

IMISE-REPORTS

Herausgegeben von Professor Dr. Markus Löffler

Birgit Brigl, Thomas Wendt, Alfred Winter

Ein UML-basiertes Meta-Modell zur Beschreibung von Krankenhausinformationssystemen

IMISE-REPORT Nr. 1/2003



UNIVERSITÄT LEIPZIG

Impressum

Herausgeber: Prof. Dr. Markus Löffler
Redakteurin: Birgit Brigl
Institut für Medizinische Informatik, Statistik und
Epidemiologie (IMISE)
Liebigstraße 27, 04103 Leipzig
Tel. 03 41 / 97 16 100, Fax 03 41 / 97 16 109
Internet: <http://www.imise.uni-leipzig.de/>
Druck des Einbands und Bindung: Buch- und
Offsetdruckerei Herbert Kirsten
Redaktionsschluss: 15.05.03
© IMISE 2003
Alle Rechte vorbehalten. Nachdruck nur mit
ausdrücklicher Genehmigung des Herausgebers und
mit Quellenangabe gestattet.
ISSN 1610-7233

Inhaltsverzeichnis

1	EINLEITUNG	2
1.1	PROBLEMATIK.....	2
1.2	ZIEL DER ARBEIT	3
2	META-MODELLE FÜR DIE MODELLIERUNG VON INFORMATIONSSYSTEMEN ...	4
3	EIN UML-BASIERTES META-MODELL ZUR BESCHREIBUNG VON KRANKENHAUSINFORMATIONSSYSTEMEN.....	5
3.1	ÜBERBLICK	5
3.2	UML-KLASSENDIAGRAMME	6
3.3	STATISCHE SICHT DES DREI-EBENEN-MODELLS.....	7
3.3.1	<i>Statische Sicht der Fachlichen Ebene</i>	<i>7</i>
3.3.2	<i>Statische Sicht der Logischen Werkzeugebene.....</i>	<i>10</i>
3.3.3	<i>Statische Sicht der Physischen Werkzeugebene</i>	<i>14</i>
3.3.4	<i>Interebenen-Beziehungen zwischen statischen Komponenten.....</i>	<i>17</i>
4	LITERATUR	22
4.1	ALLGEMEINE LITERATUR	22
4.2	LITERATUR ZUM 3LGM ²	23

1 Einleitung

1.1 Problematik

Krankenhäuser sind Dienstleistungsbetriebe, die sich durch eine sehr heterogene Unternehmensstruktur auszeichnen. Ein Krankenhaus ist aus einer Vielzahl uneinheitlicher Organisationseinheiten aufgebaut, die zur Unterstützung ihrer Informationsverarbeitung unterschiedlichste rechnerbasierte und nicht-rechnerbasierte Werkzeuge heranziehen müssen. Neben den klassischen Organisationseinheiten jedes Unternehmens wie Buchhaltung, Personalverwaltung oder Bestellwesen sind es insbesondere die Stationen und Ambulanzen, die medizinischen Leistungsstellen, die je nach ihrer Aufgabe Bilder oder Bildsequenzen (z. B. in der Röntgenabteilung), Signale (z. B. in der EKG-Abteilung), oder Text-Information 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. Die gilt auch für die externe Kommunikation mit den an der Behandlung des Patienten beteiligten Einrichtungen (z. B. Versicherungen, niedergelassene Ärzte oder Nachsorgeeinrichtungen). Die Tatsache, dass der Patient eines Krankenhauses nicht nur Kunde sondern auch Produkt ist und dass der Erfolg der erbrachten Leistungen von den über den Patienten vorliegenden Informationen abhängig ist, begründen im übrigen eine besondere Sorgfaltspflicht bei der Informationsverarbeitung.

Erfahrungen bei der Einführung von rechnerbasierten Werkzeugen in Krankenhäuser zeigen, dass es nicht möglich ist, Krankenhausinformationssysteme als abgegrenzte rechnerunterstützte Systeme zu betrachten, sondern dass sie als sozio-technisches System aufzufassen sind ([1]). Somit sind sie zugleich Teilsystem des Krankenhauses und spiegeln seine Heterogenität entsprechend wider. Aufgrund der unterschiedlichen Anforderungen an die Werkzeuge zur Informationsverarbeitung basieren diese häufig auf unterschiedlichsten Plattformen und unterschiedlichsten Softwareprodukten verschiedenster Hersteller. Hinzu kommt, dass Krankenhausinformationssysteme (im übrigen wie andere Informationssysteme im allgemeinen auch) historisch wachsen, so dass es neben modernen informationsverarbeitenden Werkzeugen immer auch sogenannte 'Altsysteme' (legacy systems; vgl. [2]) geben wird. Es kann nicht davon ausgegangen werden, dass alle informationsverarbeitenden Werkzeuge eines Krankenhausinformationssystems zu einem Zeitpunkt komplett ausgetauscht werden.

Um diese Heterogenität beherrschen zu können, sollte es für ihr Management möglich sein, Krankenhausinformationssysteme beschreiben, d. h. modellieren zu können. Geeignete Modelle sollten beispielsweise in der Lage sein folgende Fragen des Informationsmanagements zu beantworten:

- What tasks of the health professionals does a certain health professional workstation (HPW) ([3]) in a certain ward of 'my' hospital support?
- How many different applications do 'my' health professionals need for completing their work at that

ward?

- Which application is used for order entry and results reporting at that HPW? Who is the vendor of the respective software, what database management system is used, and what servers are engaged?
- The communication between the database on the server and the HPW happens to be erroneous due to timeout errors. Which network components could be contributing to those errors?
- What interfaces to the laboratory system are used for order and results communication? Is HL7 used as protocol on a server to server connection or is there a proprietary client based interface to the lab's intranet-server installed? How is visual integration ([4]) guaranteed in the latter case?
- To which network domain does the HPW belong and who is the responsible domain administrator?
- In order to realize another HPW an additional personal computer shall be installed at that ward. What plugs, wires and interface cards are necessary to connect the computer to the network at that site?

1.2 Ziel der Arbeit

Ziel dieser Arbeit ist es daher, die für das Management von Krankenhausinformationssystemen wesentlichen Komponenten zu identifizieren und sie zusammen mit ihren Beziehungen und Abhängigkeiten in einem Meta-Modell 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.“ ([5]). Das Meta-Modell soll es also ermöglichen, möglichst so detailliert (oder eben nicht-detailliert) beschreiben zu können, wie es für die entsprechenden Zwecke relevant ist.

Hierzu werden in Kapitel 2 zunächst Schwachstellen bereits existierender Meta-Modelle für die Modellierung von Informationssystemen analysiert und daraus Anforderungen an das Meta-Modell abgeleitet. In Kapitel 3 wird das Meta-Modell für Strukturen von Krankenhausinformationssystemen vorgestellt. Kapitel 4 wird dieses Meta-Modell später erweitern um die Beschreibung einer Prozesssicht. Kapitel 5 schließlich wird sich ebenfalls später mit der Darstellung von Qualitätskriterien im 3LGM² beschäftigen. Kapitel 4 und 5 sind daher in diesem Dokument noch nicht enthalten.

2 Meta-Modelle für die Modellierung von Informationssystemen

Meta-Modelle für die Beschreibung von Informationssystemen sind aus den Bereichen der Unternehmensmodellierung, der Geschäftsprozessmodellierung und der strategischen Planung von Informationssystemen hinreichend bekannt. Bei genauerem Hinsehen weisen sie jedoch folgende Schwächen auf:

- Meta-Modelle geben lediglich zu modellierenden Sichten und Ebenen vor, nicht jedoch die Strukturen innerhalb der Ebenen und Sichten (vgl. z. B. [6], [7]).
- Meta-Modelle konzentrieren sich primär auf die fachliche Ebene, Werkzeuge der Informationsverarbeitung werden lediglich als Ressourcen modelliert und nicht weiter strukturiert (vgl. z. B. [8], [9]).
- Meta-Modelle insbesondere zur Geschäftsprozess-Modellierung konzentrieren sich nicht primär auf Informationsflüsse, sondern auf im weitesten Sinne Materialflüsse oder auf verfügbare materielle Ressourcen, wie Betten, Behandlungsräume, Personal etc. (vgl. [x]).
- Meta-Modelle sind nicht durchgängig integriert modelliert, d. h. für die Ebenen und Sichten werden unterschiedliche Beschreibungssprachen vorgeschlagen oder angeboten (vgl. [10]).
- Es ist nicht diskutiert, ob die im Bereich von Krankenhausinformationssystemen existierenden Architekturstile mit Hilfe der Meta-Modelle abgebildet werden können.
- Qualitätskriterien sind lediglich syntaktischer Art im Sinne von Konsistenzbedingungen (diese beruhen meist auf Fehler beim Modellieren), nicht jedoch semantischer Art, so dass Aussagen über die Qualität des Informationssystems nicht getroffen werden können (z. B. hinsichtlich der Heterogenität).

Das vorliegende Meta-Modell behebt diese Schwachstellen. Es berücksichtigt alle relevanten Ebenen und Sichten für die Modellierung eines Krankenhausinformationssystems, jedoch in einem den zu unterstützenden Aufgaben des Informationsmanagements angemessenen Detaillierungsgrad. Es ist durchgängig modelliert und einfach erweiterbar.

3 Ein UML-basiertes Meta-Modell zur Beschreibung von Krankenhausinformationssystemen

3.1 Überblick

Das Meta-Modell zur Beschreibung von Krankenhausinformationssystemen knüpft an das in [10] veröffentlichte Drei-Ebenen-Modell an. Dieses bietet ein Grundgerüst von Begriffen, die zur Beschreibung statischer Strukturen von Krankenhausinformationssystemen geeignet sind.

Um die Komplexität des Meta-Modells beherrschbar zu machen, wird das Meta-Modell wie es beispielsweise auch mit dem in [12] vorgestellten generischen Architekturrahmen für Informationssysteme vorgeschlagen wird, in Ebenen und Sichten unterteilt. Wir unterscheiden eine fachliche Ebene, eine logische Werkzeugebene und eine physische Werkzeugebene. Innerhalb dieser Ebenen gibt es jeweils eine statische und eine dynamische Sicht. **Die dynamische Sicht ist derzeit nicht Gegenstand dieser Beschreibung.**

Fachliche Ebene: Die statische Sicht der fachlichen Ebene beschreibt ein Unternehmen als eine Ansammlung seiner Aufgaben. Im Rahmen der Erfüllung der Aufgaben werden Informationen über Objekte verwendet und auch geschaffen (vgl. auch [13]). Informationen über Objekte liegen in Form von Merkmalen vor. Objekte mit gleichen Merkmalsarten werden zu Objekttypen zusammengefasst. Die dynamische Sicht der fachlichen Ebene beschreibt die Reihenfolge der Erfüllung der Aufgaben in Abhängigkeit von Ereignissen. Diese Ereignisse beziehen sich auf Informationen.

Logische Werkzeugebene: Die statische Sicht der logischen Werkzeugebene beschreibt welche logischen Werkzeuge ein Unternehmen zur Erfüllung seiner Aufgaben einsetzt, wie diese miteinander kommunizieren müssen, damit der auf der fachlichen Ebene beschriebene Zugriff auf Informationen gewährleistet ist und wie Informationen über Objekte logisch als Daten gespeichert werden. Die dynamische Sicht der logischen Werkzeugebene beschreibt die Reihenfolge der Kommunikation zwischen logischen Werkzeugen in Abhängigkeit von Ereignissen. Diese Ereignisse beziehen auf den Abschluss von Aktivitäten einer Aufgabe.

Physische Werkzeugebene: Die statische Sicht der physischen Werkzeugebene beschreibt welche physischen Werkzeuge ein Unternehmen zur Erfüllung seiner Aufgaben einsetzt, wie diese miteinander vernetzt sein müssen, damit die auf der logischen Werkzeugebene beschriebenen Kommunikation gewährleistet ist. Die dynamische Sicht der physischen Werkzeugebene beschreibt die

Die zwischen den Elementen der drei Ebenen existierenden Beziehungen bezeichnen wir als Inter-Ebenen-Beziehungen.

Abb. 1 zeigt die grundlegende Struktur des Meta-Modells für die Beschreibung von Krankenhausinformationssystemen.

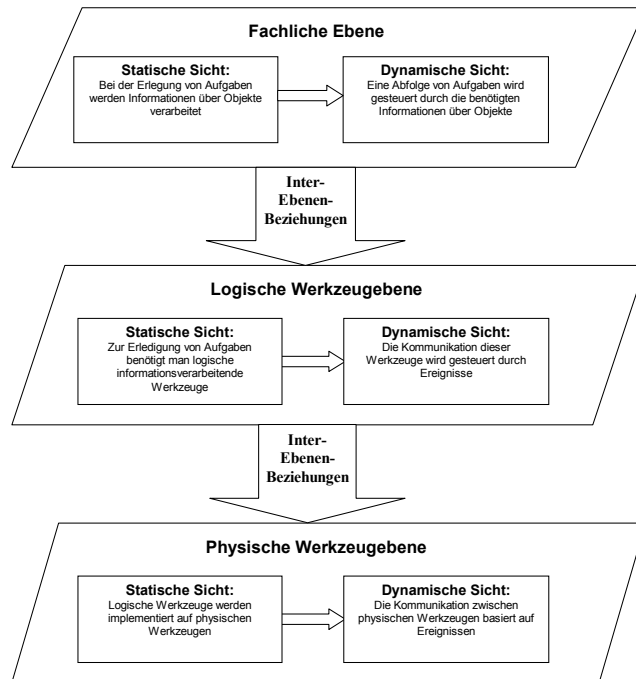


Abb. 1: Ein Meta-Modell für die Modellierung von Krankenhausinformationssystemen.

3.2 UML-Klassendiagramme

Die Struktur der drei Ebenen wird hier mit Hilfe der Klassendiagramme der Unified Modelling Language (siehe z.B. [14], [20]) dargestellt. Dabei beschränken wir uns auf die Verwendung folgender UML-Elemente (siehe auch Abb. 2):

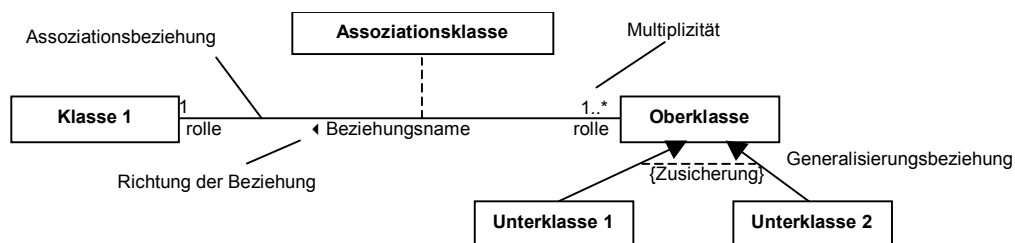


Abb. 2: Elemente der UML-Klassendiagramme.

- Eine Klasse beschreibt die Attribute und Operationen einer Menge gleichartiger Objekte.
- Klassen können in Beziehung zueinander stehen. Wir verwenden als Beziehungen die Assoziation und die Generalisierung.
- Eine Assoziation ist eine im Prinzip beliebige Beziehung zwischen Klassen und wird als Linie zwischen zwei Klassen dargestellt. Der Name einer Beziehung befindet sich in der Mitte der Linie. Ein Pfeil neben dem Beziehungsname deutet an, in welche Richtung die Beziehung zu interpretieren ist. Eine Assoziationsklasse ist eine Assoziationsbeziehung mit Attributen und ggf. Operationen.
- Eine Generalisierung ist eine Vererbungsbeziehung zwischen einer allgemeinen und einer speziellen

Klasse, die Differenzierung erfolgt anhand von Attributen. Die Generalisierung wird durch einen großen Pfeil dargestellt, wobei der Pfeil von der Unterklasse zur Oberklasse zeigt.

- Eine Zusicherung beschreibt eine Bedingung oder Integritätsregel.
- Die Multiplizität drückt aus mit wie vielen Objekten einer Klasse ein Objekt der gegenüberliegenden Klasse in Beziehung stehen darf.
- Eine Rolle beschreibt, wie ein Objekt durch das in der Assoziation gegenüberliegende Objekt gesehen wird.

3.3 Statische Sicht des Drei-Ebenen-Modells

3.3.1 Statische Sicht der Fachlichen Ebene

Die fachliche Ebene eines Krankenhausinformationssystems ergibt sich aus den Aufgaben des Krankenhauses, deren Erledigung das Krankenhausinformationssystem unterstützt, und den Objekttypen, die im Rahmen der Erledigung der Aufgaben jeweils bearbeitet bzw. interpretiert werden. Sie beschreibt damit die fachlichen Konsequenzen aus den Unternehmenszielen und abstrahiert dabei von den Werkzeugen, die zur Erledigung der Aufgaben eingesetzt werden.

Anmerkung: Das Modell der Fachlichen Ebene geht von der Vorstellung aus, dass sich die zu bearbeitenden Objekte alle in der betrachteten Welt befinden. Das macht die Vorstellung überflüssig, man müsse auf der fachlichen Ebene Objekttypen zwischen Aufgaben, bzw. Objekte zwischen Aktivitäten fließen lassen. Es wird nur noch dargestellt, im Rahmen welcher Aufgabe etwas bearbeitet wird, das im Rahmen welcher (anderen) Aufgabe möglicherweise interpretiert werden muss. Das bedeutet dann, dass darauf verzichtet wird, die Beziehungen zwischen Aufgaben zu modellieren und dies letztlich der Prozesssicht überlassen wird.

Abb. 3 zeigt die UML-Notation der fachlichen Ebene.

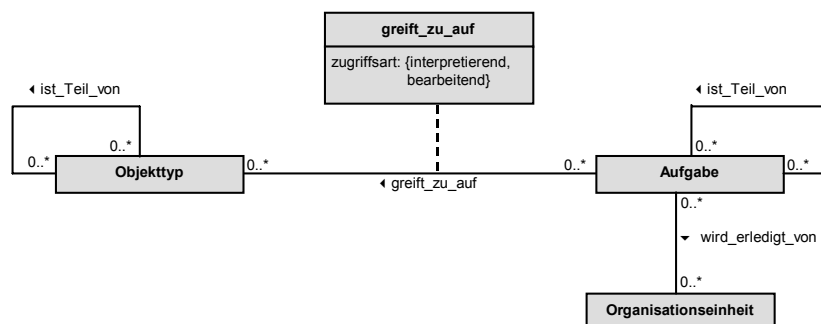


Abb. 3: Das Metamodell der fachlichen Ebene.

Objekttypen

Informationen beziehen sich auf *Objekte*. Objekte sind physische oder virtuelle Dinge eines Krankenhauses (vgl. „part of the perceivable or conceivable universe“ [15]), die durch Daten repräsentiert werden. Objekte können durch Merkmale beschrieben werden. Objekte mit denselben Merkmalsarten werden als *Objekttyp* zusammengefasst; Objekttypen werden durch ihre Merkmalsarten repräsentiert (vgl. „concept“ als „unit of thought constituted through abstraction on the basis of

properties common to a set of objects“ [15]). Durch Objekttypen kann daher beschrieben werden, welcher Art Informationen sind, die zur Bearbeitung einer Aufgabe benötigt werden oder nach Bearbeitung der Aufgabe bereitgestellt werden können. Objekttypen können verfeinert werden

Objekttypen in einem Krankenhaus sind z.B. (entsprechend [16]):

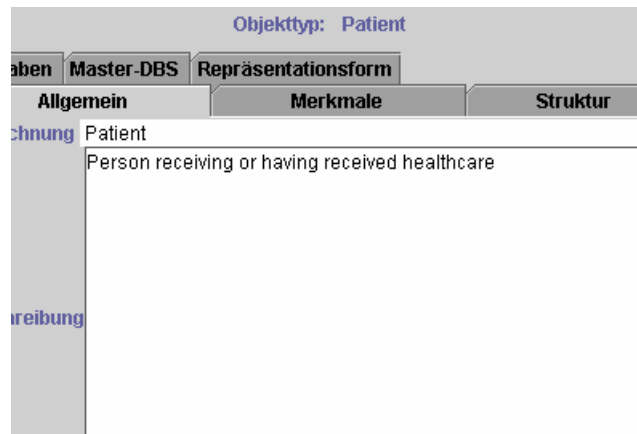


Abb. 4: Beispiel eines Objekttypen (vgl. auch [16]).

Aufgaben

Eine (Unternehmens-) Aufgabe [17] ("enterprise function" [18]) ist nach [19], 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 des Unternehmens. Eine Aufgabe hat keinen definierten Anfang und kein definiertes Ende. Aufgaben lassen sich in Teilaufgaben gliedern.

Man kann eine Aufgabe und damit auch eine Teilaufgabe als Pflicht auffassen, die sich aus den Unternehmenszielen mehr oder weniger zwingend ergibt.

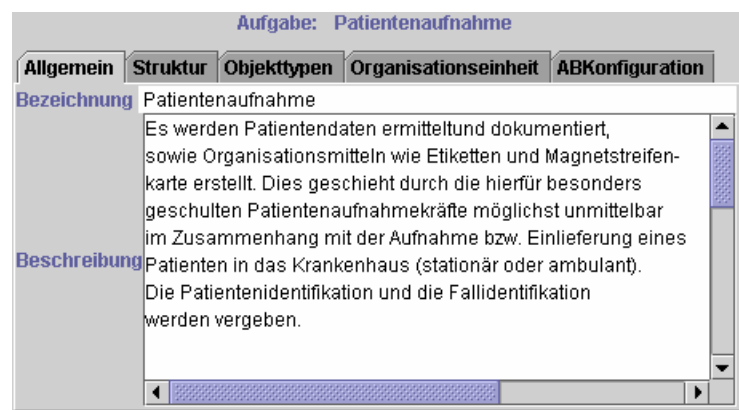


Abb. 5: Beispiel einer Aufgabe.

Das Handeln im Rahmen einer Aufgabe ist das Erfüllen der Zielvorschrift und erfolgt im Rahmen von Aktivitäten. Eine *Aktivität* der Aufgabe ist die einmalige Anwendung oder Durchführung eines *Verfahrens*, das zur Erledigung der Aufgabe zur Verfügung steht.

Aktivitäten bearbeiten Objekte, indem sie die Merkmalsausprägungen verändern. Hierzu werden ggf. Informationen benötigt, die aus den Merkmalen der bearbeiteten oder anderer Objekte gewonnen werden. Ein Objekttyp, dessen Objekte von den Aktivitäten einer Aufgabe bearbeitet werden, heißt '*zu bearbeitender Objekttyp einer Aufgabe*'. Ein Objekttyp, von dessen Objekten die Aktivitäten einer Aufgabe Informationen gewinnen, heißt '*zu interpretierender Objekttyp einer Aufgabe*'.

Im Rahmen dieser Arbeit wird zunächst darauf verzichtet, über dieses statische Modell hinaus eine Reihenfolge der Bearbeitung von Objekten durch Aktivitäten der Aufgaben zu modellieren. Dies bleibt der Erweiterung zu einem dynamischen Modell vorbehalten. Für Planung und Überwachung im strategischen Informationsmanagement ist es aber wichtig, dass für jeden Objekttyp die Aufgaben abgeleitet werden können, bei deren Erledigung der Objekttyp bearbeitet oder interpretiert wird und umgekehrt.

Organisationseinheit

Aufgaben werden in bestimmten Organisationseinheiten erledigt. Dies ist später insbesondere bei der Zuordnung von Anwendungsbausteinen bedeutsam.

Auf der fachlichen Ebene werden Anforderungen an die logische und physische Werkzeugebene formuliert. Aus diesem Grund können allein für diese Ebene keine semantischen Qualitätskriterien angegeben werden. Allerdings können auf der fachlichen Ebene Referenzmodelle beschrieben werden, die dann für die Modellierung eines konkreten Krankenhausinformationssystems lediglich noch modifiziert werden müssen. Referenzmodelle der fachlichen Ebene sind beispielsweise der Anforderungskatalog für die Informationsverarbeitung im Krankenhaus, das Good Electronic Health Record, das Reference Information Model des Kommunikationsstandards HL7, HISA oder die Europäische Electronic Healthcare Record Architecture (EHCRA).

Abb. 6 zeigt ein Beispiel einer fachlichen Ebene, das als eine Ausprägung des in Abb. 3 dargestellten Meta-Modells zu betrachten ist. Rechtecke repräsentieren dabei Aufgaben, Ovale stehen für Objekttypen. Ein Pfeil von einem Objekttyp zu einer Aufgabe markiert einen interpretierenden Zugriff, ein Pfeil von einer Aufgabe zu einem Objekttyp einen bearbeitenden Zugriff. Verfeinerungen sind dargestellt als Rechtecke, die in anderen Rechtecken enthalten sind. Typische Aufgaben eines Krankenhauses sind hier die KRANKENHAUSVERWALTUNG, die PATIENTENBEHANDLUNG oder die VERWALTUNG VON PATIENTENAKTEN. Die Aufgabe PATIENTENBEHANDLUNG ist verfeinert in weitere Aufgaben. Typische Objekttypen sind PATIENT, LEISTUNGSANFORDERUNG oder FALL.

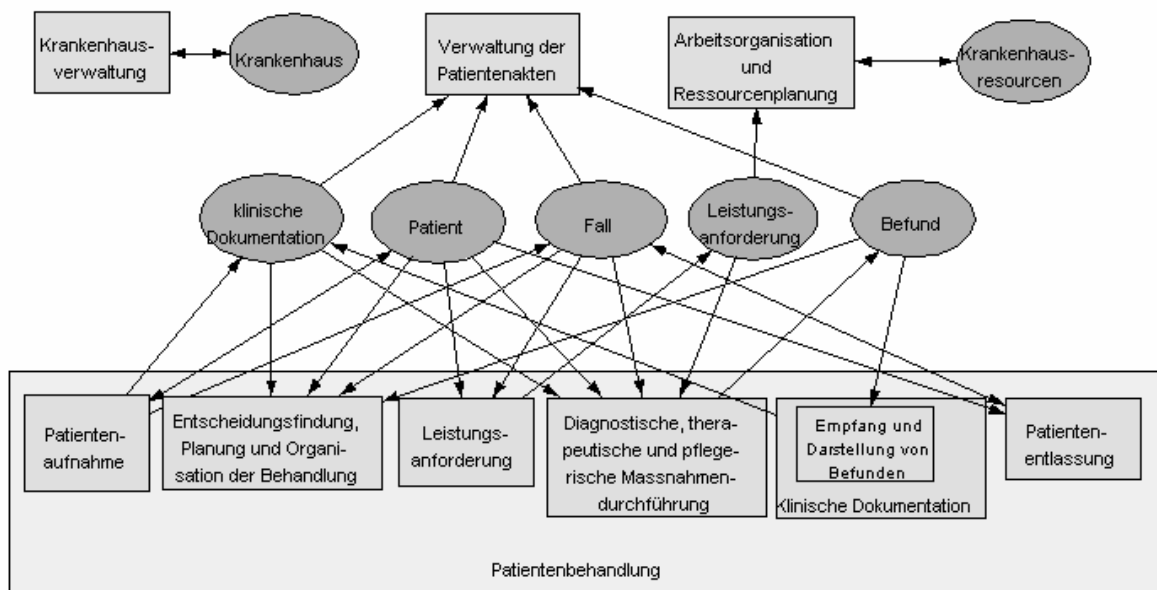


Abb. 6: Beispiel einer fachlichen Ebene.

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

3.3.2 Statische Sicht der Logischen Werkzeugebene

Auf der logischen Werkzeugebene stehen die Anwendungsbausteine im Mittelpunkt. Anwendungsbausteine unterstützen Aufgaben und verarbeiten, speichern und transportieren Daten, die Objekte bestimmter Objekttypen repräsentieren. Besonders wichtiger Aspekt dieser Ebene ist, wo Objekttypen logisch gespeichert werden und wie Anwendungsbausteine kommunizieren müssen, um zu gewährleisten, dass bei der Durchführung der Aufgaben der benötigte Zugriff auf Informationen gesichert ist.

Anwendungsbausteine (engl. application component) sind Werkzeuge der Informationsverarbeitung, die von den Nutzern des Krankenhausinformationssystems unmittelbar oder mittelbar für die Erledigung der Aufgaben des Krankenhauses verwendet werden. Ein Anwendungsbaustein kann die Erledigung einer oder mehrerer Aufgaben ganz oder auch jeweils nur teilweise unterstützen. Im letzteren Fall unterstützen möglicherweise mehrere Anwendungsbausteine gemeinsam die Erledigung einer bestimmten Aufgabe. Anwendungsbausteine gehören zu den logischen Werkzeugen. Zur Realisierung eines logischen Werkzeugs der Informationsverarbeitung sind physische Werkzeuge erforderlich (siehe Kapitel 3.3). Im Gegensatz zu den logischen Werkzeugen, sind diese – wie z.B. ein Arbeitsplatzrechner – zwar physisch greifbar – aber unmittelbar, d.h. ohne installierten Anwendungsbaustein, für den Nutzer als Werkzeug nicht nutzbar.

Ein Anwendungsbaustein wird realisiert durch

- die Installation, Adaptation und Inbetriebnahme von *Softwareprodukten* auf Rechnersystemen oder
- die Verabschiedung von *Organisationsplänen*, die mit Hilfe eines Systems physischer Werkzeuge, die keine Rechnersysteme sind (konventionelle Werkzeuge) durchgeführt werden.

Falls für ein Anwendungsbaustein ausschließlich der erste Fall gilt, wird der Anwendungsbaustein als *rechnerbasiert*, sonst als *nicht rechnerbasiert* bezeichnet.

Anwendungsbausteine können durch folgende Begriffe näher beschrieben werden:

- Ein rechnerbasierter Anwendungsbaustein wird gesteuert durch ein *Anwendungsprogramm*, das auf einem *Softwareprodukt* basiert. Anders herum formuliert ist ein Anwendungsprogramm ein adaptiertes Softwareprodukt. Ein nicht rechnerbasierter Anwendungsbaustein wird von einem *Organisationsplan* gesteuert.
- Ein *Softwareprodukt* ist ein abgeschlossenes, erworbenes oder eigenentwickeltes Programm oder Programmpaket, das auf Rechnersystemen (siehe physische Werkzeugebene) installiert werden kann.
- 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.
- 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 das Anwendungsprogramm des Anwendungsbausteins identisch mit dem Softwareprodukt.
- Ein Anwendungsbaustein kann über ein eigenes Gedächtnis verfügen, das die Speicherung von Daten erlaubt. Das Gedächtnis wird durch ein *Datenbanksystem* beschrieben, wenn der Anwendungsbaustein rechnerbasiert ist, und durch eine *Dokumentensammlung*, wenn der Anwendungsbaustein nicht rechnerbasiert ist.
- Ein Datenbanksystem speichert Datensätze bestimmter *Datensatztypen*. Sie wird gesteuert durch eine *Datenbankverwaltungssystem*.
- Eine Dokumentensammlung speichert Dokumente bestimmter *Dokumententypen*.
- Anwendungsbausteine können über *Benutzungsschnittstellen* Daten mit Benutzern austauschen.
- Anwendungsbausteine können untereinander über *Bausteinschnittstellen* Nachrichten austauschen. Eine Bausteinschnittstelle wird gegebenenfalls beschrieben durch den unterstützten *Kommunikationsstandard* und die *Ereignis-Nachrichtentypen* bzw. *Ereignis-Dokumententypen*, die versendet oder empfangen werden können.
- Ist ein Anwendungsbaustein dazu in der Lage, über Bausteinschnittstellen unmittelbar einem anderen Anwendungsbaustein Nachrichten zu senden, besteht zwischen diesen Anwendungsbausteinen eine *Kommunikationsbeziehung*. 'Unmittelbar' bedeutet dabei, dass für den Transport der Nachrichten kein dritter Anwendungsbaustein eingesetzt wird. Innerhalb einer Kommunikationsbeziehung befindet sich ein Bausteinschnittstelle in der Rolle des *Senders* und die andere Bausteinschnittstelle in der Rolle des *Empfängers*. Hierdurch wird auch eine Richtung der Kommunikation vorgegeben. Für eine Kommunikationsbeziehung wird auch angegeben, welche Ereignis-Nachrichtentypen bzw. Ereignis-Dokumententypen tatsächlich ausgetauscht werden.
- Datensatztyp, Dokumenttyp und Nachrichtentyp sind *Repräsentationsformen* für Objekttypen und sind daher insbesondere für bestimmte Inter-Ebenen-Beziehungen von Bedeutung (siehe Abschnitt Inter-Ebenen-Beziehungen).
- Eine Menge von Anwendungsbausteinen (*Anwendungsbausteinkonfiguration*) unterstützt eine oder mehrere Aufgaben (ausführliche Beschreibung von Anwendungsbaustein-Konfigurationen siehe im Abschnitt Inter-Ebenen-Beziehungen). Hinweis: Eine Anwendungsbausteinkonfiguration kann auch aus

lediglich einem Anwendungsbaustein bestehen. In diesem Fall unterstützt ein Anwendungsbaustein eine Aufgabe komplett.

***Anmerkung:** Der Zusammenhang Anwendungsbaustein – Anwendungsprogramm – Softwareprodukt soll an folgendem Beispiel verdeutlicht werden: Wenn ein Krankenhaus z. B. ein Archivverwaltungssystem für Patientenaktenarchive, nennen wir es ArchiMed, von der Firma Graecia GmbH kauft, ist das Archivverwaltungssystem zunächst 'leer'. Die Disketten oder CD-ROMs, die an andere 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 spezifischer und nutzbarer Anwendungsbaustein, der etwa mit ARCHIV bezeichnet werden kann.*

***Anmerkung:** Dasselbe Softwareprodukt kann mehrfach auf demselben oder unterschiedlichen Rechnern installiert werden. Dadurch entstehen unterschiedliche Anwendungsbausteine. Im Falle mehrfacher Installation auf verschiedenen Rechnern und jeweils identischer Adaptation lassen sich die Anwendungsbausteine (ausschließlich) durch die physischen Rechnersysteme unterscheiden, auf denen sie installiert sind. Ist ein Softwareprodukt mehrfach mit identischer Adaptation auf dem selbem Rechner installiert, ist eine Unterscheidung nicht so ohne weiteres möglich.*

Abb. 7 zeigt das Metamodell der logischen Werkzeugebene eines Krankenhausinformationssystems. Die gepunkteten Klassen und Assoziationsbeziehungen zeigen den Zusammenhang zwischen Fachlicher Ebene und Logischer Werkzeugebene (vgl. Abschnitt Inter-Ebenen-Beziehungen).

können. Solche Digital-Analog-Schnittstellen werden z.B. durch Drucker und Belegleser bzw. der entsprechenden Software realisiert

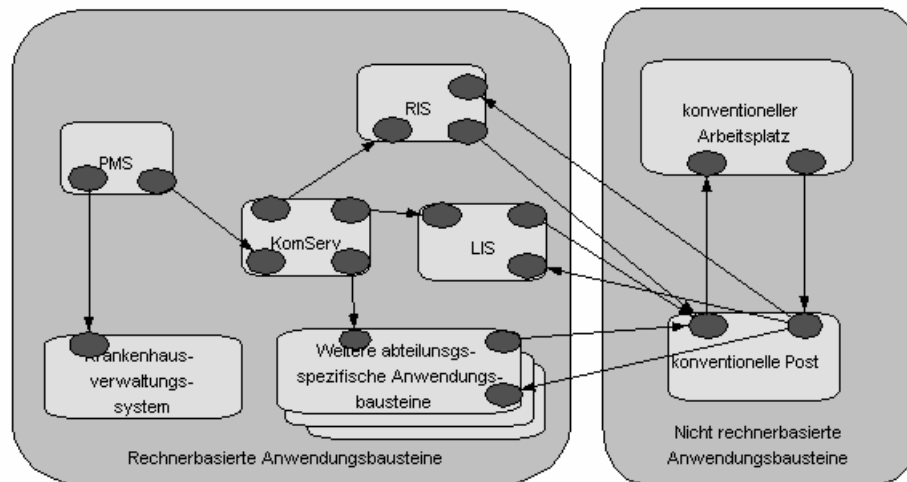


Abb. 8: Beispiel einer Logischen Werkzeugebene

3.3.3 Statische Sicht der Physischen Werkzeugebene

Die physische Werkzeugebene besteht aus einer Menge von physischen Datenverarbeitungsbausteinen. Ein physischer Datenverarbeitungsbaustein eines Krankenhausinformationssystems ist entweder

- ein System von Personen und konventionellen Werkzeugen (wie z.B. Archivregale, Posteingangs- und Postausgangskörbe) der Informationsverarbeitung (nicht-rechnerbasierter physischer Datenverarbeitungsbaustein) oder
- ein rechnerbasierter physischer Datenverarbeitungsbaustein, z.B. Rechnersysteme (PCs, Server), Monitore, Drucker, Switches, Router, Subnetze, Netzwerkkarten, Netzanschlusspunkte).

Die physischen Datenverarbeitungsbausteine können über Datenübertragungsverbindungen miteinander kommunizieren. Zwischen rechnerbasierten physischen Datenverarbeitungsbausteinen sind dies typischerweise Datenkabel. Betrachtet man allerdings Funknetze, sind diese Datenübertragungsverbindungen lediglich virtueller Art (also nicht greifbar) und lediglich durch die beiden Knoten zwischen denen sie besteht definiert. Ähnliches gilt auch für Datenübertragungsverbindungen zwischen nicht rechnerbasierten Datenverarbeitungsbausteinen.

Die Konstellation dieser Datenübertragungsverbindungen führt zu physischen Netzwerken, die auf Netzprotokollen basieren. Subnetze können definiert werden als Projektion auf das gesamte Netzwerk. Es ist zu beachten, dass auf der physischen Werkzeugebene sowohl 'echte' physische Netzwerke beschrieben werden können, die lediglich die bestehenden Datenverarbeitungsbausteine und ihre Datenübertragungsverbindungen darstellen, als auch 'logische Netzwerke', die berücksichtigen, dass bestimmte Datenverarbeitungsbausteine auch zu bestimmten Zonen gehören, in denen bestimmte Zugriffsrechte vergeben sind.

Die physische Werkzeugebene stellt also die physischen Werkzeuge bereit, die für den Betrieb von Anwendungsbausteinen erforderlich sind. Physische Datenverarbeitungsbausteine lassen sich ggf. wiederum in physische Datenverarbeitungsbausteine gliedern.

Insbesondere Rechnersysteme können z. B. durch folgende Attribute der Klasse 'Physische Datenverarbeitungsbausteine näher beschrieben werden (diese sind im UML-Klassendiagramm nicht dargestellt):

- Inventarnummer
- Seriennummer
- Typ
- Betriebssystem
- Festplattenspeicher
- Arbeitsspeicher
- Prozessor

Abb. 9 zeigt die UML-Notation der physischen Werkzeugebene.

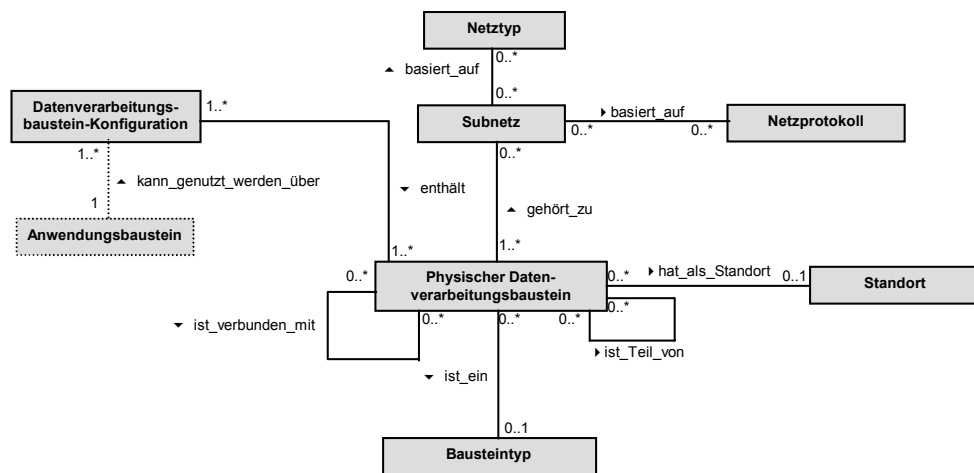


Abb. 9: Klassendiagramm 'Physische Werkzeugebene'. Gepunktete Linien und Symbole repräsentieren Inter-Ebenen-Beziehungen.

Anmerkung: Das Meta-Modell der physischen Werkzeugebene ist wesentlich offener gestaltet, d. h. es steckt inhaltlich wesentlich weniger drin als im Meta-Modell der logischen Werkzeugebene. Während dort z.B. Schnittstellen explizit modelliert sind, können Schnittstellen (z. B. Netzwerkkarten) auf der physischen Werkzeugebene als physischer Datenverarbeitungsbaustein modelliert werden, der wiederum Teil eines Rechnersystems ist.

Abb. 10 zeigt ein Beispiel eines Physischen Datenverarbeitungsbausteins

PhysischerDVBaustein: LIS Server				
Allgemein	Standort	Bausteintyp	Subnetz	Techn. Merkmale
Betriebssystem				Windows NT
Seriennummer				4356785549
Inventarnummer				03-253489568
Festplattengröße				5 GB
Arbeitsspeicher				256 MB
Prozessor				Intel III

Abb. 10: Beispiel für einen physischen Datenverarbeitungsbaustein.

Anmerkung: Bei konventionellen Datenverarbeitungsbausteinen kann man unter dem Betriebssystem die intellektuellen Fähigkeiten der beteiligten Personen verstehen. So sind beispielsweise bei einem Krankenaktenarchiv neben den Räumen und Regalanlagen Menschen notwendig, die grundlegende Kulturtechniken wie Ordnung halten, freundlicher Umgang mit Kunden und Mitarbeitern etc. beherrschen. Nur auf dieser Grundlage kann mit Hilfe eines Organisationsplans und den Anweisungen einer Leitung ein funktionierendes Archiv als nicht-rechnerbasierter Anwendungsbaustein realisiert werden.

Abb. 11 zeigt ein Beispiel einer physischen Werkzeugebene. In diesem Beispiel gibt es je einen Server für die verschiedenen Anwendungsbausteine der Funktionsabteilungen (einen LIS Server, einen RIS Server und die Server für die nicht weiter spezifizierten Anwendungsbaustein und einen zentralen Server, auf dem das PVS und das Krankenhausverwaltungssystem installiert sind. Jeder Server ist verbunden mit einigen PCs. Die schwarzen Punkte repräsentieren Netzwerkkomponenten, über die die Server miteinander verbunden sind und damit das Gesamtnetzwerk bilden. Die Datenübertragung vom und zum paper-basierten Teil des KIS wird hier lediglich am Beispiel der LIS dargestellt. Eine Laboranforderung (als Dokument) geht über den Postausgangskorb des klinischen Arbeitsplatzes zum Posteingangskorb des Labors und wird von einem Formularleser eingelesen. Das Ergebnis (also der Befund) wird ausgedruckt und geht wiederum über den Ausgangskorb des Labors zum Eingangskorb des klinischen Arbeitsplatzes. In diesem Beispiel sind keine Subnetze spezifiziert und es werden keine Informationen über den Netztyp, das Netzprotokoll oder den Standort dargestellt.

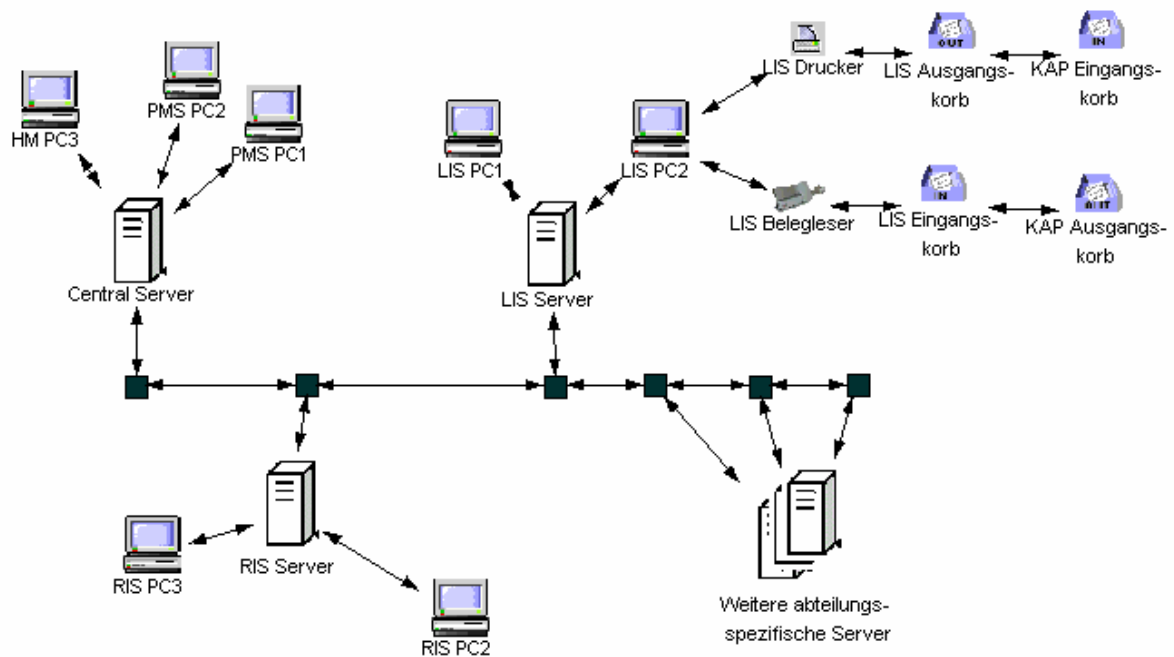


Abb. 11 Beispiel einer physischen Werkzeugebene

3.3.4 Interebenen-Beziehungen zwischen statischen Komponenten

Wie sich schon an der einen oder anderen Stelle gezeigt hat, existieren die drei Ebenen nicht unabhängig voneinander. Die vielfältigen Beziehungen zwischen den Ebenen bezeichnen wir als Interebenen-Beziehungen. In den Klassendiagrammen der Ebenen sind sie als gepunktete Klassen und Assoziationsbeziehungen sichtbar. Sowohl zwischen der Fachlichen Ebene und der Logischen Werkzeugebene als auch zwischen der Logischen und der Physischen Werkzeugebene können wir Zusammenhänge feststellen.

Beziehungen zwischen fachlicher Ebene und logischer Werkzeugebene

Aufgabe - Anwendungsbaustein

Innerhalb eines Krankenhauses kann eine Aufgabe unterstützt werden durch

- mehrere Anwendungsbausteine gemeinsam,
- durch ein einzelnen Anwendungsbaustein,
- durch Kombinationen davon.

Dieser Sachverhalt drückt sich in der Anwendungsbausteinkonfiguration aus. Die Anwendungsbausteinkonfigurationen lassen sich wie folgt ermitteln:

1. Welche Anwendungsbausteine sind gemeinsam notwendig, um die Aufgabe unterstützen zu können?

Eine Anwendungsbausteinkonfiguration enthält also alle die Anwendungsbausteine, die zur Unterstützung der Aufgabe unmittelbar erforderlich sind. Unmittelbar bezieht sich dabei auf die reine

Funktionalität, nicht auf die Lieferung von Daten. Wenn man einen Anwendungsbaustein aus einer Konfiguration entfernt, wird die Aufgabe nicht mehr unterstützt.

2. Welche möglichen Alternativen gibt es zur Unterstützung der Aufgabe?

Eine Aufgabe kann durch mehrere Anwendungsbausteinkonfigurationen unterstützt werden. Wenn man eine Anwendungsbausteinkonfiguration wegnimmt, kann die Aufgabe trotzdem noch durch eine der verbleibenden erledigt werden, ggf. unter Einschränkung der Qualität. Wenn dies nicht so ist, kann dies ein Indiz dafür sein, dass auf der Fachlichen Ebene nicht adäquat (z.B. nicht fein genug) modelliert wurde.

Abb. 12 zeigt ein Beispiel einer Anwendungsbausteinkonfiguration.

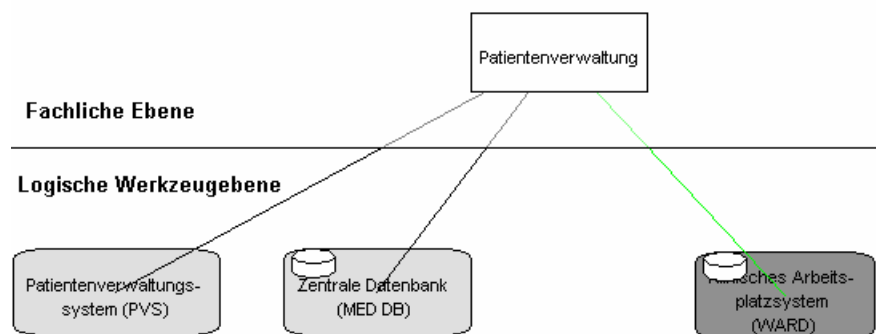


Abb. 12: Anwendungsbausteinkonfigurationen (Beispiel)

Anwendungsbaustein-Konfigurationen können Hinweise auf Redundanzen auf der logischen Werkzeugebene geben, aber auch Hinweise auf Schwächen im Modell der fachlichen Ebene. Dies wird in den folgenden Anmerkungen nochmals diskutiert. Abb. 12 zeigt ein Beispiel für Anwendungsbausteinkonfigurationen. Die Aufgabe PATIENTENAUFNAHME wird unterstützt durch zwei Anwendungsbausteinkonfigurationen. Die hellgraue besteht aus dem PVS – das allerdings keine eigene Datenbank enthält - und MED DB das die zentrale Datenbank enthält, in der die Aufnahmedaten gespeichert werden. Die Aufgabe PATIENTENAUFNAHME kann also nur ausgeführt werden, wenn beide Anwendungsbausteine, PVS und MED DB verfügbar sind. Die dunkelgraue Anwendungsbausteinkonfiguration besteht aus genau einem Anwendungsbaustein, der ausreicht, um die Aufgabe PATIENTENAUFNAHME zu unterstützen.

Anmerkungen zu dem Konzept der Anwendungsbausteinkonfigurationen: In der folgenden Abb. 13 sieht es zunächst so aus, als könnte das AB Laboratory das AB Radiology ersetzen. Das dies nicht so ist, ist offensichtlich. Es liegt also ein Modellierungsproblem vor. Um dies zu beheben, bieten sich zwei Möglichkeiten an:

- Wir verfeinern auf der fachlichen Ebene die Aufgabe Diagnostik in Labordiagnostik, Radiologiediagnostik und Pathologische Diagnostik. Jede dieser Aufgabe erhält dann den entsprechenden Anwendungsbaustein zugeordnet.
- Wir vergrößern auf der logischen Werkzeugebene und fassen die Anwendungsbausteine für die Diagnostik zu einem Anwendungsbaustein zusammen.
- Eine dritte Möglichkeit ergibt sich daraus, dass man - wie in der aktuellen Version des Metamodells getan - Organisationseinheiten einführt. Berücksichtigt man diese später bei der Zuordnung der

Anwendungsbausteinkonfigurationen – so wie im Metamodell vorgeschlagen – so kann man diese Problematik umgehen.

Hieraus kann man sich leicht klarmachen, dass die Verwendung von Anwendungsbausteinkonfigurationen im Prinzip äquivalent ist zur Vergrößerung und Verfeinerung von Anwendungsbausteinen. Man will mit der Verwendung von Anwendungsbausteinkonfigurationen allerdings umgehen, 'künstliche' Anwendungsbausteine einführen zu müssen und möchte diesen Sachverhalt lieber explizit als anderes Konzept modellieren.

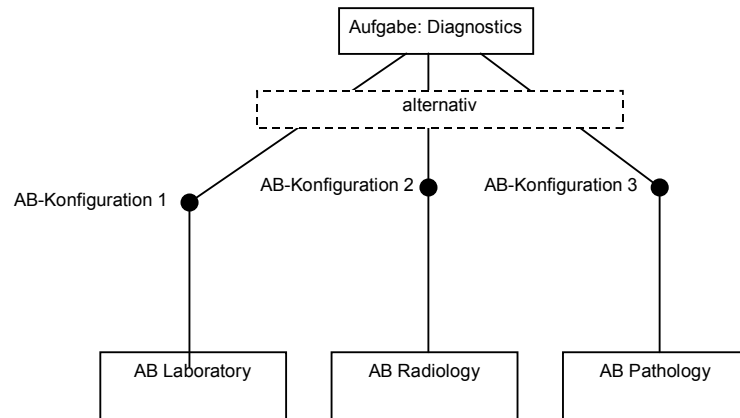


Abb. 13: Anwendungsbausteinkonfigurationen mit Hinweis auf nicht adäquate Modellierung

Hat man andererseits eine Aufgabe, die tatsächlich alternativ durch zwei Anwendungsbausteine unterstützt wird (siehe Abb. 14), wird man folgende Überlegungen durchführen müssen:

- Handelt es sich tatsächlich um dieselbe Aufgabe?
- Sind die Alternativen tatsächlich notwendig (beispielsweise aufgrund äußerer Gegebenheiten, oder weil einer der Anwendungsbausteine noch eine zusätzliche Aufgabe unterstützt)?

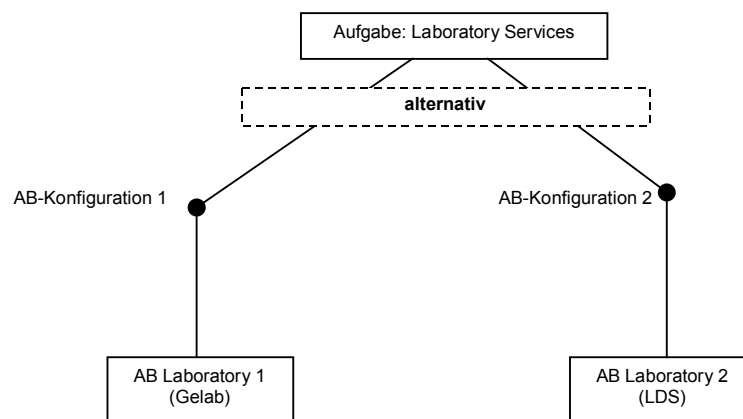


Abb. 14: Anwendungsbausteinkonfigurationen Alternativen

Im Beispiel von Abb. 14 sehen wir die Aufgabe Laboratory Services durch zwei Anwendungsbausteine alternativ unterstützt, die auch noch auf zwei verschiedenen Softwareprodukten basieren (die Softwareprodukte sind in Klammern angegeben). Hier könnte es aber durchaus sein, dass einer der Anwendungsbausteine zusätzlich noch in der Lehre eingesetzt wird, also eine weitere Aufgabe unterstützt. Allerdings müssen wir uns dann fragen, ob nicht wenigstens Anwendungsbausteine basierend auf dem gleichen Softwareprodukt verwendet werden sollen.

Ein weiteres Beispiel ist die Aufgabe Terminplanung. Zwar gibt es z.B. sowohl im Rahmen der OP-Planung eine Terminplanung als auch in einer Radiologischen Abteilung. Die entsprechenden Anwendungsbausteine können einander aber nicht ersetzen. Offenbar muss daher die Aufgabe Terminplanung verfeinert werden, oder als Teilaufgaben der Aufgaben OP-Planung und Radiologische Dienste verstanden werden. Entfernt man sich aber vom abteilungsorientierten Denken hin zum diensteorientierten Denken, so kann es tatsächlich einen generischen Dienst Terminplanung geben, der je nachdem wo er verwendet wird, unterschiedlich arbeitet.

There are some other important inter-layer-relationships between concepts of the domain and the logical tool layer which we will shortly describe in the following:

Objektyp – Repräsentationsform

Objektypen können logisch repräsentiert werden durch Datensatztypen, durch Dokumententypen und durch Nachrichtentypen. Datensatztypen beschreiben welche Informationen repräsentiert als Objektypen in einem Datenbanksystem gespeichert werden. Dokumententypen beschreiben, welche Informationen repräsentiert als Objektypen in einer Dokumentensammlung gespeichert, und welche Informationen über eine Kommunikationsverbindung transportiert werden. Ein Nachrichtentyp beschreibt welche Informationen bei nachrichtenbasierter Kommunikation transportiert werden.

Aufgabe - Softwareprodukt

Ein Softwareprodukt unterstützt die Durchführung bestimmter Aufgaben. Entsprechend einer bestimmten Parametrierung kann es aber sein, dass ein auf einem bestimmten Softwareprodukt basierender Anwendungsbaustein nicht alle diese Aufgaben unterstützt. Diese Beziehung beschreibt also nicht, welche Aufgaben ein Anwendungsbaustein basierend auf einem Softwareprodukt tatsächlich unterstützt, sondern lediglich das Potential, das ein bestimmtes Softwareprodukt bietet.

Aufgabe – Ereignis

Die Beendigung einer Aktivität einer Aufgabe kann ein Ereignis eines bestimmten Ereignistyps auslösen, das wiederum das Versenden einer Nachricht eines bestimmten Nachrichtentyps steuert. Diese Beziehung ermöglicht es insbesondere nachrichten-basierte Kommunikation zu beschreiben.

Beziehungen zwischen logischer Werkzeugebene und physischer Werkzeugebene

Anwendungsbausteine - Physische Datenverarbeitungsbausteine

Ein rechnerbasierter Anwendungsbaustein kann installiert sein auf:

- mehrere physischen Datenverarbeitungsbausteinen gemeinsam,
- auf einem einzelnen physischen Datenverarbeitungsbausteinen,
- auf Kombinationen davon.

Dieser Sachverhalt drückt sich in der Datenverarbeitungsbaustein-Konfiguration aus. Die Datenverarbeitungsbaustein-Konfigurationen lassen sich wie folgt ermitteln:

1. *Welche Datenverarbeitungsbausteine sind gemeinsam notwendig, um den Anwendungsbaustein installieren zu können?*

Eine Datenverarbeitungsbaustein-Konfiguration enthält also alle die Datenverarbeitungsbausteine, die zur Installation des Anwendungsbausteins erforderlich sind. Wenn man einen

Datenverarbeitungsbaustein aus einer Konfiguration entfernt, ist der Anwendungsbaustein nicht mehr nutzbar.

2. Welche möglichen Alternativen gibt es zur Nutzung eines Anwendungsbausteins?

Ein Anwendungsbaustein kann durch mehrere Datenverarbeitungsbaustein-Konfigurationen nutzbar gemacht werden. Wenn man eine Datenverarbeitungsbaustein-Konfiguration wegnimmt, kann der Anwendungsbaustein trotzdem noch durch eine der verbleibenden genutzt werden, ggf. unter Einschränkung der Qualität. Dies ist dann nicht der Fall, wenn ein Datenverarbeitungsbaustein ausfällt, der in beiden Konfigurationen enthalten ist.

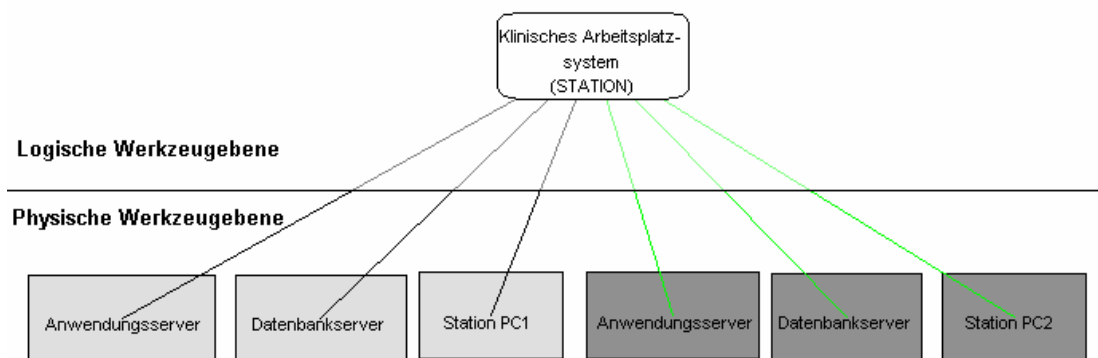


Abb. 15: Datenverarbeitungsbaustein-Konfigurationen (Beispiel)

In unserem Beispiel in Abb. 15 gibt es für den Anwendungsbaustein WARD zwei alternative Konfigurationen. Fällt beispielsweise der PDVB WARD PC1 aus, so kann zwar mit Konfiguration 1 nicht mehr gearbeitet werden, aber mit Konfiguration 2. Fällt dagegen der PDVB DB Server aus, so ist keine Konfiguration mehr nutzbar.

4 Literatur

4.1 Allgemeine Literatur

- [1] BERG M (1999): Patient care information systems and health care work: A sociotechnical approach, *Int. J. Med. Inf.* **55** (1999) 87–101.
- [2] BRODI ML, STONEBRAKER M (1995): *Migrating Legacy Systems: Gateways, Interfaces & the incremental approach*. Morgan Kaufmann.
- [3] WINTER A, LAGEMANN A, BUDIG B, GROTHE W, HAUX R, HERR S, PILZ J, SAWINSKI R, SCHMÜCKER P (1996): Health professional workstations and their integration in a hospital information system: the pragmatic approach MEDIAS. *Computer Methods and Programs in Biomedicine* **51(3)**, 193-209.
- [4] REYNOLDS M, WEJERFELD I (2000). *Health Informatics – Short Strategic Study – Health Information Infrastructure*. CEN/TC 251/SSS-HII INR, 2000.
- [5] SCHÜTTE R; ROTTHOWE T (1998): The Guidelines of Modeling as an approach to enhance the quality of information models. In: LING, TOK WANG; RAM, SUDHA; LEE, MONG LI (1998): *Conceptual Modeling - ER '98*. 17th International Conference on Conceptual Modeling, Singapore, November 16-19, 1998, Proceedings . S. 240-254.
- [6] ZACHMAN JA (1999): A framework for information systems architecture. *IBM systems journal* 1999: 38 (2&3), pp. 454-470 (Reprint).
- [7] Gartner Group (1998): *Three Documents for Healthcare IT Planning*, Gartner Group's Healthcare Executive and Management Strategies, Research Note KA-03-5074, 1998.
- [8] OBERWEIS A, SCHERRER G, STUCKY W (1994): INCOME/STAR: Methodology And Tools for the Development of Distributed Information Systems. *Information Systems* 1994. **19(8)**, 643-660.
- [9] FERSTL OK, SINZ EJ (1997): Modeling of Business Systems Using the Semantic Object Model (SOM) - A Methodological Framework -. In: BERNUS P, MERTINS K, SCHMIDT (eds.). *Handbook on Architectures of Information Systems*. International Handbook on Information Systems. Springer, 1997
- [10] SCHEER AW (1998): ARIS - House of Business Engineering. In: MOLINA A.; KUSIAKA A.; SANCHEZ J. (Hrsg.): *Handbook of Life Cycle Engineering - Concepts, models and technologies*. Kluwer Academic Publishers, Dordrecht et al, S. 332 - 357.
- [11] WINTER A, HAUX R (1995): A Three-Level Graph-Based Model for the Management of Hospital Information Systems. *Meth. Inform. Med.* **34** (4), 378-396.

- [12] SINZ EJ (1996): *Ein Architekturrahmen für die Modellierung betrieblicher Informationssysteme*. Bamberger Beiträge zur Wirtschaftsinformatik, Nr. 32.
- [13] PORTER M. *Competitive Advantage*. The Free Press, 1985.
- [14] OMG. Unified Modelling Language Specification. Version 1.3, 1999. (www.omg.org)
- [15] ISO 1087
- [16] CEN/TC251 (1997). *Healthcare Information System Architecture Part 1 (HISA) Healthcare Middleware Layer*. CEN European Committee for Standardisation. European Prestandard. prENV 12967-1:1997 Final Draft 2.
- [17] FERSTL OK, SINZ EJ (1994). *Grundlagen der Wirtschaftsinformatik*. R. Oldenbourg, München.
- [18] MARTIN J. *Information Engineering, Book II: Planning and Analysis*. Englewood cliffs: Prentice Hall, 1990.
- [19] WÖHE G. (1978). *Einführung in die Allgemeine Betriebswirtschaftslehre*. Franz Vahlen.
- [20] OESTEREICH B (1998): *Objektorientierte Softwareentwicklung*. München: Oldenbourg.

4.2 Literatur zum 3LGM²

- [1] BRIGL B., GERBER S., WENDT T., WINTER A. (2001). Ableitung von Prozessen aus statischen Modellen für Krankenhausinformationssystemen (Abstract). *Informatik, Biometrie und Epidemiologie in Medizin und Biologie* **32**(2-3), 124-125.
- [2] BRIGL B, WENDT T, WINTER A (2003): Modeling interdependencies between business and communication processes in hospitals. In: BAUD R, FIESCHI M, LE BEUX P AND RUCH P (2003): *The New Navigators: from Professionals to Patients*. Proceedings of MIE2003. Studies in Health Technology and Informatics, Volume 95. Amsterdam: IOS Press.
- [3] GRÄBER S, AMMENWERTH E, BRIGL B, DUJAT C, GROÙE A, HÄBER A, C.JOSTES, WINTER A (2002). *Rahmenkonzepte für das Informationsmanagement in Krankenhäusern: Ein Leitfaden*. Deutsche Gesellschaft für Medizinische Informatik, Statistik und Epidemiologie, Arbeitsgruppe Methoden und Werkzeuge für das Management von Krankenhausinformationssystemen.
- [4] HASSELBRING W., NIEMANN H., WENDT T., WINTER A., MEIERHOFER M. (2002). Wie eng kann/sollte die Kopplung von KIS Komponenten sein?. *Informatik, Biometrie und Epidemiologie in Medizin und Biologie* **33**(2-3), 155.
- [5] HAUX R., AMMENWERTH E., BRIGL B., WINTER A. (2001). Information Processing in Hospitals: State, Requirements, and the Need for a Modern Theory of Information System Architectures. In PATEL V.L., ROGERS R., HAUX R. (Hrsg.) (2001). *MEDINFO 2001*. IOS, Amsterdam. S. 1517.

- [6] HAUX R., WINTER A., AMMENWERTH E., BRIGL B. (2002). *Strategic Information Management in Hospitals*, New York: Springer: In Vorbereitung
- [7] HÜBNER-BLODER, BRIGL B, WENDT T, WINTER A. (2003). 3LGM²-basierte Modellierung von Krankenhausinformationssystemen am Beispiel der Tiroler Landeskrankenanstalten GmbH (TILAK). submitted for: GMDS Jahrestagung 2003.
- [8] NIEMANN H., HASSELBRING W., WENDT T., WINTER A., MEIERHOFER M. (2002). Kopplungsstrategien für Anwendungssysteme im Krankenhaus. *Wirtschaftsinformatik* (5/2002)
- [9] WENDT T., BRIGL B., WINTER A. (2001). Modellierung von Integrationstypen.. *Informatik, Biometrie und Epidemiologie in Medizin und Biologie* **32**(2-3), 331 (Abstract).
- [10] WENDT T, HÄBER A, T, BRIGL B, WINTER A: Modeling Hospital Information Systems (Part 2): Using the 3LGM² Tool for Modeling Patient Record Management Zur Veröffentlichung eingereicht.
- [11] WINTER A., AMMENWERTH E., BRIGL B., HAUX R. (2002). Krankenhausinformationssysteme. In Lehmann T., Bexten E.M.z. Hrsg. (2002). *Handbuch der Medizinischen Informatik*. Hanser, München. S. 473-552.
- [12] WINTER A, BRIGL B, WENDT T: Modeling Hospital Information Systems (Part 1): The Revised Three-Layer Graph-Based Meta Model 3LGM². *Methods Inf Med*. Zur Veröffentlichung angenommen.
- [13] WINTER A., BRIGL B., WENDT T. (2001). A UML-based Ontology for Describing Hospital Information System Architectures. In Patel V.L., Rogers R., Haux R. Hrsg. (2001). *MEDINFO 2001*. IOS, Amsterdam. S. 778-782.
- [14] WINTER AF, AMMENWERTH E, BOTT OJ, BRIGL B, BUCHAUER A, GRÄBER S, GRANT A, HÄBER A, HASSELBRING W, HAUX R, HEINRICH A, JANSSEN H, KOCK I, PENDER O-S, PROKOSCH H-U, TERSTAPPEN A, WINTER A (2001). Strategic Information Management Plans: The Basis for systematic Information Management in Hospitals. *International Journal of Medical Informatics* **64**(2-3), 99-109.
- [15] BRIGL B, AMMENWERTH E, DUJAT C, GRÄBER S, GROBE A, HÄBER A, JOSTES C, WINTER A. Preparing strategic information management plans for hospitals: a practical guideline. Submitted for publication.
- [16] WENDT T (2003): Modellierung von Architekturstilen mit dem 3LGM². Universität Leipzig, Institut für Medizinische Informatik, Statistik und Epidemiologie. IMISE Reports 2/2003, Universität Leipzig..

In der Reihe IMISE-REPORTS sind bisher erschienen:

2002

- | | | |
|--------|---|---|
| 1/2002 | Barbara Heller, Markus Löffler | Teleomatics and Computer-Based Quality Management in a Communication Network for Malignant Lymphoma |
| 2/2002 | Barbara Heller, Katrin Kühn, Kristin Lippoldt | Report OntoBuilder |
| 3/2002 | Barbara Heller, Katrin Kühn, Kristin Lippoldt | Handbuch OntoBuilder |
| 4/2002 | Barbara Heller, Katrin Kühn, Kristin Lippoldt | Leitfaden für die Eingabe von Begriffen in den OntoBuilder |
| 5/2002 | Mitarbeiter des IMISE | Skriptenheft für Medizinstudenten
Medizinische Biometrie
Medizinische Statistik und Informatik
(Kursus zum Ökologischen Stoffgebiet) |

2003

- | | | |
|--------|---|--|
| 1/2003 | Birgit Brigl, Thomas Wendt, Alfred Winter | Ein UML-basiertes Meta-Modell zur Beschreibung von Krankenhausinformationssystemen |
| 2/2003 | Thomas Wendt, Birgit Brigl, Alfred Winter | Modellierung von Architekturstilen mit dem 3LGM ² |
| 3/2003 | Birgit Brigl, Thomas Wendt, Alfred Winter | Requirements on tools for modeling hospital information systems |