

February 2005

# Nutzen-orientiertes Management einer Service-orientierten Unternehmensarchitektur

Andreas Dietzsch  
*Schweizerische Mobiliar*

Thomas Goetz  
*Schweizerische Mobiliar*

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

---

## Recommended Citation

Dietzsch, Andreas and Goetz, Thomas, "Nutzen-orientiertes Management einer Service-orientierten Unternehmensarchitektur" (2005). *Wirtschaftsinformatik Proceedings 2005*. 80.  
<http://aisel.aisnet.org/wi2005/80>

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

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

ISBN: 3-7908-1574-8

© Physica-Verlag Heidelberg 2005

# Nutzen-orientiertes Management einer Service-orientierten Unternehmensarchitektur

**Andreas Dietzsch, Thomas Goetz**

Schweizerische Mobiliar

*Zusammenfassung: Im Jahr 2001 setzte die Schweizerische Mobiliar eine Unternehmensarchitektur um. Diese basiert auf dem Management von Geschäftsprozessen, Applikationen und der Technologie-Infrastruktur als wesentliche Elemente von Veränderungen. Nach einem zweijährigen Betreiben der Unternehmensarchitektur in dieser Form wurde das Potenzial der Service-Orientierung für den ganzheitlichen Ansatz einer Unternehmensarchitektur untersucht. Die Beschränkung auf einzelne Teilarchitekturen ermöglichte dabei jedoch stets nur eine teilweise Realisierung.*

*Aus diesem Grund wurde ein ganzheitlicher Ansatz zum Management einer Service-orientierten Unternehmensarchitektur entwickelt. Ausgehend von deren spezifischen Artefakten wurden dabei die jeweiligen Stakeholder und deren Anforderungen analysiert. Auf dieser Grundlage wurden Stufen des Service-Entwurfs identifiziert, die mit einem bestimmten Nutzen des Einsatzes von Services korrelieren. Diese Stufen bildeten den Ausgangspunkt für das Formulieren von Heuristiken, die die Planung und Bewertung der Entwicklung von Services unterstützen. Dieser Beitrag stellt diesen Ansatz der Schweizerischen Mobiliar zum Management einer Service-orientierten Unternehmensarchitektur vor.*

*Schlüsselworte: Service Management, Service-orientierter Entwurf, Service Analyse Techniken, Service-Entwurfsprinzipien*

## 1 Hintergrund

Gegründet im Jahr 1826 ist die Schweizerische Mobiliar (Mobiliar) die erste private Versicherung der Schweiz. Sie bietet Produkte für Privatpersonen, Unternehmen und den öffentlichen Sektor an.

Zum Ende des Jahres 2001 erfolgte eine Revision der IT-Strategie der Mobiliar, die u. a. zur Definition und Umsetzung eines Unternehmensarchitektur-Ansatzes führte. Damit sollte ein systematischer Umbau der Applikationslandschaft bei gleichzeitiger Ausrichtung an den aus der Unternehmensentwicklung resultierenden Anforderungen sichergestellt werden.

Die Zusammenführung mehrerer bis dahin selbständiger Tochterunternehmen im Rahmen einer Reorganisation im Jahr 2003 führte zur Notwendigkeit der Integration von Geschäftsprozessen, Applikationslandschaften und Technologie-Infrastrukturen. Auf der Suche nach effizienten Wegen, diese Anforderungen zu erfüllen, rückten die Potenziale einer Service-Architektur ins Zentrum von Untersuchungen. In diesem Kontext war die Frage zu beantworten, um welche Elemente ein bestehendes Architekturrahmenwerk zu erweitern ist, um die Konzeption, Umsetzung und das Management einer Service-orientierten Unternehmensarchitektur zu unterstützen.

Hierzu wurden zunächst für die Teilarchitekturen der Unternehmensarchitektur untersucht, welche Artefakte bei einem Service-orientierten Ansatz zu berücksichtigen sind. Davon ausgehend wurden die dadurch betroffenen Interessengruppen, deren jeweilige Blickwinkel auf eine Service-orientierte Architektur und die daraus resultierenden Anforderungen bestimmt. Basierend auf diesen Analysen und auf ersten Erfahrungen bei der Umsetzung Service-orientierter Konzepte in Projekten erfolgte die Definition von Heuristiken für Planung und Umsetzung der Service-orientierten Architektur.

Wie dieses Vorgehen in der Mobiliar umgesetzt wurde und der Rahmen der Unternehmensarchitektur um Service-orientierte Konzepte erweitert und um Hilfsmittel für das Architekturmanagement ergänzt wurde, stellt dieser Beitrag vor.

## **2 Service-orientierte Erweiterung einer Unternehmensarchitektur**

### **2.1 Der Unternehmensarchitektur-Rahmen**

Grundlage der Unternehmensarchitektur der Mobiliar ist ein im Jahr 2001 entwickelter Unternehmensarchitektur-Rahmen. Dieser entstand auf Grundlage des Federal Enterprise Architecture Framework (vgl. [FEAF]). Die wesentliche Gliederung einer Unternehmensarchitektur wurde von diesem übernommen und entsprechend der Anforderungen der Mobiliar ausgestaltet.<sup>1</sup>

In der ursprünglich entwickelten Variante unterscheidet das Rahmenwerk zwei Perspektiven (Funktionen, Daten), die auf drei Abstraktionsstufen (konzeptionell, logisch und physisch) betrachtet werden. Aus dieser Gliederung leiten sich die fünf Teilarchitekturen (Business-Architektur, Applikations-Architektur, Techn-

---

<sup>1</sup> Einen Überblick über bestehende Unternehmensarchitektur-Rahmen gibt Schekkerman in [Wood03].

ische Architektur, Daten-Architektur und Sicherheits-Architektur) ab (vgl. Abbildung 1).

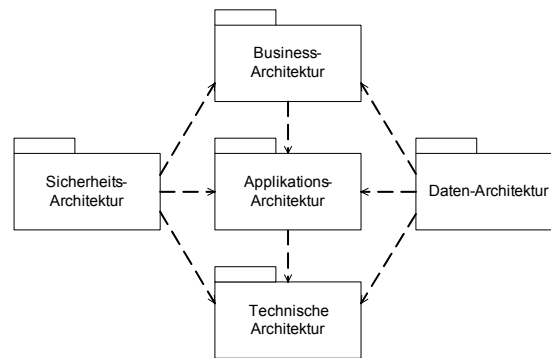


Abbildung 1: Teilarchitekturen des Mobiliar-Architekturrahmens und deren Abhängigkeiten

Die **Business-Architektur** basiert auf Elementen, die im direkten fachlichen Bezug zum Versicherungsgeschäft stehen. Sie subsumiert Artefakte der funktionalen Sicht auf konzeptioneller Ebene, wie z. B. Produkte, Ziele, Geschäftsprozesse, fachliche Aktivitäten und Geschäftsobjekte.

Die Informationssysteme, die die Geschäftsprozesse unterstützen, sind Gegenstand der **Applikations-Architektur**. Sie beschreibt u. a. die Beziehungen zwischen Komponenten und Applikationen als Bestandteile der Applikationslandschaft. Die Applikationsarchitektur sichert eine an den Geschäftsanforderungen ausgerichtete Unterstützung der Geschäftsprozesse durch die Systeme der Applikationslandschaft ab.

Die **Technische Architektur** bildet auf der physischen Ebene die Fähigkeiten von Technologietypen auf die funktionalen Anforderungen der Applikationslandschaft ab. Sie definiert die Prinzipien, die bei der Realisierung der Applikationslandschaft zugrunde zu legen sind. Zu ihrem Betrachtungsbereich gehören Teile der Infrastruktur, wie z. B. Hardware und Betriebssysteme aber auch Entwicklungsumgebungen, Programmiersprachen und die technische Infrastruktur für die Applikationsintegration.

Die Aufgabe der **Daten-Architektur** ist die Gewährleistung einer konsistenten und stabilen Abbildung der fachlichen Informationen von deren konzeptioneller Betrachtung als Geschäftsobjekte bis hin zur physischen Realisierung in einem Datenbanksystem. Sie besitzt damit Schnittstellen zu allen drei Teilarchitekturen mit funktionalem Fokus.

Die Risikobeurteilung und Definition von IT-Schutzziele auf allen Ebenen der Unternehmensarchitektur sowie die ebenenübergreifende Festlegung und Begleitung der Umsetzung von IT-Sicherheitsstandards ist Gegenstand der **Sicherheits-**

**Architektur.** Damit beeinflusst auch die Sicherheits-Architektur alle Teilarchitekturen.

Dieser Architekturrahmen bildete bis zum Jahr 2003 die Basis der Konzeption, Umsetzung und Veränderung der Unternehmensarchitektur. Da zum Zeitpunkt der Definition dieses Architekturrahmens im Jahre 2001 Elemente der Service-Orientierung nicht berücksichtigt wurden war es erforderlich, den Architekturrahmen an die neue Zielsetzung des Aufbaus einer Service-orientierten Architektur anzupassen.

## 2.2 Service-orientierte Erweiterungen des Unternehmensarchitektur-Rahmens

Mit dem Ziel der Umsetzung einer Service-orientierten Architektur wurde es deshalb erforderlich, Konzepte wie z. B. Verträge, Services, Applikationschnittstellen und Komponenten in den Architekturrahmen und damit in den Wirkungsbereich der Unternehmensarchitektur zu integrieren.

Den Ausgangspunkt für diese Integration bildete die Positionierung von Services an der Schnittstelle zwischen Business-Architektur und Applikations-Architektur. Ein Service setzt dabei eine interaktive oder vollständig automatisierte Aktivität eines Geschäftsprozesses mit einem Vertrag<sup>2</sup> über die Eigenschaften der Implementierung dieser Funktionalität in Bezug (vgl. Abbildung 2). Dies schliesst Vereinbarungen über zu realisierende Sicherheitsanforderungen und auszutauschende Geschäftsobjekte ein.

---

<sup>2</sup> Meyer definiert einen Vertrag als formale Vereinbarung zwischen dem Anbieter und dem Nutzer eines Services über die zu erbringende Leistung. Diese umfasst Zusicherungen zu Vorbedingungen, Nachbedingungen, Invarianten und Ausnahmen. Vorbedingungen beschreiben unter welchen Voraussetzungen ein Service erbracht werden kann. Nachbedingungen geben an, welche Leistungen ein Service erbringt. Mit Invarianten werden Aussagen zu Eigenschaften von Geschäftsobjekten getroffen, die durch einen Service nicht verändert werden. Die Verletzung eines mit diesen Elementen definierten Vertrags signalisiert der Anbieter eines Services durch Ausnahmen. (vgl. [Mey97, S. 331ff])

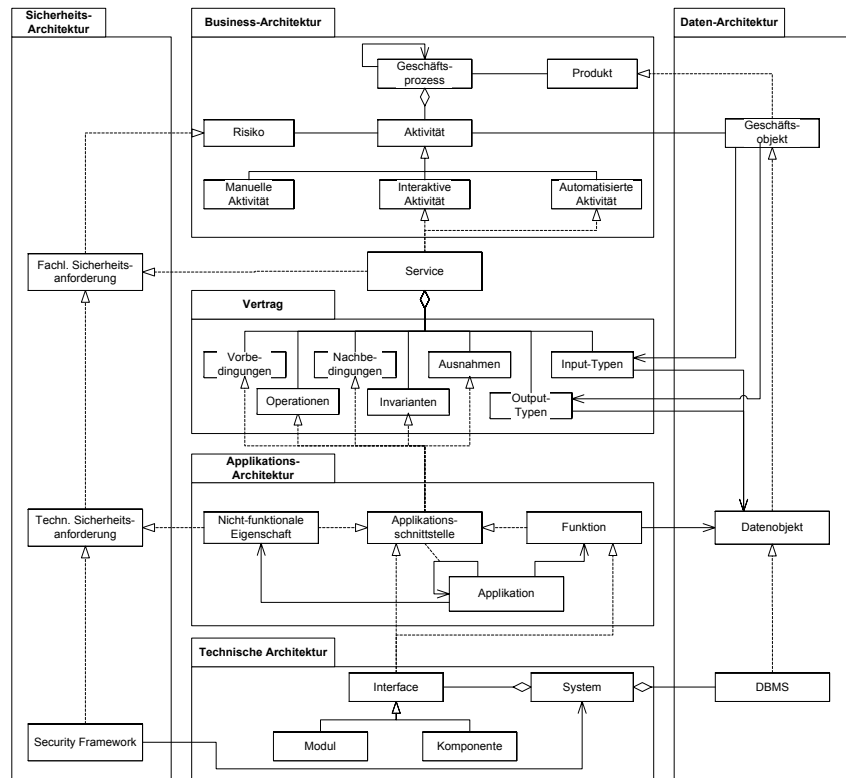


Abbildung 2: Service-orientierter Unternehmensarchitektur-Rahmen der Mobiliar

Die mit einem Service vereinbarten Operationen werden durch die Schnittstelle einer Applikation realisiert. Diese Applikationsschnittstelle kann entweder zur Realisierung der Kommunikation zwischen den Schichten einer Applikation oder zwischen zwei Applikationen genutzt werden. Dabei sind die mit den Datenobjekten der Datenarchitektur sowie den technischen Sicherheitsanforderungen zusätzlich gegebenen Rahmenbedingungen zu berücksichtigen.

Auf der Ebene der technischen Architektur werden anschliessend die definierten Applikationsschnittstellen auf Interfaces von Modulen oder Komponenten abgebildet. Dabei wird u. a. entschieden, welcher Typ Middleware für die Realisierung der Kommunikation zwischen Applikationen genutzt wird.

Mit den dargestellten Erweiterungen war es möglich, ausgehend vom bereits bestehenden Unternehmensarchitektur-Rahmen die Unternehmensarchitektur unter Beibehaltung der Kern-Architekturprozesse weiterzuentwickeln und zu betreiben. Nachfolgend wird beschrieben wie dabei die spezifischen Interessen der Stakeholder einer Service-orientierten Unternehmensarchitektur identifiziert und berücksichtigt wurden.

### 3 Stakeholder einer Enterprise Service Architektur

Die Umsetzung des Service-orientierten Architekturrahmens erfolgte bisher in Projekten mit verschiedenen Zielrichtungen. Diese bestanden u. a. in einer Überarbeitung von Geschäftsprozessen einschliesslich Anpassungen an den unterstützenden Systemen, der Neuentwicklung einer Geschäftsapplikation sowie dem Aufbau einer Integrationsinfrastruktur. Im Verlauf dieser Projekte wurden die folgenden Typen von Stakeholdern erkannt:

- Qualitätsmanager
- Softwareentwickler
- Unternehmensarchitekt
- Anwender
- Eigentümer einer Applikation
- Prozessverantwortlicher

Das generelle Ziel des Qualitätsmanagers ist die Absicherung einer Mindestqualität für Artefakte, die im Verlauf des Lebenszyklus<sup>3</sup> einer Applikation erstellt und verändert werden. Er fokussiert deshalb auf das Formulieren formaler Vorgaben sowie die Überprüfung von deren Einhaltung. Ein Service wird deshalb primär als Konvention der Beschreibung einer Schnittstelle wahrgenommen.

Die Art und Weise, in der die Funktionalität einer Applikation implementiert wird, steht im Zentrum des Interesses des Softwareentwicklers. Ein Service stellt deshalb im Wesentlichen eine Möglichkeit dar, auf eine bestehende Applikationsfunktion unabhängig von deren Lokalisierung zugreifen zu können.

In der Verantwortung des Unternehmensarchitekten liegen die Planung der (Weiter-)Entwicklung von Elementen der Applikationslandschaft und die Unterstützung von deren Umsetzung. In diesem Kontext wird ein Service als Konzept betrachtet, das die Evolution der Applikationslandschaft ermöglicht, ohne Veränderungsbedarf bei den unterstützten Geschäftsprozessen auszulösen.

Der Eigentümer einer Applikation ist für die Festlegung, Vereinbarung und Abrechnung der Service Levels für die Informatik-Dienstleistungen verantwortlich. Er fokussiert aus diesem Grund auf die Funktionalität, die eine Applikation zur Verfügung stellt, deren Verfügbarkeit und die verursachten Kosten. Aus dieser Perspektive wird ein Service deshalb primär als Möglichkeit zur Wiederverwendung und damit zur langfristigen Kostenreduktion betrachtet.<sup>3</sup>

---

<sup>3</sup> Dies wird im Wesentlichen möglich durch die Reduktion von Entwicklungskosten durch die systematische Fokussierung auf die Entwicklung von Funktionalitäten für bisher nicht unterstützte Aktivitäten sowie die Senkung langfristiger Applikationskos-



Analog zu den Aufgaben des Applikationseigentümers verantwortet der Prozessverantwortliche den Entwurf, die Umsetzung, Führung und kontinuierliche Verbesserung eines Geschäftsprozesses. Services werden deshalb primär als Konzept verstanden, dass eine Flexibilisierung der Prozesse durch die Möglichkeit einer einfachen Neuordnung fachlicher Aktivitäten des Geschäftsprozesses ermöglicht.

Die letzte Gruppe von Stakeholdern sind die Anwender der Applikationen. Diese führen Aufgaben innerhalb eines bzw. mehrerer Prozesse aus. Deshalb besteht ihr Interesse in einer einheitlichen Repräsentation von Geschäftsobjekten. Services werden deshalb als Elemente angesehen, die eine homogene Sicht auf Daten und Operationen eines Geschäftsobjektes ermöglichen.

Die dargestellten Aspekte, unter denen die Elemente des Service-orientierten Architekturr Rahmens wahrgenommen werden, fasst Tabelle 1 zusammen.

<b>Rolle</b>	<b>Fokus</b>
Qualitätsmanager	Absicherung einer Mindestqualität von Schnittstellen-Beschreibungen
Softwareentwickler	Reduktion des Entwicklungsaufwands durch die Applikations-übergreifende Nutzung existierender Funktionalität
Unternehmensarchitekt	Risikoverminderung bei der Evolution der Elemente der Applikationslandschaft
Anwender	Durchgängige und effiziente Unterstützung fachlicher Aktivitäten durch IT-Systeme
Applikations-Eigentümer	Langfristige Reduktion der Applikationskosten (Total Cost of Ownership)
Prozessverantwortlicher	Vereinfachung von Veränderungen an den Geschäftsprozessen

Tabelle 1: Stakeholder der Service-orientierten Unternehmensarchitektur

Die identifizierten Typen von Stakeholdern verdeutlichen die unterschiedlichen Erwartungen und Anforderungen, die mit der Umsetzung einer Service-orientierten Unternehmensarchitektur verbunden sind. Die Erfüllung dieser Anforderungen führt zu Lösungen und Services mit unterschiedlicher Granularität und Komplexität [Keen04, S. 46ff].

Aus diesem Grund wurde ein Ansatz erforderlich, der die systematische Bestimmung der für einen Service notwendigen Eigenschaften ermöglicht. Dieser wurde in mit den nachfolgend beschriebenen Nutzenstufen entwickelt.

---

ten durch die Mehrfachverwendung existierender Implementierungen für fachliche Aktivitäten.

## 4 Nutzenstufen des Serviceentwurfs

Um die dargestellten Anforderungen der Stakeholder bei der Architekturplanung berücksichtigen zu können, wurden sechs Stufen des Serviceentwurfs definiert. Grundlage für die Bildung der Stufen waren die spezifischen Anforderungen der Stakeholdertypen, die bei der Serviceentwicklung zu realisierenden Anforderungen sowie der Nutzen, der aus der Erfüllung dieser Anforderungen resultiert.<sup>4</sup> Die Beschreibung jeder Stufe umfasst

- Das **Ziel**, das mit einem Service auf der jeweiligen Stufe verfolgt wird,
- Das **Artefakt** des Unternehmensarchitektur-Rahmens, das im Mittelpunkt der Aktivitäten auf dieser Stufe steht,
- Den zusätzlichen **Freiheitsgrad** bei weiteren Entwicklungsaktivitäten, der aus der Implementierung eines Services auf dieser Stufe resultiert,
- Die **Anforderungen**, die aus der Implementierung eines Services auf dieser Stufe entstehen,
- Die **Einschränkungen**, die aus der Implementierung eines Services auf dieser Stufe resultieren,
- Den **Nutzen**, der durch den entwickelten Service entsteht
- Beispiele für **Entwurfsmuster**, die bei der Serviceentwicklung auf dieser Stufe relevant sind.

### 4.1 Stufe 1: Qualität

**Ziel:** Die Implementierung eines Services auf dieser Stufe erfolgt mit dem Ziel, eine einheitliche Mindestqualität von Schnittstellen-Spezifikationen zu erreichen. Damit werden die formalen Voraussetzungen für die Umsetzung einer Service-orientierten Unternehmensarchitektur geschaffen.

**Artefakt:** Aus diesem Grund ist die formale Vorgabe (Template) zur Spezifikation eines Services das primär betrachtete Artefakt.

---

<sup>4</sup> Woods stellt in [Wood03] ein Stufenmodell vor, das – inspiriert durch das Capability Maturity Model – Conformance Levels einer Service-orientierten Unternehmensarchitektur definiert. Diese beschreiben mögliche Ausprägungen der Qualität einer Architektur und Schritte, um diese zu erreichen. Im Gegensatz dazu beschreiben die hier vorgestellten Stufen primär Anforderungen und den jeweils korrespondierenden Nutzen eines Service-Entwurfs. Damit korrespondiert das von Wood vorgestellte Stufenmodell mit der initialen Nutzenstufe (Stufe 0).

**Freiheitsgrad:** Durch die Beschreibung von Vor- und Nachbedingungen, Invarianten sowie Ausnahmen bei der Ausführung eines Services, kann von einer bekannten Menge zu handhabender Zustände ausgegangen werden. Der Entwurf einer Systemarchitektur auf der Grundlage von Schnittstellen wird damit möglich.

**Anforderungen:** Um diese Stufe zu erreichen, ist es erforderlich, formale Anforderungen an die Beschreibung einer Schnittstelle zu erfüllen. Dies bedeutet, dass mit der Schnittstellendefinition die Festlegung der ausführbaren Operationen, Input- und Outputtypen, Vor- und Nachbedingungen, Invarianten sowie der Ausnahmen erfolgt. Voraussetzung für die Entwicklung von Services auf dieser Stufe ist das Etablieren von Entwicklungs- und Reviewprozessen, die sicherstellen, dass Artefakte konform zu den formalen Anforderungen entwickelt werden.

**Einschränkungen:** Die zusätzlichen formalen Anforderungen dieser Stufe führen zu erhöhten Kosten bei der Entwicklung von Schnittstellen.

**Nutzen:** Die Implementierung von Services auf dieser Stufe führt zu einer homogenen Mindestqualität von Schnittstellen-Beschreibungen. Durch die Entkopplung von Spezifikation und Implementierung wird es möglich, die Spezifikation durch Prüfung der Schnittstellen zu testen.

**Entwurfsmuster:** Keine

## 4.2 Stufe 2: Integration

**Ziel:** Mit der Entwicklung von Services, die die Integration von Applikationen ermöglichen, wird das Ziel verfolgt, die für die Umsetzung von Geschäftsanforderungen in den IT-Systemen benötigte Zeit zu reduzieren. In diesem Kontext bilden Services die Grundlage für die Nutzung von Funktionalität, die durch andere Applikationen implementiert wird [StPi02, S. 426ff].

**Artefakt:** Um die Integration von Applikationen zu unterstützen, fokussiert der Serviceentwurf auf die Realisierung des entfernten Zugriffs auf Funktionen einer Applikation.

**Freiheitsgrad:** Ein auf dieser Stufe implementierter Service reduziert die Abhängigkeit von Plattform- und Applikationsgrenzen bei der Entwicklung und dem Betrieb von Applikationen.

**Anforderungen:** Um einen Service zu implementieren, der Integration ermöglicht ist es erforderlich, dass

- Schnittstellenbeschreibungen den formalen Anforderungen genügen (Stufe 1)
- die Funktionen einer Applikation durch Schnittstellen zur Verfügung gestellt werden, die den Anforderungen an einen Vertrag genügen (vgl. Abschnitt 2)
- Services über Plattformgrenzen hinweg aufgerufen werden können

- weitere technische Anforderungen, wie Fehlerbehandlung, Transaktionsbehandlung und Sicherheit, bei der Schnittstellen-Definition spezifiziert werden<sup>5</sup>

**Einschränkungen:** Um Applikationsfunktionen in einer verteilten Umgebung plattformübergreifend effizient nutzen zu können, sind an den Applikationsfunktionalitäten ausgerichtete Services erforderlich. Damit wird die Granularität eines Services durch die Art und Weise der Implementierung einer Funktion bestimmt. Durch die Ergänzung existierender Applikationen um Schnittstellen entstehen zusätzliche Aufwände bei der Entwicklung.

Die erforderliche Integrations-Infrastruktur verursacht zusätzliche Kosten beim Betrieb der Applikationslandschaft.

Durch die mehrfache Nutzung bestehender Funktionen entstehen Abhängigkeiten zwischen Applikationen. Veränderungen an der bestehenden Applikationslandschaft werden erschwert.

**Nutzen:** Auf bestehende Funktionen kann unabhängig von Plattform- und Applikationsgrenzen zugegriffen werden. Damit kann der Entwicklungsaufwand auf die Bereitstellung neuer Funktionalität fokussiert und die Reaktionszeit auf Geschäftsanforderungen verkürzt werden. Eine durchgängige Unterstützung von Geschäftsprozessen durch die Kopplung von Applikationen wird ermöglicht.

**Entwurfsmuster:** Beispiele für Entwurfsmuster, die im Zusammenhang mit dieser Stufe Relevanz besitzen, sind die *Integration Styles* [HoWo02, S. 39ff] und die Muster der Gruppen *Message Routing* [HoWo02, S. 225ff], *Message Transformation* [HoWo02, S. 327ff] sowie *Message Endpoints* [HoWo02, S. 463ff].

### 4.3 Stufe 3: Evolution

**Ziel:** Mit der Implementierung von Services auf Stufe 3 soll die schrittweise Veränderung der Applikationslandschaft bei einer gleichzeitig stabilen Systemunterstützung für die Geschäftsprozesse ermöglicht werden.

**Artefakt:** Aus diesem Grund steht die Schnittstelle zu einer implementierten Funktion im Fokus der Entwicklungsaktivitäten.

**Freiheitsgrad:** Die Entwicklung von Services auf dieser Stufe schafft eine stabile Menge von Funktionalitäten der Applikationslandschaft, die Veränderungen an der Implementierung einer Schnittstelle ohne Auswirkungen auf die bereitgestellten Leistungen ermöglicht. Damit wird die Entwicklung der Applikationslandschaft von der Entwicklung der Geschäftsprozesse entkoppelt.

**Anforderungen:** Die Entwicklung von Services auf dieser Stufe erfordert es,

---

<sup>5</sup> Anforderungen an die Implementierung verteilter Anwendungen beschreibt u. a. Kischel in [Kisc03, S. 37f].

- die Anforderungen an Services, die Integration ermöglichen, zu erfüllen (Stufe 2)
- Applikationen in Schichten zu gliedern, die wesentliche Aspekte kapseln (z. B. Präsentation, Geschäftslogik und Persistenz)
- Eine Menge von Services der Applikationslandschaft zu definieren, die bei entsprechenden Problemstellungen zwingend zu nutzen sind
- Prozesse zum Management von Performance, Verfügbarkeit und Versionen von Services zu etablieren
- die Spezifikation einer Schnittstelle um die zu nutzenden Testdaten zu ergänzen

**Einschränkungen:** Veränderungen an fachlichen Aktivitäten und Geschäftsprozessen erschweren die Evolution der Applikationslandschaft. Deshalb wirkt sich die Entwicklung auf dieser Stufe nur positiv auf solche Geschäftsanforderungen aus, die expliziert, langfristig stabil und allgemein anerkannt sind. Dies bedeutet ebenfalls, dass die durch eine Applikation bereitgestellte Funktionalität primär durch die Menge existierender Services bestimmt wird.

**Nutzen:** Die Kosten und Risiken der Transformation der Applikationslandschaft werden durch die Entkopplung von Schnittstelle und Implementierung reduziert. Die Konformität einer Implementierung (z. B. die Applikation eines Drittanbieters) zu einer Menge von Services stellt sicher, dass die mit diesen Services in Beziehung stehenden Geschäftsanforderungen erfüllt werden.

**Entwurfsmuster:** Beispiele für Entwurfsmuster, die auf dieser Stufe Relevanz besitzen, sind *Separated Interface* [Fowl02, S. 388ff], *Remote Facade* [Fowl02, S. 476ff] und *Bridge* [Gam+95, S. 151ff].

#### 4.4 Stufe 4: Standardisierung

**Ziel:** Das Ziel der Entwicklung standardisierender Services ist es, fachlich gleiche Aktivitäten stets in der gleichen Form zu repräsentieren. Zu diesem Zweck ist eine Menge von Services erforderlich, die die Daten und Operationen eines Geschäftsobjekts vollständig und einheitlich darstellen.

**Artefakt:** Auf dieser Stufe stehen Geschäftsobjekte im Zentrum von Entwicklungsaktivitäten.

**Freiheitsgrad:** Ein Geschäftsobjekt repräsentiert Operationen auf Informationen, die auf fachlicher Ebene in Beziehung zueinander stehen. Standardisierende Services führen damit zu einer Vereinheitlichung fachlicher Aktivitäten. Damit wird es möglich, die Komplexität einzelner Systeme (bzw. der gesamten Applikationslandschaft) zu reduzieren.

**Anforderungen:** Für die Entwicklung von Services auf dieser Stufe bedarf es,

- der Erfüllung der Anforderungen an Services, die Evolution ermöglichen (Stufe 3)
- der vollständigen Definition der Operationen eines Geschäftsobjektes
- langfristig stabiler Geschäftsanforderungen an ein Geschäftsobjekt
- der Definition einer Sicht, mit der sich die Daten und Operationen des Geschäftsobjektes dem Anwender repräsentieren
- der Definition von Views auf Ebene der konzeptionellen Datenstrukturen, die die durch ein Geschäftsobjekt gekapselten Informationen umfassen

**Einschränkungen:** Die Anforderungen an ein Geschäftsobjekt müssen langfristig stabil sein. Eine einheitliche Sicht auf die Daten und Operationen eines Geschäftsobjektes führt zu einer engen Kopplung zwischen Prozess- und Datenarchitektur. Damit bestimmen die Grenzen einer fachlichen Aktivität wesentlich die Abgrenzung eines Geschäftsobjektes.

**Nutzen:** Durch eine einheitliche Repräsentation von Geschäftsobjekten steigt die Benutzerfreundlichkeit und damit die Effizienz der Anwendung.

**Entwurfsmuster:** Beispiele für Entwurfsmuster auf dieser Stufe sind *Model-View-Controller*, *Facade* und *Observer* [Gam+95, S. 293ff].

## 4.5 Stufe 5: Flexibilität

**Ziel:** Das Ziel der Implementierung von Services auf dieser Stufe ist die Reduktion solcher Abhängigkeiten zwischen den Aktivitäten eines Geschäftsprozesses, die aus durch die Applikationslandschaft gesetzten Rahmenbedingungen resultieren (vgl. [GaLu02, S. 276f]). Die Veränderung von Geschäftsprozessen durch die flexible Orchestrierung fachlicher Aktivitäten und der sie unterstützenden Systeme wird damit erleichtert.

**Artefakt:** Aus diesem Grund werden Services auf dieser Stufe primär auf Basis der Analyse und des Entwurfs von Geschäftsprozessen entwickelt.

**Freiheitsgrad:** Die Implementierung von Services auf dieser Stufe erhöht die Möglichkeiten der Kombination fachlicher Aktivitäten.

**Anforderungen:** Um Services auf dieser Stufe entwickeln zu können, müssen

- die Anforderungen an standardisierende Services erfüllt sein (Stufe 4)
- sequentielle Abhängigkeiten zwischen fachlichen Aktivitäten reduziert werden
- Geschäftsobjekte klar gegeneinander abgegrenzt sein

- die Geschäftsregeln identifiziert werden, die innerhalb eines Prozesses gelten<sup>6</sup>
- der Kontext bestimmt werden, in dem eine fachliche Aktivität realisiert wird

**Einschränkungen:** Die Anpassbarkeit, die die Systemunterstützung sich häufig ändernder Geschäftsprozesse erfordert, führt zu komplexen Applikationen. Gleichzeitig kann die Umsetzung eines Geschäftsprozess-Entwurfs nicht mehr über dessen Implementierung durch eine Applikation erzwungen werden. Damit wird der Erfolg der Umsetzung der Unternehmensarchitektur durch die Fähigkeiten des Prozessmanagements bestimmt.

**Nutzen:** Durch die Entkopplung des Geschäftsprozess-Entwurfs vom Entwurf der unterstützenden Systeme kann die Reaktionszeit auf Marktveränderungen reduziert werden. Die Flexibilität der Applikationslandschaft ermöglicht damit flexible Prozessarchitekturen.

**Entwurfsmuster:** Muster, die die Entwicklung von Services auf dieser Stufe unterstützen sind *Interpreter* [Gam+95, S. 243ff], *Chain of Responsibility* [Gam+95, S. 223ff] und *Serial Process Application*, [Endr04, S. 55f].

#### 4.6 Stufe 6: Wiederverwendbarkeit

**Ziel:** Die Schaffung der Voraussetzungen für die Wiederverwendung von Lösungen ist das Ziel der Implementierung von Services auf dieser Stufe. Damit wird die mehrfache Verwendung fachlicher Lösungen, einschliesslich deren informationstechnischer Unterstützung möglich.

**Artefakt:** Aus diesem Grund werden bei Aktivitäten auf dieser Stufe fachliche Aktivitäten, Schnittstellen, deren Implementierung sowie die zum Betreiben erforderliche technische Infrastruktur betrachtet.

**Freiheitsgrad:** Durch Services, die Wiederverwendung ermöglichen, kann eine systematische Fokussierung auf die Komposition existierender Elemente erfolgen. Das Schwergewicht von Entwicklungsaktivitäten kann damit auf Innovationen gelegt werden.

**Anforderungen:** Voraussetzung der Serviceentwicklung auf dieser Stufe ist

- die Fähigkeit Services zu implementieren, die eine flexible Kombination von Aktivitäten in einem Geschäftsprozess erlauben (Stufe 5)
- das Etablieren der Prozesse des wiederverwendungs-orientierten Entwurfs und der Entwicklung mit Wiederverwendung

---

<sup>6</sup> Business Rules sind in diesem Kontext „... conditions that govern a business event so that it occurs in such a way that is acceptable to the business.“ [vonH01, S. 28]

- die Umsetzung einer Kosten-/Nutzenkalkulation, die die Effekte von Wiederverwendung berücksichtigt (vgl. [Lim98, S. 351ff])
- die Unterstützung des Managements für das Ziel „Wiederverwendung“

**Einschränkungen:** Das systematische Fokussieren auf Wiederverwendung bei Entwurf und Entwicklung führen zu einem erhöhten initialen Entwicklungsaufwand (vgl. [Lim98, S. 118ff]). Durch die Nutzung existierender Komponenten nehmen die im Entwicklungsprozess zu berücksichtigenden Abhängigkeiten zu.

**Nutzen:** Die Bereitstellung wiederverwendbarer Komponenten schafft das Potenzial zur Reduktion von Entwicklungszeiten und –kosten sowie zur Steigerung der Qualität bei Veränderungen an Prozessen und Applikationen.<sup>7</sup>

**Entwurfsmuster:** Bei der Implementierung von Services, die Wiederverwendung ermöglichen, sind Muster wie z. B. *Service Layer* [Fowl02, S. 133ff] und *Robust Artifact* zu berücksichtigen [Ambl00].

Wie die vorgestellten Nutzenstufen als Unterstützung für die Aktivitäten des Architekturmanagements genutzt werden, beschreibt der folgende Abschnitt.

## 5 Heuristiken für den Entwurf von Services

Aufgrund der Erfahrungen, die in den in Abschnitt 3 genannten Projekten gesammelt werden konnten, wurden Heuristiken formuliert, die die Anwendung der definierten Nutzenstufen für die Entscheidungsfindung bei der Architekturplanung vereinfachen.

Mit Bezug zum angestrebten Nutzen werden Heuristiken zur Steigerung der Effizienz der Geschäftsprozesse, zur Verbesserung der Flexibilität der Applikationslandschaft sowie zur Handhabung abhängiger Applikationen unterschieden. Eine Voraussetzung für die Anwendung dieser Heuristiken ist die Dokumentation von Ist- und Ziel-Architektur.

In der Mobiliar werden dazu auf einer hoch aggregierten Stufe Versicherungsprodukte, Geschäftsprozesse, Applikationen sowie die Plattformen abgebildet, auf denen diese betrieben werden (vgl. Abbildung 3).

---

<sup>7</sup> Eine umfassende Betrachtung von Kosten und Nutzen von Wiederverwendung führt Lim in (vgl. [Lim98, S. 101ff]).



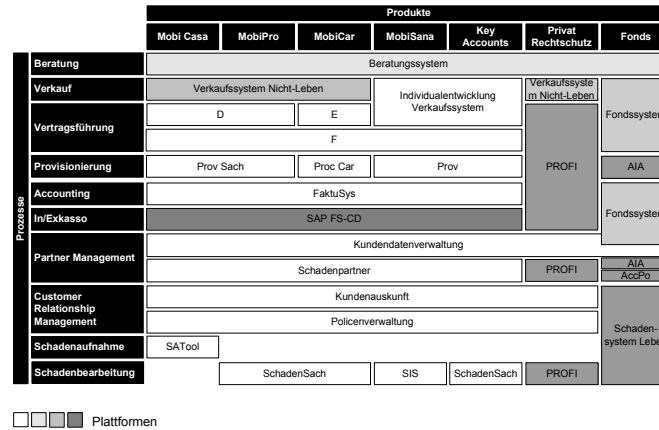


Abbildung 3: Exemplarische Abbildung der Ist-Architektur der Mobiliar

Die Analyse der Elemente dieser Architekturdokumentation sowie deren Entwicklung ist der Ausgangspunkt bei der Nutzung der folgenden Heuristiken.

### 5.1 Effizienz der Geschäftsprozesse steigern

Für die Produkte im Breitengeschäft sind die Prozesse der Produktentwicklung sowie der Bestandesverwaltung in der Mobiliar standardisiert und automatisiert. Im Gegensatz dazu führt ein Schadenfall mit seinen individuellen Merkmalen zu spezifischen Abläufen der Bearbeitung. Während dabei die Phasen des Schadenmanagement-Prozesses noch definiert werden können, ist dies für die Reihenfolge der Abarbeitung einzelner fachlicher Aktivitäten nicht allgemeingültig möglich.

Bei der Analyse potenzieller Anwendungsgebiete Service-orientierter Lösungen wurde deutlich, dass die Unterstützung standardisierter Prozesse nur teilweise das Potenzial einer Service-orientierten Unternehmensarchitektur realisiert (vgl. auch [Pic+03, S. 293f]). Die Standardisierung von Reihenfolgebeziehungen zwischen fachlichen Aktivitäten führt zur Entwicklung von Applikationen, die nur vorhersehbare und repetitive Prozesse unterstützen [JoJa00, S. 220]. Damit bleibt die Möglichkeit der flexiblen Kombination fachlicher Aktivitäten ungenutzt. Die Erfahrung, dass nicht oder nur teilweise standardisierbare Prozesse ein hohes Nutzenpotenzial für die Verwendung von Services bieten, fasst Heuristik 1 zusammen.<sup>8</sup>

<sup>8</sup> Im Zusammenhang mit der Entwicklung von Services für nicht standardisierbare Prozesse sind die Aussagen von Heuristik 4 zu berücksichtigen.

<b>H1</b>	Nicht-standardisierbarer Prozess
<b>Wenn</b>	es nicht möglich ist, einen Prozess zu definieren, der 80% der Geschäftsvorfälle abdeckt,
<b>Dann</b>	sollten Services entwickelt werden, die die flexible Orchestrierung fachlicher Aktivitäten ermöglichen, ohne die Notwendigkeit zur Anpassung von Applikationen auszulösen (Stufe 5).

Die Kernprozesse nutzen die Leistungen mehrerer unterstützender Prozesse, wie z. B. des Dokumenten-Managements, der Archivierung, des Partner- und Customer Relationship Managements. In der evolutionär entstandenen Applikationslandschaft der Mobiliar werden diese Prozesse z. T. nur partiell bzw. durch mehrere redundante Applikationen mit unterschiedlichen Formen der Repräsentation unterstützt. Aus Sicht des Benutzers wird dadurch die Ergonomie der Applikationen reduziert. Um in diesem Kontext zu einer einheitlichen Sicht auf ein Geschäftsobjekt zu gelangen, wurde Heuristik 2 formuliert.

<b>H2</b>	Verschiedene Applikationen für einen Unterstützungsprozess
<b>Wenn</b>	ein Unterstützungsprozess durch mehr als eine Applikation unterstützt wird
<b>Dann</b>	sind Services zu entwickeln, die eine standardisierte Sicht auf ein Geschäftsobjekt realisieren (Stufe 4).

In der Mobiliar wurden Services für die standardisierte Repräsentation von Geschäftsobjekten für die Applikationen zur Unterstützung des Partner- und Customer Relationship Management Prozesse implementiert (vgl. Abbildung 3).

## 5.2 Flexibilität der Applikationslandschaft steigern

Die zweite Gruppe umfasst Heuristiken, bei denen das Ziel im Vordergrund steht, bei der Transformation der Applikationslandschaft Veränderungen lokal zu begrenzen und damit das mit diesem Umbau verbundene Risiko zu reduzieren.

Veränderungen an Produkten und/oder Geschäftsprozessen führen zu Anforderungen an die Anpassung unterstützender Applikationen. Die Produkterneuerungs-Strategie der Mobiliar geht von einem zwei bis -vierjährigen Rhythmus bei der Weiterentwicklung eines einzelnen Versicherungsprodukts aus. Dies hat zur Folge, dass innerhalb des strategischen Planungshorizonts von 5 Jahren bis zu drei neue Produkte entwickelt werden, wobei Änderungen an bestehenden Produkten dabei nicht berücksichtigt sind. Aus diesem Grund wurde entschieden Applikationen zu implementieren, die den Computer-unterstützten Entwurf von Versicherungsprodukten sowie produktunabhängige Verkaufs- und Bestandesverwaltungs-Prozesse unterstützen.

Durch die Standardisierung der Bausteintypen eines Versicherungsprodukts und der darauf operierenden Prozesse wurde der Entwurf von Services möglich, die Wiederverwendung unterstützen. Es hat sich jedoch gezeigt, dass dieser Grad an

Standardisierung den Einsatz interpretierender Systeme ermöglicht, die eine Service-orientierte Implementierung aus fachlicher Sicht überflüssig werden lassen. Diese Erfahrungen fast Heuristik 3 zusammen.

<b>H3</b>	Häufige Änderungen eines Produktes
<b>Wenn</b>	zwei oder mehr Änderungen eines Produkts im Verlauf der nächsten fünf Jahre geplant sind
<b>Dann</b>	sollten Services entwickelt werden, die Wiederverwendung ermöglichen (Stufe 5). Werden das Produkt und die Prozesse auf einem generischen System umgesetzt, das zur Laufzeit entsprechende Definitionen interpretiert, so ist die Implementierung von Services nicht erforderlich.

Heuristik 1 gibt auf der Ebene der Business Architektur Empfehlungen für das Vorgehen bei nicht-standardisierbaren Prozessen. Der Bedarf nach der Unterstützung einer flexiblen Orchestrierung fachlicher Aktivitäten resultiert in der Anforderung nach Services, die Wiederverwendung ermöglichen.

Im Kontext von Geschäftsprozessen, bei denen in einer strategischen Perspektive keine Veränderungen vorgesehen sind, besteht eine solche Anforderung nicht. Ein Beispiel hierfür sind die Systeme, die die in Abschnitt 5.1 genannten Prozesse der Produktentwicklung sowie der Bestandesverwaltung unterstützen. Im Fall solcher stabiler Prozesse liegt der Schwerpunkt des Serviceentwurfs auf dem Ermöglichen von Technologieerneuerungen bei einer gleichzeitig stabilen Systemunterstützung der Geschäftsprozesse, wie dies Heuristik 4 beschreibt.

<b>H4</b>	Langfristig stabile Geschäftsprozesse
<b>Wenn</b>	innerhalb der nächsten fünf Jahre keine Veränderungen an einem Geschäftsprozess vorgesehen sind
<b>Dann</b>	sollten Services implementiert werden, die die Evolution der Applikationslandschaft bei gleichzeitig stabiler Unterstützung des Prozesses ermöglichen (Stufe 3).

Wie bereits dargestellt, führen die verschiedenen Lebenszyklen der Elemente der Unternehmensarchitektur zu einem kontinuierlichen Veränderungsprozess. Dabei sind die Lebenszyklen der Produkte und Prozesse mehrheitlich kürzer als die der unterstützenden Applikationen. Da auch die Phasen dieser Veränderungen nicht synchron sind, entsteht der Bedarf zur Erneuerung von Produkten, Prozessen, Applikationen oder Plattformen gewöhnlich zu verschiedenen Zeiten. Um die Abhängigkeiten zwischen diesen Elementen trotzdem handhaben zu können, sind Veränderungen lokal zu begrenzen. Als Folge davon muss beim Entwurf von Services eine Konzentration auf stabile Elemente der Unternehmensarchitektur erfolgen, wie dies in Heuristik 5 beschrieben wird.

<b>H5</b>	Stabile Applikationen innerhalb von Veränderungen
<b>Wenn</b>	Applikationen bzw. Komponenten existieren, die unabhängig von Produkten und den Kerngeschäftsprozessen sind UND Änderungen an mehreren in Bezug stehenden Applikationen geplant sind
<b>Dann</b>	Ist die stabile Applikation bzw. Komponente durch Services zu kapseln, die Wiederverwendung ermöglichen (Stufe 6).

### 5.3 Abhängige Applikationen handhaben

Die aktuelle Applikationslandschaft der Mobiliar ist im Verlauf der vergangenen 20 Jahre gewachsen. In diesem Zeitraum wurden die Organisationsstruktur und die Geschäftsprozesse mehrfach geändert. Die wesentliche Änderung war dabei die Umsetzung einer prozessorientierten Matrixorganisation im Jahr 1999. Vor diesem Hintergrund wird deutlich, dass der typische Lebenszyklus einer Geschäftsapplikation, der zwischen 5 und 15 Jahren beträgt, die optimale Ausrichtung auf spezifische organisatorische Strukturen oder Prozesse behindert. Dies wird durch die zunehmende und tiefere Integration von Geschäftspartnern verstärkt.

Die Ausführung solcher integrierter Prozesse erfordert Services, die den Zugriff auf Funktionalitäten unabhängig von Applikations- und Plattformgrenzen ermöglichen. Wie in Heuristik 6 beschrieben bedeutet dies, dass Services zu implementieren sind, die eine externe Schnittstelle zu den Funktionalitäten einer Applikation bilden.

<b>H6</b>	Ein Prozess, mehrere Applikationen
<b>Wenn</b>	ein produktunabhängiger Prozess existiert, der Applikationen ordnet, die verschiedene Phasen des Prozesses unterstützen
<b>Dann</b>	sind Services zu entwickeln, die die Integration von Applikationen unterstützen (Stufe 2).

Neben der historisch gewachsenen Applikationslandschaft ist die Unterstützung individueller Produkte ein weiterer Grund für die Notwendigkeit, mehrere Applikationen zur Prozessunterstützung zu orchestrieren. In der Mobiliar ist die Unterstützung der Produkte für das Grosskunden- und Spezialgeschäft hierfür ein Beispiel.

Aufgrund spezifischer Anforderungen, die aus der Individualisierung der Produkte resultieren, können die Produktentwicklungs- und Bestandsführungssysteme der Produkte des Breitengeschäfts nicht genutzt werden. Deshalb wurden die Geschäftsprozesse und mit ihnen die unterstützenden Applikationen um die Individualprodukte gruppiert.

Um in diesem Umfeld eine durchgängige Unterstützung der Geschäftsprozesse über alle Applikationen hinweg zu realisieren, sind Services zu entwickeln, die die Integration ermöglichen. Heuristik 7 fasst dies zusammen.

<b>H7</b>	Mehrere Applikationen für ein Individualprodukt
<b>Wenn</b>	die Prozesse, die auf einem Produkt operieren, durch mehrere Applikationen unterstützt werden
<b>Dann</b>	sind Services zu entwickeln, die Integration von Applikationen unterstützen (Stufe 2).

## 6 Zusammenfassung

Die Entwicklung des Unternehmensarchitektur-Rahmens der Mobiliar zeigt, dass die Erweiterung eines etablierten Architekturrahmens um Service-orientierte Konzepte möglich und sinnvoll ist. Um unter wirtschaftlichen Kriterien den Aufwand des Aufbaus und Betriebens einer solchen Architektur zu rechtfertigen, ist jedoch eine differenzierte Betrachtung der Anforderungen und des Nutzens eines Serviceentwurfs erforderlich.

Gemeinsam mit den vorgestellten Heuristiken und einer adäquaten Form der Dokumentation der Prozess- und Applikationslandschaft bilden die eingeführten Stufen des Serviceentwurfs ein wirkungsvolles, technologieunabhängiges Instrument zur Planung von Veränderungen an der Unternehmensarchitektur. Dabei werden bekannte Muster vor dem Hintergrund einer Service-orientierten Betrachtung neu organisiert.

Weitere Arbeiten in der Mobiliar fokussieren auf die Evaluation und Verbesserung des vorgestellten Ansatzes als Instrument des Architekturmanagements. Dabei wird insbesondere die Untersuchung von Trade-offs zwischen den mit verschiedenen Heuristiken vorgeschlagenen Lösungen, wie z. B. Prozessstandardisierung im Gegensatz zu flexiblen Applikationen, im Zentrum stehen.

## Literatur

- [Ambl00] Ambler, S.: Reuse patterns and antipatterns. <http://www.umlchina.com/CBD/reuse.htm>, 2000, Abruf am 2003-11-28.
- [Endr04] Endrei, M: Service-Oriented Architecture and Web Services. <http://www.redbooks.ibm.com/redbooks/pdfs/sg246303.pdf>, 2004, Abruf am 2004-06-08.
- [FEAF] o. V.: Overview of the Federal Enterprise Architecture <http://www.feapmo.gov/resources/FEAOverview.pdf>, 2001, Abruf am 2001-11-12.

- [Fowl02] Fowler, M: Patterns of Enterprise Application Architecture. Addison-Wesley: Boston et al., 2002.
- [GaLu02] Ganesarajah, D.; Lupu, E.: Workflow-based composition of web-services: a business model or programming paradigm. In: 6th International Enterprise Distributed Object Computing Conference (EDOC 2002), 17-20 September 2002, Lausanne, Switzerland, Proceedings. IEEE Computer Society Press: New York et al., 2002, S. 273-284.
- [Gam<sup>+</sup>95] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: Design Patterns - Elements of Reusable Object-Oriented Software. Addison-Wesley: Reading, 1995.
- [HoWo02] Hohpe, G.; Woolf, B.: Enterprise Integration Patterns. Addison-Wesley: Boston et al., 2002.
- [JoJa00] Johannesson, P.; Wangler, B.; Jayaweera, P.: Application and Process Integration - Concepts, Issues, and Research Directions. In: Wangler, B; Bergman, L. (Hrsg.): Advanced Information Systems Engineering, 12th International Conference CAiSE 2000. Proceedings. Springer: Heidelberg et al., 2000, S. 212-231.
- [Keen04] Keen, M.: Patterns: Implementing an SOA using an Enterprise Service Bus, <http://www.redbooks.ibm.com/redpieces/pdfs/sg246346.pdf>, 2004, Abruf am 2004-06-08.
- [Kisc03] Kischel, S.: Der „Enterprise Service Bus“: Implementierung einer Service-orientierten Architektur. In: ObjektSpektrum, 2/2003, S. 36-39.
- [Lim98] Lim, W. C.: Managing Software Reuse. Prentice Hall: Upper Saddle River, 1998.
- [Meye97] Meyer, B.: Object-oriented Software Construction. 2. Auflage, Prentice Hall: Upper Saddle River, 1997.
- [Pic<sup>+</sup>03] Piccinelli, G.; Finkelstein, A.; Williams, S. L.: Service-oriented workflows: the dysco framework. In: Chroust, G. (Hrsg.) Proceedings of the 29th Euromicro Conference. IEEE Computer Society Press: New York et al., 2003, S. 291- 297.
- [Sche04] Schekkerman, J.: How to survive in the jungle of Enterprise Architecture Frameworks. Trafford: Victoria, 2004.
- [StPi02] Stearns, M.; Piccinelli, C.: Managing Interaction Concerns in Web Service Systems. In: 22nd International Conference on Distributed Computing Systems, Workshops (ICDCSW '02) July 2-5, 2002, Vienna, Austria, Proceedings. IEEE Computer Society Press: New York et al., 2002, S. 424-429.
- [vonH01] von Halle, B.: Business Rules Applied: Building Better Systems Using the Business Rule Approach. John Wiley & Sons: New York et al., 2002.
- [Wood03] Woods. D.: Enterprise Services Architecture. O'Reilly: Sebastopol, 2003.