

IMISE-REPORTS

Herausgegeben von Professor Dr. Markus Löffler

A. Winter, L. Ißler, F. Jahn, A. Strübing, T. Wendt

Das Drei-Ebenen-Metamodell für die Modellierung und Beschreibung von Informationssystemen (3LGM² V3)

IMISE-REPORT Nr. 1/2010

UNIVERSITÄT LEIPZIG

Medizinische Fakultät

IMPRESSUM

Herausgeber: Prof. Dr. Markus Löffler

Redakteur: Franziska Jahn

Institut für Medizinische Informatik, Statistik und Epidemiologie (IMISE)

Härtelstr. 16/18, 04107 Leipzig

Tel.: (0341) 97-16100, Fax (0341) 97-16109

Internet: <http://www.imise.uni-leipzig.de>

Druck des Einbandes und Bindung: Buch- und Offsetdruckerei Herbert Kirsten

Redaktionsschluss: 09.04.2010

© IMISE 2010

Alle Rechte vorbehalten. Nachdruck nur mit ausdrücklicher Genehmigung des Herausgebers und mit Quellenangabe gestattet.

ISSN 1610-7233

Inhaltsverzeichnis

A	EINLEITUNG.....	3
A.A	PROBLEMATIK	3
A.B	ZIEL DER ARBEIT	5
A.C	AUFBAU DER ARBEIT.....	5
B	DREI GEGENSTANDEBENEN EINES INFORMATIONSSYSTEMS (DREI-EBENEN-METAMODELL).....	6
C	UML-KLASSENDIAGRAMME ZUR BESCHREIBUNG DES DREI-EBENEN-METAMODELLS	8
C.A	KLASSEN	8
C.B	ASSOZIATIONEN	9
C.C	ASSOZIATIONSKLASSEN.....	9
C.D	AGGREGATION UND KOMPOSITION ALS SPEZIELLE ASSOZIATIONEN	9
C.E	BEDINGUNGEN.....	10
D	UMSETZUNG DES DREI-EBENEN-METAMODELLS IM 3LGM²-BAUKASTEN	11
E	DAS DREI-EBENEN-METAMODELL 3LGM² ZUR BESCHREIBUNG VON INFORMATIONSSYSTEMEN	13
E.A	BASIS, ERWEITERUNGEN UND MODIFIKATIONEN DES METAMODELLS	13
E.B	DIE BASIS DES DREI-EBENEN-METAMODELLS: 3LGM ² -B	13
E.B.A	Überblick	13
E.B.B	Fachliche Ebene.....	15
E.B.C	Zentrale Begriffe der Logischen und Physischen Werkzeugebene.....	30
E.B.D	Logische Werkzeugebene.....	32
E.B.E	Physische Werkzeugebene.....	41
E.C	ERWEITERUNGEN DES METAMODELLS BZW. MODIFIKATIONEN DES METAMODELLS	51
E.C.A	Nachrichtenorientierte Kommunikation mit Nachrichtentypen: 3LGM ² -M.....	51
E.C.B	Serviceorientierte Kommunikation: 3LGM ² -S.....	57
F	ANFORDERUNGEN AN DEN 3LGM²-BAUKASTEN.....	68
F.A	DARSTELLUNG VON 3LGM ² -MODELLEN IM 3LGM ² -BAUKASTEN	68
F.A.A	Baumansicht.....	68
F.A.B	Grafische Einzelebenenansicht	69
F.A.C	Grafische 3-Ebenen-Ansicht	69
F.A.D	Matrixsicht.....	70
F.A.E	Eigenschaftsdialoge.....	71
F.B	AKTUELLE VERSION DES 3LGM ² -BAUKASTENS	71
F.C	ANFORDERUNGEN AN DEN 3LGM ² -BAUKASTEN AUS DEM METAMODELL	71
F.C.A	Anforderungen seitens der Elementklassen.....	71
F.D	HANDHABUNG DES BASIS-METAMODELLS UND ERWEITERUNGEN DES METAMODELLS	79
F.E	KONVERTIERUNG VON MODELLEN AUF DER BASIS VON METAMODELL V1.X ZU MODELLEN AUF DER BASIS VON 3LGM ²	79
F.E.A	Allgemein	79
F.E.B	Logische Werkzeugebene.....	79
F.E.C	Physische Werkzeugebene.....	80
F.F	KONVERTIERUNG ZWISCHEN DEN METAMODELLEN VON 3LGM ²	80
F.G	ASSOZIATIONSKLASSEN DES METAMODELLS IM 3LGM ² -BAUKASTEN.....	80
G	LITERATUR	82
H	STICHWORTVERZEICHNIS	84

I	DANKSAGUNG	86
J	HINWEISE.....	87

A Einleitung

A.A Problematik

Krankenhäuser sind Dienstleistungsunternehmen, die sich durch eine sehr heterogene Unternehmensstruktur auszeichnen. Ein Krankenhaus ist aus einer Vielzahl von Organisationseinheiten aufgebaut, die zur Unterstützung ihrer Informationsverarbeitung unterschiedlichste rechnerbasierte und nicht-rechnerbasierte Werkzeuge benötigen. Neben den klassischen Organisationseinheiten jedes Unternehmens wie Buchhaltung, Personalverwaltung oder Bestellwesen sind es insbesondere die Stationen und Ambulanzen sowie die medizinischen Leistungsstellen, die je nach ihrer Aufgabe Bilder oder Bildsequenzen (z. B. in der Radiologie), Signale (z. B. in der EKG-Abteilung) oder Textinformation zu verarbeiten haben. Die Menge der anfallenden Daten variiert sowohl hinsichtlich der Anzahl der Parameter als auch hinsichtlich der Sequenz der Parameter. Der Behandlungsprozess für einen Patienten gestaltet sich individuell und ist von vielen Faktoren abhängig, die zum Teil nicht im Voraus planbar sind. Es ist daher eine hohe Interoperabilität zwischen den am Behandlungsprozess beteiligten Organisationseinheiten (interne Kommunikation) notwendig. Dies gilt auch für die externe Kommunikation mit anderen an der Behandlung des Patienten beteiligten Einrichtungen (z. B. Versicherungen, niedergelassene Ärzte, andere Krankenhäuser oder Nachsorgeeinrichtungen). Da Gesundheit und Leben eines Patienten im Krankenhaus von den zur Behandlung vorliegenden Informationen und daher vom Informationssystem abhängig sind, ergibt sich eine besondere Sorgfaltspflicht bei der Informationsverarbeitung und beim Informationsmanagement.

Auch wenn mittlerweile ein Großteil der Aufgaben eines Krankenhauses durch rechnerbasierte Werkzeuge unterstützt werden kann, hat sich gezeigt, dass Krankenhausinformationssysteme nicht als abgegrenzte rechnerunterstützte Systeme, sondern als sozio-technisches System aufzufassen sind ([1], [2]).

Definition: Krankenhausinformationssystem (KIS)

Ein *Krankenhausinformationssystem* ist das sozio-technische Teilsystem eines Krankenhauses, welches alle datenverarbeitenden Prozesse und die an ihnen beteiligten menschlichen und maschinellen Handlungsträger in ihrer datenverarbeitenden¹ Rolle umfasst.

Somit sind *Krankenhausinformationssysteme* zugleich Teilsystem des Krankenhauses und spiegeln seine Heterogenität entsprechend wider. Aufgrund der unterschiedlichen Anforderungen an die Werkzeuge zur Informationsverarbeitung basieren diese häufig auf unterschiedlichsten Hardware-, Software- und anderen Produkten verschiedenster Hersteller. Hinzu kommt, dass *Krankenhausinformationssysteme* historisch wachsen, so dass es neben modernen datenverarbeitenden Werkzeugen immer auch sogenannte 'Altsysteme' (legacy systems; vgl. [3]) geben wird. Es kann nicht davon ausgegangen werden, dass alle datenverarbeitenden Werkzeuge eines *Krankenhausinformationssystems* zu einem Zeitpunkt komplett ausgetauscht werden (können).

Das bisher Gesagte gilt in ähnlicher Form nicht nur für Krankenhäuser mit ihren *Krankenhausinformationssystemen*, sondern für alle *Organisationen* (Unternehmen, Einrichtungen, Verbände, ...) des Gesundheitswesens oder gar anderer Branchen. Der Begriff des *Krankenhausinformationssystems* kann wie folgt verallgemeinert werden [4]:

Definition: Informationssystem (IS)

Ein *Informationssystem* ist das sozio-technische Teilsystem einer *Organisation*, welches alle datenverarbeitenden Prozesse und die an ihnen beteiligten menschlichen und maschinellen Handlungsträger in ihrer datenverarbeitenden Rolle umfasst.

Damit sind *Krankenhausinformationssysteme* die *Informationssysteme* von Krankenhäusern.

Um die Heterogenität eines *Informationssystems* beherrschen zu können, muss das Informationsmanagement *Informationssysteme* beschreiben, d. h. modellieren können. Geeignete Modelle sollten in der Lage sein, beispielsweise folgende Fragen des Informationsmanagements zu beantworten:

¹ Auch wenn es üblich ist, von Informationssystemen zu sprechen, werden in diesen Systemen nicht Informationen, sondern Daten, welche Informationen repräsentieren, verarbeitet. Wir wollen daher hier wieder den scheinbar veralteten, aber richtigeren Begriff der Datenverarbeitung verwenden.

- Welche Aufgaben von Medizinern unterstützt ein bestimmter Klinischer Arbeitsplatz auf einer bestimmten Station in unserem Krankenhaus?
- Wie viele verschiedene Anwendungen brauchen die Mediziner, um ihre Arbeit auf Station erledigen zu können?
- Welche Anwendung wird für die Anforderungserstellung und die Darstellung von Befunden am Klinischen Arbeitsplatz genutzt? Wer ist der Hersteller der entsprechenden Software, welches Datenbankverwaltungssystem wird genutzt und welche Server werden in Anspruch genommen?
- Bei der Kommunikation zwischen der Datenbank auf dem Server und dem Klinischen Arbeitsplatz treten manchmal Fehler aufgrund von Zeitüberschreitungen auf. Welche Netzwerkkomponenten könnten diese Fehler verursachen?
- Welche Schnittstellen zum Laborinformationssystem werden für die Kommunikation von Anforderungen und Befunden genutzt? Wird HL7 als Protokoll für die Server-Server-Kommunikation genutzt oder ist eine proprietäre Client-basierte Schnittstelle zum Intranetserver des Labors installiert? Wie wird im letzteren Fall visuelle Integration [5] garantiert?
- Zu welcher Netzwerkdomeäne gehört der Klinische Arbeitsplatz und wer ist der verantwortliche Domänen-Administrator?
- Um einen weiteren klinischen Arbeitsplatz auf der Station zu realisieren, soll ein zusätzlicher PC installiert werden. Welche Stecker, Kabel und Schnittstellenkarten sind notwendig, um den PC mit dem Netzwerk zu verbinden?
- In welcher Form müssen Informationen zwischen welchen Anwendungssystemen auf welchen Kommunikationswegen transportiert werden, wenn eine Folge bestimmter Aufgaben erledigt werden muss?
- Werden auf einem Kommunikationsweg die Informationen nur mit Hilfe elektronischer Nachrichten oder auch durch Dokumente auf gegenständlichen Datenträgern (z. B. Papier, Film, CD) transportiert.
- Zwischen welchen Geräten oder Orten können Daten ohne den Transport bestimmter gegenständlicher Datenträger übermittelt werden (z. B. über ein Kabelnetz oder über ein Funknetz) und wo ist das nicht möglich?
- Für den Fall, dass eine Datenübertragungsverbindung getrennt wird: Bei welchem Anwendungssystem kann dann ein Client-PC nicht mehr auf einen der Server über eine nicht-gegenständliche, signalbasierte Verbindung zugreifen? Sind alle Clients aller Anwendungssysteme mit ihren Servern verbunden?
- Auf welchen Geräten oder an welchen Orten werden Daten in welcher Form gespeichert bzw. gelagert?
- Welche Hardware ist vorhanden und wo befindet sie sich?

Bezogen auf konkrete Anwendungsfälle sollte ein Modell beispielsweise folgende Fragen beantworten können:

- Fragen eines Mitarbeiters im Bereitschaftsdienst der IT-Abteilung:
 - Welche Anwendungssysteme waren auf dem Stationscomputer der Urologie installiert und welche speziellen Installationsparameter sind anzugeben, wenn die unwiderruflich defekte Festplatte des PCs durch eine neue ersetzt wurde?
 - Welche Anwendungssysteme und damit verbunden welche Unternehmensaufgaben sind krankenhausweit betroffen, wenn ein zentraler Server ausfällt? Welche Abteilungen des Krankenhauses sind in einem solchen Fall zu benachrichtigen und unter welcher Rufnummer ist der zuständige Mitarbeiter oder die beauftragte Servicefirma auch an Wochenenden erreichbar?
- Fragen vor der Neueinführung eines Digitalen Archivs:
 - Ist ein Referenzmodell verfügbar, in welchem die Aufgaben, die ein digitales Archiv erfüllen muss, beschrieben sind und das für die Erstellung eines Sollmodells verwendet werden kann?
 - Welche Aufgaben aus dem Bereich der digitalen Archivierung werden bereits durch rechnerbasierte Anwendungssysteme ausgeführt? Welche im Referenzmodell beschriebenen Aufgaben werden ggf. nicht gefordert? Welche Aufgaben sollen ggf. zusätzlich erfüllt werden?
 - Welche rechnerbasierten Anwendungssysteme mit Bezug zur digitalen Archivierung sind im eigenen Krankenhaus im Einsatz? Welche Schnittstellen sind vorhanden?
 - Wie sieht die physische Werkzeugebene (Server, PCs, Netzkomponenten etc.) des eigenen Krankenhauses aus?

A.B Ziel der Arbeit

Vorrangiges Ziel dieser Arbeit ist es, die für das Informationsmanagement wesentlichen Komponenten eines *Informationssystems* zu identifizieren und sie zusammen mit ihren Beziehungen und Abhängigkeiten in einem Metamodell zusammenzufassen. Dabei wird gezielt der Modellbegriff von SCHÜTTE verwendet: „Ein Modell ist das Ergebnis einer Konstruktion eines Modellierers, der für Modellnutzer Elemente eines Systems zu einer Zeit als relevant mit Hilfe einer Sprache deklariert.“ [6]. Das Metamodell soll es also ermöglichen, möglichst so detailliert (oder eben nicht-detailliert) beschreiben zu können, wie es für die entsprechenden Zwecke relevant ist.

Gleichzeitig soll in dieser Arbeit auch der korrekte und zweckmäßige Gebrauch der Metamodell-Konstrukte bei der 3LGM²-Modellierung vermittelt werden. Innerhalb dieser Vorgehensmodelle wird insbesondere auch auf die 3LGM²-Modellierung aktueller Trends in der IT, wie z. B. Virtualisierung oder service-orientierte Architekturen, eingegangen.

Darüber hinaus soll auch die praktische Relevanz der 3LGM²-Modellierung verdeutlicht werden, indem Fragen des Informationsmanagements, die sich mit Hilfe des 3LGM² bzw. des 3LGM²-Baukastens analysieren lassen, identifiziert werden.

A.C Aufbau der Arbeit

Kapitel B führt die drei Gegenstandsebenen des Drei-Ebenen-Metamodells ein. In Kapitel C wird erklärt, welche Konstrukte der Unified Modeling Language (UML) benutzt wurden, um die Elemente auf den Gegenstandsebenen und deren Beziehungen untereinander zu beschreiben. Um die ausführliche Beschreibung des Drei-Ebenen-Modells in Kapitel E zu verstehen, ist es nötig, den grundsätzlichen Aufbau des 3LGM²-Baukastens zu kennen. Der 3LGM²-Baukasten, der die Erstellung 3LGM²-konformer Modelle von Informationssystemen unterstützt, wird in Kapitel D eingeführt. Kapitel E stellt das Metamodell 3LGM² vor, welches aus dem 3LGM²-B (Basis-Metamodell) sowie den Erweiterungen und Modifikationen 3LGM²-M (Metamodell für nachrichtenbasierte/messagebasierte Kommunikation) und 3LGM²-S (Metamodell für diensteorientierte/serviceorientierte Kommunikation) besteht. Dabei werden die vorgestellten Modellierungskonstrukte anhand eines durchgängigen, praxisnahen Beispiels erklärt. Das Beispielmodell beschreibt ein Informationssystem der Radiologieabteilung eines Krankenhauses.

Die Kapitel E.C.A-8, E.C.B-12 und E.C.B-13 beschreiben die Konvertierungsmöglichkeiten zwischen 3LGM²-B, 3LGM²-M und 3LGM²-S. Kapitel F stellt die Anforderungen an den 3LGM²-Baukasten zusammen, die für diese Version des Metamodells (3LGM² V3) erfüllt sein müssen. In Kapitel F.E wird beschrieben, wie Modelle aus dem bisherigen Metamodell in das neue Metamodell konvertiert werden können.

B Drei Gegenstandsebenen eines Informationssystems (Drei-Ebenen-Metamodell)

Das Metamodell zur Beschreibung von *Informationssystemen* knüpft an das in [7] veröffentlichte Drei-Ebenen-Modell und das in [8] weiterentwickelte Drei-Ebenen-Metamodell an. Dieses bietet ein Grundgerüst von Begriffen, die zur Beschreibung statischer Strukturen von *Informationssystemen* geeignet sind. Mit der hier vorgestellten Version 3 des Metamodells kann sich der Modellierer zwischen einer Basismodellierung (3LGM²-B), der spezielleren Modellierung der nachrichtenorientierten Kommunikation (3LGM²-M) oder der Modellierung der serviceorientierten Kommunikation (3LGM²-S) entscheiden. 3LGM²-B, das zunächst beschrieben wird, bildet dabei die Grundlage, um die Erweiterungen und Modifikationen in 3LGM²-M und 3LGM²-S zu verstehen.



Abbildung 1: Die Gegenstandsebenen des 3LGM² - Der 3LGM²- Tower

Die in 3LGM² verwendeten Konzepte lassen sich, wie es beispielsweise auch mit dem in [9] vorgestellten generischen Architekturrahmen für Informationssysteme vorgeschlagen wird, Gegenstandsebenen zuordnen. Wir unterscheiden eine *Fachliche Ebene*, eine *Logische Werkzeugebene* und eine *Physische Werkzeugebene*:

- *Fachliche Ebene*: Mit den Konzepten der fachlichen Gegenstandsebene kann eine *Organisation* anhand ihrer *Aufgaben* beschrieben werden. Im Rahmen der Erfüllung der *Aufgaben* werden Informationen über Objekte verwendet und auch geschaffen (vgl. auch [10]). Informationen über Objekte liegen in Form von Merkmalen vor. Objekte mit gleichen Merkmalsarten werden zu *Objekttypen* zusammengefasst. Innerhalb von Prozessen können *Aufgaben*, die Informationen über Objekte nacheinander bearbeiten und interpretieren, in eine logische und zeitliche Reihenfolge eingeordnet werden. Zur *Fachlichen Ebene* gehört auch, die Struktur der Organisation, also ihre Gliederung in *Organisationseinheiten* zu beschreiben.

- *Logische Werkzeugebene:* In dieser Gegenstandsebene lässt sich darstellen, welche *logischen Werkzeuge*, die hier als *Anwendungsbausteine* bezeichnet werden, eine *Organisation* zur Erfüllung ihrer *Aufgaben* einsetzt. Es wird beschrieben, wie diese miteinander kommunizieren müssen, damit der auf der *Fachlichen Ebene* beschriebene Zugriff auf Informationen gewährleistet ist. Es wird ebenfalls dargestellt, in welchen *Anwendungsbausteinen* Informationen über Objekte als Daten gespeichert werden.
- *Physische Werkzeugebene:* Mit Hilfe der Begriffe der *Physischen Werkzeugebene* kann man beschreiben, welche *Physischen Datenverarbeitungssysteme* eine *Organisation* einsetzt, und wie diese miteinander vernetzt sein müssen, damit die Kommunikation der auf der *Logischen Werkzeugebene* beschriebenen *Anwendungssysteme* gewährleistet ist.

C UML-Klassendiagramme zur Beschreibung des Drei-Ebenen-Metamodells

Das Metamodell 3LGM² wird mit Hilfe von Klassendiagrammen der Unified Modeling Language (UML) (siehe z. B. [11, 12]) formalisiert. Dabei werden folgende UML-Konzepte (mit den angegebenen, ggf. vom Standard abweichenden Interpretationen) verwendet:

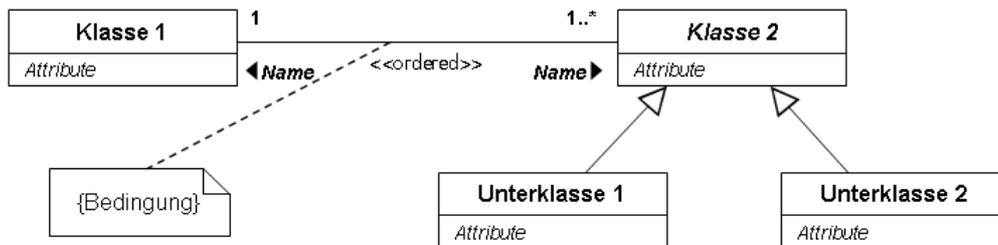


Abbildung 2: UML-Konzepte im 3LGM²-Metamodell.

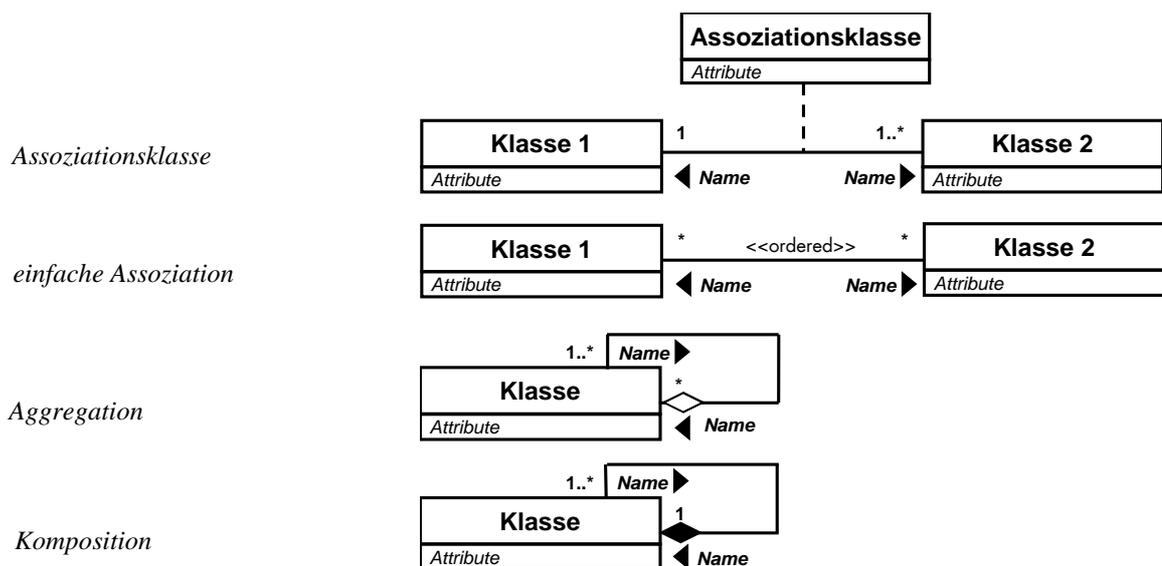


Abbildung 3: Assoziationsarten im 3LGM²-Metamodell

C.A Klassen

- Eine *Klasse* beschreibt die Attribute und Operationen einer Menge gleichartiger Objekte. Jede *Klasse* besitzt einen eindeutigen Klassennamen.
- *Klassen* können abstrakt sein. Von *abstrakten Klassen* können in einem konkreten Modell keine Instanzen gebildet werden. Der Name *abstrakter Klassen* ist kursiv gestellt. Abstrakte *Klassen* besitzen mindestens eine nichtabstrakte Unterklasse.
- Unterklassen ergeben sich als Spezialisierung einer anderen Klasse, zu der eine Generalisierungsbeziehung besteht. Sie erhalten alle Eigenschaften ihrer Oberklasse und können durch weitere *Attribute* und *Assoziationen* näher beschrieben werden.
- Beispiele für Klassen im 3LGM² sind *Aufgabe* (siehe E.B.B-1), *Anwendungsbaustein* (siehe E.B.D-1), *Physisches Datenverarbeitungssystem* (siehe E.B.E-1).

C.B Assoziationen

Alle im Metamodell 3LGM² verwendeten Arten von *Assoziationen* beschreiben nach den üblichen UML-Konventionen Beziehungen zwischen *Klassen*.

- *Klassen* können mit sich selbst über eine *Assoziation* in Beziehung stehen. Konkrete Instanzen einer solchen *Klasse* können aber nur mit anderen Instanzen dieser *Klasse* in Beziehung stehen und niemals mit sich selbst.
- Jede *Assoziation* besitzt zwei gerichtete Namen. Der Name am Ende einer *Assoziation* beschreibt, wie die gegenüberliegende *Klasse* zu der *Klasse* an dem Ende in Beziehung steht.
- Alle *Assoziationen* sind in beide Richtungen navigierbar und können durch eine Zeichenkette der Form „*Klasse 1 Name* ▶ *Klasse 2*“ eindeutig gerichtet beschrieben werden. ‚*Klasse 1*‘ und ‚*Klasse 2*‘ sind dabei die Namen der in Beziehung stehenden *Klassen* und ‚*Name* ▶‘ ist der entsprechende gerichtete Name der *Assoziation*.
- Bei jeder Assoziationsart sind immer an beiden Enden die Multiplizitäten angegeben. Die Multiplizitätsangabe an einem Ende einer *Assoziation* zeigt an, mit wie vielen Elementen der *Klasse* an dem Ende ein Element der gegenüberliegenden *Klasse* minimal verbunden sein muss und maximal verbunden sein kann.
- Bei allen *Assoziationen* (außer mit Stereotyp <<ordered>>) kann es nur eine Verbindung derselben Art zwischen zwei identischen Instanzen geben.
- Ein Beispiel für eine *Assoziation* im 3LGM² sind die Beziehungen „benötigt für Funktionalität“ und „benötigt für Speicherung“ zwischen *Anwendungsbaustein* und *Physischem Datenverarbeitungssystem* (siehe E.B.D-1.9).
- Für *Assoziationen* des Stereotyps <<ordered>> gilt, dass für wenigstens eine der durch diese *Assoziation* verbundenen *Klassen* die Reihenfolge der *Assoziationen* dieser Art eine Bedeutung hat. Für welche der jeweils zwei miteinander verbundenen *Klassen* das gilt, geht aus dem Kontext des Metamodells hervor.
- Über *Assoziationen* mit dem Stereotyp <<ordered>> können abweichend von allen anderen Assoziationsarten dieselben zwei Instanzen von *Klassen* mehrfach miteinander verbunden sein.
- *Prozesse* (siehe E.B.B-6) besitzen im 3LGM² eine *Assoziation* des Stereotyps <<ordered>> zu *Aufgaben*.

C.C Assoziationsklassen

Eine *Assoziation* kann eine *Assoziationsklasse* sein, wenn die *Assoziation* selbst durch Attribute oder eigene *Assoziationen* näher beschrieben werden soll. Beispiel für eine *Assoziationsklasse* im 3LGM² ist die Klasse *Kommunikationsverbindung* (siehe E.B.D-3), zwischen *Sende-* und *Empfangsschnittstelle*.

C.D Aggregation und Komposition als spezielle Assoziationen

Im Drei-Ebenen-Metamodell werden die UML-Konzepte der *Aggregation* und der *Komposition* ausschließlich genutzt, um Beziehungen von *Klassen* zu sich selbst zu beschreiben. Dieses Kapitel erläutert den Unterschied zwischen dem Gebrauch von *Komposition* und *Aggregation* in 3LGM². Es werden die Begriffe *Unterelement* und *Gruppenelement* verwendet, um Instanzen von einer *Klasse* zu beschreiben, die mit anderen Instanzen dieser *Klasse* über eine *Aggregation* oder *Komposition* verbunden sind. Einem *Gruppenelement* können mehrere *Unterelemente* zugeordnet werden, *Gruppenelemente* können wiederum selbst *Unterelement* eines anderen *Gruppenelements* sein.

- Über das UML-Konzept der *Aggregation* wird ausgedrückt, dass die *Unterelemente* eine spezielle Ausprägung des *Gruppenelements* repräsentieren. Z. B. kann der *Gruppenobjekttyp* „Befund“ die unterschiedlichen Ausprägungen „Radiologischer Befund“ und „Laborbefund“ als Unterobjekttypen besitzen. Man könnte die *Unterelemente* auch als Alternativen des *Gruppenelements* beschreiben. Diese *Unterelemente* werden im Folgenden als *spezialisierte Elemente* bezeichnet. Bei der Verwendung einer *Aggregation* im Metamodell wird die Unterscheidung des *Gruppenelements* in seine *Unterelemente* als *Spezialisierung* und das Verallgemeinern der *Unterelemente* zum *Gruppenelement* als *Generalisierung* bezeichnet. Bei einer *Spezialisierung* in einem konkreten Modell erhalten bzw. ‚erben‘ alle *Unterelemente* automatisch die Verbindungseigenschaften der *Gruppenelemente*. Dies bedeutet, dass die konkreten *Assoziationen* von *Gruppenelementen* in einem Modell

auch für *Unterelemente* gelten. Dabei kann Mehrfachvererbung zugelassen sein, also ein Element kann mehrere *Gruppenelemente* besitzen und deren Eigenschaften erben, wenn die maximale Kardinalität der *Aggregation* für *Gruppenelemente* größer als 1 ist. Anders als bei den Verbindungseigenschaften werden Attributwerte eines *Gruppenelements* nicht auf die ihm untergeordneten Elemente übertragen, sondern müssen für diese selbst festgelegt werden.

- Über das UML-Konzept der *Komposition* wird ausgedrückt, welche verschiedenen Teile sich zu einem Ganzen zusammenfügen lassen, also z. B. welche *Unteraufgaben* zusammen die *Gruppenaufgabe* „Patientenaufnahme“ bilden. Das *Unterelement* ist hierbei als echter Teil des *Gruppenelementes* zu verstehen und wird somit auch als *Teilelement* bezeichnet. Dieses Aufspalten von *Gruppenelementen* in *Teilelemente* wird im Folgenden *Zerlegung* genannt. Die Umkehrrichtung der Zerlegung, also das Zusammenfügen von *Unterelementen* zu einem *Gruppenelement* wird im Folgenden als *Komposition* bezeichnet. Zerlegungsbeziehungen sind in den Metamodellidiagrammen von den *Unterelementen* zum *Gruppenelement* mit ‚ist Teil von‘ und vom *Gruppenelement* zu den *Unterelementen* mit ‚hat Teil‘ benannt. Bei *Zerlegungen* ist Vererbung von den *Gruppenelementen* an die *Unterelemente* nicht in jedem Fall möglich. Deshalb wird für jede *Elementklasse*, bei der eine *Zerlegung* möglich ist, im Einzelnen definiert, welche Assoziationen von *Gruppenelementen* an *Unterelemente* vererbt werden.

Nicht für alle Elementklassen ist *Spezialisierung* oder *Zerlegung* möglich. Folgende Tabelle gibt einen Überblick aller *Elementklassen* bei denen mindestens eine Form möglich ist.

Elementklasse	Spezialisierung	Zerlegung	Verweis auf Text
Objekttyp	x	x	E.B.B-1
Aufgabe	x	x	E.B.B-2
Organisationseinheit		x	E.B.B-5
Anwendungsbaustein		x	E.B.D-1
Physisches Datenverarbeitungssystem		x	E.B.E-1
Geräteklasse	x		E.B.E-2
Standort		x	E.B.E-3

Konkrete Elemente dürfen nicht zirkulär über *Aggregationen* oder *Kompositionen* verbunden sein. D.h. ein Element darf weder sich selbst noch ein anderes *Unterelement* besitzen, das direkt oder über Beziehungen zu anderen Elementen bereits sein *Gruppenelement* ist.

Bei allen *Aggregationen* und *Kompositionen* können *Unterelemente* in einem konkreten Modell in ihrer Darstellung und inhaltlichen Interpretation ausgeblendet werden. Dabei werden alle Verbindungseigenschaften der *Unterelemente* für die Dauer des Ausblendens den *Gruppenelementen* zugerechnet. Dieses Vorgehen entspricht einer Generalisierung oder *Aggregation* auf Modellebene, da im jeweiligen Kontext nicht relevante Detailinformation verborgen werden kann. In diesem Dokument wird dieses Ausblenden der *Unterelemente* als *Vergrößern* und das Wiedereinblenden der *Unterelemente* als *Verfeinern* bezeichnet.

C.E Bedingungen

Eine *Bedingung* ist eine Zusicherung bzw. Integritätsregel. In einem metamodellkonformen Modell müssen alle *Bedingungen* eingehalten werden. Zur besseren Lesbarkeit sind alle *Bedingungen* in 3LGM² semiformal spezifiziert.

D Umsetzung des Drei-Ebenen-Metamodells im 3LGM²-Baukasten

Mit der aktuell verfügbaren Software „3LGM²-Baukasten“ lassen sich in Bezug auf das in [8] veröffentlichte Metamodell 3LGM² V1.01 3LGM²-konforme Modelle von Informationssystemen erstellen. Da in den folgenden Abschnitten häufig auf spezifische Anforderungen an das Modellierungswerkzeug zur Erstellung von Modellen mit dem neuen, in diesem Dokument beschriebenen Metamodell eingegangen wird, soll hier kurz auf die wichtigsten Bestandteile des bestehenden 3LGM²-Baukastens eingegangen werden (siehe Abbildung 4):

- Der *Modellbrowser*: Im *Modellbrowser* sieht man die momentan geöffneten Modelle sowie deren Elemente. Man kann sich wahlweise alle Elemente eines Modells oder die Elemente der einzelnen Teilmodelle eines Modells anzeigen lassen. Die Elemente sind unterteilt in Elemente der *Fachlichen Ebene*, der *Logischen Werkzeugebene* und der *Physischen Werkzeugebene*.
- Das *Dokumentfenster*: Im *Dokumentfenster* werden die grafischen Sichten des Modells dargestellt. Für jede Gegenstandsebene gibt es jeweils eine sogenannte Einzelebenenansicht, in der eine Auswahl der Konzepte der jeweiligen Gegenstandsebene grafisch visualisiert wird. Es ist ebenfalls möglich, alle drei Ebenen sowie die wichtigsten Beziehungen zwischen Elementen unterschiedlicher Ebenen im *Dokumentfenster* anzuzeigen.
- Die *Menüleiste*: Am oberen Rand des Fensters befindet sich die *Menüleiste*, die sämtliche im jeweiligen Kontext sinnvollen Funktionen zur Verfügung stellt.
- Die *Standardsymbolleiste*: Unter der *Menüleiste* befindet sich die *Standardsymbolleiste*. Sie enthält einige der am häufigsten verwendeten Funktionen des 3LGM²-Baukastens. Unter anderem kann man über diese Symbolleiste für ein Modell zwischen Einzelebenenansicht und Mehrebenenansicht umschalten.
- Die *Zeichensymbolleiste* befindet sich am unteren Rand links. Sie enthält die in der ausgewählten Ebene verfügbaren Zeichenobjekte. Da je nach ausgewählter Ebene verschiedene Objekte gezeichnet werden können, unterscheidet sich auch das Aussehen der *Zeichensymbolleiste* je nach gewählter Ebene.
- Die *Ansichtswerkzeuge*: Die *Ansichtswerkzeuge* dienen dazu, die Ansicht des *Dokumentfensters* je nach Bedarf anzupassen. Ist die Ansicht der Ebenen im *Dokumentfenster* auf Einzelansicht der Ebenen eingestellt, so kann nur der Zoomfaktor verändert werden. Bei Ansicht aller drei Ebenen im *Dokumentfenster*, können Zoomfaktor, Ebenenwinkel und Ebenenabstand verändert werden.

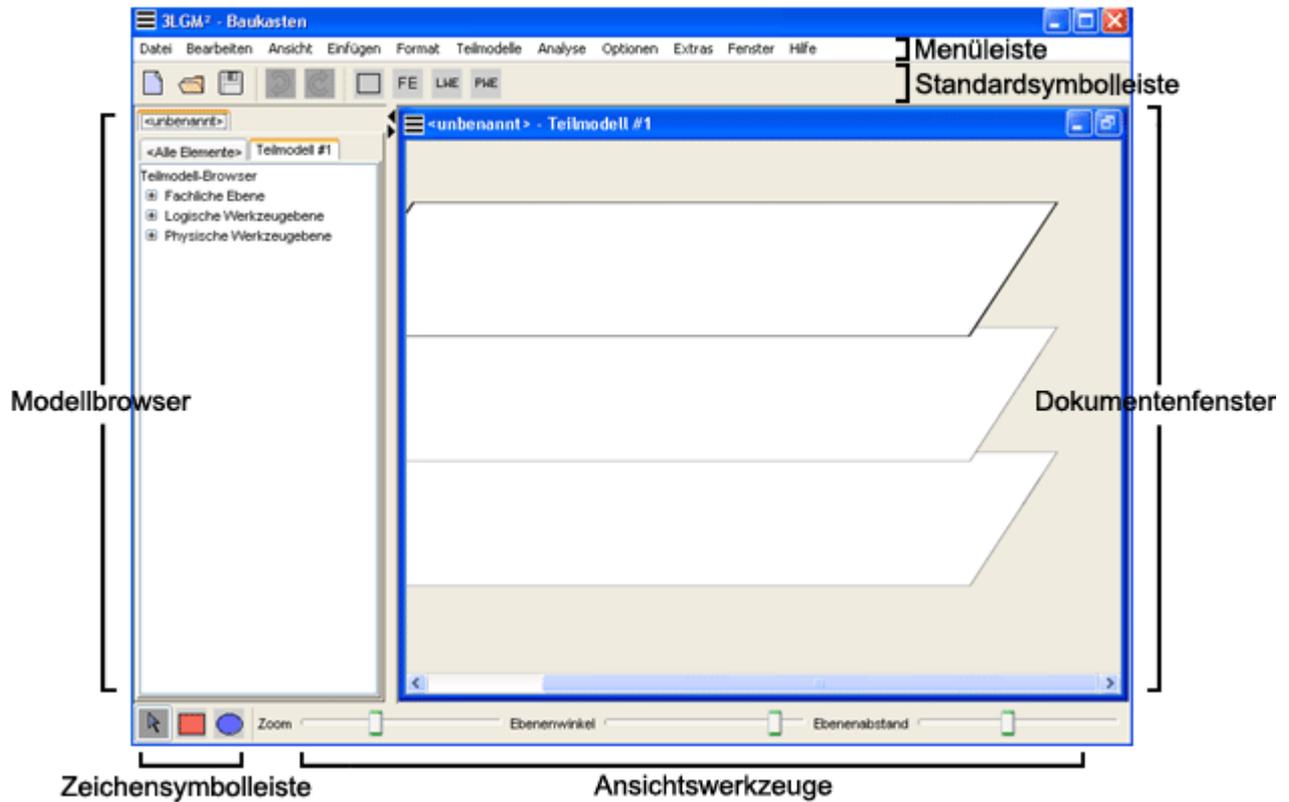


Abbildung 4: Baukastenübersicht

Fast alle in diesem Dokument enthaltenen Abbildungen wurden mit dem 3LGM²-Baukasten in der derzeit verfügbaren Version 3.2.x erstellt. Eventuelle grafische Abweichungen von späteren Baukasten-Versionen sind möglich. Modellelemente, die zum Zeitpunkt der Fertigstellung dieses Reports noch nicht im 3LGM²-Baukasten verfügbar waren, wurden mit anderen Grafikwerkzeugen beispielhaft dargestellt.

Eine detailliertere Beschreibung, insbesondere der verschiedenen Sichten auf ein Informationssystem, die der 3LGM²-Baukastens anbietet, ist in Abschnitt A gegeben.

E Das Drei-Ebenen-Metamodell 3LGM² zur Beschreibung von Informationssystemen

E.A Basis, Erweiterungen und Modifikationen des Metamodells

Ab der hier dargestellten Version 3 hat das Drei-Ebenen-Metamodell einen modularen Aufbau, bestehend aus 3LGM²-B und den Erweiterungen und Modifikationen 3LGM²-M und 3LGM²-S.

Die Basis für die 3LGM²-Modellierung und das Verständnis von 3LGM²-M und 3LGM²-S bildet das 3LGM²-B, welches die wichtigsten Elementklassen für die Modellierung von Informationssystemen auf den drei Gegenstandsebenen *Fachliche Ebene*, *Logische Werkzeugebene* und *Physische Werkzeugebene* enthält. Die Erweiterung 3LGM²-M bezieht sich auf die *Logische Werkzeugebene* und erweitert 3LGM²-B um Elementklassen zur Modellierung der nachrichtenorientierten Kommunikation. Während im 3LGM²-B der Austausch von Daten zwischen verschiedenen *Anwendungsbausteinen* nur grob beschrieben wird, kann die Kommunikation zwischen *Anwendungsbausteinen* auf der Basis von *Nachrichtentypen* und *Kommunikationsstandards* in einem 3LGM²-M-Modell ausführlich beschrieben werden. Eine Konvertierung eines Modells von 3LGM²-M nach 3LGM²-B ist demnach nicht ohne Informationsverlust möglich und lässt sich nicht rückgängig machen. Wiederum können nur Modelle auf der Basis von 3LGM²-M, d. h. Modelle in denen bereits *Nachrichtentypen* modelliert wurden, automatisch zu 3LGM²-S – Modellen erweitert werden.

3LGM²-S ist hilfreich für die Modellierung dienstorientierter (service-orientierter) Kommunikation, wie sie zum Beispiel für SOA (Service-Oriented Architectures) relevant ist. Mit 3LGM²-S ist es möglich, ein Informationssystem zu beschreiben, in welchem *Anwendungsbausteine* spezielle *Services* anbieten, die von anderen *Anwendungsbausteinen* genutzt werden können. Modelle auf der Basis von 3LGM²-B können manuell zu 3LGM²-S-Modellen erweitert werden. Eine Rück-Konvertierung von 3LGM²-S-Modellen zu 3LGM²-M oder 3LGM²-B ist prinzipiell möglich, allerdings nicht verlustlos.

E.B Die Basis des Drei-Ebenen-Metamodells: 3LGM²-B

E.B.A Überblick

Das Drei-Ebenen-Metamodell 3LGM²-B ist mit den in Abschnitt A erläuterten UML-Konzepten dargestellt. Alle dort verwendeten *Klassen* bzw. Begriffe werden in den folgenden Abschnitten näher erläutert. Die Zuordnung der Konzepte zu den drei Gegenstandsebenen erfolgt anhand derselben Farbcodierung, wie sie in Abbildung 5 zu sehen ist. D. h. die Begriffe der *Fachlichen Ebene* sind rot, die der *Logischen Werkzeugebene* grün und alle Konzepte der *Physischen Werkzeugebene* sind blau. Grau dargestellte Klassen liegen zwischen 2 Gegenstandsebenen.

E.B.B Fachliche Ebene

Die *Fachliche Ebene* eines *Informationssystems* ergibt sich in erster Linie aus den Daten, die im Rahmen der Erledigung von *Aufgaben* jeweils erzeugt bzw. benötigt werden. Die Aufgaben sind durch die *Organisation* vorgegeben und deren Erledigung muss durch das *Informationssystem* adäquat unterstützt werden. Die *Fachliche Ebene* beschreibt damit die fachlichen Konsequenzen aus den Unternehmenszielen und abstrahiert dabei von den Werkzeugen, die zur Erledigung der *Aufgaben* eingesetzt werden, und der Form und dem Ort in bzw. an dem die Daten repräsentiert sind.

Anmerkung: Auf der *Fachlichen Ebene* geht man von der Vorstellung aus, dass sich die erforderlichen Daten alle in der betrachteten Welt befinden. Das macht die Vorstellung überflüssig, man müsse auf der *Fachlichen Ebene* Daten zwischen *Aufgaben* ‚fließen‘ lassen. Es muss nur dargestellt werden, bei der Erledigung welcher *Aufgabe* ein Datum erzeugt wird, das bei der Erledigung welcher (anderen) *Aufgabe* möglicherweise benötigt wird.

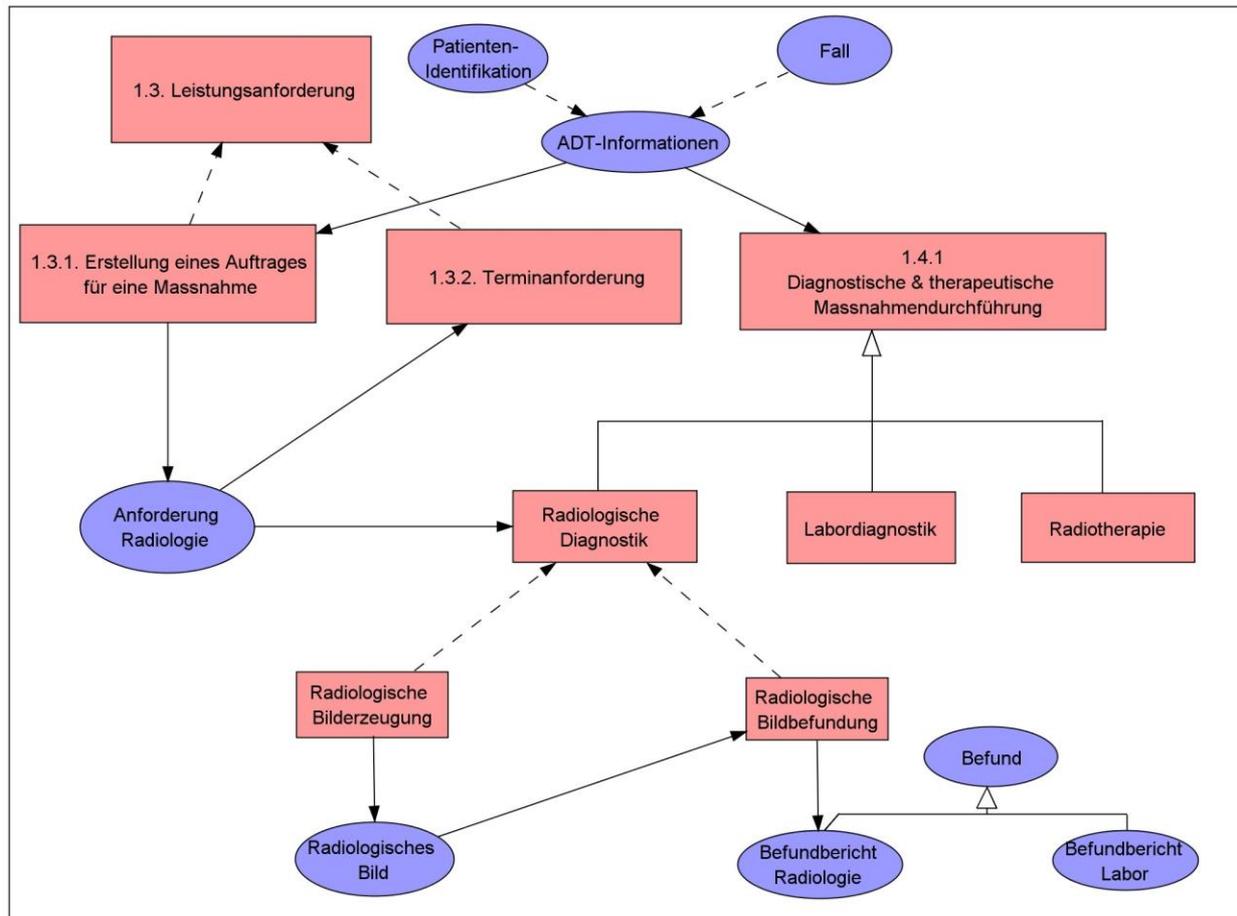


Abbildung 6: Beispiel der grafischen Sicht einer *Fachlichen Ebene*.

(Rechtecke repräsentieren *Aufgaben*, Ovale stehen für *Objekttypen*. Ein Pfeil von einem *Objekttyp* zu einer *Aufgabe* markiert einen interpretierenden Zugriff, ein Pfeil von einer *Aufgabe* zu einem *Objekttyp* einen bearbeitenden Zugriff. *Generalisierungen* bzw. *Aggregationen* sind durch gestrichelte Pfeile dargestellt, die von dem *Unterelement* ausgehen und auf das übergeordnete Element hinweisen. Die konkrete Bedeutung wird in E.B.B-1 und E.B.B-2 näher erläutert.)

Auf der *Fachlichen Ebene* kann also beschrieben werden, welche *Aufgaben* erledigt werden müssen und welche *Objekttypen* dabei jeweils bearbeitet und interpretiert werden. Somit wird auf dieser Ebene letztlich formuliert, welche Anforderungen von *datenverarbeitenden Werkzeugen* in einer *Organisation* erfüllt werden müssen, damit die Informationsverarbeitung in dieser *Organisation* tatsächlich ‚funktioniert‘.

Außerdem lassen sich auf dieser Gegenstandsebene die Unternehmensstruktur über *Organisationseinheiten* und deren Beziehung zu den *Aufgaben* darstellen sowie Prozessabläufe über Folgen von Aufgaben beschreiben.

E.B.B-1 Objekttyp

E.B.B-1.1 Beschreibung des Konzepts²

Ein Informationssystem verarbeitet Daten. Die Daten repräsentieren Informationen über physische oder virtuelle Objekte (engl. objects, vgl. „part of the perceivable or conceivable universe“ [14]). Gleichartige Dinge werden in *Objektklassen*³ zusammengefasst. In einem Krankenhaus gibt es zum Beispiel folgende Objektklassen: Patient, Pflegekraft, Ärztin/Arzt, Befund, durchgeführte Operation.

Definition: Objekttyp

Ein *Objekttyp* (engl. „entity type“) repräsentiert

1. eine Objektklasse und zeigt an, dass die repräsentierte Klasse von Objekten für das Informationssystem interessant ist und dort grundsätzlich Daten⁴, die Informationen über Objekte dieser Klasse repräsentieren, gespeichert werden können sowie
2. die Daten, die Informationen über Objekte dieser Objektklasse repräsentieren, und in dem Informationssystem gespeichert werden bzw. gespeichert werden können oder sollen..

Bei der Modellierung erhält ein *Objekttyp* als Bezeichnung die Bezeichnung der *Objektklasse*, auf die er verweist. Zur Vereinfachung sagen wir „*Daten eines Objekttyps*“ wenn wir die von dem *Objekttyp* repräsentierten Daten meinen.

Zum Beispiel repräsentiert der *Objekttyp* „Befund“ (siehe Abbildung 6) sowohl

1. die Klasse aller Befunde eines Krankenhauses und zeigt an, dass Befunde für das Informationssystem des Krankenhauses interessant sind und dort grundsätzlich Befunddaten gespeichert werden können sowie
2. die Befunddaten, die in diesem Informationssystem gespeichert werden bzw. gespeichert werden können oder sollen.

Weitere für Krankenhäuser typische *Objekttypen* werden in [15] beschrieben.

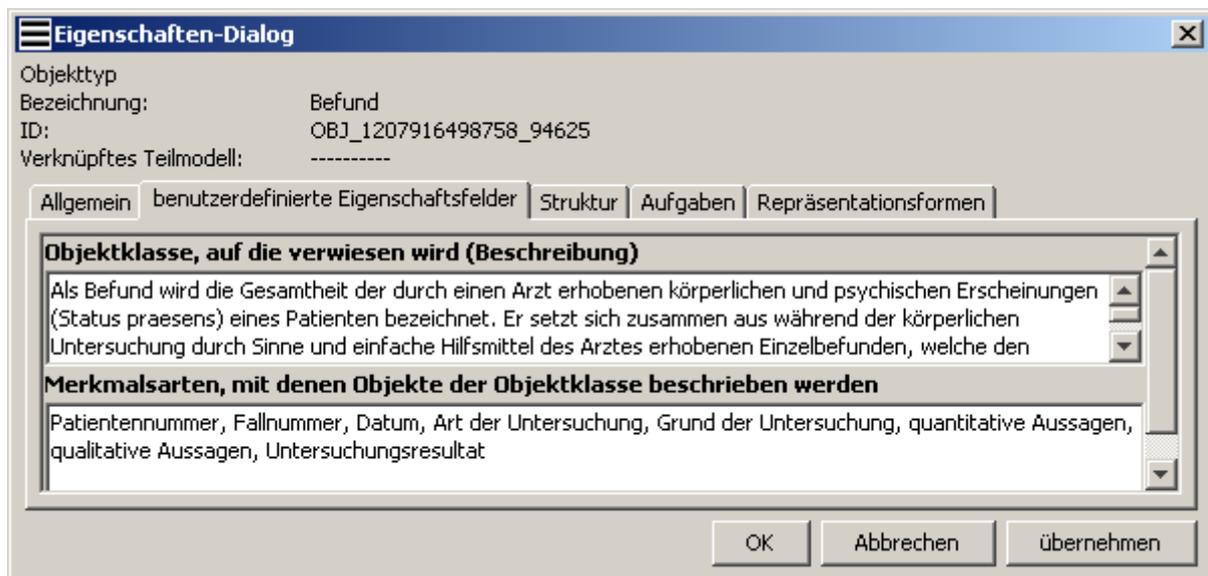


Abbildung 7: Beispiel eines *Objekttyps*

² Vgl. ([13], S. 67, 70)

³ vgl. „concept“ als „unit of thought constituted through abstraction on the basis of properties common to a set of objects“[14]).

⁴ In ([13], S. 67f) werden Daten als Explikationen sogenannter Informationsobjekte bezeichnet. Ein Informationsobjekt ist dort „eine abstrakte, aussagekräftige Repräsentation von Feststellungen über ein einzelnes, konkrete, benennbares Objekt“. Wir verzichten hier auf den Begriff des Informationsobjekts und sprechen unmittelbar von Daten als Beschreibungen von Objekten.

E.B.B-1.2 Spezielle Formen von Objekttypen

Eine spezielle Form von *Objekttypen* sind *Begriffssysteme*. Siehe E.B.B-3.

E.B.B-1.3 Spezialisierung und Zerlegung von Objekttypen

E.B.B-1.3.1 Konzept

Objekttypen können sowohl durch Spezialisierung als auch durch Zerlegung in *Unterojekttypen* gegliedert werden.

- Bei einer *Spezialisierung* kann ein *Gruppenobjekttyp* (leicht) unterschiedliche Ausprägungen haben, die ihm in Form seiner *Unterojekttypen* untergeordnet werden. Die *Unterojekttypen* werden über das UML-Konzept der *Aggregation* dem *Gruppenobjekttyp* untergeordnet. Jeder *Objekttyp* kann über diese Beziehung genau einem anderen *Objekttyp* untergeordnet sein.
- Bei einer *Zerlegung* versteht man die *Unterojekttypen* als echten Teil des *Gruppenobjekttyps*. Sie werden über das UML-Konzept der *Komposition* dem *Gruppenobjekttyp* untergeordnet. Jeder *Objekttyp* kann über diese Beziehung beliebig vielen anderen *Objekttypen* untergeordnet sein.

Abbildung 6 zeigt Beispiele für Spezialisierung und Zerlegungen von Objekttypen:

In Abbildung 6 ist der *Objekttyp* „ADT Information“ mit den *Unterojekttypen* „Patientenstamminformation“ und „Fall“ Beispiel für eine *Zerlegung*, wohingegen „Befund“ durch „Befundbericht Radiologie“ und „Befundbericht Labor“ spezialisiert wird.

E.B.B-1.3.2 Vererbung

Eine Vererbung drückt aus, dass bei *Spezialisierung* und *Zerlegung* alle Eigenschaften des *Gruppenobjekttyps* auch für die *Unterojekttypen* als gegeben angesehen werden sollen, die Bestandteile dieses *Gruppenobjekttyps* sind, d. h. Eigenschaften und *Assoziationen* werden an die *Unterojekttypen* vererbt. Dies bedeutet, dass alle Beziehungen zu *Aufgaben*, *Anwendungsbausteinen*, *Begriffssystemen*, *Sendeschnittstellen*, *Empfangsschnittstellen* und zu *Kommunikationsverbindungen*, die für den *Gruppenobjekttyp* modelliert sind, auch als für alle seine *Unterojekttypen* modelliert gelten, also vererbt werden.

Entsprechend ist in Abbildung 6 aus der Tatsache, dass die *Aufgabe* „1.4.1 Diagnostische und therapeutische Maßnahmedurchführung“ eine Beziehung zum *Objekttyp* „ADT Informationen“ hat, abzuleiten, dass dessen *Unterojekttypen* „Patientenstamminformation“ und „Fall“ ebenfalls diese Beziehung zu der Aufgabe besitzen. Die verschiedenen Ausprägungen des spezialisierten *Objekttyps* „Befund“ müssen jedoch nicht gemeinsame Beziehungen zu *Aufgaben* haben. Somit ist es häufig nur bei *Objekttypen* mit *Kompositionsbeziehung* sinnvoll, dem *Gruppenobjekttyp* Beziehungen zu anderen Elementklassen zuzuordnen.

E.B.B-1.3.3 Vergrößern

Bei dem Vergrößern eines *Gruppenobjekttyps* werden seine *Unterojekttypen* (vorübergehend) aus dem Modell entfernt. Dies bedeutet, dass auf die differenzierte Betrachtung, die durch die *Unterojekttypen* gegeben ist, verzichtet wird. Die *Assoziationen* der *Unterojekttypen* zu *Aufgaben*, *Anwendungsbausteinen*, *Sendeschnittstellen*, *Empfangsschnittstellen* und zu *Kommunikationsverbindungen* werden dann für die Dauer der Vergrößerung vollständig dem *Gruppenobjekttyp* zugeordnet.

E.B.B-1.4 Beziehungen „wird interpretiert“ und „wird bearbeitet“ zu Aufgabe

Objekttypen können von *Aufgaben* interpretiert und bearbeitet werden. Dies wird in E.B.B-2.1 und E.B.B-2.3 näher erläutert.

Zur Kardinalität der Beziehung siehe auch E.B.B-2.7.

Im Beispiel in Abbildung 6 wird die interpretierende Beziehung durch einen Pfeil vom *Objekttyp* zur *Aufgabe* und die bearbeitende Beziehung durch einen Pfeil von der *Aufgabe* zum *Objekttyp* dargestellt.

E.B.B-1.5 Interebenenbeziehung „wird gespeichert von“ zu Anwendungsbaustein

Die Daten über *Objekte*, auf die ein bestimmter *Objekttyp* verweist, werden durch *Anwendungsbausteine* gespeichert. Nur in *Anwendungsbausteinen*, die für diesen *Objekttyp* als *Master* gekennzeichnet werden, dürfen diese Daten erzeugt oder geändert werden; in anderen *Anwendungsbausteinen*, die Daten über diese *Objekte* speichern, dürfen diese Daten nur gelesen werden.

E.B.B-1.6 Interebenenbeziehung „wird gesendet von“ zu Sendeschnittstelle, „wird empfangen von“ zu Empfangsschnittstelle und „wird übertragen von“ zu Kommunikationsverbindung

Werden die Daten über *Objekte*, auf die ein bestimmter *Objektyp* verweist, in mehreren *Anwendungsbausteinen* gespeichert, dann müssen diese Daten zwischen diesen *Anwendungsbausteinen* auch ausgetauscht, d. h. kommuniziert werden. Darüber hinaus werden die Daten ggf. auch in *Anwendungsbausteinen* benötigt, in denen sie nicht gespeichert werden. Dies kann zum Beispiel dann der Fall sein, wenn ein solcher *Anwendungsbaustein* eine *Aufgabe* unterstützt, die den auf die Daten verweisenden *Objektyp* interpretiert. Zwischen den *Anwendungsbausteinen* werden die Daten, auf die ein bestimmter *Objektyp* verweist, kommuniziert.

Die *Assoziation* eines *Objektyps* zu einer *Sendeschnittstelle* beschreibt, dass diese *Sendeschnittstelle* Daten versenden kann, die Informationen über *Objekte* repräsentieren, welche durch den *Objektyp* beschrieben sind. Vereinfachend kann davon gesprochen werden, dass die *Sendeschnittstelle* den *Objektyp* versenden kann.

Die *Assoziation* eines *Objektyps* zu einer *Empfangsschnittstelle* beschreibt, dass diese *Empfangsschnittstelle* Daten empfangen kann, die Informationen über *Objekte* repräsentieren, welche durch den *Objektyp* beschrieben sind. Vereinfachend kann davon gesprochen werden, dass die *Empfangsschnittstelle* den *Objektyp* empfangen kann.

Die *Assoziation* eines *Objektyps* zu einer *Kommunikationsverbindung* beschreibt, dass diese *Kommunikationsverbindung* Daten transportiert, die Informationen über *Objekte* repräsentieren, welche durch den *Objektyp* beschrieben sind. Vereinfachend kann davon gesprochen werden, dass die *Kommunikationsverbindung* den *Objektyp* transportiert.

E.B.B-1.7 Fragen des Informationsmanagements und Unterstützung durch den 3LGM²-Baukasten

Besonders interessante Fragen in Bezug auf *Objektypen*, die für das Management von Informationssystemen relevant sind, sind in der folgenden Tabelle zusammen mit den Hinweisen auf die Stellen, wo diese Fragen näher behandelt werden, aufgeführt.

Fragen des Informationsmanagements	Verweis auf Text
Welche Aufgaben sind auf Daten eines bestimmten Objektyps angewiesen?	E.B.B-2.3
Bei der Erledigung welcher Aufgaben werden die Daten eines bestimmten Objektyps erzeugt?	E.B.B-2.3
Wo werden die Daten eines bestimmten Objektyps gespeichert?	E.B.B-1.5, E.B.D-1.8
Zwischen welchen Anwendungsbausteinen werden die Daten eines bestimmten Objektyps kommuniziert?	E.B.B-1.6, E.B.D-3
Gibt es ausreichende Kommunikationsmöglichkeiten, um die Daten eines bestimmten Objektyps zwischen den Anwendungsbausteinen zu kommunizieren, in denen sie gespeichert werden?	E.B.B-1.6, E.B.D-3
Welche Datendomäne hat ein Objektyp, d. h. welche Anwendungsbausteine benötigen den Objektyp?	[16]
Gibt es ausreichende Kommunikationsmöglichkeiten, um die Daten eines bestimmten Objektyps von den Anwendungsbausteinen, in denen sie gespeichert werden, zu den Anwendungsbausteinen zu kommunizieren, in denen sie zur Erledigung von Aufgaben benötigt werden?	[17]

E.B.B-1.8 Modellierungshinweise

- Bei der Bezeichnung eines *Objektyps* ist darauf zu achten, dass diese Bezeichnung auch für die Instanzen der Objektklasse verwendet werden kann, auf die der *Objektyp* verweist. Insbesondere sollte daher die jeweilige Singularform gewählt werden. So sollte ein *Objektyp*, der auf die Objektklasse der Patienten eines Krankenhauses verweist, mit „Patient“ und nicht mit „Patienten“ bezeichnet werden. In Anwendung dieses Prinzips sollte auch ein *Objektyp*, der auf den ICD-10 Katalog der Diagnosenklassen verweist, d. h. ein *Begriffssystem* (siehe E.B.B-3), z. B. als „ICD-10-Diagnoseklasse“ bezeichnet werden. Bei einem *Objektyp* „ICD-10-Katalog“ würde dagegen davon ausgegangen werden, dass jede Instanz ein ICD-10 Katalog ist und mit diesem *Objektyp* eine Menge von ICD-Katalogen, z. B. unterschiedlicher Versionen, gemeint ist.
- Bei der Modellierung eines *Objektyps* sollte immer beschrieben werden, was genau die *Objektklasse* ist, auf die er verweist und mit welchen *Merkmalsarten* die *Objekte* beschrieben werden. Ferner ist es hilfreich, sich

stets zu vergegenwärtigen, dass ein *Objektyp* stets Daten über die bzw. zur Beschreibung der *Objekte* der *Objektklasse* meint; nie jedoch die Objekte selbst.

- Jeder *Objektyp* sollte von mindestens einer *Aufgabe* interpretiert werden. Ein *Objektyp*, der von keiner *Aufgabe* interpretiert wird, ist in einem Modell möglicherweise überflüssig.
- Das Vererbungsprinzip ist stets sorgfältig zu beachten. So ist stets zu beachten, dass z. B. eine Beziehung die ein *Gruppenobjektyp* mit einer *Aufgabe* hat, auch für alle *Unterojekttypen* gegeben ist. Dies gilt auch dann, wenn einem *Gruppenobjektyp* erst später *Unterojekttypen* zugeordnet werden, also der *Objektyp* zum *Gruppenobjektyp* wird. Es wird nicht immer davon auszugehen sein, dass alle Eigenschaften des *Gruppenobjektyps* auch für alle *Unterojekttypen* gelten. Es sollte also immer überprüft werden, ob wirklich nur die Eigenschaften bei dem neuen *Gruppenobjektyp* belassen worden sind, die auch wirklich für alle *Unterojekttypen* gelten sollen. Diese Situation ergibt sich z. B., wenn *Objektypen* eines Referenzmodells in ein Modell übernommen, dort aber zur präziseren Modellierung aufgeteilt werden. Grundsätzlich sollte also vermieden werden, *Gruppenobjektypen* Beziehungen zuzuweisen. Zur Unterstützung der Modellierung bietet der 3LGM²-Baukasten eine Funktion, mit der alle oder eine Auswahl der vererbten Eigenschaften und Beziehungen eines *Gruppenobjektyps* explizit auf alle *Unterojekttypen* übertragen und bei dem *Gruppenobjektyp* entfernt werden.
- Ein *Gruppenobjektyp* sollte mindestens zwei *Unterojekttypen* haben. Von dieser Regel kann z. B. abgewichen werden, wenn es sich bei dem *Gruppenobjektyp* um einen *Elementarobjektyp*, d. h. um einen *Objektypen* ohne weitere *Unterelemente*, aus einem Referenzmodell handelt, aber in dem Modell eine detailliertere Aussage nötig erscheint. In diesem Fall sind wiederum die Hinweise zur Vererbung zu beachten.
- Es sollten folgende Modellierungsregeln für *Objektypen* beachtet werden, insbesondere bei Verwendung des Phasenmodells für die Modellierung von *Aufgaben* und *Objektypen* (vgl. E.B.B-2.7).
 - Jeder *Objektyp* muss genau eine *Klasse* darstellen.
 - Eine *Klasse* darf nicht von mehr als einem *Objektyp* dargestellt werden.
 - Die Bezeichnung eines *Objektyps* sollte aus einem domänenspezifischen, standardisierten Vokabular (z. B. einem Glossar, einem Fachwörterbuch, einem Standard oder einer Nomenklatur) entnommen sein.
 - Die Bezeichnung eines *Objektyps* muss ein Substantiv enthalten, das als Bezeichnung für jedes einzelne der Objekte der vom *Objektyp* dargestellten *Klasse* geeignet ist.
 - Die Bezeichnung eines *Objektyps* darf nicht geeignet sein, andere *Objekte* als die *Objekte* der vom *Objektyp* dargestellten *Klasse* zu bezeichnen.
 - Die Bezeichnung muss so kurz wie möglich gewählt sein.
 - Zu jedem *Objektyp* sollte die Definition der von diesem *Objektyp* dargestellten *Klasse* dokumentiert sein.
 - Die Definition zu einem *Objektyp* sollte nicht frei erstellt, sondern unter Angabe der Quelle einem zitierfähigen Dokument entnommen sein.
 - Für jeden *Objektyp* muss ein Identitätsprinzip angegeben werden.
 - Für *Objektypen*, die keine ein Identitätsprinzip spendende *Klasse* darstellen, kann das Identitätsprinzip angegeben werden, indem die *Klasse* angegeben wird, von dem die vom *Objektyp* dargestellte *Klasse* ihr Identitätsprinzip bezieht.

Für nähere Erläuterungen der Modellierungsregeln siehe [13].

E.B.B-2 Aufgabe

E.B.B-2.1 Beschreibung des Konzepts

Aktivitäten im Rahmen der datenverarbeitenden Prozesse eines Informationssystems verändern oder löschen bestimmte Daten und sind dabei auf bestimmte schon vorhandene Daten angewiesen. Mit einem *Objektyp* können wir alle Daten über *Objekte* einer bestimmten Klasse repräsentieren (siehe Definition in E.B.B-1.1). Das wiederum ermöglicht es, auch gleichartige *Aktivitäten* zusammenzufassen:

Wir können solche *Aktivitäten* als gleichartig zusammenzufassen, welche die *Daten* derselben *Objekttypen* verändern oder löschen, und gleichzeitig dabei auf die *Daten* derselben *Objekttypen* angewiesen sind. Eine so definierte Klasse gleichartiger *Aktivitäten* bezeichnen wir als *datenverarbeitende Aufgabe*. Eine *datenverarbeitende Aufgabe* wird folglich durch die *Objekttypen* gekennzeichnet, deren Daten ihre Aktivitäten verarbeiten bzw. auf die sie angewiesen sind.

In der wirtschaftswissenschaftlichen Literatur ist eine Unternehmensaufgabe [18] (engl. "enterprise function" [19]) nach [20], S. 147, eine Zielvorschrift für menschliches oder maschinelles Handeln. Das durch das Handeln zu erreichende Ziel deckt sich mit einem oder unterstützt die Erreichung eines (Teil-) Ziels einer *Organisation*.

Vor diesem Hintergrund sind *datenverarbeitende Aufgaben* stets Teilaufgaben einer Unternehmensaufgabe. Die erwähnte Zielvorschrift ist für *datenverarbeitende Aufgaben* als eine Vorschrift zu verstehen, wie Daten bestimmter *Objekttypen* verändert oder gelöscht werden sollen.

Definition: Aufgabe

Eine *datenverarbeitende Aufgabe* (kurz: *Aufgabe*) ist eine Zielvorschrift, wie auf Grund von Daten bestimmter *Objekttypen* Daten anderer *Objekttypen* verändert oder gelöscht werden sollen. Das durch die Veränderung oder Löschung zu erreichende Ziel deckt sich mit einem oder unterstützt die Erreichung eines (Teil-) Ziels einer *Organisation*. Eine *Aufgabe* hat keinen definierten Anfang und kein definiertes Ende.

Man kann eine *Aufgabe* als Pflicht auffassen (siehe oben „Zielvorschrift“), die sich aus den Zielen des Unternehmens mehr oder weniger zwingend ergibt. Die Zielvorschrift wird durch konkretes Handeln erfüllt. Das konkrete Handeln einer menschlichen Person oder einer Maschine in einem bestimmten Zeitraum zur einmaligen Erledigung einer *Aufgabe* ist eine *Aktivität*; eine *Aktivität* kann als Instanz einer *Aufgabe* aufgefasst werden.

Aktivitäten finden nicht nur wie eingangs erläutert, im Rahmen eines Prozesses statt, sondern durchlaufen bei Ihrer Erledigung selber wieder *Prozesse*. Daher kann eine *Aufgabe* auch als Repräsentant der *Prozesse* aufgefasst werden, die bei der Erledigung seiner *Aktivitäten* durchlaufen werden.

Als Bezeichnungen für *Aufgaben* werden Substantive bzw. substantivierte Verben gewählt.

Typische, in einem Krankenhaus zu erledigende *Aufgaben* sind z.B. Patientenaufnahme, Termin- und Ressourcenplanung, diagnostische und therapeutische Maßnahmendurchführung. Weitere für Krankenhäuser typische *Aufgaben* werden in [15] beschrieben.

E.B.B-2.2 Spezialisierung und Zerlegung von Aufgaben

E.B.B-2.2.1 Konzept

Analog zu Objekttypen wird bei Aufgaben ebenfalls zwischen Spezialisierung und Zerlegung unterschieden:

- *Spezialisierung* bedeutet, dass die *Unteraufgabe* eine spezielle Art der *Gruppenaufgabe* darstellt bzw. die *Gruppenaufgabe* eine *Generalisierung* der *Unteraufgabe* ist. Inhaltlich bedeutet dies, dass eine *Aufgabe* in bestimmten Kontexten in (leicht) unterschiedlicher Form erledigt wird. Diese unterschiedlichen Formen der Erledigung werden durch ihre *Unteraufgaben* ausgedrückt.
- Bei einer *Zerlegung* hingegen ist die *Unteraufgabe* als echter Teil der *Gruppenaufgabe* zu verstehen, d. h. erst wenn alle *Unteraufgaben* erledigt sind, dann ist auch die *Gruppenaufgabe* vollständig erledigt. In einem solchen Fall kann davon ausgegangen werden, dass ein *Prozess* existiert, der beschreibt, in welcher Reihenfolge bzw. unter welchen Bedingungen welche der *Unteraufgaben* erledigt werden sollen (allerdings wird der Prozess hier nicht modelliert). Das heißt die *Unteraufgaben* müssten immer gemeinsam bzw. in einer bestimmten Abfolge erledigt werden und nicht, wie bei einer *Spezialisierung* alternativ.

Abbildung 6 zeigt Beispiele für *Spezialisierung* und *Zerlegung* bei Aufgaben:

- Die *Aufgabe* „1.4.1 Diagnostische und therapeutische Maßnahmendurchführung hat die *Unteraufgaben* „Radiologische Diagnostik“, „Labordiagnostik“ und „Radiotherapie“ (dargestellt durch unterbrochene Pfeile, die von den *Unteraufgaben* auf die *Gruppenaufgabe* zeigen). Es handelt sich bei der Radiologischen Diagnostik, der Labordiagnostik und der Radiotherapie um *Spezialisierungen* der diagnostischen und therapeutischen Maßnahmendurchführung, d. h. sie können alternativ durchgeführt werden.
- Die *Gruppenaufgabe* „Radiologische Diagnostik“ wurde zerlegt in zwei *Unteraufgaben* (*Zerlegung*). D. h. sowohl die „Radiologische Bilderzeugung“ als auch die „Radiologische Bildbefundung“ müssen innerhalb der Radiologischen Diagnostik erfolgen.

E.B.B-2.2.2 Vererbung

Bei *Spezialisierung* und *Zerlegung* werden alle Eigenschaften und *Assoziationen*, die der *Gruppenaufgabe* zugewiesen werden, auch für ihre *Unteraufgaben* als gegeben angesehen, d. h. die Eigenschaften werden vererbt. Dies bedeutet, dass alle Beziehungen zu *Objekttypen*, *Prozessen* und *Anwendungsbausteinen*, die für eine *Gruppenaufgabe* modelliert sind, auch als für alle ihre *Unteraufgaben* modelliert gelten.

Der Pfeil des blauen Ovals „ADT Information“ auf die übergeordnete *Aufgabe* in Abbildung 6 stellt dar, dass diese *Aufgabe* einen entsprechenden *Objektyp* interpretiert. Für alle *Unteraufgaben* gilt demnach auch, dass sie „ADT Information“ interpretieren.

E.B.B-2.2.3 Vergrößern

Bei dem Vergrößern einer *Gruppenaufgabe* werden die *Unteraufgaben* (vorübergehend) aus dem Modell entfernt (vgl. C.D). Dies bedeutet, dass dann auf die differenzierte Betrachtung, die durch die *Unteraufgaben* gegeben ist, verzichtet wird. Die *Assoziationen* der *Unteraufgaben* werden dann für die Dauer der Vergrößerung wie folgt der übergeordneten *Aufgabe* zugeordnet:

- *Objekttypen*: Alle interpretierenden/bearbeitenden *Assoziationen* der *Unteraufgaben* zu *Objekttypen* (siehe E.B.B-2.3, E.B.B-1) werden der *vergrößerten Aufgabe* zugeordnet.
- *Nutzung*: Alle *Assoziationen* der *Unteraufgaben* zu *Nutzungen* (siehe auch E.B.B-4), werden der vergrößerten *Aufgabe* zugeordnet. Das bedeutet, dass aufgrund der gewollten Vergrößerung und des Verzichts auf Differenzierung, die den jeweiligen *Unteraufgaben* zugeordneten *Anwendungsbausteine* als gleichwertig und damit alternativ zur Erledigung der *Gruppenaufgabe* oder zur Vermittlung von *Anwendungsbausteinen* zur Erledigung der *Aufgabe* (siehe E.B.B-4) einsetzbar angesehen werden.
- *Prozess*: Die Zugehörigkeit einer *Unteraufgabe* zu einem *Prozess* wird ebenfalls beim Vergrößern an ihre übergeordneten *Aufgaben* übertragen (siehe E.B.B-6).

Es bleibt zu beachten, dass ein und dieselbe *Unteraufgabe* gleichzeitig mehreren anderen *Aufgaben* untergeordnet sein kann. Solange nicht alle *Aufgaben* eines Modells, denen dieselbe *Unteraufgabe* zugeordnet ist auch gleichzeitig vergrößert sind, wird diese *Unteraufgabe* nicht inhaltlich ausgeblendet.

E.B.B-2.3 Beziehungen „interpretiert“ und „bearbeitet“ zu Objekttyp

Eine *Aufgabe* ist eine Zielvorschrift, wie auf Grund von Daten bestimmter *Objekttypen* Daten anderer *Objekttypen* verändert oder gelöscht werden sollen. Die *Objekttypen*, deren Daten als Grundlage der Veränderung oder Löschung erforderlich sind, werden als interpretierte *Objekttypen* bezeichnet. Die *Objekttypen*, deren Daten verändert oder gelöscht werden sollen, werden als bearbeitete *Objekttypen* bezeichnet.

Wenn eine *Aufgabe* mehrere *Objekttypen* interpretiert, bedeutet dies, dass bei der Erledigung der *Aufgabe* Daten aller interpretierten *Objekttypen* erforderlich sind. (siehe auch die Modellierungshinweise in E.B.B-2.7)

Wenn eine *Aufgabe* mehrere *Objekttypen* bearbeitet, bedeutet dies, dass bei der Erledigung der *Aufgaben* Daten aller bearbeiteten *Objekttypen* erzeugt bzw. verändert werden. Da dies zu Interpretationsschwierigkeiten führen kann, wird empfohlen, jeweils nur einen *Objekttypen* als bearbeitet zu modellieren (siehe auch die Modellierungshinweise in E.B.B-2.7). Ggf. ist dazu in Betracht zu ziehen, die *Aufgabe* zu *zerlegen* oder zu *spezialisieren* (siehe E.B.B-2.2.1).

In der grafischen Sicht der *Fachlichen Ebene* werden die von einer *Aufgabe* interpretierten *Objekttypen* im Baukasten durch einen Pfeil von dem *Objektyp* zur *Aufgabe* kenntlich gemacht. Der von einer *Aufgabe* bearbeitete *Objektyp* wird im Baukasten graphisch durch einen Pfeil von der *Aufgabe* zum *Objektyp* dargestellt. In Abbildung 6 *bearbeitet* die *Aufgabe* „1.3.1 Erstellung eines Auftrags für ein Maßnahme“ den *Objektyp* „Anforderung Radiologie“. Dies bedeutet, dass bei der Erledigung der Auftragserstellung Daten entstehen, die Informationen über Anforderungen an die Radiologie repräsentieren. Die *Aufgabe* „Radiologische Diagnostik“ interpretiert wiederum den *Objektyp* „Anforderung Radiologie“. Dies bedeutet, dass bei der Erledigung dieser *Aufgabe* Daten, die Informationen über Anforderungen für die Radiologie repräsentieren, in andere Daten transformiert werden. Kurz gesagt wird also bei der Erstellung eines Auftrags für eine Maßnahme eine Anforderung erstellt, die dann bei der Radiologischen Diagnostik erfüllt wird.

E.B.B-2.4 Beziehung „wird erledigt über“ zu Nutzung

Durch die Beziehung zur *Klasse Nutzung* wird ausgedrückt, dass zur Erledigung einer bestimmten *Aufgabe* in bestimmten *Organisationseinheiten* jeweils bestimmte *Anwendungsbausteine* genutzt werden (oder kurz: die *Aufgabe*

wird durch *Nutzung* eines *Anwendungsbausteins* erledigt). Die Assoziation zwischen *Nutzung* und *Anwendungsbausteinen* ist über die *Assoziationsklasse Unterstützung* näher bestimmt, wodurch zusätzlich ausgedrückt werden kann, dass die Funktionalität eines *Anwendungssystem*, das für die Erledigung einer *Aufgabe* genutzt wird, durch ein anderes *Anwendungssystem* vermittelt wird (vgl. E.B.B-4).

Kann eine *Aufgabe* innerhalb einer *Organisationseinheit* oder auch in verschiedenen *Organisationseinheiten* durch mehrere *Anwendungsbausteine* unterstützt (oder zusätzlich auch vermittelt) werden, dann wird davon ausgegangen, dass jeder dieser *Anwendungsbausteine* die Erledigung der *Aufgabe* in gleicher Weise und Qualität unterstützt (bzw. einen *Anwendungsbaustein* zur Erledigung vermittelt); folglich könnten alle unterstützenden *Anwendungsbausteine* gegeneinander und alle vermittelnden *Anwendungsbausteine* auch jeweils untereinander ohne Verlust an Funktionalität wechselseitig ausgetauscht werden.

E.B.B-2.4.1 Kompatibilität zu 3LGM² V1.x

Diese Beziehung ersetzt die *Anwendungsbausteinkonfiguration*.

E.B.B-2.5 Interebenenbeziehung „kann unterstützt werden durch“ zu Softwareprodukt

Die zuvor genannte Beziehung zu *Nutzung* drückt aus, dass zur Erledigung einer bestimmten *Aufgabe* tatsächlich in einer bestimmten *Organisationseinheit* bestimmte *Anwendungsbausteine* durch Mitarbeiter der *Organisation* genutzt werden können. Demgegenüber drückt die Beziehung zu *Softwareprodukt* aus, dass die zu der *Aufgabe* in Beziehung stehenden *Softwareprodukte* zur Realisierung von *Anwendungssystemen* genutzt werden könnten, mit denen die *Aufgabe* dann unterstützt werden kann.

E.B.B-2.6 Fragen des Informationsmanagements und Unterstützung durch das 3LGM²

Fragen des Informationsmanagements	Verweis auf Text
Welche Daten werden für die Erledigung der Aufgaben benötigt und welche Daten entstehen dabei?	E.B.B-1.4, E.B.B-2.3
Welche datenverarbeitenden Werkzeuge werden in der Organisation eingesetzt, um die Erledigung der Aufgaben zu unterstützen? Welche funktionale Domäne hat eine Aufgabe, d. h. welche Anwendungsbausteine werden benötigt, um die Aufgabe zu erledigen?	E.B.C, [16]
Für die Erledigung welcher Aufgaben werden in der Organisation moderne rechnerbasierte Werkzeuge genutzt? Für welche Aufgaben nicht?	E.B.C, E.B.D-1
Ist das Informationssystem vertikal fragmentiert? Werden für die Erledigung derselben Aufgabe in unterschiedlichen Verantwortungsbereichen der Organisation unterschiedliche Werkzeuge eingesetzt? (Vertikale Fragmentierung eines Informationssystems liegt laut [16, 21] vor, wenn verschiedene Personen in verschiedenen Organisationseinheiten verschiedene Anwendungssysteme benutzen, um verschiedene oder auch gleiche Unternehmensaufgaben zu erledigen. Das bedeutet, jede Organisationseinheit hat ihre eigene Business-Architektur, Anwendungsarchitektur und Technologie-Architektur.)	E.B.D-1, E.B.C
Warum unterscheiden wir eigentlich die verschiedenen speziellen Formen einer bestimmten Aufgabe (z. B. der Leistungsanforderung)? Das ist doch eigentlich immer dasselbe (z. B. Anforderung eines Laborbefundes oder eines pathologischen Befundes). Was wäre, wenn wir das nicht mehr unterscheiden würden; auf wie viele und welche speziellen Anwendungssysteme könnten wir dann eigentlich verzichten?	E.B.B-2.2.3, [22]
Wird eine bestimmte Aufgabe oft im Kontext anderer Aufgaben erledigt? Wenn ja, sollten die unterstützenden Anwendungssysteme möglicherweise besonders eng miteinander integriert sein?	E.B.B-2.2.3
Wenn eine Aufgabe erledigt werden soll, werden dafür bestimmte Eingabedaten benötigt, die wiederum Ausgabedaten anderer Aufgaben sind. Welche anderen Aufgaben müssen also erledigt werden, damit die betrachtete Aufgabe auch erledigt werden kann?	E.B.B-2.3
Kann die Erledigung einer bestimmten Aufgabe auch durch andere Softwareprodukte unterstützt werden, als die, die bereits im Informationssystem der Organisation im Einsatz sind?	E.B.B-2.5

E.B.B-2.7 Modellierungshinweise

- *Aufgaben* bearbeiten in der Regel (mindestens) einen *Objektyp*. Wenn ein Mensch eine *Aufgabe* erledigt, und dabei Informationen verwendet, die nicht außerhalb seiner Person im *Informationssystem* durch Daten repräsentiert sind, werden diese Informationen in der Regel nicht durch einen *Objektyp* modelliert. Auf diese Weise kann es also dazu kommen, dass es in einem Modell *Aufgaben* gibt, die keinen *Objektyp* interpretieren.
- Es kann auch sinnvoll sein, *Aufgaben* zu modellieren, bei denen es auf den Aspekt der Datenverarbeitung zunächst nicht ankommt. Solche *Aufgaben* werden auch als *Unternehmensaufgaben* bezeichnet. Solche *Unternehmensaufgaben* sind z. B. Therapie, Strategischen Rahmenplan erstellen etc. Möglicherweise wird man darauf verzichten, eine Beziehung von solchen *Aufgaben* zu *Objektypen* zu modellieren. Da es im Kontext von 3LGM² aber um die Modellierung von Informationssystemen geht, sollten in solchen Fällen datenverarbeitende Teilaufgaben der Unternehmensaufgaben definiert werden, die mit *Objektypen* in interpretierender und bearbeitender Beziehung stehen.
- Das Vererbungsprinzip ist stets sorgfältig zu beachten. So ist stets zu beachten, dass z. B. eine Beziehung, die eine *Aufgabe* mit einem *Objektyp* hat, auch für alle ihre *Unteraufgaben* gegeben ist. Dies gilt auch dann, wenn einer *Aufgabe* erst später *Unteraufgaben* zugeordnet werden. Weder bei aggregierenden noch generalisierenden *Aufgaben* wird jedoch immer davon auszugehen sein, dass alle Eigenschaften einer *Aufgabe* auch für alle ihre *Unteraufgaben* gelten. Es sollte also immer überprüft werden, ob wirklich nur die Eigenschaften bei einer *Aufgabe* belassen worden sind, die auch wirklich für alle ihre *Unteraufgaben* gelten sollen. Diese Situation ergibt sich z. B., wenn *Aufgaben* eines Referenzmodells in ein Modell übernommen, dort aber zur präziseren Modellierung weiter zerlegt oder spezialisiert werden. Grundsätzlich sollten Beziehungen zu anderen Elementen möglichst immer für *Elementaraufgaben* – also *Aufgaben* ohne eigene *Unteraufgaben* – modelliert werden.
- Eine *Aufgabe* sollte bei Zerlegung in mindestens zwei *Unteraufgaben* zerlegt werden. Von dieser Regel kann z. B. abgewichen werden, wenn es sich bei der *Gruppenaufgabe* um eine *Elementaraufgabe* aus einem Referenzmodell handelt, aber in dem Modell eine detailliertere Aussage nötig erscheint. Ebenso kann eine *Unternehmensaufgabe*, die nicht direkt *Objektypen* bearbeitet oder interpretiert, aus genau einer datenverarbeitenden *Aufgabe* bestehen, welche direkt mit *Objektypen* assoziiert ist. In solch einem Fall sind wiederum die Hinweise zur Vererbung zu beachten.
- Eine *Aufgabe* muss entweder *Elementaraufgabe*, *aggregierende Aufgabe* oder *generalisierende Aufgabe* sein. Die Konzepte der aggregierenden und generalisierenden *Aufgaben* dürfen nicht vermengt werden.
- Es gibt insbesondere zwei Wege, *Aufgaben* zu modellieren:
 - Man durchläuft folgende Phasen bei der Modellierung der *Fachlichen Ebene*:
 1. Zunächst werden die *Objektypen* identifiziert, die für das Anwendungsgebiet relevant sind.
 2. Ausgehend von den *Objektypen* werden *Aufgaben* modelliert, die die *Objektypen* bearbeiten. Dabei handelt es sich um *Aufgaben*, die Daten über die *Objektypen* erfassen, verändern oder vernichten.
 3. Es werden *Objektypen* modelliert, die von den zuvor identifizierten *Aufgaben* interpretiert werden.
 4. Schließlich werden die im zweiten Schritt identifizierten datenverarbeitenden *Aufgaben* *Unternehmensaufgaben* zugeordnet. Die *Unternehmensaufgaben* werden dazu als übergeordnete *Aufgaben* eingeführt.

Bei der Anwendung des Phasenmodells sollen folgenden Modellierungsregeln für *Aufgaben* beachtet werden:

- Eine *Aufgabe* muss genau einen *Objektyp* bearbeiten.
- Eine *Aufgabe* muss mit dem bearbeiteten *Objektyp* und der Operation bezeichnet sein, also dem Schema ›Informationen über . . . erfassen/verändern/vernichten‹ entsprechen.
- Eine *Aufgabe* darf nicht die gleiche Bezeichnung wie eine andere *Aufgabe* tragen.
- Jeder *Objektyp* muss von einer *Aufgabe* bearbeitet werden, die seine Informationsobjekte erzeugt, also mit der Operation ›erfassen‹ gekennzeichnet ist.

- Eine *Unternehmensaufgabe* darf keinen *Objektyp* bearbeiten.
- Eine *Unternehmensaufgabe* kann *Objekttypen* interpretieren.
- Eine *Aufgabe* darf nicht einer anderen *Aufgabe* übergeordnet sein.
- Eine *Unternehmensaufgabe* kann übergeordnetes Element einer Unterordnungsbeziehung sein (*Zerlegung* oder *Spezialisierung*).

Ausführliche Beschreibungen der Modellierungsregeln und des Phasenmodells sowie dabei zu verwendende Entwurfsmustern findet man in [13].

- Die *Fachlichen Ebenen* von Modellen ihrer *Informationssysteme* sind in *Organisationen* derselben Branche meist gleich oder zumindest sehr ähnlich. Für Krankenhäuser wurde daher in [15] ein Referenzmodell vorgestellt, das die für Krankenhäuser typischen *Aufgaben* und *Objekttypen* sowie ihre Beziehungen darstellt. Abbildung 6 enthält einige *Aufgaben* und *Objekttypen* dieses Referenzmodells (z. B. 1.3.1 Erstellung eines Auftrags für eine Maßnahme, 1.4.1 Diagnostische und Therapeutische Maßnahmendurchführung; ADT Information). Es bietet sich an, bei der Modellierung eines bestimmten *Informationssystems* *Aufgaben* aus z. B. diesem Referenzmodell auszuwählen; ggf. kann die Modellierung durch *Spezialisierung* oder *Zerlegung* der *Elementaraufgaben* des Referenzmodells präzisiert werden. In Abbildung 6 werden z. B. *Spezialisierungen* für die *Aufgabe* „1.4.1 Diagnostische und Therapeutische Maßnahmendurchführung“ angegeben, die im Referenzmodell der *Fachlichen Ebene* nicht enthalten sind.

E.B.B-3 Begriffssystem

E.B.B-3.1 Beschreibung des Konzepts

Eine besondere Form von *Objekttypen* sind *Begriffssysteme*. Nach [23] ist ein *Begriffssystem* eine Menge von Begriffen, die je nach *Begriffssystem* mehr oder weniger geordnet sind. Gelegentlich werden *Begriffssysteme* als Ontologien bezeichnet. Beispielsweise ist die internationale Klassifikation der Krankheiten in der zehnten Revision, d. h. die ICD-10 ein solches *Begriffssystem*.

E.B.B-3.2 Spezialisierung und Zerlegung von Begriffssystemen

Es gelten die gleichen Regeln wie für *Objekttypen* (siehe E.B.B-1.3).

E.B.B-3.3 Beziehung zu Aufgabe

Ein *Begriffssystem* kann die gleichen Beziehungen zu *Aufgaben* wie ein *Objektyp* haben (siehe E.B.B-1.4).

E.B.B-3.4 Interebenenbeziehungen zu Anwendungsbausteinen, Sendeschnittstellen, Empfangsschnittstellen und Kommunikationsverbindungen

Ein *Begriffssystem* besitzt die gleichen Interebenenbeziehungen wie *Objekttypen* (siehe E.B.B-1.5 und E.B.B-1.6).

E.B.B-3.5 Fragen des Informationsmanagements und Unterstützung durch den 3LGM²-Baukasten

Fragen des Informationsmanagements	Verweis auf Text
Welche Aufgaben sind auf Verfügbarkeit eines Begriffssystems angewiesen?	E.B.B-2.3, E.B.B-1.4
Wo wird das Begriffssystem gespeichert?	E.B.B-3.4, E.B.D-1.8
Welche Begriffssysteme werden in der Organisation verwendet?	E.B.B-3.1
Gibt es ausreichende Kommunikationsmöglichkeiten, um Begriffssysteme von den Anwendungsbausteinen, in denen sie gespeichert werden, zu den Anwendungsbausteinen zu kommunizieren, in denen sie zur Erledigung von Aufgaben benötigt werden?	[17]
Welche semantische Domäne hat ein Begriffssystem, d. h. in welchen Anwendungsbausteinen wird das Begriffssystem benötigt?	[16]

E.B.B-3.6 Modellierungshinweise

Bei der Modellierung und insbesondere der Bezeichnung von *Begriffssystemen* sind die Hinweise in E.B.B-1.8 zu beachten.

E.B.B-4 Nutzung

E.B.B-4.1 Beschreibung des Konzepts

Für das Informationsmanagement ist es wichtig zu wissen, welche *Anwendungsbausteine*, d. h. *Anwendungssysteme* oder *Organisationssysteme*, die Mitarbeiter der Organisation bei der Erledigung ihrer *Aufgaben* nutzen und ob möglicherweise in unterschiedlichen *Organisationseinheiten* für die Erledigung derselben *Aufgabe* unterschiedliche *Werkzeuge* genutzt werden. Darüber hinaus können die Funktionen eines *Anwendungssystems* durch ein anderes *Anwendungssystem* (z. B. Webbrowser oder Terminalserver) vermittelt werden, so dass sie an bestimmten Arbeitsplätzen genutzt werden können.

Nutzung

Die Klasse *Nutzung* vermittelt zwischen *Aufgaben*, *Organisationseinheiten* und *Anwendungsbausteinen*.

Durch die *Klasse Nutzung* kann beschrieben werden, welche *Anwendungssysteme* bei der Erledigung von *Aufgaben* genutzt werden. Zusätzlich kann über diese *Klasse* die Verbindung zu *Organisationseinheiten* hergestellt werden. Wenn eine *Aufgabe* in einer oder mehreren *Organisationseinheiten* durch mehrere *Anwendungsbausteine* unterstützt wird, bedeutet dies stets, dass diese *Anwendungsbausteine* gleichermaßen und damit alternativ zur Erledigung der *Aufgaben* genutzt werden können. (siehe auch E.B.B-2.4). Eine solche Modellierung drückt damit eine funktionale Redundanz aus.

Durch die *Assoziationsklasse Unterstützung*, welche die Assoziation zwischen *Nutzung* und *Anwendungsbaustein* näher spezifiziert, kann beschrieben werden, dass die Erledigung einer *Aufgabe* durch ein zugangsvermittelndes *Anwendungssystem* unterstützt wird, welches die *Nutzung* von Funktionalität eines anderen *Anwendungssystems* vermittelt. Die *Assoziationsklasse Unterstützung* ist ebenfalls mit *Organisationseinheiten* assoziiert, so dass ausgedrückt werden kann, welches *Anwendungssystem* in welcher *Organisationseinheit* die *Nutzung* eines anderen *Anwendungssystems* vermittelt. Es können somit ebenfalls Redundanzen von zugangsvermittelnden *Anwendungssystemen* auftreten, wenn in unterschiedlichen *Organisationseinheiten* unterschiedliche zugangsvermittelnde *Anwendungssysteme* genutzt werden, um die *Nutzung* desselben *Anwendungssystems* vermitteln.

E.B.B-4.2 Beziehung „erledigt“ zu Aufgabe

Wenn ein *Anwendungsbaustein* in einer *Organisationseinheit* den Mitarbeitern zur Verfügung gestellt wird, dann können ihn die Mitarbeiter dort bei der Erledigung bestimmter *Aufgaben* nutzen.

E.B.B-4.3 Beziehung „wird genutzt durch“ zu Organisationseinheit

Möglicherweise ist vorgesehen, dass ein *Anwendungsbaustein* zwar die Mitarbeiter bei einer Vielzahl von *Aufgaben* unterstützen kann, aber in unterschiedlichen *Organisationseinheiten* nur jeweils individuelle Teilmengen dieser *Aufgaben* unterstützt werden (sollen). Durch die Beziehung „wird genutzt durch“ kann jeder einzelnen Nutzungsbeziehung zwischen *Aufgabe* und *Anwendungsbaustein* eine *Organisationseinheit* zugeordnet werden.

E.B.B-4.4 Interebenenbeziehung „wird unterstützt von“ zu Anwendungsbaustein

Mit dieser Beziehung kann dargestellt werden, dass die Erledigung einer *Aufgabe* in verschiedenen *Organisationseinheiten* durch die Nutzung von *Anwendungsbausteinen* unterstützt wird.

Wenn ein *Anwendungssystem* nicht direkt zur Erledigung einer *Aufgabe* genutzt werden kann, jedoch den Zugang zu dem *Anwendungssystem* ermöglicht, das für die zu erledigende *Aufgabe* genutzt wird, so kann dies durch die *Assoziationsklasse Unterstützung* ausgedrückt werden, welche der Beziehung „wird vermittelt durch“ zugeordnet ist (siehe Beispiel in E.B.D-1.11. und Abbildung 11).

E.B.B-4.5 Fragen des Informationsmanagements und Unterstützung durch das 3LGM²

Fragen des Informationsmanagements	Verweis auf Text
Werden zur Erledigung einer bestimmten Aufgabe unterschiedliche Anwendungsbausteine eingesetzt?	E.B.B-2

Werden für dieselbe Aufgabe in unterschiedlichen Organisationseinheiten unterschiedliche Anwendungsbausteine eingesetzt?	E.B.B-2
Wozu dient ein bestimmter Anwendungsbaustein?	E.B.D-1.6
Wird für eine bestimmte Aufgabe ein modernes Anwendungssystem genutzt?	E.B.B-2.4

E.B.B-4.6 Modellierungshinweise

- Über das Konzept *Nutzung* und die Interebenenbeziehung „*wird unterstützt durch*“ zu *Anwendungsbausteinen* können einer *Aufgabe* mehrere *Anwendungsbausteine* zugeordnet werden. Damit kann aber nicht ausgedrückt werden, dass für die Unterstützung einer bestimmten *Aufgabe* mehrere bestimmte *Anwendungsbausteine* erforderlich sind und z. B. bei dem Ausfall eines der *Anwendungsbausteine* die Erledigung der *Aufgabe* nicht mehr möglich wäre. Wenn dies ausgedrückt werden soll, muss die *Aufgabe* in *Unteraufgaben* zerlegt werden, die jeweils durch eines der *Anwendungsbausteine* vollständig unterstützt werden können. Diese *Unteraufgaben* werden dann eine *Komposition* bilden.
- Einer *Aufgabe* sollte immer mindestens ein Anwendungsbaustein zugeordnet sein, durch welchen sie unterstützt wird, außer die *Aufgabe* kann durch das Informationssystem nicht unterstützt werden. Die Nutzung eines *zugangsvermittelnden Anwendungssystems* über die *Assoziationsklasse Unterstützung* braucht nicht modelliert zu werden. Durch dessen Modellierung kann aber hervorgehoben werden, dass die Funktionalität des Anwendungssystems, dass die *Aufgabe* unterstützt, nicht direkt genutzt werden kann.
- Ist einer *Aufgabe* sowohl ein Anwendungssystem zugeordnet, das sie unterstützt, als auch über die *Assoziationsklasse Unterstützung* ein *Anwendungssystem*, das die Funktionalität eines anderen Anwendungssystems zur Erledigung der *Aufgabe* vermittelt, dann brauchen diese beiden *Anwendungsbausteine* auf der *Logischen Werkzeugebene* keine *Kommunikationsverbindung* haben. Allerdings müssen die entsprechenden *Physischen Datenverarbeitungssysteme*, auf denen die *Anwendungsbausteine* installiert sind, miteinander verbunden sein.

E.B.B-4.7 Kompatibilität zu 3LGM² V1.x

Dieses Konzept ersetzt die Anwendungsbausteinkonfigurationen.

E.B.B-5 Organisationseinheit

E.B.B-5.1 Beschreibung des Konzepts

Die unterschiedliche Unterstützung einer *Aufgabe* durch verschiedene Anwendungssysteme ist häufig an unterschiedliche Verantwortungsbereiche der Organisation gebunden. Die Kenntnis dieser Verantwortungsbereiche ist für das Informationsmanagement wichtig, um erkennen zu können, mit welchen Verantwortlichen z. B. bei der Reduzierung funktionaler Redundanz (vgl. [22]) zu sprechen ist.

Definition: Organisationseinheit

Eine *Organisationseinheit* ist eine durch Verantwortlichkeiten abgrenzbare organisatorische Gliederung einer Organisation.

Auch eine Organisation, z. B. ein Krankenhaus, kann *Organisationseinheit* in Bezug auf einen Verbund oder ein System von Krankenhäusern z. B. in einer Region sein.

Beispiele für *Organisationseinheiten* in Krankenhäusern sind: Abteilung für Transplantationschirurgie, Stabstelle Unternehmenskommunikation, Abteilung Patientenverwaltung, gemeinsamer Operationsbereich der Abteilung für HNO und der Abteilung für Augenheilkunde. *Organisationseinheiten* dürfen nicht mit Räumen oder *Standorten* verwechselt werden (vgl. E.B.E-3.1).

Auch wenn bei Bedarf ein 3LGM²-Modell dazu genutzt werden kann zu beschreiben, welche *Organisationseinheiten* welche *Aufgaben* erledigen, ohne dabei auf die jeweilige Werkzeugunterstützung einzugehen, dient dieses Konzept in 3LGM²-Modellen primär dazu, aufdecken zu können, wo und von wem dieselbe *Aufgabe* mit unterschiedlichen Werkzeugen erledigt wird. Über das Konzept *Nutzung* (siehe E.B.B-4) kann modelliert werden, dass eine *Aufgabe* in unterschiedlichen *Organisationseinheiten* durch *Nutzung* verschiedener *Anwendungsbausteine* unterstützt wird. Über die *Assoziationsklasse Unterstützung* kann zusätzlich modelliert werden, dass die *Nutzung* eines *Anwendungssystems* im Rahmen einer *Aufgabe* auch durch verschiedene *Anwendungssysteme* vermittelt wird.

E.B.B-5.2 Zerlegung von Organisationseinheiten

E.B.B-5.2.1 Konzept

Eine *Organisationseinheit* kann durch *Zerlegung* in *Unterorganisationseinheiten* zerlegt werden. Eine *Spezialisierung*, wie beispielweise für *Aufgaben* oder *Objekttypen* möglich, ist bei *Organisationseinheiten* nicht möglich. *Unterorganisationseinheiten* werden auch als *Teilorganisationseinheiten* bezeichnet, da sie immer eine Zerlegung einer *Gruppenorganisationseinheit* in Teile bedeuten.

Eine *Organisationseinheit* kann *Unterorganisationseinheit* genau einer *Gruppenorganisationseinheit* sein.

E.B.B-5.2.2 Vererbung

Alle Eigenschaften und *Assoziationen*, die einer *Gruppenorganisationseinheit* zugewiesen werden, werden auch für ihre *Teilorganisationseinheiten* als gegeben angesehen, d. h. die Eigenschaften und *Assoziationen* werden vererbt. Dies bedeutet, dass alle Beziehungen zu *Nutzungen*, die für die *Gruppenorganisationseinheit* modelliert sind, auch für alle ihre *Teilorganisationseinheiten* als modelliert gelten, also vererbt werden.

E.B.B-5.2.3 Vergrößern

Die *Teilorganisationseinheiten* einer *Gruppenorganisationseinheit* G können durch Vergrößern vorübergehend inhaltlich und visuell ausgeblendet werden

Bei dem Vergrößern der *Teilorganisationseinheiten* einer *Gruppenorganisationseinheit* werden die *Teilorganisationseinheiten* (vorübergehend) aus dem Modell entfernt. Dies bedeutet, dass auf die differenzierte Betrachtung, die durch die *Teilorganisationseinheiten* gegeben ist, verzichtet wird. Die *Assoziationen* der *Teilorganisationseinheiten* zu *Nutzung* werden dann für die Dauer der Vergrößerung vollständig der *Gruppenorganisationseinheit* zugeordnet.

E.B.B-5.3 Beziehung zu Nutzung

Diese Beziehung drückt aus, in welcher *Organisationseinheit* *Anwendungsbausteine* für die Erledigung von *Aufgaben* genutzt werden.

E.B.B-5.4 Beziehung zu Unterstützung

Durch diese Beziehung kann ausgedrückt werden, dass die *Nutzung* eines *Anwendungssystems* in einer *Organisationseinheit* durch bestimmte (andere) *Anwendungssysteme* unterstützt wird.

E.B.B-5.5 Fragen des Informationsmanagements und Unterstützung durch das 3LGM²

Fragen des Informationsmanagements	Verweis auf Text
Welche Anwendungsbausteine werden in einer bestimmten Organisationseinheit zur Unterstützung von Aufgaben eingesetzt?	E.B.D-1
Wird eine bestimmte Aufgabe, die in einer Organisationseinheit mit einem bestimmten Anwendungsbaustein unterstützt wird, in einer anderen Organisationseinheit mit einem anderen Anwendungsbaustein unterstützt?	E.B.B-2.4
Wird ein Anwendungsbaustein in verschiedenen Organisationseinheiten zur Unterstützung unterschiedlicher Aufgaben eingesetzt?	E.B.D-1.6

E.B.B-5.6 Modellierungshinweise

- Das Vererbungsprinzip ist stets sorgfältig zu beachten. So ist stets zu beachten, dass z. B. eine Beziehung die eine *Gruppenorganisationseinheit* mit einer *Nutzung* hat, auch für alle *Teilorganisationseinheiten* gegeben ist. Dies gilt auch dann, wenn einer *Organisationseinheit* erst später *Teilorganisationseinheiten* zugeordnet werden, also die *Organisationseinheit* zur *Gruppenorganisationseinheit* wird. Es wird jedoch nicht immer davon auszugehen sein, dass alle Eigenschaften der *Gruppenorganisationseinheit* auch für alle (neuen) *Teilorganisationseinheiten* gelten. Es sollte also immer überprüft werden, ob wirklich nur die Eigenschaften bei der neuen *Gruppenorganisationseinheit* belassen worden sind, die auch wirklich für alle *Teilorganisationseinheiten* gelten sollen. Diese Situation ergibt sich z. B., wenn *Organisationseinheiten* eines

Referenzmodells in ein Modell übernommen, dort aber zur präziseren Modellierung aufgeteilt werden. Grundsätzlich sollte also vermieden werden, *Gruppenorganisationseinheiten* Beziehungen zuzuweisen.

E.B.B-6 Prozesse

E.B.B-6.1 Beschreibung des Konzepts

Während die bisher beschriebenen Konzepte der *Fachlichen Ebene* die Datenverarbeitung aus der statischen Sicht betrachtet haben, sollen *Prozesse* die *Fachliche Ebene* um dynamische Aspekte erweitern. Dazu werden datenverarbeitende *Aufgaben*, die Objekttypen bearbeiten und interpretieren, in eine logische und zeitliche Reihenfolge gebracht. Dadurch besteht in 3LGM²-Modellen auch die Möglichkeit zu untersuchen, ob die Sequenz von datenverarbeitenden *Aufgaben* innerhalb des *Prozesses* durch die Informationssystemarchitektur unterstützt wird. D. h. es kann analysiert werden, ob es Kommunikationspfade zwischen Anwendungssystemen gibt, über die *Objekttypen* kommuniziert werden können, die von aufeinanderfolgenden *Aufgaben* bearbeitet und interpretiert werden.

Definition Prozess

Ein *Prozess* (genauer: Informationsprozess) ist eine Abfolge von *Aufgaben*, die Daten über Objekte interpretieren oder bearbeiten.

Um einen echten Informationsprozess zu definieren, müssen alle *Aufgaben* (bis auf die erste) wenigstens einen *Objekttyp* interpretieren, der durch eine der Vorgängeraufgaben in dem Prozess bearbeitet wurde. Nur dann stehen die Aufgaben aus der Sicht des Informationssystems in einem sinnvollen Prozesszusammenhang [24].

Durch den hier vorgestellten Prozessbegriff wird im Metamodell besonders der datenverarbeitende Aspekt während des *Prozesses* und damit die Beziehung des *Prozesses* zur *Logischen Werkzeugebene* hervorgehoben.

In Abbildung 8 ist ein Behandlungsablauf in der Radiologie als *Prozess* dargestellt. Nach der administrativen Patientenaufnahme, bei der die ADT-Information des Patienten bearbeitet wird, kann die Anforderung für die Radiologie erstellt werden, für die die ADT-Information des Patienten benötigt wird. Durch die Terminvergabe entsteht ein terminierter Einzelauftrag. Die Anforderung und der terminierte Einzelauftrag werden durch die Radiologische Diagnostik interpretiert. Zur Radiologischen Diagnostik gehört die Radiologische Bilderzeugung, bei der das Radiologische Bild entsteht, und die Radiologische Bildbefundung, bei der das Bild interpretiert wird.

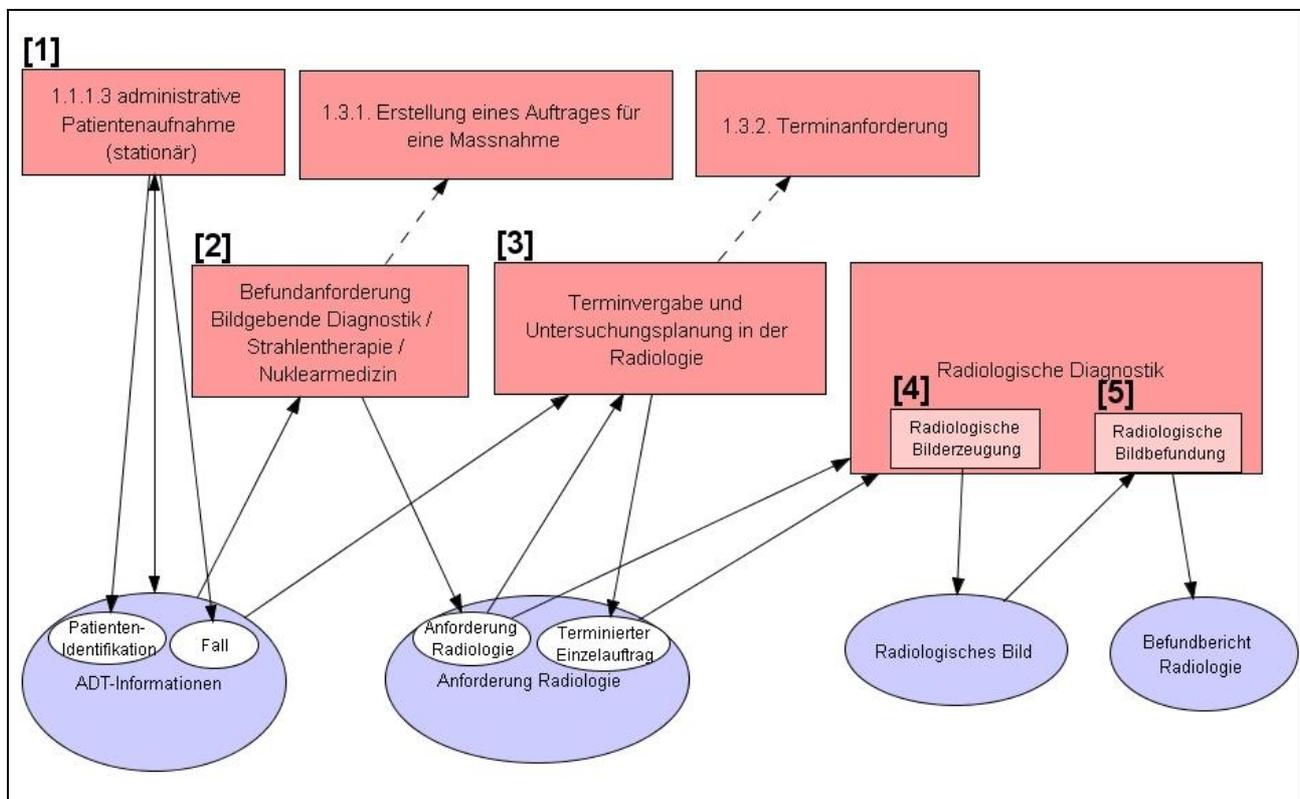


Abbildung 8: Beispielprozess „Radiologischer Behandlungsablauf“

E.B.B-6.2 Sonderstellung von Prozessen

E.B.B-6.2.1 Konzept

Prozesse unterscheiden sich von den anderen Begriffen insofern, als ihre Existenz an die Existenz einer anderen Elementklasse (d. h. an *Aufgaben*) gebunden ist und sie Elemente dieser Elementklasse gruppieren und dabei in eine zeitliche Reihenfolge bringen. *Assoziationen* von *Prozessen* zu sich selbst sind nicht vorgesehen.

E.B.B-6.3 Beziehung zu Aufgaben

Ein *Prozess* ist im Metamodell dadurch charakterisiert, dass innerhalb von ihm mindestens eine *Aufgabe* ausgeführt wird. Dabei gilt: Bei aufeinanderfolgenden datenverarbeitenden *Aufgaben* muss die erste *Aufgabe* einen *Objekttyp* bearbeiten, der von einer der nächsten *Aufgaben* interpretiert wird. Weiterhin darf keine *Aufgabe*, die zum *Prozess* gehört, *Unteraufgabe* einer anderen *Aufgabe* innerhalb des *Prozesses* sein.

E.B.B-6.4 Fragen des Informationsmanagements und Unterstützung durch das 3LGM²

Fragen des Informationsmanagements	Verweis auf Text
Treten bei einem datenverarbeitenden Prozess Medienbrüche auf, d. h. werden Daten zwischen rechnerbasierten Anwendungsbausteinen und papierbasierten Organisationssystemen ausgetauscht?	E.B.D-1
Hat das Informationssystem eine geeignete Infrastruktur (Kommunikationsverbindungen, Schnittstellen, Anwendungsbausteine) für den Transport von Daten? Werden zusätzlich geplante Komponenten hinreichend sein, um die benötigte Infrastruktur zur erschaffen?	E.B.D-1, E.B.D-3

E.B.B-6.5 Modellierungshinweise

- Aufgaben innerhalb eines *Prozesses* dürfen nicht in einer Ist-Teil-Von- oder Spezialisierungsbeziehung zueinander stehen.

E.B.C Zentrale Begriffe der Logischen und Physischen Werkzeugebene

In diesem Abschnitt werden die Begriffe *Physisches Datenverarbeitungssystem*, *Anwendungsbaustein*, *Anwendungssystem* und *Organisationssystem* eingeführt. Modellierungshinweise, Beziehungen der Begriffe untereinander und zu weiteren Begriffen, sowie weitere Begriffe der *Logischen* bzw. *Physischen Werkzeugebene* folgen in E.B.D bzw. E.B.E.

E.B.C-1 Physisches Datenverarbeitungssystem

Als *datenverarbeitende Werkzeuge* zur Unterstützung bei der Erledigung von *Aufgaben* werden zunächst physisch greifbare und damit lokalisierbare technische Dinge bzw. Gerätschaften benötigt. Mit diesen Werkzeugen müssen die Daten, welche die Informationen über *Objekte* darstellen, und ihre Datenträger gehandhabt werden können. Hierfür sind z. B. PC, Server, Kommunikationsnetzkomponenten oder Regalanlagen, Transportfahrzeuge für Akten usw. geeignet.

In modernen rechnerbasierten Informationssystemen werden häufig physisch greifbare Geräte per Software simuliert, z. B. bei sogenannten virtuellen Servern.

Definition: Physisches Datenverarbeitungssystem (PDVS)

Ein *Physisches Datenverarbeitungssystem* ist ein physisch greifbarer Gegenstand oder ein simulierter physisch greifbarer Gegenstand, mit dem Daten entgegengenommen, gespeichert, weitergeleitet oder gezielt verändert werden können. Die Entgegennahme, Speicherung, Weiterleitung oder gezielte Veränderung von Daten wird als *Verarbeitung von Daten* bezeichnet. Die Verarbeitung der Daten folgt Regeln, die vorgegeben werden können.

Wir bezeichnen *Physische Datenverarbeitungssysteme* auch als *physische Werkzeuge der Datenverarbeitung*. *Physische Datenverarbeitungssysteme* werden der *Physischen Werkzeugebene* zugeordnet (siehe E.B.E-1). Zwischen ihnen können Daten über *Datenübertragungsverbindungen* transportiert werden (siehe E.B.E-1.4).

Rechnersysteme (PCs, Server), Monitore, Drucker, Switches, Router, Subnetze, Netzwerkkarten, Netzanschlusspunkte aber auch virtuelle Server usw. sind *Physische Datenverarbeitungssysteme*, deren *Verarbeitung von Daten* und deren Zusammenwirken durch Regeln gesteuert werden, die in Form von Software gegeben sind. Sie werden als *softwaregesteuerte Physische Datenverarbeitungssysteme* bezeichnet.

Insbesondere Rechnersysteme können z. B. durch Eigenschaften wie Betriebssystem, Seriennummer, Inventarnummer, Festplattenspeicher, Arbeitsspeicher, Prozessor usw. näher beschrieben werden. Abbildung 9 zeigt ein Beispiel eines solchen *Physischen Datenverarbeitungssystems*, wie es im 3LGM²-Baukasten näher beschrieben werden kann.



Abbildung 9: Beispiel für ein *softwaregesteuertes Physisches Datenverarbeitungssystem*.

Andere, auch als konventionell bezeichnete Gegenstände oder Geräte für die Datenverarbeitung sind z. B. Archivregale, Posteingangs- und Postausgangskörbe und Transportfahrzeuge für Akten. Die *Verarbeitung von Daten* in solchen konventionellen *Physischen Datenverarbeitungssystemen* und ihr Zusammenwirken muss von Menschen je nach Art des Bausteins unter Anwendung gewisser Regeln gesteuert werden. Sie werden als *personengesteuerte Physische Datenverarbeitungssysteme* bezeichnet. Die Regeln werden als Organisationspläne bezeichnet. Die steuernden Menschen werden nicht als *Physische Datenverarbeitungssysteme* aufgefasst.

Wie *Physische Datenverarbeitungssysteme* miteinander in Beziehung stehen, wird in Kapitel E.B.E näher beschrieben.

E.B.C-2 Anwendungsbaustein, Anwendungssystem, Organisationssystem

Physische Datenverarbeitungssysteme reichen alleine nicht aus, um die Erledigung datenverarbeitender *Aufgaben* zu unterstützen. So ist z. B. ein PC, der alleine mit dem Betriebssystem ausgestattet ist, ebenso wenig hilfreich bei der Patientenaufnahme wie ein Regal ohne Archivierungsregeln, die von entsprechendem Personal bei der Archivierung von Krankenakten umgesetzt werden. Vielmehr sind die genannten Regeln und eine tatsächliche Steuerung anhand dieser Regeln erforderlich.

Definition: Anwendungsbaustein

Ein *Anwendungsbaustein* (engl. application component) ist eine Menge tatsächlich nutzbarer Regeln, welche die Datenverarbeitung in bestimmten *Physischen Datenverarbeitungssystemen* steuern.

Regeln sind dann tatsächlich nutzbar, wenn die Erledigung bestimmter *Aufgaben* oder die Kommunikation zwischen *Anwendungsbausteinen* damit tatsächlich unterstützt wird.

Regeln, die nicht tatsächlich verwendet werden, sondern nur z. B. als nicht-installierte Software auf einer CD oder als Muster eines Organisationsplans in einem Ordner festgehalten bzw. beschrieben werden, bilden also noch keinen *Anwendungsbaustein* sondern – im ersten beschriebenen Fall – ein *Softwareprodukt* (vgl. E.B.D-2).

Anwendungsbausteine werden als *logische Werkzeuge* der Informationsverarbeitung bezeichnet, da sie für die Mitarbeiter der Organisation zwar anwendbar, aber im Gegensatz zu den *physischen Werkzeugen* nicht physisch greifbar sind. *Anwendungsbausteine* werden der *Logischen Werkzeugebene* zugeordnet (siehe E.B.D).

Wenn die Regeln eines *Anwendungsbausteins* in Form von Software implementiert sind, sprechen wir von einem *Anwendungssystem*:

Definition: Anwendungssystem (Synonym: rechnerbasierter Anwendungsbaustein)

Ein *Anwendungssystem* (engl. application system) ist ein *Anwendungsbaustein*, bei dem die nutzbaren Regeln durch ein Softwareprodukt repräsentiert sind.

Ein *Anwendungssystem* wird also realisiert durch die Installation, *Adaptation* und Inbetriebnahme von *Softwareprodukten* auf durch *Software* steuerbaren *Physischen Datenverarbeitungssystemen*. Solche *Physischen Datenverarbeitungssysteme* sind z. B. Rechnersysteme.

Während das *Softwareprodukt* als eine eher abstrakte Repräsentation von Regeln aufgefasst werden kann, wie bestimmte Arten von *Physischen Datenverarbeitungssystemen* arbeiten und zusammenwirken könnten, sind bei einem *Anwendungssystem* die Regeln konkretisiert und es ist explizit und konkret festgelegt, welche bestimmten *Physischen Datenverarbeitungssysteme* in einer bestimmten *Organisation* gesteuert werden und zusammenwirken sollen.

Das Synonym *rechnerbasierter Anwendungsbaustein* wird vorzugsweise dann verwendet, wenn deutlich gemacht werden soll, dass das *Anwendungssystem* als ein Teil oder Modul eines ‚größeren‘ Systems aufzufassen ist.

Wenn die Regeln eines *Anwendungsbausteins* nicht in Form von Software implementiert sind bzw. sich die verwendeten *Physischen Datenverarbeitungssysteme* nicht mit Hilfe von Software steuern lassen, müssen die *Physischen Datenverarbeitungssysteme* durch die Anwendung von *Organisationsplänen* gesteuert werden. Die *Organisationspläne* werden von Menschen angewendet.

Definition: Organisationssystem

Ein *Organisationssystem* (engl. organizational system) ist ein *Anwendungsbaustein*, bei dem die nutzbaren Regeln durch *Organisationspläne* repräsentiert sind.

Das *Organisationssystem* entsteht also nach der impliziten oder expliziten Verabschiedung von *Organisationsplänen* in dem Moment, wo Menschen mit der Anwendung dieser *Organisationspläne* beginnen. Ein *Organisationsplan* kann ein schriftliches Dokument sein, in dem in Form von Arbeitsanweisungen beschrieben ist, wie bestimmte *Aufgaben* zu erledigen sind. In den meisten Fällen wird es sich aber eher um mündliche Vereinbarungen handeln.

Organisationssysteme haben eine gewisse Ähnlichkeit mit sogenannten Arbeitssystemen im Sinne der REFA [25]⁵ [26]. Auch Arbeitssysteme gehen von der Existenz von *Organisationssystemen* aus. Ein Arbeitssystem erhielte man z. B., wenn man zu einem *Organisationssystem* die von ihm gesteuerten *Physischen Datenverarbeitungssysteme* und die Menschen hinzunähme, die *Organisationspläne* anwenden sowie die, die bei der Erledigung der *Aufgaben* unterstützt werden.

E.B.D Logische Werkzeugebene

Die *logischen Werkzeuge*, also insbesondere die *Anwendungsbausteine* werden in der *Logischen Werkzeugebene* beschrieben.

Auf der *Logischen Werkzeugebene* stehen die *Anwendungsbausteine* im Mittelpunkt. Sie beschreibt, womit die Erledigung von *Aufgaben* unterstützt wird, wo Daten über *Objektypen* logisch gespeichert werden und wie *Anwendungsbausteine* kommunizieren müssen, um zu gewährleisten, dass bei der Erledigung von *Aufgaben* die benötigten *Objektypen* tatsächlich bearbeitbar sind.

E.B.D-1 Anwendungsbaustein*E.B.D-1.1 Beschreibung des Konzepts*

Siehe E.B.C-2

E.B.D-1.2 Spezialisierung von Anwendungsbausteinen

Ein *Anwendungsbaustein* kann *Anwendungssystem* oder *Organisationssystem* sein. Diese Spezialisierung von Anwendungsbausteinen ist Teil des Metamodells. Siehe E.B.C-2.

⁵ <http://www.refa.de/INTERNATIONAL/deutsch/system/system.htm#3>

E.B.D-1.3 Zerlegung von Anwendungsbausteinen

E.B.D-1.3.1 Konzept

Ein *Anwendungsbaustein* (*Gruppenanwendungsbaustein*) kann durch *Zerlegung* in *Unteranwendungsbausteine* zerlegt werden. Diese *Unteranwendungsbausteine* sind als echte Teile des zugehörigen *Gruppenanwendungsbausteins* zu verstehen und werden auch als *Teilanwendungsbausteine* bezeichnet.

Ein *Anwendungsbaustein* kann nur *Teilanwendungsbaustein* eines einzigen *Gruppenanwendungsbausteins* sein.

Ein *Organisationssystem* kann *Organisationssysteme*, aber auch *Anwendungssysteme* als Teile enthalten. Auf diese Weise können z. B. alle *Anwendungssysteme* eines *Informationssystems* (z. B. eines Krankenhausinformationssystems) zu einem *Organisationssystem* zusammengefasst werden, das dann wiederum als *Anwendungsbaustein* eines übergeordneten *Informationssystems* (z. B. regionales Gesundheitssystem) aufgefasst werden könnte.

Anwendungssysteme können allerdings nur aus anderen *Teilanwendungssystemen* bestehen.

E.B.D-1.3.2 Vererbung

Alle *Eigenschaften*, die einem *Anwendungsbaustein* zugewiesen werden, werden auch für seine *Teilanwendungsbausteine* als gegeben angesehen (Vererbung).

Für die Vererbung gilt insbesondere:

- Ist für einen *Anwendungsbaustein* die Speicherung eines *Objektyps* modelliert, wird davon ausgegangen, dass auch alle *Teilanwendungsbausteine* Zugriff auf die Daten zu diesem *Objektyp* haben. Ist ein *Anwendungsbaustein* ein *Master* des *Objektyps*, dann gelten auch alle *Teilanwendungsbausteine* als *Master* des *Objektyps*.
- Besitzt ein *Anwendungsbaustein* eine *Kommunikationsschnittstelle*, wird davon ausgegangen, dass auch alle *Teilanwendungsbausteine* diese *Kommunikationsschnittstelle* nutzen können.
- Hat ein *Anwendungsbaustein* eine *Beziehung zur Nutzung*, wird davon ausgegangen, dass auch alle *Teilanwendungsbausteine* zu dieser *Nutzung* in *Beziehung* stehen. Die *Komposition* bedeutet hier, dass alle *Teilanwendungsbausteine* unbedingt erforderlich sind, damit der zusammenfassende *Anwendungsbaustein* tatsächlich die *Aufgabe* unterstützen kann (vgl. differierendes Konzept in E.B.E-1.2).
- Basiert ein *Anwendungssystem* auf einem *Softwareprodukt*, wird davon ausgegangen, dass auch alle *Teilanwendungssysteme* auf diesem *Softwareprodukt* basieren.

Diese Regeln erfordern im Einzelfall eine sorgfältige Abwägung, ob solche *Eigenschaften* bzw. *Beziehungen* für *Anwendungsbausteine* modelliert werden sollen, die aus *Teilanwendungsbausteinen* bestehen oder ob diese *Eigenschaften* besser unmittelbar den *Teilanwendungsbausteinen* zugeordnet werden sollen.

E.B.D-1.3.3 Vergrößern

Bei dem Vergrößern eines *Gruppenanwendungsbausteins* werden seine *Teilanwendungsbausteine* (vorübergehend) aus dem Modell entfernt. Dies bedeutet, dass auf die differenzierte Betrachtung, die durch die *Teilanwendungsbausteine* gegeben ist, verzichtet wird. Die *Assoziationen* der *Teilanwendungsbausteine* zu *Objektyp*, *Nutzung*, *Physischem Datenverarbeitungssystem* und *Kommunikationsschnittstelle* werden dann für die Dauer der Vergrößerung dem *Gruppenanwendungsbaustein* zugeordnet.

E.B.D-1.4 Beziehung „basiert auf“ zu Softwareprodukt

Diese Beziehung besteht nur für *Anwendungssysteme*.

Ein *Anwendungssystem* kann auf mehreren *Softwareprodukten* basieren. Allerdings ist nicht vorgesehen, dass dann näher spezifiziert wird, welche *Funktionalitäten* des *Anwendungssystems* durch die einzelnen *Softwareprodukte* realisiert werden. Ist dies aber gewünscht, z. B. um zu verdeutlichen, dass die Software SQL das Datenbanksystem des Anwendungssystems verwaltet, während die Software ArchiMed die *Funktionalität* realisiert, die für die Erledigung der *Aufgaben* erforderlich ist, sollte das *Anwendungssystem* in zwei entsprechende *Teilanwendungssysteme* zerlegt werden (siehe E.B.D-1.3.1).

Der Zusammenhang *Anwendungssystem* – *Softwareprodukt* soll an folgendem Beispiel verdeutlicht werden: Wenn ein Krankenhaus z. B. ein *Softwareprodukt* für die Archivverwaltung in Patientenaktenarchiven, nennen wir es ArchiMed, von der Firma Graecia GmbH kauft, ist das *Softwareprodukt* zunächst 'leer'. Die Disketten oder CD-ROMs, die an

verschiedene Krankenhäuser ausgeliefert werden, haben üblicherweise denselben Inhalt. Die krankenhausspezifischen Anpassungen müssen erst noch durchgeführt werden, z. B. muss parametrisiert werden, d. h. mit Daten 'gefüllt', wie die Akten sortiert werden, welche Abteilungen es gibt mit welchen Kostenstellennummern, auch natürlich, wie das Krankenhaus heißt, damit der Name auf dem Bildschirm oder auf Briefen erscheinen kann. Nach diesen Anpassungen - auch Adaptierung genannt -, der Installation und der Inbetriebnahme, wird aus dem *Softwareprodukt* ArchiMed der Firma Graecia GmbH ein für dieses Krankenhaus spezifisches und nutzbares *Anwendungssystem*, das etwa als Archivverwaltungssystem bezeichnet werden kann.

E.B.D-1.5 Beziehung „besitzt“ zu Kommunikationsschnittstelle

Ein *Anwendungsbaustein* kann mit anderen *Anwendungsbausteinen* durch den Versand bzw. das Empfangen von Daten kommunizieren. Für diese Kommunikation benötigt der *Anwendungsbaustein* *Kommunikationsschnittstellen*. Es wird unterschieden, ob eine *Kommunikationsschnittstelle* als *Sendeschnittstelle* zum Senden oder als *Empfangsschnittstelle* zum Empfangen von Daten dient. Eine *Kommunikationsschnittstelle* existiert nur im Zusammenhang mit einem bestimmten *Anwendungsbaustein*.

Kommunikationsschnittstellen von *Anwendungssystemen* sind Installationen von Kommunikationssoftware und als solche über die Identifikation der Software abgrenzbar und identifizierbar.

Kommunikationsschnittstellen von *Organisationssystemen* sind in der Regel in der jeweiligen Organisation nicht eindeutig lokalisierbar oder abgrenzbar. In einem Modell sollen die *Kommunikationsschnittstellen* eines *Organisationssystems* daher auch nicht einen lokalisierbaren oder abgrenzbaren Teil der Organisation repräsentieren, der für das Senden oder Empfangen von Daten zuständig wäre. Vielmehr repräsentiert eine *Kommunikationsschnittstelle* eines *Organisationssystems* abstrakt die Möglichkeit des *Organisationssystems*, Daten an andere *Anwendungsbausteine* abzugeben (zu senden) oder sie entgegenzunehmen (zu empfangen).

E.B.D-1.6 Interebenenbeziehung „unterstützt“ zu Nutzung

Die Beziehung beschreibt, wofür ein *Anwendungsbaustein* in verschiedenen *Organisationseinheiten* genutzt werden kann. Es kann durchaus sein, dass ein *Anwendungsbaustein* in der gleichen *Organisationseinheit* oder auch in verschiedenen *Organisationseinheiten* zur Erledigung unterschiedlicher *Aufgaben* eingesetzt werden kann.

E.B.D-1.6.1 Kompatibilität zu 3LGM² V1.x

Diese Beziehung ersetzt die Anwendungsbausteinkonfigurationen.

E.B.D-1.7 Interebenenbeziehung „vermittelt“ zu Unterstützung

Die Beziehung besteht nur für *Anwendungssysteme* und beschreibt, dass ein sogenanntes *zugangsvermittelndes Anwendungssystem* die Nutzung eines anderen *Anwendungssystems* vermittelt. Es besteht die Möglichkeit, dass in verschiedenen *Organisationseinheiten* verschiedene *zugangsvermittelnde Anwendungssysteme* zur Nutzung desselben *Anwendungssystems* verwendet werden. Ein *zugangsvermittelndes Anwendungssystem* kann jedoch auch die *Nutzung* verschiedener *Anwendungssysteme* vermitteln.

E.B.D-1.8 Interebenenbeziehung „speichert“ zu Objekttyp

Anwendungsbausteine werden nicht nur bei der Erledigung von *Aufgaben* genutzt, sondern auch zur Speicherung von Daten. Die Modellierung des Speicherns eines *Objekttyps* in einem *Anwendungsbaustein* drückt aus, dass Daten über *Objekte* dieses *Objekttyps* in dem *Anwendungsbaustein* gespeichert werden.

Ein *Anwendungsbaustein* wird als *Master* eines bestimmten *Objekttyps* bezeichnet, wenn Daten über *Objekte* dieses *Objekttyps* in diesem *Anwendungsbaustein* verändert werden dürfen. Ein *Objekttyp* kann mehrere *Master* haben. Diese Modellierung ist wichtig, um bei der redundanten Speicherung von Daten über *Objekte* dieses *Objekttyps* deutlich zu machen, dass nach dem Ändern dieser Daten diese Daten von dem *Master* noch an alle anderen *Anwendungsbausteine* geleitet werden müssen, die diese Daten redundant speichern. Wenn es zu einem *Objekttyp* mehrere *Master* gibt, ist ein aufwändiges Transaktionsprotokoll, z. B. das 2-phase-commit Protokoll, erforderlich, um die Integrität der Daten in dem *Master* sicherzustellen. Dies gilt nicht nur für *Anwendungssysteme*, sondern auch für *Organisationssysteme*.

Zur Modellierung von Datenbanksystemen siehe E.B.D-1.11.

E.B.D-1.9 Interebenenbeziehungen „benötigt für Funktionalität“ und „benötigt für Speicherung“ zu Physischem Datenverarbeitungssystem

Ein *Anwendungsbaustein* wird genutzt zur Erledigung von *Aufgaben* und zur Speicherung von Daten. Für beide Zwecke werden bestimmte *Physische Datenverarbeitungssysteme* benötigt, in denen dann der *Anwendungsbaustein* die Datenverarbeitung anhand seiner Regeln steuert (vgl. Definition in E.B.C-2).

Über die *Assoziation* „benötigt für Funktionalität“ werden einem *Anwendungsbaustein* ein oder mehrere *Physische Datenverarbeitungssysteme* zugeordnet, die für die Entgegennahme, Weiterleitung oder gezielte Veränderung von Daten erforderlich sind (vgl. Definition in E.B.C-1).

Über die *Assoziation* „benötigt für Speicherung“ werden einem *Anwendungsbaustein* ein oder mehrere *Physische Datenverarbeitungssysteme* zugeordnet, die für die Speicherung von Daten erforderlich sind (vgl. Definition in E.B.C-1).

Werden dem *Anwendungsbaustein* mehrere *Physische Datenverarbeitungssysteme* über eine der beiden *Assoziationen* zugeordnet, bedeutet dies, dass **alle** jeweils zugeordneten *Physischen Datenverarbeitungssysteme* erforderlich sind um die Funktionalität des *Anwendungsbausteins* bzw. die Datenspeicherung in dem *Anwendungsbaustein* zu ermöglichen.

Mehrere *Physische Datenverarbeitungssysteme* können über die Beziehung „wird virtualisiert“ zu einem virtuellen *Physischen Datenverarbeitungssystem* bzw. Cluster gruppiert werden (siehe E.B.E-1.3). Wird einem *Anwendungsbaustein* der Cluster zugeordnet, wird damit ausgedrückt, dass nicht alle der *Physischen Datenverarbeitungssysteme* dieses Clusters notwendig sind, sondern nur mindestens eines der *Physischen Datenverarbeitungssysteme* dieses Clusters notwendig ist. Somit ist ein *virtuelles Physisches Datenverarbeitungssystem* nicht gleichzusetzen mit einem kommerziell erhältlichen Virtualisierungssystem wie z. B. VMware, sondern es ist ein verallgemeinerndes Konzept, welches aber auch zur Modellierung von Virtualisierungssystemen genutzt werden kann. Wie die „wird virtualisiert“-Beziehung zur Modellierung eines solchen Virtualisierungssystems genutzt wird, ist in E.B.E-1.3 beschrieben.

Mit Hilfe eines Clusters kann z. B. modelliert werden, dass ein Anwendungssystem für seine Funktionalität mindestens einen Server (aus einer Gruppe möglicherweise redundanter Rechner) **und** mindestens einen Client (aus einer Gruppe von Arbeitsplatzrechnern) benötigt. Abbildung 10 zeigt die physische Basis für ein Bildbefundungssystem. Damit das Bildbefundungssystem benutzt werden kann, wird mindestens ein Server aus der Gruppe „Virtuell: MagicView-Server“ sowie mindestens eine Workstation aus der Gruppe „Virtuell: Befundungsworkstations“ benötigt.

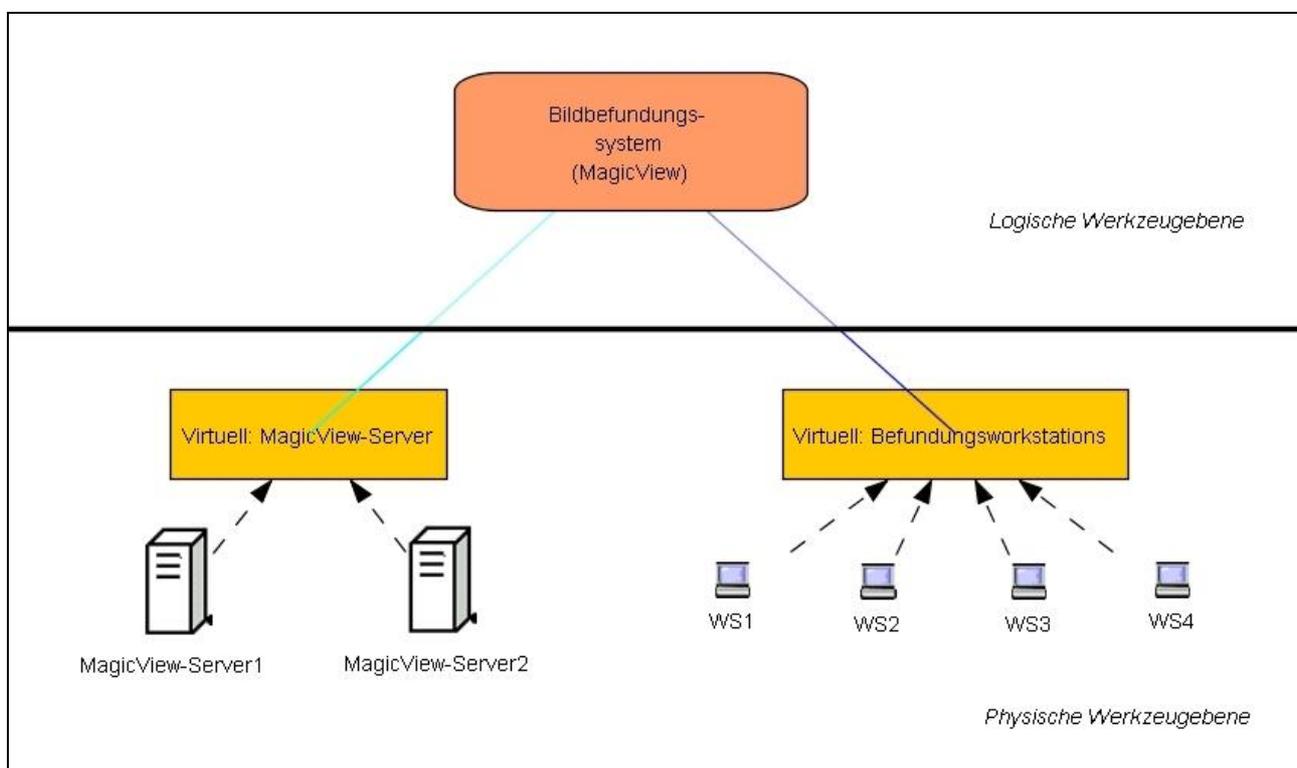


Abbildung 10: Beispiel für die Modellierung *virtueller Physischer Datenverarbeitungssysteme*

E.B.D-1.10 Fragen des Informationsmanagements und Unterstützung durch das 3LGM²

Fragen des Informationsmanagements	Verweis auf Text
Welche Daten, d. h. welche Daten über Typen von Objekten werden auf welchem Server oder in welchem Aktenlager gespeichert?	E.B.D-1.9
Welche Aufgaben können nicht mehr unterstützt werden, wenn ein Anwendungsbaustein ausfällt? Hierbei ist zu berücksichtigen, dass der Anwendungsbaustein möglicherweise selber die Erledigung von Aufgaben unterstützt oder aber dafür sorgt, dass Daten zu Objekten zu einem anderen Anwendungssystem transportiert werden, die dort für die Erledigung von Aufgaben benötigt werden.	E.B.B-2.4
Welche Datendomäne hat ein Objekttyp, d. h. in welchen Anwendungsbausteinen wird der Objekttyp benötigt?	[16]
Ist im Informationssystem / in bestimmten Teilen des Informationssystems Datenintegration gegeben, d. h. müssen bestimmte Daten nur einmal erfasst werden, auch wenn die dadurch repräsentierten Informationen bei der Arbeit mit mehreren Anwendungssystemen benötigt werden?	[16]
Welche funktionale Domäne hat eine Aufgabe, d. h. welche Anwendungsbausteine unterstützen eine Aufgabe?	[16]
Ist im Informationssystem / in bestimmten Teilen des Informationssystems funktionale Integration gegeben, d. h. sind Funktionen, die in mehreren Anwendungssystemen benötigt werden, nur einmal implementiert und können aus anderen Anwendungssystemen aufgerufen werden?	[16]
Welche semantische Domäne hat ein Begriffssystem, d. h.. in welchen Anwendungsbausteinen wird das Begriffssystem benötigt?	[16]
Ist im Informationssystem / in bestimmten Teilen des Informationssystems semantische Integration gegeben, d. h. werden in unterschiedlichen Anwendungssystemen dieselben Begriffssysteme verwendet und werden Daten somit gleich interpretiert?	[16]
Welche Zugriffsdomänen besitzt ein <i>physisches Datenverarbeitungssystem</i> , d. h. welche <i>Anwendungssysteme</i> sind darauf installiert?	[16]
Welche <i>Anwendungssysteme</i> meines Informationssystems könnten abgeschafft werden, ohne dass Funktionalität verloren geht (funktionale Redundanz)?	[27]
Habe ich unnötige Kosten, weil verschiedene Nutzer in meiner Institution verschiedene <i>Anwendungssysteme</i> benutzen, um die gleiche Aufgabe zu erfüllen? Was sind kritische Unternehmensaufgaben und die Nutzung welcher <i>Anwendungssysteme</i> sollte vermieden werden?	[27]

E.B.D-1.11 Modellierungshinweise

Anwendungsbausteine:

- Das Vererbungsprinzip ist stets sorgfältig zu beachten. So ist stets zu beachten, dass z. B. eine Beziehung, die ein *Anwendungsbaustein* mit einem *Objekttyp* hat, auch für alle *Teilanwendungsbausteine* gegeben ist. Dies gilt auch dann, wenn einem *Anwendungsbaustein* erst später *Teilanwendungsbausteine* zugeordnet werden. Nicht immer wird jedoch davon auszugehen sein, dass alle Eigenschaften des *Anwendungsbausteins* auch für alle *Teilanwendungsbausteine* gelten. Es sollte also immer überprüft werden, ob wirklich nur die Eigenschaften bei dem *Anwendungsbaustein* belassen worden sind, die auch wirklich für alle *Teilanwendungsbausteine* gelten sollen. Diese Situation ergibt sich z. B., wenn *Anwendungsbausteine* eines Referenzmodells in ein Modell übernommen, dort aber zur präziseren Modellierung aufgeteilt werden. Grundsätzlich sollte also vermieden werden, übergeordneten *Anwendungsbausteinen* Beziehungen zuzuweisen. Zur Unterstützung der Modellierung bietet der 3LGM²-Baukasten eine Funktion, mit der alle oder eine Auswahl der vererbaren Eigenschaften und Beziehungen eines *Anwendungsbausteins* explizit auf alle *Teilanwendungsbausteine* übertragen und bei dem *Anwendungsbaustein* entfernt werden.
- Grundsätzlich sollte vermieden werden zu modellieren, dass ein *Anwendungsbaustein* aus genau einem *Teilanwendungsbaustein* besteht. Vielmehr ist es erforderlich, neben einem Teil des *Anwendungsbausteins* auch die anderen Teile explizit zu beschreiben. Dies bedeutet, dass die Verfeinerung eines

Anwendungsbausteins aus mindestens zwei (Teil-)Anwendungsbausteinen besteht, die gemeinsam den übergeordneten *Anwendungsbaustein* vollständig beschreiben.

Anwendungssysteme:

- *Anwendungssysteme* können durch die *Softwareprodukte*, durch deren Installation sie entstanden sind, voneinander abgegrenzt werden. Ein *Anwendungssystem* ist daher in der Regel ein betriebswirtschaftlich relevanter Vermögensgegenstand (Asset), da er die für seinen Betrieb erworbenen und tatsächlich nutzbaren *Softwareprodukte* repräsentiert. Im Übrigen ergibt sich daher aus der Modellierung eines *Anwendungssystems*, dass für die verwendete Software, wenn deren Nutzung entgeltpflichtig ist, auch tatsächlich Nutzungsgebühren fällig werden.
- Um ein *Anwendungssystem* an einem Arbeitsplatzrechner nutzen zu können ist bestimmte Software erforderlich, die dem Benutzer den Zugang zu dem *Anwendungssystem* vermittelt. Diese Software kann in dem vorhandenen Betriebssystem integriert sein oder es ist z. B. ein Internet-Browser nötig oder die Nutzung erfolgt über eine Terminalemulation und einen Terminalserver. Die Installation der Software, die den Zugang zu einem *Anwendungssystem* vermittelt, kann bei Bedarf als *zugangsvermittelndes Anwendungssystem* modelliert werden. Die Modellierung der Beziehungen zwischen *Anwendungssystemen* und *Aufgaben* erfolgt über die *Klasse Nutzung* und die *Assoziationsklasse Unterstützung*, die der *Assoziation* zwischen *Nutzung* und *Anwendungsbaustein* zugeordnet ist. Soll modelliert werden, dass ein *Anwendungssystem* den Zugang zur *Nutzung* eines anderen *Anwendungssystems*, welches eine *Aufgabe* unterstützt, vermittelt, so geschieht dies über die *Assoziationsklasse Unterstützung*. Die direkte *Unterstützung* einer *Aufgabe* durch ein *Anwendungssystem* wird modelliert über die *Klasse Nutzung*. Es kann hilfreich sein, eine *Kommunikationsverbindung* zu modellieren, die alle zur Erledigung der *Aufgabe* erforderlichen *Objekttypen* zwischen dem vermittelten *Anwendungssystem* und dem *zugangsvermittelnden Anwendungssystem* transportiert. Allerdings ist dies nicht unbedingt nötig, wenn auf der *Physischen Werkzeugebene* eine *Datenübertragungsverbindung* zwischen den zugrundeliegenden *physischen Datenverarbeitungssystemen* des vermittelten und des *zugangsvermittelnden Anwendungssystems* hergestellt wird. In Abbildung 11 sind die *Nutzung* und die *Unterstützung* der *Nutzung* von *Anwendungssystemen* bei der Erledigung von *Aufgaben* beispielhaft dargestellt: Das Patientenverwaltungssystem ist das vermittelte *Anwendungssystem*, „Terminalserver“ das *zugangsvermittelnde Anwendungssystem*. Die *Aufgabe* „administrative Patientenaufnahme“ wird durch das Patientenverwaltungssystem unterstützt und durch den „Terminalserver“ vermittelt. In der *Kommunikationsverbindung* zwischen KDMS und „Terminalserver“ könnte ein *Nachrichtentyp* alle *Objekttypen*, die interpretiert werden müssen, zum „Terminalserver“ transportieren.

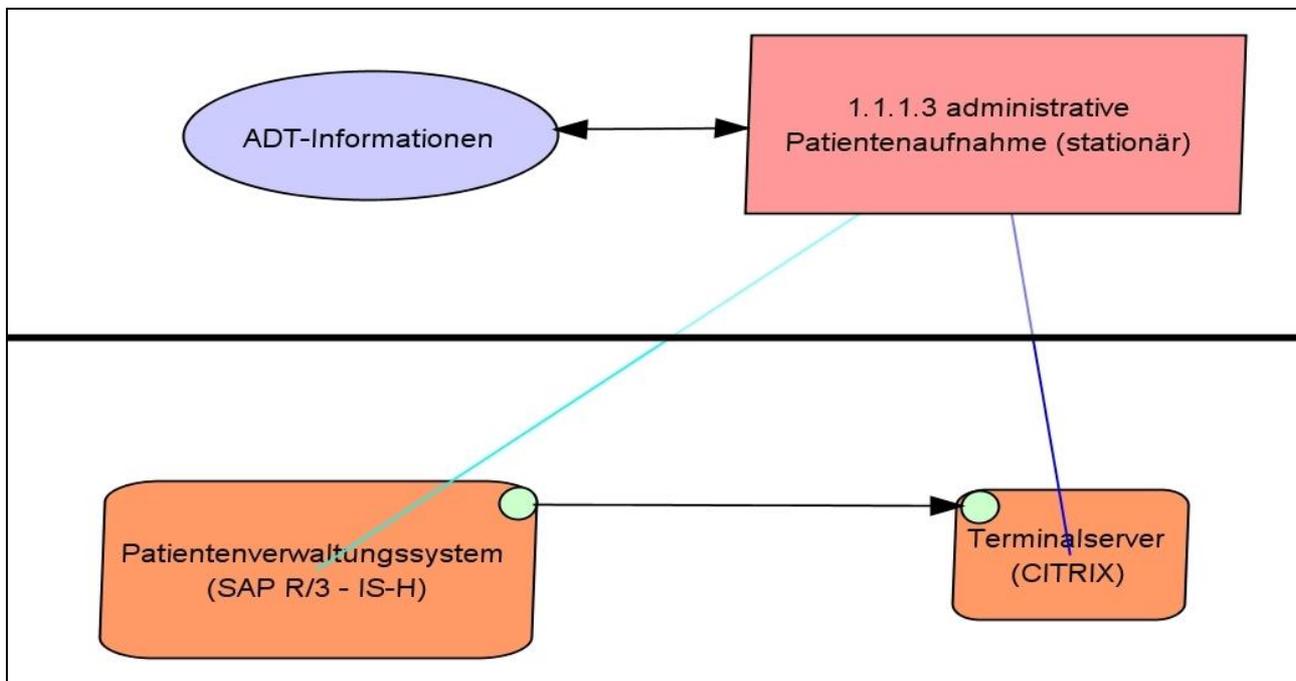


Abbildung 11: Modellierung zugangsvermittelnder Anwendungssysteme

- Die Modellierung von *Teilanwendungssystemen* sollte nicht ausschließlich dazu benutzt werden, eine Zugriffsmöglichkeit des vorgesehenen *Teilanwendungssystems* auf die Daten zu beschreiben, die in dem übergeordneten *Anwendungssystem* gespeichert sind. Die Teil-von-Beziehung kann vielmehr dazu benutzt werden zu beschreiben, dass auch das durch die Zusammenfassung der Teile sich ergebende Gebilde als Baustein des gesamten Informationssystems oder auch wiederum als Asset aufzufassen ist.
- Für die *Teilanwendungsbausteine* eines *Anwendungsbausteins* kann bei der Modellierung angenommen werden, dass sie unbeschränkt untereinander Daten austauschen können. Die Modellierung von *Kommunikationsverbindungen* zwischen den *Teilanwendungsbausteinen* desselben *Anwendungsbausteins* würde damit auch entfallen. Möchte man jedoch auch die Kommunikation innerhalb eines *Anwendungsbausteins* beschreiben, so empfiehlt es sich, für jede *Kommunikationsschnittstelle* des *Anwendungsbausteins* einen neuen *Anwendungsbaustein* mit zwei *Kommunikationsschnittstellen* – eine nach „innen“ und eine nach „außen“ – anzulegen.
- Wenn dargestellt werden soll, dass die unterschiedlichen Komponenten eines *Anwendungssystems* dadurch integriert werden, dass sie auf dieselbe Datenbank zugreifen, kann dies wie folgt modelliert werden: Die Komponenten werden als *Teilanwendungssysteme* modelliert. Insbesondere wird das aus der Datenbank und dem verwendeten Datenbankverwaltungssystem zusammengesetzte Datenbanksystem als ein *Teil-Anwendungssystem* modelliert; das Datenbankverwaltungssystem wird als *Softwareprodukt* modelliert. Über *Kommunikationsschnittstellen* und *Kommunikationsverbindungen* wird dann die Kommunikation der *Teil-Anwendungssysteme* untereinander modelliert.
- Der Modellierer hat stets darauf zu achten, dass einem Anwendungssystem nur solche *Physischen Datenverarbeitungssysteme* zugeordnet werden, die sich auch durch *Software* steuern lassen.

Organisationssysteme:

- Im Gegensatz zum *Anwendungssystem* ist es nicht immer einfach, ein *Organisationssystem* klar abzugrenzen und zu bezeichnen. Abgrenzungskriterien können z. B. die Zugehörigkeit zu einer *Organisationseinheit*, die Dominanz einer bestimmten Personengruppe (Rolle) oder aber die Erledigung bestimmter *Aufgaben* sein. Als Bezeichnung bietet sich oft die Form „Organisationssystem der/des/für ...“
- Im ersten Fall der Zugehörigkeit zu einer *Organisationseinheit* kann es sich beispielsweise um die angewendeten Regeln zum Dokumentenmanagement auf der Station AUG1 des Universitätsklinikums handeln. In diesem Fall könnte als Bezeichnung gewählt werden: „Organisationssystem des Dokumentenmanagements AUG1“. Im zweiten Fall kann es sich beispielsweise um Regeln zur Pflegedokumentation in der Klinik für Chirurgie 1 handeln. In diesem Fall könnte als Bezeichnung gewählt werden: „Organisationssystem der Pflegedokumentation Chirurgie 1“. Im dritten Fall kann es sich beispielsweise um das *Organisationssystem* zur Archivierung von Krankenakten handeln. In diesem Fall könnte als Bezeichnung gewählt werden: „Organisationssystem der Archivierung“ oder auch einfach z. B. „Krankenaktenarchiv“. Denkbar sind auch Kombinationen von Abgrenzungskriterien.
- Auf jeden Fall sollte beim Modellieren eines *personengesteuerten Informationsverarbeitungssystems* genau festgehalten werden, welches Kriterium bzw. welche Kombination von Kriterien in welcher Form zur Abgrenzung dienen. Ferner muss zwingend darauf geachtet werden, dass zwei *Organisationssysteme*, die nicht in einer Teil-von-Beziehung stehen (vgl. Kap. E.B.D), sich in einem Modell nicht inhaltlich überlagern.
- Die Menge der *Anwendungsbausteine* des *Informationssystems* einer *Organisation* kann durch ein *Organisationssystem* zusammengefasst und damit wieder als *Anwendungsbaustein* aufgefasst werden, z. B. dann, wenn das *Informationssystem* der *Organisation* als Teilsystem eines regionalen Gesundheitsinformationssystems aufgefasst wird [28].

Beziehungen zwischen Anwendungsbausteinen:

- Gelegentlich liegen Daten in einem rechnerbasierten *Anwendungssystem* vor und werden in einem anderen *Anwendungssystem* benötigt, werden aber zur Übermittlung zuerst ausgedruckt, dann wieder am Ziel abgetippt oder es wird ein Datenträger beschrieben, der am Ziel wieder eingelegt wird. In einer solchen Situation sollte keine direkte *Kommunikationsverbindung* zwischen *Kommunikationsschnittstellen* der beiden *Anwendungssysteme* modelliert werden. Vielmehr sollte der Aspekt des menschlichen Eingreifens in den Kommunikationsvorgang durch das Zwischenschalten mindestens eines *Organisationssystems* dargestellt werden. Darauf kann verzichtet werden, wenn auf Grund des Modellierungszwecks das menschliche Eingreifen als unbedeutend eingestuft wird.

E.B.D-2 Softwareprodukt

E.B.D-2.1 Beschreibung des Konzepts

Ein *Softwareprodukt* ist eine abgeschlossene, erworbene oder eigenentwickelte Software, die dediziert für die Erledigung bestimmter *Aufgaben* einer *Klasse* von *Organisationen* oder für den Austausch von Daten zwischen *Anwendungssystemen* bestimmt ist. Ein Betriebssystem wird z. B. dann als *Softwareprodukt* aufgefasst, wenn dargestellt werden soll, dass es die Mitarbeiter der Organisation durch bestimmte Funktionen direkt bei der Erledigung bestimmter *Aufgaben* unterstützt, ohne dass dabei zusätzliche Software erforderlich ist.

Während das *Softwareprodukt* als eine eher abstrakte Repräsentation von Regeln aufgefasst werden kann, wie bestimmte Arten von *Physischen Datenverarbeitungssystemen* arbeiten und zusammenwirken könnten (vgl. Definition *Anwendungsbaustein* in E.B.C-2), sind bei einem *Anwendungssystem* die Regeln konkretisiert und es ist explizit festgelegt, welche bestimmten *Physischen Datenverarbeitungssysteme* in einer bestimmten *Organisation* gesteuert werden und zusammenwirken sollen.

Ein *Softwareprodukt* wird entweder am Markt als Standardprodukt erworben oder gezielt für den Einsatz in einer Organisation entwickelt. Die Entwicklung des *Softwareprodukts* kann in der Organisation selbst oder aber durch ein anderes Unternehmen erfolgen.

E.B.D-2.2 Beziehung „dient als Basis für“ zu Anwendungssystem

Durch die Installation des *Softwareprodukts* entsteht ein *Anwendungssystem*. Werden bei der *Adaptation* des *Softwareprodukts* dessen Konstruktion nicht verändert und lediglich Tabelleneinträge vorgenommen, spricht man von *Adaptation durch Parametrierung* oder engl. *customizing*. In diesem Fall ist die ausführbare Software des *Anwendungsbausteins* identisch mit dem *Softwareprodukt*.

Ein und dasselbe *Softwareprodukt* kann in einer *Organisation* auch mehrfach installiert werden. Aus Sicht des Informationsmanagements und gerade im Hinblick auf den erforderlichen Integrationsaufwand muss jede dieser Installationen als ein eigenständiges *Anwendungssystem* aufgefasst werden; denn auch zwei *Anwendungssysteme* desselben *Softwareprodukts* benötigen oft auch aufwändige Kommunikationsschnittstellen, um einen Datenaustausch zwischen ihnen zu ermöglichen.

Ein *Softwareprodukt* muss nicht notwendigerweise eine Beziehung zu einem *Anwendungssystem* haben. Ohne diese Beziehung kann ein Katalog von z. B. am Markt verfügbaren oder im Rahmen einer Ausschreibung angebotenen *Softwareprodukte* modelliert werden, die bestimmte *Aufgaben* des *Informationssystems* der *Organisation* unterstützen könnten. Erst wenn es im Rahmen entsprechender Einführungsprojekte zur Installation eines der *Softwareprodukte* käme, würde eine Beziehung zu dem dann entstandenen *Anwendungssystem* zu modellieren sein.

E.B.D-2.3 Interebenenbeziehung „kann unterstützen“ zu Aufgabe

Das *Softwareprodukt* wurde entwickelt, um nach Installation die Erledigung bestimmter *Aufgaben* einer *Organisation* zu unterstützen. Bei der Installation (vgl. E.B.D-2.2) werden aber nicht notwendigerweise alle Möglichkeiten zur Unterstützung der Erledigung von *Aufgaben* in Betrieb genommen. Andererseits können aber auch Möglichkeiten zur Erledigung weiterer *Aufgaben* geschaffen werden.

Mit dieser *Assoziation* werden also alle *Aufgaben* bezeichnet, deren Erledigung ein durch Installation dieses *Softwareprodukts* entstehendes *Anwendungssystem* unterstützen könnte.

E.B.D-2.4 Fragen des Informationsmanagements und Unterstützung durch das 3LGM²

Fragen des Informationsmanagements	Verweis auf Text
Könnte auf bestimmte Anwendungssysteme in einem Informationssystem verzichtet werden, wenn das Potential der bereits eingesetzten Softwareprodukte, die Erledigung weiterer Aufgaben zu unterstützen, mehr ausgenutzt würde?	E.B.B-2.5
Wird das Softwareprodukt, das als Basis des Anwendungssystems A in dem Informationssystem des Krankenhauses K eingesetzt wird, auch für andere Anwendungssysteme im demselben Krankenhaus oder in anderen Krankenhäusern der Region eingesetzt?	E.B.B-2.5

Wie hoch sind die Lizenzkosten des Softwareprodukts, das als Basis des Anwendungssystems A in dem Informationssystem des Krankenhauses K eingesetzt wird? Wer sind die Ansprechpartner bei dem entsprechenden Hersteller/Lieferant?

Welche (am Markt) verfügbaren Softwareprodukte könnten zur Unterstützung der Erledigung einer bestimmten Menge von Aufgaben eingesetzt werden? E.B.B-2

Könnte auf bestimmte Anwendungssysteme in einem Informationssystem verzichtet werden, wenn ein bestimmtes anderes Softwareprodukt, das die Erledigung einer größeren Menge von Aufgaben unterstützen könnte, eingesetzt würde? E.B.B-2, E.B.D-1

E.B.D-3 Kommunikationsverbindung, Sende- und Empfangsschnittstelle

E.B.D-3.1 Beschreibung des Konzepts

Ein *Anwendungsbaustein* besitzt *Kommunikationsschnittstellen* um mit anderen *Anwendungsbausteinen* kommunizieren zu können.

Eine *Sendeschnittstelle* eines *Anwendungssystems* kann Daten an ein anderes *Anwendungssystem* versenden, das diese Daten über eine *Empfangsschnittstelle* empfangen kann. Wenn von einer *Sende-* zu einer *Empfangsschnittstelle* tatsächlich Daten versendet werden, dann werden diese *Kommunikationsschnittstellen* durch eine *Kommunikationsverbindung* verbunden, die von der *Sende-* zu der *Empfangsschnittstelle* weist.

Eine *Kommunikationsschnittstelle* kann für eine bestimmte Art von Daten *Sendeschnittstelle* und gleichzeitig für eine andere Art von Daten *Empfangsschnittstelle* sein.

Alle *Kommunikationsschnittstellen* eines *Anwendungsbausteins* gelten implizit als so untereinander verbunden, dass jede *Sendeschnittstelle* des *Anwendungsbausteins* auf alle Daten aller *Empfangsschnittstellen* des *Anwendungsbausteins* zugreifen kann.

E.B.D-3.2 Interebenenbeziehungen „kann senden“ und „kann empfangen“ von Sende- und Empfangsschnittstellen zu Objekttyp und „überträgt“ von Kommunikationsverbindung zu Objekttyp

Die *Assoziation* einer *Sendeschnittstelle* zu einem *Objekttyp* beschreibt, dass diese *Sendeschnittstelle* Daten versenden kann, die Informationen über *Objekte* repräsentieren, welche durch den *Objekttyp* beschrieben sind. Vereinfachend kann davon gesprochen werden, dass die *Sendeschnittstelle* den *Objekttyp* versenden kann.

Die *Assoziation* einer *Empfangsschnittstelle* zu einem *Objekttyp* beschreibt, dass diese *Empfangsschnittstelle* Daten empfangen kann, die Informationen über *Objekte* repräsentieren, welche durch den *Objekttyp* beschrieben sind. Vereinfachend kann davon gesprochen werden, dass die *Empfangsschnittstelle* den *Objekttyp* empfangen kann.

Die *Assoziation* einer *Kommunikationsverbindung* zu einem *Objekttyp* beschreibt, dass diese *Kommunikationsverbindung* Daten transportiert, die Informationen über *Objekte* repräsentieren, welche durch den *Objekttyp* beschrieben sind. Vereinfachend kann davon gesprochen werden, dass die *Kommunikationsverbindung* den *Objekttyp* von der *Sendeschnittstelle* zur *Empfangsschnittstelle* transportiert.

E.B.D-3.3 Fragen des Informationsmanagements und Unterstützung durch das 3LGM²

Fragen des Informationsmanagements	Verweis auf Text
Erfolgt die Kommunikation zwischen zwei bestimmten Anwendungssystemen umständlich durch den physischen Transport von DVDs, CDs oder gar per Papier, oder werden die Nachrichten komfortabel auf signalbasierten Wegen wie WLAN oder Kupferkabel transportiert?	E.B.E-5
Wenn ein Anwendungsbaustein Daten zu Objekten eines Objekttyps erhält und Daten zu Objekten desselben Objekttyps abgibt, müssen dann die Daten zwischen unterschiedlichen nicht signalbasierten Übertragungsmedien aufwändig umkopiert werden?	E.B.E-5
Wenn ein Anwendungsbaustein Daten zu Objekten eines Objekttyps erhält und Daten zu Objekten desselben Objekttyps abgibt, findet dann ein Medienbruch zwischen signalbasierten und nicht signalbasierten Übertragungsmedien mit aufwändiger Transkription statt?	E.B.E-5
Daten zu welchen Objekttypen werden zwischen zwei bestimmten Anwendungsbausteinen ausgetauscht?	E.B.B-1.6

Erhält ein bestimmter Anwendungsbaustein alle die Daten zu Objekttypen, welche die Aufgaben, die er unterstützt, interpretieren? E.B.D-1

Welche Objekttypen kann ein Anwendungsbaustein versenden oder empfangen?

E.B.D-1.5, E.B.D-3

E.B.D-3.4 Modellierungshinweise

- Soll bei einem *Anwendungsbaustein* explizit dargestellt werden, wie innerhalb des Bausteins Daten von einer *Empfangs-* zu einer *Sendeschnittstelle* geleitet und dabei möglicherweise (z. B. in einem Kommunikationsserver) umformatiert werden, kann dies wie in E.B.D-3.1 erwähnt, nicht durch eine (interne) *Kommunikationsverbindung* dargestellt werden. Zur Darstellung interner Kommunikation muss der *Anwendungsbaustein* vielmehr in *Teilanwendungsbausteine* aufgeteilt werden (siehe E.B.D-1), zwischen denen dann *Kommunikationsverbindungen* modelliert werden können.
- Bei einem *Anwendungsbaustein* kann für jeden *Objekttyp*, den er versenden/empfangen kann, eine besondere *Kommunikationsschnittstelle* modelliert werden. Alternativ kann auch für alle sendbaren *Objekttypen* eine einzige *Sende-* und für alle empfangbaren *Objekttypen* eine einzige *Empfangsschnittstelle* modelliert werden. Ferner kann eine einzige *Kommunikationsschnittstelle* modelliert werden, die als *Sende-* und *Empfangsschnittstelle* für alle versend- und empfangbaren *Objekttypen* dient. Es empfiehlt sich aber, auch *Kommunikationsschnittstellen* im Sinne von Assets (vgl. E.B.D-1.11) zu modellieren, d. h. dann eine *Kommunikationsschnittstelle* zu modellieren, wenn diese in der modellierten Form von dem jeweiligen Hersteller so hergestellt bzw. geliefert, von der einsetzenden Organisation bei Kauf bzw. Betrieb zu bezahlen ist und/oder beim Betrieb als solche zu betreiben bzw. zu warten ist.
- Die explizite Modellierung von *Kommunikationsverbindungen* zwischen *Kommunikationsschnittstellen* desselben *Anwendungsbausteins* ist nicht vorgesehen

E.B.D-4 Beispiel

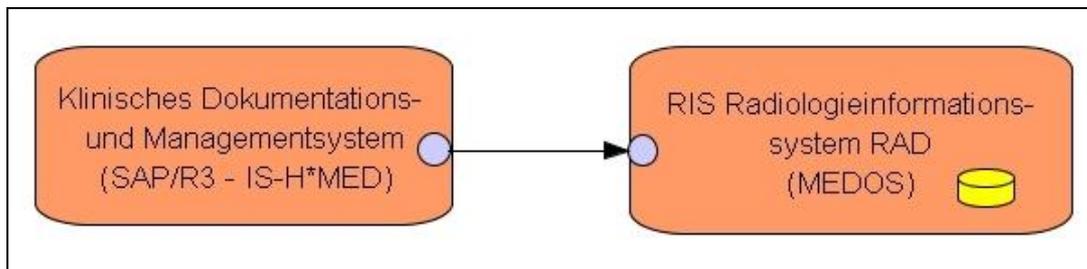


Abbildung 12: Darstellung von Kommunikationsverbindungen und Sende- und Empfangsschnittstellen

E.B.E Physische Werkzeugebene

Die physischen Werkzeuge, also insbesondere die *Physischen Datenverarbeitungssysteme*, werden in der *Physischen Werkzeugebene* zusammengefasst.

E.B.E-1 Physisches Datenverarbeitungssystem

E.B.E-1.1 Beschreibung des Konzepts

Siehe E.B.C-1.

E.B.E-1.2 Zerlegung von Physischen Datenverarbeitungssystemen

E.B.E-1.2.1 Konzept

Ein *Physisches Datenverarbeitungssystem* kann durch *Zerlegung* in seine *Unterdatenverarbeitungssysteme* zerlegt werden. Diese *Unterdatenverarbeitungssysteme* werden auch als *Teildatenverarbeitungssysteme* bezeichnet. Ein *Physisches Datenverarbeitungssystem* kann nur *Teildatenverarbeitungssystem* eines einzigen *Physischen Datenverarbeitungssystems* sein. Es findet keine Vererbung von *Gruppendatenverarbeitungssystemen* auf *Teildatenverarbeitungssysteme* statt.

Ein *Teildatenverarbeitungssystem* gilt als nicht funktionsfähig ohne das *Physische Datenverarbeitungssystem*, dessen Teil es ist. Beispielsweise ist eine Grafikkarte, die nicht Teil eines Rechners ist, nicht funktionsfähig. Für die *Teildatenverarbeitungssysteme* eines *Physischen Datenverarbeitungssystems* wird angenommen, dass sie unbeschränkt untereinander Daten austauschen können. Die Modellierung von *Datenübertragungsverbindungen* zwischen den *Teildatenverarbeitungssystemen* desselben *Physischen Datenverarbeitungssystems* ist daher nicht vorgesehen. (Vgl. differierendes Konzept in E.B.E-1.3.1)

E.B.E-1.2.2 Vererbung

Die *Teildatenverarbeitungssysteme* eines *Physischen Datenverarbeitungssystems* erben von diesem *Physischen Datenverarbeitungssystem* keine Eigenschaften oder Beziehungen!

E.B.E-1.2.3 Vergrößern

Beim Vergrößern eines *Physischen Datenverarbeitungssystems* werden seine *Teildatenverarbeitungssysteme* (vorübergehend) aus dem Modell entfernt. Dies bedeutet, dass auf die differenzierte Betrachtung, die durch die *Teildatenverarbeitungssysteme* gegeben ist, verzichtet wird. Die *Datenübertragungsverbindungs-Assoziationen* der *Teildatenverarbeitungssysteme* zu anderen *Physischen Datenverarbeitungssystemen* sowie die *Assoziationen* zu *Anwendungsbausteinen* werden dem *Physischen Datenverarbeitungssystem* zugeordnet, dessen Teil sie sind. Die anderen *Assoziationen* verschwinden mit der Vergrößerung.

E.B.E-1.2.4 Kompatibilität zu 3LGM² V1.x bzw. 3LGM² V2.x

Die bisherige Teil-von-Beziehung wurde häufig im Sinne der *Komposition* verwendet. Bei einer Konversion zu V3.x beachte bitte F.E.C-1.

E.B.E-1.3 Virtualisierung und Cluster

E.B.E-1.3.1 Konzept

Durch die *Assoziation* „virtualisiert“/“wird virtualisiert durch“ können *Physische Datenverarbeitungssysteme* mit anderen *Physischen Datenverarbeitungssystemen* in eine Beziehung gebracht werden. Über die Virtualisierungsbeziehung kann entweder ausgedrückt werden, dass ein oder mehrere *Physische Datenverarbeitungssysteme* zusammen ein *Physisches Datenverarbeitungssystem* simulieren oder dass ein *Physisches Datenverarbeitungssystem* ein oder mehrere *Physische Datenverarbeitungssysteme* simuliert. Im ersten Fall heißt das simulierte (virtuelle) *Physische Datenverarbeitungssystem* *Cluster*. Im zweiten Fall heißen die simulierten (virtuellen) *Physischen Datenverarbeitungssysteme* *virtuelle Maschinen*. Zur Veranschaulichung siehe Abbildung 13. In einem *Cluster* genügt beispielsweise die Verfügbarkeit eines zugehörigen Servers, um ein *Anwendungssystem* auszuführen. Nach außen hin kann *Cluster* wie ein einziger Server verwaltet werden. Im Gegensatz dazu können mit Hilfe *virtueller Maschinen* z. B. verschiedene Betriebssysteme oder verschiedene Instanzen eines Betriebssystems parallel auf einem Server ausgeführt werden. *Virtuelle Maschinen* und *Cluster* werden als *virtuelle Physische Datenverarbeitungssysteme* bezeichnet.

Eine *Physisches Datenverarbeitungssystem* kann *Mitglied* mehrerer *Cluster* sein.

Für die *Mitglieder* eines *Clusters* wird im Gegensatz zu E.B.D-1.3.1 nicht angenommen, dass sie allein aufgrund ihrer Mitgliedschaft unbeschränkt untereinander Daten austauschen könnten. Vielmehr sind die gegebenen Kommunikationsmöglichkeiten explizit durch *Datenübertragungsverbindungen* zu modellieren.

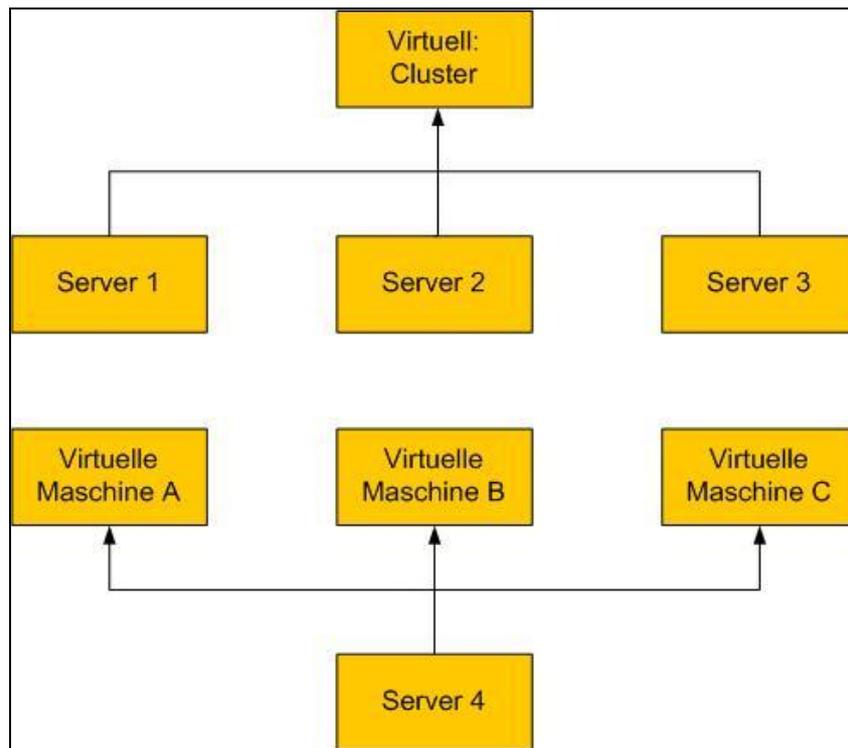


Abbildung 13. Oben: Beispiel für einen Cluster, der aus drei physischen Servern besteht. Unten: Auf einem physischen Server laufen drei virtuelle Maschinen.

Für die Beziehungen zu *Anwendungsbausteinen* sind aber folgende Besonderheiten des *Clusters* zu beachten (vgl. differierendes Konzept in E.B.D-1.3.2):

- Hat ein *Cluster C* eine Beziehung „wird für Funktionalität benötigt“ zu einem *Anwendungsbaustein A*, dann bedeutet die Gruppenbeziehung, dass bereits eines der *Mitglieder des Clusters C* ausreicht, um die Funktionalität von *A* zu gewährleisten. Auf diese Weise lässt sich *G* als ein *virtuelles Physisches Datenverarbeitungssystem*, also als virtuellen Server auffassen, dessen *Mitglieder* oder Komponenten die Funktionalität von *A* in redundanter Weise gewährleisten.
- Hat ein *Cluster C* eine Beziehung „wird für Speicherung benötigt“ zu einem *Anwendungsbaustein A*, dann bedeutet die Gruppenbeziehung, dass bereits eines der *Mitglieder des Clusters C* ausreicht, um die Speichermöglichkeit von *A* zu gewährleisten. Auf diese Weise lässt sich *G* als ein *virtuelles Physisches Datenverarbeitungssystem*, also als virtuellen Speicher auffassen, dessen *Mitglieder* oder Komponenten die Speichermöglichkeit von *A* in redundanter Weise gewährleisten.

E.B.E-1.3.2 Kompatibilität zu 3LGM² V1.x, 3LGM² V2.x

Die Virtualisierung hat in V1.x bzw. V2.x kein Pendant. Allerdings sind in der Modellierungspraxis einige der Teil-von-Beziehungen so interpretiert worden, um auf diese Weise z. B. virtuelle Maschinen darstellen zu können. Bei der Konvertierung muss im Einzelfall entschieden werden, wie die Teil-von-Beziehungen zu übersetzen sind.

Modellierungshinweis für V1.x: Wenn die Teil-von-Beziehung zur Modellierung von Gruppen verwendet wird, dann sollte ein Bausteintyp zugeordnet werden, dessen Bezeichnung mit „virtuell:...“ beginnt. Dann kann bei einer Konvertierung diese Information genutzt werden und im Modell ist die Interpretation klarer.

Siehe F.E.C-1.

E.B.E-1.4 Beziehung zur Assoziationsklasse Datenübertragungsverbindung („überträgt Daten zu“, „empfängt Daten von“)

Ein *Physisches Datenverarbeitungssystem* kann mit anderen *Physischen Datenverarbeitungssystemen* durch den Austausch von Daten kommunizieren. Für diese Kommunikation benötigen zwei *Physische Datenverarbeitungssysteme*

jeweils eine *Datenübertragungsverbindung*. Die *Datenübertragungsverbindung* wird durch eine *Assoziation* mit der *Assoziationsklasse Datenübertragungsverbindung* zwischen zwei *Physischen Datenverarbeitungssystemen* modelliert.

Bei Bedarf kann eine Schnittstelle durch die Modellierung eines entsprechenden *Physischen Datenverarbeitungssystems* dargestellt werden, das als Teil des *Physischen Datenverarbeitungssystems* modelliert wird, an das die *Datenübertragungsverbindung* durch die Schnittstelle „anzuschließen“ ist.

E.B.E-1.5 Beziehung „ist vom Typ“ zu Geräteklasse

Physische Datenverarbeitungssysteme können über *Geräteklassen* klassiert werden. Dadurch hat der Modellierer völlig freie Hand, unterschiedlichste Arten von *Physischen Datenverarbeitungssystemen* zu modellieren. Das Herstellen einer Beziehung von einem *Physischen Datenverarbeitungssystem* zu einer *Geräteklasse* bewirkt auch, dass das *Physische Datenverarbeitungssystem* graphisch mit dem Icon dargestellt ist, dass der *Geräteklasse* zugewiesen worden ist (falls eine solche Zuweisung erfolgte). Darüber hinaus wird die Beziehung von einem *Physischen Datenverarbeitungssystem* zu einer *Geräteklasse* nicht graphisch dargestellt. Ein *Physisches Datenverarbeitungssystem* kann nur zu höchstens einer *Geräteklasse* gehören.

E.B.E-1.6 Beziehung „befindet sich an“ zu Standort

Wenn sich ein *Physisches Datenverarbeitungssystem* dauerhaft an einem bestimmten Ort befindet, kann ihm dieser Ort als *Standort* zugewiesen werden. Insbesondere mobilen Geräten wie Tablet-PC oder PDA wird man möglicherweise keinen *Standort* zuweisen. Die Beziehung wird graphisch nicht dargestellt.

E.B.E-1.7 Interebenenbeziehung „wird für Funktionalität benötigt“ zu Anwendungsbaustein

Über die *Assoziation* „wird für Funktionalität benötigt“ werden einem *Physischen Datenverarbeitungssystem* die *Anwendungsbausteine* zugeordnet, die die Regeln vorgeben, mit denen das *Physische Datenverarbeitungssystem* für die Entgegennahme, Weiterleitung oder gezielte Veränderung von Daten gesteuert wird (vgl. Definition in E.B.C-1). Bei einem *Physischen Datenverarbeitungssystem*, dessen *Verarbeitung von Daten* mit Software gesteuert wird, handelt es sich um *Anwendungssysteme*, die auf diesem System installiert sind.

E.B.E-1.8 Interebenenbeziehung „wird für Speicherung benötigt“ zu Anwendungsbaustein

Über die *Assoziation* „wird für Speicherung benötigt“ werden einem *Physischen Datenverarbeitungssystem* die *Anwendungsbausteine* zugeordnet, die die Regeln vorgeben, mit denen das *Physische Datenverarbeitungssystem* für die Speicherung von Daten gesteuert wird (vgl. Definition in E.B.C-1). Bei einem *Physischen Datenverarbeitungssystem*, dessen *Verarbeitung von Daten* mit Software gesteuert wird, handelt es sich um *Anwendungssysteme*, die auf diesem System installiert sind.

E.B.E-1.9 Fragen des Informationsmanagements und Unterstützung durch das 3LGM²

Fragen des Informationsmanagements	Verweis auf Text
Welche und wie viele Rechner verschiedener Klassen (z. B. PC, Server, ...) sind in einem Informationssystem im Einsatz?	
Welche Maßnahmen zur Ausfallsicherheit eines Anwendungssystems sind durch Hardwaremaßnahmen getroffen worden? Welche Server können sich bei Ausfall gegenseitig ersetzen?	E.B.D-1
Welche virtuellen Server sind vorhanden, aus welchen Physischen Servern werden sie gebildet, den Betrieb welcher Anwendungssysteme ermöglichen sie?	E.B.E-1.3
Sind Server an unterschiedlichen Standorten installiert, die sich bei Ausfall gegenseitig ersetzen könnten?	E.B.E-3
Welche Gebäude(teile) werden durch welche Teile des Kommunikationsnetzes erreicht?	E.B.E-3
Gibt es redundante Netzverbindungen zwischen Gebäude(teile)n?	E.B.E-3
Gibt es einen zentralen, universellen Unternehmensdatenspeicher z. B. durch ein SAN oder werden die Daten der Anwendungssysteme (z. B. des PACS) in dedizierten Speichersystemen gespeichert?	E.B.D-1.9
Gibt es redundante Datenspeicher?	E.B.D-1.9

In welchen Gebäude(teile)n werden Daten auf Papier gelagert? Wie gelangen die Papier-Unterlagen (Akten) dorthin? E.B.E-3, E.B.E-4

Gibt es dort Möglichkeiten die Akten zu digitalisieren? E.B.E-3

Welche Zugriffsdomänen besitzt ein *Physisches Datenverarbeitungssystem*, d. h. welche Anwendungssysteme sind darauf installiert? [16], F.C.A-1.7

E.B.E-1.10 Modellierungshinweise

- In dem Modell eines Gesundheitsinformationssystems, das aus den Informationssystemen mehrerer Krankenhäuser bzw. anderer Versorgungseinrichtungen besteht, können die *Physischen Datenverarbeitungssysteme* jeder Versorgungseinrichtung auch durch die Kompositionsbeziehung zu einem einzigen *Physischen Datenverarbeitungssystem* je Versorgungseinrichtung zusammengefasst werden (siehe auch [28]). Wegen der Vererbungsregeln wird jedoch empfohlen, der so entstehenden *Komposition*, d. h. dem *Physischen Datenverarbeitungssystem*, das alle *Physischen Datenverarbeitungssysteme* der Versorgungseinrichtung zusammenfasst, keine Datenübertragungsverbindungen und keine Interebenenbeziehungen zu Elementen der *Logischen Werkzeugebene* zuzuordnen.
- Das Metamodell kennt keine Unterscheidung in rechnerbasierte oder nicht-rechnerbasierte *Physische Datenverarbeitungssysteme*. Wenn eine solche Unterscheidung in einem Modell gewünscht wird, können dafür entsprechende *Geräteklassen* angelegt und den *Physischen Datenverarbeitungssystemen* zugeordnet werden. Diesen *Geräteklassen* können dann auch spezifischere *Geräteklassen* (z. B. PC, Server bzw. Regalanlage, Bote) untergeordnet werden (siehe E.B.E-2.2).
- Kabel zwischen *Physischen Datenverarbeitungssystemen* werden in der Regel als *Datenübertragungsverbindungen*, d. h. als eher passive Komponenten eines Kommunikationsnetzes modelliert. Da diese *Assoziation* auch wieder eine *Klasse* ist, können bestimmte Eigenschaften dieser Kabelverbindung modelliert werden. Allerdings können Kabel oder Kabelabschnitte bzw. Stecker oder Kupplungen auch als eigene *Physische Datenverarbeitungssysteme* modelliert werden. Eine *Datenübertragungsverbindung* zwischen einem Stecker und einem Kabel oder zwischen einem Stecker und einer Kupplung müsste dann modelliert werden, um auszudrücken, dass die beiden Komponenten miteinander verbunden sind. Dann darf aber diese *Datenübertragungsverbindung* nicht physisch, d. h. als weiteres Kabel, sondern nur logisch in dem Sinne interpretiert werden, dass Daten z. B. von dem Stecker zu der Kupplung übertragen werden können.

E.B.E-2 Geräteklasse

E.B.E-2.1 Beschreibung des Konzepts

Physische Datenverarbeitungssysteme können über *Geräteklassen* klassiert werden. *Geräteklassen* können z. B. sein: Regal, Server, PC, aktive Netzkomponente, Hub etc. Im 3LGM²-Baukasten können den einzelnen *Bausteintypen* bestimmte graphische Symbole und andere Eigenschaften zugewiesen werden, die von den zugeordneten *Physischen Datenverarbeitungssystemen* übernommen werden können.

Geräteklassen werden in den graphischen Darstellungen nicht als eigene Objekte dargestellt.

E.B.E-2.2 Spezialisierung von Geräteklassen

E.B.E-2.2.1 Konzept

Eine *Geräteklasse* kann durch *Untergeräteklassen* spezialisiert werden. Eine *Geräteklasse* kann nur *Teilgeräteklasse* einer einzigen *Gruppengeräteklasse* sein.

Durch die *Spezialisierung* können die *Geräteklassen* und damit auch die zugeordneten *Physischen Datenverarbeitungssysteme* z. B. in rechnerbasierte und nicht-rechnerbasierte eingeteilt werden.

E.B.E-2.2.2 Vererbung

Die *Untergeräteklassen* einer *Geräteklasse* erben von dieser *Geräteklasse* keine Eigenschaften oder Beziehungen!

E.B.E-2.2.3 Ausblenden und Vergrößern

Bei dem Vergrößern einer *Geräteklasse* werden die *Untergeräteklassen* (vorübergehend) aus dem Modell entfernt. Die *Assoziationen* der *Untergeräteklassen* zu *Physischen Datenverarbeitungssystemen* werden der *Gruppengeräteklasse* zugeordnet, deren Spezialisierung sie sind. Die anderen *Assoziationen* verschwinden für den Zeitraum der Vergrößerung.

E.B.E-2.3 Beziehung „spezifiziert“ zu Physischem Datenverarbeitungssystem

Jeder *Geräteklasse* können beliebig viele *Physische Datenverarbeitungssysteme* zugeordnet werden.

E.B.E-2.4 Fragen des Informationsmanagements und Unterstützung durch das 3LGM²

Fragen des Informationsmanagements	Verweis auf Text
Welche und wie viele Geräte einer bestimmten Geräteklasse werden in einem Informationssystem eingesetzt? Wie viele PCs, wie viele Server sind vorhanden?	

E.B.E-2.5 Modellierungshinweise

- Umfang und Granularität der modellierten *Geräteklassen* orientieren sich klar am Modellierungszweck.
- Wenn die modellierten *Geräteklassen* zum Vergleich zwischen Informationssystemen verwendet werden sollen, ist sicherzustellen, dass in den entsprechenden Modellen dieselben *Geräteklassen* verwendet werden.

E.B.E-3 Standort

E.B.E-3.1 Beschreibung des Konzepts

Ein *Standort* ist eine geographische Position, ein Gebäude oder Gebäudeteil, ein bestimmter Raum oder eine anders zu beschreibende Position im Raum. So ist z. B. der Operationssaal O3 in der 4. Etage des Gebäudes Paracelsusstraße 17 ein *Standort*. Dieses Konzept darf nicht mit dem Konzept der *Organisationseinheit* verwechselt werden. Wenn z. B. CH101 und NCH01 die operativen Bereiche der Chirurgischen Klinik 1 und der Neurochirurgischen Klinik als *Organisationseinheit* bezeichnen, können sich beide *Organisationseinheiten* durchaus denselben Operationssaal O3 als *Standort* teilen.

E.B.E-3.2 Zerlegung von Standorten

E.B.E-3.2.1 Konzept

Ein *Standort* kann durch *Zerlegung* in *Teilstandorte* zerlegt werden.

Ein *Standort* kann nur *Teilstandort* eines einzigen *Standorts* sein.

E.B.E-3.2.2 Vererbung

Die *Teilstandorte* eines *Standorts* erben von diesem *Standort* keine Eigenschaften oder Beziehungen!

E.B.E-3.2.3 Vergrößern

Bei dem Vergrößern der *Teilstandorte* eines *Standorts* werden die *Teilstandorte* (vorübergehend) aus dem Modell entfernt. Dies bedeutet, dass auf die differenzierte Betrachtung, die durch die *Teilstandorte* gegeben ist, verzichtet wird. Die *Assoziationen* der *Teilstandorte* zu *Physischen Datenverarbeitungssystemen* werden dem *Standort* zugeordnet, dessen Teil sie sind. Die anderen *Assoziationen* verschwinden mit der Vergrößerung.

E.B.E-3.3 Beziehung „beherbergt“ zu Physischem Datenverarbeitungssystem

Wenn sich ein *Physisches Datenverarbeitungssystem* dauerhaft an einem bestimmten Ort befindet, kann ihm dieser Ort als *Standort* zugewiesen werden. Insbesondere mobilen Geräten wie Tablet-PC oder PDA wird man möglicherweise keinen *Standort* zuweisen. Die Beziehung wird graphisch nicht dargestellt.

E.B.E-3.4 Fragen des Informationsmanagements und Unterstützung durch das 3LGM²

Fragen des Informationsmanagements	Verweis auf Text
Welche Teile des Krankenhauses werden durch das Kommunikationsnetz erreicht?	
Welche Anwendungssysteme werden möglicherweise nicht mehr funktionieren, wenn es zu einem Großfeuer in einem bestimmten Gebäude kommt.	E.B.B-2, E.B.D-1, E.B.E-1
Welche Aufgaben können im Arztzimmer 3.112 auf der 3. Etage des Asklepiosweg 3 mit Rechnerunterstützung durchgeführt werden?	E.B.B-2, E.B.D-1, E.B.E-1

E.B.E-3.5 Modellierungshinweise

- Umfang und Granularität der modellierten *Standorte* orientieren sich klar am Modellierungszweck.

E.B.E-4 Datenübertragungsverbindung

E.B.E-4.1 Beschreibung des Konzepts

Zwischen zwei *Physischen Datenverarbeitungssystemen* können Daten über eine *Datenübertragungsverbindung* transportiert werden. Eine geordnete Folge von *Physischen Datenverarbeitungssystemen*, die jeweils durch eine *Datenübertragungsverbindung* verbunden sind und daher nacheinander durchlaufen werden können, wird als *Datenübertragungsstrecke* bezeichnet.

Grundsätzlich wird darauf verzichtet, bei den *Datenübertragungsverbindungen* zu beschreiben, welche Arten von Daten und damit auch, welche *Objektypen* dort transportiert werden können. Es wird vielmehr davon ausgegangen, dass dann, wenn eine *Datenübertragungsverbindung* besteht, dort grundsätzlich alle Daten transportiert werden können. Allerdings kann differenziert werden, mit welchen Trägern (siehe E.B.E-4.3) die Übertragung der Daten erfolgen kann.

E.B.E-4.2 Beziehung zu Physischem Datenverarbeitungssystem („überträgt Daten zu“, „empfängt Daten von“)

Eine *Datenübertragungsverbindung* verbindet genau zwei *Physische Datenverarbeitungssysteme*. Die Verbindung ist eine gerichtete Beziehung, d. h. das Daten von einem bestimmten *Physischen Datenverarbeitungssystem* zu einem anderen bestimmten *Physischen Datenverarbeitungssystem* transportiert werden können. *Datenübertragungsverbindungen* können sowohl einseitig als auch beidseitig gerichtet sein.

Zwischen zwei *Physischen Datenverarbeitungssystemen* können mehrere *Datenübertragungsverbindungen* modelliert werden.

E.B.E-4.3 Beziehung „überträgt“ zu Übertragungsmedium

Eine *Datenübertragungsverbindung* lässt sich durch die Zuordnung von *Übertragungsmedien* im Hinblick auf die Art der Datenübertragung näher beschreiben. Eine *Datenübertragungsverbindung* kann über beliebig viele Übertragungsmedien übertragen.

Ein Übertragungsmedium kann signalbasiert oder nicht signalbasiert sein (boolean-Wert). Daraus ergibt sich, dass eine *Datenübertragungsverbindung* gleichzeitig als „signalbasiert“ und auch als „nicht signalbasiert“ gekennzeichnet sein kann.

E.B.E-4.4 Fragen des Informationsmanagements und Unterstützung durch den 3LGM²-Baukasten

Fragen des Informationsmanagements	Verweis auf Text
Müssen zwischen zwei bestimmten <i>Physischen Datenverarbeitungssystemen</i> die Daten physisch transportiert werden (z. B. auf Disketten oder auf Papier) oder werden die Daten automatisch und ohne Eingreifen durch eine Transportperson durch eine Signalübertragung übermittelt?	E.B.E-1
Erfolgt die Signalübertragung über ein Kabel oder drahtlos, z. B. mit Bluetooth oder WLAN?	E.B.E-5
Zwischen welchen <i>Physischen Datenverarbeitungssystemen</i> werden die Daten physisch transportiert (z. B.	E.B.E-1

auf Disketten oder auf Papier) und wo werden die Daten automatisch und ohne Eingreifen durch eine Transportperson durch eine Signalübertragung übermittelt?

Welche (Gruppen von) *Physischen Datenverarbeitungssysteme(n)* können keine Daten mit anderen *Physischen Datenverarbeitungssystemen* austauschen? E.B.E-1

In welcher Form erfolgt die Anbindung des Intranets der Organisation mit dem Internet? E.B.E-4

Welche Kommunikation zwischen Anwendungssystemen wird unterbrochen, wenn eine bestimmte Datenübertragungsverbindung ausfällt? E.B.D-1

Auf welche Datenübertragungsverbindungen ist die Erledigung einer bestimmten Aufgabe angewiesen (damit die erforderlichen Daten vorhanden und die erzeugten Daten gespeichert werden können)? E.B.D-1, E.B.E-1

E.B.E-4.5 Modellierungshinweise

- Eine *Datenübertragungsverbindung* ist auch zwischen rechnerbasierten *Physischen Datenverarbeitungssystemen* nicht notwendigerweise ein Kabel und umgekehrt. Die *Datenübertragungsverbindung* drückt vielmehr lediglich aus, dass zwischen den beiden Systemen Daten fließen können. Möglicherweise erfolgt dies wie z. B. bei Stecker und Kupplung über einen unmittelbaren Kontakt, sodass es in einem solchen Fall für die *Datenübertragungsverbindung* keine physische Entsprechung gibt. (vgl. E.B.E-1.10)

E.B.E-5 Übertragungsmedium

E.B.E-5.1 Beschreibung des Konzepts

Ein *Übertragungsmedium* beschreibt, auf welche Art Daten auf einer *Datenübertragungsverbindung* übermittelt werden können. Beispielsweise kann die Datenübertragung mit Papier, CD, Kabel oder per Funk erfolgen. Grundsätzlich lassen sich *Daten* mit nicht-gegenständlichen *Signalen* oder mit *gegenständlichen Übertragungsmedien* übertragen. Ein *Übertragungsmedium* kann somit signalbasiert oder nicht signalbasiert sein (boolean-Wert), d. h. auch beides gleichzeitig ist möglich.

Übertragungsmedien mit nicht-gegenständlichen *Signalen* sind z. B. metallener Kontakt, Kupferkabel verschiedener Spezifikationen, Lichtwellenleiter, WLAN, Bluetooth usw. Ein *gegenständliches Übertragungsmedium* ist ein transportables physisches Medium, auf dem Daten für kurze oder lange Zeit ihren physischen Ausdruck finden. Dazu gehören z. B. Papierbogen, Film, Karteikarte, Diskette, Chipkarte, CD, Memory Stick usw.

E.B.E-5.2 Beziehung „wird genutzt von“ zu Datenübertragungsverbindung

Jedes *Übertragungsmedium* kann beliebig vielen *Datenübertragungsverbindungen* zugeordnet werden und beschreibt dann, auf welche Weise die Datenübertragung an dieser Stelle erfolgen kann.

E.B.E-5.3 Fragen des Informationsmanagements und Unterstützung durch den 3LGM²Baukasten

Siehe bei Datenübertragungsverbindungen E.B.E-4.4

E.B.E-5.4 Modellierungshinweise

- In den meisten Fällen werden nur Computer oder ähnliche *softwaregesteuerte Physische Datenverarbeitungssysteme* durch *Datenübertragungsverbindungen* für nicht-gegenständliche *Signale* verbunden sein. Denkbar ist aber auch, dass zwischen dem Inventar eines Archivs und dem einer Station Daten über ein Faxgerät übertragen werden. Dieses Faxgerät würde das gegenständliche Papier in *Signale* konvertieren bzw. umgekehrt. So wäre zu erklären, dass zwischen den beiden *personengesteuerten Physischen Datenverarbeitungssystemen*, die das jeweilige Inventar repräsentieren, *Signale* und nicht gegenständliches Papier ausgetauscht wird. Für das Informationsmanagement wird es in vielen Fällen ausreichen, wenn man bei einer Auswertung alle solchen Konvertierungspunkte auf einer Datenübertragungsstrecke identifizieren kann, ohne die konvertierenden *Physischen Datenverarbeitungssysteme* im Einzelfall erkennen zu können.
- Für Übertragungs- und Netzwerkprotokolle sind im 3LGM² keine expliziten *Klassen* vorgesehen. Sie können jedoch durch eine entsprechende Ausdifferenzierung der auf *Signalen* basierenden *Übertragungsmedien* dargestellt werden.

- Als *gegenständlicher Datenträger* kann z. B. der Transportdienst der Zivildienstleistenden aufgefasst werden. Diese bewegen dann z. B. beschriebenes Papier, belichtete Filme, CDs, Disketten, Bänder u.ä. und transportieren auf diese Weise die auf den Datenträgern abgelegten Daten. *Gegenständliche Datenträger* werden vorrangig bei der Datenübertragung zwischen *personengesteuerten Physischen Datenverarbeitungssystemen* modelliert, da meist von einem Eingreifen und einer aktiven Beteiligung von Menschen auszugehen ist. Allerdings kann die Modellierung der Datenübertragung mit *gegenständlichen Datenträgern* auch zwischen *softwaregesteuerten Physischen Datenverarbeitungssystemen* sinnvoll sein. Für die Zwecke des Informationsmanagements kann in den meisten Fällen darauf verzichtet werden, z. B. das Entnehmen einer CD aus einem PC bzw. das Einlegen der CD in einen PC explizit zu modellieren. Allerdings ist auch das vollautomatische und softwaregesteuerte Transportieren von CD von einem softwaregesteuerten Speicherbaustein zu einem anderen mit Hilfe eines Roboters gelegentlich Realität.

E.B.E-6 Beispiel

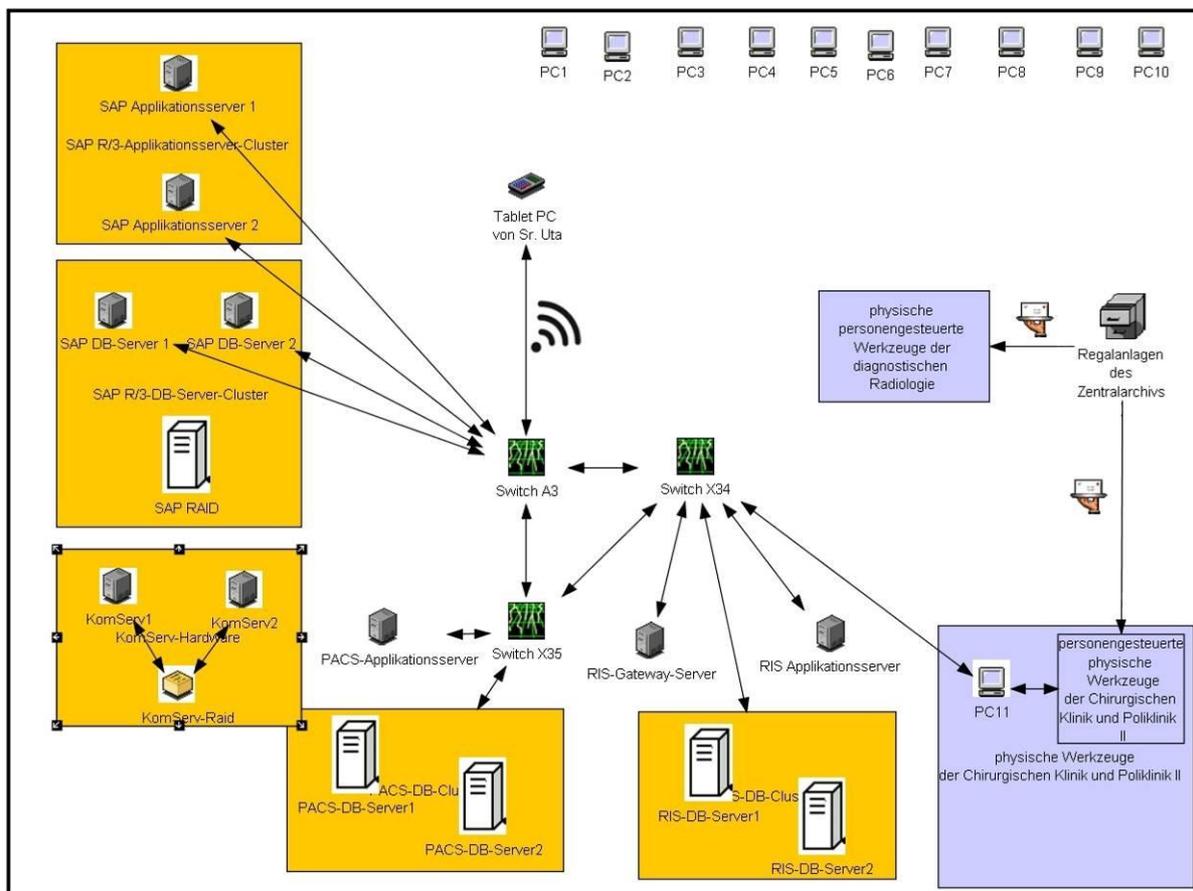


Abbildung 14: Beispiel einer *Physischen Werkzeugebene*

Abbildung 14: zeigt ein Beispiel einer *Physischen Werkzeugebene*. In diesem Beispiel gibt es *software- und personengesteuerte Physische Datenverarbeitungssysteme*. Die *personengesteuerten Physischen Datenverarbeitungssysteme* sind hellblau unterlegt. Aus dieser Darstellung ergibt sich u.a.:

- Es gibt einen Transportdienst, der eine *Datenübertragungsverbindung für gegenständliche Dokumententräger* von den Regalanlagen des Zentralarchivs zur Diagnostischen Radiologie realisiert, wobei das *Übertragungsmedium* Papier ist (symbolisiert durch ). Ebenfalls auf Papier können Daten von den Regalanlagen des Zentralarchivs zu den *personengesteuerten Physischen Datenverarbeitungssystemen* in der Chirurgischen Klinik und Poliklinik transportiert werden.

- Zu den physischen Werkzeugen der Chirurgischen Klinik und Poliklinik II (CHII) zählt auch der Arbeitsplatzrechner PC11. Er ist über eine *Datenübertragungsverbindung für nicht-gegenständliche Signale*, die durch Kabel realisiert ist (*Datenübertragungsverbindungen* dieses *Übertragungsmediums* sind in diesem Beispiel nicht näher mit einem Symbol gekennzeichnet), mit einem Switch des Kommunikationsnetzes verbunden. Über eine *Datenübertragungsverbindung für gegenständliche Dokumententräger* auf der Basis von Papier ist er mit den personengesteuerten physischen Werkzeugen der CHII verbunden. Dies bedeutet, dass eine Person die Daten von dem *Dokumententräger* abliest und in den Arbeitsplatzrechner eingibt. Dieser Arbeitsplatzrechner ist aber auch mit dem Arbeitsplatzrechner PC5 verbunden. Hier werden Disketten transportiert, die von Personen in das entsprechende Laufwerk eingelegt bzw. dort entnommen werden.
- Der Tablet-PC von Sr. Uta ist über eine drahtlose *Datenübertragungsverbindung* für nicht-gegenständliche *Signale* mit dem Kommunikationsnetz verbunden (📶). Die hierfür erforderlichen Sende- und Empfangsgeräte sind nicht modelliert worden.
- Für eine größere Anzahl von Arbeitsplatzrechnern ist nicht modelliert worden, in welcher Form sie mit anderen *Physischen Datenverarbeitungssystemen* verbunden sind. Wenn tatsächlich keine *Datenübertragungsverbindungen* für nicht-gegenständliche *Signale* z. B. mit dem Kommunikationsnetz existieren, so wird es wahrscheinlich doch Verbindungen mit *personengesteuerten Physischen Datenverarbeitungssystemen* geben. Das Modell sollte daher entsprechend komplettiert werden.

E.C Erweiterungen des Metamodells bzw. Modifikationen des Metamodells

E.C.A Nachrichtenorientierte Kommunikation mit Nachrichtentypen: 3LGM²-M

E.C.A-1 Überblick

Das Basis-Metamodell 3LGM²-B kennt weder *Nachrichtentypen* noch *Kommunikationsstandards*, sondern beschreibt die Kommunikation zwischen *Anwendungsbausteinen* unmittelbar durch die Angabe der *Objekttypen*, zu denen Daten bzw. *Nachrichten* über die *Kommunikationsverbindungen* gesendet werden. Wenn in einem Modell die Kommunikation genauer beschrieben und z. B. dargestellt werden soll, dass die Kommunikation zwischen bestimmten *Anwendungsbausteinen* auf der Basis von HL7, zwischen anderen auf der Basis von DICOM und an anderen Stellen möglicherweise mit proprietären Nachrichtentypen (Message Types) erfolgt, dann kann die Erweiterung 3LGM²-M verwendet werden.

3LGM²-M entsteht aus 3LGM²-B im Wesentlichen durch die Hinzunahme der *Klassen Nachrichtentyp* und *Kommunikationsstandard*. In den Abschnitten E.C.A-2 bis E.C.A-7 werden diese *Klassen* und weitere damit verbundene Ergänzungen erläutert. In Abschnitt E.C.A-8 wird erläutert, wie ein 3LGM²-B-Modell in ein 3LGM²-M-Modell konvertiert werden kann und umgekehrt.

3LGM²-M ist in Abbildung 15 in UML-Notation dargestellt.

E.C.A-2 Nachrichtentyp

E.C.A-2.1 Beschreibung des Konzepts

Eine *Nachricht (Message)* ist eine Menge von Daten, die zum Zwecke der Kommunikation zwischen *Anwendungsbausteinen* zusammengestellt wurde und als Einheit betrachtet und kommuniziert wird. Ein *Nachrichtentyp* beschreibt eine *Klasse* gleichartiger *Nachrichten* und legt fest, Daten welcher *Objektypen* bei einer Nachricht des Nachrichtentyps transportiert werden. Die *Anwendungsbausteine* können durchaus *Organisationssysteme* sein und die Kommunikation muss nicht notwendigerweise elektronisch und digital erfolgen. *Nachrichtentypen* sind demzufolge z. B. Aufnahmebogen, Laborbefund, Arztbrief oder „ADT Message“ und „ORU Observational Results (Unsolicited)“ aus dem Kommunikationsstandard HL7 V2.3.

E.C.A-2.2 Beziehung „gehört zu“ zu Kommunikationsstandard

Jeder *Nachrichtentyp* gehört zu einem *Kommunikationsstandard*; so können *Nachrichtentypen* gruppiert werden. Gehören *Nachrichtentypen* nicht zu einem der bekannten *Kommunikationsstandards* wie HL7 oder DICOM oder sind sie gar völlig proprietär, kann z. B. der *Kommunikationsstandard* „proprietäre Nachrichtentypen des Krankenhauses A“ angelegt und zugeordnet werden.

E.C.A-2.3 Beziehung zu „wird übertragen von“ zu Kommunikationsverbindung, „kann gesendet werden von“ und „kann empfangen werden von“ zu Sende- und Empfangsschnittstelle

Die *Kommunikationsschnittstellen* eines *Anwendungsbausteins* ermöglichen es dem *Anwendungsbaustein*, *Nachrichten* bestimmter *Nachrichtentypen* zu versenden bzw. zu empfangen. Besteht zwischen einer *Empfangs-* und einer *Sendeschnittstelle* eine *Kommunikationsverbindung*, dann kann modelliert werden, zu welchen *Nachrichtentypen* tatsächlich *Nachrichten* zwischen den beiden *Kommunikationsschnittstellen* kommuniziert werden.

E.C.A-2.4 Interebenenbeziehung „kommuniziert“ zu Objektyp

Die in einem *Nachrichtentyp* zusammengestellten Daten repräsentieren Informationen über *Objekte*, die durch *Objektypen* beschrieben sind. Ein *Nachrichtentyp* stellt daher in der Regel Daten mehrerer *Objekte* zusammen, d. h. er kommuniziert mehrere *Objektypen*. Beispielsweise kommuniziert der *Nachrichtentyp* „ADT Message“ sowohl den *Objektyp* Patient als auch den *Objektyp* Behandlungsfall (siehe Abbildung 6).

E.C.A-2.5 Fragen des Informationsmanagements und Unterstützung durch das 3LGM²

Fragen des Informationsmanagements	Verweis auf Text
Daten zu welchen Objektypen werden von Nachrichten eines bestimmten Nachrichtentyps transportiert?	E.C.A-4
Zu welchen Objektypen werden zwischen zwei bestimmten Anwendungsbausteinen Nachrichten ausgetauscht?	E.C.A-4
Welchen Teil eines bestimmten Kommunikationsstandards (z. B. HL7) unterstützt eine bestimmte Kommunikationsschnittstelle? Nachrichten welcher Nachrichtentypen kann sie senden/empfangen?	E.C.A-3
Wie uneinheitlich und komplex ist die Kommunikation? Mit wie vielen und welchen Nachrichtentypen wird ein bestimmter Objektyp versendet?	E.C.A-4

E.C.A-2.6 Modellierungshinweise

- Im Wesentlichen ist ein *Nachrichtentyp* eine Menge von *Objektypen*; nämlich die Menge der *Objektypen*, deren Daten bei einer *Nachricht* des betreffenden *Nachrichtentyps* versendet werden. Es kann aber sinnvoll sein, für ein und dieselbe Menge von *Objektypen* auch unterschiedliche *Nachrichtentypen* zu modellieren. Damit kann ausgedrückt werden, dass zwar Daten zu *Objekten* derselben *Objektypen* transportiert werden, dies aber z. B. auf Grund unterschiedlicher *Kommunikationsstandards* in unterschiedlicher Art und Weise geschieht.
- Zusammenfassungen von Daten werden in vielen Kontexten auch als *Dokumente* bezeichnet, Klassen solcher *Dokumente* als *Dokumentenarten* oder *-typen*. Im Krankenhaus sind z. B. Aufnahmebogen, Laborbefund oder Arztbrief solche *Dokumententypen*. Bei der Verwendung des Begriffs *Dokument* bzw. *Dokumententyp* wird

häufig impliziert, dass diese Zusammenfassungen von Daten auf Papier als Träger erfolgen. Auch solche *Dokumententypen* bzw. *-typen* werden in 3LGM² als *Nachrichtentypen* modelliert. Bei Bedarf können solche *Nachrichtentypen* z. B. einem *Kommunikationsstandard* „Dokumententypen des Krankenhauses A“ zugeordnet werden.

- In einem 3LGM²-Modell werden (Zusammenfassungen von) Daten auf der *Logischen Werkzeugebene* nur dann explizit modelliert, nämlich als *Nachrichtentyp*, wenn sie zwischen *Anwendungsbausteinen* kommuniziert werden. Dies ergibt sich aus dem Ziel, darzustellen, aus welchen Bausteinen, d. h. Anwendungsbausteinen, sich das gesamte Informationssystem zusammensetzt, nicht aber, wie diese Anwendungsbausteine intern aufgebaut sind. Entsprechend wird auch ein *Dokumententyp* nur dann als *Nachrichtentyp* modelliert, wenn er zwischen unterschiedlichen *Anwendungsbausteinen* kommuniziert wird. Auf die Modellierung von *Dokumententypen*, die lediglich innerhalb eines *Anwendungsbausteins* relevant sind, wird verzichtet und kann verzichtet werden. Vgl. hierzu auch E.B.D-1.2.

E.C.A-3 Kommunikationsstandard

E.C.A-3.1 Konzept

Ein *Kommunikationsstandard* beschreibt im Allgemeinen, mit welchem Datenformat zu welchen Ereignissen *Nachrichten* übertragen werden. In 3LGM²-M beschränkt sich das Konzept auf die Beschreibung der Formate durch Verweis auf entsprechende *Nachrichtentypen* und verzichtet auf die Modellierung von Ereignissen. Ein *Kommunikationsstandard* fasst also im 3LGM²-M mehrere *Nachrichtentypen* zu einer bestimmten Gruppe zusammen.

Kommunikationsstandards sind z. B. HL7 oder DICOM. *Nachrichtentypen*, die ein Hersteller proprietär für ein bestimmtes *Softwareprodukt* definiert hat, können z. B. mit dem *Kommunikationsstandard* „proprietäre Nachrichtentypen des Herstellers A“ angelegt und zusammengefasst werden.

E.C.A-3.2 Beziehung „enthält“ zu Nachrichtentyp

Einem *Kommunikationsstandard* können beliebig viele *Nachrichtentypen* zugeordnet werden.

E.C.A-3.2.1 Fragen des Informationsmanagements

Fragen des Informationsmanagements	Verweis auf Text
Welche Kommunikationsstandards werden in dem Informationssystem verwendet?	
Von welchen Anwendungssystemen wird ein bestimmter Kommunikationsstandard verwendet?	E.C.A-5
Wie uneinheitlich und komplex ist die Kommunikation? Mit wie vielen und welchen Kommunikationsstandards wird ein bestimmter Objekttyp versendet?	E.C.A-4

E.C.A-4 Objekttyp

Dieses Konzept ist in Abschnitt E.B.B-1 bereits detailliert beschrieben. Für die Erweiterung 3LGM²-M sind jedoch die folgenden Ergänzungen (bei kleineren Ergänzungen **fett** markiert) erforderlich.

E.C.A-4.1 Zerlegung und Spezialisierung

E.C.A-4.1.1 Vererbung

Wie in E.B.B-1.3.2 beschrieben, drückt eine Vererbung bei Objekttypen aus, dass alle Eigenschaften des *Gruppenobjekttyps* auch für die *Objekttypen* als gegeben angesehen werden sollen, die Bestandteile dieses *Gruppenobjekttyps* sind.

Alle Eigenschaften und *Assoziationen*, die dem *Gruppenobjekttyp* zugewiesen werden, werden auch für die *Unterojekttypen* als gegeben angesehen, d. h. die Eigenschaften und *Assoziationen* werden vererbt. Dies bedeutet, dass alle Beziehungen zu *Aufgaben*, *Anwendungsbausteinen* und **zu *Nachrichtentypen***, die für den *Gruppenobjekttyp* modelliert sind, auch als für alle seine *Unterojekttypen* modelliert gelten, also vererbt werden.

E.C.A-4.1.2 Vergrößern

Bei dem Vergrößern eines *Gruppenobjekttyps* werden seine *Unterojekttypen* (vorübergehend) aus dem Modell entfernt. Dies bedeutet, dass auf die differenzierte Betrachtung, die durch die *Unterojekttypen* gegeben ist, verzichtet

wird. Die *Assoziationen* der *Unterbjekttypen* zu *Aufgaben*, *Anwendungsbausteinen* **und** *Nachrichtentypen* werden dann für die Dauer der Vergrößerung vollständig dem *Gruppenobjekttyp* zugeordnet.

E.C.A-4.2 Interebenenbeziehung „wird kommuniziert in“ zu Nachrichtentyp

Werden die Daten über *Objekte*, auf die ein bestimmter *Objekttyp* verweist, in mehreren *Anwendungsbausteinen* gespeichert, dann müssen diese Daten zwischen diesen *Anwendungsbausteinen* auch ausgetauscht, d. h. kommuniziert werden. Darüber hinaus werden die Daten ggf. auch in *Anwendungsbausteinen* benötigt, in denen sie nicht gespeichert werden. Dies kann z. B. dann der Fall sein, wenn ein solcher *Anwendungsbaustein* eine *Aufgabe* unterstützt, die den auf die Daten verweisenden *Objekttyp* interpretiert. Zwischen den *Anwendungsbausteinen* werden die Daten, auf die ein bestimmter *Objekttyp* verweist, mit Nachrichten kommuniziert. Die *Assoziation* eines *Objekttyps* zu einem oder mehreren *Nachrichtentypen* drückt aus, dass Nachrichten dieser *Nachrichtentypen* Daten über *Objekte*, auf die der *Objekttyp* verweist, kommuniziert. Kurz: der *Objekttyp* wird durch die *Nachrichtentypen* kommuniziert.

E.C.A-5 Anwendungsbaustein

Dieses Konzept ist in Abschnitt E.B.D-1 bereits detailliert beschrieben. Für die Erweiterung 3LGM²-M sind jedoch folgende Ergänzungen erforderlich:

E.C.A-5.1 Modellierungshinweise

- *Kommunikationsverbindungen* zwischen *Anwendungssystemen* und *Organisationssystemen*: Wenn *Nachrichten* von einem *Organisationssystem* an ein *Anwendungssystem* gesendet werden bedeutet dies, das Daten, die auf nicht-rechnerbasierten Medien vorliegen, von Menschen in ein rechnerbasiertes *Anwendungssystem* eingegeben werden. Dies ist zum Beispiel dann der Fall, wenn ein auf Papier vorhandenes Dokument eingetippt, ein (aufgezeichnetes) Diktat geschrieben oder eine CD von Hand in ein Lesegerät eingelegt wird, damit die darauf enthaltenen Daten von dem *Anwendungssystem* gelesen werden können. Umgekehrt bedeutet das Senden von *Nachrichten* von einem *Anwendungssystem* an ein *Organisationssystem*, dass Daten, die in einem rechnerbasierten *Anwendungssystem* gespeichert sind, Menschen in einer nicht-rechnerbasierten Umgebung zur Verfügung gestellt werden. Dies ist zum Beispiel dann der Fall, wenn Daten auf dem Bildschirm angezeigt, auf dem Drucker ausgedruckt, in Form akustischer Signale abgehört (Musik, Text) oder auf einer DVD gespeichert werden und diese Daten anschließend von Menschen innerhalb des *Organisationssystems* weiterverarbeitet werden. In diesem Fall sollte weder ein *Nachrichtentyp* zur Modellierung verwendet werden, der angibt, die Nachricht wäre (bereits oder noch) in der digitalen Form, in der die in ihr enthaltenen Daten in dem *Anwendungssystem* gespeichert werden, noch ein *Nachrichtentyp* verwendet werden, der angibt, die Nachricht wäre (bereits oder noch) in der Form, in der die in ihr enthaltenen Daten in dem *Organisationssystem* gespeichert werden. Es empfiehlt sich vielmehr, zunächst einen *Kommunikationsstandard* zu modellieren, der Ein- bzw. Ausgaben in *Anwendungssysteme* beschreibt (z. B. „Eingaben in Anwendungssysteme“, „Ausgaben aus Anwendungssystemen“ oder auch „Ein-/Ausgaben in/aus Anwendungssystemen“); es können bei Bedarf auch *Kommunikationsstandards* separat für jedes betroffene *Anwendungssystem* modelliert werden (z. B. „Eingaben in Anwendungssystem XYZ“, „Ausgaben aus Anwendungssystem ABC“ oder auch „Ein-/Ausgaben in/aus Anwendungssystem EFG“). Diesen *Kommunikationsstandards* sollten dann *Nachrichtentypen* zugeordnet werden, die entweder recht grob beschreiben, dass eine bestimmte Menge von *Objekttypen* ein- bzw. ausgegeben werden oder aber auch detaillierter beschreiben, in welcher Form das erfolgt (z. B. „Bildschirmeingabe von Patientenstamm- und Patientenfalldaten“, „Audiowiedergabe eines Befunddiktats“, „Ein-/Ausgabe“).

E.C.A-6 Kommunikationsverbindung, Sende- und Empfangsschnittstelle

Dieses Konzept ist in Abschnitt E.B.D-3 bereits detailliert beschrieben. Für die Erweiterung 3LGM²-M sind jedoch folgende Ergänzungen erforderlich:

E.C.A-6.1 Interebenenbeziehungen „überträgt“ von Kommunikationsverbindung zu Nachrichtentyp, „kann senden“ und „kann empfangen“ von Sende- bzw. Empfangsschnittstelle zu Nachrichtentyp

Die *Assoziation* einer *Sendeschnittstelle* zu einem *Nachrichtentyp* beschreibt, dass diese *Sendeschnittstelle* Nachrichten dieses Typs versenden kann. Vereinfachend kann davon gesprochen werden, dass die *Sendeschnittstelle* den *Nachrichtentyp* versenden kann.

Die *Assoziation* einer *Empfangsschnittstelle* zu einem *Nachrichtentyp* beschreibt, dass diese *Empfangsschnittstelle* Nachrichten dieses Typs empfangen kann. Vereinfachend kann davon gesprochen werden, dass die *Empfangsschnittstelle* den *Nachrichtentyp* empfangen kann.

Die *Assoziation* einer *Kommunikationsverbindung* zu einem *Nachrichtentyp* beschreibt, dass diese *Kommunikationsverbindung* Nachrichten dieses Typs transportiert. Vereinfachend kann davon gesprochen werden, dass die *Kommunikationsverbindung* den *Nachrichtentyp* transportiert.

E.C.A-6.2 Fragen des Informationsmanagements und Unterstützung durch das 3LGM²

Fragen des Informationsmanagements	Verweis auf Text
Welche Möglichkeiten zum Senden und Empfangen von Nachrichten hat ein Anwendungsbaustein?	E.B.D-1
Welche Kommunikationsstandards und welche darin enthaltenen Nachrichtentypen unterstützt eine Kommunikationsschnittstelle?	E.C.A-2, E.C.A-3
Unterstützt eine Empfangsschnittstelle eines bestimmten Anwendungssystems, die einige HL7-Nachrichten empfangen kann, also offenbar „HL7-fähig“ ist, auch den Empfang des Nachrichtentyps ORU Observational Results (Unsolicited)?	E.B.D-1

E.C.A-6.3 Modellierungshinweise

- Bei einem *Anwendungsbaustein* kann für jeden *Nachrichtentyp*, zu dem er *Nachrichten* versenden/empfangen kann, eine besondere *Kommunikationsschnittstelle* modelliert werden. Es kann auch für alle sendbaren *Nachrichtentypen* eine einzige *Sende-* und für alle empfangbaren *Nachrichtentypen* eine einzige *Empfangsschnittstelle* modelliert werden. Es empfiehlt sich aber, auch *Kommunikationsschnittstellen* im Sinne von Assets (vgl. E.B.D-1.11) zu modellieren, d. h. dann eine *Kommunikationsschnittstelle* zu modellieren, wenn diese in der modellierten Form von dem jeweiligen Hersteller so hergestellt bzw. geliefert und von der einsetzenden Organisation bei Kauf bzw. Betrieb zu bezahlen ist.

E.C.A-7 Datenübertragungsverbindung

Dieses Konzept ist in Abschnitt E.B.E-4 bereits detailliert beschrieben. Für die Erweiterung 3LGM²-M ist jedoch folgende Ergänzung erforderlich:

Grundsätzlich wird darauf verzichtet, bei den *Datenübertragungsverbindungen* zu beschreiben, welche *Nachrichtentypen* dort transportiert werden können. Es wird vielmehr davon ausgegangen, dass dann, wenn eine *Datenübertragungsverbindung* besteht, dort grundsätzlich alle *Nachrichtentypen* transportiert werden können.

E.C.A-8 Konvertierung von Modellen zwischen 3LGM²-B und 3LGM²-M

Der 3LGM²-Baukasten unterstützt die Konvertierung von Modellen zwischen 3LGM²-B und 3LGM²-M.

E.C.A-8.1 Konvertierung von 3LGM²-B nach 3LGM²-M

Die Konvertierung ist verlustlos.

Für jeden *Objektyp*, der eine Beziehung zu einer *Kommunikationsverbindung*, *Sende-* oder *Empfangsschnittstelle* hat, wird ein *Nachrichtentyp* mit der Bezeichnung des *Objektyps* generiert. Der *Nachrichtentyp* erhält eine Beziehung zu dem *Objektyp* und die entsprechende Beziehung zu einer *Kommunikationsverbindung*, *Sende-* bzw. *Empfangsschnittstelle*. Alle generierten *Nachrichtentypen* werden einem ebenfalls generierten *Kommunikationsstandard* „generierte Nachrichtentypen“ zugeordnet.

E.C.A-8.2 Konvertierung von 3LGM²-M nach 3LGM²-B

Die Konvertierung ist nicht verlustlos und lässt sich nicht rückgängig machen.

Alle Kommunikationsstandards werden entfernt.

Für jeden *Nachrichtentyp*, der eine Beziehung zu einer *Kommunikationsverbindung*, *Sende-* bzw. *Empfangsschnittstelle* hat, werden die *Objektypen*, zu denen er in Beziehung steht, unmittelbar der jeweiligen *Kommunikationsverbindung*, *Sende-* bzw. *Empfangsschnittstelle* zugeordnet. Schließlich werden alle *Nachrichtentypen* entfernt.

E.C.B Serviceorientierte Kommunikation: 3LGM²-S

E.C.B-1 Überblick

Sowohl das Basis-Metamodell 3LGM²-B als auch die Erweiterung für nachrichtenorientierte Kommunikation 3LGM²-M gehen vom Paradigma des Austauschs von Daten als Mittel zur Integration aus. Mit diesem Mittel ist eine Datenintegration gut, eine funktionale Integration jedoch nur sehr umständlich zu realisieren. Zunehmend gewinnen aber Architekturen von Informationssystemen auch in der Praxis an Bedeutung, bei denen die einzelnen *Anwendungsbausteine* sich wechselseitig *Services* (Dienste) zur Verfügung stellen bzw. diese wechselseitig nutzen können, z. B. „Serviceorientierte Architekturen“ (SOA) [29-32]. In der Metamodell-Erweiterung 3LGM²-S für die serviceorientierte (dienstorientierte) Kommunikation werden daher *Klassen* angeboten, mit denen *Services*, ihre Bereitstellung und ihre Nutzung modelliert werden können. 3LGM²-S basiert auf [16].

Nur Modelle auf der Basis von 3LGM²-M, d. h. Modelle in denen bereits Nachrichtentypen modelliert wurden, können automatisch zu 3LGM²-S-Modellen erweitert werden. Umgekehrt ist eine Rück-Konvertierung nicht mehr möglich. Modelle auf der Basis von 3LGM²-S entstehen aus Modellen auf der Basis von 3LGM²-M im Wesentlichen durch die Beschreibung des Sendens von Nachrichten als Nutzung eines Empfangsdienstes beim Empfänger; die gesendeten Daten werden als Parameter dieses Empfangsdienstes aufgefasst (siehe E.C.B-12.1).

Modelle auf der Basis von 3LGM²-B können manuell zu 3LGM²-S-Modellen erweitert werden. Umgekehrt können Modelle auf der Basis von 3LGM²-S automatisch zu 3LGM²-M und 3LGM²-B-Modellen reduziert werden.

3LGM²-M ist in Abbildung 16 in UML-Notation dargestellt.

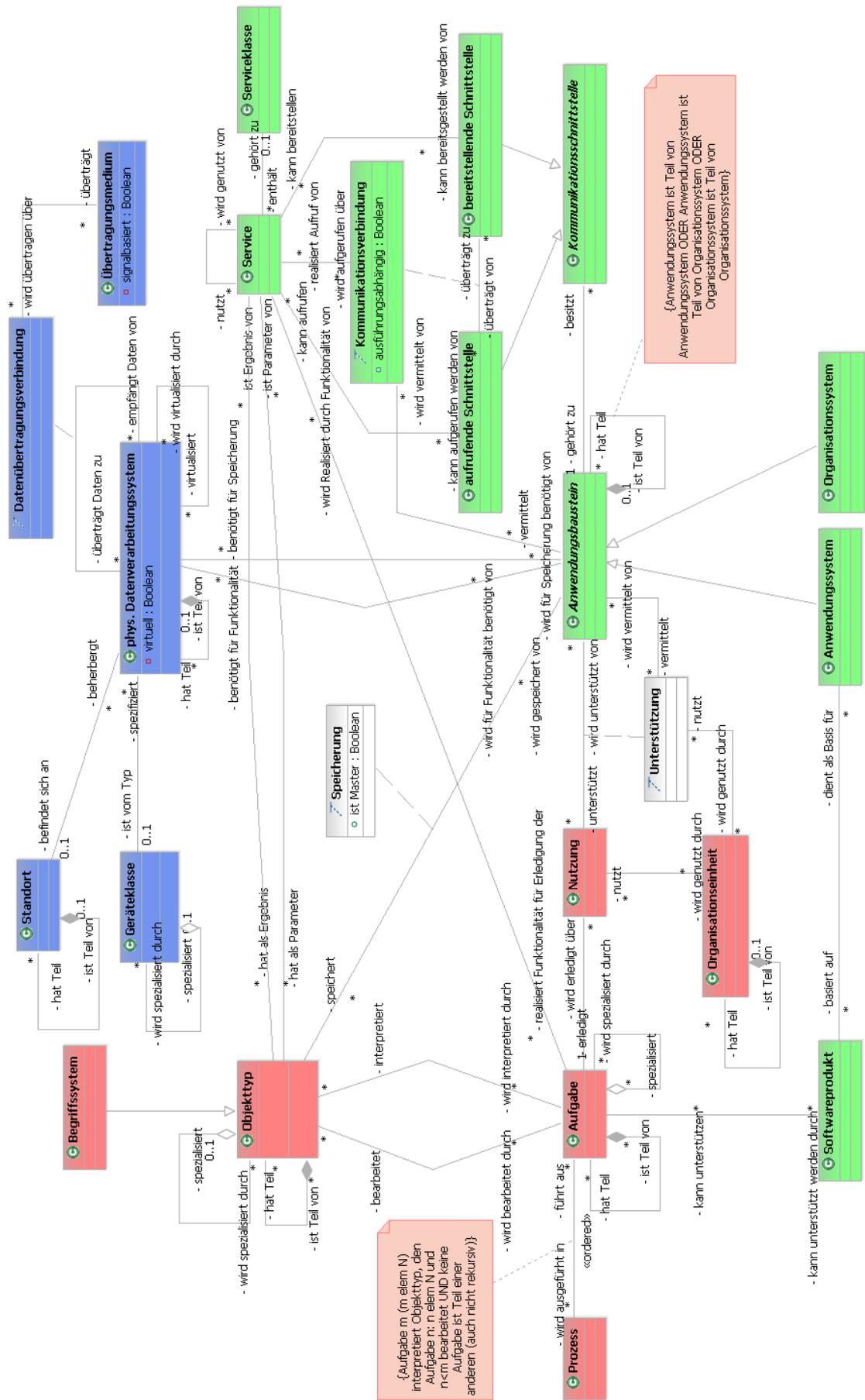


Abbildung 16: Das Metamodell 3LGM²-S

E.C.B-2 Klassen in 3LGM²-M, die durch andere Klassen ersetzt werden

Folgende Konzepte (*Klassen*) in 3LGM²-M entfallen in 3LGM²-S bzw. werden durch andere Konzepte ersetzt:

- *Sendschnittstelle* wird ersetzt durch *aufrufende Schnittstelle* (siehe E.C.B-6);
- *Empfangsschnittstelle* wird ersetzt durch *bereitstellende Schnittstelle*(siehe E.C.B-5);
- *Nachrichtentyp* entfällt;
- *Kommunikationsstandard* entfällt, kann aber durch *Serviceklasse* (siehe E.C.B-4) ersetzt werden.

In den folgenden Abschnitten werden die neuen Konzepte und Beziehungen und die Konzepte beschrieben, die eine angepasste Bedeutung erhalten.

E.C.B-3 Service

E.C.B-3.1 Beschreibung des Konzepts

Ein *Anwendungsbaustein* implementiert bestimmte Funktionalitäten, mit deren Hilfe er die Erledigung von *Aufgaben* unterstützt.

Definition: Service

Ein *Service* ist eine Funktionalität, die ein *Anwendungsbaustein* so bereitstellt, dass sie von anderen *Anwendungsbausteinen* genutzt werden kann.

Services werden auch als (Remote) Function oder (Remote) Procedure bezeichnet.

Die Ausführung eines *Services* durch den bereitstellenden *Anwendungsbaustein* hat eine oder mehrere der folgenden Auswirkungen:

- Daten, die der *Anwendungsbaustein* verwaltet, werden verändert;
- es werden Interaktionen mit einem Nutzer des *Anwendungsbausteins* ausgeführt;
- es werden *Services* aufgerufen, die dritte *Anwendungsbausteine* bereitstellen;
- es wird ein *Ergebnis* berechnet, das dem aufrufenden *Anwendungsbaustein* zurückübermittelt wird.

Einen *Service* kann ein menschlicher Nutzer des Informationssystems nicht unmittelbar nutzen, um so Unterstützung für die Erledigung einer *Aufgabe* zu erhalten. Dazu muss der Nutzer vielmehr einen *Anwendungsbaustein* nutzen, der den *Service* aufruft.

Damit ein *Anwendungsbaustein* A den *Service* eines anderen *Anwendungsbausteins* B nutzen kann, muss A bestimmte Daten als Parameter zur Verfügung stellen und erhält bestimmte Daten als Ergebnis nach Ausführen des *Services* zurück (siehe E.C.B-3.3).

Solche *Services* können z. B. sein:

- **PAT_AUFNAHME**: unter Nutzung dieses *Services* kann ein beliebiger *Anwendungsbaustein* die Erledigung der *Aufgabe* Patientenaufnahme unterstützen, ohne die dazu erforderlichen Funktionalitäten selbst implementieren zu müssen. Zur Nutzung des *Services* müssen Parameter bereitgestellt werden; nach Ausführen des *Services* wird ein Ergebnis zurückgeliefert.
- **NACHRICHTEN_EMPFANG**: Diesem *Service* kann als Parameter eine Nachricht übergeben werden, die dadurch von dem *Anwendungsbaustein*, der den *Service* bereitstellt, empfangen werden kann. Das Ergebnis könnte eine Acknowledgement-Nachricht sein, die den sendenden *Anwendungsbaustein* über den erfolgreichen Empfang der Nachricht informiert.

E.C.B-3.2 Beziehung „nutzt“ zu Service

Die Beziehung beschreibt, dass ein *Service* einen anderen *Service* in Anspruch nehmen kann. Beispielsweise könnte der in E.C.B-3.1 beschriebene *Service* **NACHRICHTEN_EMPFANG** von verschiedenen anderen *Services*, wie z. B. einem Druckservice oder einem *Service* zum Verteilen von Dokumenten genutzt werden.

E.C.B-3.3 Interebenenbeziehung „realisiert Funktionalität für Erledigung von“ zu Aufgabe

Für einen *Service* werden die *Aufgaben* benannt, für die der *Service* die entsprechende Funktionalität bereitstellt. Mit Hilfe der so bereitgestellten Funktionalität kann der den *Service* aufrufende *Anwendungsbaustein* seine Nutzer bei der Erledigung dieser *Aufgaben* unterstützen. Auf diese Weise wird beschrieben, wozu der *Service* durch einen anderen *Anwendungsbaustein* genutzt werden kann.

E.C.B-3.4 Beziehung „kann bereitgestellt werden von“ zu bereitstellende Schnittstelle

Der *Service* wird von einem *Anwendungsbaustein* mit Hilfe der *bereitstellenden Schnittstelle* so bereitgestellt, dass er von einem anderen *Anwendungsbaustein* genutzt werden kann.

E.C.B-3.5 Beziehung „kann aufgerufen werden von“ zu aufrufende Schnittstelle

Der *Service* kann von einem *Anwendungsbaustein* mit Hilfe der *aufrufenden Schnittstelle* genutzt werden.

E.C.B-3.6 Beziehung „wird aufgerufen über“ zu Kommunikationsverbindung

Der *Service* wird tatsächlich über die *Kommunikationsverbindung* von den daran beteiligten *Anwendungssystemen* genutzt.

E.C.B-3.7 Beziehung „gehört zu“ zu Serviceklasse

Mehrere *Services* können durch eine *Serviceklasse* zusammengefasst werden. So können z. B. alle *Services*, die jeweils gewisse HL7-Nachrichten empfangen können, zu einer *Serviceklasse* „HL7“ zusammengefasst werden.

E.C.B-3.8 Interebenenbeziehung „hat als Parameter“ zu Objekttyp

Damit ein *Anwendungsbaustein* einen bestimmten *Service* nutzen kann, müssen ggf. bestimmte *Daten* von der *aufrufenden Schnittstelle* des *Anwendungsbausteins* an die *bereitstellende Schnittstelle* als *Parameter* übermittelt werden. Die Art der als *Parameter* bereitzustellenden *Daten* wird durch die Angabe von *Objekttypen* beschrieben. Möglicherweise sind aber für bestimmte *Services* auch keine *Parameter* erforderlich.

E.C.B-3.9 Interebenenbeziehung „hat als Ergebnis“ zu Objekttyp

Bei der Ausführung eines *Services* durch das bereitstellende *Anwendungssystem* kann ein *Ergebnis* berechnet werden. Die Art der als *Ergebnis* zurückübermittelten *Daten* wird durch die Angabe von *Objekttypen* beschrieben. Wenn als *Ergebnis* keine *Daten* zurückübermittelt werden, die für das Modellierungsziel erscheinen, z. B. weil lediglich eine Bestätigung des Empfangs der *Parameter* bzw. des Aufrufs des *Services* zurückübermittelt werden, kann auf die Modellierung des *Ergebnisses* verzichtet werden.

E.C.B-3.10 Fragen des Informationsmanagements und Unterstützung durch das 3LGM²

Fragen des Informationsmanagements	Verweis auf Text
Welcher Anwendungsbaustein stellt anderen Anwendungsbausteinen die Funktionalität zur Verfügung, die zur Erledigung einer bestimmten Aufgabe erforderlich ist?	E.B.D-1, E.C.B-10
Welcher Service wird von Anwendungsbausteinen redundant bereitgestellt?	E.B.D-1, E.C.B-10
Welche Anwendungsbausteine stellen Services bereit, die alle auch durch (mehrere) andere Anwendungsbausteine erledigt werden können?	E.C.B-10

E.C.B-3.11 Modellierungshinweise

Durch die Modellierung von *Services* und ihren Beziehungen zu *Objekttypen* und *Aufgaben* kann dargestellt werden, auf welche Weise ein *Anwendungsbaustein* für andere *Anwendungsbausteine* nützlich ist.

E.C.B-4 Serviceklasse

E.C.B-4.1 Beschreibung des Konzepts

Eine *Serviceklasse* fasst *Services*, die vom Modellierer als gleichartig angesehen werden, zusammen.

Serviceklassen können z. B. sein: „HL7-Empfangsservice“, „Services für die Patientendatenverwaltung“, „Services für die Verwaltung der Patientenakte“, „Druckservices“ etc.

Serviceklassen werden in den graphischen Darstellungen nicht als eigene Objekte dargestellt.

E.C.B-4.2 Beziehung „enthält“ zu Service

Jeder *Serviceklasse* können beliebig viele *Services* zugeordnet werden.

E.C.B-4.3 Fragen des Informationsmanagements und Unterstützung durch das 3LGM²

Fragen des Informationsmanagements	Verweis auf Text
Welche Anwendungsbausteine stellen Services einer bestimmten Serviceklasse bereit?	E.C.B-10

E.C.B-5 Bereitstellende Schnittstelle

E.C.B-5.1 Beschreibung des Konzepts

Über eine *bereitstellende Schnittstelle* stellt ein *Anwendungsbaustein* *Services* für andere *Anwendungsbausteine* zur Verfügung.

E.C.B-5.2 Beziehung „kann bereitstellen“ zu Service

Die *bereitstellende Schnittstelle* stellt genau die hier angegebenen *Services* zur Verfügung.

E.C.B-5.3 Beziehung zu Kommunikationsverbindung

Wenn die von der *bereitstellenden Schnittstelle* zur Verfügung gestellten *Services* tatsächlich von einer *aufzufendenden Schnittstelle* genutzt werden, dann wird zwischen beiden Schnittstellen eine *Kommunikationsverbindung* modelliert.

E.C.B-6 Aufrufende Schnittstelle

E.C.B-6.1 Beschreibung des Konzepts

Über eine *aufzufende Schnittstelle* kann ein *Anwendungsbaustein* *Services* anderer *Anwendungsbausteine* nutzen.

E.C.B-6.2 Beziehung „kann aufrufen“ zu Service

Die *aufzufende Schnittstelle* kann genau die hier angegebenen *Services* aufrufen, damit der *Anwendungsbaustein* sie nutzen kann.

E.C.B-6.3 Beziehung zu Kommunikationsverbindung

Wenn von der *aufzufenden Schnittstelle* aufrufbare *Services* tatsächlich bei einer bestimmten *bereitgestellten Schnittstelle* aufgerufen werden, dann wird zwischen beiden Schnittstellen eine *Kommunikationsverbindung* modelliert.

E.C.B-7 Kommunikationsverbindung

E.C.B-7.1 Beschreibung des Konzepts

Dieses Konzept ist in Abschnitt E.B.D-3 bereits detailliert beschrieben. Für die Erweiterung 3LGM²-S sind jedoch folgende Ergänzungen erforderlich:

Eine *Kommunikationsverbindung* zwischen einer *aufzufendenden* und einer *bereitstellenden Schnittstelle* beschreibt, dass die *aufzufende Schnittstelle* tatsächlich bestimmte *Services* der *bereitstellenden Schnittstelle* aufruft.

Gehen von einer *aufzufendenden Schnittstelle* mehrere *Kommunikationsverbindungen* zu *bereitstellenden Schnittstellen* aus, die jeweils dieselben *Services* nutzen, werden diese *Kommunikationsverbindungen* als Alternativen angesehen. Dies bedeutet, dass der aufrufende *Anwendungsbaustein* zur Laufzeit entscheidet, bei welchem bereitstellenden *Anwendungsbaustein* er den *Service* tatsächlich aufruft.

E.C.B-7.2 Attribut „ausführungsabhängig“

Über das Attribut „*ausführungsabhängig*“ (ja/nein) kann modelliert werden, ob der den *Service* aufrufende *Anwendungsbaustein* mit seiner weiteren Datenverarbeitung erst fortfahren kann, wenn das Ergebnis der Serviceausführung durch den bereitstellenden *Anwendungsbaustein* zurückübermittelt wurde. Dies wäre z. B. bei synchron abgewickelten Datenbankabfragen der Fall. Bei asynchronem Nachrichtenversand wird dagegen in der Regel nicht von einer *Ausführungsabhängigkeit* auszugehen sein.

Eine *Ausführungsabhängigkeit* kann auch dann modelliert werden, wenn ein *Anwendungsbaustein* AB die Erledigung einer *Aufgabe* A durch einen Nutzer unterstützt (vgl. E.B.B-2.4 und E.B.D-1.6), aber über eine entsprechende *Kommunikationsverbindung* einen *Service* D aufruft, der die dafür erforderliche Funktionalität bereitstellt (vgl. E.C.B-3.3). Genau dann wenn AB die Erledigung der *Aufgabe* A durch einen Nutzer nur dann unterstützen kann, wenn der *Service* D nach dem Aufruf erfolgreich ausgeführt wurde, ist „*ausführungsabhängig*“ mit „ja“ zu modellieren.

Alternative *Kommunikationsverbindungen* (vgl. E.C.B-7.1) müssen alle als *ausführungsabhängig* bzw. alle als nicht *ausführungsabhängig* modelliert sein.

E.C.B-7.3 Beziehung „realisiert Aufruf von“ zu Service

Die an der *Kommunikationsverbindung* beteiligte *aufzufende Schnittstelle* ruft tatsächlich genau die hier angegebenen *Services* auf.

E.C.B-7.4 Modellierungshinweise

- Für die Bewertung der Ausführungsabhängigkeit (siehe E.C.B-7.1) gibt es keine objektiven Kriterien. Vielmehr obliegt es dem Modellierer, eine Einschätzung vorzunehmen. Dabei ist zu berücksichtigen, dass der 3LGM²-Baukasten Analysen bereitstellt, mit denen auf der Basis dieser Angaben ermittelt wird, welche Anwendungsbausteine in hohem Maße voneinander abhängig sind. Die Modellierung ist also klar an den Auswertungszielen auszurichten.

E.C.B-8 Kommunikationsschnittstelle

Dieses Konzept ist in Abschnitt E.B.D-3 bereits detailliert beschrieben. Für die Erweiterung 3LGM²-S sind jedoch folgende Ergänzungen erforderlich:

Ein *Anwendungsbaustein* besitzt *Kommunikationsschnittstellen*, um mit anderen *Anwendungsbausteinen* kommunizieren zu können.

Eine *aufzufende Schnittstelle* eines *Anwendungssystems* kann *Services* eines anderen *Anwendungssystems* aufrufen, das diese *Services* über eine *bereitstellende Schnittstelle* bereitstellen kann. Wenn von einer *aufzufendenden Schnittstelle* tatsächlich *Services* einer *bereitstellenden Schnittstelle* aufgerufen werden, dann werden diese *Kommunikationsschnittstellen* durch eine *Kommunikationsverbindung* verbunden, die von der *bereitstellenden* zu der *aufzufendenden Schnittstelle* weist (vgl.E.C.B-7).

Eine *Kommunikationsschnittstelle* kann nur *aufzufende* oder *bereitstellende Schnittstelle* sein.

Alle *Kommunikationsschnittstellen* eines *Anwendungsbausteins* gelten implizit als so untereinander verbunden, dass jede *aufzufende Schnittstelle* des *Anwendungsbausteins* auf alle Parameter aller *bereitstellenden Schnittstellen* des *Anwendungsbausteins* zugreifen kann.

E.C.B-9 Aufgabe

Dieses Konzept ist in Abschnitt E.B.B-2 bereits detailliert beschrieben. Für die Erweiterung 3LGM²-S sind jedoch folgende Ergänzungen erforderlich:

E.C.B-9.1 Interebenenbeziehung „kann aufrufen“ zu Service

Im Unterschied zur Beziehung zur *Klasse Nutzung* (vgl. E.B.B-2.4) wird durch die Beziehung einer *Aufgabe* zu einem *Service* ausgedrückt, dass der *Service* interessierten *Anwendungsbausteinen* eine Funktionalität anbietet, die es den *Anwendungsbausteinen* ermöglicht, menschliche Nutzer bei der Erledigung der *Aufgabe* zu unterstützen. D. h. der *Service* kann nicht unmittelbar die Erledigung der *Aufgabe* unterstützen, sondern muss zu diesem Zweck von einem Anwendungssystem genutzt werden.

E.C.B-9.2 Beziehungen zu Nutzung

Siehe E.B.B-2.4 (*Anwendungsbaustein*).

E.C.B-9.3 Fragen des Informationsmanagements und Unterstützung durch das 3LGM²

Fragen des Informationsmanagements	Verweis auf Text
Welche Anwendungsbausteine sind erforderlich, um menschliche Nutzer bei der Erledigung einer bestimmten Aufgabe zu unterstützen?	
Zur Erledigung welcher Aufgaben werden besonders viele verschiedene Services genutzt?	

E.C.B-9.4 Modellierungshinweise

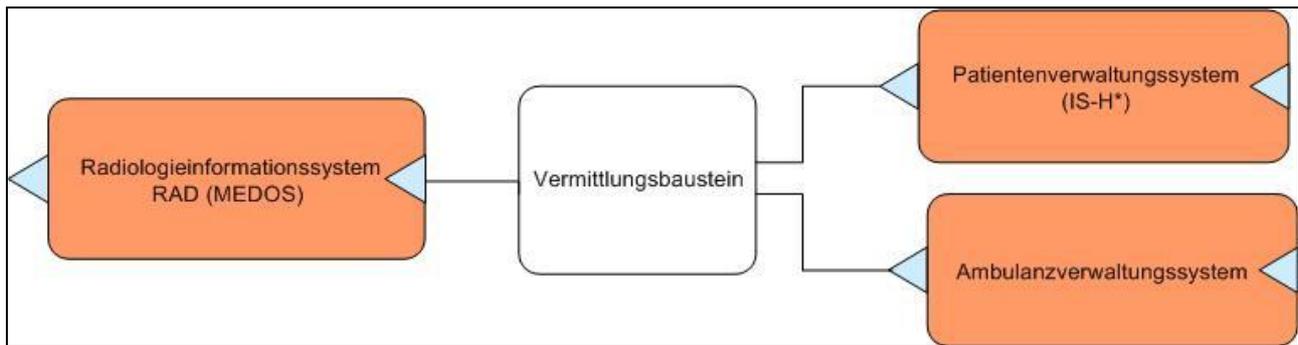
- Wenn für einen *Anwendungsbaustein* durch die Modellierung einer Beziehung zu *Nutzung* beschrieben ist, dass er in einer bestimmten *Organisationseinheit* eine bestimmte *Aufgabe* unterstützt, dann muss sichergestellt sein, dass er entweder die erforderliche Funktionalität selbst implementiert (vgl. E.C.B-3.3) oder aber über eine *aufrufende Schnittstelle* und entsprechende *Kommunikationsverbindung* mit einem anderen *Anwendungsbaustein* verbunden ist, der ihm einen *Service* bereitstellt, welcher die erforderliche Funktionalität bietet (vgl. E.C.B-9.1).
- Die explizite Modellierung von *Kommunikationsverbindungen* zwischen *Kommunikationsschnittstellen* desselben *Anwendungsbausteins* ist nicht vorgesehen (siehe E.B.D-3).

E.C.B-10 Anwendungsbaustein

Dieses Konzept ist in Abschnitt E.B.D-1 bereits detailliert beschrieben. Für die Erweiterung 3LGM²-S sind jedoch folgende Ergänzungen erforderlich:

E.C.B-10.1 Beziehung „vermittelt“ zu Kommunikationsverbindung

In Architekturen, die sich z. B. an CORBA oder DCOM orientieren, sind *Anwendungsbausteine* vorgesehen, die zur Laufzeit einem *Anwendungsbaustein* AB, der einen bestimmten *Service* D aufrufen möchte, auf Anfrage einen *Anwendungsbaustein* AB' benennen, der zu diesem Zeitpunkt dazu in der Lage ist, den *Service* D bereitzustellen und auszuführen (vgl. [16], S. 106 ff.). Durch die Beziehung „vermittelt“ wird ausgedrückt, dass ein *Anwendungsbaustein* VB in diesem Sinne die entsprechende *Kommunikationsverbindung* vermittelt. Ein solcher *Anwendungsbaustein* wird auch als *Vermittlungsbaustein* bezeichnet. Es ist nicht vorgesehen explizit zu modellieren, dass zu diesem Zweck die *Anwendungsbausteine* AB und AB' bestimmte Vermittlungsservices des *Vermittlungsbausteins* VB aufrufen (vgl. [16], S. 106 ff) oder dass der *Vermittlungsbaustein* VB solche Vermittlungsservices bereitstellt; vielmehr wird implizit davon ausgegangen, dass VB solche Vermittlungsservices bereitstellt und dass sowohl AB und AB' diese *Services* in Anspruch nehmen. Zur Darstellung siehe Abbildung 17.

Abbildung 17: Darstellung eines *Vermittlungsbausteins*

E.C.B-10.2 Fragen des Informationsmanagements und Unterstützung durch das 3LGM²

Fragen des Informationsmanagements

Verweis auf Text

Auf welche anderen Anwendungsbausteine ist ein Anwendungsbaustein angewiesen, damit er wie angegeben menschliche Nutzer bei der Erledigung bestimmter Aufgaben unterstützen kann?

E.C.B-10.3 Modellierungshinweise

Siehe E.C.B-9.3.

E.C.B-11 Objekttyp

Dieses Konzept ist in Abschnitt E.B.B-1 bereits detailliert beschrieben. Für die Erweiterung 3LGM²-S sind jedoch folgende Ergänzungen erforderlich:

E.C.B-11.1 Interebenenbeziehung „hat als Parameter“ zu Service

Wenn ein *Objekttyp Parameter* eines Services ist, bedeutet das, dass der *Service* Daten über *Objekte* der *Objektklasse* erwartet, auf die der *Objekttyp* verweist. Nur dann, wenn diese Daten bereitgestellt werden, kann der *Service* ausgeführt werden.

E.C.B-11.2 Interebenenbeziehung „hat als Ergebnis“ zu Service

Wenn ein *Objekttyp Ergebnis* eines Services ist, bedeutet das, dass der *Service* Daten über *Objekte* der *Objektklasse* erzeugt, auf die der *Objekttyp* verweist. Diese Daten werden bereitgestellt, sobald der *Service* ausgeführt wurde.

E.C.B-11.3 Fragen des Informationsmanagements und Unterstützung durch das 3LGM²

Fragen des Informationsmanagements

Verweis auf Text

Welche Anwendungsbausteine sind auf Daten über diesen Objekttyp angewiesen?

Von welchen Anwendungssystemen werden Daten über diesen Objekttyp erzeugt?

E.C.B-11.4 Modellierungshinweise

Siehe Abschnitt E.B.B-1.8.

E.C.B-12 Konvertierung von Modellen zwischen 3LGM²-B und 3LGM²-S

E.C.B-12.1 Konvertierung von 3LGM²-B nach 3LGM²-S

Die Konvertierung ist verlustlos, benötigt aber ein Eingreifen des Modellierers.

- Wenn eine *Kommunikationsschnittstelle* sowohl Sende- als auch Empfangsschnittstelle ist, muss sie in zwei getrennte Schnittstellen aufgespalten werden (vgl. E.C.B-7.4).
- Eine *Sendeschnittstelle*, die bestimmte *Objektypen* senden kann, wird wie folgt konvertiert:
 - aus der *Sendeschnittstelle* wird eine *aufrufende Schnittstelle*;
 - für jeden *Objektyp* werden n *Services* „receive_<Name des Objektyps>_Nachrichtentyp_x“ ($x=1\dots n, x \in \mathbb{N}$) eingerichtet, die die *Objektypen* als *Parameter* erwarten. Es werden also entsprechend der Anzahl n der *Kommunikationsverbindungen*, über die der *Objektyp* gesendet wird, Dummy-Nachrichtentypen zu jedem *Objektyp* erzeugt. Es wird empfohlen, dass der Modellierer den Nachrichtentypen nachträglich aussagekräftigere Namen gibt bzw. manuell gleiche Nachrichtentypen vereinigt. Die *Services* werden automatisch der *aufrufenden Schnittstelle* zugeordnet;
 - *Serviceklassen* können nicht automatisch erzeugt werden, sondern müssen manuell modelliert werden.
- Eine *Empfangsschnittstelle*, die bestimmte *Nachrichtentypen* empfangen kann, wird wie folgt konvertiert:
 - aus der *Sendeschnittstelle* wird eine *bereitstellende Schnittstelle*;
 - für jeden *Nachrichtentyp* (der beim Erzeugen der *Sendeschnittstellen* automatisch generiert wurde) wird ein *Service* „receive_<Name des Objektyps>_Nachrichtentyp_x“ eingerichtet, der die *Objektypen* als *Parameter* erwartet, zu denen der *Nachrichtentyp* in Beziehung steht. Diese *Services* werden der *bereitstellenden Schnittstelle* zugeordnet;
 - *Serviceklassen* können nicht automatisch erzeugt werden, sondern müssen manuell modelliert werden.
- Eine *Kommunikationsverbindung*, die bestimmte *Objektypen* versendet, wird wie folgt konvertiert:
 - die neue *Kommunikationsverbindung* verbindet die *aufrufende Schnittstelle* und *bereitstellende Schnittstelle*, deren vorausgehende *Sende-* und *Empfangsschnittstelle* zuvor bereits durch die *Kommunikationsverbindung* verbunden waren;
 - die neue *Kommunikationsverbindung* wird zu all den *Services* in Beziehung gesetzt, die zu den *Nachrichtentypen* passen, die über die *Kommunikationsverbindung* gesendet werden.
- Der Benutzer wird aufgefordert, die unmittelbaren Beziehungen von *Aufgabe* zu *Service* (siehe E.C.B-3.3) und *Anwendungsbaustein* (siehe E.C.B-10.1) von Hand nachzumodellieren. Die entsprechenden Informationen können nicht aus dem alten Modell entnommen werden.

E.C.B-12.2 Konvertierung von 3LGM²S nach 3LGM²B

Die Konvertierung ist nicht verlustlos, benötigt aber kein Eingreifen des Modellierers.

- Eine *aufrufende Schnittstelle*, die bestimmte *Services* aufrufen kann, wird wie folgt konvertiert:
 - eine *aufrufende Schnittstelle* wird in zwei *Schnittstellen* aufgeteilt: eine *Empfangsschnittstelle* $e1$ und eine *Sendeschnittstelle* $s1$.
 - der *Objektyp* $o1$, den der aufgerufene *Service* als *Parameter* hat, wird der *Sendeschnittstelle* $s1$ zugeordnet.
 - der *Objektyp* $o2$, den der aufgerufene *Service* als *Ergebnis* hat, wird der *Empfangsschnittstelle* $e1$ zugeordnet.
- Eine *bereitstellende Schnittstelle*, die bestimmte *Services* bereitstellt, wird wie folgt konvertiert:
 - eine *bereitstellende Schnittstelle* wird in 2 *Schnittstellen* aufgeteilt: eine *Empfangsschnittstelle* $e2$ und eine *Sendeschnittstelle* $s2$.
 - der *Objektyp* $o2$, den der bereitgestellte *Service* als *Ergebnis* hat, wird der *Sendeschnittstelle* $s2$ zugeordnet.
 - der *Objektyp* $o1$, den der bereitgestellte *Service* als *Parameter* hat, wird der *Empfangsschnittstelle* $e2$ zugeordnet.
- Eine *Kommunikationsverbindung*, die den Aufruf von *Services* realisiert, wird wie folgt konvertiert:

- Es wird eine *Kommunikationsverbindung* von s1 nach e2 angelegt, dieser wird der *Objektyp* o1 zugeordnet.
- Es wird eine *Kommunikationsverbindung* von s2 nach e1 angelegt, dieser wird der *Objektyp* o2 zugeordnet.
- Der *Service* und übergeordnete *Serviceklassen* werden gelöscht.

E.C.B-13 Konvertierung von Modellen zwischen 3LGM²-M und 3LGM²-S

Der 3LGM²-Baukasten unterstützt die Konvertierung von Modellen zwischen 3LGM²-M und 3LGM²-S.

E.C.B-13.1 Konvertierung von 3LGM²-M nach 3LGM²-S

Die Konvertierung ist verlustlos, benötigt aber ein Eingreifen des Modellierers.

- Wenn eine *Kommunikationsschnittstelle* sowohl Sende- als auch Empfangsschnittstelle ist, muss sie in zwei getrennte Schnittstellen aufgespalten werden (vgl.E.C.B-7.4).
- Eine *Sendeschnittstelle*, die bestimmte *Nachrichtentypen* versenden kann, wird wie folgt konvertiert:
 - aus der *Sendeschnittstelle* wird eine *aufrufende Schnittstelle*;
 - für jeden *Nachrichtentyp* wird ein *Service* „receive_<Name des Nachrichtentyps>“ eingerichtet, der die *Objektypen* als *Parameter* erwartet, zu denen der *Nachrichtentyp* in Beziehung steht. Diese *Services* werden der *aufrufenden Schnittstelle* zugeordnet.
 - für jeden *Kommunikationsstandard*, zu dem die *Nachrichtentypen* in Beziehung stehen, wird eine *Serviceklasse* „<Name des Kommunikationsstandards>_receive“ eingerichtet, der der jeweils entsprechende zuvor definierte *Service* zugeordnet wird.
- Eine *Empfangsschnittstelle*, die bestimmte *Nachrichtentypen* empfangen kann, wird wie folgt konvertiert:
 - aus der *Sendeschnittstelle* wird eine *bereitstellende Schnittstelle*;
 - für jeden *Nachrichtentyp* wird ein *Service* „receive_<Name des Nachrichtentyps>“ eingerichtet, der die *Objektypen* als *Parameter* erwartet, zu denen der *Nachrichtentyp* in Beziehung steht. Diese *Services* werden der *bereitstellenden Schnittstelle* zugeordnet.
 - für jeden *Kommunikationsstandard*, zu dem die *Nachrichtentypen* in Beziehung stehen, wird eine *Serviceklasse* „<Name des Kommunikationsstandards>_receive“ eingerichtet, der der jeweils entsprechende zuvor definierte *Service* zugeordnet wird.
- Eine *Kommunikationsverbindung*, die bestimmte *Nachrichtentypen* versendet, wird wie folgt konvertiert:
 - Die neue *Kommunikationsverbindung* verbindet die *aufrufende Schnittstelle* und *bereitstellende Schnittstelle*, deren vorausgehende *Sende-* und *Empfangsschnittstelle* zuvor bereits durch die *Kommunikationsverbindung* verbunden waren.
 - Die neue *Kommunikationsverbindung* wird zu all den *Services* in Beziehung gesetzt, die zuvor aus *Nachrichtentypen* abgeleitet wurden und zu den *Nachrichtentypen* passen, die über die *Kommunikationsverbindung* gesendet werden.
- Der Benutzer wird aufgefordert, die unmittelbaren Beziehungen von *Aufgabe* zu *Service* (siehe E.C.B-3.3) und *Anwendungsbaustein* (siehe E.C.B-10.1) von Hand nachzumodellieren. Die entsprechenden Informationen können nicht aus dem alten Modell entnommen werden.

E.C.B-13.2 Konvertierung von 3LGM²-S nach 3LGM²-M

Die Konvertierung ist nicht verlustlos, benötigt aber auch kein Eingreifen des Modellierers.

- Eine *aufrufende Schnittstelle*, die bestimmte *Services* aufrufen kann, wird wie folgt konvertiert:
 - eine *aufrufende Schnittstelle* wird in zwei *Schnittstellen* aufgeteilt: eine *Empfangsschnittstelle* e1 und eine *Sendeschnittstelle* s1.
 - für den *Objektyp* o1, den der aufgerufene *Service* als *Parameter* hat, wird ein *Nachrichtentyp* „o1_Nachrichtentyp_x (x=1...n, x ∈ ℕ)“ angelegt, dieser wird der *Sendeschnittstelle* s1 zugeordnet.

Die *Serviceklasse*, die dem aufgerufenen *Service* zugeordnet ist, wird in einen Kommunikationsstandard umgewandelt und der generierte *Nachrichtentyp* wird diesem zugeordnet.

- für den *Objektyp* o_2 , den der aufgerufene *Service* als Ergebnis hat, wird ein *Nachrichtentyp* „ $o_2_Nachrichtentyp_x$ ($x=1\dots n, x \in \mathbb{N}$)“ angelegt, dieser wird der *Empfangsschnittstelle* e_1 zugeordnet. Die *Serviceklasse*, die dem aufgerufenen *Service* zugeordnet ist, wird dem generierten *Nachrichtentyp* als *Kommunikationsstandard* zugeordnet.
- Eine bereitstellende Schnittstelle, die bestimmte *Services* bereitstellt, wird wie folgt konvertiert:
 - eine *bereitstellende Schnittstelle* wird in zwei *Schnittstellen* aufgeteilt: eine *Empfangsschnittstelle* e_2 und eine *Sendeschnittstelle* s_2 .
 - Der *Nachrichtentyp* „ $o_2_Nachrichtentyp_x$ “ wird der *Sendeschnittstelle* s_2 zugeordnet.
 - Der *Nachrichtentyp* „ $o_1_Nachrichtentyp_x$ “ wird der *Empfangsschnittstelle* e_2 zugeordnet.
- Werden die *Objektypen* o_1 und o_2 auch an anderer Stelle zwischen *Anwendungsbausteinen* kommuniziert, so werden neue zugehörige *Nachrichtentypen* angelegt, der Zähler x im Namen des *Nachrichtentypen* wird dabei inkrementiert. Eine nachträgliche Umbenennung der automatisch generierten *Nachrichtentypen* oder die Vereinigung von *Nachrichtentypen* seitens des Modellierers wird empfohlen.
- Eine *Kommunikationsverbindung*, die den Aufruf von *Services* realisiert, wird wie folgt konvertiert:
 - Es wird eine *Kommunikationsverbindung* von s_1 nach e_2 angelegt, dieser wird der *Nachrichtentyp* „ $o_1_Nachrichtentyp_x$ “ zugeordnet.
 - Es wird eine *Kommunikationsverbindung* von s_2 nach e_1 angelegt, dieser wird der *Nachrichtentyp* „ $o_2_Nachrichtentyp_x$ “ zugeordnet.
- Der *Service* wird gelöscht.

F Anforderungen an den 3LGM²-Baukasten

F.A Darstellung von 3LGM²-Modellen im 3LGM²-Baukasten

Ein Modell besteht im 3LGM²-Baukasten aus einem sogenannten Gesamtmodell und beliebig vielen Teilmodellen. Das Gesamtmodell enthält jeweils alle Elemente eines konkreten Modells mit all ihren Eigenschaften, die durch das Metamodell definiert sind. Teilmodelle sind eine eingeschränkt beliebige Auswahl aus den Elementen des Gesamtmodells. Eingeschränkt beliebig bedeutet, dass es Elementarten gibt, von denen sämtliche Instanzen automatisch in jedem Teilmodell vorhanden sind, und Elemente, bei denen explizit durch einen Modellierer festgelegt werden muss, ob sie in dem jeweiligen Teilmodell enthalten sind oder nicht. Zur Visualisierung des Gesamtmodells sowie der Teilmodelle stehen folgende Sichten zur Verfügung:

F.A.A Baumansicht

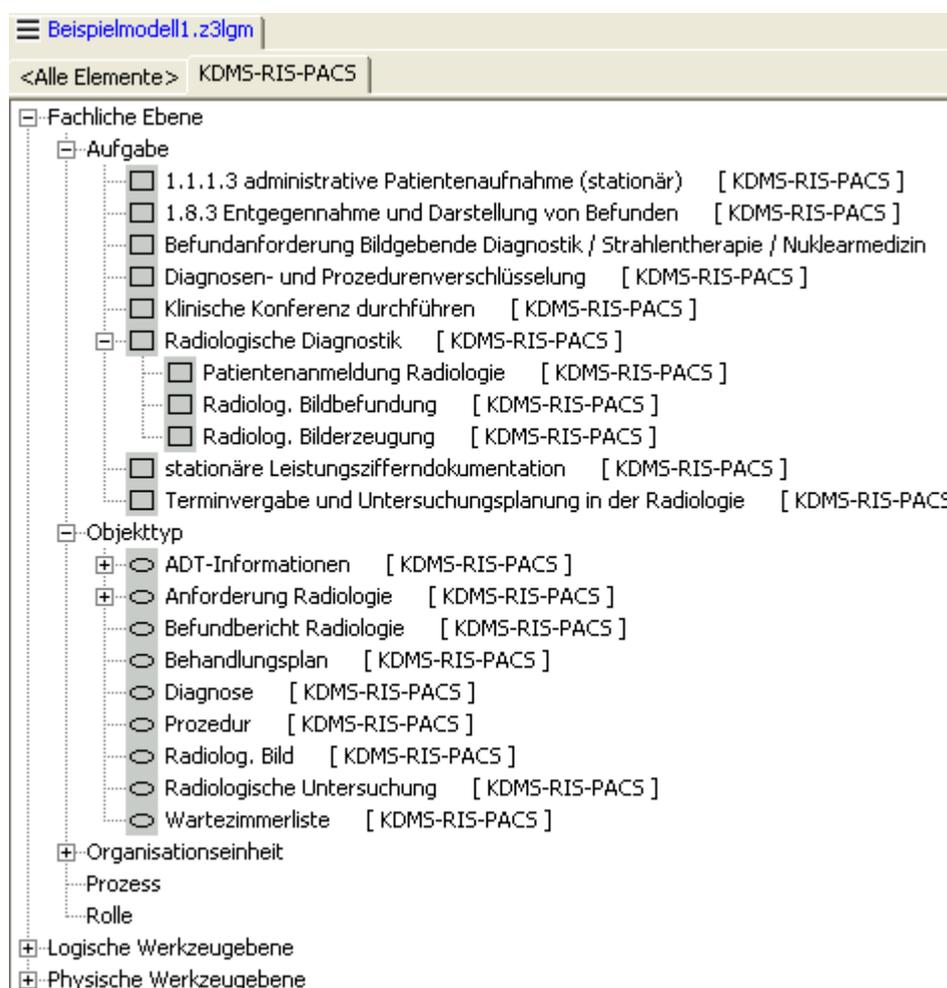


Abbildung 18: Beispiel einer Baumansicht eines Teilmodells

Die Baumansicht eines Gesamtmodells oder Teilmodells stellt in der obersten Hierarchiestufe die drei Gegenstandsebenen dar und darunter jeweils alle 3LGM²-Konzepte, die der jeweiligen Gegenstandsebene zugeordnet sind. Unterhalb der Knoten, die die 3LGM²-Elementklassen repräsentieren, werden evtl. hierarchisch gegliedert über Ist-Teil-Von-Beziehungen alle konkreten Elemente des Teil- bzw. Gesamtmodells angezeigt.

Das Beispiel in Abbildung 18 stellt eine solche Baumansicht für ein Teilmodell mit dem Namen ‚KMDS-RIS-PACS‘ dar. Der fachlichen Gegenstandsebene sind die 3LGM²-Konzepte *Aufgabe*, *Objektyp*, *Organisationseinheit*, *Prozess*

und *Rolle* des vorhergehenden Metamodells zugeordnet. Unterhalb des Knotens *Aufgabe* kann man die in diesem Teilmodell enthaltenen konkreten *Aufgaben* erkennen. Die *Aufgabe* ‚Radiologische Diagnostik‘ besitzt in diesem Beispiel drei *Teilaufgaben*, die ihr untergeordnet sind.

F.A.B Grafische Einzelebenenansicht

Zu jedem Teilmodell (aber nicht zum Gesamtmodell) gibt es jeweils drei sogenannte Einzelebenenansichten. Jede dieser Einzelebenenansichten stellt ausschließlich eine Auswahl der Konzepte jeweils einer der drei Gegenstandsebenen dar. Sie werden somit als grafische Sicht der *Fachlichen Ebene*, der *Logischen Werkzeugebene* und der *Physischen Werkzeugebene* bezeichnet. Die Sichten selbst sind zweidimensionale Graphen auf einer in ihren Ausmaßen beschränkten Zeichenebene. Aufgrund dieser Eigenschaften werden die Begriffe *Fachliche Ebene*, *Logische Werkzeugebene* und *Physische Werkzeugebene* oft synonym für die jeweilige Gegenstandsebene des Metamodells oder ganz speziell für die grafische Darstellung eines Teilmodells in der jeweiligen Einzelebenenansicht verwendet.

In Abbildung 6 ist ein Beispiel für die grafische Sicht einer *Fachlichen Ebene* zu sehen und in Abbildung 14 ist die grafische Repräsentation der *Physischen Werkzeugebene* eines Teilmodells dargestellt. Abbildung 6 zeigt eine *Fachliche Ebene*, wie sie sowohl im bestehenden 3LGM²-Baukasten, als auch in Zukunft mit den in diesem Dokument definierten Konzepten aussieht bzw. aussehen wird. Keine der drei grafischen Einzelebenenansichten stellt alle Konzepte der zugehörigen Gegenstandsebene dar, sondern sie beschränken sich aus Gründen der Übersichtlichkeit auf die in den meisten Anwendungsfällen wichtigsten Konzepte und deren Zusammenhänge der jeweiligen Gegenstandsebene. So besitzen z. B. Elemente der *Klasse Organisationseinheit* in der grafischen Ansicht der *Fachlichen Ebene* anders als in der Baumansicht keine Repräsentation.

Alle diese Einzelsichten beschreiben Informationssysteme vorwiegend statisch in ihrer Struktur. Zusätzlich dazu kann man in der grafischen Sicht der *Fachlichen Ebene* und der *Logischen Werkzeugebene* noch eine dynamische Perspektive einblenden, die die Abläufe auf den Strukturen verdeutlichen (vgl. oberer Teil von Abbildung 8).

F.A.C Grafische 3-Ebenen-Ansicht

Diese Sicht stellt die drei grafischen Einzelsichten eines Teilmodells übereinander in der Reihenfolge *Fachliche Ebene*, *Logische Werkzeugebene* und *Physische Werkzeugebene* (von oben nach unten) dar. Abbildung 19 zeigt die grafische 3-Ebenen-Ansicht eines Beispielteilmodells. Über diese Ansicht können auch die Beziehungen grafisch repräsentierter Elemente der direkt übereinander liegenden Einzelebenen visualisiert werden. Diese Beziehungen verbinden im Metamodell die einzelnen Gegenstandsebenen und finden in der grafischen 3-Ebenen-Ansicht ihre Entsprechung als Linien zwischen den grafischen Einzelebenenansichten. Sie werden sowohl im Metamodell als auch in der grafischen Darstellung eines konkreten Modells als *Inter-Ebenen-Beziehungen* bezeichnet.

Beispiel.z3lgm - KDMS-RIS-PACS - Matrix-Sicht #1

	ADT-Informationen	Anforderung Radiologie	Befundbericht Radiologie	Behandlungsplan	Diagnose	Einzelauftrag Radiologie	Fall	Leistungsziffern	Patienten-Identifikation	Prozedur	Radiolog. Bild	Radiologische Untersuchung	Terminierter Einzelauftrag	Wartezimmerliste
1.1.1.3 administrative Patientenaufnahme (stationär)	■						■	■						
1.8.3 Entgegennahme und Darstellung von Befunden		■								■				
4.3 Kosten- und Leistungsrechnung						■								
Befundanforderung Bildgebende Diagnostik / Strahlentherapie	■				■	■		■						
Diagnosen- und Prozedurenverschlüsselung		■	■						■					
Klinische Konferenz durchführen		■	■							■				
Patientenabrechnung (stationär)	■			■		■		■	■					
Patientenanmeldung Radiologie	■					■		■			■	■	■	
Radiolog. Bildbefundung	■	■				■		■		■	■	■	■	
Radiolog. Bilderzeugung	■					■		■		■	■	■	■	
Radiologische Diagnostik	■					■		■		■	■	■	■	
stationäre Leistungszifferndokumentation							■				■			
Terminvergabe und Untersuchungsplanung in der Radiologie	■				■	■		■				■		

Auswahl
 Zeilen: Aufgabe
 Spalten: Objekttyp
 nur Teilelemente anzeigen

Verbindungstyp
 ■ Aufgabe bearbeitet und interpretiert Objekttyp
 ■ Aufgabe bearbeitet Objekttyp
 ■ Aufgabe interpretiert Objekttyp

Position
 Zeile:
 Spalte:

F.A.E Eigenschaftsdialoge

Für jedes Modellelement existiert ein Eigenschaftsdialog, in dem u.a.

- der Name und die Beschreibung des Elements editiert werden können,
- Verbindungen zu anderen Elementen hergestellt werden können,
- benutzerdefinierte Eigenschaften in Form von Text, Auswahllisten, Radiobuttons, Hyperlinks, Checkboxes und Kennzahlen bzw. Kennzahlformeln angegeben werden können. Der 3LGM²-Baukasten verfügt ab der Version 3.2 über komplexe Funktionen zur Kennzahlverrechnung⁶ gemäß [33].

Siehe als Beispiel Abbildung 9.

F.B Aktuelle Version des 3LGM²-Baukastens

April 2010: Der 3LGM²-Baukasten ist in der Version 3.2.6 verfügbar⁷.

F.C Anforderungen an den 3LGM²-Baukasten aus dem Metamodell

F.C.A Anforderungen seitens der Elementklassen

F.C.A-1.1 Aufgabe

- Siehe E.B.B-2.6

⁶ http://www.3lgm2.de/Publikationen/Dokumente/Kennzahlverrechnung_Anwender_Dokumentation_V06.pdf

⁷ http://www.3lgm2.de/Downloads/3LGM2_Baukasten/index.jsp

- Alle *Aufgaben* finden und markieren können, die keinen *Objektyp* interpretieren.
- Alle *Aufgaben* finden und markieren können, die keinen *Objektyp* bearbeiten.
- Zu einer ausgewählten *Aufgabe* alle *Aufgaben* finden und markieren können, die einen *Objektyp* bearbeiten, der von der ausgewählten *Aufgabe* interpretiert wird.
- Es wird eine Funktion benötigt, mit der alle oder eine Auswahl der vererbten Eigenschaften und Beziehungen einer *Gruppenaufgabe* explizit auf alle *Unteraufgaben* übertragen und bei der *Gruppenaufgabe* entfernt werden können.
- *Unteraufgaben* einer *Gruppenaufgabe* können in der graphischen Darstellung durch eine spezielle Funktion ausgeblendet und später wieder eingeblendet werden (siehe E.B.B-2.2.3), wenn keine der *Unteraufgaben* der *Gruppenaufgabe* auch *Unteraufgabe* einer anderen *Gruppenaufgabe* ist. Im Browser ist das Ausblenden in jedem Fall möglich, da dort die *Unteraufgaben* bei multipler Zugehörigkeit zu *Gruppenaufgaben* auch entsprechend oft aufgeführt werden. Das Ausblenden dient ggf. der besseren Übersicht, verändert aber in keiner Weise das eigentliche Modell und hat keinen Einfluss auf Auswertungen und Analysen. Das Aus-/Einblenden erfolgt stets teilmodellspezifisch. Die *Gruppenaufgabe* wird nach dem Ausblenden der *Unteraufgaben* in der graphischen Darstellung so markiert, dass erkennbar ist, dass die *Unteraufgaben* ausgeblendet wurden. Bei Bedarf können die *Unteraufgaben* wieder eingeblendet und die Markierung aufgehoben werden. Alle Beziehungen der *Unteraufgaben* zu *Objektypen* und zu *Nutzungen* werden bei dem Ausblenden graphisch so dargestellt, als ob sie zur *Gruppenaufgabe* gehörten. Dabei sind solche Beziehungen, die für alle *Unteraufgaben* gelten, so darzustellen wie die Originalbeziehungen. Beziehungen, die nur für einige der *Unteraufgaben* gelten, sind ‚schwächer‘, z. B. durch graue oder unterbrochene Linien, darzustellen. Das Ausblenden und wieder Einblenden erfolgen jeweils durch eine einfache Aktion auf der *Gruppenaufgabe* (z. B. Shift-Doppelklick) in der graphischen Ansicht. Nach dem Ausblenden der *Unteraufgaben* kann die *Gruppenaufgabe* in der graphischen Darstellung beliebig formatiert und positioniert werden. Mit dem Wieder-Einblenden der *Unteraufgaben* wird die Formatierung und Position der *Gruppenaufgabe* wieder auf die letzten Werte vor dem Ausblenden zurückgesetzt. Bei einem erneuten Ausblenden der *Unteraufgaben* wird die Formatierung und Position der *Gruppenaufgabe* wieder auf die letzten Werte vor dem Einblenden zurückgesetzt. Auch diese Formatierungsregeln gelten teilmodellspezifisch.
- Das *Vergrößern* einer *Gruppenaufgabe* wird wie in E.B.B-2.2.3 beschrieben unterstützt, wenn keine ihrer *Unteraufgaben* auch *Unteraufgabe* einer anderen *Gruppenaufgabe* ist. Dabei ist sicherzustellen, dass für die vergrößerten *Gruppenaufgaben* die ursprüngliche *Verfeinerung* gespeichert bleibt und jederzeit wiederhergestellt werden kann. Nach dem *Vergrößern* von *Gruppenaufgaben* beziehen sich alle Analysen und Algorithmen im 3LGM²-Baukasten nur auf die durch das *Vergrößern* entstandene Variante des Modells. In der Darstellung im 3LGM²-Baukasten bewirkt das *Vergrößern* das Ausblenden der *Unteraufgaben*. Die *Gruppenaufgabe* wird in sowohl im Browser als auch in den graphischen Darstellungen aller Teilmodelle so markiert, dass erkennbar ist, dass *vergrößert* wurde. Bei Bedarf kann jede *Vergrößerung* rückgängig gemacht, die *Unteraufgaben* wieder eingeblendet und die Markierung aufgehoben werden. Da alle *Assoziationen* der *Unteraufgaben* zu *Objektypen* und zu *Nutzungen* wie in E.B.B-2.2.3 beschrieben der *Gruppenaufgabe* zugeordnet werden, können sie beim Ausblenden in üblicher Weise dargestellt werden. Das *Vergrößern* und wieder *Verfeinern* erfolgen jeweils durch eine einfache Aktion auf der *Gruppenaufgabe* (z. B. Shift-STRG-Doppelklick) im „Alle-Elemente-Browser“. Die *Vergrößerung* gilt für das ganze Modell und ist nicht teilmodellspezifisch. Nach dem *Vergrößern* kann die *Gruppenaufgabe* in den graphischen Darstellungen der Teilmodelle jeweils beliebig formatiert und positioniert werden. Mit dem Rückgängigmachen, d. h. dem *Verfeinern* der *Gruppenaufgabe* wird die Formatierung und Position der *Gruppenaufgabe* in jedem Teilmodell wieder auf die letzten Werte vor dem *Vergrößern* zurückgesetzt. Bei einem erneuten *Vergrößern* wird die Formatierung und Position der *Gruppenaufgabe* wieder auf die letzten Werte vor der *Verfeinerung* zurückgesetzt. Eigenschaften und *Assoziationen* von *Gruppenaufgaben*, die als *Vergrößerung* ihrer *Unteraufgaben* gekennzeichnet sind, dürfen nicht verändert werden können. Beim Speichern des Modells wird der aktuelle *Vergrößerungs-/Verfeinerungs-Zustand* mit gespeichert. Eine *Gruppenaufgabe*, die als *Vergrößerung* gekennzeichnet ist, kann durch eine spezielle Funktion endgültig in eine *Elementaraufgabe* unter Beibehaltung der bei der *Vergrößerung* generierten Eigenschaften und *Assoziationen* umgewandelt werden. Die Rekonstruktion der ursprünglichen *Unteraufgaben* ist dann nicht mehr möglich.
- Bei allen Analysen müssen die Vererbungsregeln (siehe E.B.B-2.2.2) berücksichtigt werden. Dies gilt besonders bei den Beziehungen zu *Nutzung*.

- Wird einer *Aufgabe* über die Beziehung *Nutzung* ein *Anwendungsbaustein* zugeordnet und ist diesem *Anwendungsbaustein* bereits ein *Softwareprodukt* zugeordnet, so sollte der Modellierer gefragt werden, ob die dem *Anwendungsbaustein* zugeordnete *Aufgabe* auch gleich dem *Softwareprodukt* zugeordnet werden soll.

F.C.A-1.2 Objekttyp

- Siehe E.B.B-1.7.
- Im Eigenschaftsdialog für *Objekttypen* Felder für *Objektklasse* und *Merkmalsarten* vorsehen
- *Unterobjekttypen* eines *Gruppenobjekttyps* können in der graphischen Darstellung sowie im Browser durch eine spezielle Funktion ausgeblendet und später wieder eingeblendet werden (siehe E.B.B-1.3.3). Das Ausblenden dient ggf. der besseren Übersicht, verändert aber in keiner Weise das eigentliche Modell und hat keinen Einfluss auf Auswertungen und Analysen. Das Aus-/Einblenden erfolgt stets teilmodellspezifisch. Der *Gruppenobjekttyp* wird nach dem Ausblenden der *Unterobjekttypen* in der graphischen Darstellung so markiert, dass erkennbar ist, dass die *Unterobjekttypen* ausgeblendet wurden. Bei Bedarf können die *Unterobjekttypen* wieder eingeblendet und die Markierung aufgehoben werden. Alle Beziehungen der *Unterobjekttypen* zu *Aufgaben*, *Anwendungsbausteinen*, *Sendeschnittstellen*, *Empfangsschnittstellen* und zu *Kommunikationsverbindungen* werden bei dem Ausblenden graphisch so dargestellt, als ob sie zum *Gruppenobjekttyp* gehörten. Dabei sind solche Beziehungen, die für alle *Unterobjekttypen* gelten, so darzustellen wie die Originalbeziehungen. Beziehungen, die nur für einige der *Unterobjekttypen* gelten, sind ‚schwächer‘, z. B. durch graue oder unterbrochene Linien, darzustellen. Das Ausblenden und wieder Einblenden erfolgen jeweils durch eine einfache Aktion auf dem *Gruppenobjekttyp* (z. B. Shift-Doppelklick) in der graphischen Ansicht. Nach dem Ausblenden der *Unterobjekttypen* kann der *Gruppenobjekttyp* in der graphischen Darstellung beliebig formatiert und positioniert werden. Mit dem Wieder-Einblenden der *Unterobjekttypen* wird die Formatierung und Position des *Gruppenobjekttyps* wieder auf die letzten Werte vor dem Ausblenden zurückgesetzt. Bei einem erneuten Ausblenden der *Unterobjekttypen* wird die Formatierung und Position des *Gruppenobjekttyps* wieder auf die letzten Werte vor dem Einblenden zurückgesetzt. Auch diese Formatierungsregeln gelten teilmodellspezifisch.
- Das *Vergrößern* von *Gruppenobjekttypen* wird unterstützt. Dabei ist sicherzustellen, dass für die bei der Vergrößerung vorübergehend aus dem Modell entfernten *Unterobjekttypen* die ursprüngliche *Verfeinerung* gespeichert bleibt und jederzeit wiederhergestellt werden kann. Nach dem *Vergrößern* beziehen sich alle Analysen und Algorithmen im 3LGM²-Baukasten nur auf die durch das *Vergrößern* entstandene Variante des Modells. In der Darstellung im 3LGM²-Baukasten bewirkt das *Vergrößern* das Ausblenden der *Teilobjekttypen*. Der *Gruppenobjekttyp* wird sowohl im Browser als auch in den graphischen Darstellungen aller Teilmodelle so markiert, dass erkennbar ist, dass *vergrößert* wurde. Bei Bedarf kann jede *Vergrößerung* rückgängig gemacht, die *Unterobjekttypen* wieder eingeblendet und die Markierung aufgehoben werden. Da alle *Assoziationen* der *Unterobjekttypen* wie in E.B.B-2.2.3 beschrieben dem *Gruppenobjekttyp* zugeordnet werden, können Sie beim Ausblenden in üblicher Weise dargestellt werden. Das *Vergrößern* und wieder *Verfeinern* erfolgen jeweils durch eine einfache Aktion auf dem *Gruppenobjekttyp* (z. B. Shift-STRG-Doppelklick) im „Alle-Elemente-Browser“. Die Vergrößerung gilt für das ganze Modell und ist nicht teilmodellspezifisch. Nach dem *Vergrößern* kann der *Gruppenobjekttyp* in den graphischen Darstellungen der Teilmodelle jeweils beliebig formatiert und positioniert werden. Mit dem Rückgängigmachen, d. h. dem *Verfeinern* des *Gruppenobjekttyps* wird die Formatierung und Position des *Gruppenobjekttyps* in jedem Teilmodell wieder auf die letzten Werte vor dem *Vergrößern* zurückgesetzt. Bei einem erneuten *Vergrößern* des *Gruppenobjekttyps* wird seine Formatierung und Position wieder auf die letzten Werte vor der *Verfeinerung* zurückgesetzt. Eigenschaften und *Assoziationen* von *Gruppenobjekttypen*, die als Vergrößerung gekennzeichnet sind, dürfen nicht verändert werden können. Beim Speichern des Modells wird der aktuelle Vergrößerungs-/Verfeinerungs-Zustand mit gespeichert. Ein *Gruppenobjekttyp*, der als Vergrößerung gekennzeichnet ist, kann durch eine spezielle Funktion endgültig in einen *Elementarobjekttyp* unter Beibehaltung der bei der Vergrößerung generierten Eigenschaften und *Assoziationen* umgewandelt werden. Die Rekonstruktion der ursprünglichen *Unterobjekttypen* ist dann nicht mehr möglich.
- Bei allen Analysen müssen die Vererbungsregeln (siehe E.B.B-1.3.2) berücksichtigt werden.

F.C.A-1.3 Begriffssystem

- Siehe F.C.A-1.2.
- *Begriffssysteme* sollten sich optisch (z.B. farblich) von normalen Objekttypen unterscheiden.

F.C.A-1.4 Nutzung

- siehe Kapitel F.G

F.C.A-1.5 Organisationseinheit

- Siehe E.B.B-5.5
- Siehe Modellierungshinweise zur Vererbung (E.B.B-5.6)
- Bei allen Analysen müssen die Vererbungsregeln (siehe E.B.B-5.2.2) und die Modellierungshinweise zur Vererbung (E.B.B-5.6) berücksichtigt werden.
- Das Ausblenden von *Organisationseinheiten* ist im Browser möglich, da dort die *Teilorganisationseinheiten* bei multipler Zugehörigkeit zu *Gruppenorganisationseinheiten* auch entsprechend oft aufgeführt werden. Das Ausblenden dient der besseren Übersicht, verändert aber in keiner Weise das eigentliche Modell und hat keinen Einfluss auf Auswertungen und Analysen. Das Aus-/Einblenden erfolgt teilmodellspezifisch. Bei Bedarf können die *Teilorganisationseinheiten* wieder eingeblendet werden. (z. B. Doppelklick auf ein plus/minus-Symbol) im Browser.
- Bei der *Vergrößerung* von *Gruppenorganisationseinheiten* wie in E.B.B-5.2.3 beschrieben ist sicherzustellen, dass für die vergrößerten *Teilorganisationseinheiten*, die ursprüngliche *Verfeinerung* gespeichert bleibt und jederzeit wiederhergestellt werden kann. Nach dem *Vergrößern* von *Gruppenorganisationseinheiten* beziehen sich alle Analysen und Algorithmen im 3LGM²-Baukasten nur auf die durch das *Vergrößern* entstandene Variante des Modells. In der Darstellung im 3LGM²-Baukasten bewirkt das *Vergrößern* das Ausblenden der *Teilorganisationseinheiten*. Die vergrößerte *Organisationseinheit* wird im Browser so markiert, dass erkennbar ist, dass *vergrößert* wurde. Bei Bedarf kann jede *Vergrößerung* rückgängig gemacht, die *Teilorganisationseinheiten* wieder eingeblendet und die Markierung aufgehoben werden. Da alle *Assoziationen* der *Teilorganisationseinheiten* wie in E.B.B-5.2.3 beschrieben der *Gruppenorganisationseinheit* zugeordnet werden, können Sie beim Ausblenden in üblicher Weise dargestellt werden. Das *Vergrößern* und wieder *Verfeinern* erfolgen jeweils durch eine einfache Aktion auf der *Gruppenorganisationseinheit* (z. B. Shift-STRG-Doppelklick) im „Alle-Elemente-Browser“. Die *Vergrößerung* gilt für das ganze Modell und ist nicht teilmodellspezifisch. Eigenschaften und *Assoziationen* von vergrößerten *Gruppenorganisationseinheiten* dürfen nicht verändert werden können. Beim Speichern des Modells wird der aktuelle *Vergrößerungs-/Verfeinerungs-Zustand* mitgespeichert. Eine vergrößerte *Gruppenorganisationseinheit* kann durch eine spezielle Funktion endgültig in eine *Elementarorganisationseinheit* unter Beibehaltung der bei der *Vergrößerung* generierten Eigenschaften und *Assoziationen* umgewandelt werden. Die Rekonstruktion der ursprünglichen *Teilorganisationseinheiten* ist dann nicht mehr möglich.
- Zur Unterstützung der Modellierung bietet der 3LGM²-Baukasten eine Funktion, mit der alle oder eine Auswahl der vererbaren Eigenschaften und Beziehungen einer *Gruppenorganisationseinheit* explizit auf alle *Teilorganisationseinheiten* übertragen und bei der *Gruppenorganisationseinheit* entfernt werden.

F.C.A-1.6 Prozess

- Siehe E.B.B-6.4
- Während der Modellierung des *Prozesses* überwacht der 3LGM²-Baukasten die in E.B.B-6.3 beschriebenen Bedingungen, die für *Aufgaben* innerhalb eines Prozesses erfüllt sein müssen.
- Die aufeinanderfolgenden *Aufgaben* des Informationsprozesses sind mit natürlichen Zahlen durchnummeriert.
- Ein im 3LGM²-Baukasten implementierter Algorithmus soll automatisch bestimmen, über welche Pfade Daten auf der *Logischen Werkzeugebene* kommuniziert werden können, die innerhalb einer *Aufgabe* eines Prozesses erzeugt werden und bei weiteren *Aufgaben* benötigt werden. Die *Kommunikationsverbindungen*, über welche die Daten transportiert werden, sind ebenfalls durchnummeriert.
- Im Eigenschaftsdialog eines *Prozesses* soll man durch die Informationsprozessschritte navigieren können und dabei die Kommunikationspfade von Objekttypen nachvollziehen können.

F.C.A-1.7 Anwendungsbaustein

- *Teilanwendungsbausteine* eines *Anwendungsbausteins* können in der graphischen Darstellung durch eine spezielle Funktion ausgeblendet und später wieder eingeblendet werden (siehe E.B.D-1.3.3). Dies dient ggf.

der besseren Übersicht, verändert aber in keiner Weise das eigentliche Modell und hat keinen Einfluss auf Auswertungen und Analysen. Das Aus-/Einblenden erfolgt stets teilmodellspezifisch. Der *Anwendungsbaustein* wird nach dem Ausblenden der *Teilanwendungsbausteine* in der graphischen Darstellung so markiert, dass erkennbar ist, dass die *Teilanwendungsbausteine* ausgeblendet wurden. Bei Bedarf können die *Teilanwendungsbausteine* wieder eingeblendet und die Markierung aufgehoben werden. Alle Beziehungen der *Teilanwendungsbausteine* zu *Objektyp*, *Nutzung*, *Physischem Datenverarbeitungssystem* und *Kommunikationsschnittstelle* werden beim Ausblenden graphisch so dargestellt, als ob sie zum *Anwendungsbaustein* gehörten. Dabei sind solche Beziehungen, die für alle *Teilanwendungsbausteine* gelten, so darzustellen wie die Originalbeziehungen. Beziehungen, die nur für einige der *Teilanwendungsbausteine* gelten, sind ‚schwächer‘, z. B. durch graue oder unterbrochene Linien, darzustellen. Das Ausblenden und wieder Einblenden erfolgen jeweils durch eine einfache Aktion auf dem *Anwendungsbaustein* (z. B. Shift-Doppelklick) in der graphischen Ansicht. Nach dem Ausblenden der *Teilanwendungsbausteine* kann der *Anwendungsbaustein* in der graphischen Darstellung beliebig formatiert und positioniert werden. Mit dem Wieder-Einblenden der *Teilanwendungsbausteine* wird die Formatierung und Position des *Anwendungsbausteins* wieder auf die letzten Werte vor dem Ausblenden zurückgesetzt. Bei einem erneuten Ausblenden der *Teilanwendungsbausteine* wird die Formatierung und Position des *Anwendungsbausteins* wieder auf die letzten Werte vor dem Einblenden zurückgesetzt. Auch diese Formatierungsregeln gelten teilmodellspezifisch.

- Das *Vergrößern* von *Gruppenanwendungsbausteinen* wird wie in E.B.D-1.3.3 beschrieben unterstützt. Dabei ist sicherzustellen, dass für die vergrößerten *Gruppenanwendungsbausteine* die ursprüngliche *Verfeinerung* gespeichert bleibt und jederzeit wiederhergestellt werden kann. Nach dem *Vergrößern* von *Gruppenanwendungsbausteinen* beziehen sich alle Analysen und Algorithmen im 3LGM²-Baukasten nur auf die durch das *Vergrößern* entstandene Variante des Modells. In der Darstellung im 3LGM²-Baukasten bewirkt das *Vergrößern* das Ausblenden der *Teilanwendungsbausteine*. Der vergrößerte *Anwendungsbaustein* wird sowohl im Browser als auch in den graphischen Darstellungen aller Teilmodelle so markiert, dass erkennbar ist, dass er *vergrößert* wurde. Bei Bedarf kann jede *Vergrößerung* rückgängig gemacht, die *Teilanwendungsbausteine* wieder eingeblendet und die Markierung aufgehoben werden. Da alle *Assoziationen* der *Teilanwendungsbausteine* zu *Objektyp*, *Nutzung*, *Physischem Datenverarbeitungssystem* und *Kommunikationsschnittstelle* wie in E.B.D-1.3.3 beschrieben dem vergrößerten *Gruppenanwendungsbaustein* zugeordnet werden, können sie beim Ausblenden in üblicher Weise dargestellt werden. Das *Vergrößern* und wieder *Verfeinern* erfolgen jeweils durch eine einfache Aktion auf dem vergrößerten *Anwendungsbaustein* (z. B. Shift-STRG-Doppelklick) im „Alle-Elemente-Browser“. Die *Vergrößerung* gilt für das ganze Modell und ist nicht teilmodellspezifisch. Nach dem *Vergrößern* eines *Gruppenanwendungsbausteins* kann dieser in den graphischen Darstellungen der Teilmodelle jeweils beliebig formatiert und positioniert werden. Mit dem Rückgängigmachen, d. h. dem *Verfeinern* des vergrößerten *Anwendungsbausteins* wird seine Formatierung und Position in jedem Teilmodell wieder auf die letzten Werte vor dem *Vergrößern* zurückgesetzt. Bei einem erneuten *Vergrößern* eines *Gruppenanwendungsbausteins* wird seine Formatierung und Position wieder auf die letzten Werte vor der *Verfeinerung* zurückgesetzt. Eigenschaften und *Assoziationen* eines vergrößerten *Gruppenanwendungsbausteins* dürfen nicht verändert werden können. Beim Speichern des Modells wird der aktuelle *Vergrößerungs-/Verfeinerungs-Zustand* mitgespeichert. Ein vergrößerter *Anwendungsbaustein* kann durch eine spezielle Funktion endgültig in einen *Elementar-Anwendungsbaustein* unter Beibehaltung der bei der *Vergrößerung* generierten Eigenschaften und *Assoziationen* umgewandelt werden. Die Rekonstruktion der ursprünglichen *Teilanwendungsbausteine* ist dann nicht mehr möglich.
- Bei allen Analysen müssen die Vererbungsregeln (siehe E.B.D-1.3.2) berücksichtigt werden.
- Die Modellierung von *Kommunikationsverbindungen* zwischen den *Teilanwendungsbausteinen* desselben *Anwendungsbausteins* ist zu verhindern (siehe E.B.D-1.3.1).
- Beim Modellieren einer Teil-von-Beziehung zwischen einem *Anwendungsbaustein* und dem künftigen *Teilanwendungsbaustein* wird eine Unterstützung dafür angeboten, Eigenschaften, wenn gewünscht, von dem übergeordneten *Anwendungsbaustein* auf den *Teilanwendungsbaustein* zu übertragen.
- Bei einem *Anwendungsbaustein*, der *Objektypen* speichert, soll dies durch das Anzeigen eines Datenbanksymbols („gelbe Tonne“) angezeigt werden.
- Es sind Analysen erforderlich, die für einen *Anwendungsbaustein* anzeigen, welche *Aufgaben* nicht mehr unterstützt werden können, wenn er ausfällt (vgl. E.B.D-1.10)

- Zu E.B.D-1.10: Um Fragen nach der Erfüllung von Integrationsarten beantworten zu können, muss im 3LGM²-Baukasten ein Domänenkonzept in Anlehnung an [16] implementiert werden. Domänen sind Mengen von *Anwendungsbausteinen*. Domänen werden unterteilt in Anforderungsdomänen und Kommunikationsdomänen. Anforderungsdomänen geben vor, für welche *Anwendungsbausteine* die Forderung nach Erfüllung einer bestimmten Integrationsart besteht. Die zugehörige Kommunikationsdomäne gibt die Menge von *Anwendungsbausteinen* vor, für die die Integrationsart tatsächlich umgesetzt ist. Durch den Vergleich von Anforderungs- und Kommunikationsdomänen lässt sich die Erfüllung einer Integrationsart bestimmen. Die Anforderungsdomäne muss zur Erfüllung der Integrationsanforderungen eine Teilmenge der Kommunikationsdomäne sein.

F.C.A-1.8 Softwareprodukt

- *Softwareprodukte* werden nicht graphisch dargestellt sondern erscheinen nur im „Alle-Elemente“-Browser.
- Der Name des *Softwareprodukts*, auf dem ein *Anwendungssystem* basiert, wird in der grafischen Ansicht des *Anwendungssystems* in Klammern hinter dem Namen des *Anwendungssystems* angegeben.

F.C.A-1.9 Kommunikationsverbindung, Sende- und Empfangsschnittstelle

- Sowohl *Sende-* als auch *Empfangsschnittstellen* werden graphisch mit demselben ausgefüllten kleinen Kreis dargestellt. Eine *Kommunikationsverbindung* wird als Pfeil von der *Sende-* zu der *Empfangsschnittstelle* dargestellt (siehe Abbildung 11).
- Die Modellierung von *Kommunikationsverbindungen* zwischen *Kommunikationsschnittstellen* desselben *Anwendungsbausteins* ist zu verhindern (vgl. E.B.D-3.1).

F.C.A-1.10 Physisches Datenverarbeitungssystem

- *Teildatenverarbeitungssysteme* eines *Physischen Datenverarbeitungssystems* können in der graphischen Darstellung durch eine spezielle Funktion ausgeblendet und später wieder eingeblendet werden (siehe E.B.E-1.2.3). Dies dient ggf. der besseren Übersicht, verändert aber in keiner Weise das eigentliche Modell und hat keinen Einfluss auf Auswertungen und Analysen. Das Aus-/Einblenden erfolgt stets teilmodellspezifisch. Das *Physische Datenverarbeitungssystem* wird nach dem Ausblenden der *Teildatenverarbeitungssysteme* in der graphischen Darstellung so markiert, dass erkennbar ist, dass die *Teildatenverarbeitungssysteme* ausgeblendet wurden. Bei Bedarf können die *Teildatenverarbeitungssysteme* wieder eingeblendet und die Markierung aufgehoben werden. Alle Beziehungen der *Teildatenverarbeitungssysteme* zu *Anwendungsbausteinen* und die *Datenübertragungsverbindungen* zu anderen *Physischen Datenverarbeitungssystemen* werden bei dem Ausblenden graphisch so dargestellt, als ob sie zu dem *Physischen Datenverarbeitungssystem* gehörten. Dabei sind solche Beziehungen, die für alle *Teildatenverarbeitungssysteme* gelten, so darzustellen wie die Originalbeziehungen. Beziehungen, die nur für einige der *Teildatenverarbeitungssysteme* gelten, sind ‚schwächer‘, z. B. durch graue oder unterbrochene Linien, darzustellen. Das Ausblenden und wieder Einblenden erfolgen jeweils durch eine einfache Aktion auf dem *Physischen Datenverarbeitungssystem* (z. B. Shift-Doppelklick) in der graphischen Ansicht. Nach dem Ausblenden der *Teildatenverarbeitungssysteme* kann das *Physische Datenverarbeitungssystem* in der graphischen Darstellung beliebig formatiert und positioniert werden. Mit dem Wiedereinblenden der *Teildatenverarbeitungssysteme* wird die Formatierung und Position des *Physischen Datenverarbeitungssystems* wieder auf die letzten Werte vor dem Ausblenden zurückgesetzt. Bei einem erneuten Ausblenden der *Teildatenverarbeitungssysteme* wird die Formatierung und Position des *Physischen Datenverarbeitungssystems* wieder auf die letzten Werte vor dem Einblenden zurückgesetzt. Auch diese Formatierungsregeln gelten teilmodellspezifisch.
- Das *Vergrößern* von *Gruppdatenverarbeitungssystemen* wird wie in E.B.E-1.2.3 beschrieben unterstützt. Dabei ist sicherzustellen, dass für die vergrößerten *Gruppdatenverarbeitungssysteme* die ursprüngliche *Verfeinerung* gespeichert bleibt und jederzeit wiederhergestellt werden kann. Nach dem *Vergrößern* von *Gruppdatenverarbeitungssystemen* beziehen sich alle Analysen und Algorithmen im 3LGM²-Baukasten nur auf die durch das *Vergrößern* entstandene Variante des Modells. In der Darstellung im 3LGM²-Baukasten bewirkt das *Vergrößern* das Ausblenden der *Teildatenverarbeitungssysteme*. Das vergrößerte *Physische Datenverarbeitungssystem* wird sowohl im Browser als auch in den graphischen Darstellungen aller Teilmodelle so markiert, dass erkennbar ist, dass *vergrößert* wurde. Bei Bedarf kann jede *Vergrößerung* rückgängig gemacht, die *Teildatenverarbeitungssysteme* wieder eingeblendet und die Markierung aufgehoben werden. Da alle *Datenübertragungsverbindungs-Assoziationen* der *Teildatenverarbeitungssysteme* zu anderen

Physischen Datenverarbeitungssystemen sowie die *Assoziationen* zu *Anwendungsbausteinen* wie in E.B.D-1.3.3 beschrieben dem vergrößerten *Physischen Datenverarbeitungssystem* zugeordnet werden, können sie beim Ausblenden in üblicher Weise dargestellt werden. Die anderen *Assoziationen* entfallen. Das *Vergrößern* und wieder *Verfeinern* erfolgen jeweils durch eine einfache Aktion auf dem zusammenfassenden *Physischen Datenverarbeitungssystem* (z. B. Shift-STRG-Doppelklick) im „Alle-Elemente-Browser“. Die Vergrößerung gilt für das ganze Modell und ist nicht teilmodellspezifisch. Nach dem *Vergrößern* eines *Gruppendatenverarbeitungssystems* kann es in den graphischen Darstellungen der Teilmodelle jeweils beliebig formatiert und positioniert werden. Mit dem Rückgängigmachen, d. h. dem Verfeinern des vergrößerten *Physischen Datenverarbeitungssystems* wird seine Formatierung und Position in jedem Teilmodell wieder auf die letzten Werte vor dem *Vergrößern* zurückgesetzt. Bei einem erneuten *Vergrößern* eines *Gruppendatenverarbeitungssystems* wird seine Formatierung und Position wieder auf die letzten Werte vor der *Verfeinerung* zurückgesetzt. Eigenschaften und *Assoziationen* von vergrößerten *Physischen Datenverarbeitungssystemen* dürfen nicht verändert werden können. Beim Speichern des Modells wird der aktuelle Vergrößerungs-/Verfeinerungs-Zustand mit gespeichert. Ein vergrößertes *Physisches Datenverarbeitungssystem* kann durch eine spezielle Funktion endgültig in ein *Elementar-Datenverarbeitungssystem* unter Beibehaltung der bei der Vergrößerung generierten Eigenschaften und *Assoziationen* umgewandelt werden. Die Rekonstruktion der ursprünglichen *Teildatenverarbeitungssysteme* ist dann nicht mehr möglich.

- Die Modellierung von *Datenübertragungsverbindungen* zwischen den *Teildatenverarbeitungssystemen* desselben *Physischen Datenverarbeitungssystems* ist zu verhindern (siehe E.B.E-1.2.1).
- Beim Modellieren einer Teil-von-Beziehung zwischen einem *Physischen Datenverarbeitungssystem* und dem künftigen *Teildatenverarbeitungssystem* wird eine Unterstützung dafür angeboten, Eigenschaften, wenn gewünscht, von dem übergeordneten *Physischen Datenverarbeitungssystem* auf das *Teildatenverarbeitungssystem* zu übertragen.
- *Mitglieder* einer *Datenverarbeitungssystemgruppe* können in der graphischen Darstellung durch eine spezielle Funktion ausgeblendet und später wieder eingeblendet werden, wenn keines der *Mitglieder* auch *Mitglied* einer anderen *Datenverarbeitungssystemgruppe* ist, deren *Teildatenverarbeitungssysteme* nicht bereits selbst ausgeblendet sind. Im Browser ist das Ausblenden in jedem Fall möglich, da dort die *Mitglieder* bei multipler Zugehörigkeit zu *Datenverarbeitungssystemgruppen* auch entsprechend oft aufgeführt werden. Das Ausblenden dient ggf. der besseren Übersicht, verändert aber in keiner Weise das eigentliche Modell und hat keinen Einfluss auf Auswertungen und Analysen. Das Aus-/Einblenden erfolgt teilmodellspezifisch. Die *Datenverarbeitungssystemgruppe* wird nach dem Ausblenden der *Mitglieder* in der graphischen Darstellung so markiert, dass erkennbar ist, dass die *Mitglieder* ausgeblendet wurden. Bei Bedarf können die *Mitglieder* wieder eingeblendet und die Markierung aufgehoben werden. (z. B. Doppelklick auf ein Plus/Minus-Symbol) im Browser.

Bei allen Analysen müssen die Vererbungsregeln (siehe E.B.E-1.2.2) berücksichtigt werden.

F.C.A-1.11 Geräteklasse

- Das einer *Geräteklasse* zugeordnete Symbol wird auch jedem *Physischen Datenverarbeitungssystem* zugewiesen, das dieser *Geräteklasse* zugeordnet ist.
- Die wichtigsten bzw. die hier erwähnten Beispiele für *Geräteklassen* sollten mit dem 3LGM²-Baukasten voreingestellt schon ausgeliefert werden.
- Der Katalog der *Geräteklassen* wird im Browser dargestellt. Dabei sind ggf. zugewiesene graphische Symbole anzuzeigen.
- Zum Ausblenden und Vergrößern siehe F.C.A-1.10.

F.C.A-1.12 Standort

- Zum Ausblenden und Vergrößern siehe F.C.A-1.10.

F.C.A-1.13 Datenübertragungsverbindung

- Im 3LGM²-Baukasten können den einzelnen *Übertragungsmedien* bestimmte graphische Symbole zugewiesen werden, die von den zugeordneten *Datenübertragungsverbindungen* übernommen werden können.

- Eine *Datenübertragungsverbindung* ist eine gerichtete Beziehung. Der 3LGM²-Baukasten soll aber bei dem Wunsch zwei *Physische Datenverarbeitungssysteme* zu verbinden als Default stets zwei Verbindungen in jeweils gegensätzlicher Richtung anlegen und diese als eine mit zwei Pfeilspitzen gekennzeichnete Verbindung darstellen. Wenn eine Verbindung nur in einer Richtung gewünscht ist, soll nur eine Verbindung angelegt und diese mit einem Pfeil in Übertragungsrichtung dargestellt werden.
- Zwischen zwei *Physischen Datenverarbeitungssystemen* können mehrere *Datenübertragungsverbindungen* modelliert werden, die dann auch unterscheidbar graphisch dargestellt werden.

F.C.A-1.14 Übertragungsmedium

- Im 3LGM²-Baukasten können den einzelnen *Übertragungsmedien* bestimmte graphische Symbole zugewiesen werden, die von den zugeordneten *Datenübertragungsverbindungen* übernommen werden können.
- Die wichtigsten bzw. die hier erwähnten Beispiele für *Übertragungsmedien* sollten mit dem 3LGM²-Baukasten voreingestellt schon ausgeliefert werden.
- Der Katalog der *Übertragungsmedien* wird im Browser dargestellt. Dabei sind ggf. zugewiesene graphische Symbole anzuzeigen.
- Die Ausnahme von der Vererbung in E.B.E-5 ist besonders bei Analysen zu beachten! Ansonsten sind Vergrößern und Verfeinern ähnlich wie bei *Organisationseinheiten* (vgl. E.B.B-5) zu handhaben, da die *Übertragungsmedien* als Katalog nicht auf den graphischen Darstellungen der Ebenen erscheinen.

F.C.A-1.15 Nachrichtentyp (3LGM²-M)

- Es muss sichergestellt werden, dass jeder *Nachrichtentyp* einem *Kommunikationsstandard* zugeordnet wird. Es soll leicht möglich sein, bei dem Modellieren eines *Nachrichtentyps* einen neuen *Kommunikationsstandard* anzulegen.
- Die *Nachrichtentypen* sollen im Browser gruppiert nach *Kommunikationsstandard* aufgeführt werden.
- Die wichtigsten bzw. die hier erwähnten Beispiele für *Nachrichtentypen* und *Kommunikationsstandards* sollten mit dem 3LGM²-Baukasten voreingestellt schon ausgeliefert werden.
- Für jeden Objekttyp wird automatisch ein Dummy-Nachrichtentyp angelegt.

F.C.A-1.16 Service (3LGM²-S)

- *Services* werden im 3LGM²-Baukasten nicht grafisch dargestellt. Im Eigenschaftsdialog von *Services* sollten deshalb alle möglichen Beziehungen zu anderen Elementen leicht hergestellt werden können.

F.C.A-1.17 Serviceklasse

- *Serviceklassen* werden wie *Kommunikationsstandards* nicht graphisch dargestellt.

F.C.A-1.18 Aufrufende und bereitstellende Schnittstelle (3LGM²-S)

- *Aufrufende* und *bereitstellende Schnittstellen* werden wie in Abbildung 20 dargestellt.

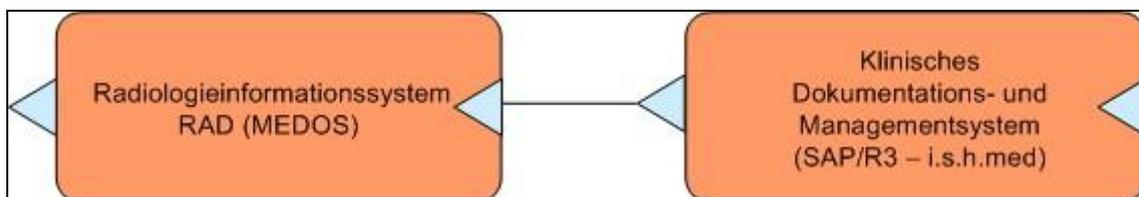


Abbildung 20: Darstellung von aufrufenden und bereitstellenden Schnittstellen: eine aufrufende Schnittstelle ist ein Dreieck mit der Spitze ins Innere des Anwendungsbausteins; eine bereitstellende Schnittstelle ist ein Dreieck mit der Spitze außerhalb des Anwendungsbausteins. Die Verbindung ist eine Kommunikationsverbindung.

F.C.A-1.19 Kommunikationsverbindung (3LGM²-S)

- Es ist sicherzustellen, dass die Menge der *Services*, zu denen die *Kommunikationsverbindung* in Beziehung steht, eine Teilmenge der Schnittmenge der jeweils von der beteiligten *aufzurufenden* und *bereitstellenden Schnittstelle* aufgerufenen bzw. bereitgestellten *Services* ist.

F.C.A-1.20 Anwendungsbaustein (3LGM²-S)

- Zu E.C.B-10.1:
 - Wenn für eine *Kommunikationsverbindung* ein *Vermittlungsbaustein* existiert, wird diese *Kommunikationsverbindung* immer so gezeichnet, dass sie durch den *Vermittlungsbaustein* führt. Ggf. wirkt der *Vermittlungsbaustein* wie ein Knickpunkt in der entsprechenden Linie. In Abbildung 17 ist ein *Vermittlungsbaustein* dargestellt, der dem Radiologieinformationssystem einen Service vermitteln kann, den sowohl das Patientenverwaltungssystem als auch das Ambulanzverwaltungssystem zur Verfügung stellt.
 - Für Analysen, insbesondere auf der PWE, ist zu berücksichtigen, dass zwischen dem *Vermittlungsbaustein* und den jeweiligen aufrufenden und bereitstellenden *Anwendungsbausteinen* Datenübertragung auf der physischen Werkzeugebene möglich sein muss.

F.D Handhabung des Basis-Metamodells und Erweiterungen des Metamodells

- Beim Anlegen eines neuen Modells sollte der Benutzer die Möglichkeit haben zum Basis-Metamodell beliebig viele von ihm gewünschte Erweiterungen zuzuschalten
- Der 3LGM²-Baukasten sollte erklären, welche Konsequenzen bzw. Möglichkeiten das bringt.
- Für bestehende Modelle: im Nachhinein Erweiterungen vornehmen können. Dabei sollte der 3LGM²-Baukasten soweit wie möglich automatisch die Klassen anlegen, die für die Erweiterungen benötigt werden (z. B. bei Nachrichtentypen). Die Erweiterung gilt immer für das gesamte Modell.
- Analysen/Exporte/Reports: Diese müssen für jede Erweiterung speziell angelegt werden.
- Reduzieren (nicht verwechseln mit Vergrößern/Verfeinern!!!):
 - Rein visuell: Vorübergehend nur bei der Darstellung die reduzierten Klassen weglassen
 - Tatsächlich: Das Modell tatsächlich (vorübergehend) ändern. (Könnte darüber gesteuert werden, dass der Benutzer gezwungen wird, vor dem Reduzieren das alte Modell zu speichern)

F.E Konvertierung von Modellen auf der Basis von Metamodell V1.x zu Modellen auf der Basis von 3LGM²**F.E.A Allgemein**

- Eine Konvertierung von V1.x ist nur nach 3LGM²-B und nicht direkt nach den Erweiterungen vorgesehen.

F.E.B Logische Werkzeugebene**F.E.B-1 Anwendungsbausteinkonfigurationen**

Siehe E.B.D-1.6.1, E.B.B-4.7, E.B.B-2.4.1

- Zunächst muss (von Hand) sichergestellt werden, dass das zu konvertierende Modell keine Anwendungsbausteinkonfiguration hat, die mehr als einen Anwendungsbaustein enthält. Der 3LGM²-Baukasten muss das prüfen und ggf. Meldungen ausgeben.
- Die verbliebenen Konfigurationen sind durch Einfügen von Instanzen der *Klasse Nutzung* zu ersetzen.

F.E.B-2 Nachrichtentypen und Kommunikationsstandards

- In V1.x sind im Gegensatz zu V3 *Nachrichtentypen* nicht unmittelbar *Kommunikationsstandards* zugeordnet. Die beim Konvertieren nötige Zuordnung soll wie folgt erfolgen: Für jeden *Nachrichtentyp* jede *Bausteinschnittstelle* bestimmen, in der er gesendet oder empfangen werden kann; ferner den *Kommunikationsstandard* bestimmen, der dieser *Bausteinschnittstelle* zugeordnet ist. Dann den *Nachrichtentyp* jedem dieser *Kommunikationsstandards* zuordnen. Es ist klar, dass dabei ein *Nachrichtentyp* auch mehreren *Kommunikationsstandards* zugeordnet werden kann.

F.E.C Physische Werkzeugebene

F.E.C-1 Teil-von-Beziehung

Siehe E.B.E-1.2.4, E.B.E-1.3.2.

Bei der Konvertierung muss für jede Teil-von-Beziehung gefragt werden, ob sie in eine *Zerlegung* oder in eine *Spezialisierung* übersetzt werden soll.

F.F Konvertierung zwischen den Metamodellen von 3LGM²

Zur Konvertierung zwischen 3LGM²-B, 3LGM²-M und 3LGM²-S siehe E.C.A-8 und E.C.B-12.1.

F.G Assoziationsklassen des Metamodells im 3LGM²-Baukasten

Assoziationsklassen sind *Assoziationen* mit Klasseneigenschaften, d. h. eine solche *Assoziation* wird durch Attribute oder Beziehungen zu anderen *Klassen* oder *Assoziationsklassen* näher beschrieben. Wie bei einfachen *Assoziationen* haben auch *Assoziationsklassen* zwei gerichtete Namen. Zusätzlich dazu besitzt die *Assoziationsklasse* einen Klassennamen. Je nach Kontext wird eine solche *Assoziationsklasse* als *Klasse* oder als *Assoziation* aufgefasst.

Im 3LGM²-Basis-Metamodell treten *Assoziationsklassen* an folgenden Stellen auf:

- 1.) Klassenname: *Nutzung*
 - Name 1: „Aufgabe wird unterstützt durch Anwendungsbaustein“
 - Name 2: „Anwendungsbaustein unterstützt Aufgabe“
- 2.) Klassenname: *Speicherung*
 - Name 1: „Objektyp wird gespeichert von Anwendungsbaustein“
 - Name 2: „Anwendungsbaustein speichert Objektyp“
- 3.) Klassenname: *Kommunikationsverbindung*
 - Name 1: „Sendeschnittstelle überträgt zu Empfangsschnittstelle“
 - Name 2: „Empfangsschnittstelle überträgt von Sendeschnittstelle“
- 4.) Klassenname: *Datenübertragungsverbindung*
 - Name 1: „Physisches Datenverarbeitungssystem überträgt Daten zu Physischem Datenverarbeitungssystem“
 - Name 2: „Physisches Datenverarbeitungssystem empfängt Daten von Physischem Datenverarbeitungssystem“

Im den erweiterten 3LGM²- Metamodellen gibt es folgende weitere *Assoziationsklassen*:

- 5.) Klassenname: *Datenübertragungsverbindung*
 - Name 1: „Physisches Datenverarbeitungssystem überträgt Daten zu Physischem Datenverarbeitungssystem“
 - Name 2: „Physisches Datenverarbeitungssystem empfängt Daten von Physischem Datenverarbeitungssystem“

6.) Klassenname: *Datenübertragungsverbindung*

- Name 1: „*Physisches Datenverarbeitungssystem überträgt Daten zu Physischem Datenverarbeitungssystem*“
- Name 2: „*Physisches Datenverarbeitungssystem empfängt Daten von Physischem Datenverarbeitungssystem*“

G Literatur

1. Berg M. Medical Work and the Computer-Based Patient Record: A Sociological Perspective. *Methods of Information in Medicine* 1998;37:294-301.
2. Winter AF, Zimmerling R, Bott OJ, Gräber S, Haas P, Hasselbring W, Haux R, Heinrich A, Jaeger R, Kock I, Möller DPF, Penger O-S, Prokosch H-U, Ritter J, Terstappen A, Winter A. Das Management von Krankenhausinformationssystemen: Eine Begriffsdefinition. *Informatik, Biometrie und Epidemiologie in Medizin und Biologie* 1998;29(2):93-105.
3. Brodie ML, Stonebraker M. *Migrating legacy systems*. San Francisco: Morgan Kaufmann; 1995.
4. Ferstl OK, Sinz EJ. *Grundlagen der Wirtschaftsinformatik*. München: R. Oldenbourg; 1994.
5. Reynolds M, Wejerfeld I. Health Informatics – Short Strategic Study – Health Information Infrastructure. Short Strategic Study: CEN/TC 251/SSS-HII INR; 2000 2000-05-11. Report No.: interim report v0.4.
6. Schütte R, Rothhove T. The Guidelines of Modeling as an approach to enhance the quality of information models. In: LING TW, RAM S, LEE ML, editors. *Conceptual Modeling - ER '98, 17th International Conference on Conceptual Modeling*, Singapore, November 16-19: Springer; 1998. p. 240-254.
7. Winter A, Haux R. A Three-Level Graph-Based Model for the Management of Hospital Information Systems. *Methods of Information in Medicine* 1995;34(4):378-396.
8. Winter A, Brigl B, Wendt T. Modeling Hospital Information Systems (Part 1): The Revised Three-Layer Graph-Based Meta Model 3LGM². *Methods Inf Med* 2003;42(5):544-551.
9. Sinz EJ. Ein Architekturrahmen für die Modellierung betrieblicher Informationssysteme. *Bamberger Beiträge zur Wirtschaftsinformatik: Otto-Friedrich-Universität Bamberg*; 1996. Report No.: 32.
10. Porter M. *Competitive Advantage: The Free Press*; 1985.
11. Object Management Group (OMG). *OMG Unified Modeling Language Specification: Object Management Group*; 2003 March 2003. Report No.: Version 1.5, formal/03-03-01.
12. Oestereich B. *Objektorientierte Softwareentwicklung - Analyse und Design mit der Unified Modeling Language*. München: Oldenbourg; 1998.
13. Ißler L. *Informationsverarbeitung im Informationsmanagement*. Leipzig: Universität Leipzig; 2008. doi:http://dol.dl.uni-leipzig.de/receive/DOLDissHabil_disshab_00001249
14. ISO. ISO 1087.
15. Hübner-Bloder G, Ammenwerth E, Brigl B, Winter A. Specification of a Reference Model for the Domain Layer of a Hospital Information System. In: Engelbrecht R, Geissbuhler A, Lovis C, Mihalas G, editors. *Connecting Medical Informatics and Bio-Informatics. Proceedings of Medical Informatics Europe (MIE 2005)*, Geneva, Aug 08 - Sep 01 2005. Amsterdam: IOS Press; 2005. p. 497-502.
16. Wendt T. *Modellierung und Bewertung von Integration in Krankenhausinformationssystemen [Dissertation]*. Leipzig: Universität Leipzig; 2005.
17. Winter A, Strübing A. Model-Based Assessment of Data Availability in Health Information Systems. *Methods Inf Med* 2008;47:417-424. doi:10.3414/ME9123
18. Ferstl OK, Sinz EJ. *Grundlagen der Wirtschaftsinformatik*. München: Oldenbourg; 2001.
19. Martin J. *Information Engineering, Book II: Planning & Analysis*. Englewood Cliffs: Prentice Hall; 1990.

20. Wöhe G. Einführung in die Allgemeine Betriebswirtschaftslehre: Franz Vahlen; 1978.
21. Hasselbring W. Information System Integration. *Communications of the ACM* 2000;43(6):33-38. doi:10.1145/336460.336472
22. Winter A, Strübing A, Brigl B, Haux R, Ißler L. Ontology-Based Assessment of Functional Redundancy in Health Information Systems. *Lecture Notes in Computer Science (LNCS)* 2009;5421:213–226.
23. Leiner F, Gaus W, Haux R, Knaup-Gregori P, Pfeiffer KP. *Medizinische Dokumentation: Grundlagen einer qualitätsgesicherten integrierten Krankenversorgung*. 4. Auflage ed. Stuttgart: Schattauer; 2003.
24. Brigl B, Strubing A, Wendt T, Winter A. Modeling Interdependencies between Information Processes and Communication Paths in Hospitals. *Methods Inf Med* 2006;45(2):216-24.
25. REFA Bundesverband e.V. Arbeitssystem: <http://www.refa.de/INTERNATIONAL/deutsch/system/system.htm#3>, accessed 2006-09-01.
26. Herbig B. Kurze Einführung in den soziotechnischen Systemansatz. In: Herbig B, Büssing A, editors. *Informations- und Kommunikationstechnologien im Krankenhaus*. Stuttgart: Schattauer; 2006. p. 1-4.
27. Winter A, Brigl B, Haux R, Ißler L, Strübing A. Ontology-Based Assessment of Functional Redundancy in Health Information Systems. In: Mothis; 2008; Toulouse; 2008. p. accepted.
28. Staemmler M. Towards sustainable e-health networks: does modeling support efficient management and operation? *Medinfo* 2007;12(Pt 1):53-7.
29. Stein S, Kühne S, Wagner J. Das Forschungsprojekt OrViA - Orchestrierung und Validierung integrierter Anwendungssysteme. In: Fähnrich K-P, Kühne S, Speck A, Wagner J, editors. *Integration betrieblicher Informationssysteme - Problemanalysen und Lösungsansätze des Model-Driven Integration Engineering*. Leipzig: Universität Leipzig, Leipziger Informatik Verbund; 2006. p. 3-12.
30. Fassnacht A, Widmann T. Geschäftslösungen entwickeln auf Basis von SOA. *OBJEKTSpektrum Online-Ausgabe* 2005(3/05):1-5.
31. Thränert M, Kühnert S. Model-Driven Integration Engineering und seien Anwendung im Projekt OrVia. In: Fähnrich K-P, Kühne S, Speck A, Wagner J, editors. *Integration betrieblicher Informationssysteme - Problemanalysen und Lösungsansätze des Model-Driven Integration Engineering*. Leipzig: Universität Leipzig, Leipziger Informatik Verbund; 2006. p. 13-19.
32. Winkler V. Identifikation und Gestaltung von Services. *Wirtschaftsinformatik* 2007;49(4):257-266.
33. Kutscha A. Modellierung von ökonomischen Bewertungskriterien zur Unterstützung des strategischen Informationsmanagements bei der Beurteilung von Krankenhausin-formationssystemen [Dissertation]. Leipzig: Universität Leipzig; 2006.

H Stichwortverzeichnis

A

abstrakte Klasse · 8
Adaptation · 39
Adaptation durch Parametrierung · 39
Aggregation · 9
Aktivität · 19
Ansichtswerkzeuge · 11
Anwendungsbaustein · 31, 55, 63
Anwendungssystem · 32
Arbeitssystem · 32
Assoziation · 9
Assoziationsklasse · 9
Aufgabe · 20, 21, 62
Aufrufende Schnittstelle · 61
ausführungsabhängig · 62

B

Bedingung · 10
Begriffssystem · 24
Bereitstellende Schnittstelle · 61

C

Cluster · 42
customizing · 39

D

Daten eines Objekttyps · 16
Datenübertragungsstrecke · 47
Datenübertragungsverbindung · 56
datenverarbeitende Werkzeuge · 30
Dokument · 53
Dokumententyp · 53
Dokumentfenster · 11

E

Elementaraufgabe · 23
Empfangsschnittstelle · 40, 56
Ergebnis · 60

F

Fachliche Ebene · 6, 15

G

Generalisierung · 9
Geräteklasse · 45
Gruppenanwendungsbaustein · 33
Gruppenelement · 9
Gruppengeräteklasse · 45
Gruppenobjekttyp · 17

I

Informationssystem · 3

K

Klasse · 8
Kommunikationsschnittstelle · 40, 62
Kommunikationsstandard · 54
Kommunikationsverbindung · 40, 55, 61
Komposition · 10
Krankenhausinformationssystem · 3

L

logische Werkzeuge · 31
Logische Werkzeugebene · 6, 32

M

Master · 17, 33, 34
Menüleiste · 11
Message · 53
Mitglied · 42
Modellbrowser · 11

N

Nachricht · 53
Nachrichtentyp · 53

O

Objektklasse · 16
Objekttyp · 16, 54, 64
Organisation · 3
Organisationseinheit · 26

Organisationssystem · 32

P

Parameter · 60

personengesteuerte Physische Datenverarbeitungssysteme · 31

Physische Werkzeugebene · 6

Physisches Datenverarbeitungssystem · 30

physisches Werkzeug der Datenverarbeitung · 30

R

rechnerbasierter Anwendungsbaustein · 32

S

Sendeschnittstelle · 40, 55

Service · 59

Serviceklasse · 61

Signal · 48

softwaregesteuerte Physische Datenverarbeitungssysteme · 30

Softwareprodukt · 39

Spezialisierung · 9, 20

Standardsymbolleiste · 11

Standort · 46

T

Teilanwendungsbaustein · 33

Teilanwendungssysteme · 33

Teilaufgabe · 20

Teilelement · 9

Teilorganisationseinheiten · 27

Teil-Standort · 46

U

Übertragungsmedium · 48

Unternehmensaufgabe · 23

Unterstützung · 25

V

Verarbeitung · 30

Verarbeitung von Daten · 30

Vermittlungsbaustein · 63

virtuelle Maschinen · 42

virtuelle physische Datenverarbeitungssysteme · 42

virtueller Server · 30

virtuelles Physisches Datenverarbeitungssystem · 43

Z

Zeichensymbolleiste · 11

Zerlegung · 20

zugangsvermittelndes Anwendungssystem · 37

I Danksagung

Wir bedanken uns für intensive, kritische und konstruktive Diskussionen und für vielfältige Anregungen bei

- Birgit Brigl
- Gert Funkat
- Anke Häber
- Reinhold Haux
- Uta Knöchel

Diese Arbeit wurde gefördert durch die Deutsche Forschungsgemeinschaft (Wi 1605/4-1, Wi 1605/4-2).

J Hinweise

Dieses Dokument beschreibt verbindlich das 3LGM² in der hier erwähnten Version. Dieses Dokument dient als Grundlage für alle weiteren Publikationen über 3LGM². Publikationen, die sich auf diese Version beziehen, dürfen nicht im Widerspruch zu diesem Dokument stehen.