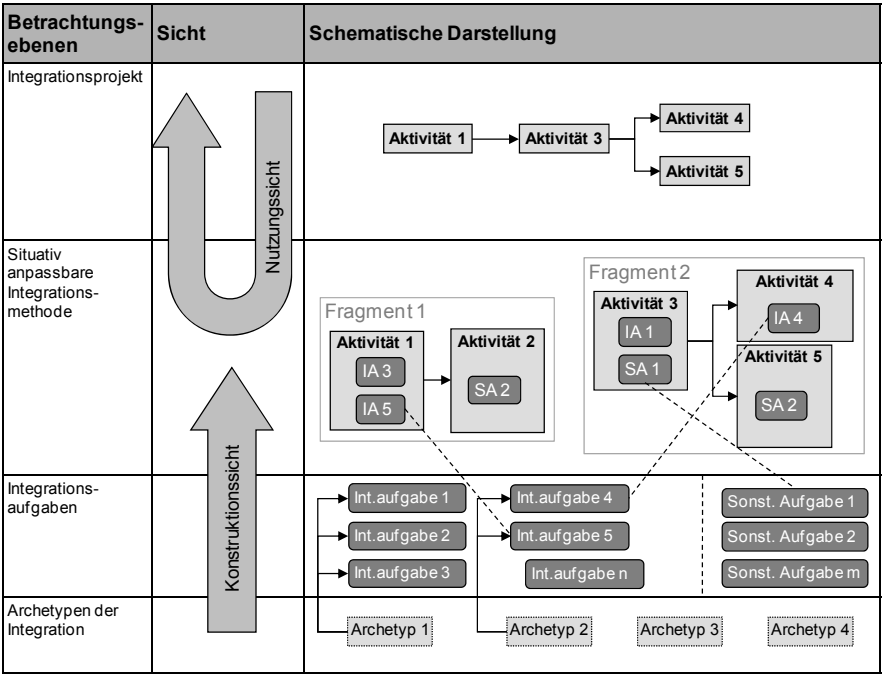


Abb. 2.10. Integrationsaufgaben als Bindeglied

Die Zusammenstellung von Integrationsaufgaben trifft noch keine Aussage über die Relevanz für bestimmte Integrationsprojekte und deren Eigenschaften. Integrationsaufgaben dienen aus Sicht der Konstruktion dazu, die Integrationsarchetypen so zu beschreiben, dass letztendlich eine Übersetzung in Methodenfragmenten-

te ermöglicht wird. Zur Bildung einer situativen Integrationsmethode werden dann Aktivitäten definiert, welche u. a. durch Integrationsaufgaben bzw. Bündel dieser charakterisiert sind. Weitere sonstige Aufgaben, z. B. aus dem Bereich des Projektmanagements, vervollständigen solche Aktivitäten, sind jedoch nicht Gegenstand der Betrachtung in diesem Buch.

Die Konfiguration einer Integrationsmethode geschieht durch die situative Kombination von Methodenfragmenten, welche Aktivitäten zusammenfassen. Einzelne Methodenfragmente können kombiniert und dabei selbst konfiguriert werden, um die benötigte situative Methode zu erstellen. Einzelne Aktivitäten innerhalb der Methodenfragmente können dabei ausgetauscht werden (Abbildung 2.10). Um die Bildung solcher Methodenfragmente zu unterstützen, bedarf es zusätzlich Aussagen zur sachlogischen Kombinierbarkeit und zeitlichen Ablauffolge von Aktivitäten und damit auch von Integrationsaufgaben. Diese Aussagen lassen sich im Detail nur nach Analyse von Integrationsprojekten und den dort getroffenen Entscheidungen zur Durchführung einzelner Integrationsaufgaben treffen. Eine erste Annäherung aus Konstruktionssicht kann jedoch unter bewusster Umgehung spezifischer Projekteigenschaften durch die Angabe generischer Konfigurationsvorschläge für Integrationsaufgaben erfolgen. Exemplarisch stellt dazu Abb. 2.11 Integrationsaufgaben sowie deren sachlogische Kombinierbarkeit und zeitliche Ablauffolge dar.

Zusammenspiel von Integrationsaufgaben	Definieren von Geschäftsfähigkeiten	Definieren von Domänen	Strategische Kennzahlen aus Zielen ableiten	Prozessgestaltung aus Zielsystem ableiten	Aufbauorganisation aus Ablauforganisation ableiten	Geschäftsprozess via Workflowmgmt. verbinden	...	Integrationsaufgabe (n)
Definieren von Geschäftsfähigkeiten	X							
Definieren von Domänen	vor	X						
Strategische Kennzahlen aus Zielen ableiten	vor	vor	X					
Prozessgestaltung aus Zielsystem ableiten			vor	X				
Aufbauorganisation aus Ablauforganisation ableiten				vor	X			
Geschäftsprozess via Workflowmgmt. verbinden		vor		vor		X		
...	X	
Integrationsaufgabe (n)	X

Die Matrix ist von den Zeilen links zu den Spalten rechts zu lesen.

Abb. 2.11. Generische Konfiguration von Integrationsaufgaben aus Konstruktionssicht

So ist es z. B. ohne weiteres denkbar, die Definition von Geschäftsfähigkeiten sowie die Definition von Domänen innerhalb eines Integrationsprojektes durchzuführen. Jedoch wird empfohlen, zunächst Geschäftsfähigkeiten als mögliche fachliche Ausgangsbasis für die Ableitung von Domänen zu definieren.

Insgesamt ermöglicht die Angabe von Integrationsaufgaben die Überführung abstrakter Integrationsarchetypen in Kandidaten für Methodenfragmente. Aus Nutzersicht müssen diese Methodenfragmente und damit auch die hierin abgebildeten Integrationsaufgaben entsprechend der Eigenschaften des vorliegenden Integrationsprojektes kombiniert werden. Dazu ist es notwendig, Integrationssituationen (Kapitel 3) und Fallbeispiele aus der Praxis (Kapitel 4) zu analysieren. Erst die Analyse konkreter Integrationsprojekte, ihrer Auslöser, Ziele und durchgeführten Integrationsaufgaben gibt Aufschluss darüber, welche Integrationsaufgaben zu Aktivitäten zusammengefasst und als Methodenfragmente einer Integrationsmethodik auf bestimmte Weise konfiguriert werden können (Kapitel 5).

Management von Integrationsprojekten
Konzeptionelle Grundlagen und Fallstudien aus
fachlicher und IT-Sicht

Winter, R. (Hrsg.)

2009, XI, 206 S., Hardcover

ISBN: 978-3-540-93772-2