

Agile modellorientierte DWH-Entwicklung mit IBM InfoSphere

White Paper der solvistas GmbH

Johannes Hiebl
Katharina Hörak
Konrad Linner
Thomas Neuböck
Jürgen Raab
Andreas Weißenböck

Linz, Juli 2014

1 Einleitung

In (Neuböck T., Raab J., Weißenböck A., 2014) wurde ein Ansatz für eine agile modellorientierten¹ Data-Warehouse-Entwicklung vorgestellt. Es wurden dort vier Dimensionen (Sichtweisen) von Business Intelligence (BI) und Data Warehouse (DWH) Projekten betrachtet: (1) Managementsicht und Projektvorgehensweise, (2) fachliche Sichtweise, (3) Architektursicht und (4) Modellsicht. Als Projektvorgehensweise wurde das Spiral-Modell nach Barry W. Boehm² als Meta-Modell und Scrum als konkrete agile Vorgehensweise vorgeschlagen. Im vorliegenden White Paper wollen wir diesen Ansatz anhand der konkreten DWH-Plattform basierend auf der DWH-Appliance IBM Netezza und der Information-Management-Umgebung IBM InfoSphere demonstrieren. Als Proof-of-Concept (PoC) wurde dieses Vorgehen mit dieser Plattform bei einem Kunden im öffentlichen deutschen Krankenversicherungsbereich durchgeführt.

Abschnitt 2 geht auf die zugrundeliegende DWH-Architektur ein. In Abschnitt 3 werden die Modellschichten vorgestellt. Es werden dabei zum Unterschied zur Darstellung in (Neuböck T., Raab J., Weißenböck A., 2014) neue Begriffe verwendet, die für die potenziellen Benutzer des Kunden verständlicher sind. Die modellorientierte Vorgehensweise, die auf

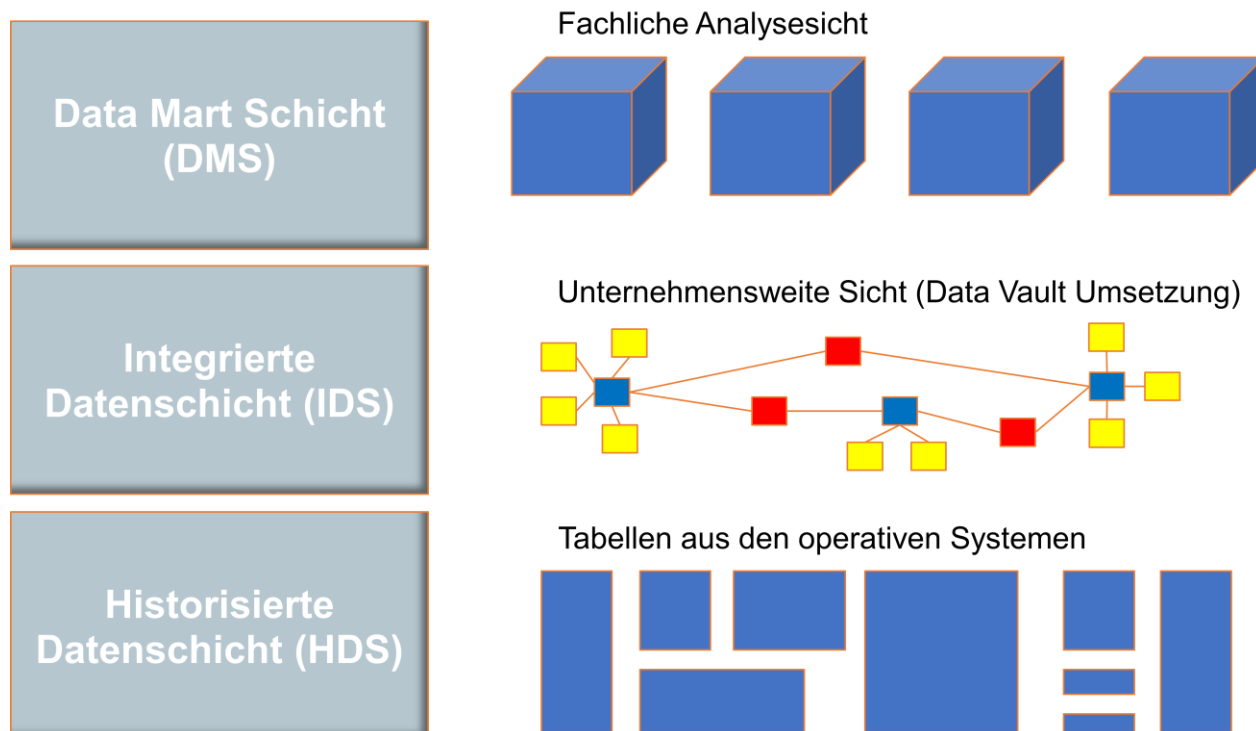
¹ In (Neuböck T., Raab J., Weißenböck A., 2014) wurde der Begriff „modellgetriebene DWH-Entwicklung“ verwendet. Die Erfahrung hat gezeigt, dass dieser Begriff häufig mit einer Model Driven Architecture (MDA) in dem Sinn gleichgestellt wurde, dass eine automatische Modelltransformation und Code-Generierung ausgehend von Modellen erwartet wird. Da diese Erwartungshaltung im DWH-Bereich in dieser vollen Automatisierbarkeit in der Regel nicht erfüllt werden kann und sie ursprünglich in dieser Form auch nicht angedacht wurde, ist der Begriff „modellgetrieben“ durch „modellorientiert“ ersetzt worden.

² siehe beispielsweise (Balzert H., 1998)

Scrum als agile Methode basiert, wird in Abschnitt 4 vorgestellt. Abschnitt 5 beschreibt die zugrundeliegende und für den PoC zur Verfügung stehende Entwicklungsumgebung IBM InfoSphere. Die plattformsspezifische Vorgehensweise auf Basis von IBM InfoSphere wird in Abschnitt 6 vorgestellt. In Abschnitt 7 wird noch auf Modellierungselemente im Detail (im Sinne von Modellierungsrichtlinien) eingegangen. Abschnitt 8 fasst die Erkenntnisse in einem Fazit zusammen.

2 Architektur

Wie bereits in (Neuböck T., Raab J., Weißböck A., 2014) demonstriert wurde, wird eine dreischichtigen Architektur vorausgesetzt, die im folgenden Diagramm dargestellt ist:



Die Data Mart Schicht (DMS) repräsentiert eine fachspezifische Sichtweise und bildet die Zugriffsschicht für Berichte und Analysen. Ihre Daten sind in einem Star-Schema³ (Dimensionen und Fakten) umgesetzt.

In der Integrierte Datenschicht (IDS) wird eine unternehmensweite Sicht der Daten geboten. Dabei soll eine Trennung zwischen Geschäftsobjekt, Kontext und Beziehungen erfolgen. Um diese erreichen wird die IDS in Data Vault⁴ umgesetzt mit den wesentlichen Modellelementen Hub, Satellit (einschließlich Business-Satellit) und Link. Dies erlaubt insbesondere höchste Flexibilität bei Erweiterungen und Änderungen. Zusätzlich werden in der IDS ein abgeleiteter fachlicher Kontext (abgeleitete Kennzahlen, fachspezifische Kennzahlen, Geschäftsregeln) gepflegt. Data Vault harmonisiert zudem mit der agilen Vorgehensweise und einer fachlich fokussierten Entwicklung.

³ siehe beispielsweise (Kimball R, Ross M., 2013)

⁴ siehe beispielsweise (Linstedt D., 2010)

In der Historisierte Datenschicht (HDS) werden die Quelldaten bereitgestellt. Die Befüllung erfolgt über eine Staging Area. Eine hohe Aktualität und eine feinste Granularität der Daten werden gefordert. Die Daten werden pseudonymisiert in Deltas bereitgestellt und technisch geprüft. Die HDS erlaubt eine vollständige Datenrekonstruktion.

Betrachtet man die Architektur des gesamten BI-Systems, dann liegt über der DMS noch eine Präsentationsschicht (Reporting-Schicht), auf die nicht näher eingegangen wird.

3 Modellschichten

Im Proof-of-Concept wurden die folgenden Modelle für die vorgeschlagene Vorgehensweise als sinnvoll ausgewählt:

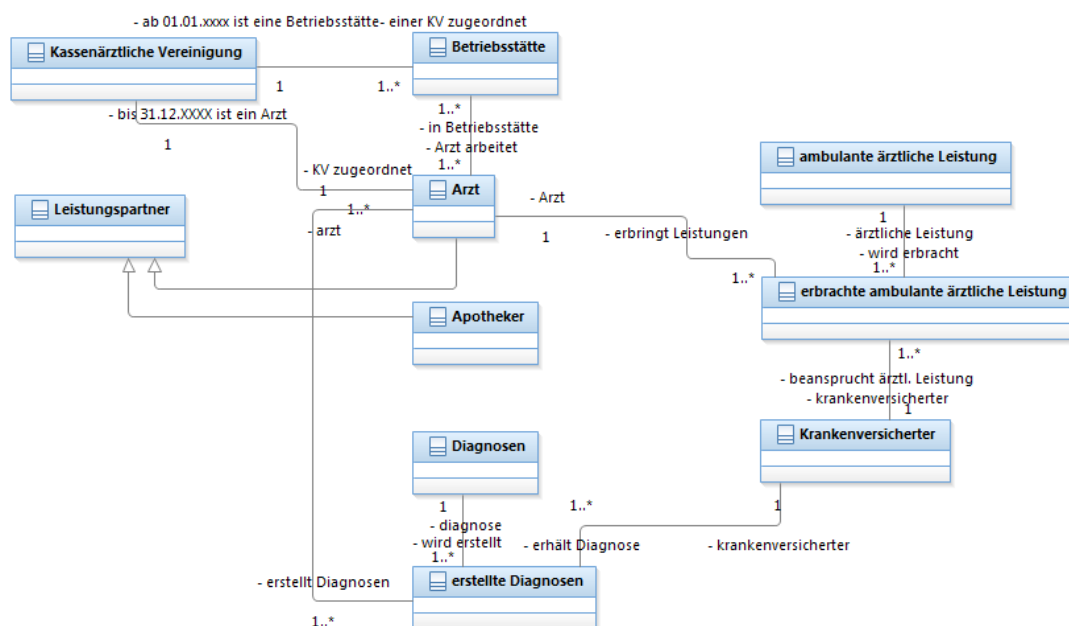
Fachliche Modelle	Architekturunabhängige Darstellung der Fachlichkeit: <ul style="list-style-type: none"> • Domänenmodell • Dimension Fact Model (DFM) • Kennzahlenbeschreibung
Logische Modelle	Plattformunabhängige Darstellung der Architektur: <ul style="list-style-type: none"> • Logisches Datenmodell 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: <ul style="list-style-type: none"> • 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

Wir verwenden an dieser Stelle die neuen Begriffe

- fachliche Modelle (oder auch Konzeptuelle Modelle),
- logische Modelle und
- physische Modelle.

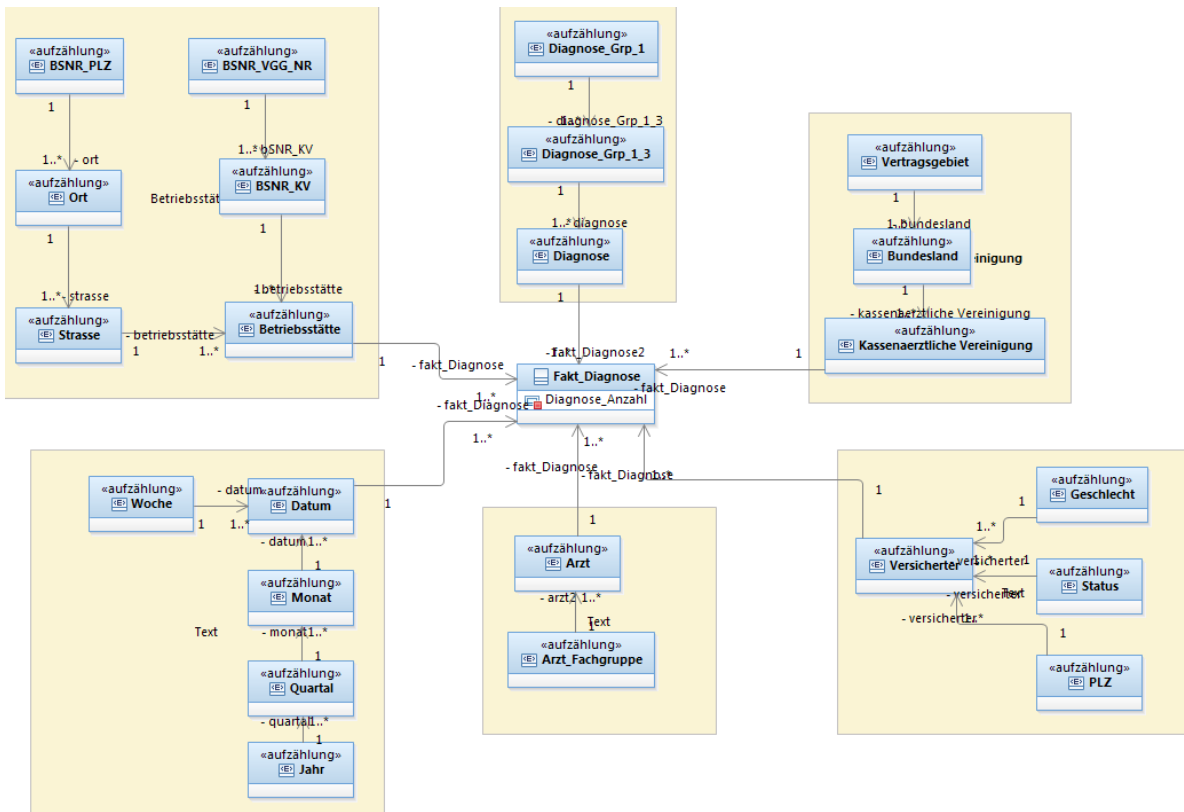
In (Neuböck T., Raab J., Weißböck A., 2014) wurde für „fachliches Modell“ der Begriff „Computation Independent Model (CIM)“, für „logisches Modell“ der Begriff „Platform Independent Model (PIM)“ und für „physisches Modell“ der Begriff „Platform Specific Model (PSM)“ herangezogen. Es hat sich im PoC gezeigt, dass für Benutzer diese neuen Begriffe verständlicher sind. Die folgenden Darstellungen demonstrieren beispielhafte Modelle (Darstellung bereits auf Basis der Werkzeuge von IBM InfoSphere) dieser drei Modellebenen.

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:



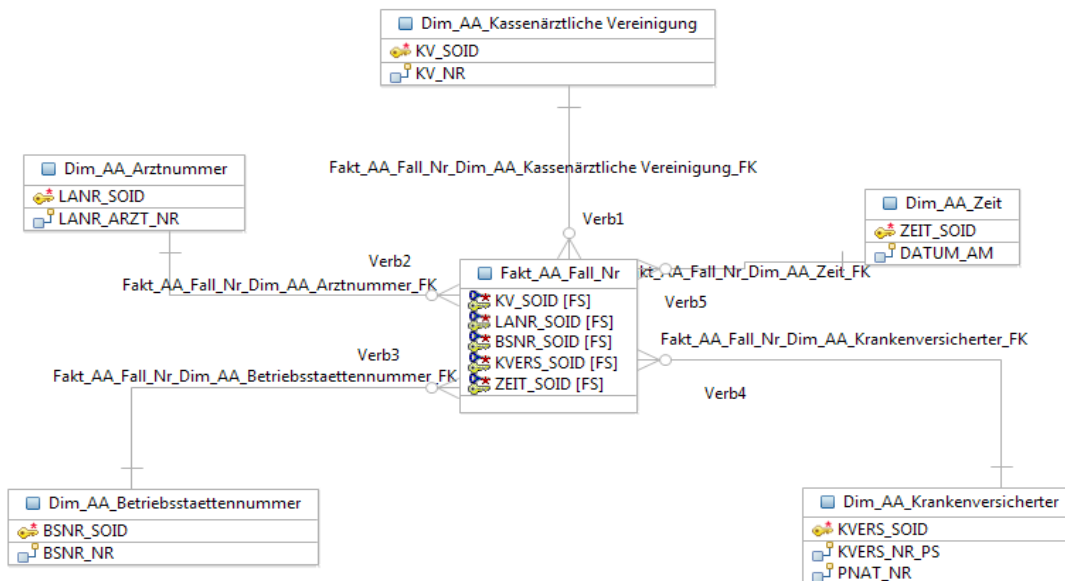
Das Dimensional Fact Model (in Anlehnung an Golfarelli)⁵ stellt die Kennzahlen, deren Dimensionen und deren Auswertungshierarchien dar. Es liefert einen Überblick über die dimensionalen hierarchischen Analysemöglichkeiten:

⁵ siehe beispielsweise (Golfarelli M., Maio D., Rizzi S., 1998)

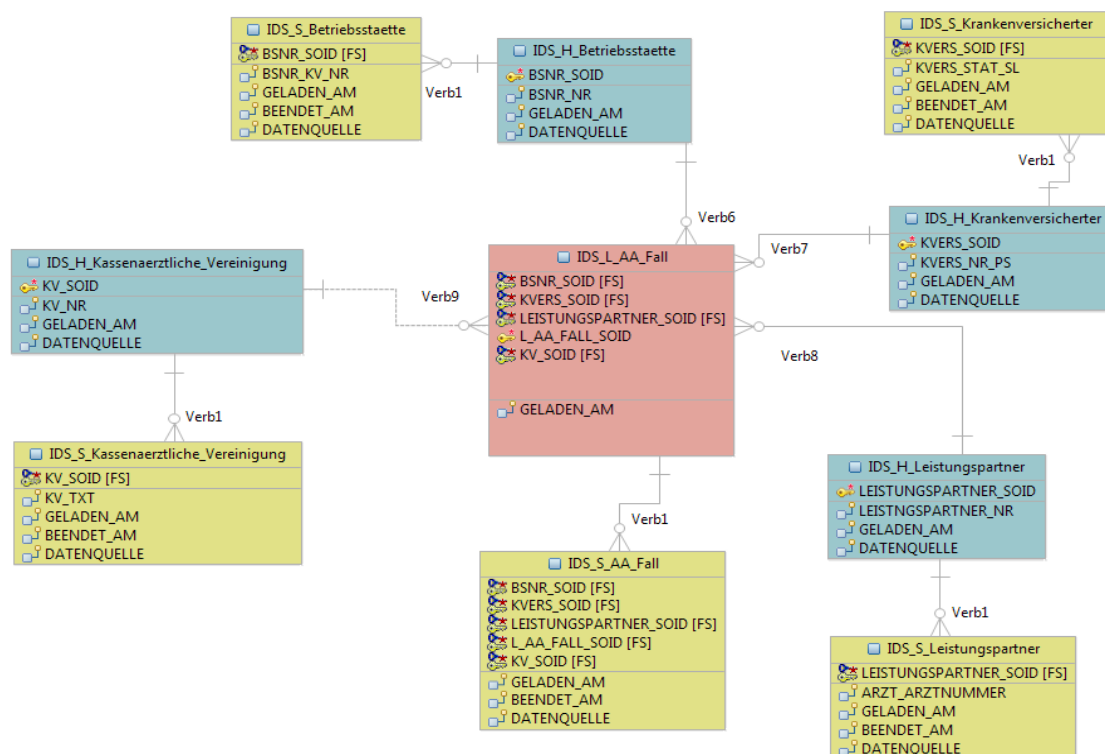


Kennzahlenbeschreibungen und Beschreibungen fachlicher Begriffe werden zwar nicht als Modelle erfasst, können aber mit den eingesetzten IBM Werkzeugen als Business Glossary verwaltet werden.

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



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



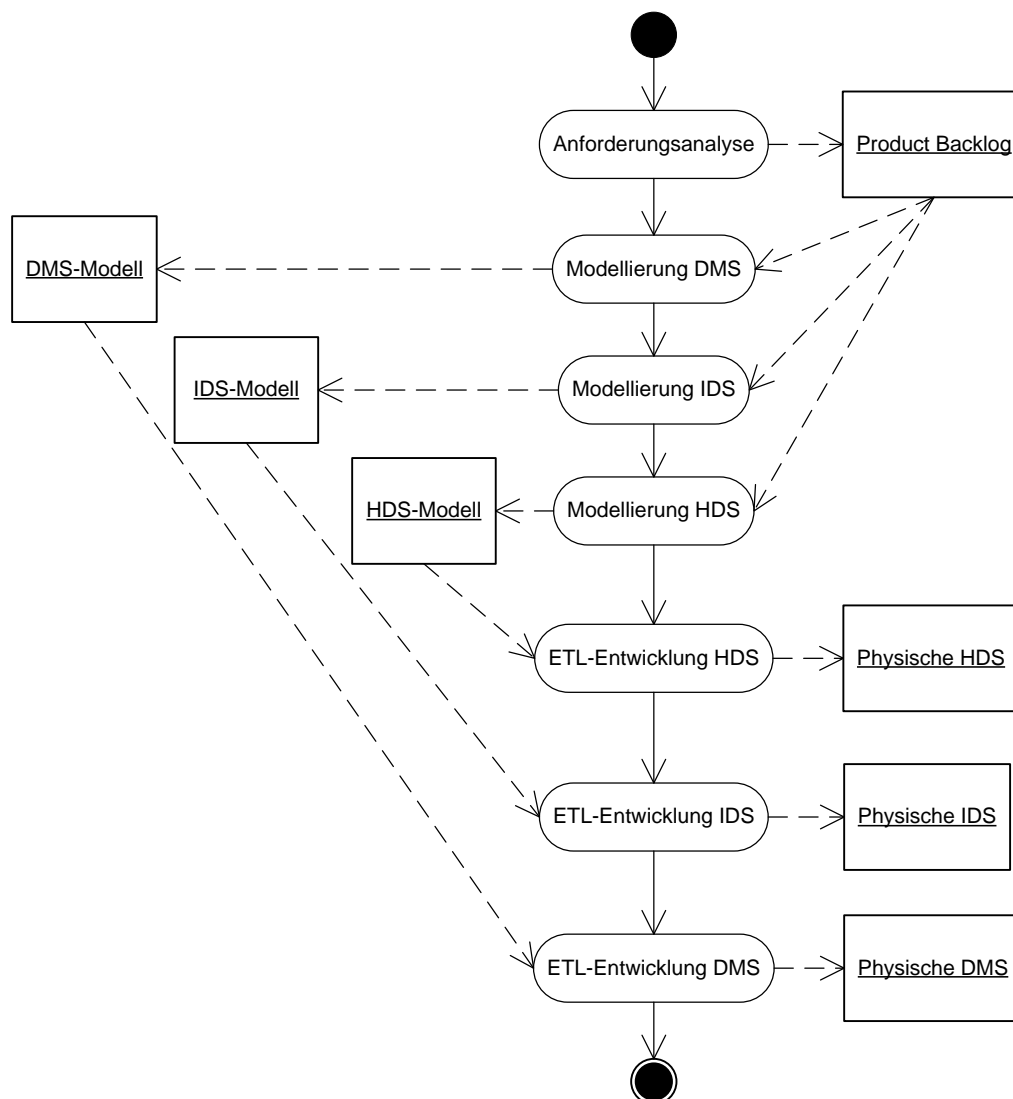
Das HDS-Modell kann über Reverse Engineering generiert werden und stellt lediglich ein System of Records dar, um das anschließende Mapping spezifizieren zu können.

Die Abbildungsvorschriften und –prozesse können mit den IBM-Werkzeugen Fast Track und Blue Print definiert werden. Modellierung und Entwicklung gehen dabei ineinander nahtlos über. Aus den mit Fast Track erstellten Abbildungen werden fertige ETL-Jobs generiert. Mit Blue Print kann die hierarchische Prozess-Struktur grafisch dargestellt werden.

Die physischen Modelle werden direkt aus den logischen generiert, z.B. als DDL oder ETL-Job. Eine weitere detaillierte Beschreibung der physischen Modellschicht wird daher nicht mehr weiter ausgeführt.

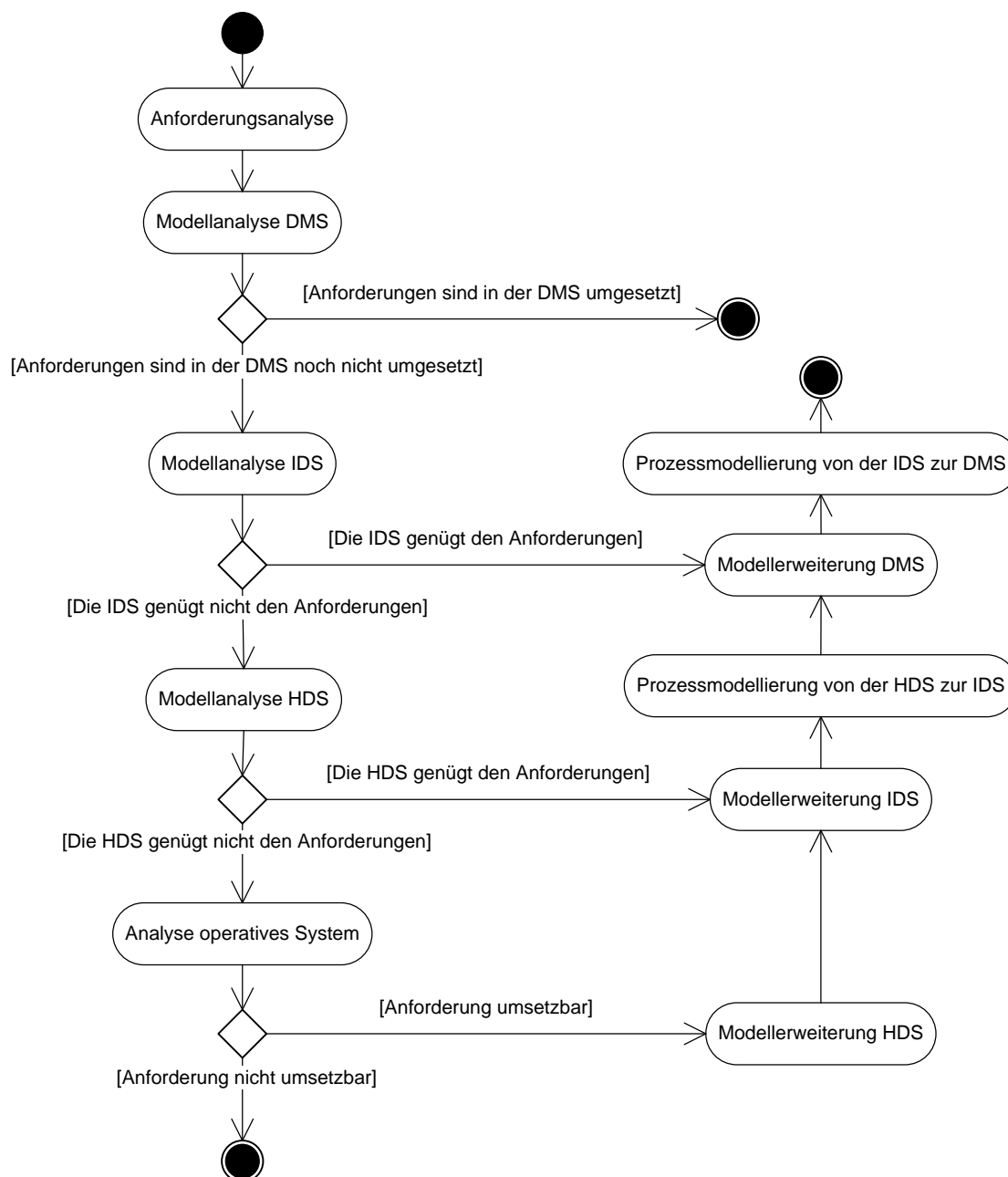
4 Werkzeugunabhängige Vorgehensweise

In der folgenden Abbildung ist der allgemeine (werkzeugunabhängige) Modellierungs- und Entwicklungsprozess der vorgeschlagenen Vorgehensweise dargestellt:



Aus der Anforderungsanalyse heraus werden Aufgaben in den Product Backlog übertragen. In Sprints werden Modellerweiterungen durchgeführt. Dies betrifft sowohl die konzeptuellen Modelle als auch die logischen Modelle der drei Architekturschichten DMS, IDS und HDS. Die Modelle dienen als Vorgabe für die ETL-Entwicklung.

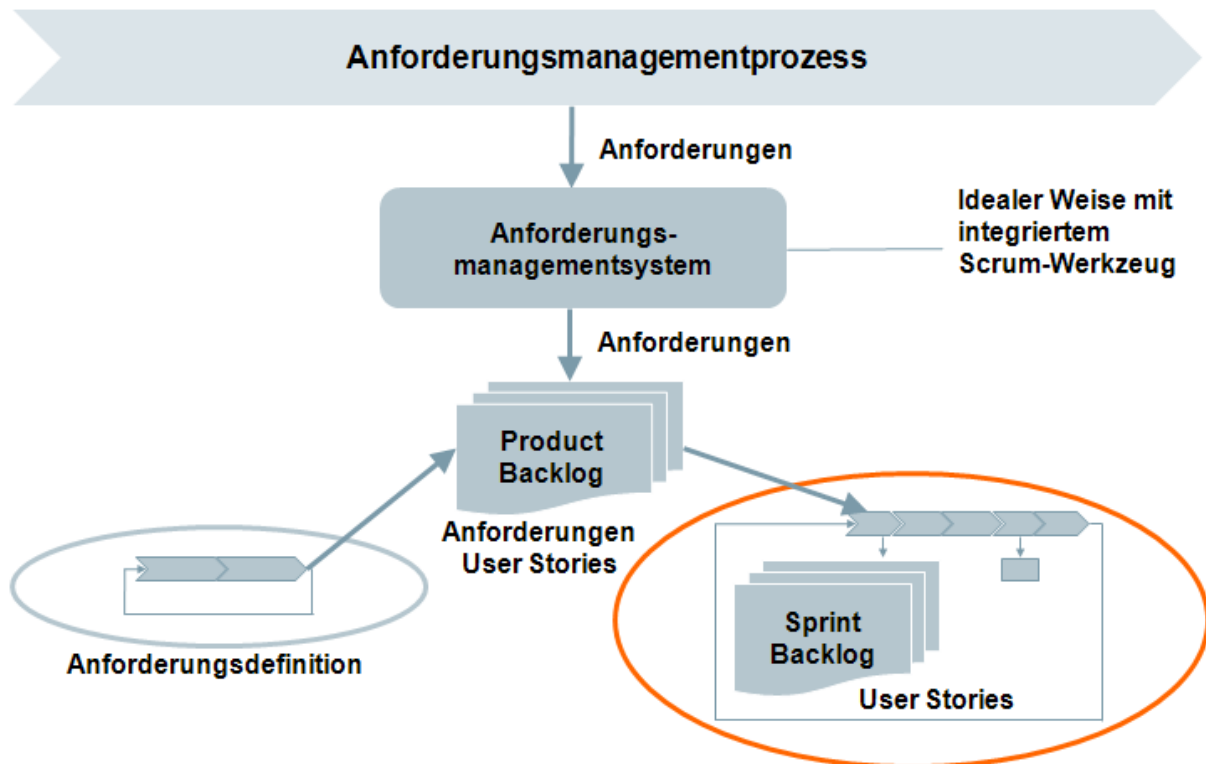
Die folgende Abbildung stellt den Modellierungsprozess in einer bereits weiter verfeinerten Form dar. Insbesondere wird hier operativ zwischen Modellanalyse und Modellerweiterung unterschieden:



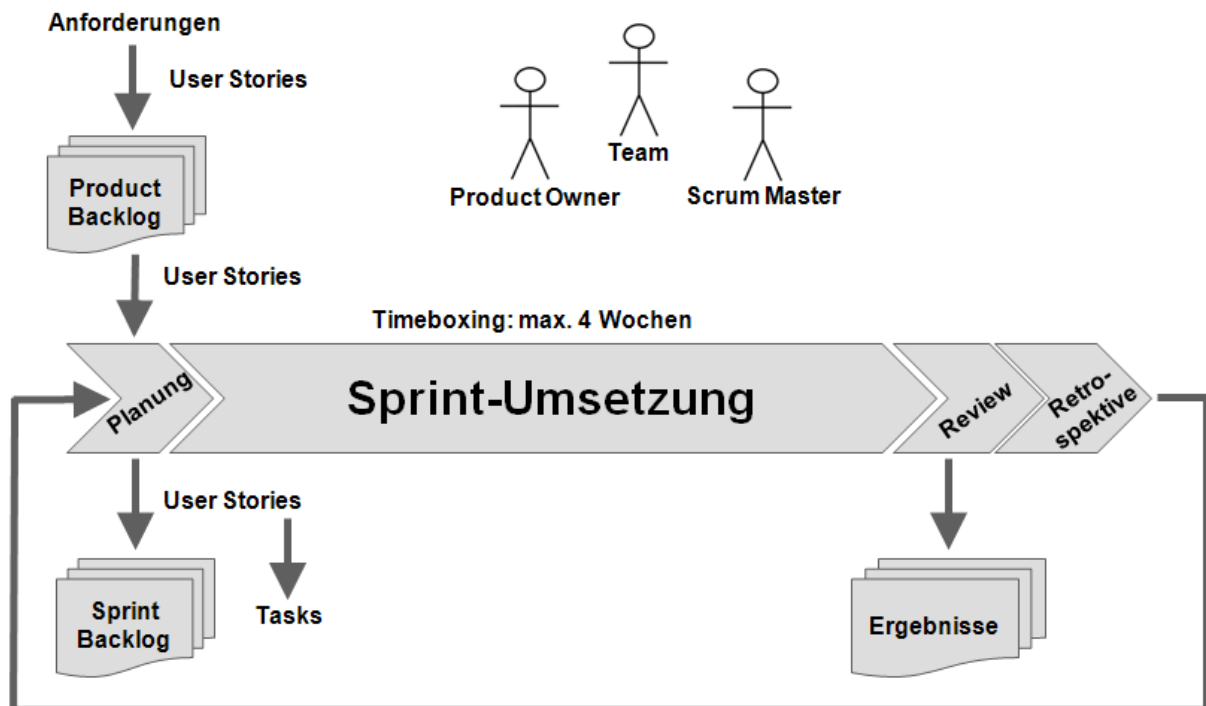
In den folgenden Abschnitten wird der Modellierungs- und Entwicklungsprozess so weit angepasst, sodass die eingesetzten IBM-Werkzeuge sinnvoll eingesetzt werden können.

Der gesamte Prozess ist einer agilen Vorgehensweise (Scrum) eingebettet. Agile Vorgehensweisen haben in den letzten Jahren auch im BI- und DWH-Bereich Einzug gehalten. Die Möglichkeit über kleine Umsetzungszyklen schneller, effizienter und auch qualitativ hochwertiger die Projektziele zu erreichen, eröffnet einen besonderen Reiz gegenüber schwerfälligen rein phasenorientierten Prozessen. Ergebnisse werden schneller sichtbar und können mit dem Kunden diskutiert werden. Bei falschen Produktentwicklungen kann schneller gegengesteuert werden.

Der Scrum-Prozess ist mit einem Anforderungsmanagementprozess verbunden, bei dem idealerweise integrierte Scrum-Werkzeuge (z.B. Agile Jira) eingesetzt werden:



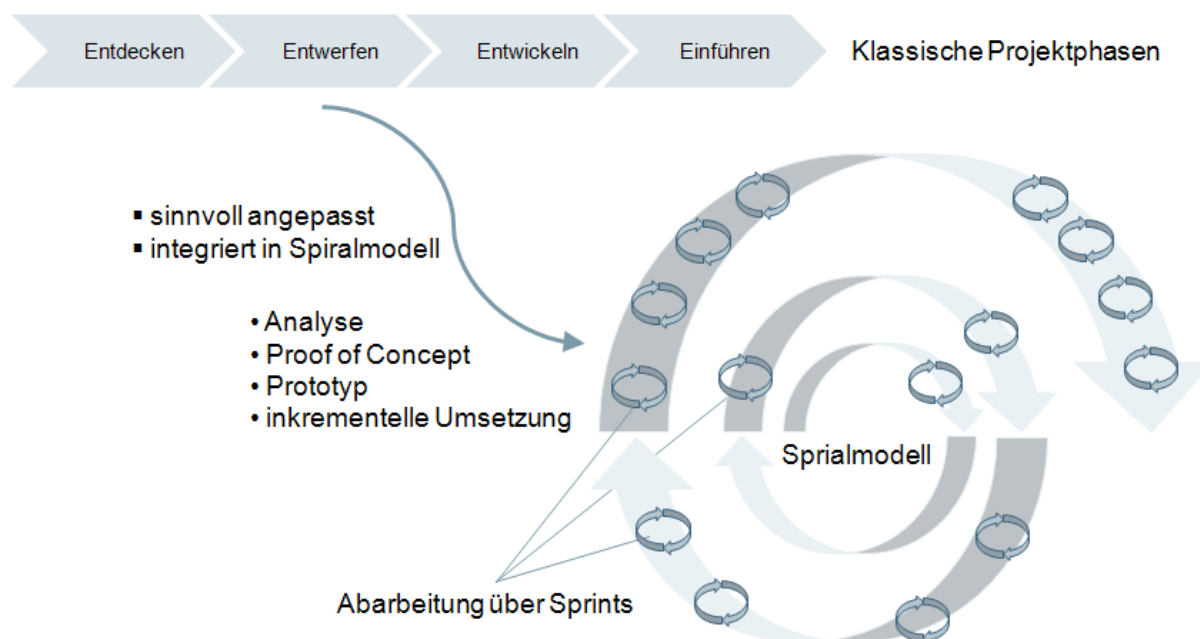
Anforderungen werden vom Product Owner als User Stories formuliert und in den Product Backlog gestellt. Die folgende Abbildung zeigt den Sprint-Prozess und die beteiligten Rollen:



Die im Product Backlog gesammelten User Stories für einen neuen Sprint werden vom Product Owner und dem Team im Planungsmeeting zur Umsetzung ausgewählt und in den

Sprint Backlog gestellt. Am Sprint-Ende erfolgt die Abnahme (oder Nicht-Abnahme) dieser User Stories (im Sprint Review). Die Sprint Retrospektive dient dazu, Prozesse auf Grund der vergangenen Erfahrungen zu verbessern.

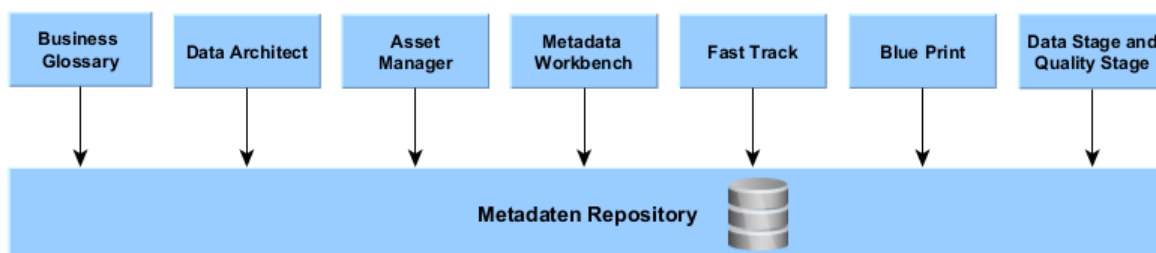
Eine generelle Anforderung des vorgestellten Vorgehensmodells stellt die Komplexitätsbewältigung dar. Beim Spiralmodell handelt es sich um ein Metamodell, das eine inkrementelle prototyporientierte Entwicklung in mehreren Zyklen erlaubt. Auch im DWH- und BI-Bereich erscheint diese Vorgehensweise in vielen Situationen angebracht.



Falls erforderlich wird in einem Zyklus eine prototyporientierte Umsetzung gestartet (z.B. als Proof of Concept). Nach einem abgeschlossenen Zyklus wird entschieden, wie mit der Umsetzung fortgefahren werden soll. Zu Beginn werden nur kleine Zyklen durchlaufen (geringe Kosten). Je mehr Zyklen abgeschlossen sind, umso höher ist auch die Planungssicherheit und umso umfangreicher können die nächsten Zyklen vollzogen werden. Einzelne Zyklen können auch als separate Projektphasen definiert werden.

5 Information Server

Die folgende Abbildung gibt eine Übersicht über die wichtigsten eingesetzten IBM InfoSphere Werkzeuge:



5.1 IBM InfoSphere Information Server 9.1.2.0

Der Information Server ist eine Web-Browser-Applikation und wird in der für den PoC installierten Entwicklungsumgebung über einen entsprechenden Link aufgerufen:



Der Information Server stellt eine Suite mit den folgenden Komponenten zur Verfügung:

- Administration Console
- Metadata Asset Manager
- Business Glossary
- Metadata Workbench
- Standardization Rule Designer
- Operations Console

Im Folgenden werden nur jene Komponenten kurz beschrieben, die für die Modellierung und Entwicklung in erster Linie relevant sind.

Darüber hinaus werden noch Client-Applikationen dargestellt, die nicht als Web-Anwendung aufrufbar sind:

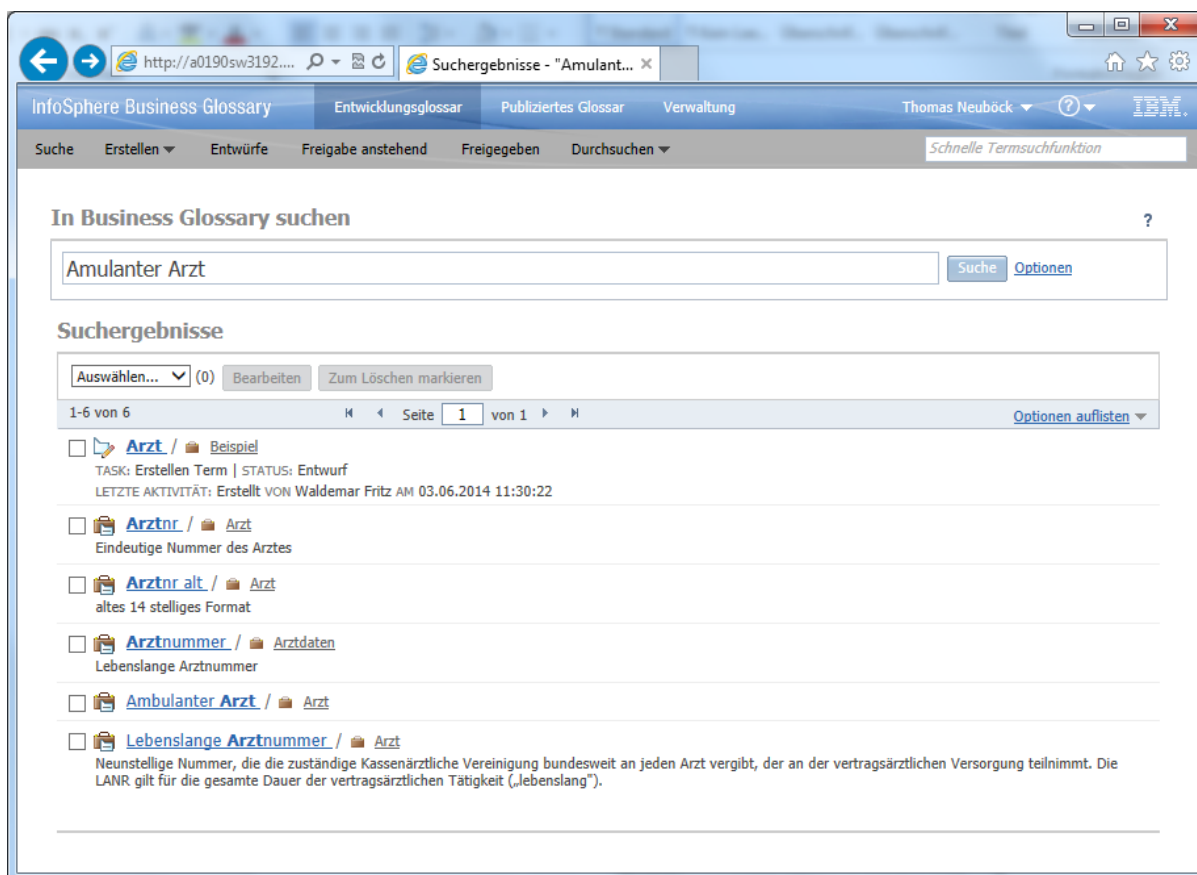
- DataArchitect
- FastTrack
- DataStage and QualityStage

5.2 IBM InfoSphere Business Glossary

IBM InfoSphere Business Glossary ermöglicht ein webbasiertes Management von geschäftlichen Begriffen, Definitionen und Kategorien, sodass ein einheitliches Vokabular für technische und fachliche Benutzer geschaffen werden kann. Sie können den Geschäfts- und den IT-Bereich für Initiativen wie Stammdatenverwaltung, Data Warehousing, Business Intelligence und Anwendungskonsolidierungsprojekte besser abstimmen:

- Management von Businessbegriffen und Kategorien
 - Erstellung und Verwaltung von einem Vokabular und Klassifizierungssystem mit Hilfe von Begriffen, die in der Branche Standard sind.
- Zusammenarbeit
 - in einer Umgebung, in der Benutzer im gesamten Unternehmen Informationen gemeinsam entwickeln und weiterentwickeln können
- Eine Webanwendung zur Untersuchung von Kategorien und Begriffen, die das Unternehmen beschreiben.
- Import-, Exportfunktionalität, um bestehende Glossare einzubinden
- Lösung, um anderen Anwendungen und Benutzern schnelleren Zugriff auf Glossareinträge zu bieten

Beispiel (Suche im Business Glossary):



The screenshot shows a web browser window displaying the IBM InfoSphere Business Glossary search results for the term "Amulanter Arzt".

InfoSphere Business Glossary | Entwicklungsglossar | Publiziertes Glossar | Verwaltung | Thomas Neuböck

Suche | Erstellen | Entwürfe | Freigabe anstehend | Freigegeben | Durchsuchen | *Schnelle Termsuchfunktion*

In Business Glossary suchen

Amulanter Arzt [Suche] [Optionen]

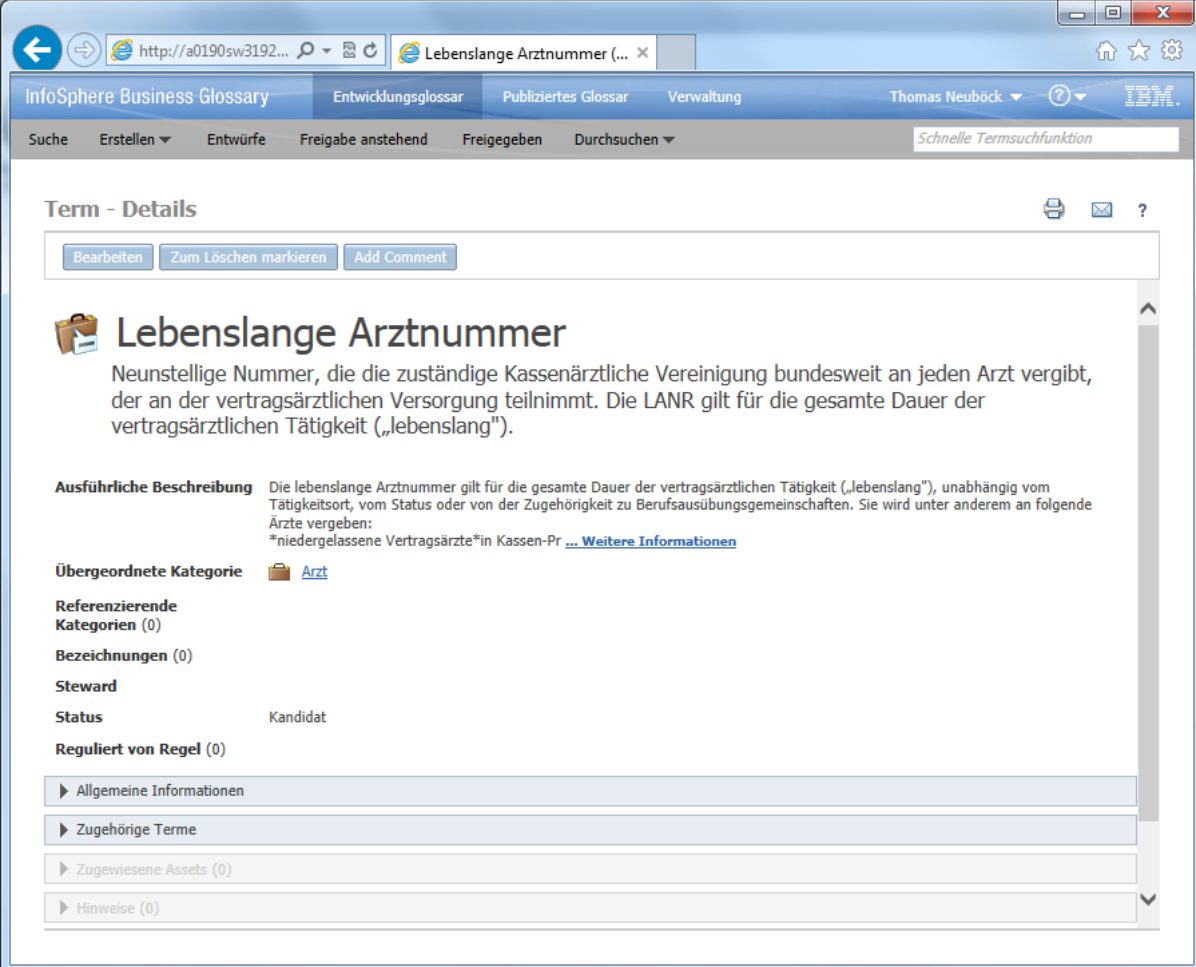
Suchergebnisse

Auswählen... (0) [Bearbeiten] [Zum Löschen markieren]

1-6 von 6 | Seite 1 von 1 | [Optionen auflisten]

- [Arzt](#) / [Beispiel](#)
TASK: Erstellen Term | STATUS: Entwurf
LETZTE AKTIVITÄT: Erstellt VON Waldemar Fritz AM 03.06.2014 11:30:22
- [Arztnr.](#) / [Arzt](#)
Eindeutige Nummer des Arztes
- [Arztnr.alt](#) / [Arzt](#)
altes 14 stelliges Format
- [Arztnummer](#) / [Arztdaten](#)
Lebenslange Arztnummer
- [Ambulanter Arzt](#) / [Arzt](#)
- [Lebenslange Arztnummer](#) / [Arzt](#)
Neunstellige Nummer, die die zuständige Kassenärztliche Vereinigung bundesweit an jeden Arzt vergibt, der an der vertragsärztlichen Versorgung teilnimmt. Die LANR gilt für die gesamte Dauer der vertragsärztlichen Tätigkeit („lebenslang“).

Beispiel (Beschreibung im Business Glossary):



The screenshot shows a web browser window displaying the IBM InfoSphere Business Glossary interface. The browser address bar shows the URL <http://a0190sw3192...> and the page title is "Lebenslange Arztnummer (... x)". The application header includes "InfoSphere Business Glossary" and navigation tabs for "Entwicklungsglossar", "Publiziertes Glossar", and "Verwaltung". A search bar with the text "Schnelle Termsuchfunktion" is visible. The main content area is titled "Term - Details" and features a "Lebenslange Arztnummer" term with a brief description: "Neunstellige Nummer, die die zuständige Kassenärztliche Vereinigung bundesweit an jeden Arzt vergibt, der an der vertragsärztlichen Versorgung teilnimmt. Die LANR gilt für die gesamte Dauer der vertragsärztlichen Tätigkeit („lebenslang“)." Below this, there is an "Ausführliche Beschreibung" section, an "Übergeordnete Kategorie" of "Arzt", and several expandable sections for "Allgemeine Informationen", "Zugehörige Terme", "Zugewiesene Assets (0)", and "Hinweise (0)".


Term - Details

Bearbeiten Zum Löschen markieren Add Comment

Lebenslange Arztnummer

Neunstellige Nummer, die die zuständige Kassenärztliche Vereinigung bundesweit an jeden Arzt vergibt, der an der vertragsärztlichen Versorgung teilnimmt. Die LANR gilt für die gesamte Dauer der vertragsärztlichen Tätigkeit („lebenslang“).

Ausführliche Beschreibung Die lebenslange Arztnummer gilt für die gesamte Dauer der vertragsärztlichen Tätigkeit („lebenslang“), unabhängig vom Tätigkeitsort, vom Status oder von der Zugehörigkeit zu Berufsausübungsgemeinschaften. Sie wird unter anderem an folgende Ärzte vergeben:
*niedergelassene Vertragsärzte*in Kassen-Pr ... [Weitere Informationen](#)

Übergeordnete Kategorie  [Arzt](#)

Referenzierende Kategorien (0)

Bezeichnungen (0)

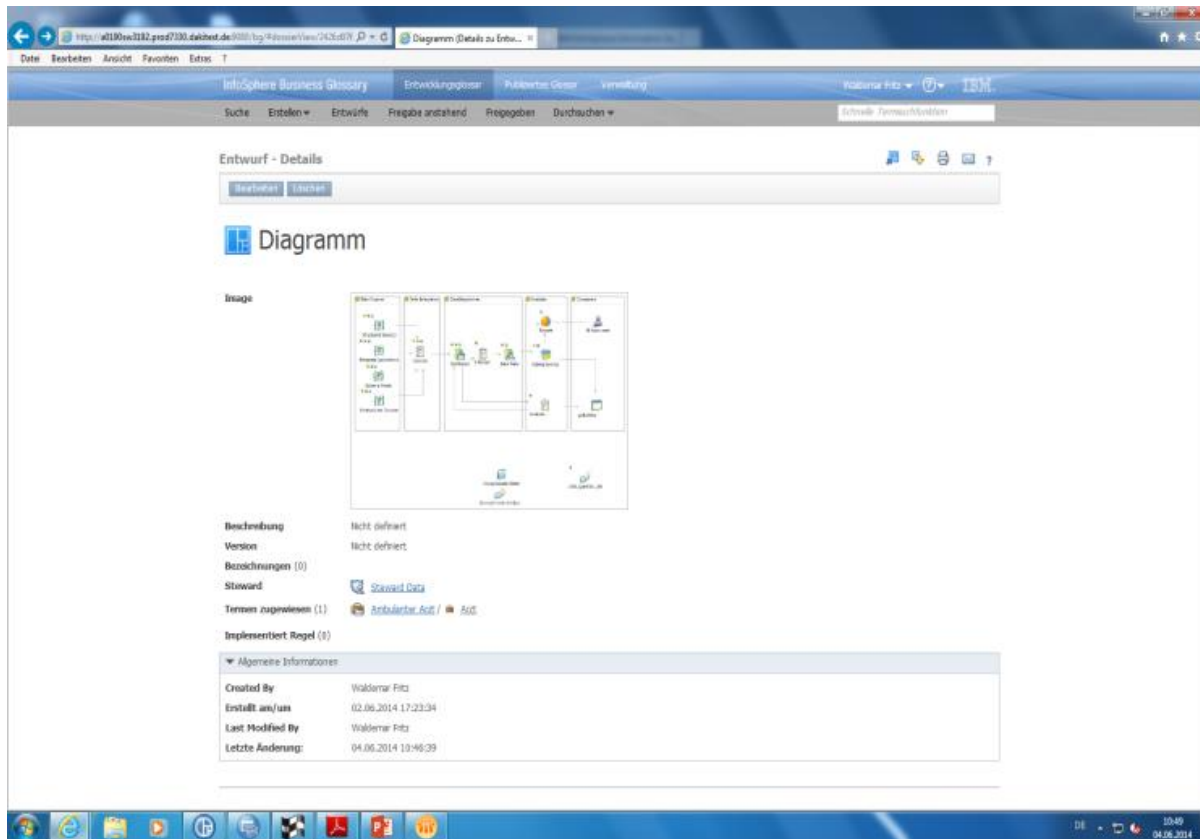
Steward

Status Kandidat

Reguliert von Regel (0)

- ▶ Allgemeine Informationen
- ▶ Zugehörige Terme
- ▶ Zugewiesene Assets (0)
- ▶ Hinweise (0)

Beispiel (Beschreibung im Business Glossary inkl. Diagrammdarstellungen):



The screenshot shows the IBM InfoSphere Business Glossary interface. The main content area displays a diagram titled "Diagramm" with a sub-label "Image". Below the diagram, there is a "Beschreibung" (Description) section with the following details:

Beschreibung	Nicht definiert
Version	Nicht definiert
Bezeichnungen (0)	
Steward	Steward Data
Termin zugewiesen (1)	Ankuler Act / Act
Implementiert Regel (1)	

Below this, there is a section for "Allgemeine Informationen" (General Information):

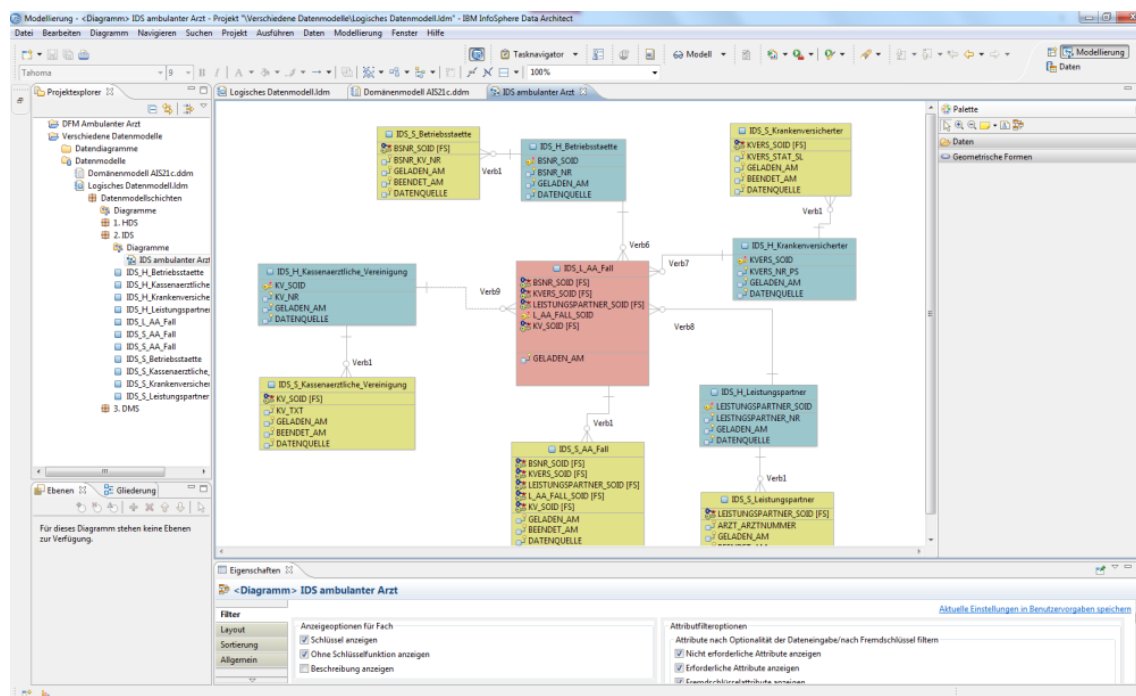
Created By	Waldemar Fitz
Erstellt am/um	02.06.2014 17:23:34
Last Modified By	Waldemar Fitz
Letzte Änderung	04.06.2014 10:46:39

5.3 IBM InfoSphere Data Architect

Mit dem Data Architect wird der gemeinschaftlichen Datenentwurf erstellt (Datenmodellierung), mit dem unternehmensweit vielfältige und verteilte Datenassets erkannt, modelliert, in Beziehung gesetzt, standardisiert und integriert werden können. Data Architect kann ein besseres Datenverständnis bieten und so die Effizienz steigern:

- Datenmodellierungswerkzeug
- Größere Effizienz durch Kenntnis der Datenassets und effektiveres Arbeiten mit Datenmodellen.
- effektives Arbeiten mit Datenmodellen
- Vereinfachte Data-Warehouse-Gestaltung, Änderungsmanagementaufgaben, Dimensionsmodellierung und

Beispiel (IDS-Modellierung im Data Architect):



Anmerkung:

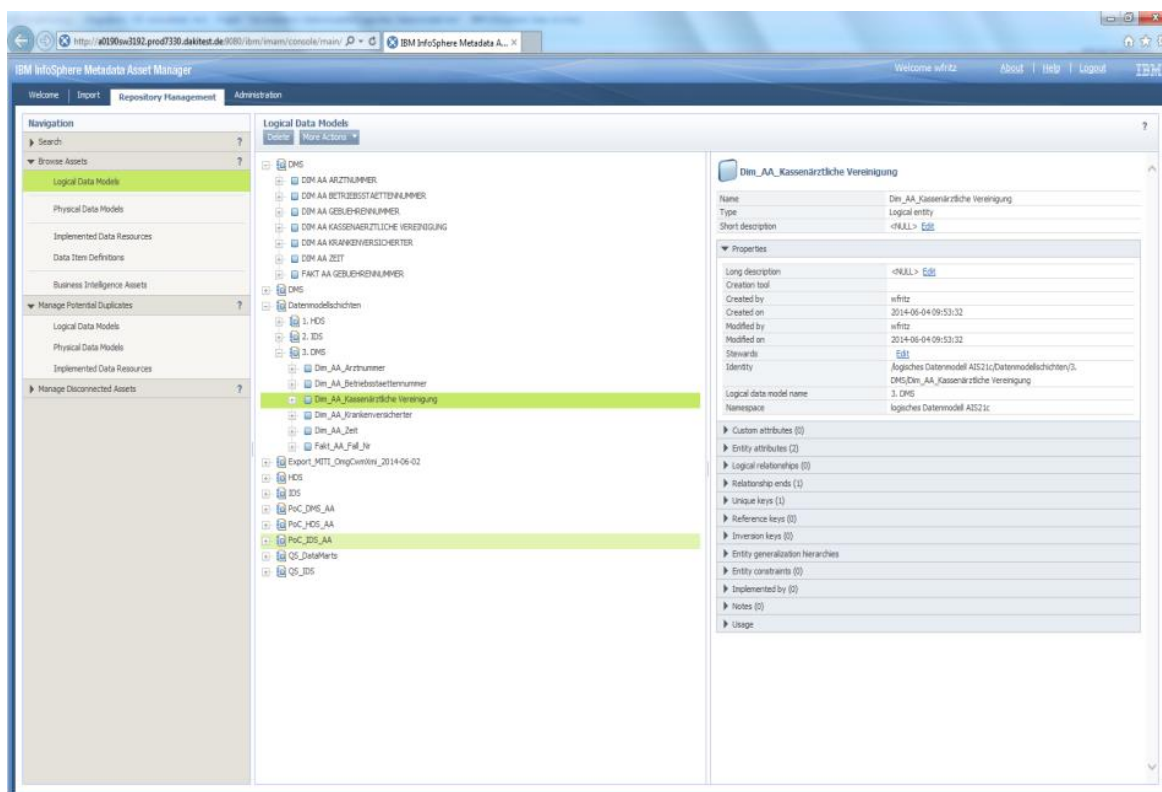
Ursprünglich wurde in diesem kundespezifischen Projekt die Erstellung von Modellen das Modellierungswerkzeug Innovator der Firma MID angestrebt. Innovator erlaubt eine integrierte Modellierung sowohl der Datenschichten als auch der Mapping-Prozesse (in BPMN). Außerdem erlaubt Innovator einen echten Mehrbenutzerbetrieb, d.h. Modelle können sehr feingranular von mehreren Benutzern bearbeitet werden. Diese Modelle werden in einem Repository verwaltet. Der DataArchitect erlaubt ausschließlich die Erstellung von Objekt- und Datenmodellen. Prozessmodell können über andere Werkzeuge (Blueprint Director, FastTrack) nur sehr eingeschränkt dargestellt werden. Die Mehrbenutzerfähigkeit, wie sie der Innovator bietet, fehlt.

5.4 IBM InfoSphere Metadata Asset Manager

Der Metadata Asset Manager erfüllt folgende Funktionen:

- Import- und Export der Metadaten in das Metadaten-Repository
- Analyse der importierten Daten, Anschauen und Verteilen der Metadaten zum Metadaten-Repository
- Der Metadaten Asset Manager wird dort eingesetzt, wo die IBM InfoSphere Werkzeuge nicht direkt in das Metadaten Repository schreiben.

Beispiel (Metadata Asset Manager):



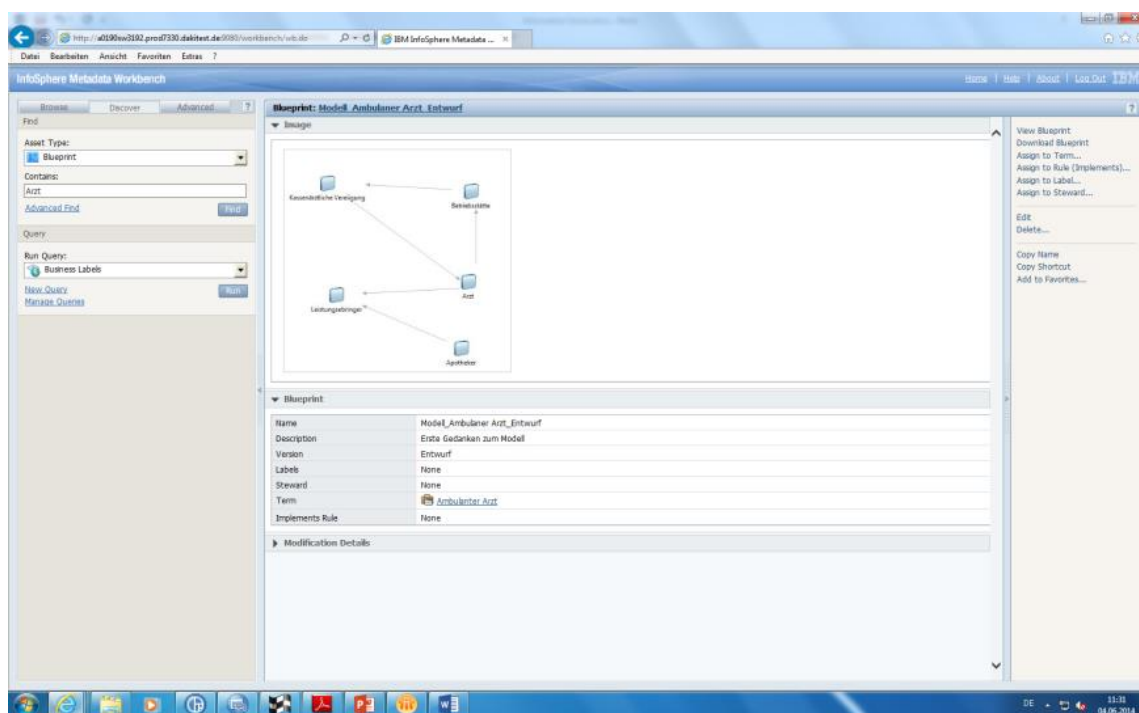
5.5 IBM InfoSphere Metadata Workbench

Die Metadata Workbench ermöglicht eine Datenabstammungsanalyse, mit der sich Datenelemente zu ihren Quellen zurückverfolgen lassen. Mit dieser Software können geschäftliche, technische und operative Metadaten durchsucht und untersucht werden, um sich ein Bild der Herkunft, Konvertierungen und der Nutzung der Daten zu machen.

InfoSphere Metadata Workbench unterstützt in folgender Hinsicht:

- Informationsverfolgung und -analyse – Ermittlung der Datenherkunft und des Wegs der Daten durch den Datenintegrationsprozess
- Änderungsmanagement – Bestimmung der Auswirkungen von Änderungen
- Zusammenarbeit – Entwicklung einer Umgebung, in der die Benutzer Informationen gemeinsam nutzen und verfolgen können

Beispiel (Metadata Workbench):

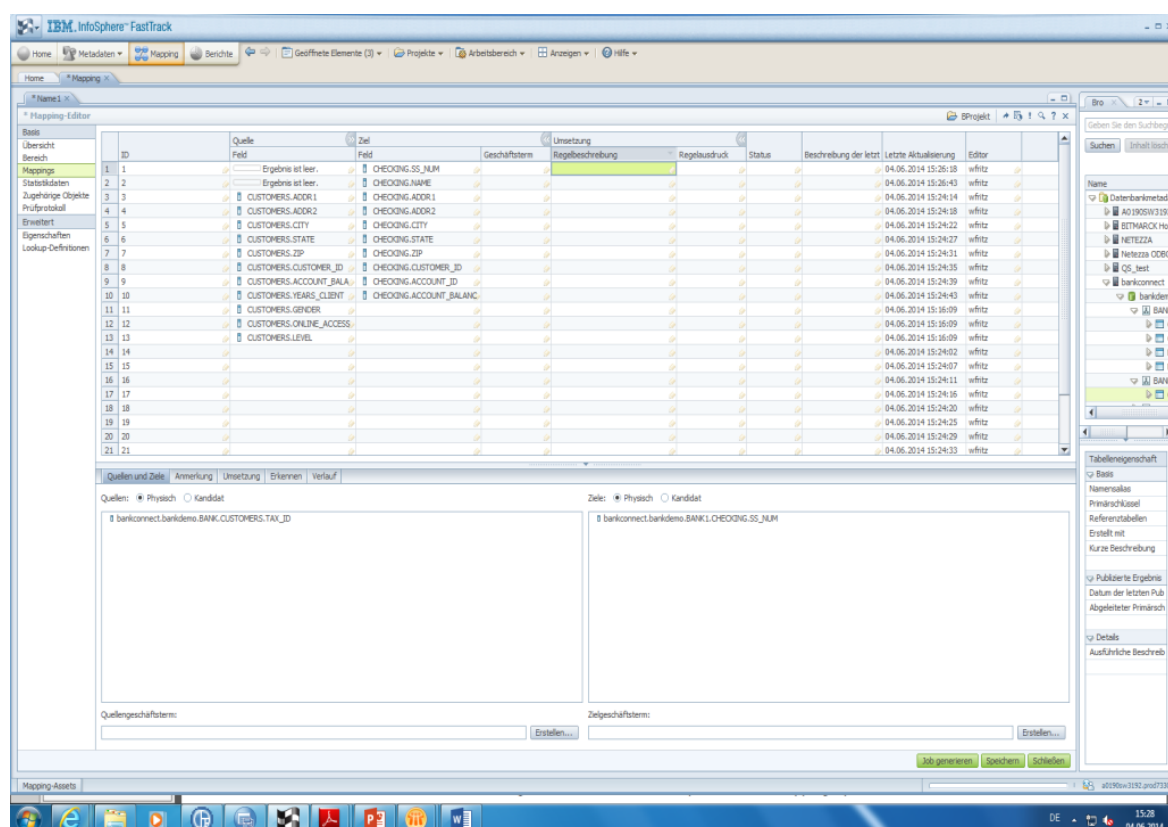


5.6 IBM InfoSphere Information Server FastTrack Client

Fast Track bietet folgende Funktionalitäten und Eigenschaften:

- Technische Definition der Datenmappings
- Ähnlich wie Excel
- Generierung von DataStage-Jobs bzw. Hüllen
- Erstellt und verbindet Begriffe des Business Glossary und ihre Beziehung zu den entsprechenden physischen Spalten im Rahmen des Zuordnungs- und Dokumentationsprozesses.

Beispiel (FastTrack):



ID	Quelle	Ziel	Unsetzung
1	Ergebnis ist leer.	CHEQING_SS_NUM	
2	Ergebnis ist leer.	CHEQING_NAME	
3	CUSTOMERS_ADDR1	CHEQING_ADDR1	
4	CUSTOMERS_ADDR2	CHEQING_ADDR2	
5	CUSTOMERS_CITY	CHEQING_CITY	
6	CUSTOMERS_STATE	CHEQING_STATE	
7	CUSTOMERS_ZIP	CHEQING_ZIP	
8	CUSTOMERS_CUSTOMER_ID	CHEQING_CUSTOMER_ID	
9	CUSTOMERS_ACCOUNT_BALA	CHEQING_ACCOUNT_ID	
10	CUSTOMERS_YEARS_CLIENT	CHEQING_ACCOUNT_BALANC	
11	CUSTOMERS_GENDER		
12	CUSTOMERS_ONLINE_ACCESS		
13	CUSTOMERS_LEVEL		
14			
15			
16			
17			
18			
19			
20			
21			

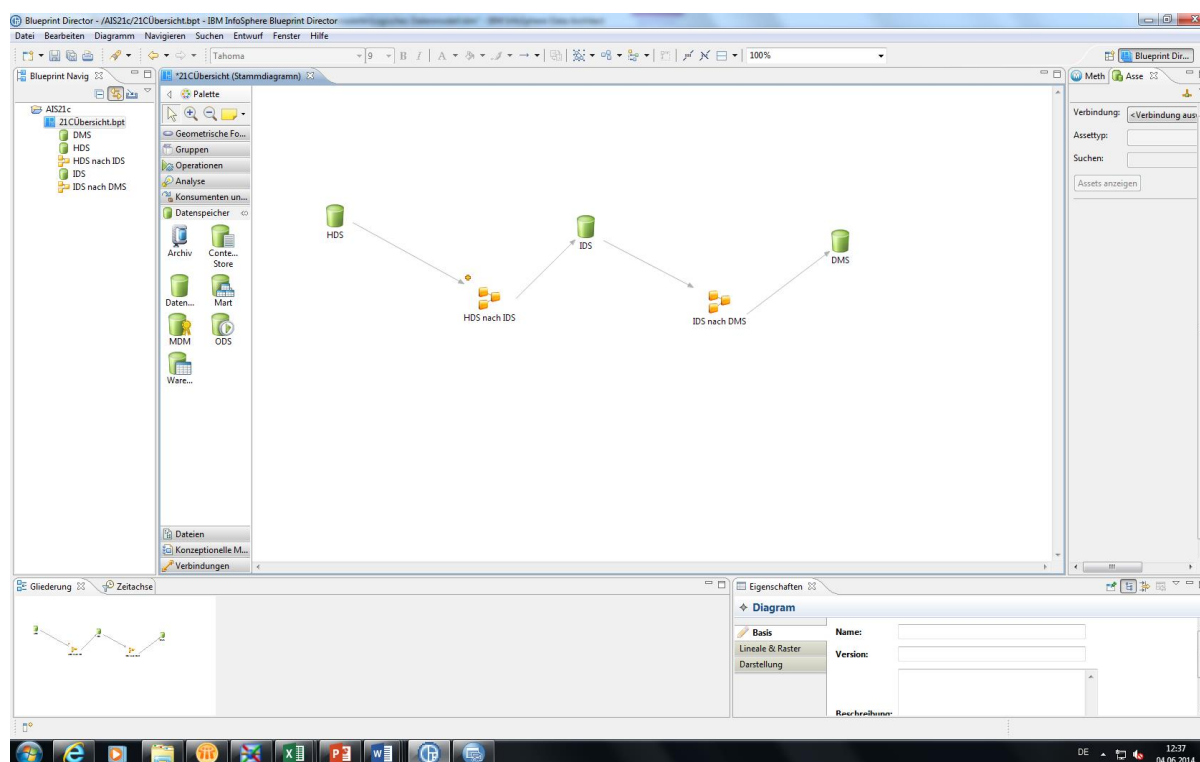
5.7 IBM InfoSphere Blueprint Director

Mit Blueprint Director werden Prozesse, angefangen von den ersten Skizzen bis hin zur Bereitstellung definiert und dargestellt.

Teammitglieder sind dadurch in der Lage, das Projekt während seiner Entstehung nachzuvollziehen. Das Erstellen eines gut dokumentierten und vollständigen Entwurfs sowohl von der Geschäftsvision als auch von der technischen Sicht unterstützt bei der Abstimmung der Geschäftsanforderungen mit der technischen Architektur:

- Erfassung und Abbildung von Prozessen
- Nutzung des gemeinschaftlichen Wissens
- Grafische Aufbereitung von Prozessen

Beispiel (BluePrint):



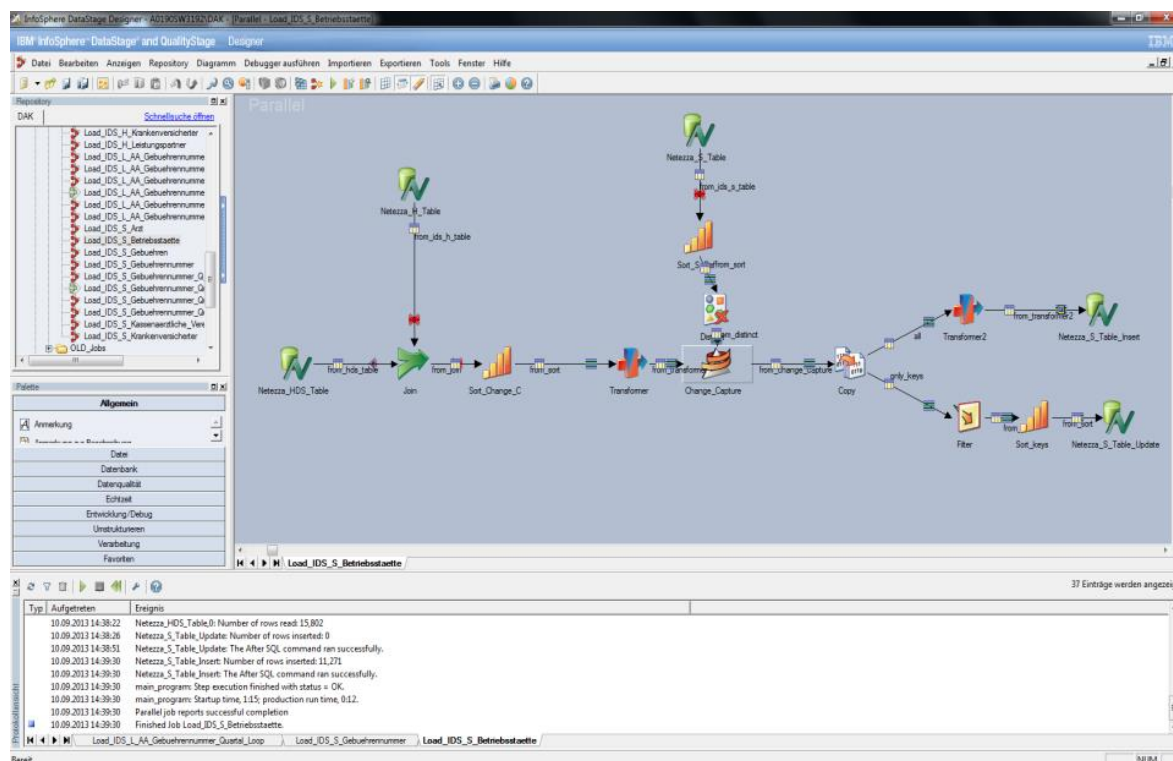
5.8 IBM InfoSphere Information Server DataStage and QualityStage Designer

DataStage integriert Daten aus mehreren Systemen und unterstützt das Management erweiterter Metadaten sowie unternehmensspezifische Konnektivität. Die skalierbare Plattform ermöglicht eine flexiblere Integration aller Datentypen auf verteilten Plattformen und Mainframeplattformen.

DataStage bietet folgende Funktionen und Vorteile:

- Leistungsfähige, skalierbare ETL-Plattform, um die Erfassung, Integration und Transformation großer Datenmengen mit Datenstrukturen, die von einfach bis komplex reichen, zu unterstützen.
- Unterstützung von Big Data und Hadoop, um den direkten Zugriff auf Big Data in einem verteilten Dateisystem zu ermöglichen.
- Echtzeitnahe Datenintegration sowie Konnektivität zwischen Datenquellen und Anwendungen.
- Workload- und Geschäftsregelmanagement, um Sie dabei zu unterstützen, die Hardwareauslastung zu optimieren und geschäftskritische Aufgaben zu priorisieren.
- Benutzerfreundlichkeit, um die Geschwindigkeit, Flexibilität und Effektivität bei Erstellung, Implementierung, Aktualisierung und Verwaltung der Infrastruktur für die Datenintegration zu verbessern

Beispiel (DataStage):



Data Profiling mit IBM Information Server: Die Bewertung der Datenqualität der Quellsysteme und Schnittstellen führt zu einer besseren Risiko- und Aufwand-Abschätzung. Die Veränderung der Datenqualität kann mit dem Modul Information Analyzer über den Vergleich von ‚Baselines‘ über die Zeit gemessen werden.

Beispiele zum Data Profiling:

The screenshot displays the IBM InfoSphere Information Server interface for Data Profiling. The main window shows the 'View Analysis Status' section with a 'Data Sources' table. Below this, a 'General Format' section includes a bar chart and a table of format statistics.

Data Sources							Primary Key	Defined
Name	Analysis	Data Class	Properties	Domain	Format	Key and Cross-Domain	Defined	
▼ All Hosts								
▼ A0190SW3192							--	
▼ SAS_AADM							--	
▼ aadm							--	
TDA000_KV.csv							No	
TDA070_DIAG.csv							No	

Total Rows	Data Class	Cardinality	Minimum Value	Maximum Value
154615	Code	27	0.0175%	?

General Format	Count	Percent	Status
A99.9	79606	51,48659574	Conform
A99.99	49885	32,26401061	Conform
A99.-	14227	9,20156518	Conform
A99.9-	8196	5,30090871	Conform
A99	2700	1,74627300	Conform

6 Vorgehensweise mit Information Server

Die folgenden Tabellen zeigen, mit welchen Werkzeugen aus IBM InfoSphere die einzelnen Modelle umgesetzt bzw. weiterverarbeitet werden. Sie sind nach der fachlichen, logischen und physischen Modellebene getrennt.

Umsetzung fachlicher Modelle mit IBM InfoSphere:

Fachliche Modelle	Domänenmodell	<ul style="list-style-type: none"> • Business Glossary (Metadata Repository) • DataArchitect (Übernahme ins Metadata Repository: Metadata Asset Manager, Metadata Workbench)
	Dimension Fact Model (DFM)	<ul style="list-style-type: none"> • Business Glossary (Metadata Repository) • DataArchitect (Übernahme ins Metadata Repository: Metadata Asset Manager, Metadata Workbench)
	Kennzahlenbeschreibung	<ul style="list-style-type: none"> • Business Glossary (Metadata Repository)

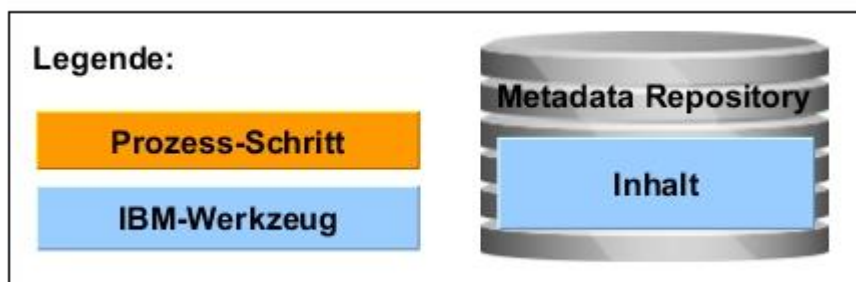
Umsetzung logischer Modelle mit IBM InfoSphere:

Logische Modelle	Logisches Datenmodell der DMS	<ul style="list-style-type: none"> • Business Glossary (Metadata Repository) • DataArchitect (Übernahme ins Metadata Repository: Metadata Asset Manager, Metadata Workbench)
	Logisches Datenmodell der IDS (als Data Vault)	<ul style="list-style-type: none"> • Business Glossary (Metadata Repository) • DataArchitect (Übernahme ins Metadata Repository: Metadata Asset Manager, Metadata Workbench)
	Logisches Datenmodell der HDS („System of Records“)	<ul style="list-style-type: none"> • Business Glossary (Metadata Repository) • DataArchitect (Übernahme ins Metadata Repository: Metadata Asset Manager, Metadata Workbench)
	Prozessmodell (Mapping) von IDS nach DMS	<ul style="list-style-type: none"> • Business Glossary (Metadata Repository) • BluePrint Director (Übernahme ins Metadata Repository: Metadata Asset Manager, Metadata Workbench) • FastTrack (Metadata Repository)
	Prozessmodell (Mapping) von HDS nach IDS	<ul style="list-style-type: none"> • Business Glossary (Metadata Repository) • BluePrint Director (Übernahme ins Metadata Repository: Metadata Asset Manager, Metadata Workbench) • FastTrack (Metadata Repository)

Umsetzung physischer Modelle mit IBM InfoSphere:

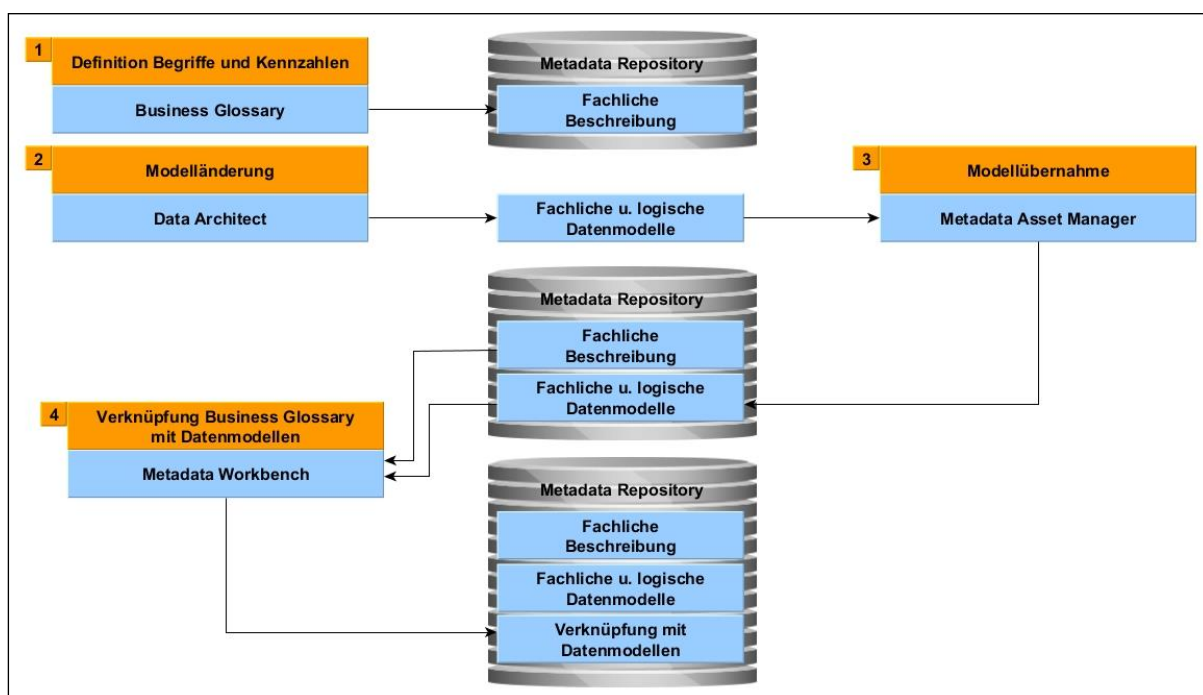
Physische Modelle	Physisches Datenbankschemas für die DMS	<ul style="list-style-type: none"> • Business Glossary (Metadata Repository) • DataArchitect (Übernahme ins Metadata Repository: Metadata Asset Manager, Metadata Workbench)
	Physisches Datenbankschemas für die IDS	<ul style="list-style-type: none"> • Business Glossary (Metadata Repository) • DataArchitect (Übernahme ins Metadata Repository: Metadata Asset Manager, Metadata Workbench)
	Physisches Datenbankschemas für die HDS	<ul style="list-style-type: none"> • Business Glossary (Metadata Repository) • DataArchitect (Übernahme ins Metadata Repository: Metadata Asset Manager, Metadata Workbench)
	ETL-Prozesse von HDS in die IDS	<ul style="list-style-type: none"> • Business Glossary (Metadata Repository) • DataStage (Metadata Repository)
	ETL-Prozesse von IDS in die DMS	<ul style="list-style-type: none"> • Business Glossary (Metadata Repository) • DataStage (Metadata Repository)

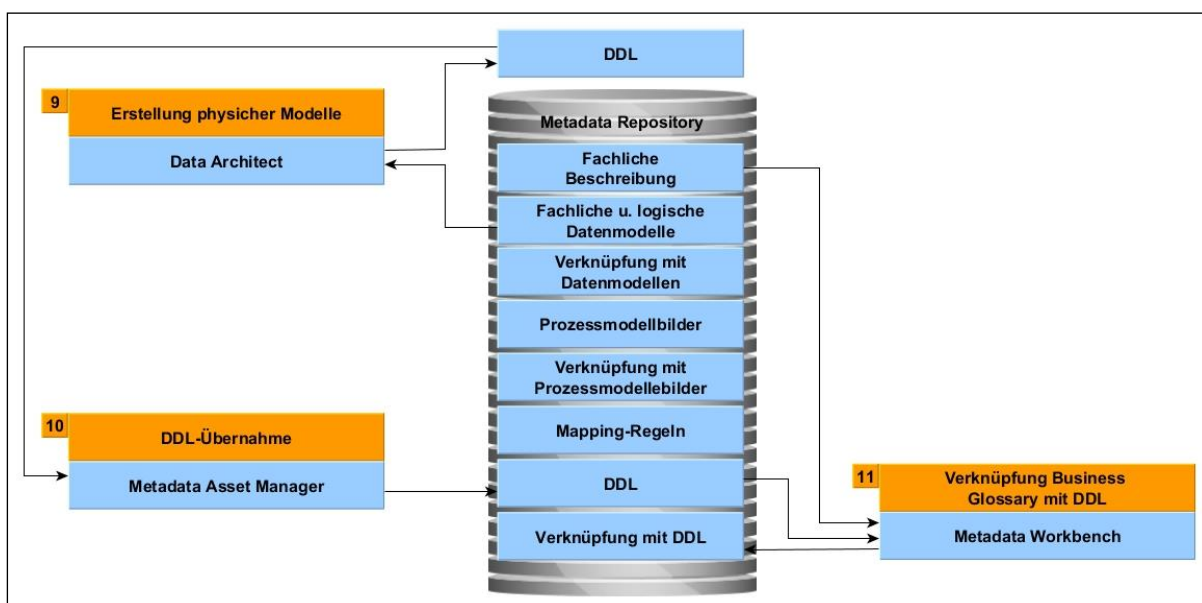
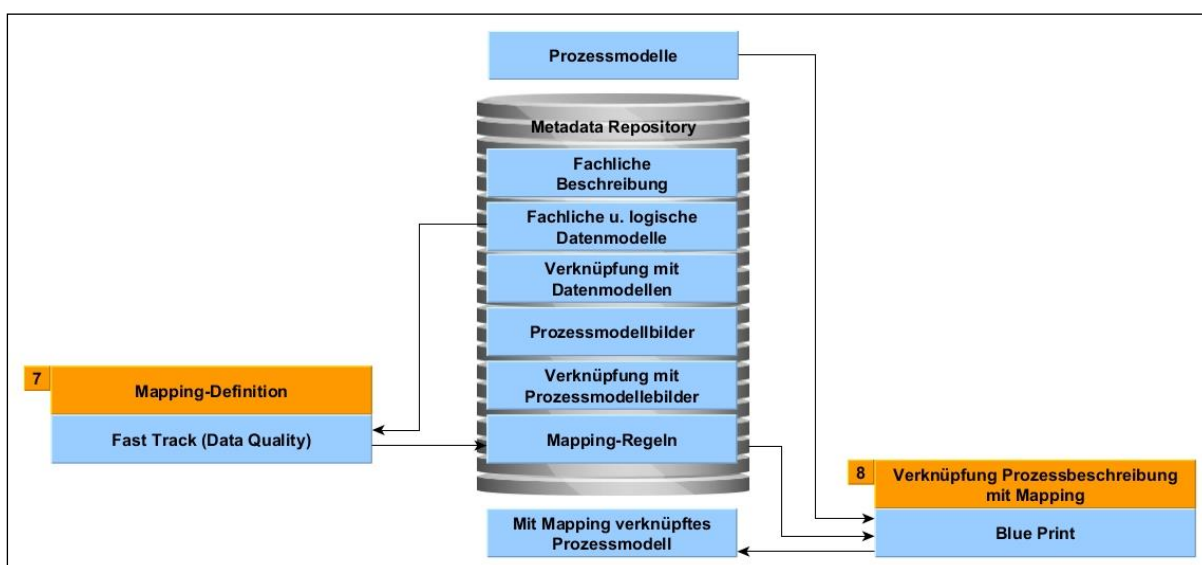
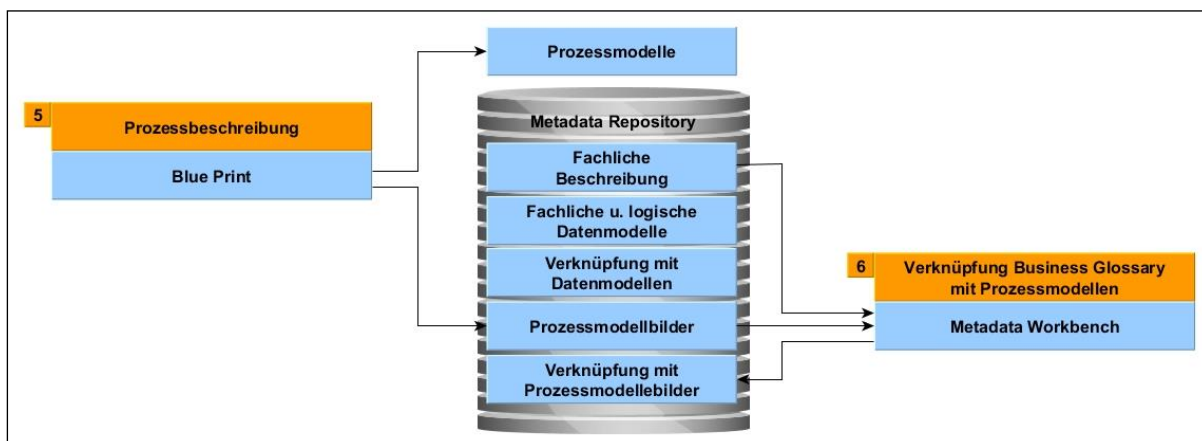
In den folgenden Abbildungen wird eine grobe Sicht auf den gesamten Modellierungs- und Entwicklungsprozess unter Berücksichtigung der vorgesehenen Werkzeuge geboten. Es werden dabei folgende Darstellungselemente verwendet:

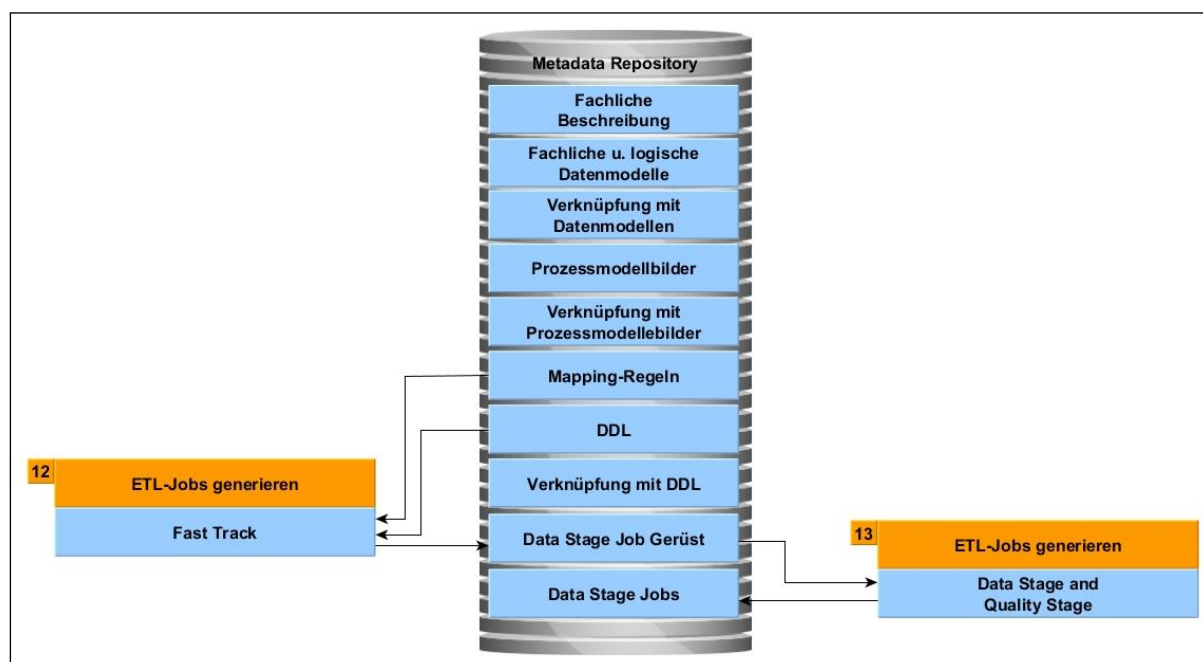


In den Abbildungen wird dargestellt, mit welchem „IBM-Werkzeug“ ein abstrakter „Prozess-Schritt“ umgesetzt werden kann und welcher „Inhalt“ im „Metadata Repository“ abgelegt oder verwendet wird.

Es folgend die Abbildungen mit den einzelnen Prozess-Schritten, die weiter unten in knapper Form erläutert werden:







Anforderungen des Fachbereichs werden in einem Anforderungssystem erfasst. Auf Basis dieser Anforderungen werden fachliche Begriffe und Kennzahlen definiert und im Business Glossary beschrieben. Diese sogenannten Terme werden direkt im Metadaten Repository abgelegt. Nach der fachlichen Beschreibung kann im Anforderungssystem eine Freigabe der Anforderung an das Business Intelligence Competence Center (BICC) zur Umsetzung erfolgen.

Zunächst werden an allen statischen Modellen mit dem Data Architect die erforderlichen Änderungen vorgenommen. Dazu werden die folgenden Modelle erweitert:

- Dimension Fact Model (DFM)
- Domänenmodell
- DMS-Modell
- IDS-Modell (in Data Vault)
- HDS-Modell (nur als System of Records um das nachfolgende Mapping erstellen zu können; wird über Reverse Engineering aus den physischen Datenbanktabellen gewonnen)

Diese Modelle werden außerhalb des Metadata Repository geführt. Die Integration dieser konzeptuellen und logischen Datenmodelle im Metadata Repository erfolgt mit Hilfe des Metadata Asset Managers und der Metadata Workbench bzw. dem Business Glossary.

Nach Erstellung der statischen Modelle kann mit BluePrint eine Prozessübersicht dargestellt werden. Diese vermittelt einen groben Einblick über die Abbildungen von der HDS in die IDS auf der einen und von der IDS in die DMS auf der anderen Seite. Blue Print Diagramme werden prinzipiell außerhalb des Metadata Repository abgelegt. Es gibt aber die Möglichkeit über eine explizite „Veröffentlichung“ die Diagramme in das Metadata Repository zu importieren und über die Metadata Workbench bzw. dem Business Glossary Verknüpfungen herzustellen.

Mit FastTrack wird das Mapping von der HDS in die IDS und von der IDS in die DMS mit spezifiziert. Dieses Werkzeug legt die Informationen direkt im Metadata Repository ab.

Außerdem erfolgt mit FastTrack der Übergang von der Modellierung zur Entwicklung, da damit bereits eine technische Definition der ETL-Jobs erstellt wird. FastTrack enthält auch Möglichkeiten der Verbesserung der Datenqualität. Die BluePrint-Diagramme können (außerhalb des Metadata Repository) mit den Mapping-Definitionen verknüpft werden.

Die logischen Datenmodelle der DMS und IDS werden mit dem DataArchitect als DDL (physisches Modell) erzeugt werden. Die Erstellung der physischen HDS erfolgt außerhalb dieses Prozesses, in dem Sinn, dass die HDS für die Umsetzung von Anforderungen bereitgestellt wird. Die durch die DDL definierten physischen Datenstrukturen sind auf der Zieldatenbank (Netezza) anzulegen.

DataStage wird zur Entwicklung der ETL-Jobs herangezogen. Im dargestellten Prozess werden ETL-Jobs bereits als Ergebnis des FastTrack-Mappings generiert, d.h. im Idealfall müssen in DataStage nur jene Teile ergänzt werden, die in FastTrack nicht vollständig ausspezifiziert oder nicht optimal erzeugt wurden. DataStage greift auf die Informationen der physischen Datenstrukturen zu und schreibt Informationen direkt in das Metadata Repository. QualityStage ist ein in DataStage integrierter Teil zur Überprüfung und Verbesserung der Datenqualität.

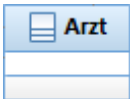


7 Modellierungsrichtlinien

In diesem Abschnitt werden einzelne Modellelement zur Modellierung in der fachlichen und logischen Modellebene (im Sinne von Modellierungsrichtlinien) dargestellt. Auf der fachlichen Modellebene wird auf das Domänenmodell und Dimensional Fact Model (DFM) eingegangen. Die Unterabschnitte der logische Modellebene fokussieren das Star-Schema der DMS und das Data Vault Modell der IDS.

7.1 Fachliche Objektmodelle

7.1.1 Modellierung des Domänenmodells

Das Domänenmodell ist ein konzeptuelles Modell, in dem relevante Geschäftsobjekte und deren Beziehungen dargestellt werden, um den fachlichen Kontext besser verstehen zu können. Die Erstellung erfolgt mit einfachen UML-Klassendiagrammen im IBM InfoSphere DataArchitect. Dazu werden die folgenden Modellelemente verwendet:



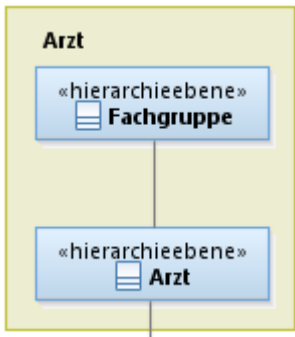

Modellelement	Symbol	Erläuterung
Fachliches Objekt (Klasse)		Eine Klasse beschreibt ein fachliches Objekt der Domäne. Beispiele: Leistungserbringer, Arzt, ambulante ärztliche Leistung, erbrachte ambulante ärztliche Leistung
Eigenschaft eines fachlichen Objekts (Attribut)		Wichtige Eigenschaften fachlicher 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 Objekten (Assoziation)	1-----^* $0..1 \text{-----}^*$ $* \text{-----}^*$	Eine Assoziation stellt eine fachliche Beziehung zwischen fachlichen Objekten dar (soll auch fachlich beschriftet werden). Dabei 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.

Folgende Richtlinien sind zu berücksichtigen:

1. Das gesamte Domänenmodell wird in fachliche Teilmodelle gegliedert (z.B. Versicherter, Krankenhaus, ...). Jedes Modellelement ist genau einem Teilmodell zugeordnet, kann aber in anderen Teilmodellen als Verknüpfung verwendet werden. In einem separaten Modell können alle Teilmodelle als Verknüpfungen zusammengeführt werden, um einen Gesamtüberblick zu erhalten. Der Name jedes Domänenmodells setzt sich aus einem Präfix und einem fachlichen Namen wie folgt zusammen: `DOM_fachlicherName` (z.B. `DOM_Versicherter`)
2. Ein Objekt (Klasse) im Domänenmodell (z.B. Leistungserbringer, Arzt) muss einem Business Term im Business Glossary entsprechen und muss sich dort auch wiederfinden. Gleiches gilt für Attribute (z.B. Arztnummer).
3. Als Attribute sollen maximal wichtige fachliche Schlüssel in den Klassen eingetragen werden. Ein detailliertes Ausspezifizieren von Klassen bis auf Attributsebene ist zu unterlassen. Das Domänenmodell soll einen fachlichen Überblick geben und erhebt nicht den Anspruch auf Detailtreue.
4. Objekte, Attribute und Beziehungen sind fachlich sprechend möglichst ohne Abkürzungen zu bezeichnen (z.B. Leistungserbringer und nicht LB). Bei der Benennung ist der fachlich übliche Sprachgebrauch anzuwenden (z.B. Ambulante ärztliche Leistung). Es muss dabei auf keine maximale Zeichenanzahl geachtet werden. Der gesamte Zeichenvorrat kann herangezogen werden. Die Namensvergabe soll möglichst an 21c (und nicht an alte Systeme) angelehnt werden.
5. Jede Beziehung zwischen Objekten (Klassen) ist fachlich zu benennen. Leserichtung und Kardinalitäten sind anzugeben.
6. Gibt es zu Objekten (Klassen) oder Beziehungen weitere Anmerkungen (z.B. Bedingungen) können diese verbal formuliert und als UML-Notiz dem Modellelement angefügt werden.

7.1.2 Modellierung des Dimension Fact Model (DFM)

Das Dimension Fact Model (DFM) stellt wie das Domänenmodell ein konzeptuelles Modell dar und wird mit IBM InfoSphere DataArchitect als UML-Klassendiagramm erstellt. Darin werden Kennzahlen, Dimensionen und deren Auswertungshierarchien abgebildet. Folgende Modellelemente werden dazu herangezogen:

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.

Folgende Richtlinien sind zu berücksichtigen:

1. Das Dimension Fact Model (DFM) wird in fachliche Teilmodelle untergliedert. Der Name jedes DFM setzt sich aus einem Präfix und einem fachlichen Namen wie folgt zusammen: DFM_ fachlicherName (z.B. DFM_Versichertenbestand)
2. Um in die Modellelemente Hierarchieebenen und Fakt unterscheiden zu können, werden zwei Stereotypen eingeführt: hierarchieebene, fakt
3. Die Beziehungen werden ohne Kardinalität und Beschriftung angegeben, da die Bedeutung der Beziehungen im DFM unmissverständlich ist.

4. Falls alternative Hierarchien dargestellt werden müssen, kann das UML-Element XOR herangezogen werden.

5. Im Modellelement Hierarchieebenen werden keine Attribute angegeben.

6. Alle Zeichen zur Vergabe des fachlichen Bezeichners sind erlaubt.


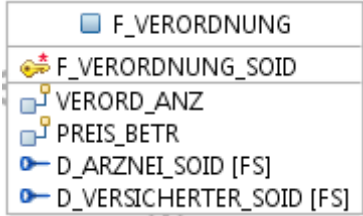

7.2 Logische Datenmodelle

Logische Datenmodelle werden 1:1 in physische abgebildet. Drei Architekturschichten sind dabei zu berücksichtigen:

Architekturschicht	Name des Schemas	Anmerkung
Data Mart Schicht	DMS	Aus dem logischen DMS-Datenmodell wird mit IBM InfoSphere DataArchitect das physische Datenbankschema generiert.
Integrierte Datenschicht	IDS	Aus dem logischen IDS-Datenmodell wird IBM InfoSphere DataArchitect das physische Datenbankschema generiert.
Historisierte Datenschicht	HDS	Über Reverse-Engineering wird IBM InfoSphere DataArchitect aus dem physischen HDS-Schema das HDS-Datenmodell als „System of Records“ generiert.

7.2.1 Modellierung der Data Mart Schicht (DMS)

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:

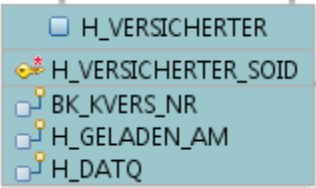
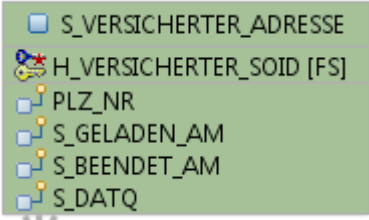
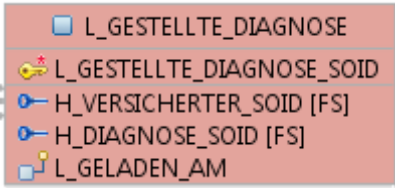
Modellelement	Symbol	Erläuterung
Dimension (Entität)		Eine Dimension wird als Entität (Tabelle) dargestellt. <u>Konventionen:</u> <ul style="list-style-type: none"> Der Name einer Dimension beginnt mit dem Präfix D_.
Fakten (Entität)		Fakten werden als Entität (Tabelle) dargestellt, die Kennzahlen enthält 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.


Folgende Richtlinien sind zu berücksichtigen:

1. Das DMS-Modell wird in Teilmodelle untergliedert. Der Name jedes Teilmodells setzt sich aus einem Präfix und einem fachlichen Namen wie folgt zusammen: DMS_ *fachlicher Name* (z.B. DMS_ Versichertenbestand). Ein Teilmodell der DMS entspricht einem DFM-Teilmodell (z.B. zum Modell DMS_Versichertenbestand gibt es auch ein Modell DFM_Versichertenbestand).
2. Bezeichner für Dimensionen und Fakten werden aus einem Präfix (D_ für Dimensionen und F_ für Fakten) und einem sprechenden Namen gebildet (z.B. D_VERSICHERTER, F_VERSBESTAND).
3. Für Dimensions- und Faktentabellen werden jeweils eigene Primärschlüssel vergeben. Der Schlüsselname beinhaltet dabei den Tabellennamen und den Suffix _SOID (z.B.: D_VERSICHERTER_SOID, F_VERSBESTAND_SOID).
4. Beziehungen werden nicht beschriftet. Die Kardinalität wird in „Krähenfußnotation“ angegeben.
5. Attributnamen werden gemäß den bestehenden Namensrichtlinien festgelegt.

7.2.2 Modellierung der Integrierten Datenschicht (IDS)

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)		<p>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.</p> <p>Zusätzlich werden noch administrativen Attribute, die das Ladedatum und die Datenquelle angeben, definiert.</p>
Satellit (Entität)		<p>Im Satelliten 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.</p> <p>Mit einem Endedatum als zusätzliches Attribut wird die Historisierung gesteuert. Dieses gibt an, ob der Datensatz noch gültig ist oder bereits „beendet“ wurde.</p> <p>Zusätzlich werden noch administrativen Attribute, die das Ladedatum und die Datenquelle angeben, definiert.</p>
Link (Entität)		<p>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.</p>

Beziehungen (Relationship)		Beziehungen sind logisch zu sehen. Fachliche Beziehungen werden über Links abgebildet. Logische Beziehungen existieren <ul style="list-style-type: none"> • zwischen Hubs und Satelliten, • zwischen Links und Satelliten, • zwischen Hubs und Links und • zwischen Links und Links.
-------------------------------	---	--

Folgende Richtlinien sind zu berücksichtigen:

1. Das IDS-Modell wird in Teilmodelle gegliedert. Der Name jedes Teilmodells setzt sich aus einem Präfix und einem fachlichen Namen wie folgt zusammen: *IDS_fachlicher Name* (z.B. *IDS_Versicherter*). Ein Teilmodell der IDS entspricht einem DOM-Teilmodell (z.B. zum Modell *IDS_Versicherter* gibt es auch ein Modell *DOM_Versicherter*).
2. Referenztabellen werden mit Hubs, Satelliten und Links abgebildet.
3. Hubs besitzen den Präfix *H_* gefolgt von einem sprechenden Namen (z.B. *H_VERSICHERTER*).
4. Satelliten besitzen den Präfix *S_* gefolgt von einem sprechenden Namen (z.B. *S_VERSICHERTER_KONTAKT*, *S_VERSICHERTER_ADRESSE*).
5. Business Satelliten sind spezielle Satelliten, die abgeleitete Werte beinhalten. Sie beginnen mit dem Präfix *SB_* (steht für „Satellit Business“).
6. Links besitzen den Präfix *L_* gefolgt von einem möglichst sprechenden Namen, der die Bedeutung der Beziehung widerspiegelt (z.B. *L_GESTELLTE_DIAGNOSE*).
7. Da aus dem IDS-Modell ein physisches Datenbankschema abgeleitet wird, sind die Regeln der Datenbank bei der Namensvergabe einzuhalten: keine Umlaute, als Sonderzeichen ist nur „_“ erlaubt, Berücksichtigung der max. Länge von Bezeichnern.
8. Es sollen möglichst deutsche Bezeichner verwendet werden (siehe auch § 119 SGB10)
9. Die Namen von Tabellen und Attributen sind in Großbuchstaben zu schreiben. Wörter werden dabei mit „_“ getrennt.
10. Für Attribute gelten die bestehenden Standards und Richtlinien.
11. Business Keys in Hubs beginnen mit dem Präfix *BK_*.
12. Die folgenden Standardattribute sind bei Hubs zu führen:
 - Primärschlüssel: *H_Hubname_SOID* (z.B. *H_VERSICHERTER_SOID*)

- Ladedatum: H_GELADEN_AM (timestamp)
- Datenquelle (Name der Quelltable): H_DATQ (varchar(254))

13. Die folgenden Standardattribute sind bei Satelliten zu führen:

- Ladedatum: S_GELADEN_AM (timestamp)
- Endedatum: S_BEENDET_AM (timestamp); leer, falls der Datensatz noch gültig ist, ansonsten belegt mit dem Tag, ab dem der Datensatz nicht mehr gültig ist.
- Datenquelle (Name der Quelltable): S_DATQ (varchar(254))
- Achtung: Satelliten besitzen keinen eigenen Primärschlüssel

14. Die folgenden Standardattribute sind bei Links zu führen:

- Primärschlüssel: L_Linkname_SOID (z.B. L_GESTELLTE_DIAGNOSE_SOID)
- Ladedatum: L_GELADEN_AM (timestamp)

7.2.3 Modellierung der Historisierten Datenschicht (HDS)

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 im DataArchitect nur Entitäten ohne Beziehungen zur Verfügung. Die Namen werden aus dem physischen Datenbankschema übernommen. Eine Richtliniendefinition ist aus diesem Grunde nicht erforderlich.

8 Fazit

Diese Arbeit zeigt einen real durchgeführten Proof of Concept, in dem die in (Neuböck T., Raab J., Weißenböck A., 2014) eingeführte agile modellorientierte Vorgehensweise zur Entwicklung von BI- und DWH-Systemen am Beispiel einer realen DWH-Umgebung (IBM Netezza und IBM InfoSphere) demonstriert wird. Es wurde dargestellt, welche abstrakte Prozess-Schritte mit welchen Werkzeugen durchgeführt und welche Artefakte damit erstellt werden können. Es konnte gezeigt werden, dass mit geeigneter Werkzeugunterstützung alle Modellebenen zumindest bis zu einem gewissen Grad abgedeckt werden können und damit eine modellorientierte DWH-Entwicklung vollzogen werden kann, wobei zuerst fachliche (konzeptuelle) Modelle und danach logische Modelle erstellt wurden. Aus den logischen Modellen werden physische (häufig automatisiert oder semi-automatisiert) generiert. Auch wenn in den einzelnen Abbildungen auf ein konkretes Werkzeug Unterschiede zwischen Werkzeugherstellern bestehen, war es das vordergründige Ziel, die generelle Sinnhaftigkeit und Realisierbarkeit (auf Basis konkreter Werkzeuge) einer agilen modellorientierten Vorgehensweise in der DWH-Entwicklung aufzuzeigen.

9 Literatur

Balzert H. (1998). *Lehrbuch der Software-Technik. Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung*. Berlin: Spektrum Akademischer Verlag.

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.

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.