Agile modellorientierte DWH-Entwicklung – Modellebenen und Architektur

White Paper der solvistas GmbH

Thomas Neuböck
Jürgen Raab
Andreas Weißenböck

Linz, Dezember 2014

1 Einleitung

In (Neuböck T., Raab J., Weißenböck A., 2014) wurde ein Ansatz zur modellorientierten DWH-Entwicklung vorgestellt, die in einer agilen Vorgehensweise eingebettet ist. Dabei sind die in Abbildung 1 dargestellten Projektdimensionen (Sichtweisen) zu beachten (siehe Abschnitt 2): Managementsicht und Vorgehensweise, fachliche Sicht, Architektursicht und Modellsicht. Diese agile modellorientierte Vorgehensweise wurde in einer deutschen öffentlichen Krankenversicherung eingeführt. Eine Umsetzung dieser Vorgehensweise auf Basis einer konkreten Umgebung (IBM Netezza und IBM InfoSphere) wird in (Hiebl J., Hörak K., Linner K., Neuböck T., Raab J., Weißenböck A., 2014) gezeigt.

Der Fokus in diesem White Paper liegt auf der Architektur des BI- und DWH-Systems (3-Schichtenarchitektur) und auf den drei Modellebenen. Diese Präsentation stellt eine konsolidierte Version der Arbeiten in (Neuböck T., Raab J., Weißenböck A., 2014) und (Hiebl J., Hörak K., Linner K., Neuböck T., Raab J., Weißenböck A., 2014) dar, wobei die Agilität hier nicht mehr näher beschrieben wird, jedoch als inhärenter Bestandteil der propagierten Vorgehensweise zu betrachten ist.

In der Architekturdarstellung von Abschnitt 3 wird neben den drei Datenschichten (historisierte Datenschicht, integrierte Datenschicht und Data Mart Schicht) auch die Präsentationsschicht hervorgehoben. Die ETL-Umsetzung und die Bereitstellung unterschiedlicher Umgebungen für Entwicklung, Test und Produktion wird ebenfalls beschrieben. Abschnitt 4 zeigt die Modellebenen der modellorientierten Vorgehensweise. Die genauere Betrachtung liegt auf der fachlichen und logischen Modellebene. Als Abschluss folgt ein kurzes Fazit in Abschnitt 5.

2 Überblick

Es folgt ein knapper Überblick über die gesamte Vorgehensweis. Dieser wird in Abbildung 1 mit der Darstellung der vier Projektdimensionen gestartet.

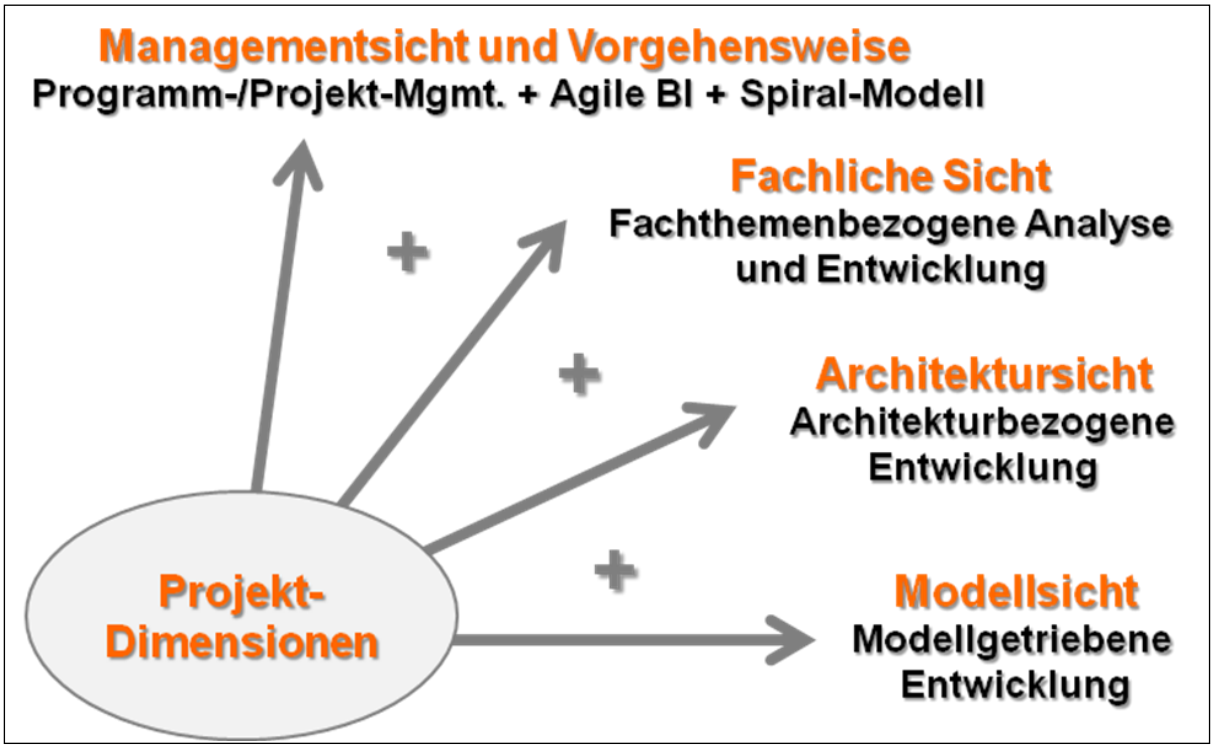


Abbildung 1

Im übergeordnetes Programm- und Projektmanagement werden Projekte in einer agilen Weise in Anlehnung an ein Spiral-Modell durchgeführt (agile BI). Das Vorgehensmodell erleichtert eine fachthemenbezogene DWH-Entwicklung.

Im Vorgehensmodell ist eine Umsetzung gemäß der vorgegebenen DWH-Architektur zu gewährleisten. DWH-Schichten sind voneinander zu trennen und zwischen den Schichten sind Abbildungsprozesse definiert. Es werden die folgenden DWH-Schichten geführt: Historisierte Datenschicht (HDS), Integrierte Datenschicht (IDS), Data Mart Schicht (DMS). Im gesamten BI-System ist noch eine Präsentationsschicht (Frontends) zu berücksichtigen, welche an der DMS anschließt.

Das vorgeschlagene Vorgehensmodell fordert eine modellgetriebene DWH-Entwicklung. Modelle erlauben die Einbringung verschiedener Sichtweisen und ergeben eine formale Dokumentation. Drei Modellebenen werden unterschieden:

- Fachliche (Konzeptuelle) Modellschicht
- Logische Modellschicht
- Physische Modellschicht

Abbildung 2 visualisiert diese vier Projektdimensionen in einer kombinierten Art und Weise.

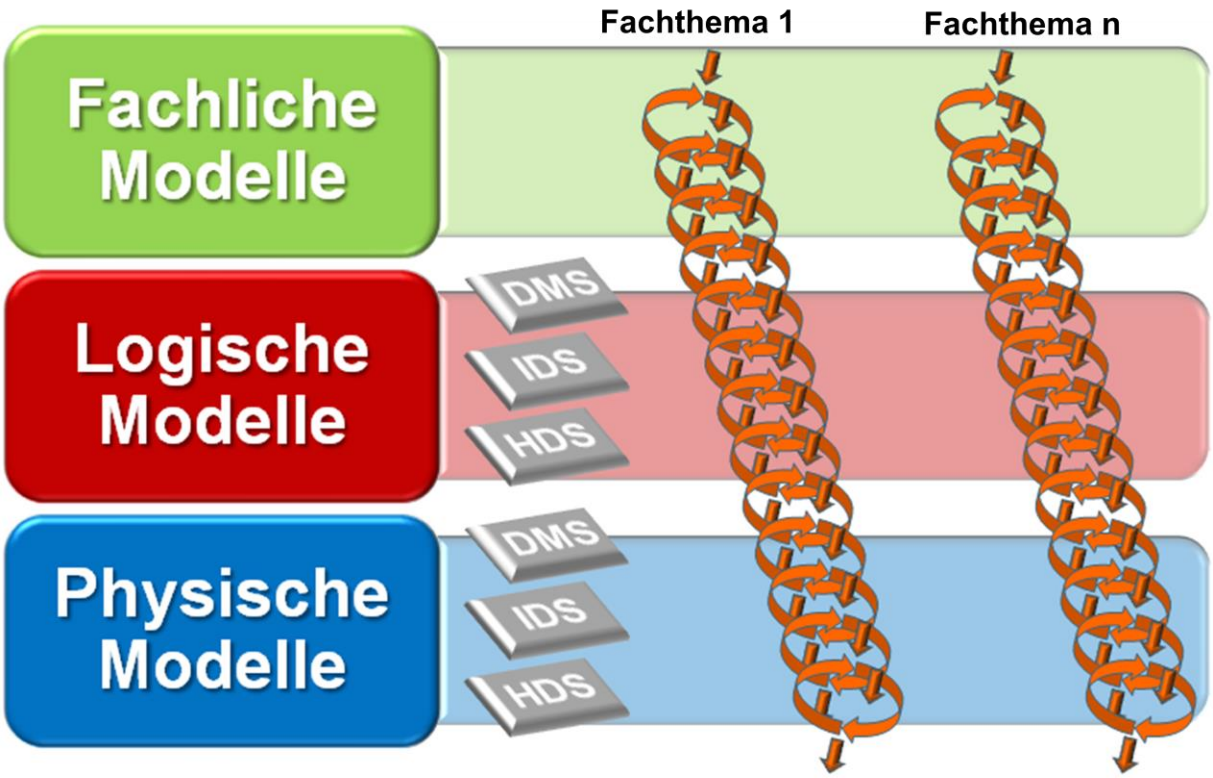


Abbildung 2

Fachthemen werden iterativ umgesetzt und liefern inkrementelle Ergebnisse („orange Spiralen“). Bei der Umsetzung eines Fachthemas wird modellorientiert vorgegangen. Zuerst werden fachliche Modell und anschließend logische Modelle erstellt. Physische Modelle liefern die direkte Implementierungsanweisungen (z.B. DDL-Skripts zur Erstellung von Datenbankschemata). Auf der logischen und physischen Modellschicht ist auf eine architekturkonforme Umsetzung (HDS, IDS, DMS, Präsentationsschicht) zu achten.

3 Architektur

In der DWH-Architektur werden folgende Schichten gepflegt:

- Präsentationsschicht (PRS)
- Data Mart Schicht (DMS)
- Integrierte Datenschicht (IDS)
- Historisierte Datenschicht (HDS)

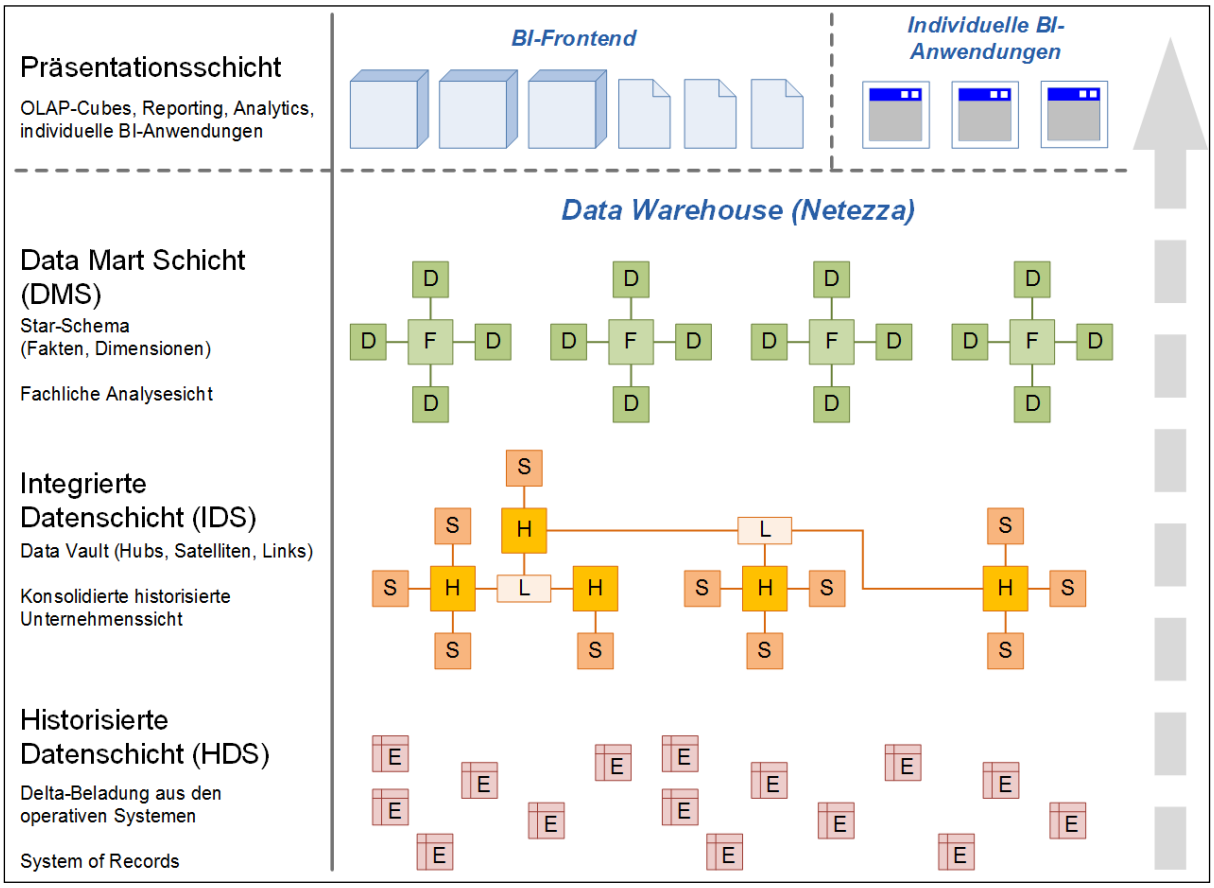


Abbildung 3

Die DMS, IDS und HDS stellen die Datenschichten des DWHs dar. Sie werden beispielsweise in der DWH-Appliance IBM Pure Data for Analytics (Netezza) verwaltet. Die Präsentationsschicht stellte Frontends zur Verfügung, welche auf die Daten der DMS zugreifen und diese für die Analyse und für das Reporting aufbereiten und darstellen. Falls erforderlich können individuell entwickelte BI-Anwendungen ebenfalls die Daten der DMS verarbeiten. Die Präsentationssicht wird nicht zum eigentlichen Data Warehouse, das auf der Netezza liegt, gezählt. Sie ist aber ein Bestandteil des gesamten BI-Systems.

3.1 Präsentationsschicht

Die Präsentationsschicht stellte die Zugriffsschicht auf oberster Ebene dar, mit der ein Benutzer konfrontiert ist (z.B. Berichte, BI-Applikationen). Die Präsentationsschicht greift ausschließlich auf die DMS zu. Ein Durchgriff direkt auf die IDS oder HDS ist prinzipiell nicht erlaubt.

3.2 Data Mart Schicht (DMS)

In der Data Mart Schicht (DMS) werden alle Fachthemen abgebildet. Diese Schicht bildet die Basis für die Präsentationsschicht. Die DMS wird ausschließlich aus der IDS aufgebaut. Kennzahlen sollten weitgehend bereits in der IDS berechnet werden.

Die Daten der DMS werden in einem Star-Schema angeboten. Ein Star-Schema gliedert die Tabellen in Dimensionen und Fakten. Mit Conformed Dimensions (mehrere Stars verwenden gemeinsame Dimensionen) wird der Wiederverwendbarkeitsgrad erhöht.

3.3 Integrierte Datenschicht (IDS)

Die Integrierte Datenschicht (IDS) beinhaltet alle Daten des Unternehmens in einer konsolidierten, historisierten und leicht erweiterbaren Form. Sie stellt den Single Point of Truth des gesamten DWHs dar. Die IDS enthält auch alle erforderlichen Kennzahlen, die später in der DMS benötigt werden.

Um die Anforderung der IDS zu erfüllen, werden die Datenstrukturen nach einem Data Vault Modell umgesetzt. Die Data Vault Modellierung liegt zwischen der Modellierung in dritter Normalform und der Modellierung in Star-Schemata. Es werden dabei folgende Tabellentypen unterschieden: Hubs repräsentieren Geschäftsobjekte und enthalten nur den Business Key. Die eigentlichen Informationen werden kontextspezifisch in einem oder mehreren zu einem Hub gehörenden Satelliten historisiert abgelegt. Die Beziehungen zwischen Geschäftsobjekten (Hubs) werden in separaten Tabellen, den sog. Links verwaltet. Abgeleitete Kennzahlen werden dazu in zusätzlichen Business-Satelliten abgelegt.

Data Vault ist äußerst flexibel hinsichtlich Änderungen und Erweiterungen, die sich aus der fachlichen Realität ergeben. Insbesondere mit der Trennung in Geschäftsobjekt (Hub) und dessen Daten (Satelliten) und die Umsetzung von Beziehungen über eigenen Tabellen (Links) wird diese Flexibilität unterstützt. Der Ansatz erlaubt auch eine schrittweise Umsetzung eines Fachthemas nach dem anderen – es muss nicht das gesamte DWH in einem Zuge umgesetzt werden. Durch die Aufteilung der historisierten Daten eines Geschäftsobjektes auf verschiedene Satelliten wird eine speicheroptimale Datenrepräsentationsform gewählt. Diese erlaubt auch eine bestmögliche Parallelisierung, was den Datenaufbau betrifft, was zu zusätzlichen Performance-Vorteilen führt.

Die ID wird ausschließlich aus der HDS heraus aufgebaut und dient als exklusive Quelle für den Aufbau der DMS.

3.4 Historisierte Datenschicht (HDS)

Die HDS stellt die Schnittstelle zur Staging Area eines IT-Providers dar und beinhaltet alle Delta-Beladungen aus der Staging Area heraus, die wiederum die Daten von den operativen Systemen erhält. Die Staging Area wird vom IT-Provider befüllt und dient zur Erstellung der HDS, die dann vom Kunden in den weiteren Prozessen verwendet wird. Aus DWH-Sicht stellt die HDS ein System of Records dar.

3.5 ETL-Umsetzung

Der Aufbau der einzelnen Datenschichten erfolgt beispielsweise mit IBM InfoSphere DataStage als ETL-Werkzeug und SQuirreL als allgemeines SQL-Client-Werkzeug.

Folgende Schritte werden bei der ETL-Umsetzung durchgeführt:

- Erstellung bzw. Anpassung des Datenbankschemas der HDS

- Erstellung bzw. Anpassung des Datenbankschemas der IDS

- Erstellung bzw. Anpassung des Datenbankschemas der HDS

- Entwicklung der ETL-Jobs zur Übernahme der Daten in die HDS

- Entwicklung der ETL-Jobs zum Aufbau der IDS aus der HDS heraus:
  - Entwicklung der ETL-Jobs zum Aufbau der Hubs
  - Entwicklung der ETL-Jobs zum Aufbau der Links
  - Entwicklung der ETL-Jobs zum Aufbau der Satelliten

- Entwicklung der ETL-Jobs zum Aufbau der DMS aus der IDS heraus:
  - Entwicklung der ETL-Jobs zum Aufbau der Dimensionen
  - Entwicklung der ETL-Jobs zum Aufbau der Fakten

3.6 Umgebungen

Es werden zum Entwickeln, Testen und für den Produktivbetrieb jeweils drei Umgebungen unterschieden (siehe Abbildung 4). Zur Entwicklung und für den Fachtest stehen jeweils zwei Datenbanken zur Verfügung, die auf ein und derselben Maschine (Netezza) liegen. Die Produktionsdatenbank ist auf einer separaten DWH-Appliance (Netezza) untergebracht. Analog sind Anwendungen hinsichtlich Entwicklungs-, Test- und Produktionsumgebung zu unterscheiden.

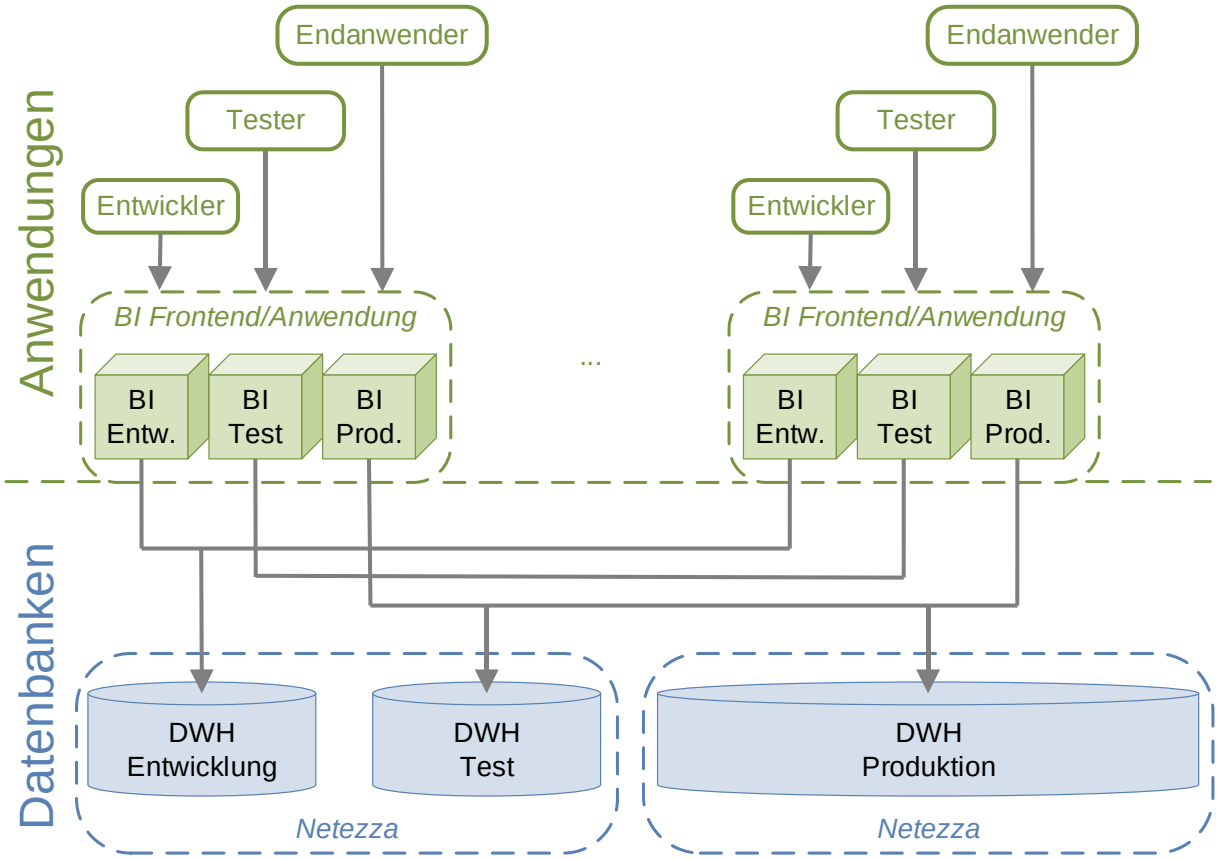

Anwendungen
Endanwender
Tester
Entwickler
BI Frontend/Anwendung
BI Entw.
BI Test
BI Prod.
...
Endanwender
Tester
Entwickler
BI Frontend/Anwendung
BI Entw.
BI Test
BI Prod.
Datenbanken
DWH Entwicklung
DWH Test
Netezza
DWH Produktion
Netezza

Abbildung 4

4 Modellorientierte Vorgehensweise

Modellorientierung steht im Zentrum der allgemeinen Vorgehensweise. Hier wurde bewusst der Begriff modellorientierte statt modellgetriebener Vorgehensweise gewählt, da die Vollautomatisierung einer Model Driven Architecture (MDA) nicht erzwungen werden muss. Stattdessen steht im Vordergrund, dass vor der eigentlichen Umsetzung eine saubere und vollständige Modellierung stattzufinden hat. Dies fördert zum einen die Analysephase und liefert zum anderen Metadaten und Dokumentationen, die hinsichtlich Wartbarkeit und Erweiterbarkeit unerlässlich sind.

Drei Modellsichten werden dabei berücksichtigt. Fachliche Modelle liefern eine architekturunabhängige Darstellung der Fachlichkeit. Logische Modelle repräsentieren eine plattformunabhängige Darstellung der Architektur. Die plattformspezifische Umsetzung der logischen Modelle findet man als physische Modellsicht:

Fachliche Modelle	**Architekturunabhängige Darstellung der Fachlichkeit:** • Dimension Fact Model (DFM) • Domänenmodell • Kennzahlenbeschreibung
Logische Modelle	**Plattformunabhängige Darstellung der Architektur:** • Logisches Datenmodell der DMS • Logisches Datenmodell der IDS (als Data Vault) • Logisches Datenmodell der HDS (nur als „System of Records“ damit das Mapping von der HDS in die IDS beschrieben werden kann) • Prozessmodell (Mapping) von IDS nach DMS • Prozessmodell (Mapping) von HDS nach IDS
Physische Modelle	**Plattformspezifische Umsetzung der logischen Modelle:** • Erstellung des physischen Datenbankschemas für die DMS • Erstellung des physischen Datenbankschemas für die IDS • Erstellung des physischen Datenbankschemas für die HDS • Erstellung der ETL-Prozesse von HDS in die IDS • Erstellung der ETL-Prozesse von IDS in die DMS

4.1 Arbeitsschritte bei der Modellierung

Folgende Reihenfolge in der Vorgehensweise wird für die Modellierung neuer Anforderungen vorgeschlagen:

1. Anforderungsanalyse und Erweiterung/Anpassung der fachlichen Modelle
    a. Analyse der Anforderungen
    b. Erweiterung/Anpassung des DFM's
    c. Erweiterung/Anpassung des Domänenmodells
    d. Erweiterung/Anpassung der Kennzahlenbeschreibung

2. Erweiterung/Anpassung der logischen Modelle
    a. Erweiterung/Anpassung der logischen Datenmodelle
        i. Erweiterung/Anpassung des DMS-Modells
        ii. Erweiterung/Anpassung des IDS-Modells
        iii. Erweiterung/Anpassung des HDS-Modells
    b. Erweiterung/Anpassung der logischen Prozessmodelle
        i. Erweiterung/Anpassung des Prozessmodells von der IDS in die DMS
        ii. Erweiterung/Anpassung des Prozessmodells von der HDS in die IDS

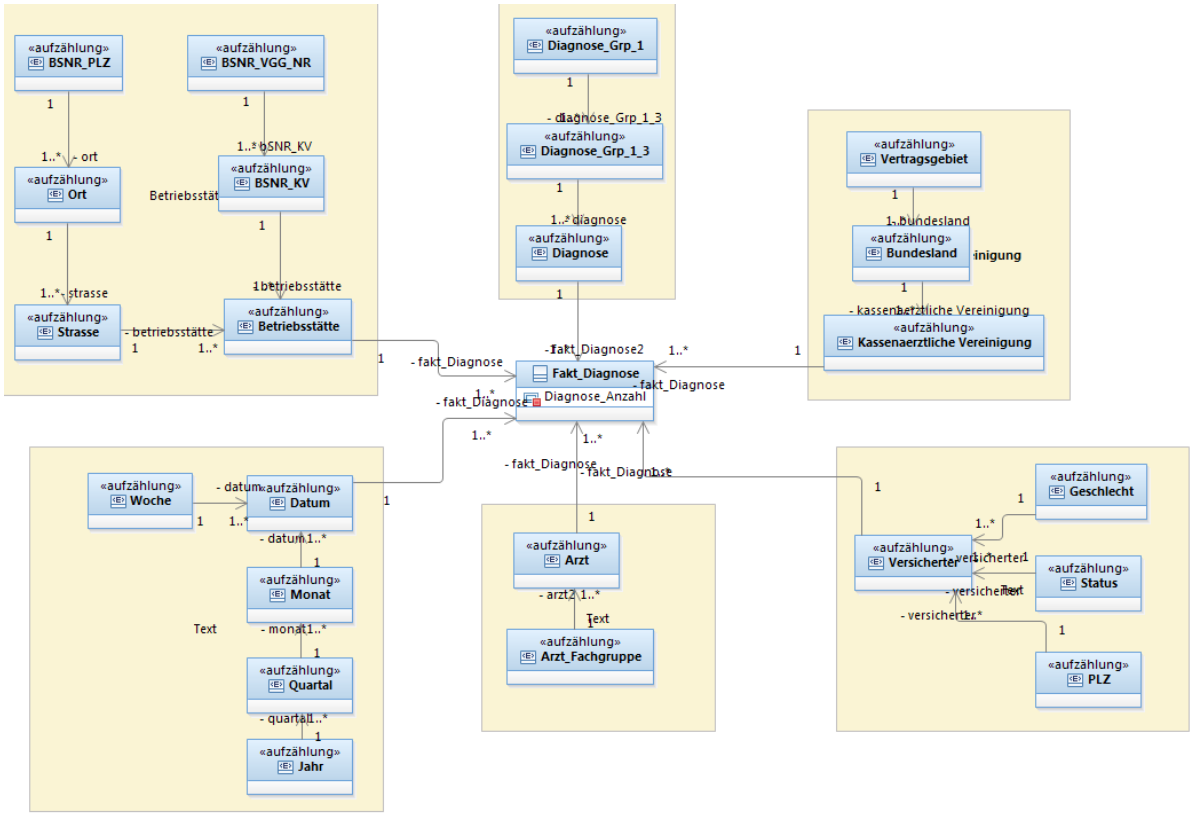
3. Erweiterung/Anpassung der physischen Modelle
    a. Erweiterung/Anpassung der physischen Datenmodelle
        i. Erstellung des physischen Datenbankschemas für die DMS
        ii. Erstellung des physischen Datenbankschemas für die IDS
        iii. Erstellung des physischen Datenbankschemas für die HDS
    b. Erweiterung/Anpassung der physischen Prozessmodelle (ETL-Entwicklung)
        i. Erstellung der ETL-Prozesse von HDS in die IDS
        ii. Erstellung der ETL-Prozesse von IDS in die DMS

4.2 Fachliche Modellschicht

Als fachliche Modelle (konzeptuelle Modelle) werden das Dimension Fact Model (DFM), das Domänenmodell und Kennzahlenbeschreibungen geführt. Darin ist die Fachlichkeit in einer architekturunabhängigen Weise darzustellen.

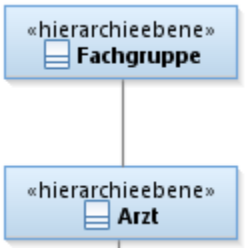
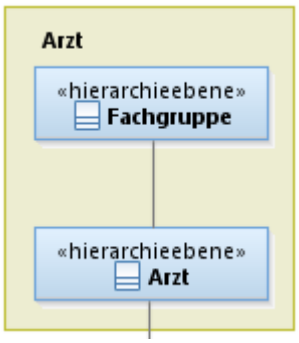
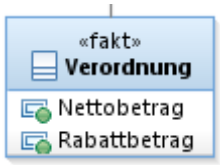
4.2.1 Dimension Fact Model (DFM)

Das Dimensional Fact Model (DFM)[1] stellt die Kennzahlen, deren Dimensionen und deren Auswertungshierarchien dar. Es liefert einen Überblick über die dimensionalen hierarchischen Analysemöglichkeiten:



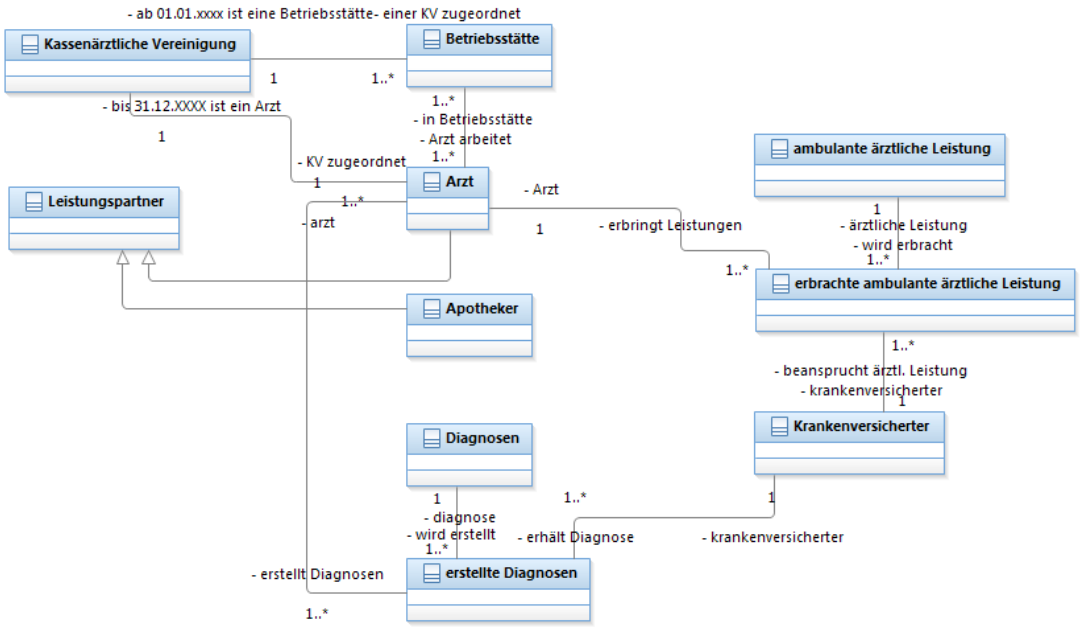
Die Erstellung eines DFMs erfolgt mit einfachen UML-Klassendiagrammen. Folgenden Modellelemente werden dazu verwendet:

[1] siehe zum Beispiel (Golfarelli M., Maio D., Rizzi S., 1998)

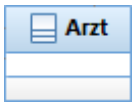
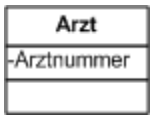
Modellelement	Symbol	Erläuterung
Hierarchieebene (Klasse)		Eine Klasse beschreibt eine Hierarchieebene einer Dimension. Beispiele: Fachgruppe und Arzt stellen Hierarchieebenen der Dimension Arzt dar (Anm.: der Dimensionsname „Arzt“ fällt in diesem Beispiel mit dem Namen einer Hierarchieebene zusammen)
Rollup-Beziehung (Assoziation)		Die Assoziation wird im DFM verwendet, um hierarchische Rollup-Beziehungen darzustellen. Beispiele: Ein Arzt gehört zu einer Fachgruppe.
Dimension (Rechteck)		Für Dimensionen werden keine speziellen Modellelemente verwendet. Hierarchieebenen werden lediglich grafisch mit Rechtecken zusammengefasst. Beispiel: Die Dimension Arzt beinhaltet die Hierarchieebenen Arzt und Fachgruppe.
Kennzahlen, Fakten (Klasse)		Fakten werden mit Klassen beschrieben. Beispiel: Der Fakt Verordnung enthält die Kennzahlen Nettobetrag und Rabattbetrag.

4.2.2 Domänenmodell

Das Domänenmodell und Dimensional Fact Model (DFM) stellen fachliche (konzeptuelle) Modelle dar. Im Domänenmodell werden relevante Geschäftsobjekte und deren Beziehungen dargestellt, um den fachlichen Kontext besser verstehen zu können:



Die Erstellung eines Domänenmodells erfolgt mit einfachen UML-Klassendiagrammen. Folgenden Modellelemente werden dazu verwendet:

Modellelement	Symbol	Erläuterung
Fachliches Objekt (Klasse)		Eine Klasse beschreibt ein fachliches Objekt der Domäne. Beispiele: Leistungserbringer, Arzt, ambulante ärztliche Leistung, erbrachte ambulante ärztliche Leistung
Eigenschaft eines fachlichen Objekts (Attribut)		Wichtige Eigenschaften fachlicher Objekte können mit Attributen spezifiziert werden. Dabei ist zu beachten, dass ein Domänenmodell nur eine grobe Darstellung liefern soll. Eine detaillierte Ausspezifizierung auf Attributsebene soll daher nicht erfolgen. Beispiel: Die Arztnummer ist ein wichtiges Attribut des fachlichen Objektes Arzt.
Beziehung zwischen fachlichen	1_____*	Eine Assoziation stellt eine fachliche Beziehung zwischen fachlichen Objekten dar (soll auch fachlich beschriftet werden). Dabei

Objekten (Assoziation)	0..1_____* *_____*	werden auch Kardinalitäten zwischen den Objekten berücksichtigt. Beispiel: Ein Arzt erbringt mehrere ambulante ärztliche Leistungen.
IS-A-Beziehung zwischen fachlichen Objekten (Generalisierung)		Die Generalisierung bzw. Spezialisierung stellt einen besonderen Beziehungstyp (IS-A-Beziehung) zwischen zwei fachlichen Objekten dar. Beispiel: Ein Arzt ist ein Leistungserbringer.

4.2.3 Kennzahlenbeschreibung

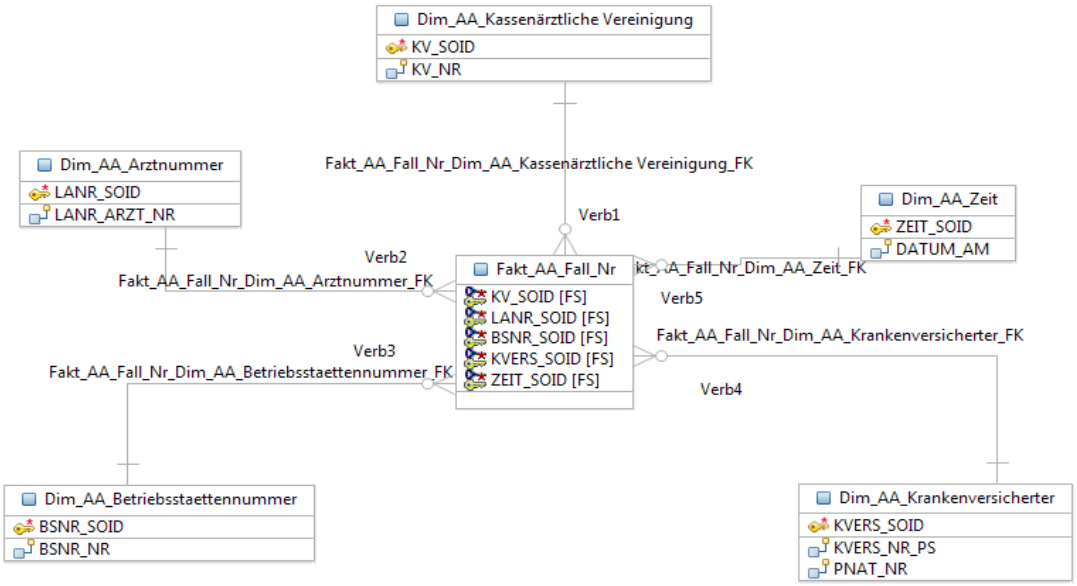
Kennzahlenbeschreibungen und Beschreibungen fachlicher Begriffe werden nicht als Modelle erfasst, sondern fließen in ein Business Glossary ein.

4.3 Logische Modellschicht

In der logischen Modellschicht werden die umzusetzenden Architekturschichten DMS, IDS und HDS dargestellt. Zudem wird die Abbildung zwischen diesen Schichten spezifiziert.

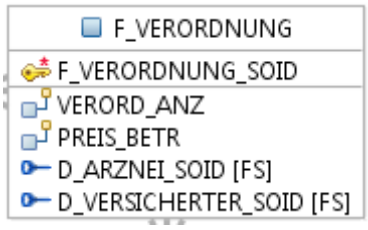
4.3.1 DMS-Modell

In der DMS werden Fakten- und Dimensionstabellen modelliert:[2]



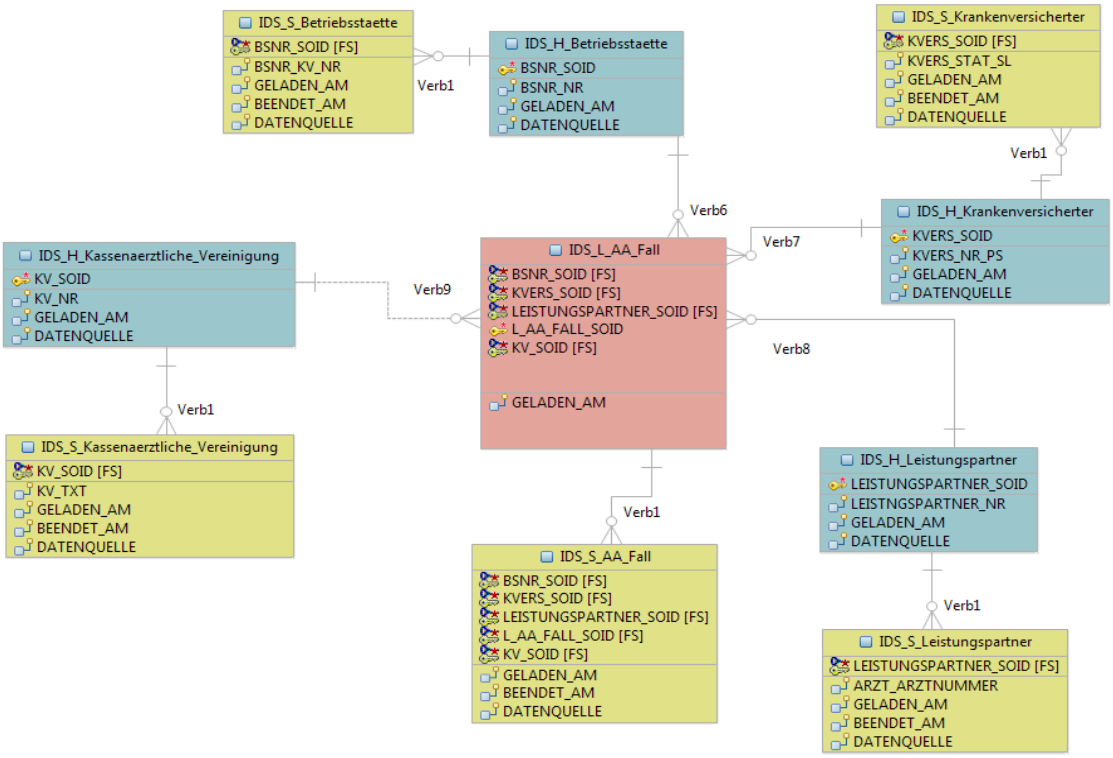
In der Data Mart Schicht (DMS) werden Fakten- und Dimensionstabellen als logische Datenmodelle (Entity-Relationship) dargestellt. Aus diesem Modell wird 1:1 eine DDL generiert, mit der die physischen Datenstrukturen erstellt werden können. Folgende Modellelemente werden verwendet:

[2] siehe zum Beispiel (Kimball R, Ross M., 2013)

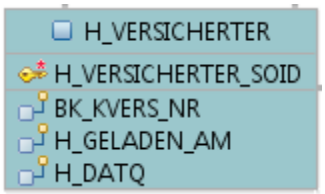
Modellelement	Symbol	Erläuterung
Dimension (Entität)		Eine Dimension wird als Entität (Tabelle) dargestellt. <u>Konventionen:</u> • Der Name einer Dimension beginnt mit dem Präfix D_.
Fakten (Entität)		Fakten werden als Entität (Tabelle) dargestellt, die Kennzahlen enthält und Dimensionen referenziert. <u>Konventionen:</u> Der Name einer Dimension beginnt mit dem Präfix F_.
Beziehung zwischen Dimension und Fakt (Relationship)		Beziehungen zwischen Dimensionen und Fakten werden als Beziehungen dargestellt und ergeben ein klassisches Star-Schema.

4.3.2 IDS-Modell

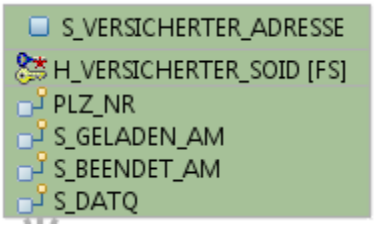
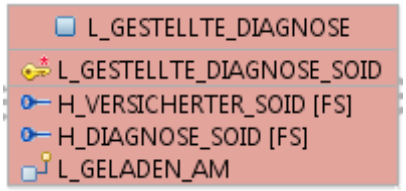
Die IDS-Modellierung erfolgt in Data Vault.[3] Die Modellelemente Hub, Satellit und Link sind farblich speziell gekennzeichnet:



In der Integrierten Datenschicht (IDS) werden alle Daten des unternehmensweiten Data Warehouse gesammelt und historisiert abgelegt. Es werden auch alle Kennzahlen, die in der Data Mart Schicht erforderlich sind, bereits in der IDS errechnet. Die IDS ist ein logisches Datenmodell (Entity-Relationship), welches nach Data Vault geführt wird. Folgende Modellelemente werden verwendet:

Modellelement	Symbol	Erläuterung
Hub (Entität)		Ein Hub ist eine Tabelle, die ein Geschäftsobjekt repräsentiert. Sie enthält nur Business-Keys, die sich nicht verändern. In einem Hub wird ein künstlicher Primärschlüssel definiert. Zusätzlich werden noch administrativen Attribute, die das Ladedatum und die

[3] siehe zum Beispiel (Linstedt D., 2010)

		Datenquelle angeben, definiert.
Satellit (Entität)		Im Satellit werden die eigentlichen Informationen eines Geschäftsobjekts (Hubs) in historisierter Form verwaltet. Pro Hub können mehrere Satelliten definiert werden, die verschiedene Kontexte darstellen. Ein Satellit wird genau von einem Hub referenziert und enthält Attribute zu einem Geschäftsobjekt in historisierter Form. Mit einem Endedatum als zusätzliches Attribut wird die Historisierung gesteuert. Dieses gibt an, ob der Datensatz noch gültig ist oder bereits „beendet“ wurde. Zusätzlich werden noch administrativen Attribute, die das Ladedatum und die Datenquelle angeben, definiert.
Link (Entität)		Beziehungen zwischen Geschäftsobjekten werden als eigene Objekte dargestellt. Ein Link wird von mehreren Hubs oder anderen Links referenziert. Zusätzlich wird noch das Ladedatum als administratives Attribut geführt.
Beziehungen (Relationship)		Beziehungen sind logisch zu sehen. Fachliche Beziehungen werden über Links abgebildet. Logische Beziehungen existieren • zwischen Hubs und Satelliten, • zwischen Links und Satelliten, • zwischen Hubs und Links und • zwischen Links und Links.

4.3.3 HDS-Modell

Das HDS-Modell kann über Reverse Engineering generiert werden und stellt lediglich ein System of Records dar, um das anschließende Mapping spezifizieren zu können. Insofern stehen nur Entitäten ohne Beziehungen zur Verfügung. Die Modellelemente werden aus dem physischen Datenbankschema übernommen.

4.3.4 Mapping von der HDS in die IDS

Die Abbildungsvorschriften, wie aus den Objekten der HDS die IDS-Ergänzungen durchgeführt werden, werden als Prozesse und Mappings von der HDS zur IDS modelliert.

4.3.5 Mapping von der IDS in die DMS

Die Abbildungsvorschriften, wie aus den Objekten der IDS die DMS-Ergänzungen durchgeführt werden, werden als Prozesse und Mappings von der IDS zur DMS modelliert.

4.4 Physische Modellschicht

Die physischen Modelle werden direkt aus den logischen generiert, z.B. als DDL oder ETL-Jobs.

5 Fazit

Diese White Paper zeigt eine konsolidierte und kompakte Darstellung der Inhalte aus (Neuböck T., Raab J., Weißenböck A., 2014) und (Hiebl J., Hörak K., Linner K., Neuböck T., Raab J., Weißenböck A., 2014). Die vier Projektdimensionen (Managementsicht und Projektvorgehensweise, fachliche Sichtweise, Architektursicht und Modellsicht) sind als gleichwertig zu betrachten, wenngleich in diesem White Paper die Architektur- und Modellsicht im Vordergrund stehen.

7 Literatur

Golfarelli M., Maio D., Rizzi S. (1998). The dimensional fact model: A conceptual model for data warehouses. *Int. J. Cooperative Inf. Syst.*, 7(2-3):215–247.

Hiebl J., Hörak K., Linner K., Neuböck T., Raab J., Weißenböck A. (2014). *Agile modellorientierte DWH-Entwicklung mit IBM InfoSphere.* Linz: solvistas GmbH.

Kimball R, Ross M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling (3rd ed.).* Wiley.

Linstedt D. (2010). *Super Charge your Data Warehouse.*

Neuböck T., Raab J., Weißenböck A. (2014). *Eine modellgetriebene Vorgehensweise in der Data Warehouse Entwicklung.* Linz: solvistas GmbH.