

# Realweltbasiertes Informationsmodell als grundlegender Bestandteil einer serviceorientierten und flexiblen Hochschul-IT

Sebastian Gogolok, Andreas Homrighausen

Zentrum für Medien und IT  
FernUniversität in Hagen  
Universitätsstraße 21, 58084 Hagen  
{sebastian.gogolok, andreas.homrighausen}@fernuni-hagen.de

**Abstract:** Auf dem Weg zur Hochschule 2020 sind hohe Betriebskosten, kaum vorhandene Erweiterbarkeit der Anwendungslandschaft aufgrund eng gekoppelter Anwendungen, schlechte Datenqualität und aufwendige Datenmigrationen Herausforderungen, die jede Hochschule zu lösen hat. Gerade Projekte zur Erneuerung der IT-Landschaft geraten durch diese Rahmenbedingungen in zeitliche und finanzielle Bedrängnis. Als Lösungsmöglichkeit wird in diesem Paper ein anwendungsunabhängiges Informationsmodell als grundlegender Bestandteil einer serviceorientierten und flexiblen Hochschul-IT im Kontext einer Enterprise Architektur vorgestellt. Es wird gezeigt, wie mit Hilfe dieses Modells eine über Jahre gewachsene, unflexible IT-Landschaft in eine qualitativ hochwertige Hochschul-IT überführt werden kann, welche eine hohe Datenqualität besitzt und einfache Datenmigration erlaubt. Hierbei bildet die Modellierung der (Universitäts-)Realwelt die Grundlage des Informationsmodells und ermöglicht die anwendungsunabhängige Definition, Speicherung und Verarbeitung von Informationen über alle Ebenen der Enterprise Architektur hinweg. Der vorgestellte Ansatz wurde im Rahmen eines großen IT-Projekts zur Neuausrichtung und Modernisierung der IT-Landschaft an der FernUniversität in Hagen entwickelt und wird aktuell erfolgreich eingesetzt.

## 1 Einleitung

Auf dem Weg zur Hochschule 2020 gibt es viele Herausforderungen. Eine der größten Herausforderungen ist es, die Generation Facebook [LR11] mit ihren Ansprüchen und Anforderungen an Dienstleistung und Verfügbarkeit von Informationen zufrieden zu stellen und im Hochschulalltag an die Hochschule zu binden. Denn anders als in den Jahrzehnten zuvor, werden die Universitäten aufgrund des demographischen Wandels in Konkurrenz um die knapper werdende Ressource der Studierenden stehen [KMK05].

Um sich innerhalb dieser Konkurrenz hervorzuheben, ist es notwendig, sich einerseits den äußeren Gegebenheiten, wie neuen Hochschulgesetzen, flexibel anzupassen und andererseits auf die Kundenwünsche und –vorstellungen zeitnah eingehen zu können. Gerade der Trend des „Mobile Shift“, der die Verlagerung von Diensten auf mobile Endgeräte beschreibt, und der dadurch resultierende Zugriff von heterogenen Endgeräten auf IT-Ressourcen, bedingt eine flexible und dienstorientierte Ausrichtung der IT-

Architektur einer Organisation. Diese ist im Ist-Zustand jedoch nicht an vielen Universitäten so anzutreffen. Zum einen ist dies dadurch bedingt, dass die zentrale Vorgabe einer einheitlichen Software für das Campus-Management in der Vergangenheit zu ähnlichen, zentralen und an der Linienorganisation ausgerichteten IT-Strukturen an allen Hochschulen beigetragen hat. Zum anderen sind Themen wie Flexibilität, Dienstleistungsorientierung und digitaler Campus in der Hochschulwirklichkeit erst in der jüngeren Vergangenheit zum Thema geworden.

Somit starten fast alle Hochschulen – ausgenommen sind Neugründungen – mit einer gewachsenen IT-Landschaft, die starke Abhängigkeiten der verwendeten Anwendungen (enge Kopplung) untereinander und auch der Anwendungen zu ihren Daten aufweist. Die Fähigkeit der flexiblen und vor allem überschaubaren Anpassung solcher Systemverbünde tendiert gegen Null. Zusammengenommen mit den in weiten Teilen nicht dokumentierten Prozessen der Hochschulen fehlen für eine dienstorientierte Architektur (SOA) somit wichtige Grundlagen.

Um den Herausforderungen der Hochschule 2020 begegnen zu können, setzt der hier vorgestellte Ansatz zur Erlangung und Erhaltung einer serviceorientierten, flexiblen Hochschul-IT auf eine behutsame, aber ganzheitliche Neukonzeption aller Strukturen der Hochschule. Im Mittelpunkt der Betrachtungen steht dabei die Auseinandersetzung mit dem wichtigsten Betriebsmittel der Hochschule, den Informationen.

## 2 Moderne Hochschul-IT - Ein ganzheitlicher Ansatz

Die ganzheitliche Sicht auf die IT-Architektur einer Hochschule setzt sich vereinfacht betrachtet aus den Ergebnissen der Analyse der Bereiche Strategien, Geschäftsprozesse, Anwendungen, technologische Basis und Informationen zusammen [Bil 1].



Abbildung 1: Zusammenhänge der Bestandteile einer Architekturbetrachtung

Wie in Abbildung 1 zu erkennen ist, beeinflussen sich die einzelnen Bestandteile der Betrachtungen gegenseitig. Auffällig ist dabei, dass die Informationen im Gegensatz zu allen anderen betrachteten Elementen eine bidirektionale Beziehung zu allen Elementen besitzen. Dies unterstreicht die zentrale Bedeutung von Informationen für die IT-Architektur einer Organisation.

Damit die eingangs genannte, notwendige Evolution des Ist-Zustandes zur Hochschul-IT 2020 gelingen kann, ist es notwendig, die Geschäftsprozesse der Hochschule an den Strategien und Zielen auszurichten, zu dokumentieren und auf ihrer Basis die heutige heterogene Anwendungslandschaft dienstorientiert zu positionieren. Ziel der Aktivitäten muss es sein, die Anwendungen der Anwendungslandschaft zu integrieren ohne sie zu koppeln. Durch die hohe Relevanz der Information in der Architektur muss es zudem das Ziel sein, die enge Kopplung aller Elemente der Architektur mit den Informationen zu vermeiden. Aktuell ist es ein weit verbreiteter Standard, dass das Zielsystem die Geschäftslogik des Quellsystems kennen und zum Teil implementieren muss, damit Daten und Informationen verwertet werden können. Um diesem Defizit zu begegnen und die Evolution zur Hochschul-IT 2020 zu unterstützen, wurde der Ansatz ORBIT an der FernUniversität in Hagen entwickelt.

### **3 Realweltbasiertes Informationsmodell – Der Ansatz ORBIT (Objective Reality Based Information Tank)**

Von den Autoren wurde ein Ansatz entwickelt, der sich den eingangs genannten Herausforderungen stellt und es ermöglicht, eine alte, durch enge Kopplung geprägte IT-Landschaft, sukzessive in eine erweiterbare, modulare IT-Landschaft zu entwickeln und gleichzeitig die Datenqualität anzuheben. Dieser Ansatz wird im Folgenden vorgestellt.

#### **3.1 Einführung**

Wenn Anwendungen Daten direkt austauschen, sind sie gekoppelt. Je mehr Wissen die eine Anwendung von der anderen benötigt, umso enger ist die Kopplung. Ziel ist eine IT-Architektur, in der die kommunizierenden Anwendungen für den Datenaustausch nichts voneinander wissen müssen, sondern sich über einen „Vermittler“ (EAI<sup>1</sup>-Framework, Middleware, z.B. SAP PI) austauschen. Anwendungen müssen lediglich diesen Vermittler kennen und ihre Anfragen nach Daten an diesen richten.

Anwendungen verarbeiten Daten in unterschiedlichster Syntax und Semantik. Abbildung 2 zeigt den Zusammenhang zwischen Syntax, Semantik von Daten und der dazugehörigen Information in Anlehnung an das aus der Sprachwissenschaft stammende semiotische Dreieck (z.B. [OR23]):

---

<sup>1</sup> EAI = Enterprise Application Integration

Die Studierende Karin Mustermann (Matrikelnummer 5123434) hat die Prüfung Nachrichtentechnik (Pnr.:02024) mit der Note sehr gut (1,0) bestanden

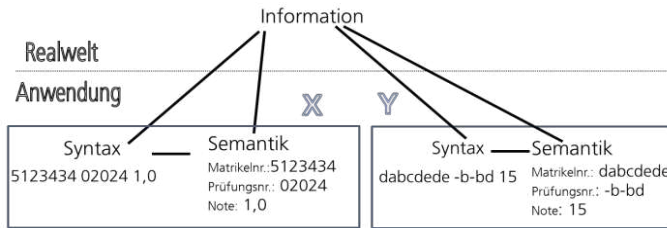


Abbildung 2: Syntax, Semantik und Information

Zwei Anwendungen X und Y speichern Prüfungsleistungen von Studierenden. Die Syntax bestimmt den Aufbau der Daten, die Semantik definiert deren Bedeutung. Zusammen bilden sie eine Information der Realwelt ab. Tauschen die Anwendungen X und Y nun direkt Prüfungsleistungen aus, muss jede von der anderen wissen, wie die auszutauschende Information abgelegt ist. Es kommt zu einer engen Kopplung.

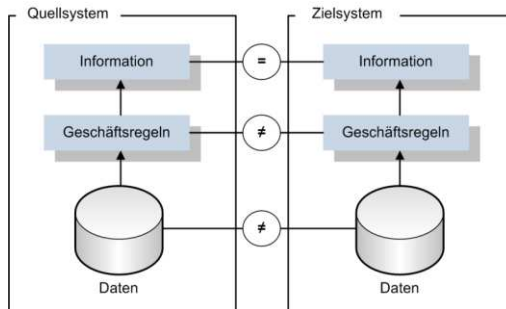


Abbildung 3: Daten, Geschäftsregeln und Information [Ma07]

Bei dem Austausch von Daten geht es aber in erster Linie um den Austausch von Information (Abbildung 3). Man möchte zwischen den Systemen die Information austauschen, dass Frau Mustermann die Prüfung Nachrichtentechnik mit der Note 1,0 bestanden hat. Wie das in der Anwendung X oder Y auf Datenebene interpretiert und in Werten ausgedrückt wird, soll dabei für die übermittelnden und empfangenden Anwendungen irrelevant sein. Der „Vermittler“ muss daher dafür sorgen, dass auf Basis eines allgemeinen Informationsmodells eine Übersetzung zwischen den Anwendungen X und Y erfolgt, damit die Informationen verstanden werden können.

Informationen anwendungsunabhängig zu speichern und der IT-Landschaft samt Prozessen zur Verfügung zu stellen, ist wichtige Voraussetzung zur Lösung der oben vorgestellten Probleme. Im Folgenden wird beschrieben, wie dies realisiert werden kann.

### 3.2 Domänenklassenmodellierung im Requirements Engineering

Das Requirements Engineering (RE) stellt Techniken, Modelle und Vorgehensweisen zur Verfügung, um die tatsächlichen Anforderungen an die zu entwickelnde Anwendung ingenieurmäßig zu erarbeiten (vgl. z.B. [Po96],[Wi05]). Um die Anforderungen eindeutig identifizieren zu können, wird ein Verständnis der Realität im Kontext der zu entwickelnden Anwendung (Problemdomäne genannt) benötigt. Zu diesem Ziel beschreibt die Domänenklassenmodellierung Strukturen der Realwelt und deren Beziehungen untereinander z.B. als UML-Klassenmodell. Wichtige Voraussetzung für den Erfolg einer solchen Modellierung ist die Problemadäquatheit: Die Realität der Problemdomäne bestimmt die Modellierung, nicht mögliche Implementierungsdetails. Alles Modellerte muss eine Entsprechung in der Realität haben.

### 3.3 Vom Domänenklassenmodell zum konzeptuellen Informationsmodell - Theoretische Betrachtung

Informationen gehören zur Realität, sind durch Zustände und Beziehungen von Personen, Gegenständen bzw. Aspekten der Realität ausdrückbar. Das Domänenklassenmodell modelliert Gegenstände und Beziehungen der Realität. Also lag es nahe, Informationen und Domänenklassenmodellierung in Verbindung zu setzen: Informationen lassen sich ausdrücken als Objektzustände eines Domänenklassenmodells. Im Folgenden wird ein Domänenklassenmodell zur Speicherung von Informationen konzeptuelles Informationsmodell genannt.

In [WW96] wird ein ontologischer Ansatz von Datenqualität definiert. Grundlage bildet das Abbilden von Zuständen der Realwelt in Zustände des Informationssystems (Repräsentation, Rep) und umgekehrt (Interpretation, Int). Dabei führen die Autoren nicht aus, was genau unter Zuständen der Realwelt und Zuständen des Informationssystems zu verstehen ist oder wie die Abbildungen definiert werden können.

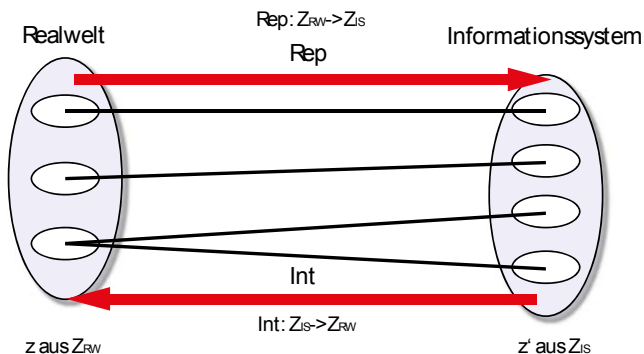


Abbildung 4: Der ontologische Ansatz

Bei der Archivierung oder der Datenmigration ist der Informationserhalt bzw. der Informationstransport entscheidend. Dazu muss in der Regel nicht nur die Menge der

Zustände des Informationssystems  $Z_{IS}$ , sondern auch die Interpretation  $Int$  archiviert werden. Letzteres z.B. durch Archivierung der Anwendung, die in der Lage ist, die Interpretation auszuführen. Das stellt zum einen großen Aufwand dar, zum anderen ist nicht sichergestellt, dass diese Anwendung nach 10 Jahren noch lauffähig ist. Die Interpretation würde im Fall der Nichtfunktion der Anwendung unmöglich sein. Hier setzt der neue Ansatz an, erweitert den ontologischen Ansatz und definiert den Begriff Information, wobei der Kontext  $K$  dabei die Menge aller Realweltzustände der Domäne ist:

„Information ( $I_K$ ) ist die objektiviert vorhandene Kombination von Daten einer bestimmten Syntax aus den Datenbanken der Domäne ( $D_D$ ) mit in der Domäne bekannter Semantik ( $S_B$ ) in einem Kontext  $K$ .“

$$I_K = \{x \in D_D \times S_B \mid Int(x) \in K\}$$

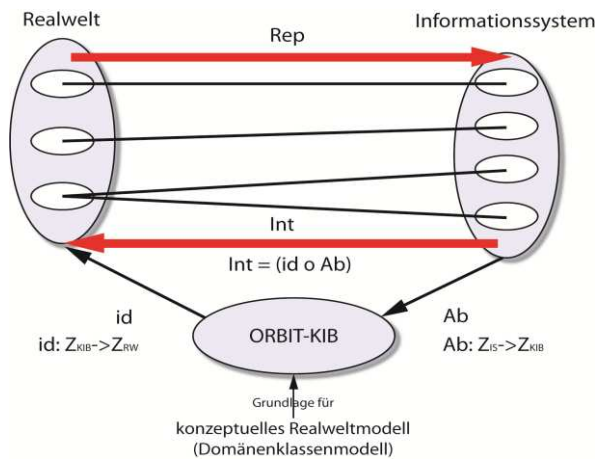


Abbildung 5: Erweiterung des ontologischen Ansatzes durch ORBIT

Ziel von ORBIT ist es, die bekannte Semantik der Domäne zu dokumentieren, so dass daraus ein konzeptuelles Informationsmodell erstellt werden kann. Zu diesem Zweck wird eine Top-Down (Geschäftsprozesse, Anwendungsfälle) und Bottom-Up (Datenbankmodelle und Quellcode der Altsysteme)-Analyse aller semantiktragenden Dokumente der Domäne durchgeführt. Die Ergebnisse werden im konzeptuellen Informationsmodell dokumentiert. Ein Mehrwert dieses Vorgehens ist die Nachvollziehbarkeit des Einflusses von Prozessen und Anforderungen auf die Informationen der Domäne.

Zudem kann ein solches konzeptuelles Informationsmodell für die Domäne Hochschule in Form eines Referenzmodells für mehrere Hochschulen dazu genutzt werden, semantische Synchronizität zu erreichen. Hochschulübergreifende IT-Projekte werden in ihren Risiken minimiert, da der Ansatz den Austausch der Informationen stark vereinfacht.

Auf Basis des konzeptuellen Informationsmodells wird z.B. unter Nutzung der XML oder durch Implementierung eines Operational Data Stores (ODS)<sup>2</sup> ein anwendungsunabhängiger Speicher für Informationen gebildet, die konzeptuelle Informationsbasis (KIB). Zustände des Informationssystems werden dabei abgebildet in (Objekt-)Zustände der KIB (Funktion Ab). Aufgrund der Definition des konzeptuellen Informationsmodell ist die Abbildung von Zuständen der KIB in Zustände der Realwelt kanonisch möglich (Funktion id). Durch die anwendungsunabhängige, zeitlose Speicherung von Informationen in ORBIT reicht es zur Archivierung nun aus, nur diese Zustände zu archivieren. Die Funktion Int und damit die Altanwendung ist im Rahmen der Archivierung nicht mehr zu betrachten, denn es gilt:  $Int = (id \circ Ab)$  (s. Abbildung 5). Informationen liegen anwendungsunabhängig abgespeichert vor.

### **3.4 Stabilität von ORBIT durch ständige Validierung des konzeptuellen Informationsmodells**

Im Folgenden wird beschrieben, dass diese Art der Informationsspeicherung deutlich zeitloser ist als die Speicherung der Anwendungsdaten. [Hb11] zeigt einen Ausschnitt des Vorlesungsverzeichnisses der Universität Heidelberg aus dem Jahre 1789. Im Rahmen einer Fallstudie zum Ansatz konnte dieser Ausschnitt sofort als Informationen in ORBIT-KIB gespeichert werden, da das zugrundeliegende konzeptuelle Informationsmodell in den relevanten Elementen stabil ist. So sind im Vorlesungsverzeichnis beispielsweise Informationen darüber zu erkennen, wie das Lehrangebot Gottesgelehrtheit aufgebaut ist. Die Informationen über Vorlesungen, Dozenten, Zeiten etc. können entnommen und mit ihnen Objekte der KIB erzeugt werden. Nicht direkt aus dem Vorlesungsverzeichnis zu entnehmende Informationen (Orte, Wochentage etc.) müssen eventuell über weitere Quellenrecherchen ermittelt werden, um die Menge der Informationen zu ergänzen. Informationen die im Jahr 1789 definitiv nicht zur Verfügung standen (z.B. ECTS-Punkte) werden dabei nicht betrachtet.

Generell kann damit die aus dem konzeptuellen Informationsmodell (s. Abbildung 6) abgeleitete ORBIT-KIB aus dem Jahr 2011 Angaben aus dem Vorlesungsverzeichnis des Jahres 1789 aufnehmen. Nach ORBIT gesicherte Informationen hätten somit 222 Jahre überdauert und wären auch weiter les- und auswertbar. Das zu Grunde liegende konzeptuelle Informationsmodell ist seit Jahrhunderten in den entscheidenden Aspekten stabil.

Zwar sind Informationen langlebiger als die Anwendungsebene, aber trotzdem führen neue Anforderungen, sich ändernde Nebenbedingungen wie rechtliche Vorgaben oder Veränderungen in der IT-Welt zu notwendigen Anpassungen des Informationsmodells. Dieses muss ständig im Rahmen eines Datenmanagementansatzes validiert und ggf. angepasst werden. Wäre das konzeptuelle Informationsmodell aus Abbildung 6 im Jahre 1789 entstanden, hätte es für den heutigen Einsatz z.B. um den Aspekt der ECTS-Punkte erweitert werden müssen.

---

<sup>2</sup> Realisiert als eigene Datenbank, deren Schemata das konzeptuelle Informationsmodell umsetzen

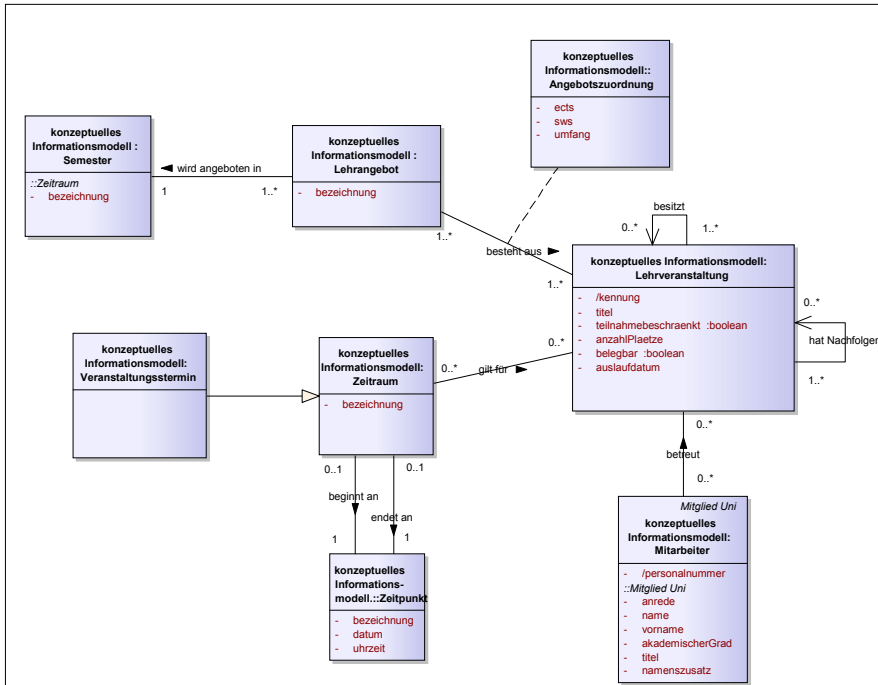


Abbildung 6: Konzeptuelles Informationsmodell (Ausschnitt)

## 4 Praktische Relevanz von ORBIT

Im Folgenden wird die praktische Relevanz von ORBIT aufgezeigt. Die gewonnenen Erkenntnisse sind insbesondere für die meist heterogene IT-Umgebung Hochschule wie auch für alle anderen Domänen wie Industrieunternehmen, Behörden etc. nutzbar.

### 4.1 ORBIT-basiertes Datenmanagement

Schlechte Datenqualität ist teuer. Eine Studie der TDWI Research Group zeigte 2002 [T02], dass der amerikanischen Wirtschaft jährlich ein Schaden von 600 Milliarden Dollar allein aufgrund schlechter Datenqualität entsteht. Schlechte Datenqualität führt insbesondere durch Kopplung der Anwendungen zu einer schlechten Wartbarkeit und Erweiterbarkeit und ist typisch für eine lange im Einsatz befindliche IT-Landschaft. Neben inhaltlichen Datendefiziten (fehlerhafte Adressen, falsche Namen etc.) spielen strukturelle Probleme wie Feldumdeutungen eine wichtige Rolle. Weitergehende Informationen zu diesem Thema sind z.B. in [Ho08] zu finden. ORBIT erlaubt ein effizientes und effektives Datenmanagement zum Erreichen und Halten hoher Datenqualität.



[WW96] definiert schlechte Datenqualität über Widersprüche, die sich aus der realen Welt und der Abbildung dieser in ein IT-System ergeben. Sowohl die inhaltliche als auch die strukturelle Ebene von Datenqualität wird dabei berücksichtigt. Hohe Datenqualität liegt dann vor, wenn Daten „einfach“ zu einer Realweltinformation abbildbar sind. Die Abbildung Int (Abbildung 4) spielt eine zentrale Rolle bei dieser ontologischen Definition von Datenqualität. Die Auswirkungen schlechter Datenqualität sind weitreichend und teuer. Wegen der immensen Bedeutung von Daten für eine Universität sollte man diese wie ein eigenständiges, von den Anwendungen losgelöstes Betriebsmittel auffassen. Es ist Aufgabe eines Datenmanagements dies sicherzustellen.

Aufgrund der Definition von ORBIT wird auf Basis des oben vorgestellten ontologischen Datenqualitätsbegriffs deutlich, dass hohe strukturelle Datenqualität vorliegt: ORBIT mit der KIB als Informationsspeicher ist speziell so konzipiert worden. Inhaltliche Datendefizite werden dabei nicht sofort behoben, können aber aufgrund der kanonischen Abbildung in die Realweltzustände einfacher identifiziert werden als in der alten IT-Landschaft.

Zur Nutzung der gewonnenen Erkenntnisse über Datenqualität wird in [BH10] beschrieben, wie ein Datenmanagement basierend auf Realweltmodellierung funktionieren kann. Oberstes Ziel dabei ist die Etablierung von Daten als eigenständiges, von den Anwendungen losgelöstes Betriebsmittel. Dies ist eine der wichtigsten Voraussetzungen für eine änderbare, flexibel erweiterbare Anwendungslandschaft. Abbildung 7 beschreibt den an der FernUniversität in Hagen eingeschlagenen Weg des Datenmanagements. Kernaufgabe dabei ist die ständige Validierung des Informationsmodells (s. Kapitel 3.4) und der Abbildungsregeln Ist-Datenbestand zu Informationen.



Abbildung 7: Der Weg zum Betriebsmittel "Daten"

## 4.2 ORBIT-basierte Entkopplung und Integration von Anwendungen

Eine weitere grundlegende Voraussetzung für eine flexibel erweiterbare Anwendungslandschaft ist neben einem effizienten Datenmanagement die Entkopplung

der Anwendungen von den Daten und die Entkopplung der Anwendungen untereinander. Die nachfolgende Abbildung beschreibt dabei den Weg, eine Ist-Anwendungslandschaft in eine Sollarchitektur zu überführen.

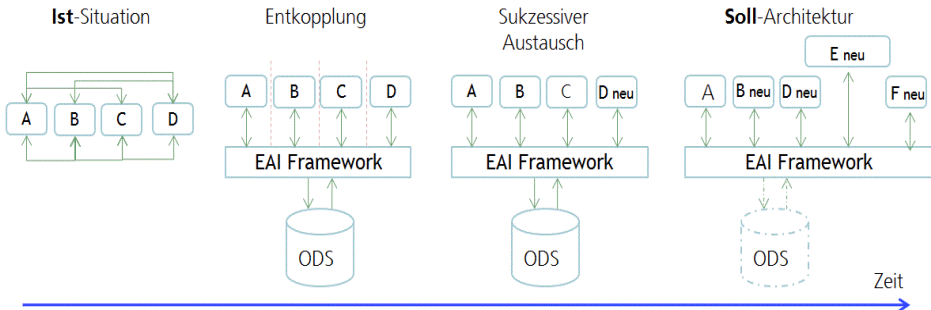


Abbildung 8: Von der Ist-Situation zur Soll-Architektur

Die Ist-Situation ist geprägt durch individuelle Schnittstellen zwischen Anwendungen, direkte Kenntnisse der Systeme untereinander sind erforderlich, es liegen direkte Kopplungen vor. Die Anwendungslandschaft ist schlecht wartbar und nicht erweiterbar.

Notwendige Tätigkeit für alle weiteren Schritte ist die Entkopplung der Anwendungen. Dazu werden die auszutauschenden Daten als Informationen anwendungsneutral abgelegt. Dies geschieht mit Hilfe eines ODS, der auf Basis des konzeptuellen Informationsmodells Informationen speichert. Über ein EAI-Framework stehen die Informationen allen Anwendungen zur Verfügung. Eine Aufgabe dieses EAI-Frameworks ist die Abbildung der anwendungsunabhängig im ODS liegenden Informationen in die von der anfordernden Anwendung benötigte Daten-Syntax und -Semantik.

Nach der erfolgten Entkopplung können die Anwendungen sukzessive ausgetauscht werden. Es muss vor dem Austausch sichergestellt sein, dass eine neue Anwendung mindestens die Informationen für den ODS zur Verfügung stellt, die von der Altanwendung bereitgestellt wurden und selber benötigte im ODS zur Verfügung stehen.

Über mehrere Iterationen wird die Zielarchitektur erreicht. Diese Architektur ist flexibel und erweiterbar. Die Anwendungen sind integriert, aber nicht gekoppelt, da diese nur mit dem EAI-Framework kommunizieren und andere Anwendungen nicht kennen.

### 4.3 ORBIT-basierte Datenmigration

Im Rahmen der dargelegten Evolution zur Soll-Architektur ist die Datenmigration eine notwendige und immer wiederkehrende Tätigkeit. Doch diese Tätigkeit ist, klassisch durchgeführt, risikobehaftet, wie in [HP07] dargelegt wird: 84% aller Datenmigrationsprojekte werden nicht im Zeit- und Budgetrahmen abgeschlossen und stellen ein hohes Risiko für ein IT-Projekt zur Erneuerung der Anwendungslandschaft dar. ORBIT senkt dieses Risiko signifikant, da mit dem anwendungsunabhängigen

Informationsspeicher eine sofort nutzbare Migrationsgrundlage ständig existiert. Im Folgenden wird die ORBIT-basierte Migration vorgestellt.

In der Regel kann man Daten aus Quellsystemen nicht direkt in ein Zielsystem migrieren, da die Ausgangsdatenqualität niedrig ist und viele Vorarbeiten zur Identifikation und Behebung der Defizite zu leisten sind. Würde man mit Migrationsüberlegungen inkl. der Anhebung der Datenqualität erst dann starten, wenn das Zielsystem bekannt ist, verzögerte dies das Gesamtprojekt erheblich. Neben dem Management der Daten auf Basis des konzeptuellen Informationsmodells, wird die Migration von Daten aus Quell- in Zielanwendungen durch ORBIT vereinfacht. Dazu wird das konzeptuelle Informationsmodell um Datensenzen der Zielsysteme und Transformationsregeln zur Abbildung in die Zielsysteme zu einem konzeptuellen Migrationsmodell erweitert. Da, nach ORBIT, die Daten der Altsysteme schon in der ORBIT-KIB vorhanden sind, startet die Migration der Daten von einem hohen Abstraktions- und Qualitätsniveau aus. Notwendige Vorarbeiten im Rahmen einer Datenmigration wie die Auseinandersetzung mit der Semantik und den Daten der Quelldomäne, die Anhebung der Datenqualität, die Datenextraktion etc. sind bei der Übertragung der Daten der Quellsysteme in die ORBIT-KIB schon im Vorfeld durchgeführt worden. Dies entlastet den Zeitplan einer Datenmigration in einem hohen Maße. Die Planung der Übertragung und Transformation der Daten in das Zielsystem steht somit bei einer Migrationsplanung im Vordergrund. Durch ein solches Vorgehen wird ein Migrationsprojekt entlastet. Die Erfolgswahrscheinlichkeit (Einhaltung von Zeit und Kosten) erhöht sich. Abbildung 9 fasst das Vorgehen zusammen.

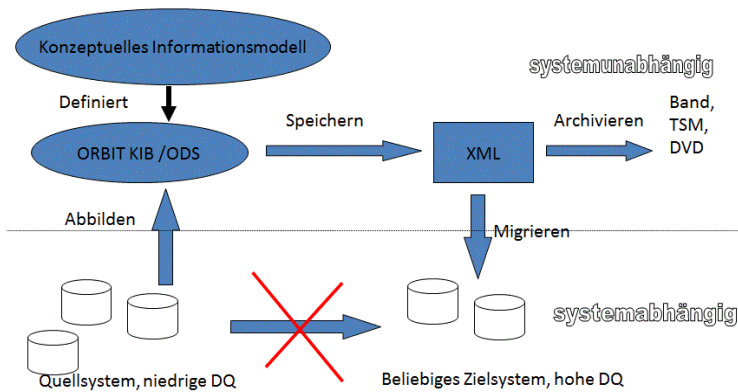


Abbildung 9: ORBIT-basierte Datenmigration

#### 4.4 ORBIT- und modellbasierter Software-Engineering Ansatz

In einer IT-Landschaft müssen regelmäßig neue Anforderungen realisiert werden. Mit Hilfe von ORBIT wurde eine modellbasierte Software-Engineering Methodik entwickelt, welche den Ansatz [Wi05] um explizite Geschäftsprozessmodellierung und Informationserhebung im Kontext der formalen Anwendungsfallbeschreibungen

erweitert. Sie erlaubt die Umsetzung neuer Anforderungen in hoher Softwarequalität und begegnet der Gefahr, die Flexibilität und Erweiterbarkeit der Anwendungslandschaft zu reduzieren. Abbildung 10 zeigt die dabei berücksichtigten Modellebenen am Beispiel des Level 0 Prozesses „Studierendenmanagement“. Es werden Geschäftsprozesse der Ebenen 0 bis 2 erfasst, Anwendungsfälle modelliert und textuell formuliert. Aus diesen werden IT-Dienste abgeleitet. Über alle Aktivitäten hinweg steht das Informationsmodell im Focus der Tätigkeiten. Auf Ebene der Prozesse werden Businessobjekte identifiziert und dadurch ein Glossar definiert. Auf Ebene der Anwendungsfälle werden Informationen identifiziert, die erzeugt, verarbeitet und zwischen Anwendungsfällen ausgetauscht werden.

Dieses Vorgehen hat mehrere Vorteile: Zum einen wird das realweltbasierte Informationsmodell ständig validiert, zum anderen definiert das Informationsmodell über alle Aktivitäten hinweg einen einheitlichen Informationsraum, definiert ein Glossar. Die Verständlichkeit nimmt deutlich zu, Konsistenzkriterien sind definierbar, Schwächen identifizierbar. Beispiele: Zu einem Business-Objekt gehört keine Information auf Anwendungsfallebene oder umgekehrt. Auf Datenbankebene werden fachliche Daten gespeichert, die keine Information abbilden. Letzteres deutet auf eine unvollständige Prozessmodellierung hin.

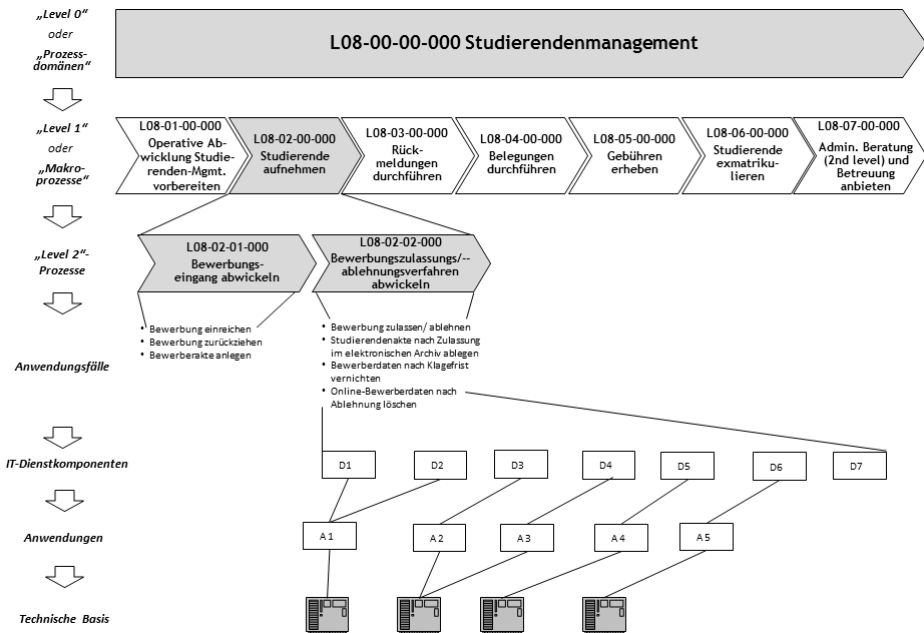


Abbildung 10: Modellebenen des ORBIT-basierten Software-Engineering-Ansatzes

Dieses Vorgehen ermöglicht eine Verfolgbarkeit über die verschiedenen Abstraktionsstufen hinweg. Zur Laufzeit der Anwendungslandschaft können z.B. beim Ausfall einer Anwendung oder eines Dienstes sofort die betroffenen Prozesse identifiziert werden.

## 5 Verwandte Ansätze und Erfahrungen im universitären Umfeld

Domänenklassenmodellierung wird im Requirements Engineering eingesetzt, um ein Grundverständnis der Domäne zu erhalten, in dessen Kontext eine Anwendung zu entwickeln ist (z.B. [Wi05]). Zur Definition und anwendungsunabhängigen Speicherung der Domäneninformationen wird das Domänenklassenmodell nicht verwendet.

1975 wurde von ANSI/SPARC [AS75] die Dreischichten-Architektur für Datenbanken entwickelt, die eine konzeptionelle Ebene enthält. Seit dieser Zeit ist der konzeptionelle Entwurf fester Bestandteil des Datenbankentwurfs. In einer von der Zieldatenbank abstrahierenden Form werden die Daten, die eine Anwendung speichern soll, mit Realweltbezug beschrieben und beispielsweise in einem ER-Modell [Ch76] festgehalten (z.B. [Ja10]). Auf Basis des konzeptionellen Entwurfs finden der logische und physische Entwurf sowie die tatsächliche Implementierung der Datenbank statt. Für die Kommunikation von Anwendungen untereinander hat die konzeptionelle Ebene keine Bedeutung, die Anwendungen tauschen Daten, keine Informationen aus. Das führt zu einer Kopplung der Anwendungen.

Der Ansatz von ORBIT ist umfassender: Über alle Ebenen der Enterprise Architektur hinweg werden Informationen für das konzeptuelle Informationsmodell gesammelt und gespeichert, nicht nur im Focus einer zu entwickelnden Anwendung, sondern auch rein realitätsbezogen auf Ebene der Strategie bzw. der Nicht-IT-unterstützten und IT-unterstützten Geschäftsprozesse (Abbildung 11). Durch ORBIT erhält man eine Verfolgbarkeit von Aspekten der strategischen Ebenen bis zur technischen Abbildung. Zudem erleichtert ORBIT die Kommunikation zwischen den Ebenen, da eine einheitliche Informationswelt definiert wird [BH10]. Der konzeptionelle Datenbankentwurf ist ein Ausschnitt des konzeptionellen Informationsmodells.



Abbildung 11: Ebenen einer ORBIT-basierten Enterprise Architektur

Als weiteren Mehrwert definiert ORBIT eine gemeinsame, zeitlose Sprache, welche die Anwendungen „sprechen“ und ermöglicht so über den Einsatz eines EAI-Frameworks die Integration von Anwendungen ohne Kopplung.

Dabei ist in Bezug auf eine servicebezogene Integration der Enterprise Service Bus seit vielen Jahren fester Bestandteil einer Service-orientierten IT-Landschaft (z.B. [Ch04]). Dieser Begriff bezeichnet dabei mehr ein Werkzeug als ein Konzept. ORBIT stellt im Unterschied dazu sowohl einen konzeptionellen Integrationsrahmen als auch ein

Implementierungswerkzeug dar, der durch den genutzten Realweltbezug im EAI-Framework in weiten Teilen deutlich stabiler ist, als klassische SOA-Ansätze. (s. Kapitel 3.4)

Zudem gibt es eine Vielzahl von Data Mediation Ansätzen, die das Ziel haben, Daten zwischen heterogene Datenquellen zu tauschen (z.B. [MC07]). Solche Ansätze arbeiten mit Datenreferenzmodellen, die die Schnittmenge der zu tauschenden Daten repräsentieren [DM]. ORBIT wählt einen anderen Ansatz: Im anwendungsunabhängigen Informationsmodell werden alle relevanten Informationen der Domäne speicherbar, unabhängig davon, ob diese als Daten zwischen Anwendungen getauscht werden sollen. Die Menge zu tauschender Informationen ist als echte Teilmenge im Modell enthalten.

Allgemein ist ORBIT in eine modellbasierte Softwareentwicklungsmethodik eingebunden, die den modellbasierten Ansatz aus [Wi05] erweitert. Diese Methodik ermöglicht eine einfache, angemessene Umsetzung erarbeiteter Anforderungen bei Beibehaltung bzw. Schaffung hoher Daten- und IT-Architekturqualität, was sich in hoher Änderbarkeit, Erweiterbarkeit und Modularität zeigt. Insbesondere sind die Lücken zwischen den tatsächlichen Kundenwünschen und den dokumentierten Anforderungen sowie zwischen den Anforderungen und der IT-Realisierung klein und kontrollierbar zu überbrücken.

ORBIT erweitert ferner den ontologischen Ansatz zur Definition von Datenqualität [WW96] und ermöglicht eine konkrete, praktisch nutzbare Beschreibung der Abbildung Int. Dies erlaubt die Speicherung von Informationen und erleichtert insbesondere die Archivierung von Daten. So sind keine Anwendungen mehr zu archivieren, mit denen Jahre später archivierte Daten interpretiert werden. Im Rahmen einer Fallstudie an der FernUniversität in Hagen wurden über ORBIT alle relevanten Hochschulinformationen in XML-Dateien gespeichert, die gezippt eine Größe von nur 50 GB einnehmen.

Die Erfahrungen, die mit ORBIT im Rahmen eines großen Projekts an der FernUniversität in Hagen bisher gemacht wurden und werden, bestätigen den großen Nutzen des Ansatzes. Dort läuft aktuell das mehrjährige Projekt Hagen System Relaunch (hs.r) zur Erneuerung der IT-Landschaft im Campusmanagementbereich. Mit der in Kapitel 4.4 beschriebenen Methodik entstehen mehrere hundert Anwendungsfälle, die in die Lastenhefte einfließen und Grundlage der Abnahmetests bilden. Die Entkopplung der Altanwendungen erfolgt über den in Kapitel 4.2 dargelegten Ansatz. Als EAI-Framework wird SAP PI benutzt, alle Informationen des Campusmanagementbereichs liegen in einem ORBIT-ODS vor, der als ORACLE-Datenbank realisiert ist. Das konzeptuelle Informationsmodell existiert für den Campusmanagementbereich. Die technische Datenmigration erfolgt in hs.r nach dem in Kapitel 3.7 beschriebenen Konzept. Dadurch wurde und wird eine hohe Zeit- und insbesondere Kostenersparnis erreicht.

Das in Kapitel 4.3 beschriebene Datenmanagement ist in eine Linienaufgabe des Zentrums für Medien und IT (ZMI) der FernUniversität in Hagen übergegangen und hat zur deutlichen Anhebung der Datenqualität geführt. Durch die ständige Validierung des konzeptuellen Informationsmodells liegt zu jedem Zeitpunkt ein validiertes Domänenklassenmodell vor, das in neuen IT-Projekten unmittelbar verwendet werden kann.

## 6 Literaturverzeichnis

- [AS75] Steel: Interim report of the ANSI-SPARC study group. In ACM SIGMOD Conf. on the Management of Data, 1975
- [BH10] Baumöl, U.; Homrighausen, A.: Datenmanagement mittels Realweltmodellierung. In: Controlling, Zeitschrift für erfolgsorientierte Unternehmenssteuerung, 8/9 August/September 2010, Verlage C.H.Beck, Vahlen, München, Frankfurt a.M. 2010; S.497-500.
- [Bi11] BITKOM: Enterprise Architecture Management – neue Disziplin für die ganzheitliche Unternehmensentwicklung; Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V., Berlin, 2011
- [Ch04] Dave Chappell: Enterprise Service Bus. O'Reilly. Juni 2004
- [Ch76] Chen, Peter Pin-Shan: The Entity-Relationship Model--Toward a Unified View of Data. In: ACM Transactions on Database Systems 1976 ACM-Press ISSN , S. 9–36
- [DM] Data Mediation Forum: [http://www.dm-forum.org/deu/dmf\\_vorteile.htm](http://www.dm-forum.org/deu/dmf_vorteile.htm)
- [Hb11] Heidelberger historische Bände – digital;<http://digi.ub.uni-heidelberg.de/diglit/VV1784WSbis1790SS/0149>
- [Ho08] Homrighausen, A.: Realweltmodellierung-Ein Weg aus der Datenqualitätskrise. In: Tagungsband Lernen Organisation Gesellschaft-logOS08, epOs media, Osnabrück 2008, Seiten 86-95.
- [HP07] Howard, P.; Potter, C.: Data Migration in the Global 2000; Bloor Research 2007.
- [Ja10] Jarosch, Helmut: Grundkurs Datenbankentwurf; Springer Vieweg, 2010.
- [KMK05] Kultusministerkonferenz: Prognose der Studienanfänger, Studierenden und Hochschulabsolventen bis 2020, Sekretariat der Ständigen Konferenz der Kultusminister der Länder in der Bundesrepublik Deutschland, Bonn, 2005
- [LR11] Leistert, Oliver; Röhle, Theo; Generation Facebook - über das Leben im Social Net; Transcript, Bielefeld, 2011
- [Ma07] Maydanchik, Arkady: Data Quality Assessment; Hrsg. Technics Publications, LLC Bradley Beach 2007
- [MC07] Mocan,A; Cimpian, E: An Ontology Based Data Mediation Framework For Semantic Environments; International Journal on Semantic Web and Information Systems, Vol.3 Iss. 2, IGI Global, 2007
- [OR23] Ogden, C.; Richards I.A.: Meaning of Meaning (1923). London, Boston and Henley, ARK Paperbacks, 1985.
- [Po96] Pohl, K.: Requirements Engineering: An Overview. In Aachener Informatik- Berichte Nr. 96-05. RWTH Aachen, 1996.
- [T02] TDWI Research Group: Data Quality and the Bottom Line: Achieving Business Success through a Commitment to High Quality Data, 2002.
- [Wi05] Winter, M.: Methodische Objektorientierte Softwareentwicklung. dpunkt.Verlag 2005; S.187.
- [WW96] Wand,Y.; Wang, R: Anchoring Data Quality Dimensions in Ontological Foundations. In: Communications of the ACM, November 1996/Vol.39, No.11, S.86-95.