



Institut für Wirtschaftsinformatik

Universität St.Gallen



# GI-Arbeitskreis EA Frühjahrskonferenz 2003



Enterprise Architecture und  
Enterprise Application Integration (EAI)

***Vorabversion***

28. Mai 2003  
Weiterbildungszentrum (WBZ)  
der Universität St. Gallen

<http://aim.iwi.unisg.ch/veranstaltungen/akea.php>

**Veranstalter:**

Competence Center Application Integration Management (CC AIM)  
Institut für Wirtschaftsinformatik  
der Universität St. Gallen (IWI-HSG)  
Müller-Friedberg-Strasse 8, CH-9000 St. Gallen  
<http://aim.iwi.unisg.ch>

## Vorwort

Im Mittelpunkt auch früherer Modewellen der Wirtschaftsinformatik stand regelmäßig der Anspruch, die verschiedenen Komponenten der Applikationslandschaft miteinander zu integrieren, Redundanzen wie Brüche zu beseitigen und eine Architektur zu schaffen, die unternehmerische Entscheidungen erleichtert, anstatt sie zu verhindern bzw. zu erschweren. Mittel zur Erreichung eines „optimalen Integrationsgrades“ sind die Fokussierung auf die Geschäftsprozesse, die Einführung integrierter Standardanwendungssoftware, die Integration durch Portale oder die Nutzung immer neuer Varianten von Middleware-Technologien. Allerdings ist optimale Integration ein „Moving Target“, weil sich durch neue unternehmerische Konzepte (z.B. verringerte Fertigungstiefe), aber nicht zuletzt auch durch die immer raschere Einführung neuer Technologien stets neue Gestaltungs- und Anpassungsbedarfe ergeben. Erst in neuester Zeit zeichnet sich ab, dass bestimmte Applikationsblöcke identifiziert werden können, die durch spezielle Integrationstechnologien handhabbar miteinander so verbunden werden können, dass Flexibilitätsspielräume auch hinsichtlich zukünftiger Änderungen entstehen:

- Die u.a. auf EAI-Technologien basierende "Business Integration-Infrastruktur" für die betriebliche Applikationsintegration
- Die Data Warehouse-Infrastruktur für die Versorgung analyseorientierter Applikationen
- Die "Business Collaboration-Infrastruktur" für die überbetriebliche Applikationsintegration

Keine dieser Infrastrukturschichten schafft isoliert und dauerhaft eine optimale Integration. Wohl aber bieten sie zusammen mit einer geeigneten Umsetzung von Architekturmanagement-Prozessen die Chance, die Schnittstellenproblematik langfristig überschaubar und handhabbar zu machen. Auch diese Infrastrukturschichten bedürfen einer fortlaufenden Anpassung und Änderung. Jedoch sind diese Herausforderungen angesichts der technologischen Fortschritte der letzten Jahre, z.B. bei Portalen oder Enterprise Application Integration (EAI), heute leichter und realistischer zu bewältigen. Aber auch bei einer verbesserten technologischen Basis bleibt die grösste Problematik bestehen: Ohne eine durchgängige Modellierung über die Ebenen der Geschäftsmodelle, Prozesse, Applikationen und schliesslich der zugrundeliegenden Informationssysteme werden Projekte scheitern. Eine nachhaltig betreibbare betriebliche Applikationslandschaft zu schaffen bleibt eine konzeptionelle Herausforderung für jedes Unternehmen, der nur auf methodisch sauberer Grundlage erfolgreich begegnet werden kann.

Unter dem Titel „Enterprise Architecture und Enterprise Application Integration (EAI)“ wird in dem vorliegenden Tagungsband ein Ausschnitt aus der aktuellen Diskussion in Forschung und Praxis wiedergegeben. Mit einer Mischung aus wissenschaftlichen Beiträgen und Praxisberichten wendet sich der Band an Wissenschaftler und an Praktiker sowie an Studierende mit entsprechenden Studienschwerpunkten. Die Beiträge wurden anlässlich der Veranstaltung „Frühjahrskonferenz 2003 des GI-Arbeitskreises Enterprise Architecture“ am 28. Mai 2003 verfasst und im Weiterbildungszentrum der Universität St. Gallen präsentiert.

Tagungen und auch Bücher sind immer das Ergebnis eines ideell motivierten Engagements, für das die Herausgeber insbesondere den *Mitgliedern des Programmkomitees*, die alle eingegangenen Beiträge begutachteten, und den Autoren und Referenten herzlich danken. Ein ganz wesentlicher Baustein des Erfolgs einer Tagung sind selbstverständlich auch die *Teilnehmer*, die durch ihre Anwesenheit und Diskussionsbereitschaft den Austausch von Wissen und Erfahrungen erst möglich machen.

Ein herzlicher Dank gebührt auch den Personen, die an der Organisation der Tagung sowie an der Erstellung dieses Bandes unmittelbar mitgewirkt haben. Besonders zu erwähnen ist Herr *MScIS. Felix Wortmann*, Mitarbeiter am Institut für Wirtschaftsinformatik, sowie den studentischen Hilfskräften Andrea Müller und Mario Schoeb.

St.Gallen, im Mai 2003

Joachim Schelp      Robert Winter

# Inhaltsverzeichnis

## Beiträge

*Dr. Michael Rohloff*

Ein Ordnungsrahmen zur Beschreibung von Unternehmensarchitekturen....2

*Alexander Schwinn*

Formen der Datenintegration ..... 10

*Prof. Dr. Alfred Winter / Dr. Birgit Brigl / Thomas Wendt*

3LGM<sup>2</sup>: Methode und Werkzeug zur Modellierung von  
Unternehmensarchitekturen im Krankenhaus Systems ..... 20

*Dr. Richard Lenz*

Integration und Evolution ablaufunterstützender Informationssysteme  
im Krankenhaus..... 34

*Dr. Michael Rosemann / Marit Schallert*

Issues in the Design of Enterprise Architectures:  
Insights from a Case Study ..... 42

*Dr. Andreas Dietzsch*

Positionierung eines Unternehmensarchitektur-Ansatzes ..... 50

## **Programmkomitee**

Prof. Dr. U. Frank, Universität Koblenz-Landau

Dr. S. Gröger, Deutsche Bank AG

Dr. H. Günzel, IBM

Prof. Dr. H. Krallmann, TU Berlin

K. Kreplin, SAP AG

Prof. Dr. P. Loos, Universität Mainz

Dr. M. Reichert, Universität Ulm

Dr. M. Rohloff, Siemens AG

Dr. J. Schelp, Universität St. Gallen

Prof. Dr. K. Turowski, Universität Augsburg

Prof. Dr. R. Winter, Universität St. Gallen

## **Bausteine einer Unternehmensarchitektur: Die Integration von Geschäfts- und IT-Architektur**

**Dr. Michael Rohloff, Siemens AG, Corporate Information and Operations, CIO  
Strategy, Planning and Controlling  
michael.rohloff@siemens.com<sup>1</sup>**

***Zusammenfassung:** Der Aufsatz stellt einen Ordnungsrahmen für Unternehmensarchitekturen in der Übersicht vor. Dieser setzt sich aus den Teilarchitekturen Geschäftsarchitektur, Mehrwertdienste und Infrastruktur zusammen. Wesentlicher Bezugspunkt ist eine Referenzarchitektur für Geschäftsprozesse, die alle wesentlichen Prozesse eines Unternehmens erfaßt. Bebauungspläne bilden die Ausgangsbasis für die systematische Analyse der Geschäftsprozesse und ihrer Unterstützung durch Anwendungssysteme. Mit dem Applikation Navigator wird ein methodisches Vorgehen für die Bebauung der IuK-Landschaft zur Verfügung gestellt. Es hilft bei der systematischen Erfassung von Synergien und Integrationspotentialen und unterstützt die Ausrichtung von Prozessen und Applikationen an der Geschäftsstrategie.*

### **1 Bausteine einer Unternehmensarchitektur**

Gegenstand der Architektur ist die Strukturierung des Unternehmens insgesamt, mit allen bestimmenden Komponenten, ihren Schnittstellen und Beziehungen untereinander. Ausgangsbasis ist die Unternehmensstrategie, welche die Rahmenbedingungen und Zielrichtung für die Gestaltung der Unternehmensarchitektur vorgibt. Die Architektur umfaßt sowohl die Gestaltung der Organisation wie auch die der IuK-Ressourcen. Die Gestaltung der Organisation und ihrer Geschäftsprozesse ist Gegenstand der Geschäftsarchitektur. Die IuK-Ressourcen beschreiben die gesamte IuK-Landschaft eines Unternehmens und setzen sich aus den Mehrwertdiensten, insbesondere aus den Anwendungen und der erforderlichen Infrastruktur zusammen.

Die hier vorgestellte Referenzarchitektur setzt sich demnach aus den drei Teilarchitekturen

- Geschäftsarchitektur,
- Mehrwertdienste,
- Infrastrukturdienste

Zusammen (vergleiche z.B. [1]). Der Architekturrahmen mit den drei Teilarchitekturen sowie die jeweiligen Architekturbausteine sind in der Übersicht in Abbildung 1 dargestellt.

Die *Geschäftsarchitektur* beschreibt die grundlegenden Strukturen und Anforderungen des Geschäfts und wird bestimmt durch die Unternehmensziele. Sie setzt sich aus den folgenden Architekturbausteinen zusammen:

Das *Geschäftsmodell* beschreibt in aggregierter Form die jeweilige Leistungserstellung von Produkten und Dienstleistungen am Markt durch Nutzung spezifischer Kombinationen von Ressourcen und Informationen.

---

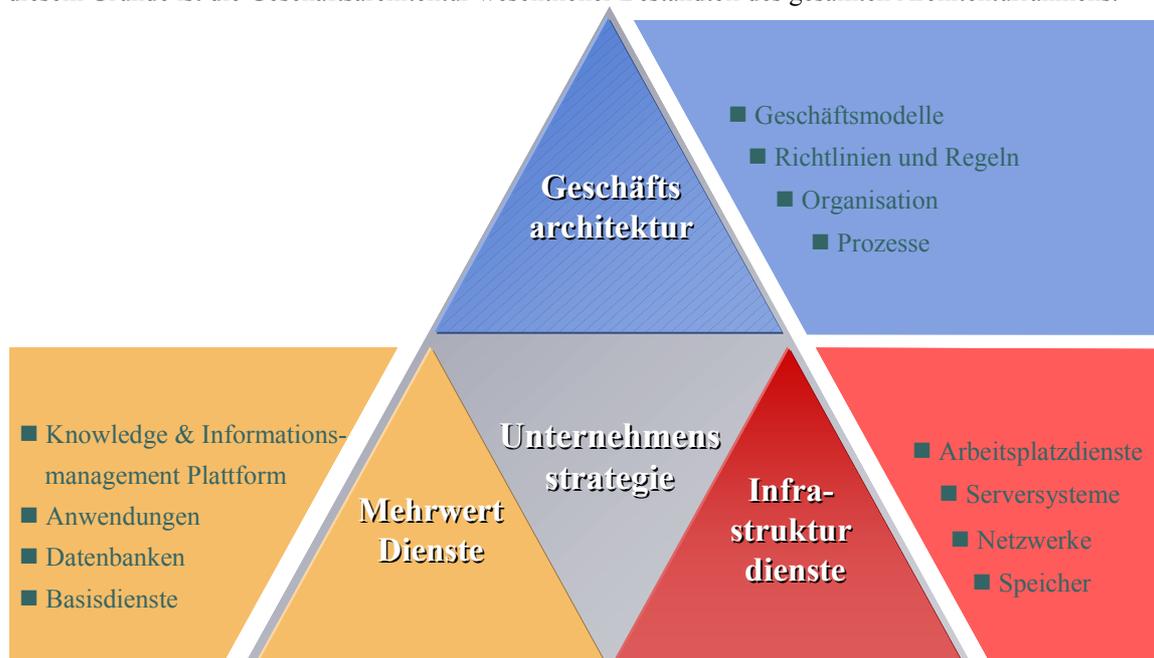
<sup>1</sup> Dieser Beitrag wurde als eingeladene Key-Note auf der Frühjahrskonferenz präsentiert.

Die *Unternehmensrichtlinien und -regeln* beschreiben die allgemeinen Grundlagen des Unternehmens für sein wirtschaftliches Handeln und geben damit die Rahmenbedingungen vor. Diese Richtlinien können sowohl durch das Unternehmen selbst festgelegt sein, wie auch durch Gesetze und allgemeine Rahmenbedingungen in Umwelt und Markt vorgegeben sein.

Die *Organisationsarchitektur* beschreibt die Unternehmensstrukturen und die Kooperations- und Leistungsbeziehungen mit Marktpartnern auf der Unternehmensgesamtebene und die Organisationsstrukturen und Modelle für die betrieblichen Teilbereiche.

Die *Prozeßarchitektur* systematisiert und beschreibt die betrieblichen Geschäftsprozesse und ihren Beitrag zur Leistungserstellung.

Alle vier Bausteine der Geschäftsarchitektur bestimmen die Anforderungen für die IuK-Landschaft. Aus diesem Grunde ist die Geschäftsarchitektur wesentlicher Bestandteil des gesamten Architekturrahmens.



**Abbildung 1: Architekturrahmen und Teilarchitekturen einer Unternehmensarchitektur**

Die IuK-Landschaft setzt sich aus den Mehrwertdiensten und der Infrastruktur zusammen. Die Mehrwertdienste setzen sich aus den folgenden Bausteinen sowie Diensten zur Informationssicherheit zusammen:

Die *Knowledge und Informationsmanagement Plattform* dient zur strukturierten Suche und Darstellung von Informationen und Wissen im Unternehmen.

Der Baustein der *Applikationen* setzt sich aus den Portalen, den Unternehmensanwendungen und den Middleware Services zusammen. Portale sind der universelle Zugang zu Anwendungen und der Präsentation von Inhalten. Unternehmensanwendungen dienen der Planung, Steuerung und Kontrolle von betrieblichen Prozessen. Diese Kategorie umfaßt alle Anwendungssysteme, die im Unternehmen entlang der Wertschöpfungskette von der Beschaffung bis zum Absatz eingesetzt werden. Anwendungssysteme sind als Unterstützung für alle Unternehmensprozesse, wie sie in der Prozeßarchitektur skizziert sind, zu finden. Die Middleware Services integrieren Applikationen und unterstützen die Bearbeitung entlang der Prozeßkette.

Die *Datenbanken* bilden eine unternehmensweite und integrierte Sicht auf alle Unternehmensdaten wie z. B. Produktdaten, Kunden- und Geschäftspartnerdaten oder Logistikdaten.

Unter *Basisdiensten* sind grundlegende, von einzelnen Geschäftsprozessen unabhängige Anwendungen zu verstehen, die als Bausteine in verschiedenen Anwendungen eingesetzt werden können. Hierzu zählen beispielsweise Email, Directory Dienste, Single Sign On, Web Services etc.

Die Mehrwertdienste setzen auf der IuK-Infrastruktur auf. Deren Dienste sind entsprechend den Anforderungen der Mehrwertdienste zu dimensionieren und bereitzustellen. Sie setzen sich aus den folgenden Bausteinen und den Diensten zur Informationssicherheit zusammen:

Die *Arbeitsplatzdienste* dienen der Präsentation und Bearbeitung von Informationen sowie zur Produktivitätsunterstützung am Arbeitsplatz (Büro, Fertigung etc.). Sie beinhalten die Auswahl an Endgeräten (fix und mobil) auf der Client Seite, sowie Grundapplikationen wie Office-Anwendungen und Browser.

Die *Server Systeme* dienen zur Abwicklung der Backend Ressourcen wie beispielsweise Applikationen, Integrationsdienste oder Web Anwendungen.

Der *Speicher* Baustein stellt die Speicherkapazitäten sowohl am Client wie für die Anwendungen auf dem Server zur Verfügung.

Die Netzwerke sichern die Einbindung in den Informationsverbund im LAN und WAN Bereich, sowohl unternehmensintern wie auch zwischen Unternehmen und in Internet.

Der Architekturrahmen dient somit als Grundlage für die geschäftsorientierte Bebauung der IuK-Landschaft. Die Entwicklung der Architektur ist die Voraussetzung für eine dynamische und schnelle Anpassung an Marktanforderungen. Drei wesentliche Zielsetzungen werden mit der Architekturentwicklung verfolgt:

- *Flexibilität* einer offenen, auf Standards basierenden Architektur mit einem modularen und komponentenbasierten Aufbau,
- *Anpassungsfähigkeit* durch Verwendung wiederverwendbarer Lösungen und Skalierbarkeit der Bausteine,
- *Verlässlichkeit* aufgrund der vollständigen Erfassung aller Bausteine, der Definition von Qualitätsstandards, sowie der Verfügbarkeit der Bausteine auf Basis definierter Dienste.

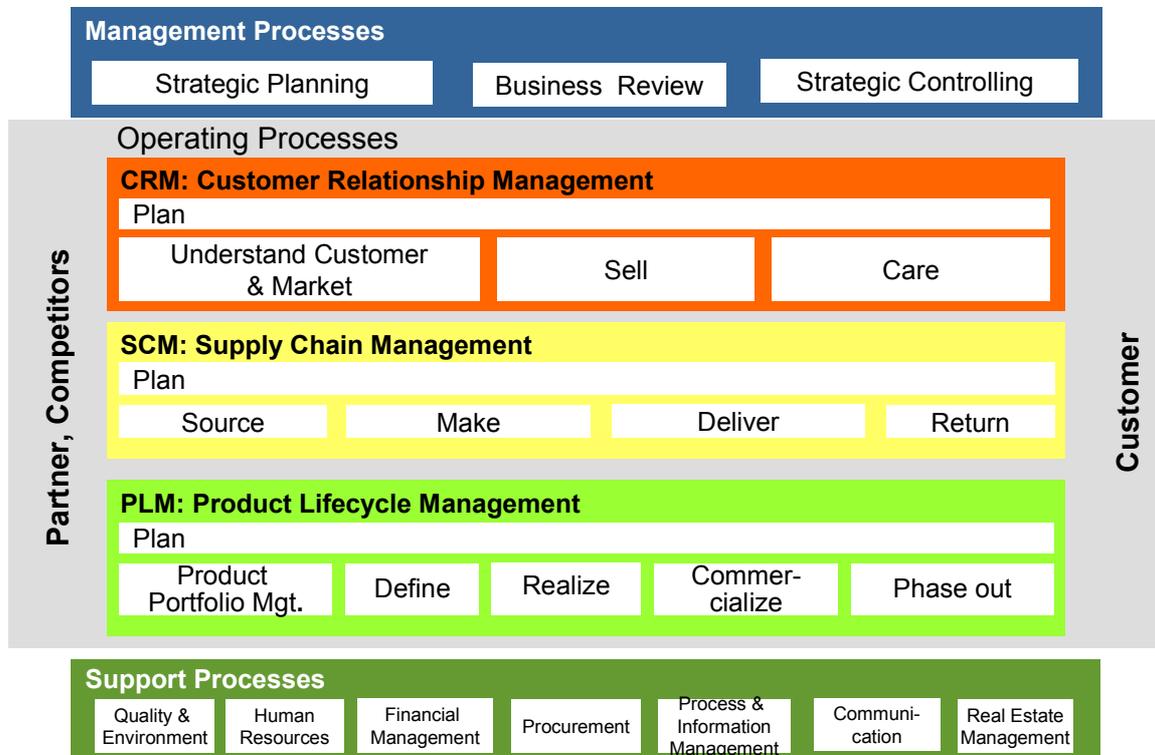
Der Architekturrahmen zeigt Strukturen und Zusammenhänge der Bausteine der IuK-Landschaft und ist damit ein entscheidendes Instrument für die systematische Weiterentwicklung der IuK-Landschaft.

## 2 Prozeßarchitektur

Das Prozeßhaus der Abbildung 2 stellt eine Referenzarchitektur für Geschäftsprozesse zur Verfügung. Grundlage für das Prozeßhaus waren verschiedene unternehmensinterne Systematiken und Prozeßbeschreibungen (siehe [2] beschrieben in [3]) sowie externe Publikationen, z.B das Supply Chain Operational Model [4] oder die Prozeßsystematik des American Productivity & Quality Center (APQC) [5]. Es beschreibt 25 Geschäftsprozesse, die im wesentlichen die Prozeßlandschaft eines Unternehmens abdecken. Diese sind in die folgenden Prozeßcluster eingeteilt:

- Managementprozesse,
- Kunden/Partner Beziehungsprozesse,
- Supply Chain Management,
- Product Life Cycle Management,
- und Support Prozesse

Die Prozesse werden nach den vier Geschäftsarten Produktgeschäft, Systemgeschäft, Projekt-/Lösungsgeschäft und Dienstleistungsgeschäft differenziert.



**Abbildung 2: Struktur des Prozeßhauses**

Die *Managementprozesse* beinhalten die übergeordnete Planung und Überwachung des Gesamtgeschäftes. Sie dienen der Führung sowie Aufbau und Erhaltung des Gesamtunternehmens und seiner Wertschöpfung. Gegenstand dieser Prozesse sind die Geschäftspolitik und -strategie (Umsatz, Kosten, Ergebnis, Finanzen, Investitionen und Personal).

Die *operativen Prozesse* sind die Prozesse, die aus Sicht des Kunden eine Wertschöpfung erbringen. Es handelt sich um Kernprozesse des Unternehmens wenn neben dem wahrnehmbaren Kundennutzen die Prozesse einmalig sind, indem unternehmensspezifische Ressourcen genutzt werden, die nicht leicht imitierbar oder zu substituieren sind [6]. Zu den wertschöpfenden Aktivitäten des Unternehmens gehören die Kunden/Partner Beziehungsprozesse, das Supply Chain Management und die Product Life Cycle Prozesse.

*Kunden/Partner-Beziehungsprozesse* umfassen die Betreuung von Kunden und potentiellen Marktpartnern von der Anbahnung über die Abwicklung und Nachbetreuung von Geschäften für kundenorientierte Leistungen und Service.

Das *Supply Chain Management* umfaßt die gesamte logistische Kette vom Sourcing bis zur Auslieferung von Produkten und Dienstleistungen.

Der *Product Life Cycle* umfaßt den gesamten Lebenszyklus von Produkten von der ersten Idee über die Entwicklung bis zur Rücknahme der Produkte vom Markt.

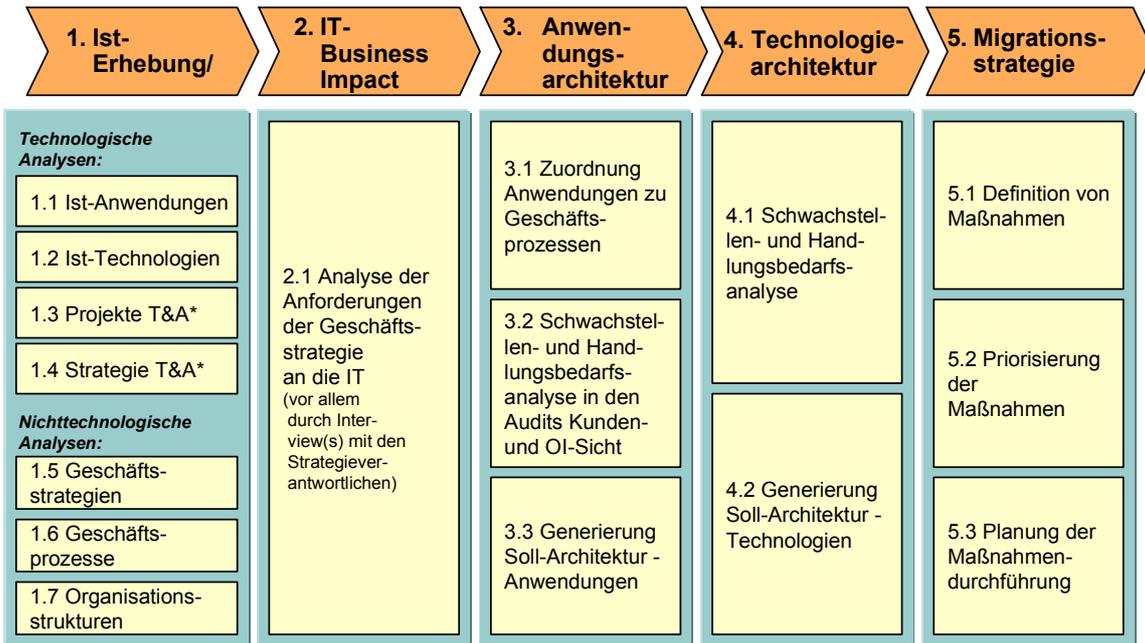
*Supportprozesse* unterstützen die operativen Prozesse, stellen aber für sich genommen noch keine eigene Wertschöpfung aus Sicht des Kunden dar. Die *Supportprozesse* dienen der Bereitstellung der Infrastruktur für die Geschäftsabwicklung. Sie beinhalten z.B. Personalmanagement, Qualitätsmanagement und Umweltschutz, Einkauf, Financial Management, Bereitstellung von IT-Dienste und Infrastruktur, Kommunikation und Real Estate Management.

Die Prozeßsystematik wird nach vier verschiedenen Geschäftstypen differenziert. Typische Anforderungen an Prozesse können so systematisch erfaßt und für das durch die Geschäftstypen repräsentierte Geschäft grundlegend beschrieben werden.

### **3 Vorgehen und Integration der Teilarchitekturen**

Das Prozeßhaus dient als Ausgangsbasis für die systematische Bebauung der IuK-Landschaft. Grundlage der Bebauung sind die im Architekturrahmen beschriebenen Bausteine und ihre Abhängigkeiten.

Der Applikation Navigator stellt ein methodisches Vorgehen für die IuK-Bebauung zur Verfügung [7]. Das prinzipielle Vorgehen ist in der Abbildung 3 in der Übersicht dargestellt.



\* T & A: Technologie und Anwendungen

Abbildung 3: Vorgehen für die IuK Bebauung

In einer Ist-Analyse wird die derzeitige Bebauung mit Applikationen und die technologische Infrastruktur erhoben. Dies erfolgt systematisch anhand des Architekturrahmens für die IuK-Landschaft. Dabei werden auf der Grundlage des Prozeßhauses jedem Geschäftsprozeß die unterstützenden Applikationen in sogenannten „Bebauungsplänen“ zugeordnet.

Methodisches Hilfsmittel für die Beschreibung der Applikationsarchitektur sind Templates, die auf horizontalen Achse jeweils die Prozesse analog dem Prozeßhaus aufgetragen haben. Hier sind unterschiedliche Detaillierungen möglich, von einer Übersicht über die Gesamtbebauung der Prozeßlandschaft bis zur Darstellung der Unterstützung einzelner Prozeßschritte eines Prozesses. Die vertikale Achse wird genutzt um das Einsatzfeld der Applikation wie die Organisation (z.B. Geschäftszweige), Region oder ähnliches aufzuzeigen. Abbildung 4 zeigt ein fiktives Beispiel einer High Level Applikationsarchitektur für die CRM und PLM Prozesse differenziert nach Geschäftszweigen. Templates mit dieser Grundstruktur gibt es für alle 25 Prozesse des Prozeßhauses, die mit der Ist-Situation bzw. Zielplanung des Untersuchungsbereichs gefüllt werden. Auf diese Weise können übersichtlich Lücken oder Überdeckungen von Anwendungen für jeden Prozeß aufgezeigt werden.

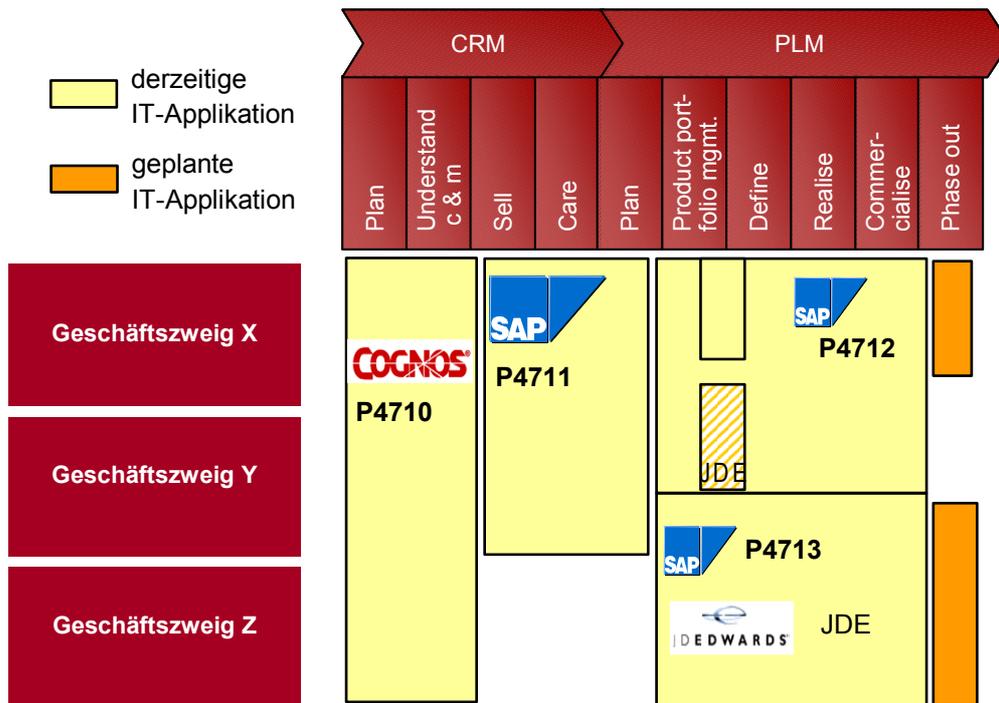


Abbildung 4: fiktives Beispiel eines Bebauungsplan

Ausgehend von der Geschäftsstrategie und Business Impact Interviews wird die Zielarchitektur für die Applikationen und Infrastruktur (Technologiearchitektur) definiert. Über eine GAP-Analyse werden die erforderlichen Maßnahmen zur Erreichung der Zielarchitektur formuliert und priorisiert und. Diese münden dann in eine Migrationsstrategie und entsprechende IT Projekte zur Weiterentwicklung der Architektur.

Bebauungspläne sind damit ein geeignetes Instrumentarium um Prozeß- und Applikationsarchitektur zu integrieren. Auch zwischen den anderen Teilarchitekturen bestehen Zusammenhänge, so ergeben sich beispielsweise aus der Organisationsarchitektur geographische Verteilungen, die spezifische Anforderungen an die Infrastruktur definieren.

## 4 Literatur

- [1] Foegen, M./ Battenfeld, J.: Die Rolle der Architektur in der Anwendungsentwicklung, Informatik Spektrum 24(2001)5, S. 290-301
- [2] IK Siemens AG, Prozeßhaus: Prozeßhaus Version 3.0, Siemens AG, IK, München 2001
- [3] Rohloff, M.: Das Prozeßrahmenwerk der Siemens AG: Ein Referenzmodell für betriebliche Geschäftsprozesse als Grundlage einer systematischen Bebauung der IuK-Landschaft, in: Becker, J.; Knackstedt, R. (Hrsg.): Wissensmanagement mit Referenzmodellen - Konzepte für die Anwendungssystem- und Organisationsgestaltung, Heidelberg 2002, S. 227-235;
- [4] SCOR: Supply Chain Operations Reference-Model, Version 5, [www.supply-chain.org/](http://www.supply-chain.org/)

- [5] APQC: American Productivity & Quality Center: Process Classification Framework,  
<http://www.apqc.org/free/framework.cfm>
- [6] Osterloh, M./ Frost, J.: Prozeßmanagement als Kernkompetenz: Wie Sie Business Reengineering strategisch nutzen können, Wiesbaden 1998
- [7] IK Siemens AG: Application Navigator: Methodik und Templates, Siemens AG, IK, München 2001

# Logische Datenintegrationstypen

**Alexander Schwinn**

**Institut für Wirtschaftsinformatik, Universität St. Gallen**

**alexander.schwinn@unisg.ch**

***Zusammenfassung:** Die Applikationslandschaft grösserer Unternehmen birgt eine komplexe, vernetzte Struktur von Applikationen. Diese erfordert immer wieder, dass Daten zwischen Applikationen ausgetauscht oder verteilt werden. Der vorliegende Beitrag identifiziert und systematisiert verschiedene Typen der Datenintegration. Zu den einzelnen erkannten Integrationstypen werden Vor- und Nachteile sowie Anwendungsbereiche aufgezeigt. Dadurch wird zudem bei zukünftigen Datenintegrationsprojekten eine Entscheidungsunterstützung gegeben, bei welchen Anforderungen welcher Datenintegrationstyp gewählt werden sollte.*

***Schlüsselworte:** Datenintegration, Datenredundanz, EAI, Enterprise Application Integration*

## 1 Einleitung

Diese Arbeit entstand im Rahmen eines Forschungsprojektes im Bereich Enterprise Application Integration (EAI). Einen Schwerpunkt bildet die Entwicklung einer geeigneten Technik zur systematischen Darstellung von Applikationsbeziehungen auf unterschiedlichen Abstraktionsniveaus. Eine Form der Integration ist die Integration von Daten. Dieser Beitrag systematisiert verschiedene Typen der Datenintegration auf logischer Ebene und diskutiert dabei Anwendungsbereiche, Vor- und Nachteile der unterschiedlichen Varianten.

Im nächsten Abschnitt wird zunächst die Ausgangssituation beschrieben, anschliessend werden vorhandene Integrationsansätze aus der Literatur vorgestellt. Auf dieser Basis erfolgt die eigentliche Diskussion der unterschiedlichen Datenintegrationstypen auf logischer Ebene und ihren Eigenschaften im vierten Abschnitt.

## 2 Ausgangssituation

Die Applikationslandschaft eines Unternehmens wird mit ihrer Weiterentwicklung zunehmend komplexer, sofern keine Standardisierung erfolgt. Neue Technologien im Zusammenhang mit dem Electronic Business erhöhen die Komplexität. In der evolutionär gewachsenen Applikationslandschaft der einzelnen Unternehmen finden sich Applikationsgruppen, die entlang der funktionalen Grenzen entsprechend der Unternehmensorganisation entstanden sind. Neue Geschäftsmodelle, die sich z.B. durch Electronic Commerce ergeben, führen zu einem höheren Integrationsbedarf zwischen den verschiedenen Anwendungen. Fusionen führen in der Praxis dazu, dass funktional ähnliche Systeme aus den Vorgängerunternehmen auch nach der Fusion noch parallel betrieben werden [KrSt02], sie jedoch miteinander gekoppelt werden.

### 3 Vorhandene Integrationsansätze

Gegenwärtig werden unter dem Stichwort „Enterprise Application Integration“ (EAI) verschiedene Integrationsansätze diskutiert, wobei unterschiedliche Klassen der Integration gebildet werden. In [SMFS02] und [BFGH02] wird zwischen einer daten-, ereignis- oder funktionsorientierten Integration unterschieden, wobei unter einer ereignisorientierten Kopplungsarchitektur die Übertragung von Ereignissen und den dazugehörigen Daten zwischen Applikationen verstanden wird. Der Austausch erfolgt dabei in Form loser Kopplung durch Übertragung von Nachrichten [FeSi01, S. 225]. Bei der datenorientierten Kopplung werden gemeinsam genutzte Daten mehrerer Applikationen manipuliert. Eine datenorientierte Kopplung erfolgt dabei durch enge Kopplung [FeSi01, S. 225]. Eine funktionsorientierte Kopplung dient zur gemeinsamen Nutzung von Funktionen durch mehrere Applikationen [SMFS02, S. 463]. Eine alternative Unterteilung von EAI wird von [DLPR02] vorgenommen. Sie unterscheiden zwischen Integration auf Datenebene, auf API-Ebene, auf Prozessebene und Integration auf Userebene. Im Unterschied zu [SMFS02] wird in [DLPR02, S. 65] unter der Integration auf Datenebene die Integration auf Datenbankebene und die damit verbundenen Technologien verbunden. Eine Integration auf API-Ebene erfolgt durch Aufruf angebotener API-Funktionen. Unter Integration auf Prozessebene verstehen [DLPR02, S. 66] die gemeinsame Nutzung von Business-Logik. Die Integration auf Userebene schliesslich wird als „primitive“ Methode der Integration bezeichnet. Unter ihr ist die Integration auf Präsentationsebene zu verstehen. Eine weitere Einteilung unterschiedlicher Integrationsformen findet sich in [ÖRV96, S. 29]. In diesem Ansatz erfolgt eine Aufteilung gemäss dem klassischen Schichtenmodell von Applikationen in Integration auf Präsentations-, Applikationslogik- und Datenebene. [RiK00] unterscheidet in Integration auf Prozess-, Objekt- und Datenebene. Eine Aufteilung in diese drei Ebenen erfolgt anhand der Kriterien „ausgetauschte Information“ und „gemeinsame Metadaten“. Die ausgetauschte Information wird auf der Prozessebene als die Bedeutung, auf der Objektebene als die Nachricht und auf der Datenebene als die Bits definiert. Die gemeinsamen Metadaten sind auf der Prozessebene die Prozessdefinitionen, auf Objektebene das Vokabular und auf Datenebene das Datenübertragungsprotokoll.

Teilweise haben einzelne Ansätze Gemeinsamkeiten, jedoch existieren auch zahlreiche Unterschiede bezüglich der Anwendbarkeit. So sind einzelne Ansätze eher logisch/konzeptionell, andere behandeln bereits konkrete Implementierungen oder gehen auf die darunterliegende Infrastruktur oder Produkte ein.

Die in vielen Ansätzen [RiK00, ÖRV96, DLPR02, FeSi01] vorgenommene Unterteilung in funktions- und datenorientierte Integration wird in diesem Beitrag nicht vorgenommen. Vielmehr wird hier unter Datenintegration alle Formen der Integration verstanden, die Nutzdaten zwischen Applikationen austauschen. Ausgangspunkt der Betrachtungen ist der fachliche Informationsbedarf einer Applikation (z.B. Kundendaten, Produktdaten, usw.). In welchen Formen dieser Bedarf erfüllt werden kann ist Gegenstand des Beitrags, unabhängig davon, ob der Bedarf über Funktionsintegration (z.B. Aufruf einer API-Funktion, Aufruf eines Corba-Services, usw.) oder über Datenintegration im Sinne oben genannter Definitionen (z.B. über ODBC, Datenbankreplikation über das DBMS, usw.) erfüllt wird. Der Aufteilung in funktions- und datenorientierter Integration wird hier nicht gefolgt, weil Datenabhängigkeiten auf logischer Ebene erkannt werden sollen, unabhängig von verwendeten Technologien, Infrastruktur oder Produkten. Somit ist der hier vorgestellte Ansatz implementierungs- und infrastrukturneutral. Ebenso wird auch nicht auf zukünftige Technologien eingegangen. Zielsetzung ist es, aufgrund der identifizierten logischen Datenintegrationstypen zukünftige Implementierungsentscheidungen zu unterstützen. Die identifizierten Datenintegrationstypen werden im nächsten Abschnitt dargestellt.

## 4 Varianten der Datenintegration

Im den folgenden Abschnitten wird zunächst eine Übersicht über die hier vorgestellten Typen der Datenintegration gegeben. Anschliessend werden die einzelnen Varianten detailliert vorgestellt und Anwendungsbereiche, sowie Vor- und Nachteile der einzelnen Typen aufgezeigt.

### 4.1 Übersicht

Die hier vorgestellte Klassifikation von Datenzugriffen geht immer davon aus, dass eine Applikation bestimmte Daten zu einem vorgegebenen Zeitpunkt, in einem bestimmten Format und in einer vorgegebenen Qualität benötigt (Informationsbedarf). Die Klassifikation zeigt nun auf, wie der Bedarf der Applikation erfüllt werden kann. Ziel des hier vorgestellten Ansatzes ist es, nicht nur zeitliche und örtliche Datenabhängigkeiten, sondern auch die Entstehung von Redundanz zu erkennen. Gerade bei der Einführung neuer Applikationen oder durch Ablösen alter Applikationen liefern diese Abhängigkeiten eine wertvolle Information. Abbildung 3 zeigt die im Folgenden vorgestellten möglichen Typen, einem Informationsbedarf gerecht zu werden:

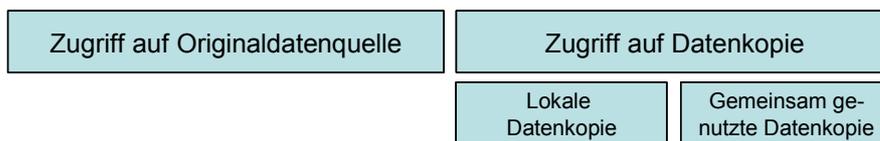


Abbildung 3: Typen der Datenintegration

Hauptunterscheidungskriterium ist dabei, ob die Applikation auf die Originaldatenquelle oder auf eine Datenkopie zugreift. Zunächst werden redundanzfreie Typen vorgestellt (Zugriff auf Originaldatenquelle), anschliessend werden Zugriffsalternativen auf Datenkopien vorgestellt. Bei Zugriff auf Datenkopien werden zwei unterschiedliche Typen unterschieden: Lokale Datenkopien und gemeinsam genutzte (share) Datenkopien.

### 4.2 Redundanzfreie Integration

Unter Redundanz wird die mehrfache Speicherung derselben Daten verstanden. Diese ist insofern problematisch, als dass sie bei geänderten Daten erfordert, dass diese Änderungen auch an allen Kopien der Originaldaten vollzogen werden. Andernfalls sind Inkonsistenzen die Folge, welche die Verarbeitung der Daten behindern oder gar unmöglich machen können. Konsequenterweise wird in der Literatur empfohlen, Redundanz zu vermeiden (z.B. in [Dit99]). Um Redundanz zu vermeiden, muss immer auf das Original der Datenquelle zugegriffen werden. Dies ist in einigen Szenarien unproblematisch und sinnvoll, jedoch kann ein direkter Zugriff auf eine Datenquelle nur unter bestimmten Voraussetzungen erfolgen. Der Aufruf erfolgt dann meist entweder über die beteiligten Datenbankmanagementsysteme (DBMS) oder mittels eines API-Aufrufs (Application Programming Interface) [AKVG01]. Der Aufruf über das Datenbanksystem kann jedoch nur implementiert werden, wenn ein Zugriff auf das System gegeben ist und die Applikation keine „Black Box“ ist. Standardapplikationen erlauben häufig keinen direkten Zugriff auf das Datenbanksystem, bieten jedoch Schnittstellen in Form von APIs an. Ein API-Aufruf kann aber nur dann erfolgen, wenn zum einen der angebotene Aufruf genau die Anforderungen erfüllt, bzw. wenn die aufrufende Applikation dementsprechend manipuliert werden kann. Eine Erweiterung des angebotenen API ist wiederum in den meisten Fällen nicht möglich, sofern es sich nicht um Eigen- oder Auftragsent-

wicklungen handelt. Tabelle 1 gibt eine Übersicht über die Anwendungsbereiche, sowie Vor- und Nachteile bei Zugriff auf die Originaldatenquelle:

<b>Zugriff auf Originaldatenquelle</b>	
<b>Anwendungsbereiche</b>	
<ul style="list-style-type: none"> <li>• Bei Industriestandards</li> <li>• Bei Eigenentwicklung (inkl. Auftragsentwicklung)</li> </ul>	
<b>Vorteile</b>	
<ul style="list-style-type: none"> <li>• Redundanzfrei → keine Kosten für Redundanz (Speicher, Synchronisation, usw.)</li> <li>• Keine Dateninkonsistenzen</li> <li>• Einfach zu implementieren (i.d.R. durch einen Adapter)</li> <li>• Geringste Komplexität</li> <li>• Kein Overhead durch zusätzliche Komponenten, Synchronisation der Daten</li> </ul>	
<b>Nachteile</b>	
<ul style="list-style-type: none"> <li>• SW-Komponenten sind nicht immer zugänglich und beeinflussbar</li> <li>• Performance-Probleme beim Zugriff vieler Applikationen</li> <li>• Enge Kopplung (Verfügbarkeit geringer, Change Management komplexer)</li> <li>• Locking-Probleme bei komplexen Transaktionen</li> <li>• Plattformübergreifende Kommunikation unter Umständen sehr komplex</li> </ul>	

Tabelle 1: Anwendungsbereiche, Vor- und Nachteile bei Zugriff auf die Originaldatenquelle

Ist ein direkter Zugriff auf die Originaldaten aus genannten Gründen nicht möglich oder gewünscht, kann zusätzlich ein Vermittler eingesetzt werden, über den eine Applikation Daten von einer Originaldatenquelle empfangen kann. Die Kopplung wird dadurch loser. Dabei wird die Integrationslogik auf einen Vermittler übertragen, da auf die Applikation, die den Originaldatenbestand hält, kein Zugriff gewährt wird, weder auf Applikations- noch auf Datenbankebene. Im Wesentlichen übernimmt der Vermittler folgende Aufgaben:

- **Transformation:** Eine Transformation kann sowohl auf semantischer Ebene (Mapping von Dateninhalten, z.B. Umwandlung von Ländercodes oder Währungscodes) als auch auf syntaktischer Ebene (Transformation unterschiedlicher Datenformate oder Nachrichtenformate) erfolgen.
- **Routing:** Das Routing ist für die Verteilung der Nachrichten zu den beteiligten Applikationen verantwortlich. Zudem werden Mechanismen zur Pufferung von Nachrichten über die Routingfunktionalität abgebildet.
- **Komposition/Dekomposition:** Die Kompositionskomponente setzt mehrere Nachrichten zu einer zusammen, die Dekompositionskomponente teilt eine Nachricht in mehrere auf.
- **Steuerung:** Die Steuerungskomponente steuert zeitliche Abläufe und löst Abhängigkeiten auf.

Diese Grundfunktionen finden sich – teilweise implizit – auch in anderen Ansätzen wieder, wenngleich die Bezeichnungen und Abgrenzungen vielgestaltig sind (vgl. z.B. [RieVog96, SMFS02]). Weitere allgemeine Vor- und Nachteile von enger und loser Kopplung sind in [RuhBro01, Cum02] zu finden. Die Kopplung gibt den Grad der Unabhängigkeit zwischen zwei Komponenten und die Auswirkungen bei Änderungen einer Komponente auf eine andere an. Bei der losen Kopplung wird anhand weniger konkreter Schnittstellen integriert, während bei der engen Kopplung in der Regel starke Implementierungsabhängigkeiten entstehen. Lose Kopplung minimiert demnach die Abhängigkeiten (zeitliche, technische und syntaktische) zwischen einzelnen Systemen. Zudem wird durch einen Vermittler die Integrationslogik außerhalb der Applikation verwaltet, wodurch die Flexibilität bei der Konfiguration oder beim Austausch einzelner Applikationen erhöht wird.

### 4.3 Redundanz erzeugende Integration

Integrationslösungen, bei denen Redundanz entsteht, sind immer mit Mehraufwand – im Vergleich zu einer redundanzfreien Lösung – verbunden, da die Datenkopien synchronisiert werden müssen. Es besteht die Gefahr, dass Inkonsistenzen entstehen, wodurch die Qualität der Daten leidet. Ausführliche Informationen zum Thema Datenqualität sind in [Hel02] zu finden. Im Folgenden werden zwei Typen von Redundanzlösungen unterschieden: Lokale Datenkopien und gemeinsam genutzte (shared) Datenkopien. In der Literatur wird zudem häufig eine Unterscheidung getroffen (z.B. [AKVG01]), ob eine Datenkopie auf Applikationsebene oder auf Datenbankebene angelegt wird. Diese Unterscheidung wird hier nicht weiter verfolgt, weil sie zur Erkennung und Darstellung von Datenabhängigkeiten und Redundanzbeziehungen keine Relevanz hat. Über welche Mechanismen die Datenkopie angelegt wird (Datenbanksystem, Filetransfer, Publish-/Subscribe, usw.), wird erst beim physischen Design bzw. bei der Implementierung entschieden, nicht aber auf der logischen Ebene.

#### 4.3.1 Lokale Datenkopie

Bei der lokalen Datenkopie hält eine Applikation eine Datenkopie lokal, d.h. die benötigten Daten werden von einer zentralen Datenbank geliefert und von der Applikation lokal abgespeichert. Die Applikation, die eine Kopie hält, arbeitet immer mit dieser. Soll die Autonomie der datenbeziehenden Applikation gewährleistet sein, muss eine lokale Datenkopie angelegt werden. Dadurch kann die Applikation autonom arbeiten und ist nicht von Ausfällen der Masterdatenbank betroffen. Zudem ist eine lokale Datenkopie meist leistungsfähiger, und kann somit Anforderungen an Transaktionszeiten gerecht werden. Folgende Tabelle fasst Anwendungsbereiche, Vor- und Nachteile einer lokalen Datenkopie zusammen.

<b>Lokale Datenkopie</b>	
<b>Anwendungsbereiche</b>	
<ul style="list-style-type: none"> <li>• Höhere Verfügbarkeit als die der Zulieferapplikationen erforderlich</li> <li>• Autonomie des Systems (Bereitstellung und Betrieb) muss gegeben sein</li> <li>• Beispiele: DWH, Channel- und Vertriebsapplikationen, Packaged Applications</li> </ul>	
<b>Vorteile</b>	
<ul style="list-style-type: none"> <li>• Hohe Performance</li> <li>• Hohe Verfügbarkeit</li> <li>• Stand-alone-fähig</li> <li>• Single-Source zur Auswertung von Daten (optimale Datensicht für analytisches System/Analysen, keine verteilten Abfragen über mehrere Datenquellen hinweg nötig)</li> </ul>	
<b>Nachteile</b>	
<ul style="list-style-type: none"> <li>• Aktualität der Daten (in Abhängigkeit von Synchronisationsabständen) geringer</li> <li>• Overhead durch Synchronisationsmechanismen (Datenkonsistenz)</li> <li>• Kosten für Redundanz (Speicher, Management der Redundanz und Change Management)</li> <li>• Mögliche inkonsistente Sichten (z.B. auf Kundenseite: Internetbanking, Bankomat)</li> </ul>	

Tabelle 2: Anwendungsbereiche, Vor- und Nachteile lokaler Datenkopien

#### 4.3.2 Gemeinsam genutzte (Shared) Datenkopie

Bei der gemeinsam genutzten (shared) Datenkopie handelt es sich um eine von mehreren Applikationen gemeinsam genutzte Datenquelle. Der Datenbestand in einer gemeinsam genutzten Datenkopie beinhaltet in der Regel Datenkopien aus mehreren Quellen. Dabei werden Daten aus heterogenen Datenhaltungssystemen zusammengeführt. Ein typisches Beispiel für eine gemeinsam genutzte Datenkopie wäre ein Ope-

rational Data Store (ODS), in dem benötigte operative Daten aus mehreren Quellsystemen integriert werden, auf die unterschiedliche Applikationen zugreifen können [Inm99]. Der Zugriff erfolgt dabei transparent, da die Applikation alle benötigten Daten aus einer Quelle – nämlich dem ODS – geliefert bekommt. Ein verteilter Zugriff/verteilte Abfragen entfallen somit, wodurch die Komplexität seitens der abfragenden Applikation reduziert wird. Im Vergleich zur lokalen Datenkopie entsteht zudem weniger Redundanz, da nicht eine Datenbank für eine Applikation kopiert wird, sondern nur benötigte Daten, zugänglich für mehrere Applikationen. Ein weiteres Anwendungsbeispiel für eine Shared Datenkopie ist die Kopplung von Rechenzentren (mittlere bis grosse geographische Entfernung). Die Applikationslogik kann so verteilt in mehreren Rechenzentren ablaufen. Dies fordert zudem die Wiederverwendung von Applikationslogik, da der Zugriff für die Clients transparent erfolgt. Eine gemeinsam genutzte Datenkopie kann jedoch nicht immer angelegt werden, da z.B. ein Zugriff auf das Datenbanksystem einer Standardsoftware nicht immer möglich ist. Tabelle 3 fasst Anwendungsbereiche, Vor- und Nachteile der gemeinsam genutzten Datenkopie zusammen.

<b>Gemeinsam genutzte (Shared) Datenkopie</b>	
<b>Anwendungsbereiche</b>	
<ul style="list-style-type: none"> <li>• Verteilte Rechenzentren (mittlere bis grosse geographische Entfernung)</li> <li>• Zusammenführen der Daten aus heterogenen Datenhaltungssystemen</li> <li>• Beispiel: ODS</li> </ul>	
<b>Vorteile</b>	
<ul style="list-style-type: none"> <li>• Für Clients ist ein transparenter Zugriff möglich</li> <li>• Lose Kopplung zwischen verteilten Rechenzentren</li> <li>• Reduzierung der Abfragekomplexität (keine verteilten Abfragen nötig)</li> <li>• Förderung der Wiederverwendung von Applikationslogik</li> <li>• Weniger Redundanz als bei lokalen Datenkopien und somit weniger Probleme, die mit Entstehung von Redundanz verbunden sind</li> </ul>	
<b>Nachteile</b>	
<ul style="list-style-type: none"> <li>• Replikation von Daten bei Packaged Applications oft nicht realisierbar</li> <li>• Entstehung von Redundanz, und die damit verbundenen Nachteile (vgl. Tabelle 2)</li> </ul>	

Tabelle 3: Anwendungsbereiche, Vor- und Nachteile der gemeinsam genutzten (shared) Datenkopie

Wird eine redundante Lösung zur Integration von Daten gewählt, ist zudem entscheidend, zu welchen Zeitpunkten die Kopie angelegt/aktualisiert wird (z.B. wie aktuell werden die Daten im ODS gehalten). Es wurden drei Kategorien von Zeitpunkten identifiziert: Unbekannte, periodische und Near Realtime Kopien/Aktualisierungen. Diese drei Kategorien werden im Folgenden kurz vorgestellt.

Die einfachste, jedoch meist vernachlässigte Form der Datenkopie ist die Manuelle. Bei ihr wird zu einem unbekanntem Zeitpunkt manuell (beispielsweise durch eine Benutzereingabe) eine Datenkopie erzeugt. Eine manuelle Integration von Daten ist immer dann sinnvoll, wenn eine Integration nicht automatisiert erfolgen kann, d.h. wenn keine Implementierung vorhanden ist oder eine Implementierung nicht wirtschaftlich wäre. Dies kann immer nur dann der Fall sein, wenn sehr selten Aktualisierungen erfolgen und die Datenmengen und die Anzahl betroffener Applikationen gering sind. Typische Anwendungsbereiche sind Änderungen von Länder- oder Sprachcodes. Ändert sich ein Ländercode, müsste dieser dann bei den entsprechenden Applikationen manuell nachgepflegt werden. Der einzige Vorteil dieser Lösung besteht in der Wirtschaftlichkeit. Es müssen keine Integrationstools angeschafft oder entwickelt werden (allenfalls Eingabemasken bei den betroffenen Applikationen). Die Nachteile einer manuellen Lösung liegen in der Aktualität der Daten, der Gefahr von Inkonsistenzen und der Fehleranfälligkeit. Die Aktualität der Daten ist von dem Mitarbeiter abhängig, der für die Erstellung der manuellen Datenkopie verantwortlich ist, womit auch organisatorische Probleme verbunden sind. Inkonsistenzen können entstehen, wenn einige

Systeme bereits eine Kopie besitzen, andere noch nicht. Schliesslich ist die Fehleranfälligkeit höher, da durch manuelle Dateneingabe Tippfehler entstehen können.

Bei der periodischen Integration werden Daten zu einer zuvor festgelegten Periode (z.B. 1 mal pro Tag, 1 mal pro Stunde, usw.) integriert. Eine typische Ausprägung der periodischen Datenkopie ist die Batch-Verarbeitung, die über einen Scheduler angestossen wird. Eine periodische Integration ist immer dann sinnvoll, wenn die Anforderungen an die Aktualität der Daten nicht sehr hoch sind und grosse Datenmengen integriert werden müssen. Eine Near Realtime Integration würde bei sehr grossen Datenmengen das operative System zu sehr belasten und somit die Verfügbarkeit und das Leistungsverhalten beeinträchtigen. Durch eine periodische Integration können Zeiten gewählt werden, zu denen das operative System keine grosse Last verarbeiten muss. Zudem hat die periodische Integration den Vorteil, dass in der Regel Fehler leichter rückgängig gemacht werden können (durch Rollback-Mechanismen).

Die Near Realtime Integration ist am aufwändigsten zu implementieren, sofern keine leistungsfähige Infrastruktur existiert. Sie wird immer dann gewählt, wenn die Aktualitätsanforderungen sehr hoch sind (z.B. Prüfung von Limits beim Bezug von Bargeld am Bankomaten). Bei der Near Realtime Integration sind sehr leistungsfähige Systeme erforderlich, die auch zu Peak-Zeiten die Last verarbeiten müssen. Dies wiederum ist mit hohen Kosten verbunden. Ein weiterer Nachteil der Near Realtime Integration besteht darin, dass die Daten nicht nur im Bedarfsfall integriert werden, sondern immer sofort nach der Entstehung.

Zur Orientierung, in welcher Situation, welche Art der Datenintegration (manuell, periodisch oder Near Realtime) angewendet werden kann, soll folgende Matrix einen groben Ordnungsrahmen geben:

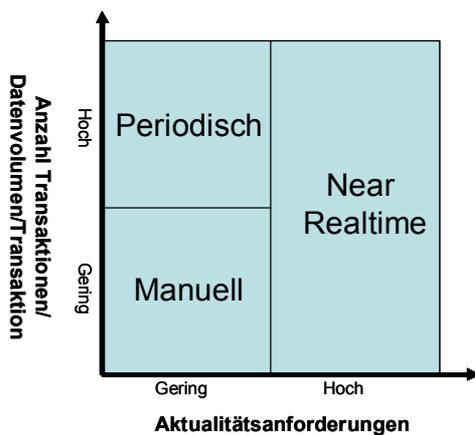


Abbildung 4: Entscheidungsunterstützung zur Wahl der Integrationsart

Zum einen wird hier nach den Aktualitätsanforderungen der Daten unterschieden, zum anderen nach Anzahl zu erwartender Transaktionen, bzw. nach zu erwartenden Datenvolumen pro Transaktion. Sind die Aktualitätsanforderungen hoch, so bleibt nur die Near Realtime Integration. Sind die Aktualitätsanforderungen eher gering, die zu integrierende Datenmenge jedoch gross, so kann die periodische Variante gewählt werden. Bei geringen Aktualitätsanforderungen und geringer Anzahl von Transaktionen kann die manuelle Integration in Erwägung gezogen werden, jedoch sollte diese aufgrund ihrer dargestellten Nachteile möglichst immer vermieden werden.

## 5 Zusammenfassung und Ausblick

Der hier vorgestellte Ansatz dient zur Systematisierung von Datenabhängigkeiten zwischen Applikationen auf logischer Ebene. Die präsentierte Systematik mit den jeweils diskutierten Anwendungsbereichen, Vor- und Nachteilen der einzelnen Varianten kann z.B. bei zukünftigen Implementierungsentscheidungen herangezogen werden. Die unterschiedlichen Typen der Datenintegration wurden bei einem Schweizer Unternehmen identifiziert, das Bankapplikationen entwickelt und betreibt. Über diese identifizierten Datenintegrationstypen wurden bei bislang drei weiteren Grossunternehmen aus dem schweizerischen Finanzdienstleistungssektor Untersuchungen durchgeführt. Dabei stellte sich heraus, dass die gleichen logischen Datenintegrationstypen in der Applikationslandschaft anzutreffen waren. Zukünftige Untersuchungen sollen (weitere) abstrakte, implementierungsneutrale Typen der Integration – nicht ausschliesslich auf der Ebene der Datenintegration – identifizieren.

Der hier vorgestellte Ansatz berücksichtigt hauptsächlich technische Aspekte bei der Entscheidung der geeignetsten Integrationsform. Bei der Entscheidung spielen aber auch organisatorische Aspekte (z.B. Datenhoheit, organisatorische Verantwortung für Daten) und finanzielle Aspekte (z.B. Berücksichtigung der vorhandenen Infrastruktur) eine wichtige Rolle, deren Betrachtungen in diesem Beitrag vernachlässigt wurden. Diese sollten bei zukünftigen Integrationsanforderungen jedoch unbedingt berücksichtigt werden.

Um die Datenabhängigkeiten erkennen und weiterführende Auswertungen vornehmen zu können, müsste ein durchgängiger Modellierungsansatz nebst zugehörigem Metamodell entwickelt werden. Beim Austausch von Applikationen oder bei der Einführung neuer Applikationen könnte so sichergestellt werden, dass alle Datenabhängigkeiten berücksichtigt werden. Die Entwicklung eines entsprechenden Modellierungsansatzes und eines zugehörigen Metamodells ist derzeit Gegenstand weiterer Arbeiten des eingangs erwähnten Forschungsprojektes.

## 6 Literaturverzeichnis

- [AKVG01] Adams, J.; Koushik, S.; Vasudeva, G; Galambos, G.: Pattern for e-business- A Strategy for Reuse, IBM Press, Double Oak, 2001.
- [BFGH02] Bunjes, B.; Friebe, J.; Götze, R.; Harren, A.: Integration von Daten, Anwendungen und Prozessen am Beispiel des Telekommunikationsunternehmens EWE TEL, in: Wirtschaftsinformatik, Jg. 44., Nr. 5, S. 415-423.
- [Cum02] Cummins, F. A.: Enterprise Integration – An Architecture for Enterprise Application and System Integration, OMG Press, 2002.
- [DLPR02] Dangelmaier, W.; Lessing, H.; Pape, U.; Rüter, M.: Klassifikation von EAI-Systemen, in: HMD – Praxis der Wirtschaftsinformatik, Jg. 39, Nr. 225, 2002, S. 61-71.
- [Dit99] Dittrich, K.: Datenbanksysteme, in: Rechenberger, P.; Pomberger, G. (Hrsg.): Informatik-Handbuch, 2. Aufl., Hanser, München, Wien 1999, S. 875-908.
- [FeSi01] Ferstl, O.; Sinz, E. J.: Grundlagen der Wirtschaftsinformatik, 4. Auflage, Oldenbourg, München 2001.
- [Hel02] Helfert, M.: Planung und Messung von Datenqualität in Data-Warehouse-Systemen. Universität St. Gallen, Dissertation 2002, Nr. 2648, difo-druck, Bamber 2002.
- [Inm99] Inmon, W. H.: Building the Operational Data Store, 2. Aufl., John Wiley, New York et al. 1999.

- [KrSt02] Kromer, G.; Stucky, W.: Die Integration von Informationsverarbeitungsressourcen im Rahmen von Mergers & Acquisitions, in: Wirtschaftsinformatik, Jg. 44, Nr. 6, 2002, S. 523-533.
- [ÖRV96] Österle, H.; Riehm, R.; Vogler, P.: Middleware. Grundlagen, Produkte und Anwendungsbeispiele für die Integration heterogener Welten, Vieweg, 1996.
- [RieVog96] Riehm, R.; Vogler, P.: Middleware – Infrastruktur für die Integration, in: Österle, H.; Riehm, R.; Vogler, P. (Hrsg.): Middleware – Grundlagen, Produkte und Anwendungsbeispiele für die Integration heterogener Welten, Vieweg, Braunschweig et al. 1996, S. 25-135.
- [RiK00] Ring, K.: EAI: Making the right Connections, Ovum Reports, Boston, 2000.
- [RuhBro01] Ruh, W. M., F.; Brown, W.: Enterprise Application Integration – A Wiley Tech Brief, John Wiley, 2001.
- [SMFS02] Schissler, M.; Mantel, S; Ferstl, O. K.; Sinz, E. J.: Kopplungsarchitekturen zur überbetrieblichen Integration von Anwendungssystemen und ihre Realisierung mit SAP R/3, in: Wirtschaftsinformatik, Jg. 44, Nr. 5, 2002, S. 459-468.



## 3LGM<sup>2</sup>: Methode und Werkzeug zur Modellierung von Unternehmensarchitekturen im Krankenhaus

Alfred Winter, Birgit Brigl, Thomas Wendt  
Institut für Medizinische Informatik, Statistik und Epidemiologie,  
Universität Leipzig  
alfred.winter@imise.uni-leipzig.de

***Zusammenfassung:** Krankenhäuser benötigen – wie auch Unternehmen anderer Branchen - eine Strategie für die Informationsverarbeitung als Teil einer Geschäftsstrategie. Im Rahmen dieser Strategie wird das Krankenhausinformationssystem (KIS) wie ein Gebäude aus einzelnen, möglicherweise heterogenen Bausteinen aktiv gestaltet. Für das Zusammenfügen der Bausteine ist ein ‚Bauplan‘ erforderlich, der in der Granularität den Aufgaben des strategischen Informationsmanagements angemessen und im Sinne der in dieser Arbeit vorgeschlagenen Definition von KIS auch umfassend sein muss. Diese Arbeit stellt ein Begriffssystem zur Beschreibung von KIS und ihrer Komponenten vor, das auch als formales Meta-Modell zur Modellierung von KIS und als Spezifikation für ein Modellierungswerkzeug dient.*

***Schlüsselworte:** Informationssysteme, Krankenhaus, Modellierung, Strategisches Informationsmanagement, Enterprise Architecture Planning*

### 1 Einleitung

Krankenhäuser entwickeln sich konsequent zu betriebswirtschaftlich geführten Unternehmen, die – wie auch Unternehmen anderer Branchen - eine Strategie für die Informationsverarbeitung als Teil einer Geschäftsstrategie [Brenner'94, Glaser+'99] benötigen.

Im Fokus des Interesses steht bei der Informationsverarbeitung das Informationssystem des Krankenhauses. Dieses **Krankenhausinformationssystem (KIS)** ist das sozio-technische Teilsystem des Krankenhauses, welches alle informationsverarbeitenden (und informationsspeichernden) Prozesse und die an ihnen beteiligten menschlichen und maschinellen Handlungsträger in ihrer informationsverarbeitenden Rolle umfasst [Winter+'01](siehe bereits [Ferstl+'94]). Der Begriff KIS meint also erheblich mehr als einzelne Anwendungssysteme. Vielmehr schließt er sowohl rechnerbasierte als auch papierbasierte Informationsverarbeitung ein und bezieht sich auf *alle* Bereiche des Unternehmens Krankenhaus.

Der medizinische Fortschritt und die damit verbundene Spezialisierung und Arbeitsteilung bei dem Prozess der Patientenversorgung erfordern eine Vielzahl fachbezogener Einrichtungen mit unterschiedlichen Aufgaben innerhalb und außerhalb eines modernen Krankenhauses. Dies führt zu einem in hohem Maße arbeitsteiligen Unternehmen mit heterogener Struktur. Andererseits muss eine *integrierte* Versorgung gerade chronisch erkrankter Patienten das Ziel sein [Holm+'02, van der Haak+'02]. Dies setzt eine intensive Kommunikation sowohl zwischen den Einrichtungen innerhalb des Krankenhauses als auch zwischen dem Krankenhaus und externen Einrichtungen (z.B. Versicherungen, Lieferanten, niedergelassene Ärzte, Rehabilitationseinrichtungen etc.) voraus.

Es ist daher naheliegend, dass die Heterogenität des Krankenhauses sich auch in seinem KIS widerspiegelt. Aufgrund der sehr speziellen Anforderungen der verschiedenen Fachabteilungen eines

Krankenhauses basieren die eingesetzten rechnerbasierten Werkzeuge häufig auf unterschiedlichen Hardware-Plattformen und Softwareprodukten verschiedener Hersteller [Kuhn+'01]. Hinzu kommt, dass neben modernen rechnerbasierten Werkzeugen auch ‚Altlasten‘ (legacy systems) [Brodie+'95] existieren und nach wie vor papierbasierte ‚Datenbanken‘ – die klassische Krankenakte – das Feld beherrschen. In diesem Umfeld kann eine mit dem Auftrag und den Zielen des jeweiligen Krankenhauses abgestimmte Strategie für die Informationsverarbeitung nur in einem umfassenden, strategischen **Informationsmanagement** entstehen und umgesetzt werden [Austin+'95, Heinrich'99, Winter+'01].

Die für den rechnerbasierten Teil des KIS benötigte Anwendungssoftware wird in aller Regel nicht in den Krankenhäusern bzw. für einzelne Krankenhäuser individuell entwickelt. Vielmehr werden im allgemeinen marktgängige Softwareprodukte im Sinne von „Fachkomponenten“ [Turowski'02] eingesetzt. Das ‚Innenleben‘ dieser Fachkomponenten und die Probleme bei ihrer (Software-) Entwicklung sind für das strategische Informationsmanagement von geringerem Interesse. Im Zentrum des Interesses steht vielmehr die Frage danach, welche Fachkomponenten für die Erfüllung welcher Unternehmensaufgaben erforderlich sind und wie die Fachkomponenten als Bausteine zusammengefügt werden können. Allerdings muss angemerkt werden, dass die am Markt bisher für KIS angebotenen Anwendungssoftwareprodukte zwar die in [Turowski'02] geforderten Eigenschaften der Wiederverwendbarkeit, Abgeschlossenheit und Vermarktbarkeit erfüllen, aber die „plug-and-play-artige Wiederverwendung“ oder gar eine Standardisierung der Funktionalität auch trotz umfangreicher Bemühungen z.B. der Domain Task Force CORBAmed [Object Management Group (OMG)'00] in der Realität nicht gegeben sind.

Für das Zusammenfügen der Bausteine zu einem KIS-‚Gebäude‘ ist ein Bauplan erforderlich. Um dies zu erreichen, erscheinen die Methoden zur Planung von **Unternehmensarchitekturen** (enterprise architecture) geeignet (z.B. [Chief Information Officer Council'01, Martin'90, Spewak+'92]).

Bereits die genannten Autoren geben z.B. mit „function“ oder „application“ auf informelle Weise ein Begriffssystem zur Beschreibung von Unternehmensarchitekturen vor. Dieses Begriffssystem kann im Sinne von [Strahinger'98] als Meta-Sprache und – da eine isomorphe bzw. homomorphe Abbildung, also ein Modell des Unternehmens in die Unternehmensarchitektur anzustreben ist – auch als sprachliches Meta-Modell aufgefasst werden. Eine systematische und stringente Unterstützung der Modellierung durch rechnerbasierte Entwurfswerkzeuge ist dabei wünschenswert, kann aber um so besser gelingen, je detaillierter und präziser dieses Begriffssystem in Syntax und Semantik beschrieben ist. Gerade wegen des Ziels der Rechnerunterstützung erscheint es sinnvoll, sich zur Definition des Begriffssystems, d.h. des Meta-Modells, als Meta-Meta-Modell einer formalen anstatt einer natürlichen (Modellierungs-) Sprache zu bedienen.

Ziel dieser Arbeit ist es, auf diesen Ansätzen und früheren Arbeiten zur Modellierung von KIS [Winter+'95] aufbauend ein sprachliches Meta-Modell [Brinkkemper'90, Strahinger'98] zur Beschreibung bzw. zur Modellierung von KIS und ihrer Architekturen vorzustellen, das in dem Sinne ein formales Meta-Modell darstellt, dass zu seiner Beschreibung als Meta-Meta-Modell die formale Sprache UML [Object Management Group (OMG)'03] eingesetzt wird.

Zunächst wird untersucht, inwieweit andere Methoden und Werkzeugen zur Beschreibung und Modellierung von Informationssystemen zur Modellierung von KIS eingesetzt werden können, um das Informationsmanagement zu unterstützen. Als Konsequenz wird **3LGM<sup>2</sup>** (*Three Layer Graph Based Meta-Model*) als Meta-Modell vorgeschlagen. Da mit einem Meta-Modell alleine in der Praxis keine KIS-Modelle gebaut werden können, wird anschließend als ein Modellierungswerkzeug der *3LGM<sup>2</sup>-Baukasten* kurz vorgestellt. Die Arbeit schließt mit einer kritischen Diskussion des Nutzens des vorgestellten Ansatzes und der noch erforderlichen Forschungsarbeiten.

## 2 Anforderungen und Ansätze

In Abhängigkeit von den unterschiedlichen Aufgaben bei dem Informationsmanagement sind auch unterschiedliche Arten von Modellen erforderlich [Zachman'99]. Es ist daher nützlich, Informationsmanagement zu differenzieren und zwischen dem strategischen Informationsmanagement als Teil der strategischen Unternehmensplanung [Spewak+'92] auf der einen Seite und dem taktischen und operativen Informationsmanagement auf der anderen Seite zu unterscheiden. **Strategisches Informationsmanagement** befasst sich mit der Informationsverarbeitung des Krankenhauses als Ganzes und über einen längeren Zeitraum. Er orientiert sich strikt an dem Auftrag (mission) und den Zielen des Krankenhauses und übersetzt sie in eine passende Strategie für die Informationsverarbeitung [Austin+'95, Winter+'01].

Für das strategische Informationsmanagement ist es von herausragender Bedeutung, neben den aus dem Auftrag und den Zielen des Krankenhauses abgeleiteten (Unternehmens-) **Aufgaben** (enterprise functions) [Martin'90, Spewak+'92] des Krankenhauses auch die informationsverarbeitenden Werkzeuge, d.h. zunächst die Anwendungssysteme und dann die zugrundeliegenden Hardwarekomponenten ganzheitlich im Blick zu haben. Die Architektur des KIS muss also diese drei Ebenen in ihrem Zusammenhang berücksichtigen und ein Modell der Architektur muss dies darstellen können.

Insbesondere solche - auch formalen - Modellierungsansätze, die aus dem Bereich des Software-Engineering bzw. der Geschäftsprozessmodellierung und -optimierung stammen (z.B. [Ferstl+'98, Oberweis+'94, Scheer'94]) erscheinen für das Modellieren von "Architecture in the Large" im Rahmen des strategischen Informationsmanagements aus folgenden Gründen weniger geeignet:

- Die meisten Ansätze konzentrieren sich auf die Unternehmensaufgaben. Einige betrachten die informationsverarbeitenden Werkzeuge als Ressourcen, die selbst und deren Zusammenwirken nicht weiter beschrieben werden müssen. Andere Ansätzen berücksichtigen zwar diese Anforderung und schlagen entsprechende Sichten und Ebenen vor, beschreiben aber nicht, was auf diesen Ebenen modelliert werden soll (z.B.[Gartner Group'98, Zachman'99]).
- Keiner der untersuchten Ansätze ist in der Lage unterschiedliche Repräsentationsformen für Informationen über Objekte explizit darzustellen. Gerade dies ist jedoch im Krankenhaus mit seinem nach wie vor großen Anteil von papierbasierter Datenspeicherung unbedingt erforderlich.
- Die Stärke der Ansätze liegt in der Modellierung der (kausalen oder zeitlichen) Abhängigkeiten zwischen den Aufgaben in Form von Geschäftsprozessen. Aber die sich aus diesen Geschäftsprozessen ergebenden Kommunikationsprozesse zwischen den informationsverarbeitenden Werkzeugen sind zwar mit entsprechendem Aufwand modellier- aber nicht automatisch ableitbar.
- Die Ansätze stellen kein Meta-Modell zur Verfügung, das dem Modellierer als konkreter Leitfaden bei der Modellierung einer KIS-Architektur dienen könnte.

Als eine universelle Modellierungsmethode steht die Unified Modeling Language [Object Management Group (OMG)'03] zur Verfügung. Ohne die Verwendung des Konzepts der Stereotypen [Heverhagen+'01] ist UML jedoch als Methode für die Modellierung von KIS-Architekturen zu unspezifisch. Voraussetzung für die Nutzung der Stereotypen wäre ein Meta-Modell.

## 3 3LGM<sup>2</sup>

Das Drei-Ebenen Meta-Modell **3LGM<sup>2</sup>** (*Three Layer Graph Based Meta-Model*) ist ein Meta-Modell für die Modellierung von KIS bzw. von KIS-Architekturen. 3LGM<sup>2</sup> ist mit UML spezifiziert und unterscheidet drei Ebenen zur Beschreibung von KIS.

Die **fachliche Ebene** eines Krankenhausinformationssystems ergibt sich aus den (Unternehmens-) **Aufgaben** des Krankenhauses, deren Erledigung das KIS unterstützt, und den Informationen, die im Rahmen der Erledigung der Aufgaben jeweils bearbeitet bzw. interpretiert werden. Informationen beziehen sich auf Objekte. Objekte sind physische oder virtuelle Dinge eines Krankenhauses („part of the perceivable or conceivable universe“<sup>2</sup>), die durch Daten repräsentiert werden. Objekte mit denselben Merkmalsarten werden als **Objektyp** zusammengefasst. Objektypen in einem Krankenhaus sind z.B. (entsprechend [CEN/TC251'97]) *Patient* oder *Behandlungsfall*.

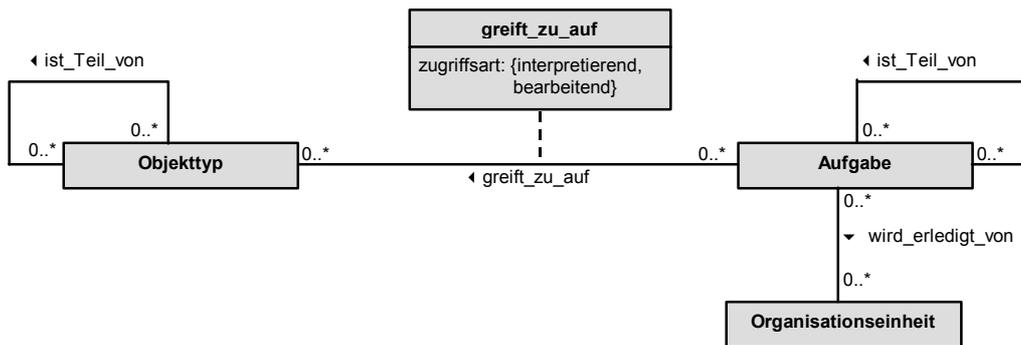


Abbildung 1: Das Meta-Modell der fachlichen Ebene.

Man kann eine Aufgabe und damit auch eine Teilaufgabe als Pflicht auffassen, die sich aus den Unternehmenszielen mehr oder weniger zwingend ergibt. Eine Aufgabe hat keinen definierten Anfang und kein definiertes Ende. Aufgaben lassen sich in Teilaufgaben gliedern. Eine typische Aufgabe eines Krankenhauses ist z.B. die *Patientenaufnahme*.

Ein Objektyp, dessen Objekte im Rahmen der Erledigung einer Aufgabe bearbeitet werden, heißt '*zu bearbeitender Objektyp einer Aufgabe*'. Ein Objektyp, von dessen Objekten Informationen gewonnen werden, heißt '*zu interpretierender Objektyp einer Aufgabe*'. Aufgaben werden in bestimmten **Organisationseinheiten** erledigt. Dies ist insbesondere bei der Zuordnung von Anwendungsbausteinen (s.u.) bedeutsam, da zur Unterstützung derselben Aufgabe häufig unterschiedliche Anwendungsbausteine eingesetzt werden [Simoneit'98].

Welche Objektypen, Aufgaben und Organisationseinheiten in einem konkreten Krankenhaus zu modellieren sind, hängt natürlich von diesem Krankenhaus ab. Allerdings können verschiedene Standardisierungsbemühungen als Grundlage von Referenzmodellen dienen, die dann für die Modellierung eines konkreten Krankenhausinformationssystems lediglich noch modifiziert werden müssen (z.B. [Ammenwerth+'02, Ingram+'95, Schadow+'00])

<sup>2</sup> ISO 1087

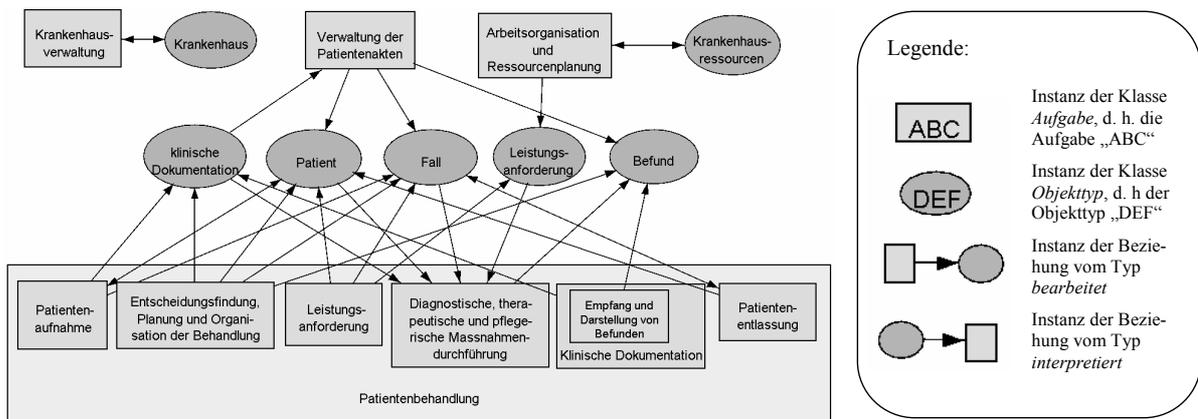


Abbildung 2: Beispiel einer fachlichen Ebene.

Abbildung 1 zeigt die UML-Notation der fachlichen Ebene. Abbildung 2 zeigt ein Beispiel einer fachlichen Ebene, das als eine Ausprägung des in Abbildung 1 dargestellten Meta-Modells zu betrachten ist. Rechtecke repräsentieren dabei Aufgaben, Ovale stehen für Objekttypen. Ein Pfeil von einem Objekttyp zu einer Aufgabe markiert einen interpretierenden Zugriff, ein Pfeil von einer Aufgabe zu einem Objekttyp einen bearbeitenden Zugriff. Verfeinerungen sind dargestellt als Rechtecke, die in anderen Rechtecken enthalten sind. Typische Aufgaben eines Krankenhauses sind hier die KRANKENHAUSVERWALTUNG, die PATIENTENBEHANDLUNG oder die VERWALTUNG VON PATIENTENAKTEN. Die Aufgabe PATIENTENBEHANDLUNG ist verfeinert in weitere Aufgaben. Typische Objekttypen sind PATIENT, LEISTUNGSANFORDERUNG oder FALL.

Im Gegensatz zum 3LGM<sup>2</sup> basieren die auf der Grundlage von 3LGM<sup>2</sup> erzeugten Modelle ausdrücklich nicht auf einer UML-Notation. Hierfür wurde eine einfache, intuitive, graphische Darstellungsweise gewählt. UML-Kenntnisse sind für die Modellierung nicht erforderlich.

Die Aufgaben werden von den in einem Krankenhaus tätigen *Personen* erledigt. Diese benutzen dafür *informationsverarbeitende Werkzeuge*. Wir unterscheiden im folgenden logische Werkzeuge, die auf Softwareprodukten oder Organisationsplänen basieren und physische Werkzeuge wie z.B. Rechnersysteme.

Auf der **logischen Werkzeugebene** (Abbildung 3) stehen die Anwendungsbausteine im Mittelpunkt. **Anwendungsbausteine** (syn. auch Anwendungssysteme) unterstützen die Erledigung von Aufgaben und sind verantwortlich für die Verarbeitung, die Speicherung und den Transport der Daten, die die Objekte der fachlichen Ebene repräsentieren.

Rechnerbasierte Anwendungsbausteine werden durch *Anwendungsprogramme* gesteuert, die ihrerseits durch das Customizing von *Softwareprodukten* (die man kaufen kann) entstehen. Wenn Softwareprodukte den Anforderungen in [Turowski'02] genügen, können sie als *Fachkomponenten* aufgefasst werden. Ein einziges Softwareprodukt in einem Krankenhaus kann auch mehrfach installiert und dabei jeweils unterschiedlichem und jeweils erneut auch sehr teurem Customizing unterzogen werden. Dies macht es für das Informationsmanagement sehr wichtig, sorgfältig zwischen diesem Produkt und damit der Fachkomponente auf der einen Seite und den jeweils entstehenden Anwendungsbausteinen auf der anderen Seite zu unterscheiden.

Konventionelle oder papierbasierte Anwendungsbausteine werden durch *Organisationspläne* gesteuert.

In rechnerbasierten Anwendungsbausteinen werden die Daten in *Datenbanksystemen* gespeichert. *Kommunikationsschnittstellen* ermöglichen die Kommunikation zwischen Anwendungsbausteinen



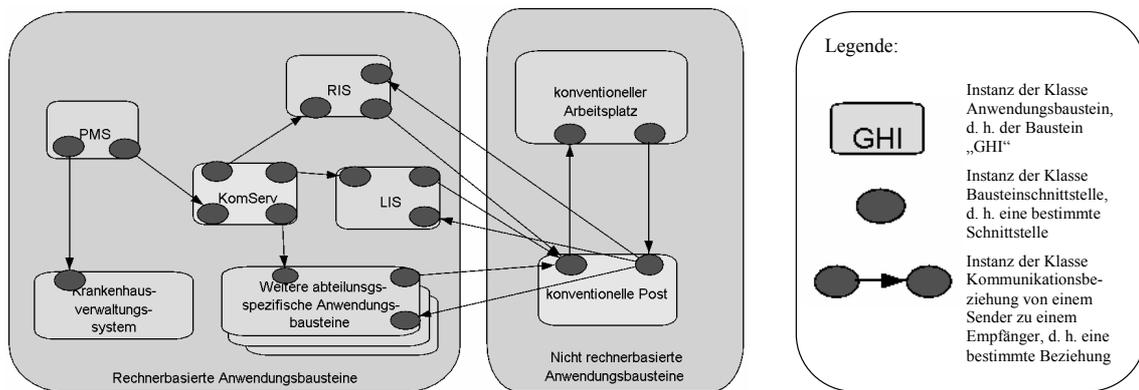


Abbildung 4: Beispiel einer logischen Werkzeugebene

*Anwendungsbaustein-Konfigurationen* dienen der Herstellung der Beziehung zur fachlichen Ebene und werden wie die ebenfalls hierzu dienenden gepunkteten Klassen und Assoziationsbeziehungen in Abbildung 3 weiter unten erläutert. Anwendungsbaustein-Konfigurationen entsprechen „Komponenten-Konfigurationen“ [Turowski'02] nach Installation und Customizing.

Ein Beispiel einer logischen Werkzeugebene eines KIS findet sich in Abbildung 4. Die abgerundeten Rechtecke zeigen hier Anwendungsbausteine, die kleinen Ovale innerhalb der Anwendungsbausteine zeigen die Bausteinschnittstellen. Die gerichteten Kanten zwischen Bausteinschnittstellen unterschiedlicher Anwendungsbausteine repräsentieren die Kommunikationsbeziehungen.

Der linke Teil der Abbildung zeigt den rechnerunterstützten Teil eines Krankenhausinformationssystems: ein Patientenverwaltungssystem (PMS), ein Radiologie-Informationssystem (RIS), ein Labor-Informationssystem (LIS), ein Kommunikationsserver (KomServ), ein Anwendungsbaustein für die Krankenhausverwaltung und weitere nicht näher spezifizierte Abteilungsinformationssysteme. Der rechte Teil zeigt konventionelle Anwendungsbausteine, die konventionelle Post und einen konventionellen klinischen Arbeitsplatz

Dieses Beispiel ist vereinfacht und fiktiv. Allerdings zeigt es eine ganz typische Situation. Während die Krankenhausverwaltung (hier Patientenmanagementsystem PMS) und die Funktionsbereiche durch rechnerbasierte Anwendungsbausteine (hier Laborinformationssystem LIS, Radiologieinformationssystem RIS) unterstützt werden, existieren für typische klinische Aufgaben lediglich papierbasierte Anwendungsbausteine. Für die Kommunikation zwischen den rechnerbasierten Anwendungsbausteinen ist ein Kommunikationsserver (message queuing system) verfügbar, der aber offensichtlich nicht von allen Anwendungsbausteinen genutzt wird oder genutzt werden kann. Als Konsequenz werden auch proprietäre Bausteinschnittstellen benötigt. Die Tatsache, dass typische klinische Aufgaben nicht durch rechnerbasierte Anwendungsbausteine unterstützt werden, erfordert viele Bausteinschnittstellen zwischen dem rechnerunterstützten Teil des KIS und dem nicht rechnerunterstützten Teil, die zu Problemen beispielsweise hinsichtlich Medienbrüchen, führen können. Solche Digital-Analog-Schnittstellen werden z.B. durch Drucker und Belegleser bzw. der entsprechenden Software realisiert

Die **physische Werkzeugebene** (siehe Abbildung 5) stellt die physischen Datenverarbeitungsbausteine bereit, die für den Betrieb von Anwendungsbausteinen erforderlich sind. Ein **physischer Datenverarbeitungsbaustein** ist entweder ein System von Personen und konventionellen Werkzeugen (wie z.B. Archivregale, Posteingangs- und Postausgangskörbe) der Informationsverarbeitung (nicht-rechnerbasierter physischer Datenverarbeitungsbaustein) oder ein rechnerbasierter physischer

Datenverarbeitungsbaustein, z.B. Rechnersysteme (PCs, Server), Monitore, Drucker, Switches, Router, Subnetze, Netzwerkkarten, Netzanschlusspunkte).

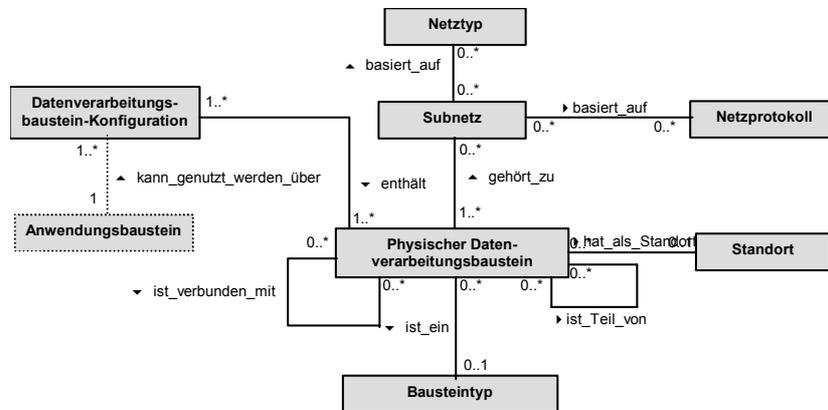


Abbildung 5: Klassendiagramm 'Physische Werkzeugebene'. Gepunktete Linien und Symbole repräsentieren Inter-Ebenen-Beziehungen.

Die physischen Datenverarbeitungsbausteine können über *Datenübertragungsverbindungen* miteinander kommunizieren. Die Konstellation dieser Datenübertragungsverbindungen führt zu physischen Netzwerken, die auf *Netzprotokollen* basieren. *Subnetze* können definiert werden als Projektion auf das gesamte Netzwerk. Es ist zu beachten, dass auf der physischen Werkzeugebene sowohl 'echte' physische Netzwerke beschrieben werden können, als auch 'logische Netzwerke'.

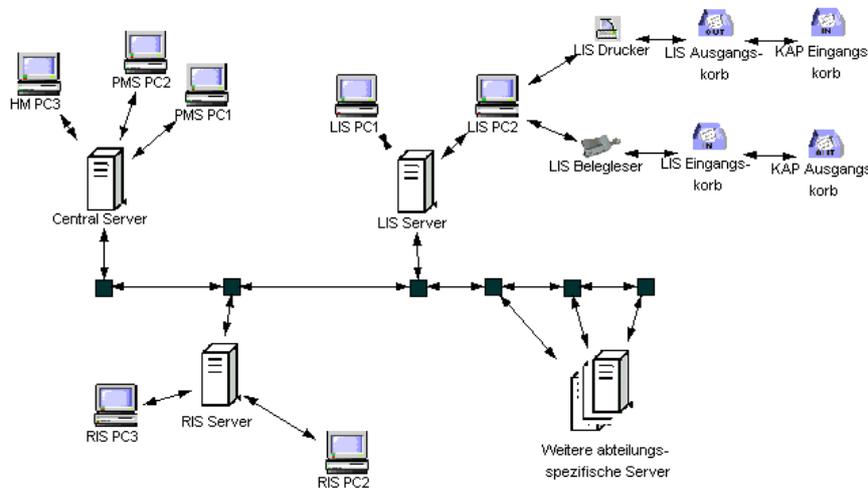


Abbildung 6: Beispiel einer physischen Werkzeugebene

Abbildung 6 zeigt ein Beispiel einer physischen Werkzeugebene. Es gibt je einen Server für die verschiedenen Anwendungsbausteine der Funktionsabteilungen: einen LIS Server, einen RIS Server und die Server für die nicht weiter spezifizierten Anwendungsbausteine, und einen zentralen Server, auf dem u.a. das PMS installiert ist. Jeder Server ist verbunden mit einigen PCs. Die schwarzen Punkte repräsentieren Netzwerkkomponenten, über die die Server miteinander verbunden sind und damit das Gesamtnetzwerk bilden. Die Datenübertragung vom und zum papierbasierten Teil das KIS wird hier lediglich am Beispiel der LIS dargestellt. Eine Laboranforderung (als Dokument) geht über den

Postausgangskorb des klinischen Arbeitsplatzes zum Posteingangskorb des Labors und wird von einem Formularleser eingelesen. Das Ergebnis (also der Befund) wird ausgedruckt und geht wiederum über den Ausgangskorb des Labors zum Eingangskorb des klinischen Arbeitsplatzes. Im Beispiel in Abbildung 6 sind keine Subnetze spezifiziert, und es werden keine Informationen über den Netztyp, das Netzprotokoll oder den Standort dargestellt.

Zwischen den drei Ebenen existieren vielfältige Beziehungen (*Inter-Ebenen-Beziehungen*). Diese Beziehungen können genutzt werden, um Schwächen der logischen oder physischen Werkzeugebene zu entdecken

Zwischen fachlicher Ebene und logischer Werkzeugebene bestehen die wichtigsten Beziehungen zwischen den Aufgaben und den Anwendungsbausteinen. Eine Aufgabe benötigt möglicherweise einen einzelnen Anwendungsbaustein oder mehrere Bausteine gemeinsam. Eine *Anwendungsbaustein-Konfiguration* enthält die Anwendungsbausteine, die gemeinsam für eine Aufgabe erforderlich sind. Es kann sein, dass eine Aufgabe alternativ durch mehrer *Anwendungsbaustein-Konfigurationen* unterstützt werden kann. Eine solche Situation kann auf Redundanz hinweisen.

Weitere wichtige Inter-Ebenen-Beziehungen sind:

- *'hat\_führendes'*: Für den Fall redundanter Speicherung eines Objekttyps, wird dargestellt, welches Datenbanksystem bzw. welche Dokumentensammlung für die Aktualität und Korrektheit der Daten dieses Objekttyps verantwortlich ist und wo diese Daten geändert werden dürfen.
- *'wird ausgelöst durch eine Aktivität der'*: Die Erledigung der zugeordneten Aufgabe löst ein Ereignisses des zugeordneten Ereignistyps aus. Auf der Basis dieses Ereignisses wird dann ggf. eine Kommunikation ausgelöst.
- *'wird repräsentiert durch'*: Informationen über Objekte eines Objekttyps können durch einen *Datensatztyp*, *Dokumententyp* oder *Nachrichtentyp* repräsentiert werden. Datensätze werden in Datenbanksystemen und Dokumente in Dokumentensammlungen gespeichert. Nachrichten dienen dem Transport der Daten.

Die *'Datenverarbeitungsbaustein-Konfiguration'* repräsentiert eine Beziehung zwischen Anwendungsbausteinen und Physischen Datenverarbeitungsbausteinen. Sie drückt aus, dass ein Anwendungsbaustein auf einem oder auch auf mehreren Physischen Datenverarbeitungsbausteinen gemeinsam installiert sein kann (z.B. typische Client/Server Installation). Würde man aus einer Datenverarbeitungsbaustein-Konfiguration einen Datenverarbeitungsbaustein entfernen, wären die verbleibenden nicht mehr für den Anwendungsbaustein ausreichend. Gibt es mehrere Datenverarbeitungsbaustein-Konfiguration, kann eine dieser Konfigurationen alternativ genutzt werden. Andererseits zeigt die Existenz mehrere Konfigurationen Redundanz auf.

## 4 Der 3LGM<sup>2</sup>-Baukasten zur Modellierung von Krankenhausinformationssystemen

Der 3LGM<sup>2</sup>-Baukasten ist eine Software für ein graphisches Modellierungswerkzeug, mit dem Informationsmanager graphische Modelle von Krankenhausinformationssystemen erstellen können, die exakt dem 3LGM<sup>2</sup> Meta-Modell entsprechen. Der 3LGM<sup>2</sup>-Baukasten ist für alle Plattformen verfügbar, die Java 1.3.1 unterstützen.

Ähnlich wie andere graphische Modellierungswerkzeuge präsentiert der Baukasten das Modell über einen Modellbrowser und einen graphischen Darstellungsbereich. Die Dateneingabe und Steuerung wird durch verschiedene (Auswahl-) Menüs und Suchdialoge unterstützt (Abbildung 7).

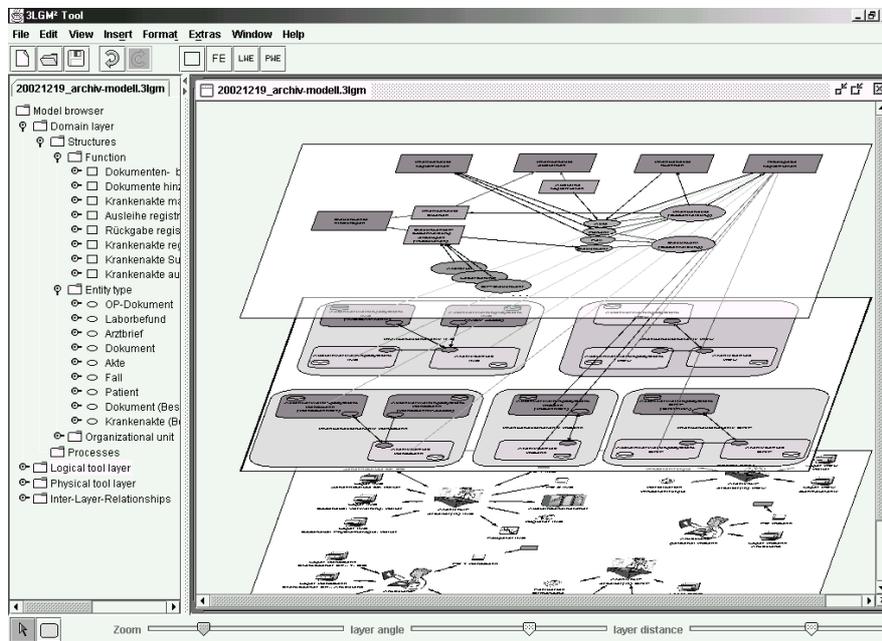


Abbildung 7: Benutzungsoberfläche des 3LGM<sup>2</sup>-Baukastens

Alle drei Ebenen des Modells (siehe Kapitel „3LGM<sup>2</sup>“) können sowohl separat als auch in einer Mehr-Ebenen-Sicht betrachtet und bearbeitet werden. Auf diese Weise wird ein guter Überblick über das gesamte Modell vermittelt und es können auch die Inter-Ebenen-Beziehungen dargestellt werden. Die strikte Ausrichtung an dem Meta-Modell ermöglicht bei der Modellierung einen permanenten Konsistenz-Check, der den Modellierer vor formalen Fehlern bewahrt.

3LGM<sup>2</sup>-Modelle werden im XML-Format gespeichert und sind damit auch für andere Anwendungsprogramme zugänglich. Darüber hinaus können die graphischen Darstellungen im GIF oder TIFF Format ausgegeben werden. So können die Modelle z.B. in Planungsunterlagen für Integrationsprojekte oder in einen Strategischen Plan für das Informationssystem integriert werden.

Es ist ein Auswertungswerkzeug realisiert, mit dem für ein gegebenes Modell z.B. die Frage beantwortet werden kann, „welche Aufgaben des Krankenhauses werden eigentlich durch die Server unterstützt, die im Raum 104 der Zentralgebäudes untergebracht sind?“

Das in Abbildung 7 gezeigte Modell stammt aus einem Projekt zur Einführung eines Dokumenten-Management- und Archivierungs-Systems (DMAS) am Universitätsklinikum Leipzig AöR. Hintergrund ist das Bestreben, im klinischen Bereich die Patientenakte und im administrativen Bereich die dortigen Dokumente digital verfügbar zu machen und digital zu archivieren.

In dem Modell wird der Ist-Zustand des Teil-Informationssystems des Klinikums dargestellt, das sich in den Kliniken für Innere Medizin, Kinderchirurgie, Konservative Zahnheilkunde und Orthopädie mit der Archivierung von Krankenaktenunterlagen befasst. Das Modell hilft zu erkennen:

- Auf der fachlichen Ebene unterscheiden sich die in den verschiedenen Präsenz-Archiven der genannten Kliniken zu erledigenden Aufgaben nicht.

- Auf der logischen Werkzeugebene werden die Archive als konventionelle Anwendungsbausteine dargestellt, die zum Teil auch rechnerunterstützte Bausteine (z.B. Aktenverwaltungssystem INZ (INZ/Access)) enthalten. Obwohl die Aufgaben identisch dargestellt sind, werden unterschiedliche Bausteine in den Kliniken zur Erledigung der Aufgaben eingesetzt.
- Auf der physischen Werkzeugebene wird deutlich, dass die Archivmitarbeiterinnen klar im Zentrum des Geschehens stehen und nur vereinzelt Arbeitsplatzrechner im Einsatz sind. Die Kommunikation der Rechner mit den Lagern für die Akten bzw. den Karteikästen mit den Aktennachweisen erfolgt über diese Mitarbeiterinnen.

## 5 Diskussion

Das vorgestellte Meta-Modell 3LGM<sup>2</sup> dient als Basis des 3LGM<sup>2</sup>-Baukastens, mit dem Modelle von KIS für das strategische Informationsmanagement erstellt werden können. Allerdings konnte bislang noch nicht gezeigt werden, dass dieser Ansatz außerhalb unserer Arbeitsgruppe und in anderen Krankenhäusern einen sichtbaren Nutzen bringen kann. Hierzu ist eine weitergehende Evaluation zur Prüfung der Frage erforderlich, ob die von dem Werkzeug angebotenen Funktionen zur Bildung von Teilmodellen auch große und komplexe Modelle beherrschbar machen.

3LGM<sup>2</sup> ist nicht krankenhausspezifisch, da auch die Informationssysteme von Krankenhäusern sich von Informationssystemen anderer Branchen nicht grundsätzlich unterscheiden. Die Besonderheiten von Krankenhaus-Informationssystemen ergeben sich durch besondere Aufgaben und Objekttypen auf der fachlichen Ebene und auf der logischen Werkzeugebene z.B. durch besondere Anwendungsbausteine und Kommunikationsstandards. Sie lassen sich daher sinnvollerweise durch Referenzmodelle [Simoneit'98, Winter+'99] darstellen. Solche Referenzmodelle für z.B. typische Aufgaben, Objekttypen, Anwendungsbausteine, Middleware-Techniken, Standards etc. sind daher auch eine wesentliche Voraussetzung für die praktische Nutzbarkeit des Ansatzes beim Informationsmanagement im Krankenhaus. Folgeprojekte widmen sich daher insbesondere dieser Problematik.

Die bei den Anforderungen genannte Notwendigkeit, Prozesse auf der fachlichen Ebene modellieren und daraus Kommunikationsprozesse auf der Logischen Werkzeugebene abzuleiten wird zur Zeit aufbauend auf [Brigl+'03] realisiert. Es werden Algorithmen implementiert, die auf dem Suchen kürzester Wege in Graphen basieren.

Über die Verwendung des Meta-Modells 3LGM<sup>2</sup> als Grundlage für ein Modellierungswerkzeug hinaus hat 3LGM<sup>2</sup> Bedeutung als ein semantisches Netz von Begriffen und damit als Ontologie. Die in dieser Ontologie bereitgestellten Begriffe können auch in der natürlichen Alltagssprache von Informationsmanagern verwendet werden, um Aussagen wie „Das System ist nicht verfügbar“ zu präzisieren. Konsequenterweise wird die Ontologie auch bei der akademischen Ausbildung von Medizinischen Informatikern an der Universität Leipzig und der Privaten Universität für Medizinische Informatik und Technik Innsbruck eingesetzt und ist die terminologische Basis zweier Lehrbücher über Krankenhausinformationssysteme [Winter+'02][Haux+to appear].

## 6 Danksagungen

Diese Arbeit entstand im Forschungsprojekt „Integrative Modellierung von Strukturen und Prozessen von Krankenhausinformationssystemen“, das durch die Deutsche Forschungsgemeinschaft (DFG) gefördert wird (WI 1605/2-1).

## 7 Literaturverzeichnis

- [Austin+95] Austin C, Trimm J, Sobczak P. Information systems and strategic management. *Health Care Manage Rev* 1995;20(3):26-33.
- [Brenner94] Brenner W. *Grundzüge des Informationsmanagements*. Berlin: Springer; 1994.
- [Brigl+03] Brigl B, Wendt T, Winter A. Modeling interdependencies between business and communication processes in hospitals. In: Baud R, Fieschi N, Beux PL, Ruch P, editors. *The New Navigators: from Professionals to Patients*. Amsterdam: IOS Press; 2003. p. 863-868.
- [Brinkkemper90] Brinkkemper S. *Formalisation of Information Systems Modelling*. Amsterdam: Thesis Publishers; 1990.
- [Brodie+95] Brodie ML, Stonebraker M. *Migrating legacy systems*. San Francisco: Morgan Kaufmann; 1995.
- [CEN/TC251'97] CEN/TC251. *Healthcare Information System Architecture Part 1 (HISA) Healthcare Middleware Layer*. European Prestandard: CEN European Committee for Standardisation; 1997 03.1997. Report No.: prENV 12967-1:1997 Final Draft 2.
- [Chief Information Officer Council'01] Chief Information Officer Council. *A Practical Guide to Federal Enterprise Architecture*. Boston: Chief Information Officer Council c/o Rob C. Thomas, U.S. Customs Service 7681 Boston Boulevard Springfield, VA 22153; 2001 February 2001. <http://www.cio.gov>.
- [Ferstl+94] Ferstl OK, Sinz EJ. *Grundlagen der Wirtschaftsinformatik*. München: R. Oldenbourg; 1994.
- [Ferstl+98] Ferstl OK, Sinz EJ. Modeling of Business Systems Using SOM. In: Bernus P, Mertins K, Schmidt G, editors. *Handbook on Architectures of Information Systems*. International Handbook on Information Systems: Springer; 1998. p. 339 - 358.
- [Gartner Group'98] Gartner Group. *Three Documents for Healthcare IT Planning*. Gartner Group's Healthcare Executive and Management Strategies Research Note; 1998. Report No.: KA-03-5074.
- [Glaser+99] Glaser JP, Hsu LD. *The strategic application of information technology in healthcare organizations*. New York: McGraw-Hill; 1999.
- [Haux+to appear] Haux R, Winter A, Ammenwerth E, Brigl B. *Strategic Information Management in Hospitals*. New York: Springer; to appear.
- [Heinrich'99] Heinrich LJ. *Informationsmanagement: Planung, Überwachung und Steuerung der Informations-Infrastruktur*. München: Oldenbourg; 1999.
- [Heverhagen+01] Heverhagen T, Tracht R. Using Stereotypes of the Unified Modeling Language in Mechatronic Systems. In: *Proc. of the 1. International Conference on Information Technology in Mechatronics, ITM'01*, October 1-3, 2001, Istanbul, UNESCO Chair on Mechatronics. Istanbul: Bogazici University, Istanbul, Turkey; 2001. p. 333-338.
- [Holm+02] Holm T, Lassen JF, Husted SE, Christensen P, Heickendorff L. A randomized controlled trial of shared care versus routine care for patients receiving oral anticoagulant therapy. *J Intern Med* 2002;252(4):322-31.
- [Kuhn+01] Kuhn KA, Giuse DA. From Hospital Information Systems to Health Information Systems. *Methods of Information in Medicine* 2001;40:275-287.
- [Martin'90] Martin J. *Information Engineering, Book II: Planning & Analysis*. Englewood Cliffs: Prentice Hall; 1990.
- [Oberweis+94] Oberweis A, Scherrer G, Stucky W. INCOME/STAR: Methodology and Tools for the Development of Distributed Information Systems. *Information Systems* 1994;19(8):643-660.
- [Object Management Group (OMG)'00] Object Management Group (OMG). *CORBAmed*. <http://www.omg.org/homepages/corbamed/>; 2000.
- [Object Management Group (OMG)'03] Object Management Group (OMG). *OMG Unified Modeling Language Specification: Object Management Group*; 2003 March 2003. Version 1.5, formal/03-03-01.

- [Scheer'94] Scheer A-W. ARIS Toolset: A Software Product Is Born. *Information Systems* 1994;19(8):607-624.
- [Simoneit'98] Simoneit M. Informationsmanagement in Universitätsklinika: Konzeption und Implementierung eines objektorientierten Referenzmodells. Wiesbaden: Gabler Verlag; 1998.
- [Spewak+'92] Spewak SH, Hill SC. Enterprise Architecture Planning: Developing a blueprint for Data, Applications and Technology. New York: John Wiley & Sons; 1992.
- [Strahinger'98] Strahinger S. Ein sprachbasierter Metamodellbegriff und seine Verallgemeinerung durch das Konzept des Metaisierungsprinzips. In: Klaus Pohl AS, Gottfried Vossen, editor. Proceedings Modellierung '98. Münster: Angewandte Mathematik und Informatik, Universität Münster; 1998.
- [Turowski'02] Turowski K, editor. Vereinheitlichte Spezifikation von Fachkomponenten - Memorandum des Arbeitskreises 5.10.3 Komponentenorientierte betriebliche Anwendungssysteme: Gesellschaft für Informatik, Arbeitskreis 5.10.3 "Komponentenorientierte betriebliche Anwendungssysteme"; 2002.
- [van der Haak+'02] van der Haak M, Mludek V, Wolff AC, Bulzebruck H, Oetzel D, Zierhut D, et al. Networking in shared care--first steps towards a shared electronic patient record for cancer patients. *Methods Inf Med* 2002;41(5):419-25.
- [Winter+'95] Winter A, Haux R. A Three-Level Graph-Based Model for the Management of Hospital Information Systems. *Methods of Information in Medicine* 1995;34(4):378-396.
- [Winter+'99] Winter A, Winter A, Becker K, Bott OJ, Brigl B, Gräber S, et al. Referenzmodelle für die Unterstützung des Managements von Krankenhausinformationssystemen. *Informatik, Biometrie und Epidemiologie in Medizin und Biologie* 1999;30(4):173-189.
- [Winter+'01] Winter AF, Ammenwerth E, Bott OJ, Brigl B, Buchauer A, Gräber S, et al. Strategic Information Management Plans: The Basis for systematic Information Management in Hospitals. *International Journal of Medical Informatics* 2001;64(2-3):99-109.
- [Zachman'99] Zachman JA. A framework for information systems architecture (Reprint). *IBM systems journal* 1999;38(2&3):454-470.



# Integration und Evolution ablaufunterstützender Informationssysteme im Krankenhaus

**Richard Lenz**  
**Philipps-Universität Marburg**  
**lenzr@mail.uni-marburg.de**

***Zusammenfassung:** Krankenhausinformationssysteme (KIS) haben die Aufgabe, den kooperativen und arbeitsteiligen Behandlungsprozess in angemessener Weise zu unterstützen. Dabei ist insbesondere zu berücksichtigen, dass die Anforderungen an die Prozessunterstützung sich rasch ändern können. Infolgedessen entsteht einerseits ein Bedarf zur integrierten IT-Unterstützung abteilungsübergreifender Abläufe, andererseits auch ein Bedarf zur flexiblen Evolution und Anpassung von Informationssystemen. Die häufig anzutreffende Strategie, neue Funktionalität durch „best-of-breed“-Komponenten einzukaufen und dann über nachrichtenbasierte EAI-Werkzeuge in eine Gesamtarchitektur einzubeziehen, wird den Anforderungen an eine abteilungsübergreifende Prozessunterstützung aufgrund unzureichender Standards noch nicht gerecht. In diesem Artikel wird ein Ansatz zur kontinuierlichen Systemevolution vorgestellt, bei dem die Zahl der heterogenen Systemkomponenten zugunsten integrierter Komponenten reduziert wird. Ein Kernsystem wird dabei mit Hilfe eines Generatorwerkzeugs zur Erstellung dokumentenbasierter klinischer Anwendungen an die konkreten Anforderungen eines Krankenhauses angepasst. Neben der Reduktion der Zahl der Integrationsinseln und Schnittstellen besteht eine wesentliche Zielsetzung in der bedarfsgerechten Einbettung der Softwareentwicklung in einen kontinuierlichen Managementprozess sowie in der Förderung der Benutzerakzeptanz durch ein interdisziplinäres und partizipatives Vorgehensmodell*

## 1 Anforderungen an Informationssysteme im Krankenhaus

Krankenhausinformationssysteme (KIS) werden häufig als Schlüssel zur Ablaufoptimierung im Krankenhaus angesehen. Dabei ist insbesondere zu berücksichtigen, dass die zu unterstützenden Abläufe das Ergebnis einer eingespielten sozialen Interaktion im Kontext einer komplexen Organisation darstellen. Die Abläufe im Krankenhaus sind typischerweise interdisziplinär, hochgradig arbeitsteilig, abteilungsübergreifend und selten automatisierbar [1]. Im Rahmen der Patientenbehandlung wird an vielen Orten von vielen Beteiligten patientenbezogene Information erfasst, die in den korrekten Kontext zu bringen ist und in geeigneter Form am „point of care“ bereitzustellen ist. In der Konsequenz lautet eine zentrale Forderung an Krankenhausinformationssysteme: *Integration*. Ziel ist hierbei die Vermeidung von Mehrfacheingaben, die Minimierung von Inkonsistenzen durch Reduzierung unkontrollierter Redundanz sowie die Herstellung einheitlicher Konzepte zur Benutzerinteraktion (Single System Image). Die konkreten funktionalen Anforderungen an Krankenhausinformationssysteme sind nicht in allen Häusern gleich und ändern sich zudem fortwährend. So ändern sich beispielsweise die Rahmenbedingungen für die Prozessunterstützung sehr häufig. Durch die Einführung der DRGs (Diagnosis Related Groups) wird z.B. vom Gesetzgeber ein neues pauschalisierendes Abrechnungssystem vorgegeben, das sich gravierend auf die Optimierung prozessrelevanter Zielgrößen (z.B. Liegezeiten) auswirkt. Somit sind die Möglichkeiten zur Anpassbarkeit und Weiterentwicklung eines Systems ebenfalls wesentliche Forderungen, die vor allem die Struktur und den Aufbau (Architektur) eines KIS betreffen.

## 2 Kontinuierliche Weiterentwicklung von Informationssystemen

Die Fähigkeit eines Systems zur kontinuierlichen Erweiterbarkeit beruht vor allem darauf, bestimmte Funktionen in mehr oder weniger autonomen Komponenten zu kapseln, die bei Bedarf zum Gesamtsystem ergänzt, ausgetauscht oder verändert werden können, ohne dass dadurch der Rest des Systems allzu sehr betroffen wäre. Die Umsetzung einer solchen Komponenten-Architektur in die Praxis lässt allerdings eine große Bandbreite an Strategien zur Systemevolution zu. Eine weit verbreitete Strategie der Systemevolution beruht beispielsweise darauf, bedarfsorientiert am Markt Komponenten einzukaufen, die eine bestimmte Funktionalität abdecken sollen (z.B. ein System zur OP-Dokumentation oder Abteilungsinformationssysteme für z.B. Pathologie oder Kardiologie). Diese Methode hat vor allem den Nachteil, dass unabhängig voneinander entwickelte autonome Komponenten typischerweise weder in ihrer Funktionalität noch in der Semantik der zu handhabenden Daten aufeinander abgestimmt sind. Die nachträgliche Integration dieser Komponenten erfordert daher im Allgemeinen einen sehr hohen technischen Aufwand, wobei nicht selten in die Implementierung der Komponenten eingegriffen werden muss. Die Aufgabe des IT-Personals im Krankenhaus liegt dabei vor allem bei der Identifikation auszutauschender Daten zwischen verschiedenen Systemen, der Abbildung verschiedener Schnittstellen aufeinander, sowie der technischen Umsetzung der entsprechenden Datenströme. EAI-Werkzeuge, wie z.B. spezielle Message Broker, können diese Arbeit sehr erleichtern. Vor allem aber sind standardisierte Schnittstellen erforderlich, die den Integrationsaufwand und die semantische Abstimmung der verschiedenen Komponenten erheblich erleichtern. Im Krankenhausumfeld hat sich in diesem Zusammenhang vor allem der nachrichtenbasierte Standard HL7 [2] etabliert. HL7 basiert auf einem umfassenden Katalog von nachrichtenauslösenden „Events“ und zugehörigen Nachrichtenformaten. HL7 ist aber kein Plug-and-Play-Standard: Kaum ein Hersteller hat den kompletten Standard implementiert, sondern bietet üblicherweise nur häufig vorkommende Formate an. Eine Anwendungsintegration auf der Basis von HL7 erfordert immer die Definition einer Integrationsstrategie, wobei der Standard eine gewisse Bandbreite erlaubt, indem beispielsweise benutzerdefinierte Felder und Segmente und unterschiedliche Propagierungsstrategien ermöglicht werden (z.B. Push-Strategie oder Pull-Strategie). Obwohl die Verbreitung des HL7 Standards bereits wesentlich zur Reduzierung der semantischen Inkompatibilität heterogener Subsysteme im Krankenhaus beigetragen hat, sind die Integrationsprobleme auch unter Verwendung von EAI-Werkzeugen aus verschiedenen Gründen noch nicht gelöst:

- Ein nachrichtenbasierter Datenaustausch zwischen weitgehend autonomen Subsystemen stellt noch keine einheitlichen Benutzerschnittstellen und Interaktionsformen sicher (z.B. auch noch kein „single sign on“).
- Die visuelle Integration verschiedener Anwendungen erfordert neben dem (asynchronen) Austausch von Nutzdaten auch die Synchronisation von Kontextdaten – d.h. wenn in einer Applikation der Kontext verändert wird (z.B. Benutzer, Patient, Fall, etc.), so muss dies mit den anderen aktiven Applikationen synchronisiert werden.
- Die Funktionalität der Teilsysteme ist nach wie vor nicht abgestimmt, so dass viele Funktionen redundant in vielen Systemen vorkommen (Funktionale Überlappung). Typische redundante Funktionen sind z.B. Autorisierung und Patientenidentifikation.
- Nicht zuletzt durch die überlappende Funktionalität der Komponenten kommt es zu redundanten Datenbeständen, die durch einen asynchronen nachrichtenbasierten Datenaustausch nur unzureichend synchronisiert werden.
- Die Unterstützung abteilungsübergreifender Abläufe wird erschwert, wenn dabei Systemgrenzen überschritten werden müssen.

Die Integration heterogener Systemkomponenten erfordert die Spezifikation einer Systemarchitektur, bei der wesentliche Funktionalitäten zentral zur Verfügung gestellt werden müssen. Mindestvoraussetzung ist

eine systemweit eindeutige Patientenidentifikation, die erforderlich ist, um eine falsche oder unvollständige Zuordnung von Patientendaten zu vermeiden („Master Patient Index“). Weitere zentrale Komponenten werden je nach Integrationsgrad benötigt – z.B. für die Autorisierung, die Kontextverwaltung (vgl. [3]) und die Sicherstellung der terminologischen Konsistenz [4]. Jedes einzubindende Teilsystem erfordert eine Abstimmung mit mindestens diesen zentralen Komponenten, was einen sehr hohen Anpassungs- und Integrationsaufwand verursacht. Standardisierungsbestrebungen zur Erleichterung dieser Integration betreffen nicht nur die Syntax und Semantik auszutauschender Daten sondern auch grundlegende Annahmen über die Komponentenarchitektur eines Krankenhausinformationssystems. CCOW („Clinical Context Object Working Group“) bezeichnet beispielsweise einen aufkommenden Standard der für die Kontextsynchronisation einschliesslich „Single sign on“ für heterogene Applikationen sorgen soll. Ein derartiger Standard muss naturgemäss Annahmen über die Rolle zentraler Systemkomponenten machen und damit Rahmenbedingungen für eine CCOW-konforme Systemarchitektur abstecken. CCOW-konforme Applikationen müssen in der Lage sein, in eine solche Architektur eingebettet zu werden (und damit einen Teil der eigenen Autonomie aufzugeben). Die meisten der heute erhältlichen autonomen Anwendungssysteme weisen eine entsprechende Kompatibilität noch nicht auf und müssen mit entsprechendem Aufwand nachträglich an eine vorgegebene Architektur angepasst werden.

Der hohe Integrationsaufwand kann gespart werden wenn Anwendungskomponenten von Anfang an so entwickelt werden, dass sie aufeinander abgestimmt sind. Dazu ist ein Rahmenwerk erforderlich, das bei der Entwicklung von neuen Anwendungskomponenten verwendet wird, um die zu entwickelnden Komponenten in ein laufendes System einzubetten. Hierbei werden also nicht mehr fertige Komponenten gekauft und nachträglich in eine heterogene Systemlandschaft integriert, sondern es werden neue Komponenten entwickelt, die von Beginn an in eine weitgehend homogene Systemlandschaft eingebettet werden. Damit der Anwendungsentwickler sich bei der Neuentwicklung solcher Komponenten nicht in erster Linie mit technischen Fragestellungen, sondern vorwiegend mit der bedarfsorientierten Umsetzung der funktionalen Anforderungen der End-Anwender befassen kann, ist die Verwendung eines entsprechenden RAD-Werkzeugs (RAD = Rapid Application Development) von großem Vorteil. Ein derartiges Entwicklungswerkzeug wird am Universitätsklinikum in Marburg eingesetzt, und im nachfolgenden Abschnitt kurz beschrieben.

### **3 Ein Rahmenwerk zur Entwicklung integrierter klinischer Anwendungen**

In Abb. 1 wird die Architektur des Krankenhausinformationssystems schematisch dargestellt. Mit Hilfe des integrierten RAD-Werkzeugs (im Folgenden auch kurz als „Generator“ bezeichnet) wird der Anwendungsentwickler in die Lage versetzt, dokumentenbasierte klinische Anwendungen zu entwickeln, die in das Gesamtsystem integriert sind. Bei der Entwicklung eines Formulars zur Erstellung von Arztbriefen kann der Entwickler z.B. Bezug nehmen auf Daten, die bereits im System vorhanden sind, um diese in neu generierten Formularen zu verwenden und somit Mehrfacheingaben zu vermeiden und das Risiko von Inkonsistenzen zu minimieren. Der Entwickler definiert mit der Spezifikation eines neuen Formulars für Arztbriefe praktisch eine Schablone für neu zu erstellende Arztbriefe. Diese Schablone kann dann zur Generierung von elektronischen Dokumenten (Formular-Instanzen) immer wieder verwendet werden. Zuvor erfasste Diagnosen, Befunde und Massnahmen können automatisch in einen so generierten Arztbrief übernommen werden. Mit Hilfe von Zustandsvariablen kann ein Dokumentenfluss realisiert werden, der ausgenutzt werden kann, um auch abteilungsübergreifende Abläufe bedarfsorientiert zu unterstützen. So kann beispielsweise die Auftragskommunikation zwischen Station und Radiologie über ein entsprechendes elektronisches Anforderungsformular abgewickelt werden, das auf der Station ausgefüllt wird und dann von dort in eine elektronische Arbeitsliste an der radiologischen Leitstelle weitergeleitet wird.

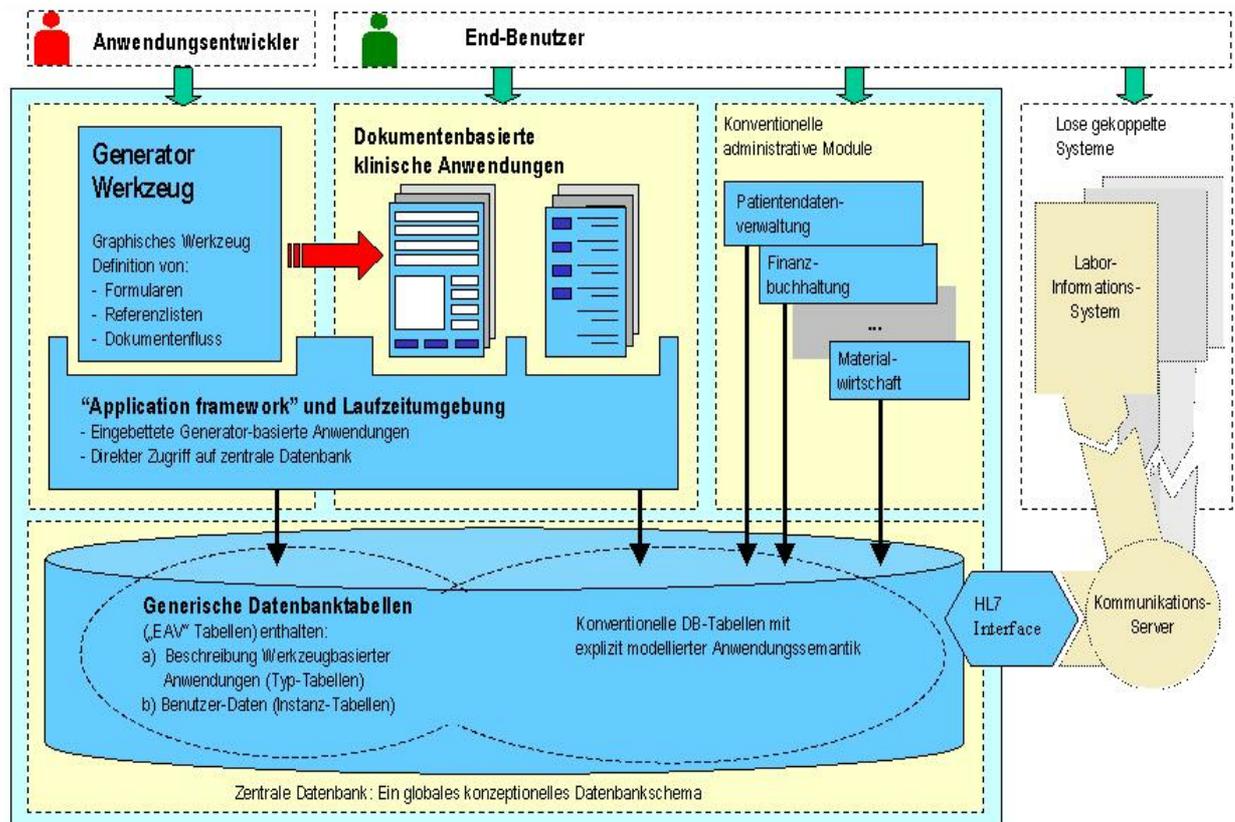


Abb. 1: Architektur eines integrierten erweiterbaren Krankenhausinformationssystems [5].

Der Anwendungsentwickler kann mit Hilfe des Generator-Werkzeugs in vergleichsweise kurzer Zeit Formulare entwickeln, die in ein homogenes Gesamtsystem integriert sind. Autonome, unabhängig entwickelte Anwendungen können nach wie vor über Schnittstellen an das System gekoppelt werden. Ein Message Broker (Kommunikationsserver) erleichtert die Konfiguration und das Schnittstellenmanagement. Importierte Daten aus Subsystemen können ebenfalls in der zentralen Datenbank abgelegt werden und über benutzerdefinierte Formulare im Rahmen des integrierten Systems zugänglich gemacht werden.

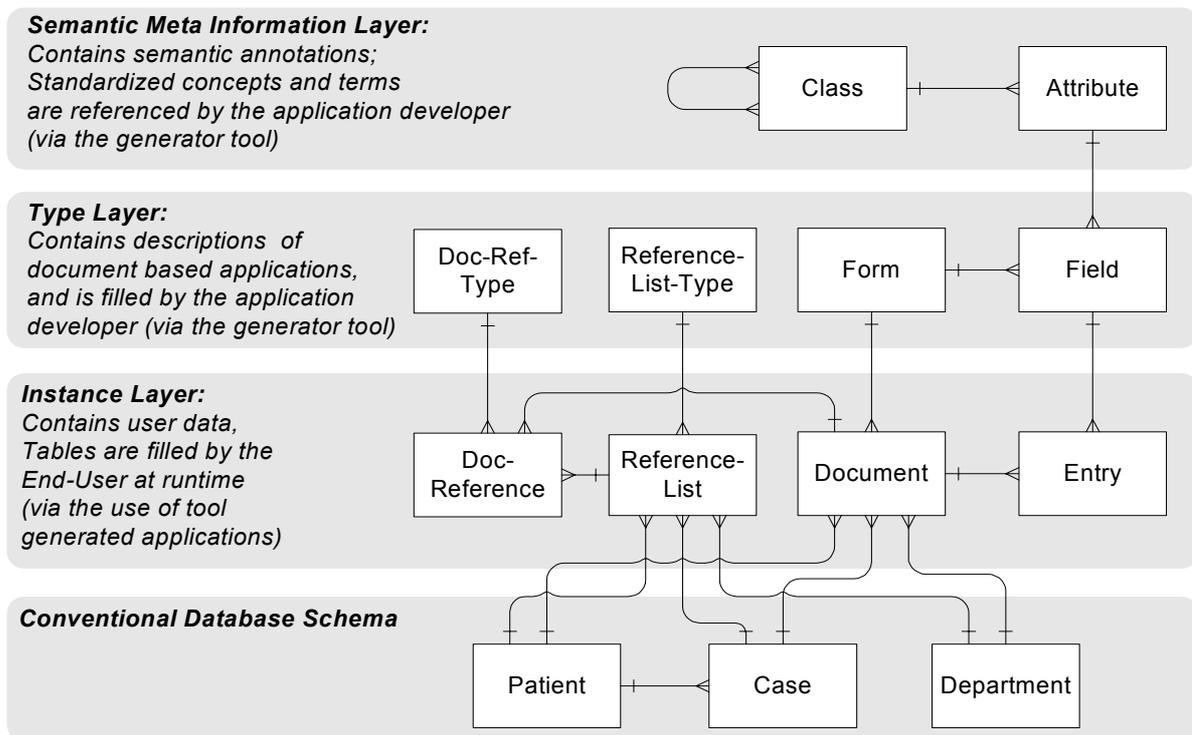


Abb. 2: Generisches Datenbankschema für dokumentenbasierte Anwendungen [6]

Entscheidend für die flexible Erweiterbarkeit des Systems ist die Möglichkeit, neue Formulare (und Formularinhalte) zu definieren, ohne dafür das Datenbankschema der zentralen Datenbank erweitern zu müssen. Dazu muss die zentrale Datenbank generische Tabellen vorhalten, die sowohl die Beschreibung neuer Formulare (Typebene) als auch die Instanzen dieser Formulare abbilden können. Das verwendete Datenbankschema beruht auf einer abgewandelten EAV-Technik („Entity-Attribute-Value“) [7], die es ermöglicht, Daten unterschiedlicher Semantik in eine Tabelle abzubilden. Ein Ausschnitt des generischen DB-Schemas wird in Abb. 2 gezeigt. Eine genauere Beschreibung DB-Schemas und eine eingehende Diskussion der zugrunde liegenden Konzepte findet sich in [5].

## 4 Ein mehrschichtiger Software-Entwicklungsprozess

Die Verwendung des Generator-Werkzeugs zur Anwendungsentwicklung ermöglicht einen mehrschichtigen Ansatz zur Systemevolution, bei dem im Rahmen der Software-Entwicklung verschiedene Rollen unterschieden werden können:

- Der Framework-Entwickler ist für die Funktionalität des Kernsystems und der Laufzeitumgebung zuständig.
- Der Generator-Entwickler ist für die Integration und die Funktionalität des Generatorwerkzeugs verantwortlich.
- Der Anwendungsentwickler ist für die Entwicklung und Anpassung der End-Anwender-Funktionalität zuständig.

Ziel dieser Schichtung ist, den Anwendungsentwickler durch die Wiederverwendung generischer Framework Funktionalität von Codierungs- und Debuggingaufgaben zu entlasten, damit er sich auf die Anforderungsanalyse konzentrieren kann. Durch die Verwendung anwendungsnaher Entwurfsprimitive im Generatorwerkzeug (wie z.B. papierähnliche Formulare) wird die Umsetzung der aus der klinischen Praxis resultierenden Anforderungen beschleunigt und die Fehleranfälligkeit durch die Übersetzung auf ein anwendungsfernes Programmierparadigma reduziert. Die Vorteile der werkzeuggestützten Anwendungsentwicklung kommen aber erst dann zur Geltung wenn die Software-Entwicklung in einen bedarfsorientierten kontinuierlichen Entwicklungsprozess eingebettet wird. Dabei kann auf der Basis moderner iterativer Software-Engineering-Techniken die Optimierung der Abläufe im Krankenhaus stärker in den Vordergrund treten: Die Systemevolution soll nicht mehr technikorientiert sondern bedarfsorientiert erfolgen [6]. Der eigentlichen Softwareentwicklung geht dabei eine Prozessanalyse und -optimierung voraus, damit nicht bestehende Mängel durch die Nachimplementierung der konventionellen eingefahrenen Kommunikationswege sozusagen zementiert werden. Optimierungen und Prozessveränderungen müssen allerdings durch ein sorgfältiges „Change Management“ und eine Fehleranalyse begleitet werden, um die Prozessbeteiligten in geeigneter Weise auf neue oder veränderte Aufgaben im Rahmen der neugestalteten Prozesse vorzubereiten. Die Implementierung von neuen IT-Komponenten ergibt sich dann im Rahmen solcher Optimierungsprojekte aus einem konkreten Bedarf heraus und kann aufgrund der vorangegangenen Prozessanalyse auch bewertet und priorisiert werden. Zur Anpassung der zu erstellenden Software-Komponenten an die tatsächlichen Anforderungen ist dabei eine iterative und partizipative Vorgehensweise zu empfehlen: Die Endanwender werden von vornherein in die Entwicklung neuer Anwendungen mit einbezogen, damit zum einen die tatsächlichen Anforderungen zur Unterstützung der Abläufe besser erfasst und abgebildet werden können und zum anderen auch eine hohe Akzeptanz der resultierenden Applikationen erzielt werden kann. Das verwendete Entwicklungswerkzeug sollte dabei nach Möglichkeit die rasche Erstellung von Prototypen ermöglichen, um auf diese Weise auch möglichst schnell zu Rückmeldungen von den Endanwendern zu gelangen. Damit wird vermieden, dass Fehler bei der Anforderungsanalyse erst zu einem sehr späten Projektzeitpunkt bemerkt werden und dann hohe Kosten verursachen („late design breakage“) [8].

## 5 Integration autonomer Subsysteme

Die konsequente Anwendung der holistischen Strategie zur Systemevolution hat zu einer erheblichen Reduzierung der Zahl der autonomen Anwendungen am Universitätsklinikum in Marburg geführt. Der Ansatz ersetzt jedoch nicht die Notwendigkeit zur Integration autonomer Subsysteme: Viele Teilsysteme im Krankenhaus können nicht durch generatorbasierte Komponenten abgelöst werden, da sie beispielsweise eng mit diagnostischen oder therapeutischen Modalitäten verknüpft sind (z.B. Herzkatheter-Messplatz, Strahlentherapie-Geräte etc.) [9]. Die Ablösung komplexer autonomer Anwendungen, wie z.B. das Laborinformationssystem (LIS), würde zudem einen unverhältnismäßig hohen Aufwand im Vergleich zum Nutzen verursachen. Offenheit und Kompatibilität mit Standards wie HL7 bleiben daher auch für holistische und modular erweiterbare Systeme zentrale Forderungen.

Da das vorgestellte Kernsystem die zentrale Funktion der Patientendatenverwaltung übernimmt, fungiert es im Rahmen der krankenhausweiten Systemarchitektur auch als Master Patient Index: Aufnahmen, Verlegungen, Entlassungen und Modifikationen an identifizierenden Stammdaten erfolgen ausschliesslich über dieses führende System. Abteilungsspezifische autonome Subsysteme werden über eine HL7-Schnittstelle und einen Message Broker an dieses zentrale System gekoppelt. Zur Integration des Laborsystems wurde z.B. eine HL7-Schnittstelle etabliert, welche die strukturierte Übertragung von Laborergebnissen ermöglicht. Generatorbasierte Anwendungen ermöglichen dann die bedarfsorientierte Präsentation der Labordaten in geeigneten Formularen auf der Station, bzw. in den anfordernden Abteilungen. Zur

Umsetzung dieser Lösung waren auch Eingriffe in das Laborsystem erforderlich (z.B. Freigabestatus von Laborergebnissen muss mit im System abgelegt werden).

Zur Integration bildgebender Modalitäten (z.B. CRT, MRT, etc.) werden über HL7 hinaus weitere Standards benötigt. Hier hat sich vor allem DICOM (Digital Imaging and Communication in Medicine) als Standard für medizinische Bilddaten und zugehörige Kontextdaten etabliert [10]. Neben dem Informationsmodell, das den DICOM-Kontextdaten zugrunde liegt, werden im Rahmen des Standards auch Dienste spezifiziert, welche die im Zusammenhang mit dem Bilddatenaustausch erforderliche Kommunikation regeln. Dazu gehört beispielsweise auch die Übermittlung von Arbeitslisten an bildgebende Modalitäten, so dass die zur DICOM-konformen Bilddatenerzeugung erforderlichen Kontextdaten nicht an der Modalität redundant eingegeben werden müssen. DICOM ist aber auch kein Plug-and-Play-Standard, weil es beispielsweise – genau wie HL7 – nicht die funktionale Überlappung der zu integrierenden Teilsysteme vermeidet und nicht regelt, welche Funktionalität von welchem System wahrzunehmen ist. Die Initiative IHE (Integrating the Healthcare Enterprise) versucht auf der Basis der vorhandenen Standards HL7 und DICOM die Integration verschiedener KIS Komponenten zu verbessern [11]. Dabei werden vor allem Integrationsprofile und die daran beteiligten Akteure definiert. Durch die Spezifikation von Akteuren (Rollen) im Zusammenhang mit Integrationsprofilen wird genau festgelegt, welche IHE-Transaktionen wann aufgerufen werden müssen. IHE-Transaktionen spezifizieren die Interaktion zwischen Akteuren, wobei jeweils Kontext, Ablauf und Rollenverhalten festgelegt werden. Eine Integration verschiedener Systemkomponenten auf der Basis von IHE erfordert dann die Aufteilung der erforderlichen Akteure (Rollen) auf die realen Systeme.

Im Rahmen der Einbindung bildgebender Modalitäten ist auch mit der holistischen Strategie zur Systemevolution ein hoher Integrationsaufwand verbunden. Durch die Vorgaben von IHE kann dieser Aufwand zu einem Teil reduziert werden. Desweiteren ist eine Aufwandsreduktion dadurch möglich, dass dem zentralen System ggf. mehrere Rollen zugeordnet werden können, so dass ein Teil der spezifizierten Kommunikationsprotokolle nicht explizit etabliert werden muss.

## 6 Diskussion

Die vorgestellte Systemarchitektur basiert auf einer zentralistischen Strategie zur Systemevolution. Ein hoher Integrationsgrad wird durch Wiederverwendung eines zentralen Kernsystems im Rahmen dokumentenbasierter Anwendungen erreicht. Gleichzeitig wird durch die Verwendung eines RAD-Werkzeugs die Anwendungsentwicklung beschleunigt und in einen kontinuierlichen Managementprozess eingebettet. Dadurch wird eine bedarfsorientierte Steuerung der Systemevolution unterstützt. Die Umsetzung dieses Konzeptes ist derzeit nur auf der Basis eines kommerziellen Systems mit integriertem Generatorwerkzeug möglich. Demzufolge ist als wesentlicher Nachteil dieser Evolutionsstrategie (zumindest heute noch) die hohe Abhängigkeit von einem einzigen Hersteller zu nennen.

Komplementär zu der holistischen Evolutionsstrategie muss immer noch der Weg zur Kopplung autonomer Anwendungen auf der Basis von Standards wie HL7 und DICOM offen bleiben. Während innerhalb des homogenen Kernsystems der Generator Ansatz eine praktikable Möglichkeit darstellt, ablaufunterstützende Anwendungen bedarfsorientiert zu generieren, so sind Möglichkeiten zur flexiblen systemübergreifenden Ablaufspezifikation derzeit noch sehr begrenzt. Mit der Konsolidierung der Komponententechnologie und entsprechender Domain-Standards für das Gesundheitswesen (z.B. CORBAMED [12]) bahnt sich eine neue Entwicklung an, für die aber noch abzuwarten bleibt wie sie von den Herstellern aufgenommen wird. Sollte sich eine Standardisierung von Diensten für Informationssysteme im Gesundheitswesen durchsetzen so ist zu erwarten dass damit auch neue Werkzeuge aufkommen die eine bedarfsorientierte Systemevolution auf der Basis generischer Dienste unterstützen.

## 7 Literaturverzeichnis

- [1] Berg M, Toussaint P. The mantra of modeling and the forgotten powers of paper: a sociotechnical view on the development of process-oriented ICT in health care. *Int J Med Inf* 2003.
- [2] Schadow G, Fohring U, Tolxdorff T. Implementing HL7: from the standard's specification to production application. *Methods Inf Med* 1998; 37(1):119-123.
- [3] Seliger R. Overview of HL7's CCOW Standard.  
[http://www.hl7.org/library/committees/sigvi/ccow\\_overview\\_2001.doc](http://www.hl7.org/library/committees/sigvi/ccow_overview_2001.doc). 2001. Health Level Seven, Inc.
- [4] Cimino JJ. From data to knowledge through concept-oriented terminologies: experience with the Medical Entities Dictionary. *J Am Med Inform Assoc* 2000; 7(3):288-297.
- [5] Lenz R, Elstner T, Siegele H, Kuhn KA. A practical approach to process support in health information systems. *J Am Med Inform Assoc* 2002; 9(6):571-585.
- [6] Kuhn KA, Lenz R, Elstner T, Siegele H, Moll R. Experiences with a generator tool for building clinical application modules. *Methods Inf Med* 2003; 42(1):37-44.
- [7] Nadkarni PM, Marenco L, Chen R, Skoufos E, Shepherd G, Miller P. Organization of heterogeneous scientific data using the EAV/CR representation. *J Am Med Inform Assoc* 1999; 6(6):478-493.
- [8] Versteegen G. *Projektmanagement mit dem Rational Unified Process*. Berlin: Springer-Verlag, 2000.
- [9] Lenz R, Blaser R, Kuhn KA. Hospital information systems: chances and obstacles on the way to integration. *Stud Health Technol Inform* 1999; 68:25-30.
- [10] Bidgood WD, Horii SC, Prior FW, Van Syckle DE. Understanding and Using DICOM, the Data Interchange Standard for Biomedical Imaging. *Journal of the American Medical Informatics Association* 1997; 4(3):199-212.
- [11] Wein BB. [IHE (Integrating the Healthcare Enterprise): a new approach for the improvement of digital communication in healthcare]. *Rofo Fortschr Geb Rontgenstr Neuen Bildgeb Verfahr* 2003; 175(2):183-186.
- [12] Jagannathan V, Wreder K, Glicksman B, alSafadi Y. Objects in Healthcare-focus on standards. *ACM Standards View* 1998;22-26.

# Issues in the Design of Enterprise Architectures: Insights from a Case Study

**Marit Schallert and Michael Rosemann**

**Centre for IT Innovation, Queensland University of Technology,  
Brisbane, Australia**

**{m.schallert|m.rosemann}@qut.edu.au**

***Abstract:** The need to align business and IT-related perspectives motivates an increasing number of organisations to design Enterprise Architectures. A wide range of general frameworks provides valuable guidance for the possible structure of such architecture. More detailed models for the entire architecture lifecycle management are, however, not available. This research explores through a case study at an experienced Australian utility provider the perceived issues in the design, communication, use and maintenance of such an Enterprise Architecture.*

***Keywords:** Enterprise Architecture, Enterprise Design, Enterprise Modelling, Enterprise Integration*

## 1 Introduction

According to a recent survey investigating the modelling practice in Australia the design of Enterprise Architecture was the fourth most popular purpose for undertaking modelling activities (Davies et al., 2002). It seems the design of Enterprise Architectures has never been more popular. Being faced with a continuously growing complexity, organisations (as heterogeneous systems) are realising the critical requirement for accomplishing a mature level of enterprise integration. Only based on the integration of processes, organisational structures, technology, applications and data, the benefits of large implementations of information systems can be obtained. Current developments in the areas of Web Services and Enterprise Application Integration enable organisations on the one hand to respond accordingly to needs of customers and suppliers. On the other hand, however, these developments cause additional complexity and drastically increase the demand for integration and business / IT alignment.

In most publications, Enterprise Architecture is referred to as a comprehensive description of all key elements and relationships that constitute an organisation (Harmon, 2003; O'Rourke, 2003; Armour et al. 1999-1). Thus, the four domains of Business Architecture, Application Architecture, Technology and Information Architecture have to be described to succeed with alignment efforts (Stevenson, 1995). Identifying and modelling the elements of each of these domains in an integrated way is crucial to the success and use of an Enterprise Architecture.

The key contribution of this study is the identification of current issues in the design of Enterprise Architectures. An explorative case study provides insights into the incremental mechanisms and procedures used in practice to design an Enterprise Architecture. The study's secondary contribution is the development of linkages between individual project purposes and the affected Enterprise Architecture domains. This will be used for future research targeting on the development of a universal procedure model for the design of Enterprise Architecture. This paper reports on the findings obtained through a case study conducted at a utility company in Western Australia.

This paper is structured as follows. First, the motivation underlying this research is discussed. The research methodology and the actual case site are described in the following sections. The findings of this case study are discussed in section 5 before a brief section summarises the contribution of this paper.

## 2 Motivation

Manifold approaches exist to describe the elements of an Enterprise Architecture. Reference architectures and frameworks have evolved since the fundamental work of John A. Zachman in 1987 (Zachman, 1987). Many attempts have been made to develop architectures for information systems that provide a framework for integration. Early architectures, such as the Network Application Support (NAS) of Digital Equipment Corp. and Systems Application Architecture (SAA) of IBM, have been developed by hardware and software vendors. Other architectures, such as CIM-OSA, the Computer Integrated Manufacturing-Open Systems Architecture are outcomes of comprehensive research projects (Petrie, 1992). Finally, many architectures have been derived by modifications of the Zachman Framework. Overall, more than 25 frameworks and reference architectures can be identified. The most of them are derivatives of the Information System Architecture Framework (ISA) of Sowa and Zachman in 1992 (Bernus et al., 1996; Sowa, 1992; Vernadat, 1996).

Although significant research has been undertaken in the structure and components of universal frameworks for Enterprise Architectures, only a minority focuses on the actual Architecture Lifecycle Model organisations should follow when they develop a corporate Enterprise Architecture. Furthermore, an analysis of the critical success factors and success measures of an Enterprise Architecture is still missing. Most publications in this area only define general steps, such as

1. agreement on the need and drivers for an Enterprise Architecture,
2. selection of framework, tool and repository,
3. actual design of the architecture,
4. application of the architecture, and
5. architecture extension and maintenance.

These steps can be referred to as Enterprise Architecture Lifecycle (Harmon, 2003; Armour et al. 1999-1). Such a universal lifecycle does not spell out, however, that in reality an Enterprise Architecture is often designed as a sequence of incremental projects over time. Thus, designing an Enterprise Architecture has to be seen as a continuous process rather than a resultant product. Armour et al. refer to building an Enterprise Architecture as a five-year project; this is why organisations often do not take a systematic design approach (Armour et al. 1999-2).

## 3 Research Methodology

A comprehensive literature review was conducted at the initial stage of this research in order to identify and analyse all reference architectures, frameworks and procedure models for the design of Enterprise Architecture. Related work on lifecycle models can only be found in some of the developed architecture

frameworks, such as the Generic Enterprise Reference Architecture and Methodology (GERAM) and CIM-OSA (Bernus, 1998).

A case study has been selected as the appropriate research methodology because of the lack of theoretical models and the explorative nature of this research. The case study was conducted at a leading Australian utility provider. The organisation was chosen due to its relative high degree of maturity and its leading role in the design of an Enterprise Architecture in Australia. The organisation is the owner of the Corporate Information Management Model (CIMod). The corporate Enterprise Architecture repository of CIMod contains currently more than 15,000 objects and 20,000 relationships in approx. 2,000 models. The design of the architecture started in 1998, while an ERP system was implemented.

Particular emphasis of this explorative case study laid on (a) the organisation's objectives and requirements for the Enterprise Architecture, and (b) on the design approach. Within the design approach the following key elements were investigated

- the identification of elementary architecture components
- the actual procedural modelling approach taken
- the critical success factors for the design.

Objective of this paper is to collect findings about perceived shortcomings and critical issues of the utilised Enterprise Architecture framework and limitations of the design approach. This will contribute to the development of a general procedure model for the design of Enterprise Architectures.

## 4 The Case Site

The case study was conducted at Water Corporation, in Perth, Western Australia. The Water Corporation produces annually 323,477 mega litres of water and provides water and wastewater services to 0.8M customers. With a total annual revenue in 2001/2002 of 1,051M AUD, 2,084 employees and 9,579M AUD invested in water services assets and infrastructure the Water Corporation is one of Australia's largest water service providers.

The organisational structure of the Water Corporation reflects a high customer and process-oriented focus. The organisational areas are Finance, Engineering Contracts, Planning & Development, Customer Services and Bulk and Waste Water. These areas are concerned with the service and output they deliver, rather than with the associated functional areas. In order to deliver quality and trusted water services the Information Planning branch, which was initially assigned to the Planning & Development division within Water Corporation, has a strong focus on improving the access to information supplied to customers and business partners. Information technology is seen as an enabler that allows efficiently capturing data and converting it into useful information. Water Corporation has extensive application architecture with over 300 systems. The organisation has 45 supporting ERP applications and 54 core business applications. The focus of the IT Management lies on a throughout integration of the numerous applications and alignment of the above service areas with modern IT systems.

The data collection for the case study is based on nine interviews with sponsors, designers and users of the Corporate Information Management Model and the detailed analysis of over ten project related documents, as well as the Enterprise Architecture repository of CIMod.

## 5 The Findings

### 5.1 The organisation's objective and requirement for Enterprise Architecture

*1. The original Information Architecture evolved to an Enterprise Architecture within less than five years due to large implementations of information systems, predominately driven by ERP implementations.*

The Water Corporation started since its inception in 1995 various IT-driven developments, such as the comprehensive implementation of ERP software, and Customer Service Software as well as upgrades to desktop and server technologies. These developments were driven by the business requirement to replace aging technologies and the Y2K issues. After Y2K, immediate efforts were taken by the Corporate Information Support Branch to assess the information architecture and receive advice on architecture content, usage, documentation, promotion and governance. The objective was to ensure that IS developments do not result in silo solutions and fragmentation, increased system integration costs and lack of interoperable applications. For this reason an EA as a framework was established in 2001 to ensure that individual developments are planned, designed and built complementary (non redundant) in functionality and purpose, that data are effectively shared and the overall business is supported.

*2. Enterprise Architecture (EA) requires top management support. The strategic architecture sponsor is convinced to utilise an EA for management purposes of planning and control.*

Efforts to establish an EA were initiated within the planning and development environment of the organisation. Such a strategic organisational assignment seems to be more than appropriate to push through enterprise-wide integration and alignment, but also acceptance of the corporate Enterprise Architecture. Results of the case study have shown that the responsibility for an EA can not be one-sided; EA is not a question of either Business or IT sponsorship, but requires the alignment of both.

*3. Process and goal documentation as well as measures for process innovation and Key Performance Indicators have to be integral part of the Enterprise Architecture.*

The increasing strategic role of an EA sets a new perspective for traditional frameworks. Key Performance Indicators measuring the performance of all architecture domains are essential for the continuous evaluation of the usefulness of an EA for planning and control. The case study indicated that the organisation is confronted with difficulties integrating motivation, goals and performance on all design levels of the architecture. Desirable would be performance measurements and assessments of architecture domains against each other, such as Application Architecture against Business Architecture. However, the design and integration of a Motivation Architecture can already be found in the approach of Sowa and Zachman. The new challenge is linking the EA to performance measurement approaches, such as the Balance Scorecard approach.

Summarising for the driving objectives of an organisation to design and implement an Enterprise Architecture it can be concluded that the original design leads back primarily to IS integration efforts, but finds increasing focus in its true purpose and high potential for planning, decision making and control. This emphasises the new attitude towards the actual value of an EA – it does not automatically add value by designing it on an operational level, the design is only a precondition for providing strategic value to the organisation (Boster et al. 2000).

## 5.2 The Enterprise Architecture design approach

4. *Enterprise Architecture design efforts should be incrementally taken; driven by projects associated with the architecture domains and levels. High contextual and conceptual levels of the Enterprise Architectures should be established in initial enterprise wide design effort which serves as starting point for all sub design efforts*

The question the CIMod designer faced in 1998 was “It is great to have such a framework, but how to fill it out?” In the initial EA planning phase a big bang approach was proposed. It was expected to deliver nominally 80% of the process and data architecture within two years based on ERP application reference models, which need to be tailored to the organisations requirements and extended for non-ERP based processes and data. Assessing this objective again it had to be realised that the pure design of an enterprise model leads to tremendous update and maintenance costs, due to the fact that an Enterprise Architecture is basically always outdated. For this reason the organisation implemented a gradual design approach. This means, that every project which can be associated with or has an impact on the Enterprise Architecture has to model the respective part of CIMod. In dependence on the project objective, such as business process improvement, workflow management or system design, the project has to deliver the relevant enterprise models. The model owner is responsible for the continuous update. Thus, incrementally an Enterprise Architecture is established, which is up to date and can therefore be used as a planning and controlling tool. Issues arisen in this respect are the challenge to overcome inconsistencies between many partial enterprise models, which were designed based on different purposes. The following table summarises which architecture domains were addressed with the background of different project purposes:

Project objective	Business Architecture		Information	Application	Technology
	Process Architecture	Organisation Architecture	Architecture	Architecture	Architecture
<b>Business Process Improvement</b>	x		x		
<b>e-business integration</b>	x				x

Table 1: Project objectives drive architecture domains

It can be concluded that specific project objectives are the driver for the architectural design. Thus, a relationship can be established between the incremental design efforts and the architecture domain to be addressed.

This incremental approach resolves also the decision between modelling from bottom-up or top-down. The continuously partial population of the Enterprise Architecture framework is not a question of bottom up or to down, but in linking the project objective with the respective architecture domain(s) and level(s). Nevertheless, in this case study it was found, that the high contextual and conceptual levels of the Enterprise Architecture were already completely populated in a first enterprise-wide design effort. For example, a system overview model and business process architecture model (Figure 1) depict in an integrated way as-is and to-be scenario. This high modelling level provides strategic direction (to-be) and sets the organisational principles (as-is), while at the same time being the planning and decision support instruments for the corresponding managers. One can say that Enterprise Architecture is here intended to be used in a strong diagnostic sense.

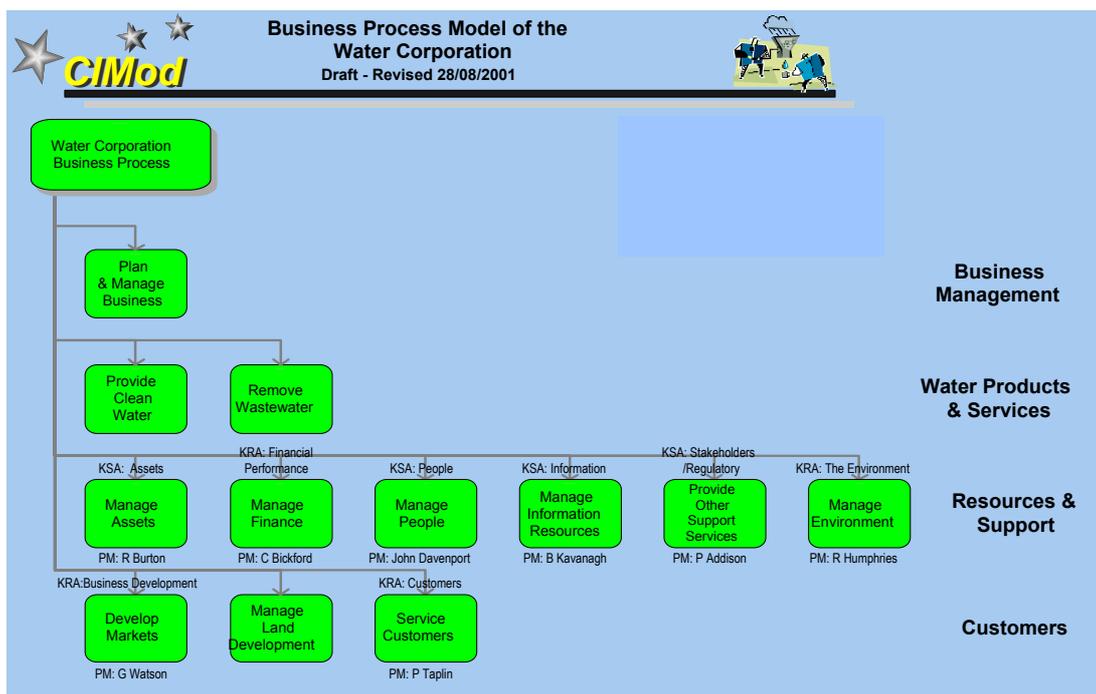


Figure 1: Business process architecture model (Function Tree, ARIS Design Suite)

Such a business process architecture model as depicted in the above figure as function tree also supports on the other hand the assignment of each IS project to its influenced business processes. Respectively a system overview model is depicted in an application system type diagram. Those high contextual and conceptual levels of Enterprise Architectures can serve as starting point for all following incremental 'sub' design efforts.

5. Enterprise Architecture requires a portal for enterprise-wide communication and a high level of acceptance and commitment. Such a portal can only be realised based on a strict assertion of an enterprise-wide modelling tool, modelling notations and disseminated modelling standards.

The capabilities of a modelling tool are critical success factors for the implementation of an Enterprise Architecture. The enterprise modelling tool needs to be able to supply modelling techniques for all domains on all architecture levels. Therefore a notational standard needs to be developed providing direction for populating the Enterprise Architecture. An enterprise-wide modelling standard and procedure model have to be put in place for the Enterprise Architecture design. The modelling tool widely

used by the utility service provider is the design suite of ARIS (Architecture for Integrated Information Systems) with an underlying repository. All architecture domains of processes, organisation, information and application systems are depicted within the corporate Enterprise Architecture repository. The technology architecture is currently designed in Visio. The main advantage the CIMod designer outlined is the integrated nature of modelling techniques within this ARIS tool. Thus, an enterprise wide modelling tool seems to be advantageous due to the common language used throughout the organisation. The Enterprise Architecture standards and procedures for the CIMod design address the governance of CIMod, the relationships within the CIMod framework, as well as procedures for project progression, change control and minimum model requirements including all notational standards.

*6. An Enterprise Architecture framework requires an integrated perspective on Business, Application, Information and Technology Architecture. Enterprise Architecture frameworks have for example to cater for e-business developments in applications.*

The integration of customers and supply chain partners in one organisation's business processes set new challenges for application systems. The utility service provider focuses for example on customer self services based on e-billing and call services. If the customer is enabled to draw and change specific data out of databases the organisation is facing enormous security issues. This effects Application and Technology Architecture, both strongly dominated by the defined process architecture. Conclusively, the Application and Technology Architecture must be strongly linked with the Business Architecture. E-business developments are one example for the increasing complexity and therefore require mounting enterprise integration. Specific modelling notations need to supply those design requirements. Furthermore, the framework should be free of redundancies and needs therefore to have an integrated architectural view, while at the same time splitting complexity into different domains.

Summarising the above three design matters of Enterprise Architecture one can conclude that the design takes place bottom up and top down; incrementally; driven by projects associated with the architecture domains and levels. High contextual and conceptual levels of the Enterprise Architectures should be established to provide strategic direction. A portal for enterprise-wide communication of the EA based on a strict assertion of an enterprise-wide modelling tool, modelling notations and disseminated modelling standards is perceived to ensure a high level of acceptance and commitment across the organisation.

## **6 Summary and Outlook**

This paper presents the results of an explorative case study undertaken at a utility service provider. The objective of this research was to study perceived issues in the design, use and maintenance of Enterprise Architectures.

Enterprise Architectures are often motivated by the desire to manage complexity and guide enterprise integration efforts. A key finding of this case study was that an Enterprise Architecture requires a strategic architecture sponsor, who sets the framework, rules and standards. The framework needs to guarantee an integrated perspective on Business, Application, Information and Technology Architecture. This is of particular interest for development projects in e-business. The Enterprise Architecture design efforts should be incrementally taken; project by project linked to the respective part of the Enterprise Architecture. Furthermore, Enterprise Architectures require an enterprise-wide acceptance and commitment, tight in Enterprise Architecture standards and procedures and made accessible in an Enterprise Architecture Portal.

Potentials for future research are seen in a further exploration of the interrelationships of Enterprise Architecture project objectives and their concerning Architecture domains and levels. This can in the long term build the basis for a procedure model. Further emphasis can also be laid on the integration of architecture performance assessment and the linkage to performance indicators. Another area of significant interest can be seen in the development of widely accepted Enterprise Architecture Lifecycle and Maturity Models.

## 7 References

- Armour, F., Kaisler, S., Liu, S.: A big-picture look at Enterprise Architectures. IEEE, Jan/Feb 1999.
- Armour, F., Kaisler, S., Liu, S.: Building an Enterprise Architecture step by step. IEEE, July/Aug 1999.
- Bernus, P., Nemes, L., Williams, T.J.: Architectures for Enterprise Integration. London, Chapman & Hall 1996.
- Bernus, P., K. Mertins, et al.. Handbook on Architectures of Information Systems. Berlin, New York, Springer, 1998.
- Boster, M., Liu, S., Thomas, R.: Getting the most from your Enterprise Architecture. IEEE, July/Aug 2000.
- Davies, I., Green, P., Rosemann, M.: Modelling in the Australian Practice – Preliminary Insights, Information Age. Journal of the Australian Computer Society. February 2002
- Harmon, P.: Business Process Change, San Francisco, Morgan Kaufmann 2003.
- O'Rourke, C., Fishman, N., Selkow, W.: Enterprise Architecture: using the Zachman Framework., Boston, Thomson, 2003.
- Petrie, C. J.: Enterprise Integration Modelling - Proceedings of the First International Conference, Cambridge, MIT Press 1992.
- Sowa, J. F., Zachman, J. A.: Extending and formalizing the framework for information systems architecture. IBM Systems Journal 31 (3), 1992, pp. 590-616.
- Stevenson, D. A.: Business and Enterprise Architecture, University of Cape Town, Cape Town, <http://users.iafrica.com/o/om/omisditd/denniss/text/busthem0.html> 1995.
- Vernadat, F.B.: Enterprise Modelling and Integration, Principles and Applications, London, Chapman & Hall 1996.
- Zachman, J. A.: A framework for information systems architecture. IBM Systems Journal 26(3): 276-292, 1987.

# Positionierung eines Unternehmensarchitektur-Ansatzes: Erfahrung der Schweizerischen Mobiliar im Architekturmanagement

**Andreas Dietzsch**  
**Schweizerische Mobiliar Versicherungsgesellschaft**  
**Andreas.Dietzsch@mobi.ch**

***Zusammenfassung:** Die langjährigen Erfahrungen der Mobiliar mit Architekturansätzen zeigen, dass für eine erfolgreiche Architekturumsetzung sowohl die Festlegung der internen Organisation als auch der Schnittstelle zu deren Kunden erforderlich ist. Der vorliegende Beitrag hat die für diese Positionierung genutzten Rahmen und die Positionierung der Unternehmensarchitektur der Mobiliar vorgestellt.*

*Dabei wurde deutlich, dass es für diese Schritte neben der Festlegung der Beziehung der Architektur zu ihren Kunden sowie ihrer inneren Struktur auch einer pragmatischen Relativierung der dabei getroffenen Vorgaben bedarf.*

*Mit den skizzierten Regeln wurde aufgezeigt, mit welchen Massnahmen beim Unternehmensarchitektur-Ansatz der Mobiliar versucht wird, die Bedürfnisse einzelner Projekte mit dem Interesse einer langfristigen und systematischen Entwicklung und Umsetzung der Unternehmensarchitektur auszubalancieren.*

## 1 Hintergrund

Die Schweizerische Mobiliar Versicherungsgesellschaft (Mobiliar) bietet Dienstleistungen für Privatpersonen, Unternehmen und den öffentlichen Sektor an. Gegründet 1826 ist sie die erste private Versicherung der Schweiz.

Als in der Mitte der 1990er Jahre der schweizer Versicherungsmarkt dereguliert wurde, bestand die Applikationslandschaft der Mobiliar aus einer breiten Palette, hauptsächlich eigenentwickelter Applikationen. Die Komplexität dieser gewachsenen Applikationslandschaft ist ein wesentlicher Treiber der IT Kosten. Um den zunehmenden Kostendruck auf die Versicherungen und die daraus resultierenden neuen Anforderungen an die IT der Mobiliar gerecht zu werden, wurde ein Instrument benötigt, das eine systematische Transformation der Applikationslandschaft ermöglicht. Als Lösung dieses Problems wurde ein architekturzentrierter Ansatz als Kern der IT-Strategie gewählt.

In diesem Zusammenhang wurde Ende des Jahres 2001 die Organisationseinheit „Business Engineering & Architektur“ mit dem Ziel der Umsetzung einer Unternehmensarchitektur implementiert.

Im Folgenden wird der Rahmen vorgestellt, der genutzt wurde, um den Unternehmensarchitektur-Ansatz der Mobiliar und insbesondere das Architekturmanagement zu positionieren. Dabei wird gezeigt, wie die Beweglichkeit der Architektur und damit deren Wirksamkeit sichergestellt wird.

## 2 Erfahrungen der Mobiliar mit Architekturansätzen

### 2.1 Historie

Das Thema "Architektur" wurde in der Mobiliar erstmals in der Mitte der 1960er Jahre im Zusammenhang mit Hardware-orientierten Assembler-Architekturen diskutiert,

Die Realisierung modellbasierter Architekturprozesse beginnt 1992 mit der Beschaffung und dem Einsatz von IBMs Insurance Application Architecture (IAA), einem Referenzmodell für Daten und Funktionen im Bereich des Versicherungswesens. Die Architekturleistungen beschränken sich zu dieser Zeit primär auf die Datenarchitektur und die einheitliche Umsetzung aller Geschäftsanforderungen. Das Verwalten von Artefakten, die Vorgabe von Richtlinien und Durchsetzung von Modellierungsrichtlinien ist zu dieser Zeit eine zentrale Architekturaufgabe.

Der Entschluss 1996 ein Standard Bestandesführungssystem einzuführen und die Fokussierung des Managements auf einen dezentralen Architekturansatz führten Ende der 90er Jahre dazu, dass die Koordination der Entwicklung und Umsetzung der Architektur durch die Projekte selbst realisiert wurde. Es wurde eine Architektur implementiert hauptsächlich reaktiven Charakter besass, kaum in Projekten verankert war und sich primär auf die Bereitstellung eines konsistenten Technologieportfolios beschränkte.

Im Jahr 1999 wurden durch die IT mögliche Architekturen für die Applikationsintegration untersucht. Ebenfalls in diesem Jahr wurde auf Seite der Fachbereiche ein Projekt mit dem Ziel begonnen, die Grundlagen für eine prozessorientierte Neugestaltung der Unternehmensstruktur zu schaffen. Um dabei die systematische Entwicklung der Geschäftsprozesse des Unternehmens abzusichern, wurde die Business Architektur aufgebaut.

Zum Ende des Jahres 2001 erfolgte die Zusammenlegung der Bereiche Geschäftsentwicklung und Informatik. Damit wurde der Betrachtungsbereich der Architektur auf einen unternehmensweiten Ansatz ausgedehnt. Alle architekturbezogenen Aktivitäten sowie die Verantwortung für die Entwicklung und Umsetzung der Unternehmensarchitektur wurden in einer Organisationseinheit zusammengefasst.

Die beschriebenen Architekturansätze und deren zeitliche Einordnung fasst Abb. 1 zusammen.

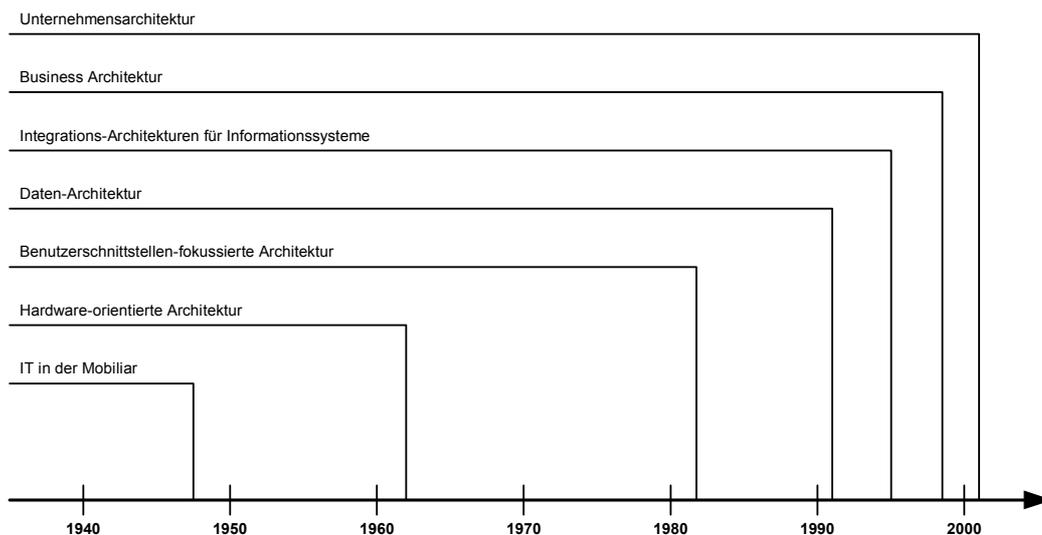


Abb. 1: Zeitliche Einordnung der in der Mobiliar verfolgten Architekturansätze

## 2.2 Erfahrungen

Bei der Verfolgung der beschriebenen Architekturansätze konnten Erfahrungen zu verschiedenen Aspekten architekturzentrierter Veränderungen gewonnen werden.

Eine wichtige Erkenntnis ist, dass der Nutzen eines Architekturansatzes stets durch dessen Fokus begrenzt wird. Dabei steigt der Nutzen einer Architektur als Rahmen für systematische Veränderungen mit dem Anwachsen des Umfangs des durch die Architektur adressierten Bereichs.

Als problematisch erwies sich die Umsetzung eines passiven Architekturkonzepts, dessen wesentliche Aufgaben das Verwalten von Artefakten, die Vorgabe von Richtlinien und deren administrative Durchsetzung sind. Durch eine geringe Verankerung in Projekten waren bei diesem Ansatz Architekturvorgaben nur wenig akzeptiert. Daraus resultierte ebenfalls eine Wahrnehmung der Architektur als Ansatz, der nicht an den täglichen Problemen der Kunden ausgerichtet ist.

Daneben wird der Erfolg der Umsetzung eines Architekturansatzes durch die aufbauorganisatorische Zuordnung der Architekten bestimmt. So war festzustellen, dass auf der einen Seite die Umsetzung einer Architektur nur möglich ist, wenn diese dezentral organisiert wird. Auf der anderen Seite zeigte es sich jedoch, dass die konzeptionelle Entwicklung einer Architektur nur dann realisiert werden kann, wenn die Verantwortung dafür zentral zusammengefasst wird.

Auf Basis der mit den verschiedenen Architekturansätzen gewonnenen Erfahrungen wurde der Unternehmensarchitektur-Ansatz der Mobiliar positioniert. Wie dies erfolgte beschreiben die nachfolgenden Abschnitte.

## 3 Positionierung eines Architekturansatzes

Die Positionierung eines Architekturansatzes bestimmt die Form der Entwicklung und Umsetzung einer Architektur im Unternehmen. Dabei ist zunächst die Aussensicht des Architekturansatzes, d. h. dessen Bezug zu den Kunden der Architekturleistungen, zu definieren. Anschliessend wird mit der Beschreibung der Innensicht des Architekturansatzes, dessen interne Organisation festgelegt. Nachfolgend wird dargestellt wie der Architekturansatz dieser Gliederung entsprechend in der Mobiliar positioniert wurde.

### 3.1 Aussensicht

Die Aussensicht eines Architekturansatzes setzt diesen in Bezug zu den Kunden der Architekturleistungen. Nach DIKEL ET AL., die zu diesem Zweck die Metapher „Rhythmus einer Architektur“ nutzen, definiert die Aussensicht „... the recurring, predictable exchange of work products within an architecture group and across their customers and suppliers.“ ([3], S. 74) Der „Rhythmus einer Architektur“ wird durch die Elemente Inhalt, Tempo und Qualität bestimmt.

<b>Rhythmus</b>	<b>Inhalt</b>	Der Inhalt beschreibt, <b>was</b> , d. h. welche Artefakte, innerhalb der Architektur sowie mit Kunden ausgetauscht werden.
	<b>Tempo</b>	Das Tempo bestimmt, <b>wann</b> die im Inhalt festgelegten Artefakte bereitgestellt werden. Es definiert somit Zeitpunkt oder Frequenz der Bereitstellung von Architekturleistungen.
	<b>Qualität</b>	Die Qualität definiert, <b>wie</b> Architekturergebnisse an die Kunden weitergegeben werden. Dabei wird festgelegt wie sichergestellt wird, dass die Architektur frei von Unzulänglichkeiten ist oder in welchen Bearbeitungsstadien Ergebnisse der Architektur weitergegeben werden.

**Tab. 1: "Rhythmus-Struktur" der Aussensicht eines Architekturansatzes**

Um einen Architekturansatz aus Kundensicht zu definieren ist deshalb anzugeben, *was* an Ergebnissen *wann* und *wie* den Kunden der Architektur bereitgestellt wird. Die Beziehung der Unternehmensarchitektur in der Mobiliar zu ihren Kunden wird nachfolgend beschrieben.

### 3.1.1 Inhalt

Die Unternehmensarchitektur der Mobiliar stellt Ergebnisse sowohl für die Fachprozesse „Produktentwicklung“, „Versichern“ und „Schaden“ als auch für die IT-Entwicklung und den IT-Betrieb bereit. Dabei liegt die Verantwortung für die Erstellung, Umsetzung und Weiterentwicklung der Ergebnisse bei diesen Bereichen. Die Architektur unterstützt die jeweiligen Verantwortlichen.

Bestandteile der Business-Architektur sind Ziele, Dienstleistungen, Geschäftsprozesse, Geschäftsvorfälle und Geschäftsobjekte.

Systeme, die die Geschäftsprozesse informationstechnisch unterstützen sind Gegenstand der Applikations-Architektur. Die Architektur auf dieser Ebene beschreibt Beziehungen zwischen den Bestandteilen der Applikationslandschaft, z. B. unter Verwendung von Komponenten, Konnektoren und Protokolle.

Die Technische Architektur definiert, wie die Komponenten der Applikationslandschaft durch Module und Laufzeiteinheiten realisiert werden. Hauptergebnisse dieser Teilarchitektur sind Konzepte zu Plattformen, Programmierumgebungen, Datenbanken aber z. B. auch die Konzeption und Umsetzung einer Plattform zur Applikationsintegration.

### 3.1.2 Tempo

Die Unternehmensarchitektur wirkt im Wesentlichen im Rahmen von Projekten. Aus diesem Grund ist der Zeitpunkt der Bereitstellung der meisten Architekturergebnisse nicht fest definiert. Er wird vielmehr durch den jeweiligen Projektplan bestimmt. Die Umsetzung der Unternehmensarchitektur wird dabei als Bestandteil der Projektarbeiten behandelt. Eine Ausnahme sind die Beurteilungen von Projektanträgen, die in die Unternehmensplanung aufgenommen werden sollen. Da die Unternehmensplanung einem festen jährlichen Rhythmus folgt, gelten die in diesem Rahmen festgelegten Phasen und Termine für die Bereitstellung der Architekturbeurteilungen.

### 3.1.3 Qualität

Für in Projekten erbrachte Leistungen erfolgt die Bestimmung der für ein Architekturergebnis erforderlichen Qualität ebenfalls im Rahmen der Projektplanung. Hierbei wird auch festgelegt durch wen die Abnahme erfolgt. Für die Artefakte der Business Architektur sind dies im Normalfall die Prozessver-

antwortlichen der Geschäftsprozesse. Ergebnisse der Applikations- und Technischen Architektur werden dementsprechend durch die Verantwortlichen der IT-Entwicklung bzw. des IT-Betriebs bewertet. Für Ergebnisse der konzeptionellen Architekturentwicklung erfolgt zunächst eine ausschliesslich interne Kontrolle. Deren Ergebnisse werden dann durch die Verwendung des Ergebnisses in einem Projekt evaluiert.

### 3.2 Innensicht

Die Innensicht eines Architekturansatzes wird durch vier Facetten charakterisiert - Fokus, Philosophie, Prozesse und Aufbauorganisation der Architektur.

#### 3.2.1 Fokus

Der Fokus eines Architekturansatzes bestimmt die Themenfelder, die durch die Architektur adressiert werden. Er gibt an, welche Bereiche des Unternehmens auf der Grundlage von Architekturen entwickelt werden. Dies sind z. B. das IT-Technologieportfolio, die Anwendungsintegration, Datenstrukturen und Prozesse.

Wie bereits dargestellt, ist die Unternehmensarchitektur der Mobiliar in die Ebenen Business-, Applikations- und technische Architektur gegliedert. Damit umfasst sie sowohl die Gestaltung der Geschäftsprozesse als auch deren Abbildung auf die unterstützenden Informationssysteme und die Technologien, auf deren Basis diese entwickelt und betrieben werden. Die folgende Tabelle stellt die Ebenen der Unternehmensarchitektur mit ihren jeweiligen Wirkungsbereich und den wesentlichen Artefakten zusammenfassend dar.

Architekturebene	Hauptelemente	
<b>Business Architektur</b>	Wirkungsbereich	Grundsätze, Unternehmensstrategie
	Artefakte	Geschäftsprozesse und -vorfälle, Fachkomponenten
<b>Applikations-Architektur</b>	Wirkungsbereich	Beziehungen zwischen den Geschäftsprozessen und den unterstützenden Applikationen.
	Artefakte	Applikationskomponenten, Applikationen, Schnittstellen zwischen Applikationen
<b>Technische Architektur</b>	Wirkungsbereich	Beziehungen zwischen der Applikationslandschaft und der technischen Infrastruktur
	Artefakte	Konzepte zu Plattformen, Netzwerken, Datenbank-Systemen, Entwicklungsumgebungen, Integrationsplattformen

**Tab. 2: Charakterisierung der Ebenen der Mobiliar-Unternehmensarchitektur**

Wegen der besonderen Bedeutung einer konsistenten Datenbasis, existiert neben diesen Teilarchitekturen die Datenarchitektur, die übergreifend über die genannten Ebenen definiert ist.

#### 3.2.2 Philosophie

In welcher Art und Weise ein Architekturansatz umgesetzt wird, beschreibt die Philosophie, wobei eine Umsetzung aktiv-kooperativ oder reaktiv-administrativ erfolgen kann [4]. Schwerpunkt einer reaktiv-administrativen Architekturumsetzung ist die Definition einzuhaltender Vorgaben, wie z. B. Richtlinien

und Standards. Ergebnisse werden zunächst im Rahmen eines Projekts erstellt und anschliessend hinsichtlich ihrer Konformität zu den getroffenen Vorgaben beurteilt.

Die aktiv-kooperative Umsetzung ist die Grundphilosophie des Unternehmensarchitektur-Ansatzes der Mobiliar. Dessen Basis ist die vorausschauende Beteiligung der Architektur bei der Lösungsfindung und -umsetzung. Um ein aktiv-kooperatives Wirken der Architektur zu gewährleisten, verfügt die Organisationseinheit „Architektur“ nicht über ein Budget für die Realisierung von Projekten. Die Umsetzung der Zielarchitektur kann aus diesem Grund nur durch Kooperation der Architektur mit anderen Projekten erfolgen.

### **3.2.3 Prozesse**

Mit der Definition der Prozesse der Architektur ist die Frage zu beantworten, welche Abläufe für das erfolgreiche Entwickeln und Betreiben der Architektur erforderlich sind und wie diese mit den Unternehmensprozessen verbunden werden.

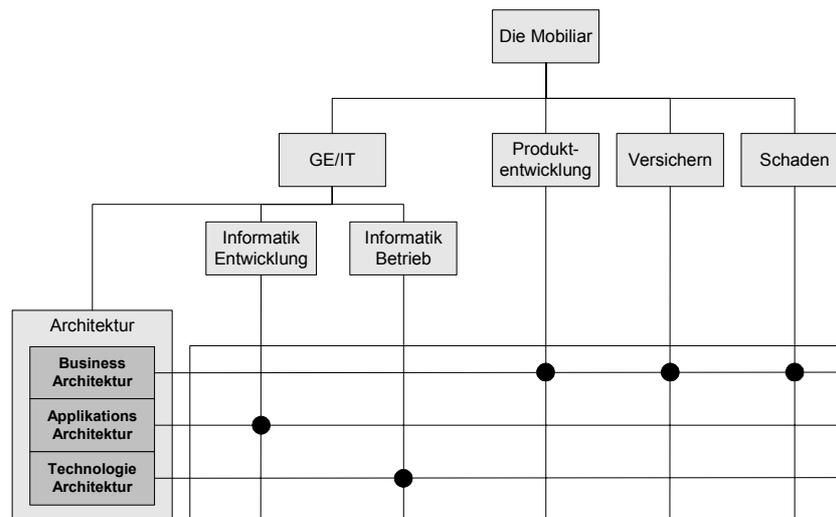
Grundsätzlich ist es hierbei zum einen möglich, die Architekturprozesse in einem Unternehmensprozess zusammenzufassen oder spezifische Architekturprozesse als Teilprozesse z. B. von Prozessen der Fachseite oder IT zu betreiben.

Die Unternehmensarchitektur der Mobiliar ist als eigenständiger Prozess organisiert, der neben dem Architekturmanagement aus den Teilprozessen „Architektur entwickeln“, „Architektur vertreten“ und „Architektur beurteilen“ besteht.

Die Bereitstellung der Architekturergebnisse sowie der Prozesse und Hilfsmittel zu deren Anwendung ist Gegenstand des Prozesses „Architektur entwickeln“. Als Ausgangspunkt dieses Prozesses werden Ergebnisse des Prozesses „Architektur vertreten“, d. h. architekturrelevante Ergebnisse aus Projekten, genutzt. Durch das aktive Vertreten der Architektur werden Projekte, mit einer hohen Bedeutung für die Weiterentwicklung der Architektur, bei der Lösungsfindung und -umsetzung unterstützt. Die architekturkonforme Projektrealisierung wird dabei durch Betreiben eines Architekturbüros, Befähigung von Mitarbeitern oder die aktive Mitarbeit von Architekten bei der Entwicklung von Lösungen sichergestellt. Der Prozess „Architektur beurteilen“ umfasst alle Ebenen der Unternehmensarchitektur und sichert die Architekturkonformität von Vorhaben ab. Die Hauptaufgabe ist die Bewertung von Vorhaben, die in die Unternehmensplanung aufgenommen werden sollen oder von Ergebnissen solcher Projekte, bei denen eine aktive Beteiligung von Architekten nicht möglich oder sinnvoll war.

### **3.2.4 Aufbauorganisation**

Mit der Definition der Aufbauorganisation wird die Rolle der Architekten innerhalb des verfolgten Architekturansatzes sowie deren Verankerung in der Organisationsstruktur bestimmt. Dabei ist zu entscheiden, ob „der Architekt“ als Rolle oder als Funktion definiert wird, d. h. ob die Architektur in einer zentralen Organisationseinheit des Unternehmens zusammengefasst oder in Form einer Matrixorganisation im Unternehmen verteilt wird.



**Abb. 2: Aufbauorganisatorische Einordnung der Unternehmensarchitektur der Mobiliar**

In der Mobiliar wurde eine aufbauorganisatorische Mischform gewählt. Die Organisationseinheit „Architektur“ wurde implementiert. Deren wesentliche Aufgaben bestehen in der Architekturentwicklung und der Koordination und Unterstützung der dezentralen Umsetzung. Die Aufgabe des Architekturmanagements ist ebenfalls dieser Organisationseinheit zugeordnet. Daneben werden für die Erarbeitung spezifischer Architekturergebnisse Mitarbeiter aus anderen Fachbereichen, der IT-Entwicklung und dem IT-Betrieb in Form einer Matrix-Organisation integriert.

### 3.3 Erfahrungen mit dem Unternehmensarchitektur-Ansatz der Mobiliar

Die Unternehmensarchitektur der Mobiliar wird in der beschriebenen Form seit ca. 1½ Jahren betrieben. In dieser Zeit hat es sich gezeigt, dass auf den verschiedenen Ebenen der Architektur unterschiedliche Treiber wirken, die asynchrone Lebenszyklen der Architekturergebnisse auslösen. Dies und die hohe Anzahl der Architekturergebnisse verursachen die folgenden Probleme:

- Konflikte zwischen verschiedenen Interessengruppen werden durch die Architektur explizit.
- Die Unternehmensarchitektur ist stets unvollkommen.
- Der Nutzen der Architektur ist schwer nachzuweisen.

Die mit der Unternehmensarchitektur zusammengefassten Architekturergebnisse adressieren Inhalte in der Verantwortung verschiedener Organisationseinheiten. Konflikte zwischen deren Interessen werden deshalb häufig im Prozess der Definition der Architektur deutlich.

So ist das wesentliche Interesse der Fachprozesse „Versichern“ und „Schaden“ die abgesicherte Verfügbarkeit der im Prozess genutzten Applikationen sowie deren Anpassung an neue fachliche Anforderungen. Eine wesentliche Anforderung des Produktentwicklungs-Prozesses ist dagegen die mit den Produktlebenszyklen abgestimmte Entwicklung der Applikationslandschaft. Auch die Ziele der IT-Prozesse sind nicht vollständig kongruent. Während das Interesse der IT-Entwicklung in der Berücksichtigung aktueller Technologien bei der Applikationsentwicklung liegt, ist die Stabilität und langfristige Entwicklung der Applikationslandschaft ein wesentliches Ziel des IT-Betriebs. Dabei wird die Reduktion der Komplexität der Applikationslandschaft angestrebt.

Dieses Konfliktpotenzial berücksichtigend muss die Architektur nicht als das Problem, sondern als Weg zu dessen Lösung kommuniziert werden.

Aufgrund der Vielzahl enthaltener Architekturelemente und deren asynchroner Lebenszyklen ist die Unternehmensarchitektur einer kontinuierlichen Verschiebung ausgesetzt, die zu Inkonsistenzen innerhalb der Architektur führt (vgl. [5], [7]). Da zu jeder Zeit auf jeder Ebene der Architektur Veränderungen stattfinden, sind diese Inkonsistenzen als inhärentes Merkmal der Unternehmensarchitektur anzusehen. Die Aufgabe des Architekturmanagements ist es deshalb nicht diese Inkonsistenzen zu vermeiden sondern diese zu handhaben. Der dafür erforderliche Aufwand ist mit den Anforderungen an die Qualität der Architektur auszubalancieren.

Die Architekturentwicklung unterliegt Restriktionen, die sich aus der Begrenzung der verfügbaren Ressourcen sowie betriebswirtschaftlichen Überlegungen ergeben. Mit Bezug zu diesen Einschränkungen und mit Rücksicht auf den jeweils projektbezogen zu realisierenden Nutzen ist die vollständige Definition einer Unternehmensarchitektur kein Ziel.<sup>3</sup> Mit der Unternehmensarchitektur sind vielmehr Konzepte bereitzustellen, die das Handhaben von Unvollständigkeit bei der Umsetzung der Architektur erlauben.

In der Mobiliar hat sich im Verlauf mehrerer Projekte gezeigt, dass die Erarbeitung von Architekturergbnissen trotz der beschriebenen Einschränkungen umfangreiche zeitliche und personelle Ressourcen bindet. Die langfristige Akzeptanz und Unterstützung eines architekturzentrierten Ansatzes hängt deshalb wesentlich davon ab, dessen Nutzen klar darstellen zu können. Eine Hürde hierbei ist, dass der Fokus eines Projekts gegenüber dem der Unternehmensarchitektur sowohl zeitlich als auch inhaltlich eingeschränkt ist. Dies hat zur Folge, dass in Projekten stets nur einzelne, spezifische Teile einer Unternehmensarchitektur genutzt werden. Um dennoch einen Gesamtnutzen aufzeigen zu können ist es erforderlich, nutzenstiftende Ergebnisse für jedes Einzelprojekt zu realisieren.

Diese Erfahrungen mit dem Mobiliar-Ansatz zur Unternehmensarchitektur zeigen deutlich, dass die ausschliessliche Definition von Aussensicht und Innensicht eines Architekturansatzes unzureichend ist. Vielmehr sind diese beiden Schritte durch eine pragmatische Relativierung zu ergänzen.

Dies bedeutet, dass Regeln zu bestimmen sind, nach denen die Architektur dann umgesetzt werden soll, wenn keine spezifischen Vorgaben existieren oder die Einhaltung dieser Vorgaben im Konflikt zu anderen Unternehmenszielen steht.

## 4 Konsequenzen für das Architekturmanagement

Aus den dargestellten Erfahrungen wird deutlich, dass es für den Erfolg eines Unternehmensarchitektur-Ansatzes entscheidend ist, die Architekturprozesse in die Lage zu versetzen, Inkonsistenzen systematisch zu handhaben und projektspezifische Lösungen zeitnah und an der Projektplanung ausgerichtet bereitzustellen.

Zu diesem Zweck wurden für die Unternehmensarchitektur der Mobiliar neun Regeln formuliert, mit denen die Flexibilität der Architektur, insbesondere gegenüber ihren Kunden, sichergestellt werden soll. Die nachfolgende Tabelle fasst diese Regeln zusammen.

---

<sup>3</sup> Es würde sich vielmehr um ein Beispiel für das Antimuster "Big Design Up Front" handeln (vgl. [6]).

<b>Rhythmus</b>	<b>Inhalt</b>	<ul style="list-style-type: none"> <li>• Zweckgebundene Architekturen</li> <li>• Lokale Veränderungen</li> <li>• Einfachheit voraussetzen</li> </ul>
	<b>Tempo</b>	<ul style="list-style-type: none"> <li>• Schnelle Rückmeldung</li> <li>• Zeitscheiben bilden</li> <li>• Nächste Schritte vorbereiten</li> </ul>
	<b>Qualität</b>	<ul style="list-style-type: none"> <li>• Erwarte Instabilität</li> <li>• Schrittweise Veränderungen</li> <li>• Multiple Architekturen</li> </ul>

**Tab. 3: Regeln zur pragmatischen Relativierung des Architekturansatzes der Mobiliar**

#### 4.1 Inhaltliche Relativierung

Mit der inhaltlichen Relativierung wird festgelegt, wann es zulässig ist, dass ein Architekturergebnis von den zu beschreibenden Inhalten abweicht. Dabei ist die Akzeptanz der Unternehmensarchitektur durch deren Kunden als ein kritischen Erfolgsfaktor zu berücksichtigen.

Aus diesem Grund werden in der Mobiliar Architekturergebnisse ausschliesslich mit Bezug zu Projekten entwickelt. Damit wird abgesichert, dass alle Bestandteile der Architektur zweckgebunden und mit den jeweiligen Abnehmern abgestimmt sind.

Um trotz inhärenter Inkonsistenzen die Weiterentwicklung der Unternehmensarchitektur zu ermöglichen, werden im Rahmen von Projekten nur lokale Veränderungen realisiert. Dies bedeutet, dass Veränderungen stets nur an abgegrenzten Teilen der bestehenden Architektur vorgenommen werden. Den beteiligten Architekten fällt dabei die Aufgabe zu, Veränderungen in ihren Auswirkungen zu dokumentieren und zu kontrollieren. Führt ein Projektergebnis zu Inkonsistenzen, wird die Architekturplanung so angepasst, dass diese bei der Architekturumsetzung eliminiert werden.

Um die Komplexität der Unternehmensarchitektur beherrschen zu können, ist es Architekturprinzip, stets die einfachste Lösung zu einem Problem zu realisieren. Aus der Anwendung dieser Regel resultiert das Ziel Redundanzen, z. B. bei Strukturen von Geschäftsprozessen oder Applikationen, zu reduzieren bzw. zu vermeiden. Ein weiteres resultierendes Ziel ist die "Anspruchsreduktion". Dabei wird die Bedeutung einer Anforderung und damit ihrer Umsetzung daran gemessen, wie diese auf die Fähigkeit des Unternehmens wirkt, das Tagesgeschäft aufrechtzuerhalten sowie zur Umsetzung der angestrebten Architektur beizutragen.

Die folgende Tabelle fasst die Massnahmen zusammen, die eine inhaltliche Relativierung von Architekturvorgaben sicherstellen sollen.

Herausforderung	Massnahmen
<ul style="list-style-type: none"> <li>• Eine problemadäquate Granularität ist die Basis für den Nutzen eines Architekturelements.</li> <li>• Die Akzeptanz durch die Anwender ist ein kritischer Erfolgsfaktor für eine Architektur.</li> <li>• Jedes Projekt beeinflusst die Unternehmensarchitektur mit seinen Ergebnissen.</li> <li>• Inkonsistenzen sind ein inhärentes Merkmal einer Unternehmensarchitektur.</li> <li>• Die gesamte Architektur muss zwischen Qualität und Wirtschaftlichkeit balanciert werden.</li> </ul>	<ul style="list-style-type: none"> <li>• Keine Architektur wird ausserhalb eines Projekts entwickelt.</li> <li>• Keine Architekturentwicklung erfolgt ohne Einbezug der Stakeholder.</li> <li>• Bei der Definition einer Architektur ist die einfachste Lösung eines Problems die beste.</li> <li>• Die Architektur wird niemals als Ganzes verändert.</li> <li>• Redundanzen sind zu vermeiden.</li> <li>• Bei der Architekturentwicklung werden die Anforderungen berücksichtigt, die der Aufrechterhaltung oder strategischen Entwicklung des Geschäfts dienen.</li> </ul>

## 4.2 Zeitliche Relativierung

Die Erfahrungen der Mobiliar zeigen, dass ein weiterer kritischer Erfolgsfaktor die Zeitspanne zwischen dem Formulieren eines Problems, z. B. durch ein Projekt, und der entsprechenden Antwort durch die Architektur ist.

Um eine architekturkonforme Lösungsfindung abzusichern sind kurze Antwortzeiten der Architektur auf Anfragen aus Projekten erforderlich. Da für eine schnelle Rückmeldung eine enge Zusammenarbeit mit dem Kunden erforderlich ist, werden Architekten in der Mobiliar als Mitglieder des Projektteams eingesetzt. Dies ermöglicht es, Kundenanforderungen frühzeitig zu erkennen und Lösungsvarianten proaktiv zu entwickeln.

Die ausschließliche Fokussierung auf Probleme einzelner Projekte führt jedoch zu einer Architektur, die aus projektspezifisch optimierten Einzelementen besteht. Um die Entwicklung der Unternehmensarchitektur als Ganzes zu planen, werden ausgehend von der IST-Architektur die Schritte zur Realisierung der Ziel-Architektur in Form von zeitlich aufeinanderfolgenden SOLL-Architekturen bestimmt.

Die Projektaktivitäten richten sich an der nächsten zu erreichenden Architektur aus. Eine Lösung, die eine solche Ausrichtung nicht aufweist, wird nur dann zugelassen, wenn für deren Realisierung, Betrieb und Entsorgung, ein wirtschaftlicher Nutzen nachweisbar ist.

Um das Problem der Koordination von Inkonsistenzen zu lösen, ohne die Robustheit der Unternehmensarchitektur zu gefährden, werden Zeitscheiben gebildet. Diese korrespondieren mit der Projektplanung und geben an, wann welche Teile einer Soll-Architektur realisiert werden. Das Bestimmen der Folgen einer Inkonsistenz und deren Auswirkungen auf die Soll-Architektur, ist die Grundlage der Bildung von Zeitscheiben. Das Management von Inkonsistenzen wird somit durch die Definition von Zeitplänen für lokale Veränderungen möglich. In der Mobiliar umfassen diese einen Zeitraum von maximal drei Jahren. Damit wird sichergestellt, dass der Erfolg der Realisierung gemessen und wenn notwendig die Soll-Architektur korrigiert werden kann.

Die folgenden Massnahmen dienen somit der zeitlichen Relativierung von Architekturvorgaben.

Herausforderung	Massnahmen
<ul style="list-style-type: none"> <li>• Die Konzentration auf einzelne Projekte führt zu einer lokal optimierten Architektur.</li> <li>• Die Sicherung der Stabilität der Architektur bedeutet die Koordination von Inkonsistenzen.</li> <li>• Je länger zu einem Problem keine Position der Architektur besteht, um so höher ist das Risiko einer nicht architekturkonformen Lösung des Problems.</li> </ul>	<ul style="list-style-type: none"> <li>• Grundsätzlich muss sich jede Projektaktivität an den Anforderungen der Soll-Architektur ausrichten.</li> <li>• Mit einem positiven Business Case sind provisorische Lösungen auch dann zulässig, wenn sie zu Inkonsistenzen führen.</li> <li>• Die maximale Lebensdauer einer Inkonsistenz wird durch Zeitscheiben definiert.</li> <li>• Eine schnelle Rückmeldung wird durch die Mitarbeit der Architekten als Mitglieder der Projektteams sichergestellt.</li> </ul>

### 4.3 Relativierung der Qualität

Die Erfahrung hat gezeigt, dass eine Unternehmensarchitektur einem ständigen Wandel unterworfen ist. Ohne eine Kontrolle der sich daraus ergebenden Veränderungen wird die Zahl von Inkonsistenzen zunehmen, bis die Architektur ihren Zweck nicht mehr erfüllt.

Das Architekturmanagement der Mobiliar berücksichtigt dies und geht von Veränderungen und damit der Instabilität der Architektur aus. Das Ziel ist es, an Stelle der Architektur den Veränderungsprozess der Architektur zu stabilisieren. Da Prinzipien die Elemente einer Architektur mit der höchsten Stabilität sind [3], basiert die Unternehmensarchitektur der Mobiliar auf Architekturprinzipien, die eine Orientierung für die Realisierung der Architekturprozesse bieten [1].

Eine Unternehmensarchitektur muss das vielschichtige System "Unternehmen" und die damit möglichen Sichten berücksichtigen. Um alle aus einer Unternehmensarchitektur resultierenden Anforderungen zu erfüllen, müssen deshalb multiple Architekturen unterstützt werden. Aus diesem Grund ist die Unternehmensarchitektur der Mobiliar, neben den bereits beschriebenen Ebenen, in Aspekte geteilt. Dies sind z. B. Geschäftsprozesse, die Unternehmensdaten, interagierende Applikationen oder Komponenten. Zu diesen Aspekten existieren jeweils angepasste Methoden, Hilfsmittel und Vorgaben der Architektur.

Das Streben nach einer vollständigen Definition einer Architektur vor dem Beginn von Umsetzungsaktivitäten, führt mit hoher Wahrscheinlichkeit zu einem langen Zeitraum ohne Ergebnisse. Umfangreiche Projekte der Mobiliar zeigen, dass es ein solcher Ansatz dem Projektmanagement erschwert, den Projektfortschritt zu kontrollieren und zu kommunizieren. Um dennoch eine „Analyse Paralyse“ [2] zu verhindern, werden die durch ein Projekt benötigten Teile der Unternehmensarchitektur schrittweise entwickelt. Im Kontrast zu einer vollständigen Architekturdefinition impliziert dies ein (kontrolliertes) Fehlen von Spezifikation. In der Mobiliar wird dies realisiert, indem bestimmte projektspezifische Architekturlösungen entwickelt und durch das Team der Architekten, unabhängig von spezifischen Projekten, in die Unternehmensarchitektur integriert werden. Unzureichend oder nicht spezifizierete Teile der Unternehmensarchitektur werden als Inkonsistenzen gehandhabt.

Die Massnahmen zur Relativierung von Architekturvorgaben zur Qualität fasst die folgende Tabelle zusammen.

Herausforderung	Massnahmen
<ul style="list-style-type: none"> <li>• Die Architekturelemente verändern sich im Zeitverlauf mit unterschiedlicher Geschwindigkeit.</li> <li>• Ohne Kontrolle der daraus resultierenden Änderungen steigt die Zahl der Inkonsistenzen bis die Architektur ihre Funktion nicht mehr erfüllt.</li> <li>• Die vollständige Definition führt zu einem langen Zeitraum ohne "konkrete" Ergebnisse.</li> <li>• Die Phasen eines Projekts führen zu unterschiedlichen Anforderungen an Tiefe und Breite der Definition einer Architektur.</li> <li>• Verschiedene Sichten und Ebenen der Architektur erfordern spezifische Methoden und Werkzeuge.</li> </ul>	<ul style="list-style-type: none"> <li>• An Stelle der Architektur selbst werden die Veränderungsprozesse stabilisiert.</li> <li>• Prinzipien sind die Basis der Definition der Unternehmensarchitektur.</li> <li>• Architekturlösungen werden projektspezifisch entwickelt.</li> <li>• Projektergebnisse werden unabhängig von spezifischen Projekten in die Architektur integriert und gepflegt.</li> <li>• Verschiedene, an den Teilarchitekturen ausgerichtete Methodenteile existieren, die miteinander gekoppelt sind.</li> </ul>

## 5 Literaturverzeichnis

- [1] Bredemeyer, D.: Architecting Process. <http://www.bredemeyer.com/pdffiles/ProcessGuide.pdf>, Download: 2001-08-12.
- [2] Brown, W. H.; Malveau, R. C.; McCormick III, H. W.; Mowbray, T. J.: AntiPatterns - Refactoring Software, Architectures and Projects in Crisis. Wiley Computer Publishing: New York, 1998.
- [3] Dikel, D. M., Kane, D., Wilson, J. R.: Software Architecture: Organizational Principles and Patterns. Prentice Hall, Upper Saddle River. 2001.
- [4] Maier, M. W.; Rehtin, E.: The Art Of Systems Architecting. 2. Auflage, CRC Press LLC: Boca Raton, 2002.
- [5] Nuseibeh, B., Easterbrook, S.: The Process of Inconsistency Management: A framework for understanding. In: Proceedings of the DEXA Workshop, 1999. S. 364-368.
- [6] o.V.: Big Design Up Front. <http://xp.c2.com/BigDesignUpFront.html>. Download: 20.07.2002.
- [7] Paulish, D. J.: Architecture-Centric Software Project Management: A Practical Guide. Pearson Education, Boston. 2002.

Positionierung eines Unternehmensarchitektur-Ansatzes: Erfahrung der Schweizerischen Mobiliar im Architekturmanagement