



Fakultät II – Informatik, Wirtschafts- und Rechtswissenschaften
Department für Informatik

Bachelor-Arbeit

Entwicklung eines Patientendaten Verwaltungs-Prototypen mit dem besonderen Fokus: Datenintegration

vorgelegt von

Sascha-Ulf Habenicht

Gutachter:

Prof. Dr. Dr. h.c. H.-J. Appelrath

Dr. Marco Grawunder

3. Januar 2010

Danksagung

An dieser Stelle bedanke ich mich bei all jenen, die mich bei der Erstellung dieser Bachelor-Arbeit unterstützt haben.

An erster Stelle sind hier die Cochlear Implant Zentren der Universitätskliniken Köln und Essen zu nennen. Ohne die gute Zusammenarbeit wäre es mir nicht möglich gewesen, die Zusammenhänge innerhalb der Zentren zu verstehen.

Zu großem Dank verpflichtet bin ich auch Dr. Marco Grawunder, der sich viel Zeit für die Betreuung genommen hat.

Erwähnen möchte ich an dieser Stelle außerdem, dass die zu Grunde liegende MS Access-Datenbank vom Cochlear Implant Zentrum Köln insbesondere von Dirk Fürstenberg seit über zehn Jahren entwickelt wird und für eine Neuentwicklung uneingeschränkt zur Verfügung stand. Nicht unerwähnt darf der Einsatz von Heinz Klünter bleiben, der viele Anregungen lieferte und mich bei der Kommunikation mit dem Rechenzentrum in Köln sowie der Firma Avantgarde und Auritec unterstützte.

Zusammenfassung

Innerhalb dieser Bachelor-Arbeit wird ein Patientendaten-Verwaltungs-Prototyp für Cochlear-Implant(CI)-Zentren mit dem besonderen Fokus auf Datenintegration entwickelt.

Ein CI ist eine Hörprothese für Gehörlose mit funktionierenden Hörnerv. Ein CI-Zentrum ist oft Teil einer Hals-, Nasen- und Ohrenklinik und besteht aus einem interdisziplinären Team. Die CI-Zentren an den Universitätskliniken Köln und Essen verwalten Patientendaten jeweils mit Hilfe selbst entwickelter MS Access-Datenbanken. Diese werden seitens der Kliniken als nicht ausreichend empfunden. Patientendaten (Stammdaten, Ergebnisse von Hörtests, etc.) müssen zur Zeit manuell aus verschiedenen Softwareprodukten in die MS Access-Datenbanken übertragen werden. Zur Ablösung der MS Access-Datenbanken soll ein neues System entwickelt werden. Hierzu sollen die vorhandenen Datenbanken und weitere Anforderungen mittels eines Prototypen untersucht werden.

Das neue System soll an das Krankenhausinformationssystem und an medizinische Geräte angeschlossen werden. Des Weiteren werden die Benutzeroberflächen neu gestaltet, da die bisherigen Datenbanken von den Benutzern als unübersichtlich bemängelt werden.

Der Entwurf schlägt die Entwicklung einer Eclipse Rich Client Plattform vor, da diese die Anforderungen am Besten erfüllen kann. Die Datenintegrationen werden als eigene Systeme entwickelt und beliefern im Prototypen das Datenbankmanagementsystem mit neuen Informationen. Bei Health Level 7 (HL7) kommt ein Open-Source Parser zum Einsatz, die Gerätedatentransfer(GDT)-Nachrichten werden nur mit ausgewählten Testdaten im Prototypen umgesetzt.

Es wird darauf hingewiesen, dass das endgültige System eine Mehrschichten-Architektur aufweisen sollte, so dass die Systeme nur über ein Application-Server kommunizieren können.

Inhalt

1	Einleitung	1
1.1	Ziele der Arbeit	2
1.2	Aufbau der Arbeit	3
2	Grundlagen	5
2.1	Datenintegration	5
2.2	Krankenhausinformationssystem	7
2.3	Medizinische Geräte	10
2.4	Eclipse	13
2.5	Zusammenfassung	17
3	Systemanalyse	19
3.1	Beschreibung der Ist-Situation/Bestandsaufnahme	19
3.2	Ermittlung der Anforderungen	22
3.3	Untersuchung verschiedener Lösungsansätze	25
3.4	Zusammenfassung	31
4	Entwurf und Implementierung	33
4.1	Entwurf	33
4.2	Implementierung	40
4.3	Zusammenfassung	43
5	Fazit	45
5.1	Zusammenfassung	45
5.2	Ergebnisse	45
5.3	Einführung in den Arbeitsalltag	46
5.4	Projektstand	46
5.5	Ausblick	47
A	Datenbankschema	49
B	Beispieldatensätze	57
	Glossar	61
	Abkürzungen	65
	Abbildungen	67
	Literatur	69

1 Einleitung

In dieser Bachelor-Arbeit wird ein Patientendaten-Verwaltungs-Prototyp für Cochlear-Implant(CI)-Zentren mit dem besonderen Fokus auf Datenintegration entwickelt.

Ein CI ist eine Hörprothese für Gehörlose mit funktionierenden Hörnerv. Ein CI-Zentrum ist oft Teil einer Hals-, Nasen und Ohrenklinik und besteht aus einem interdisziplinären Team. Die CI-Zentren an den Universitätskliniken Köln und Essen verwalten Patientendaten jeweils mit Hilfe selbst entwickelter MS Access-Datenbanken. Diese werden seitens der CI-Zentren als nicht ausreichend empfunden. Patientendaten (Stammdaten, Ergebnisse von Hörtests, etc.) müssen zur Zeit manuell aus verschiedenen Softwareprodukten in die MS Access-Datenbanken übertragen werden. Dies geschieht nicht vollständig, da einige Fachbereiche nicht mit den MS Access-Datenbanken arbeiten oder die Datenmengen aus der jeweiligen Datenquelle zu umfangreich sind. Mit Hilfe der Datenintegration sollen hierzu Informationen zum Patienten automatisiert in der zu entwickelnden Datenbank persistent¹ gespeichert werden. Vorhandene Datenquellen sind das Krankenhausinformationssystem (KIS) und Informationssysteme von medizinischen Geräten in den CI-Zentren.

Beide CI-Zentren wachsen. Die steigende Anzahl von CI-Trägern ist auf verbesserte Techniken und neue Verfahren bei der Früherkennung zurückzuführen. Die MS Access-Datenbanken sind nicht weiter ausbaubar und zeigen Probleme auf, wie z.B. lange Zugriffszeiten. Dabei muss die Entwicklungsdauer, der Wartungs- und Pflegeaufwand der Software überschaubar bleiben. Zur Einschätzung des Entwicklungsaufwandes des Systems wird in dieser Arbeit ein Prototyp entwickelt. Der Prototyp wird mit Hilfe eines Frameworks entwickelt, welches auch in der Entwicklung des eigentlichen Systems benutzt werden soll.

Damit auch im Fall von Netzwerkproblemen Patientendaten weiter bearbeitet werden können, wird ein Rich Client entwickelt. Für die Eclipse Rich Client Plattform (ERCP) spricht die Plug-in-Technologie, die neben der Entwicklung auch die Wartung und Pflege im Betrieb vereinfachen kann. Mit Hilfe der Plug-in-Technologie der ERCP können für jede Universitätsklinik nicht nur allgemeine, sondern auch individuelle Benutzergruppen und Schnittstellen entwickelt werden. Der Prototyp besteht aus drei Komponenten:

1. **Datenbank:** Alle Daten werden auf einem zentralen Server in einer Datenbank gespeichert.
2. **Benutzerschnittstellen:** Die Benutzergruppen „CI-Technik“ und „Audiologie und Pädaudiologie“ erhalten ein eigenes Plug-in als Benutzerschnittstelle. Es können nur Daten verarbeitet werden, die für den Fachbereich benötigt werden.
3. **System-Plug-ins:** Zur Kommunikation des Prototypen mit dem Krankenhausinformationssystem (KIS) wird das Kommunikationsprotokoll Health Level Seven (HL7) zum Nachrichtenaustausch verwendet. Der Prototyp wird diese Nachrichten nur lesend verarbeiten, also nicht Nachrichten in HL7 ins KIS senden. Die Kliniken haben zudem verschiedene Diagnosegeräte, die jeweils mit einem PC verbunden sind. Auswertungen werden zur Zeit ausgedruckt bzw. in einem lokalen Informationssystem gespeichert. Um diese Daten auch innerhalb des CI-Zentrums verarbeiten zu können, wird zunächst die Schnittstelle Gerätedatentransfer (GDT) in Betracht gezogen.

Nach Abbildung 1.1 besteht das CI-Zentrum der Universitätsklinik Köln aus den interdisziplinären Fachbereichen CI-Sekretariat, HNO-Ärzte, CI-Technik und Pädagogik. Andere Funktionsbereiche sind innerhalb der HNO-Klinik angesiedelt und können als Dienstleister des CI-Zentrums gesehen werden. Ein CI-Zentrum implantiert CIs an den Hörnerv, betreut die Technik und therapiert diese um

¹ aus lat. *persistere* „verharren“, materialisiert, dauerhaft speichern

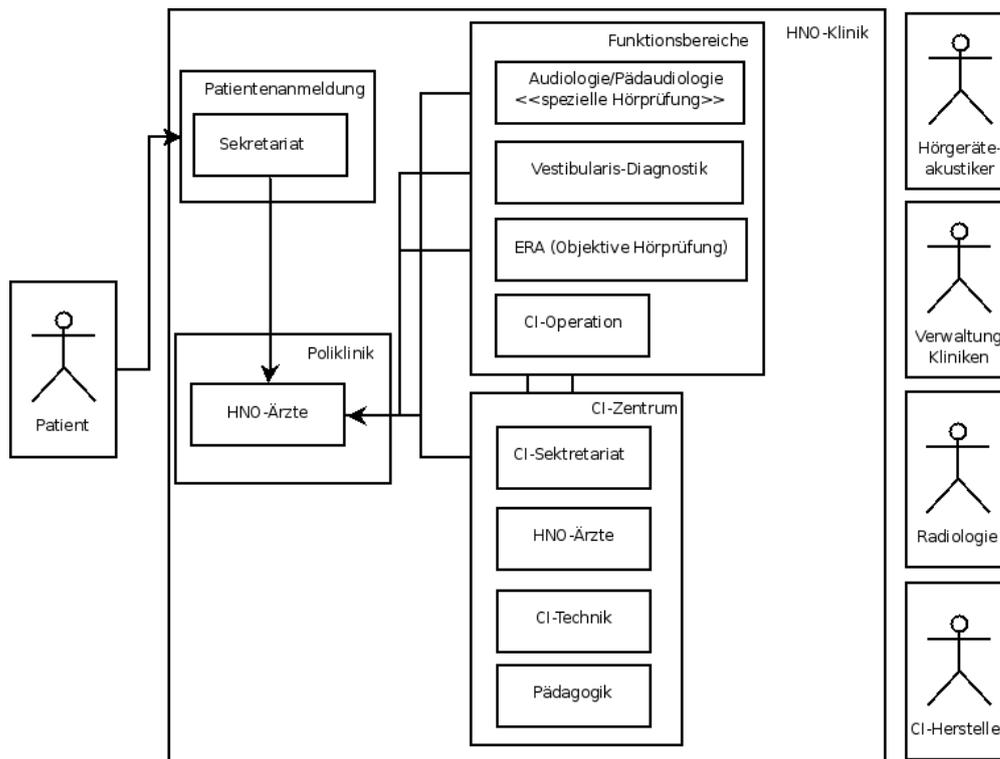


Abbildung 1.1: Fachlandkarte HNO-Uniklinik (Köln)

mit CI-Implantaten hören zu können. Des Weiteren wird regelmäßig mit Hörgeräteakustikern, CI-Herstellern, der Verwaltung der Klinik und der klinischen Radiologie kommuniziert.

Die Gruppe „Audiologie und Pädaudiologie“ misst die jeweils aktuelle Hörfähigkeit des Patienten. Diese Messergebnisse sind für die Versorgung des Patienten unentbehrlich. Die „Audiologie und Pädaudiologie“ benötigt aber keine Ergebnisse der intraoperativen Messungen der „CI-Technik“. Das zu entwickelnde System soll eine Nutzer- und Rechteverwaltung erhalten, in der Mitarbeiter Gruppen zugeordnet werden können.

1.1 Ziele der Arbeit

Die Ziele dieser Arbeit bestehen aus:

- der Erhebung der funktionalen Anforderungen der Benutzergruppen,
- der Untersuchung der notwendigen nichtfunktionalen Anforderungen,
- der Nennung von Rahmenbedingungen zum Informationsaustausch mit dem Krankenhausinformationssystem,
- der Klärung des Datenaustauschs mit Medizinischen Geräten,
- der Untersuchung des Einsatzes der ERCP Frameworks zur Umsetzung des Systems,
- der Vorbereitung der Zusammenführung aller Patienteninformationen in einer Datenbank und
- der Auswertung der Ergebnisse des Prototypen für die Softwareentwicklung.

Zur Überprüfung der funktionalen Anforderungen einer Benutzergruppe werden die zugehörigen Anforderungen zusammengefasst und in einer Ansicht im Prototypen umgesetzt. Die nichtfunktionalen Anforderungen gelten für alle Benutzergruppen und werden ebenfalls zusammengefasst. Der Prototyp wird mit Hilfe eines Frameworks entwickelt, um sich hier mit diesem vertraut zu machen und eventuelle Probleme hieraus für die Softwareentwicklung im Voraus zu erkennen. So kann es z.B. sein, dass die Software nicht auf allen Rechnern im Universitätsklinikum Köln funktionieren. Hier befinden sich noch viele ältere Thin-Clients im Einsatz, welche gegebenenfalls nicht genug Leistung für jedes Plug-in erbringen. Bei der Datenintegration muss das Umfeld der anderen Informationssysteme dahingehend untersucht werden, welche Formate und Protokolle verwendet werden.

Da der Datenaustausch ein wesentliches Thema dieser Arbeit ist, müssen spezielle, für den Gesundheitsstandard entwickelte Protokolle untersucht werden. Diese sind Health Level Seven (HL7) und Gerätedatentransfer (GDT). HL7 dient zum Datenaustausch mit dem KIS und GDT wird zum Datenaustausch mit medizinischen Geräten verwendet. Probleme mit anderen Informationssystemen können dann entstehen, wenn z.B. Informationen der Hersteller zu ihren Schnittstellen, Schnittstellen nicht aktiviert werden oder zeitliche Verzögerungen seitens der Hersteller eine Entwicklung behindern.

1.2 Aufbau der Arbeit

Das zweite Kapitel befasst sich mit den notwendigen Grundlagen Datenintegration, KIS, Medizinischen Geräten und der Rich Client Plattform von Eclipse. Die Datenintegration ist weiter untergliedert in Überblick, Verteilung, Autonomie und Heterogenität sowie einen Beispielprozess. Anschließend wird KIS eingeführt. Die Software wird innerhalb des KIS Nachrichten zum Patienten erhalten, wie z.B. Aufnahme oder Adressänderungen. Eine weitere Datenquelle zur Datenintegration stellen die medizinischen Geräte dar. Hier wird oft GDT, welches zur Gruppe der xDT Datenaustauschformate gehört, zum Datenaustausch benutzt. Im dritten Kapitel wird eine Systemanalyse mit einer Ist-Situation durchgeführt und darauf aufbauend die Anforderungen ermittelt. Am Schluss des dritten Kapitels werden verschiedene Lösungsansätze miteinander verglichen. Das Kapitel vier beschreibt den Entwurf und die Implementierung des Prototypen. Im fünften Kapitel werden die wichtigsten Ergebnisse zusammengefasst und es wird ein Ausblick auf die notwendigen weiterführenden Arbeiten gegeben.

2 Grundlagen

In diesem Kapitel werden die Grundlagen der Datenintegration und Grundlagen zur technischen Umsetzung des Prototypen vorgestellt. Dabei wird in Abschnitt 2.1 auf die Datenintegration eingegangen. Hier werden eine Definition, ein Überblick, Grundprobleme und ein Beispielprozess zur Datenintegration dargestellt. In Abschnitt 2.2 wird der Begriff „Krankenhausinformationssystem (KIS)“ eingeführt. Kommunikationsstandards reduzieren die Implementierung von Kommunikationsbeziehungen zwischen Softwaresystemen. Die Protokolle HL7 und GDT sind im klinischen Bereich am häufigsten anzutreffen. Weitere Kommunikationsprotokolle sind EDIFACT (Electronic Data Interchange for Administration, Commerce and Transport) und DICOM (Digital Imaging and Communication in Medicine). EDIFACT konnte sich bislang nicht gegen HL7 durchsetzen, DICOM ist geeignet für die Bildübermittlung im Krankenhaus und Datenübertragung [LEH05]. Nach einem Abschnitt über HL7 in 2.2 wird zur Überführung in den xDT Kommunikationstandard im Abschnitt 2.3 aufgezeigt, weshalb nicht alle medizinischen Geräte HL7 unterstützen. Zur Repräsentation der medizinischen Geräte ist ein Stellvertreter ausgewählt, ein klinisches Audiometer. Es werden Details zum Gerätedatentransfer (GDT) beschrieben. Das Kapitel 2.4 liefert die Grundlagen zur Auswahl des Frameworks zur Systementwicklung, zeigt Grundlagen der verteilten Anwendung und stellt die Eclipse Rich Client Plattform vor. Anschließend wird auf die Benutzerschnittstellen, der Oberflächengestaltung der geplanten Software eingegangen.

2.1 Datenintegration

Datenintegration beschreibt das Problem, Daten aus unterschiedlichen Quellen so zu integrieren, dass sie dem Anwender in einer einheitlichen, konsistenten Sicht präsentiert werden [LEN02]. Unter Datenintegration wird auch die redundanzfreie (oder wenigstens redundanzarme) Repräsentation von Informationen verstanden, die in einer Menge von sich inhaltlich überlappenden Systemen vorliegen [FN07]. Die Redundanzfreiheit gilt für die Schemaebene als auch für die Datenebene. So kann es z.B. zu Redundanz führen, wenn eine Person mit zwei verschiedenen Adressen vorkommt. Diese Person muss als dieselbe erkannt werden und sollte in einem Datensatz zusammengefasst werden. Der Konflikt mit den Adressen muss aufgelöst werden.

Zusammenfassend ist die Aufgabe der Datenintegration also eine konsistente Sicht, welche bereinigte Daten zur Verfügung stellt. Die Datenintegration gehört zu den Problemfeldern der Informatik und hat die Aufgabe, Informationseinheiten zum selben Patienten in den Benutzergruppen zusammen zu fassen [REI73].

Der erste Schritt zur Datenintegration besteht darin, Informationsquellen zu identifizieren. Nach der Identifizierung muss der Zugriff auf die Daten geklärt werden. Dies geschieht oft mit Hilfe von Datenformaten oder Protokollen. Der Datenaustausch kann auf mehreren Wegen erfolgen und hängt von der Quelle ab. Dies kann z.B. über Nachrichten, Extraktion- oder Reportingwerkzeuge geschehen. Wenn die Zugriffsart auf die Daten geklärt ist, müssen die Daten transformiert und den Benutzern je nach Szenario persistent oder virtuell zur Verfügung gestellt werden.

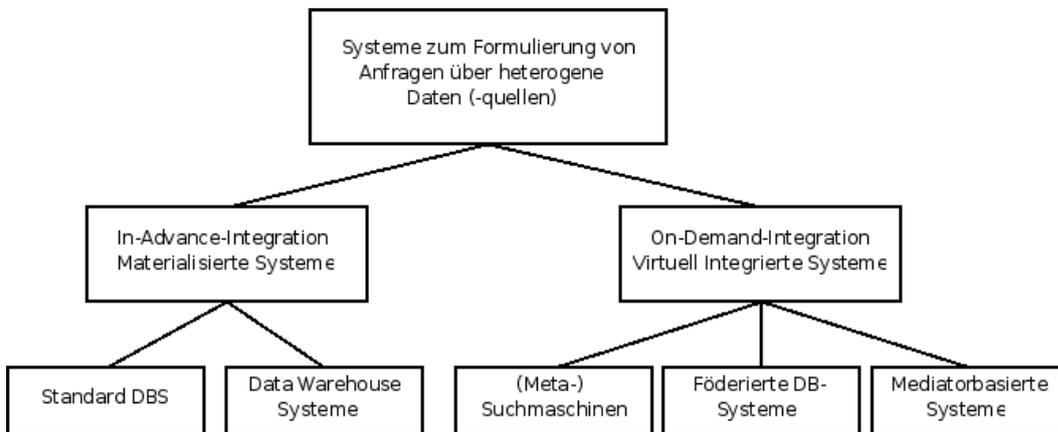


Abbildung 2.1: Klassifikation von Anfragesystemen nach [KRD99]

2.1.1 Klassifikation von Anfragesystemen

Zur Integration von Daten werden, wie schon aufgezeigt wurde, Quellen abgefragt oder Daten empfangen. Anfragesysteme können dahingehend unterschieden werden, ob die Daten in einer Datenbank persistent gespeichert werden oder den Benutzern virtuell zur Verfügung gestellt werden (vgl. Abb. 2.1). Virtuuell integrierte Systeme sind z.B. Meta-Suchmaschinen. Mit Hilfe dieser werden die Anfragen über ein Webformular an mehreren Quellen gestellt. Durch eine solche Abfrage können die Daten gegenüber einer Suchmaschine für den Anwender bereinigt, bewertet und vervollständigt werden. Ein weiterer Vertreter eines virtuell integrierten Anfragesystems sind die Föderierten Datenbanksysteme. Mit ihrer Hilfe wird dem Anwender eine Sicht über mehrere Datenbanken geboten, wobei Abfragen in Echtzeit ausgeführt werden. Ein großer Nachteil virtueller Systeme sind u.a. lange Antwortzeiten bei hoher Auslastung der Quellen oder keine Verarbeitung von Anfragen bei Ausfall von Quellen. Vertreter persistenter Speicherung der integrierten Daten sind Data-Warehouse-Systeme (DWHS) und klassische Datenbanksysteme. DWHS liegen nach [HG09] in einem Dualismus von Informatik und Betriebswirtschaft. Der Schwerpunkt liegt bei der Auswertung unter betriebswirtschaftlichen Gesichtspunkten. Auf der Anwenderseite werden die Daten zu Analyse-Zwecken verwendet.

2.1.2 Verteilung, Autonomie und Heterogenität

Bei der Konstruktion von verteilten Softwaresystemen mit Datenintegration existieren die Grundprobleme Verteilung, Autonomie und Heterogenität. Diese drei Problemfelder werden auch als orthogonale Dimensionen der Datenintegration bezeichnet [FN07]. Die Dimensionen können bei der Beschreibung von Informationsquellen hilfreich sein. Von Verteilung wird gesprochen, wenn Daten auf unterschiedlichen Systemen liegen. Die Verteilung kann physisch (geografisch getrennt) oder logisch verteilt (mehrere Orte zur Speicherung) sein. Mit Autonomie wird die Freiheit der Datenquellen bezeichnet, Entscheidungen über die verwalteten Daten, ihre Struktur und Zugriffsmöglichkeiten zu treffen. Die Autonomie kann weiter unterteilt werden, wie z.B. in Designautonomie oder Schnittstellenautonomie. Die Schnittstellenautonomie bezeichnet die Freiheit der Datenquelle, mit welchen technischen Verfahren auf die Daten zugegriffen werden kann. Dies umfasst Entscheidungen über Anfragesprache und Protokolle. Entscheidungen innerhalb der Schnittstellenautonomie haben direk-

ten Einfluss auf die Designautonomie. Die Designautonomie trifft Entscheidungen über die Art und Weise, wie Daten zur Verfügung gestellt werden. Dies umfasst neben der Art des Zugriffs, Datenformat, Datenmodell, syntaktische Darstellung der Daten, Einheiten von Werten und die Verwendung von Schlüssel- und Begriffssystemen. Bei unterschiedlichen Informationssystemen kommt es i.d.R. zu Heterogenität, da die Datenmodelle selten exakt gleich sind, unterschiedliche Methoden zur Verfügung stehen und Zugriffe oft unterschiedlich sind.

2.1.3 Prozesse

Im Prinzip besteht der Vorgang der Datenintegration immer aus drei Schritten: Extraktion-, Transformation- und Ladeprozess. Diese drei Prozesse werden meist im Kontext von DWHS als Extract-Transform-Load, abgekürzt ETL-Prozess, bezeichnet. Anhand des ETL-Prozesses werden die Arbeitsschritte eines möglichen Vorgehens der Datenintegration näher erläutert.

1. **Extraktion:** In der Phase der Extraktion werden z.B. mit Hilfe von Skripten, abhängig von der jeweiligen Datenquelle, die Informationen ausgelesen. Dies geschieht bei DWHS oft einmal am Tag. Dazu werden z.B. SQL-Skripte, aber auch Export- und Reportingwerkzeuge benutzt.
2. **Transformation:** In diesem Schritt werden die Informationen in das Format und die Struktur des Data-Warehouse (DWH) transformiert. Dazu gehören auch die Umformung von Datumswerten oder die Umrechnung von Einheiten, wie z.B. Dollar in Euro.
3. **Laden (Load):** Beim Laden werden die Informationen in einen Arbeitsbereich geladen und von dort aus in Schema des DWHS persistent gespeichert.

2.2 Krankenhausinformationssystem

Der Begriff KIS wird in Krankenhäusern, bei Wissenschaftlern und der Software- und Hardwareindustrie nicht einheitlich verwendet. Erschwerend kommt hinzu, dass weitere Begriffe verwendet werden. Lehmann [LEH05] definiert ein KIS als sozio-technisches System aller Informationsverarbeitung, -übermittlung und -speicherung im Krankenhaus. Es wird nicht nur als administrative Informationsverarbeitung des Krankenhauses verstanden.

„Ein Krankenhausinformationssystem (KIS) ist das soziotechnische Teilsystem eines Krankenhauses, das alle informationsverarbeitenden (und -speichernden) Prozesse und die an ihnen beteiligten menschlichen und maschinellen Handlungsträger in ihrer informationsverarbeitenden Rolle umfasst. Das KIS dient dazu, die Mitarbeiter des Krankenhauses bei der Erledigung der Aufgaben des Krankenhauses zu unterstützen. Es umfasst daher

- alle Bereiche des Krankenhauses,
- alle Gebäude des Krankenhauses und
- alle Personengruppen, die im Krankenhaus tätig sind.“

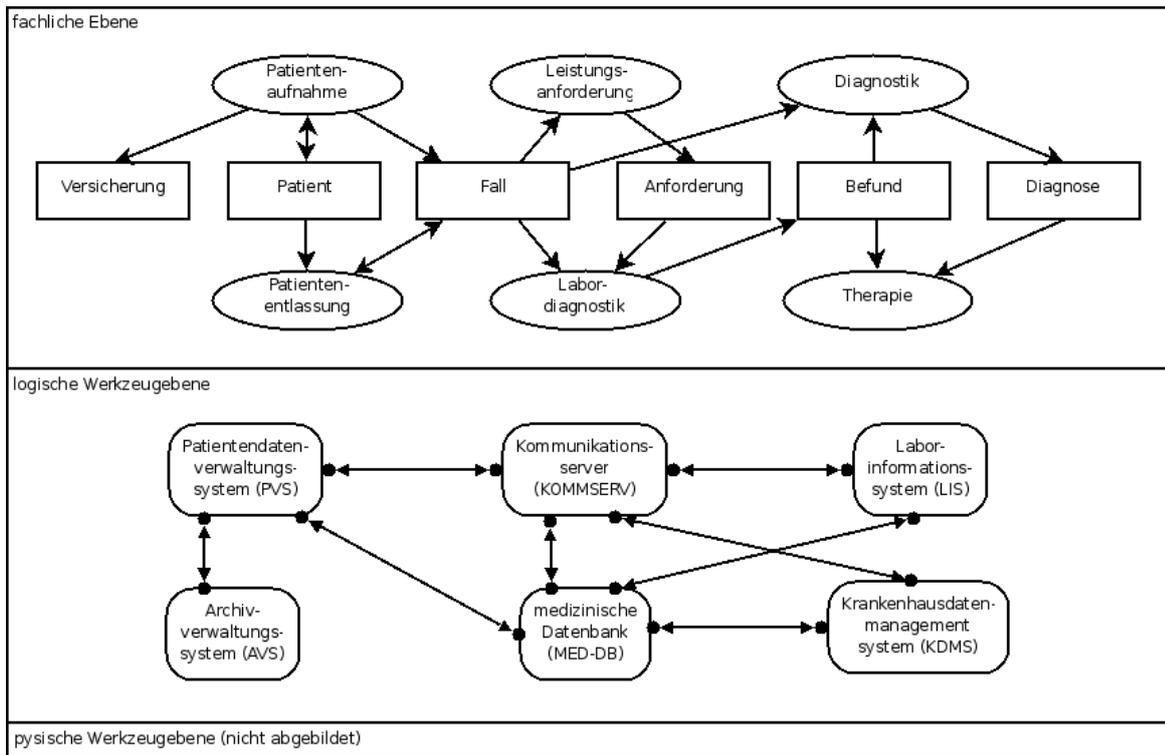


Abbildung 2.2: Das 3-Ebenen-Meta-Modell (3LGM²) [LEH05]

Die Informationen im Krankenhaus werden bei Lehmann [LEH05] in zwei Bereiche untergliedert:

1. Informationen über einzelne Patienten
2. Patientenunabhängiges Wissen

Zu den Informationen über einzelne Patienten gehören unter anderem Patientenstammdaten, Voraufenthalte, Diagnosen, Laborbefunde. Dagegen fallen unter patientenunabhängiges Wissen aktuelle Informationen über Krankheiten und Therapien sowie Management und Abrechnungsdaten. Informationen können als Nachrichten über das KIS versendet oder im KIS gespeichert werden. Datenintegration liegt in einem KIS vor, wenn Informationen nur einmal im KIS erfasst werden und nicht bei weiteren Aufgaben erhoben werden müssen [LEH05].

2.2.1 Architektur

Informationssysteme im Gesundheitswesen lassen sich mit Hilfe des Drei-Ebenen-Meta-Modell (3LGM²) beschreiben, bewerten und planen [W3L09]. Ein 3LGM² zum Aufbau eines KIS ist in Abbildung 2.2 zu sehen, es besteht auf der logischen Werkzeugebene aus folgenden Anwendungsbausteinen:

1. Patientenverwaltungssystem (PVS)
2. Archivverwaltungssystem (AVS)
3. Krankenhausdatenmanagementsystem (KDMS)
4. Laborinformationssystem (LIS)
5. medizinische Datenbank (MED-DB)
6. Kommunikationsserver (KOMMSERV)

Im PVS liegen die Patientenstammdaten der aktuellen Patienten, sowie die zugehörigen administrativen Daten (z.B. Zimmernummer). Im AVS werden die Daten nach einer Entlassung oder nach einer ambulanten Therapie langfristig gespeichert. Das KDMS verwaltet die notwendigen Informationen die zur Verarbeitung des administrativen Personals notwendig sind. Hier befinden sich alle Informationen zur Abrechnungen der Leistungen und liefert notwendige betriebswirtschaftliche Informationen. Das LIS nimmt für eine Laboruntersuchung die Patientenstammdaten und Untersuchungsanfrage entgegen. Nach der Laboruntersuchung wird das Laboreergebnis des Patienten an das PVS geschickt. In der MED-DB liegt das medizinische Wissen aufbereitet vor, damit das medizinische Personal notwendige Informationen für die Behandlung und Pflege der Patienten zur Verfügung hat. Ein KOMMSERV muss nicht zwangsläufig bei einem KIS vorhanden sein, erleichtert jedoch das Zusammenspiel der Anwendungsbausteine untereinander. Ein KOMMSERV nimmt Informationen entgegen und sendet diese an ein oder mehreren Empfänger aus einer Sendeliste weiter. Der KOMMSERV kann dabei das empfangene Datenformat in ein anderes Datenformat transformieren, je nachdem was der Empfänger verarbeiten kann. Als Datenformat hat sich in den Kliniken HL7 durchgesetzt, das im nächsten Abschnitt vorgestellt wird.

2.2.2 Health Level Seven

Wie schon erwähnt, werden in einem KIS Nachrichten ausgetauscht. Zum technischen Austausch wird eine standardisierte Syntax benutzt, da Algorithmen zur Auswertung von Freitext nicht zuverlässig genug sind. Durch die Standardisierung werden Inhalte vorweggenommen, woraus sich Klassen von Nachrichtentypen ergeben. Der Health Level 7 (HL7) Kommunikationsstandard ist im klinischen Bereich am weitesten vertreten. Im Universitätsklinikum Köln befindet sich HL7 in der zweiten Version im Einsatz.

HL7 ist ereignisgetrieben, Nachrichten werden durch Ereignisse in der realen Welt ausgelöst. Beispiele für Ereignistypen sind:

- A01: Aufnahme eines Patienten
- A02: Verlegung eines Patienten innerhalb des Krankenhauses
- A03: Entlassung eines Patienten

Ereignisse treten bei neuen Informationen, Anforderung und Übermittlung von Informationen auf [SCH00]. Ereignistypen sind Nachrichtentypen zugeordnet, welche Aufschluss über die Segmente der Nachricht geben, z.B. ist ORU ein Nachrichtentyp für einen Befund. Der Aufbau einer Nachricht wird in Abb. 2.3 gezeigt. Segmente sind Datensätze, die in Felder unterteilt sind. Ein Segment liegt in einer Zeile, am Anfang steht die Bezeichnung des Segments. Segmente wiederum gliedern die Datenfelder in eine bestimmte Reihenfolge auf, die durch ein Trennzeichen voneinander getrennt sein müssen. Nachfolgend ist ein Beispiel für eine HL7-Nachricht gegeben ¹:

¹ HL7 Datensatz wurde vom ZIK Köln zur Verfügung gestellt.

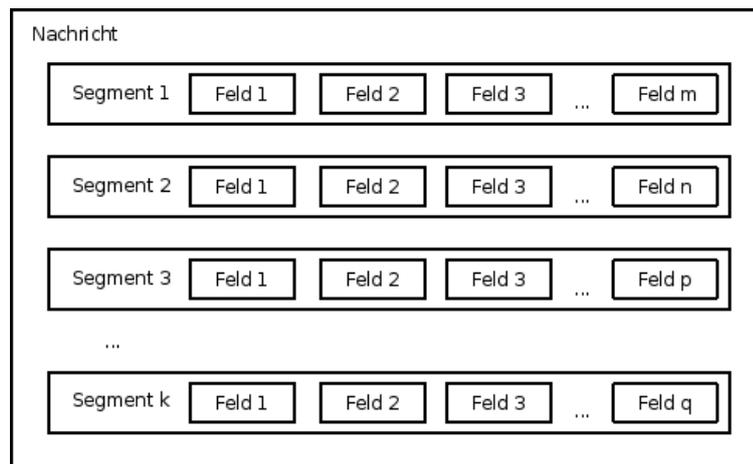


Abbildung 2.3: Aufbau einer HL7-Nachricht [KOC03]

1. **MSH**^ &IISHI0001|Orbis|0001|20090806064858||ADT^A03|2009080606490400329949|P|2.3|||ALI
NEIDE<CR>
2. **EVN**|A03|20090803100345||NP97I0|IEHLERS<CR>
3. **PID**||5824882^8^M11||Mustermann^Manuela^^^^Dr.|Hoppel19410913|F|||Mustermann-Str. 2^^
Gummersbach^^51645^D||0226175921|<CR>
4. **PV1**||TS|00000615^^^06TK^^^1|AS||00000615^^|^^^^^^|^^^^^^|0004024348||N|||3562542^2^M11|
|K|||K|||P9|||010|||20090803100206|20090803100345|||H PV2||PTDERM|||EH<CR>
5. **ZBE**|0356254200002^ISHI|20090803100345||INSERT<CR>

Alle Nachrichten beginnen mit MSH (Message Header). Hier werden der Empfänger und Absender festgehalten. EVN (Event) gibt das Ereignis an. Beim PID (Patient Identification) wird der Patient identifiziert. Das Segment PV1 (Patient Visit) enthält Daten zur Beschreibung von Krankheits- bzw. Behandlungsfällen, wie z.B. einer Fallnummer. ZBE beinhaltet Bewegungsdaten des Patienten. Mit A03 in MSH handelt es sich bei dieser Nachricht um eine Entlassung der Patienten Manuela Mustermann.

Aktuell ist die dritte Version von HL7, die große Unterschiede zu Version 2.x hat. Version 3 verwendet XML Encoding, statt einen Nachrichten-Paradigma wird ein Architektur-Paradigma verfolgt. Als Grundlage dient ein generisches Reference Information Model (RMI) mit sechs Kernklassen (Entitäten, Rollen, Beteiligungen, Aktionen, Rollenbeziehungen und Aktionsbeziehungen) [KOC03].

2.3 Medizinische Geräte

Viele medizinische Geräte unterstützen kein HL7. Dies sind oft Geräte, die für den Einsatz in Arztpraxen entwickelt wurden. Eine Arztpraxis kommuniziert mit Laboren, Krankenkassen und mit Kliniken. Bei vielen medizinischen Geräten wäre eine Anbindung an das KIS zu aufwendig, da die gewonnenen Daten z.B. nicht klinikweit von Interesse sind oder das Papierformat gegenüber dem Aufwand einer Einbindung an das KIS als ausreichend empfunden wird. Bei einer klinikweiten Anbindung mit HL7 müssen die Geräte z.B. an den KOMMSERV angeschlossen werden, wohingegen ein Gerätedatentransfer lokal mit einer Software verarbeitet werden kann. Als Stellvertreter für die medizinischen Geräte wird ein Audiometer vorgestellt, anschließend wird der Gerätedatentransfer im Detail betrach-



Abbildung 2.4: Audiometriemodul AT900

tet. Ein Audiometer stellt die Hörfähigkeit des Patienten fest und dient der Beurteilung und auch der Erfolgskontrolle. Kein CI-Zentrum wird ohne einem Audiometer Patienten versorgen können.

2.3.1 Klinisches Audiometer AT900

In den Unikliniken werden Audiometer zur Bestimmung der Hörfähigkeit der Patienten verwendet. Bei einem Audiometer können Frequenzen verschiedener Tönhöhe erzeugt werden. Tonmessungen können dabei über ein Luftleitungskopfhörer, Knochenleitungshörer oder per Lautsprecher übermittelt werden. Über ein Pegelregler können verschiedene Lautstärken eingestellt werden. Mit Hilfe von genormten Sprachmaterial wird das Sprachverständnis ermittelt. Durch Auswertung dieser Messergebnisse können Ärzte Diagnosen stellen oder überprüfen. In der Rehapphase, z.B. im Anschluss an eine CI-Implantation, können Hörerfolge nachgewiesen werden. Audiologen und Hörtherapeuten können anhand der Ergebnisse den Therapieverlauf besser planen.

Im Benutzerhandbuch [AUR07] zum Klinischen Audiometer AT900 steht: “Das Audiometriesystem AT900 ist ein PC-gesteuertes Audiometer. Das System bietet die Möglichkeit der audiologischen Messdatenspeicherung und der Datenübertragung in die elektronische Patientenakte bzw. in ein Datennetzwerk. [...] Die Audiometriesoftware AT 904-1 bildet mit dem Audiometriemodul eine Einheit.” Das klinische Audiometer AT900 ist in Abbildung 2.4 zu sehen, es gibt neben dem Audiometer AT900 auch das Audiometer AT700 und die AT960 Kinderaudiometrieanlagen. Die AT960 baut auf AT900 auf, das AT700 ist ein kleinere Variante des AT900. Die verschiedenen Audiometer befinden sich alle in den CI-Zentren im Einsatz. Das Steuerungsmodul wird benutzt um die Frequenz, Lautstärke und Antworten des Patienten einzugeben. Wird z.B. ein Satztest durchgeführt, sitzt der Patient im Abstand von einem Meter vor dem Messlautsprecher. Schallquellen befinden sich aus Sicht des Patienten links,

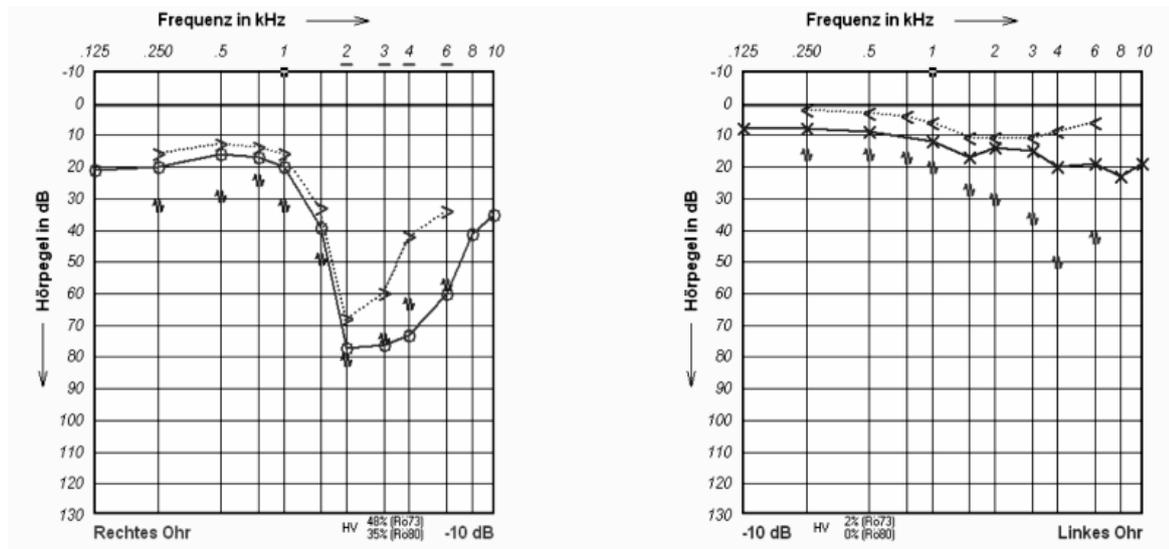


Abbildung 2.5: Grafische Auswertung Tonaudiometrie¹

vorne links, vorne, vorne rechts und rechts. Ein Sichtschutz verhindert, dass der Patient Einstellungen durch den Bediener sieht. Der Patient versucht, die ihm vorgesprochenen Sätze bestmöglich zu wiederholen. Dabei wird z.B. der prozentuale Anteil der richtig wiedergegebene Sprache gespeichert. Die Ergebnisse werden mit Hilfe der Audiometriesoftware oder bei Vergleichen mit vorherigen Ergebnissen in der Software Avantgarde, grafisch aufbereitet (zu sehen in Abbildung 2.5). Bei einer Tonaudiometrie werden z.B. Frequenz in kHz gegenüber dem Hörpegel in dB grafisch dargestellt. Für jedes Ohr gibt es getrennte Audiogramme. Die Befunde des rechten Ohres werden auf der linken Seite und die des linken Ohres auf der rechten Seite dargestellt. Dies entspricht der Darstellungsart, die Ohrenärzte gewöhnt sind [RL09].

2.3.2 Gerätedatentransfer

Daten werden in Arztpraxen meist mit Hilfe der GDT-Schnittstelle ausgetauscht, welche zur Gruppe der xDT-Datenaustauschformate gehört. xDT ist ein Standard der Kassenärztlichen Bundesvereinigung und ist ein Synonym für den elektronischen Datenaustausch in der Arztpraxis [KAS06]. Das Know-How wird nach Auskunft der Kassenärztlichen Bundesvereinigung vom Qualitätsring medizinischer Software e.V. weiter entwickelt². Dieser Verein ist ein Zusammenschluss von Lösungsanbietern und Dienstleistern im Gesundheitswesen, die Kassenärztliche Vereinigung ist als Mitglied weiterhin vertreten. Zu der Gruppe der xDT-Schnittstellen gehören Abrechnungsdatentransfer (ADT), Behandlungsdatentransfer (BDT), Gerätedatentransfer (GDT) und Labordatentransfer (LDT). Die Kommunikation kann bei GDT V2.0 über die Datei-Schnittstelle oder serielle Schnittstelle erfolgen, eine Programm-Programm Schnittstelle ist in der Version 2.0 als Erweiterung geplant [QUA99]. Jedes Gerät hat eine eindeutige GDT-ID, die zur Identifizierung bei der Kommunikation dient und bei der Installation festgelegt wird. Als Zeichensatz ist der IBM-8-Bit Zeichensatz (Codepage 437)

¹Copyright by Ing.-Büro J. Näß / AURITEC GmbH

²E-Mail vom 24.09.09 mit P. Becker, Kassenärztliche Bundesvereinigung.

mit Zeichen kleiner gleich 20 hex (32. dez.) erlaubt, andere Zeichensätze können gegebenenfalls unterstützt werden. Bei der Datei-Schnittstelle werden Dateien in einem Verzeichnis unter bestimmten Dateinamen abgelegt. Der Dateiname kann eindeutig (<Empfänger><Sender>.GDT) oder mit einer hochzählenden Nummer (<Empfänger><Sender>.<Nr.>) aufgebaut sein. Die Verarbeitung der Dateien erfolgt nach FIFO (First-In-First-Out), verarbeitete Dateien werden durch den Empfänger gelöscht.

Beispiel: Ein Satztest soll durchgeführt werden. Das Audiometer ist dabei ein Client, die Software mit Patientenverwaltung der Server. Innerhalb der Software ist der Patient bekannt. Der Client startet eine Anfrage per Tastendruck und schreibt hierbei eine Datei an einem vordefinierten Ort ab, z.B. /GDT/SERVER_LUFU.001. Es wird die Satzart 6300 zur Anforderung des Patienten benutzt. Der Server liest die Datei und löscht diese. Nun schreibt der Server eine Datei in /GDT/LUFU_SERVER.001 mit Satzart 6301 zur Übermittlung des aktuellen Patientenstammsatzes. Der Client liest die Datei und löscht diese, der Satztest kann nun durchgeführt werden. Wenn der Client die Daten aus dem Sprachtest speichern möchte, legt dieser eine Datei /GDT/SERVER_LUFU.002 mit Satzart 6310 zur Übermittlung des Ergebnisses ab. Dies kann beliebig oft wiederholt werden. Der Server liest die Datei oder die Dateien ein und löscht diese und ordnet die Daten den Patienten zu.

GDT ist aufgebaut wie BDT, so ist *0163101Schmidt<CR><LF>*³ die Zeilenlänge inkl. CR LF (016), Feldkennung des Zeileninhalts (3101), Inhalt des BDT-Feldes (Schmidt) und Zeilenende (CR LF / 0D 0A hex). Die Information 3101 ist der Name des Patienten.

In der Satztafel von GDT sind folgende Sätze definiert:

- 6300: Stammdaten anfordern
- 6301: Stammdaten übermitteln (Antwort auf 6300)
- 6302: Neue Untersuchung anfordern
- 6310: Daten einer Untersuchung übermitteln
- 6311: Daten einer Untersuchung zeigen

2.4 Eclipse

Eclipse ist eine bekannte Java-Entwicklungsumgebung. Darüber hinaus ist Eclipse eine Laufzeitumgebung für Plug-ins, genau genommen ist die Java-Entwicklungsumgebung selbst ein Plug-in. Wird Eclipse als Framework zur Entwicklung der eigenen Software benutzt, wird das Plug-in zur Java-Entwicklungsumgebung meist nicht mit ausgeliefert. Mit Hilfe der Plug-in-Architektur kann Eclipse als Rich Client benutzt werden. Ein Rich Client implementiert Funktionalitäten direkt beim Client. Es ist kein Problem, Eclipse mit eigenen Plug-ins auszuliefern. Der Benutzer merkt keinen Unterschied zu klassischen proprietären Anwendungen. Der Aufbau von Eclipse ist in Abbildung 2.6 dargestellt. Das Eclipse Software Development Kit (SDK) besteht aus der Eclipse Plattform und zwei Plug-ins, der Java Development Tooling (JDT) und Plug-in Developer Environment (PDE). Die Eclipse Plattform besteht aus Workbench, Workspace, Help und Team. Die Workbench ist die Benutzeroberfläche, welche mit SWT und JFace aufgebaut wird. Help stellt eine Umgebung zur Dokumentation und Hilfe da (mit Einsatz eines eigenen Webserver). Team liefert Support für alle Plug-ins, wie z.B. Versionierung. Dieses Kapitel ist in zwei Abschnitten unterteilt, Aufbau und Ablauf der Rich Client Plattform und die Verwendung der UI-Architektur, der sogenannten Workbench.

³ Beispiel ist entnommen aus [QUA99]

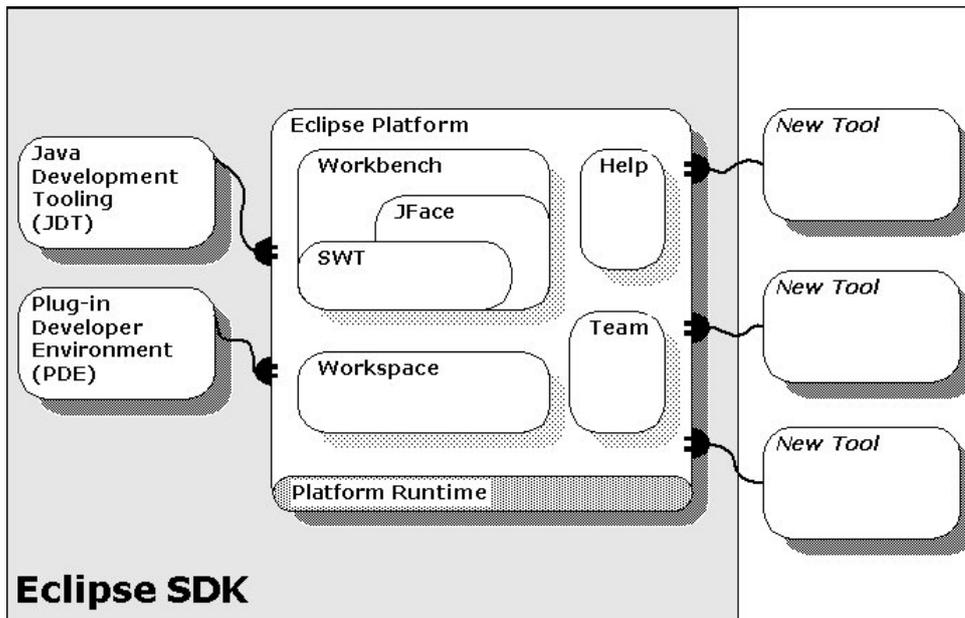


Abbildung 2.6: Eclipse SDK [ERC09]

2.4.1 Rich Client Plattform

Die Eclipse Rich Client Plattform (ERCP) basiert auf dem OSGi-Framework, das in der Lage ist, Klassen zur Laufzeit in sogenannten Bundles nachzuladen. OSGi ermöglicht die dynamische Verwaltung von Java-Komponenten [REI09]. Die Objektorientierung kann nicht spezifizieren, wer welche Klasse nutzt. Durch Typumwandlungen sind z.B. Schnittstellen nutzbar. Eine OSGi-Komponente definiert die exportierten Elemente und Schnittstellen der Klasse. Hierdurch kann genau gesteuert werden, welche Klasse auf die Komponente zugreifen kann, Schnittstellen sind in einer Komponente nicht durch z.B. Typumwandlungen nutzbar. Der Entwickler kann in einer Komponente steuern, welche Schnittstellen welchen Klassen zur Verfügung gestellt wird.

Durch das Plug-in Konzept müssen neben den Arbeiten am Quellcode bestimmte Schnittstelle zur Laufzeitumgebung des Plug-in konfiguriert und verwaltet werden. Die Konfiguration verteilt sich auf mehrere Dateien, die mit Hilfe des Plug-in Manifest Editor verwaltet lassen. Zusammengefasst besteht ein Plug-in aus zwei Teilen, der Konfiguration und der Implementierung. Zum Verständnis der ERCP wird die Konfiguration und ein Lebenszyklus zum Aufbau einer Benutzeroberfläche dargestellt.

2.4.1.1 Konfiguration

Mit Hilfe des Plug-in-Manifest Editor werden die Einstellungen zu Plug-in, ERCP, Buildprozess und OSGi-Bundle übersichtlich dargestellt und verwaltet. Es werden dazu Tools angeboten, die z.B. helfen ein neues View anzulegen.

1. Overview: Übersicht, Konfiguration des Activators.
2. Dependencies: Abhängigkeiten, die zur Laufzeit des Plug-in benötigt werden.
3. Runtime: Regelt die Sichtbarkeit der Klassen zu anderen Plug-ins.
4. Extensions: Schnittstellen, die ausserhalb des Plug-in benutzt werden können.

5. Extensions Points: präsentiert die Erweiterungspunkte, welche ein Plug-in zur Verfügung steht.
6. Build: Konfiguration des Build-Prozesses.
7. MANIFEST.MF: Konfiguration des OSGi-Manifest.
8. plugin.xml: Klassen, die bei Bedarf zur Laufzeit geladen werden, Plug-in-Manifest.
9. build.properties: Einstellungen zum compilieren, wie z.B. Pfadangaben.

Im Overview (1) kann ein Activator angegeben werden. Wenn ein Activator angegeben ist, wird diese Klasse vor allen anderen des Plug-in gestartet. Hierdurch kann z.B. ein Logmechanismus eingerichtet werden. Dependencies (2) regelt die Paket-Importen, also Abhängigkeiten zu anderen Plug-ins. Die Extensions (4) deklariert, welche Stecker das Plug-in zur Verfügung stehen [JB08]. Somit muss statisch deklariert werden, wie das Plug-in in Eclipse eingebunden wird. Die Einstellungen zum OSGi-Manifest werden in MANIFEST.MF (7) hinterlegt, wie z.B. welche Manifest-Version benutzt wird. Plugin.xml (8) ordnet Perspektiven, Views usw. mit Hilfe z.B. einer ID den jeweiligen Java-Klassen zu.

2.4.1.2 Lebenszyklus

Nachdem das Plug-in konfiguriert ist, kann dieses gestartet werden [DAU08]. Die nachfolgenden Java-Klassen sind im Plug-in-Manifest registriert, anhand der Klasse Application wird der Lebenszyklus aufgezeigt, in Abbildung 2.7 ist der Workbench Lebenszyklus in einem Sequenzdiagramm dargestellt.

1. Activator: Vor Starten der Application
2. Application: Eintrittspunkt
3. ApplicationWorkbenchAdvisor: Lebenszyklus Workbench
4. ApplicationWorkbenchWindowAdvisor: Lebenszyklus Fenster
5. ApplicationActionBarAdvisor: Aktionen

Die Benutzung eines Activators (1) ist nicht zwingend vorgeschrieben, erlaubt aber Zugriff auf die Einsatzumgebung. Es kann u.a. die Eclipse-Logdatei oder auf Benutzereinstellungen zugegriffen werden. Der Activator wird vor allen anderen Klassen verarbeitet. Mit der Application (2) beginnt der Lebenszyklus der Workbench. Sukzessiv werden nun die Klassen durch den Aufruf von Application gestartet, so startet z.B. ApplicationWorkbenchWindowsAdvisor (4) den ApplicationActionBarAdvisor (5). Die Application ist der Eintrittspunkt gegenüber Eclipse. Hier können weitere Schritte zum Lebenszyklus der Application hinzugefügt werden, wie z.B. Authentifizierung des Benutzers. Dagegen konfiguriert ApplicationWorkbenchAdvisor die Workbench. Hier wird z.B. eine ID abgelegt, mit welcher die Perspektive eindeutig zugeordnet werden kann. Es stehen mehrere Methoden zur Verfügung, wie z.B. preShutdown. Die Methode preShutdown wird aufgerufen, wenn die Ereignisschleife beendet wurde, aber bevor die Fenster geschlossen werden. Die Klasse ApplicationWorkbenchWindowAdvisor wird während der einzelnen Workbench-Fenster aufgerufen. Die Methode preWindowsShellClose wird z.B. beim Schließen eines Fensters aufgerufen. Die Klassen ApplicationActionBarAdvisor baut die Aktionsleisten (Menü, Werkzeugleiste etc.) auf.

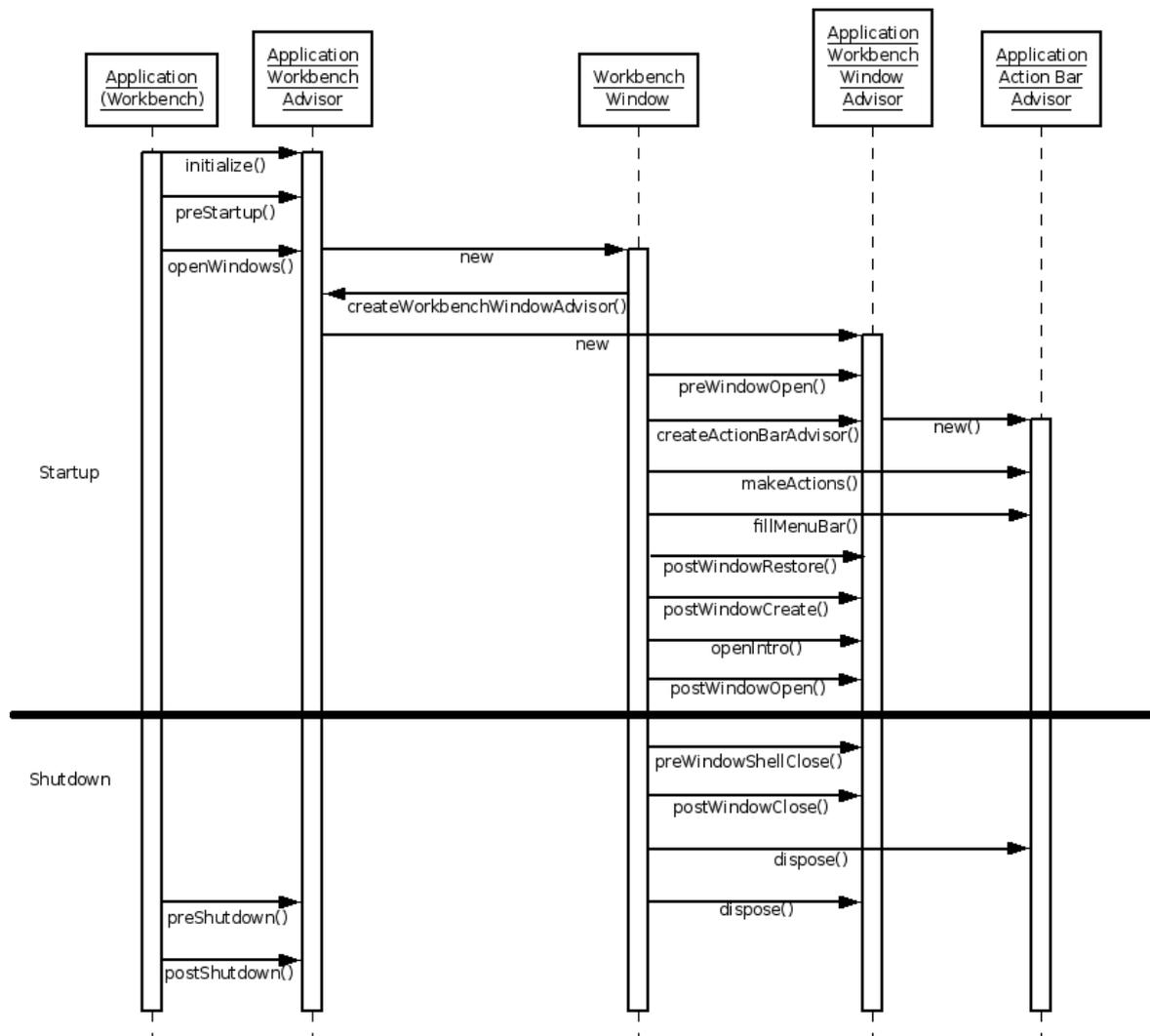


Abbildung 2.7: Workbench Lebenszyklus [JM06]

2.4.2 UI-Architektur

Die Benutzerschnittstellen (die sichtbare Oberfläche, Workbench) basieren auf Java SWT und JFace. Zu einer Display-Instanz wird die Oberfläche aufgebaut, aus Views und Editoren. Die Benutzeraktionen werden mit Hilfe von Actions bearbeitet, welche durch vorgegebene Listener aufgerufen werden.

SWT: SWT ist eine eigene GUI-Bibliothek von Eclipse. SWT nutzt im Gegensatz zu z.B. GUI-Bibliothek SWING grafische Ressourcen des Betriebssystems. SWT besitzt ein Erscheinungsbild (Look & Feel) wie das Betriebssystem, nur wenn ein GUI-Element (Widget) in einem Betriebssystem nicht vorhanden ist, wird es emuliert⁴. Der Benutzer kann deshalb mit einer gewohnten Benutzeroberfläche arbeiten. Durch die Nutzung der Betriebssystemressourcen verlangt SWT weniger Ressourcen als z.B. SWING.

In SWT werden Listener auf ein Widget bezogen. Der Quellcode von Aktionen wie z.B. Copy & Past muss für jedes Widget geschrieben, gewartet und gepflegt werden.

Bei der Anordnung von Widgets auf der Oberfläche können Layout-Manager helfen. Die Klasse GridLayout ist eine vordefinierte Layout-Klasse und ähnlich dem Aufbau einer HTML-Tabelle [DAU08]. Neben GridLayout gibt es noch weitere vier vordefinierte Layout-Klassen: FillLayout (füllt die Oberfläche mit einem GUI-Element aus), RowLayout (wird z.B. eingesetzt bei Werkzeugleisten) und FormLayout (erlaubt die Anordnung auf eine zweidimensionalen Fläche). Es ist auch möglich, eigene Layout-Klassen zu entwerfen.

JFace: JFace nutzt die Ressourcen von SWT und stellt weitere bereit, wie z.B. Wizards. Mit Hilfe von JFace können Listener und deren Aktionen gruppiert werden, es ist nicht mehr notwendig redundanten Quellcode für jedes Widget zu pflegen.

2.5 Zusammenfassung

Zusammenfassend lässt sich feststellen, dass die Betrachtung der Probleme Verteilung, Autonomie und Heterogenität Datenquellen hinsichtlich der Datenintegration beurteilen lassen. Der ETL-Prozess beschreibt den grundlegenden Aufbau der Datenintegration zwischen zwei verschiedenen Systemen. Die Kommunikation mit dem KIS wird mit Hilfe eines KOMMSERV auf eine Datenquelle reduziert. Der KOMMSERV verwendet das am weitesten verbreitete Kommunikationsstandard im klinischen Bereich HL7. Medizinische Geräte, welche kein HL7 unterstützen, können Daten in der Regel mit GDT austauschen. Eclipse kann nicht nur als Entwicklungsplattform genutzt werden, es bietet sich auch als Framework zur Entwicklung als Rich Client an. Die ERCP nutzt das OSGi-Framework und kann auf mehrere Betriebssysteme eingesetzt werden. Durch die Verwendung von SWT, JFace und dem GridLayout lassen sich aus Sicht der Entwickler überschaubare, wartungs- und pflegbare Benutzeroberflächen entwerfen.

⁴ Eine Emulation bezeichnet ein technisches Verfahren zur Nachbildung bestimmter Eigenschaften.

3 Systemanalyse

Dieses Kapitel befasst sich mit der Analyse. Zunächst wird die Ist-Situation und eine Bestandsaufnahme der CI-Zentren dargestellt. Zur Nutzung der Schnittstellen müssen die Gegebenheiten in den Kliniken näher betrachtet werden. Anschließend werden die Anforderungen der Zielgruppen ermittelt und verschiedene Lösungsansätze diskutiert.

3.1 Beschreibung der Ist-Situation/Bestandsaufnahme

Zur Zeit werden die Patienten-, Behandlungs- und Abrechnungsdaten in mehreren Softwareanwendungen verarbeitet. Eine Zusammenfassung der Patienten- und Behandlungsdaten der CI-Zentren ist immer noch die papierbasierte Patientenmappe. Beide CI-Zentren haben deshalb eigene MS Access-DB entwickelt, um die notwendigen Daten der Patienten in einer Anwendung zu bündeln. In Abbildung 3.1 ist die Ansichtsmaske der Patientenstammdaten der MS Access-DB des CIK zu sehen. Beide MS Access-DB werden als Grundlage zur Systementwicklung heran gezogen, eine ausführliche Ist-Beschreibung der DB ist in [AHL10] zu finden.

Zum CI-Zentrum Köln gehören die Benutzergruppen CI-Sekretariat, HNO-Ärzte, CI-Technik und Pädagogik. Die MS Access-DB wird zur Zeit nicht von allen Benutzergruppen benutzt. Ob ein Patient ein CI-Implantat bekommt, wird mit Hilfe von Voruntersuchungen geklärt. Hierzu greift das CI-Zentrum auch auf andere Funktionsbereiche der HNO-Klinik zu. Eine grafische Übersicht der Funktionsbereiche ist in Abbildung 1.1 zu sehen. Neben der HNO-Klinik gibt es einen regelmäßigen Informationsaustausch mit Hörgeräteakustikern, Verwaltung der Uniklinik, Radiologie und CI-Herstellern. Mit der Verwaltung werden Abrechnungsdaten abgeglichen und bei der Radiologie Untersuchungen angefordert. Ein weiterer Informationsaustausch findet durch die Zusammenarbeit mit Instituten, z.B. über Studien, statt.

Bisher wurden Datensätze manuell aus anderen Anwendungen übernommen. Dies mindert die Qualität der vorhandenen Daten, so kommt es z.B. zu Fehlern durch Falscheingaben, fehlenden Daten aufgrund von Zeitmangel der Bediener. Folgende in den Kliniken vorhandenen Datenquellen sind zur Untersuchung als Stellvertreter von Schnittstellen zur Datenintegration ausgewählt:

1. Patientenstammdaten vom KOMMSERV, Protokoll HL7
2. Messdaten des Audiometer AT900, Protokoll GDT

Durch eine Verbindung mit dem KOMMSERV soll das zu entwickelnde Softwaresystem an das KIS angeschlossen werden. Aufgrund der Komplexität von HL7 werden Nachrichten in der ersten Phase der Entwicklung nur lesend verarbeitet und keine HL7-Nachrichten in das KIS gesendet. Dagegen soll mit dem Protokoll GDT eine Kommunikation mit medizinischen Geräten aufgebaut werden. Um die Besonderheiten mit GDT im Detail zu klären wird dies am klinischen Audiometer AT900 analysiert.

Beide MS Access-DBen sind in der vorliegenden Fassung nicht weiter ausbaubar. Eine Neuentwicklung wird von den Beteiligten Entwicklern als weniger aufwendig betrachtet, als die MS Access-DBen an die Anforderungen anzupassen.

Patienten-Stammblatt

Klinik-ID: Aufnahme-Datum OK:

Geburtsdatum:

Nachname: Vorname: Patientenbild:

Stasse: PLZ/Ort: Patiententyp: Bilateral Binodal Studie

Bezugsperson: Muttersprache: Weiblich

Telefon-Nummern:

zuständige Therapeutin:

bekannte Prozessoren des Patienten						
Seite	Hersteller	Typ	Ser.-Nr	Custom	Datum	control
rechts		Freedom				<input type="checkbox"/>
rechts		Nucleus Sprint				<input type="checkbox"/>
*						<input type="checkbox"/>

bekannte Implantate des Patienten					
Seite	OPDat	Implant-Typ	Ort	SerienNr.	Explant
rechts	11.12.2002	Nucleus CI24M	Köln	CI71725	<input type="checkbox"/>
*					<input type="checkbox"/>

genehmigt implantiert, unbeantwortet Post-Nachsorgephase
 beantragt geplant unversorgt
 privat gekürzt
 Klinik (J-Schein) gelöscht

Scans/Schreiben Voruntersuchungen
 Audio-Daten Krankenkasse
 EPA-Daten Systembestellungen

Audietermine: 10 Mon
 3 Mon 24 Mon
 6 Mon 36 Mon
 9 Mon 48 Mon
 12 Mon 60 Mon

Nach den grün hinterlegten Feldern kann gesucht werden!
 Zum Suchen bitte ein Feld "0000000000000000" eingeben!

Abbildung 3.1: Ansicht Patientenstammdaten der MS Access-DB Uniklinikum Köln

3.1.1 Architektur Krankenhausinformationssystem im Universitätsklinikum Köln

Die Speicherung und Verarbeitung der medizinischen Daten geschieht mit AGFA ORBIS und AGFA RIS im Universitätsklinikum Köln. RIS ist ein Radiologie-Informationssystem und ORBIS das PVS. Die Leistungen werden abgerechnet durch SAP als KDMS, die Anbindung von SAP an das KIS geschieht mit GSD-Siemens IS-H med. Der KOMMSERV ist der Integrationsserver e*Gate der Firma SUN. Das LIS ist eine Eigenentwicklung, als AVS wird ein PACS benutzt.

Das KIS in Köln, das in Abbildung 3.2 im 3LGM²-Modell skizziert ist, zeigt auch den Nachrichtenfluss des Uniklinikums. Patientendaten werden in Orbis, Abrechnungsdaten in SAP verwaltet. Das IS-H-System sendet alle 10 Sekunden Nachrichten zum Integrationsserver e*Gate, wo Nachrichten unter anderem in HL7 an registrierte Systeme weiter versendet werden. Orbis sendet Daten direkt an IS-H, empfängt die Nachrichten von SAP aber über e*Gate. Pro Tag werden ca. 8000 HL7-Nachrichten in der Universitätsklinik Köln verarbeitet. Schnittstellen außerhalb des Integrationsservers sind sehr wartungsintensiv, da hier Herstellerupdates oft Funktionalitäten überschreiben.

Im Uniklinikum Köln wird die Patientennummer zusammen mit einer Fallnummer als Identifikation der Behandlung verwendet. Ein Patient kann sich in mehreren Behandlungen befinden, so z.B. CI-Träger sein und unter einer anderen Fallnummer als Notfall in der Ambulanz behandelt werden. Durch die verschiedenen Fallnummern werden diese Behandlungen getrennt.

Es gibt ein Patientennetz (MedLAN-P) und Wissenschaftsnetz (MedLAN-W). Im MedLAN-P liegen die Patientendaten, dieses Netzwerk unterliegt strengen Sicherheitsvorkehrungen. So kann Software z.B. nur durch die Zusammenarbeit mit dem Rechenzentrum installiert werden. Dagegen steht das MedLAN-W für wissenschaftliche Arbeiten zur Verfügung, es gibt keine so strengen Sicherheitsvorkehrungen wie im MedLAN-P. Die Installation eigener Software stellt kein Problem dar, die Rechner sind lokal zugänglich. Die Trennung hat zur Folge, dass die zu entwickelnde Software und der dazu-

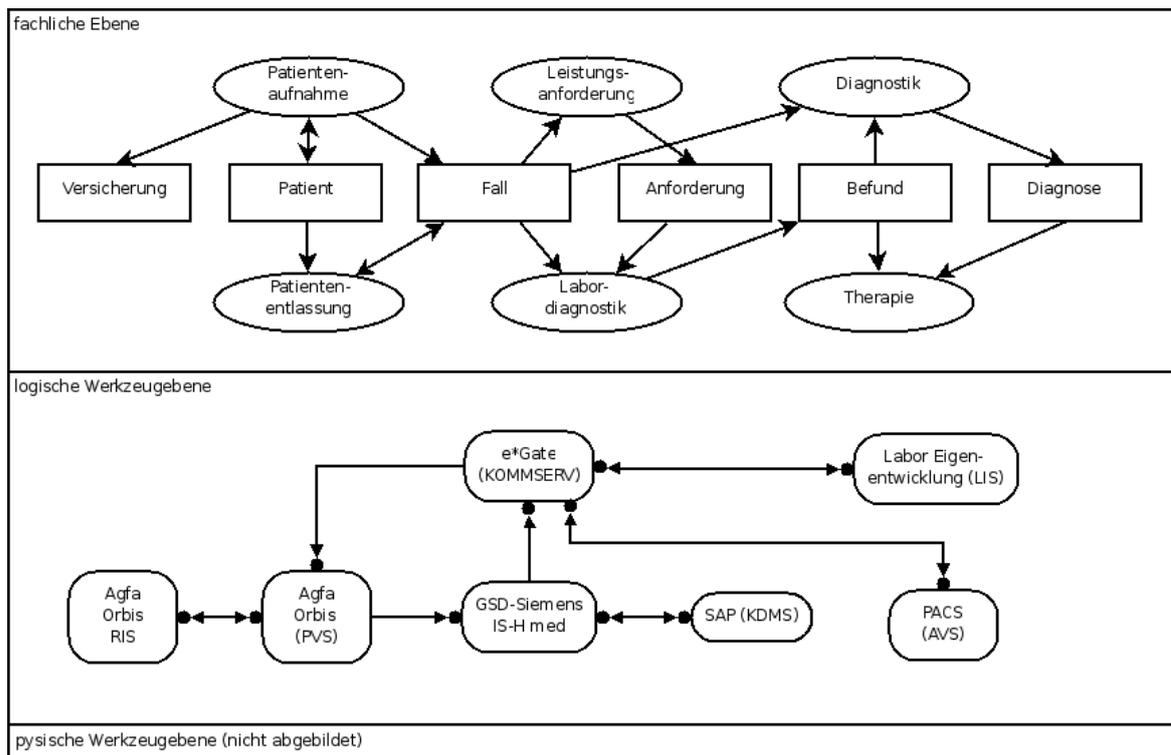


Abbildung 3.2: KIS im 3LGM²-Modell (Universitätsklinikum Köln)

gehörige Server im MedLAN-P zu betreiben sind. Sonst ist eine Anbindung an KOMMSERV oder Audiometer nicht möglich. Der KOMMSERV e*Gate kann die HL7-Nachrichten auf drei Arten zur Verfügung stellen:

1. Datei: Informationen werden in einer Datei angefügt
2. Datei-Schnittstelle: Die Kommunikation erfolgt über Dateien, die in einem bestimmten Verzeichnis abgelegt werden.
3. Programm-Programm Schnittstelle: z.B. Unix-Pipes

Persistente Daten müssen mit den Informationen der anderen Anwendungen in der Uniklinik einheitlich bleiben. Hierzu sollen HL7-Nachrichten vom KOMMSERV z.B. auf Änderungen von Patientendaten hin permanent überwacht werden und bei neuen Informationen in der DB aktualisiert werden.

3.1.2 Klinisches Audiometer AT900

Die klinischen Audiometer AT900 und AT700 unterstützen die Feldkennzeichnung GDT 2.0¹. Das Audiometer AT700 ist neben dem AT900 im CI-Zentrum Essen in Verwendung und eine Variante des AT900 mit weniger Funktionalität. Messdaten müssen langfristig sicher gespeichert werden und übersichtlich dargestellt werden. Dies geschieht zur Zeit nur in Papierform in der jeweiligen Patientenakte und auf dem Audiometriesystem. Es gibt zwei verschiedene Ansätze Daten aus dem

¹ E-Mail vom 16. Dezember mit A. Müller, Auritec Medizindiagnostische Systeme GmbH.

Audiometer zu integrieren. Als Informationssystem nutzt das Audiometer eine Paradox-Datenbank. Bei der Paradox-Datenbank wird SQL als Abfragesprache benutzt. Eine Kommunikation ist mit Hilfe von ODBC möglich. Sollte GDT nicht alle Anforderungen erfüllen können, kann dies gegebenenfalls mit Hilfe der ODBC Schnittstelle als Alternative erfolgen. Im- und Export von Daten können über eine BDT/GDT Schnittstelle, jeweils zur Patientensitzung, durch eine Speicherung der aktuellen Patientensitzung benutzt werden. Bei einer Datenübernahme der gesamten Informationen ist die GDT-Schnittstelle deshalb nicht einsetzbar. Da die Informationen in einer Paradox-Datenbank liegen, wäre hier eine Datenübernahme mit Hilfe der ODBC-Schnittstelle zu empfehlen.

3.2 Ermittlung der Anforderungen

Unter Zuhilfenahme der vorhandenen MS Access-DB und durch Gespräche mit den zukünftigen Benutzergruppen wurden die Anforderungen ermittelt. Aufgrund der Speicherung von Patientendaten müssen zudem auch Anforderungen aus Datenschutz, Datensicherheit usw. beachtet werden. Die zu entwickelnde Software muss durch einen Ethikrat der Kliniken vor dem Einsatz begutachtet und ein Einsatz erlaubt werden.

Die zu entwickelnde Software wird von mehreren Benutzergruppen verwendet. Teil dieser Bachelor-Arbeit ist die Anforderungen der Gruppen „CI-Technik“ und „Audiologie und Pädaudiologie“ zu untersuchen². Die Benutzergruppen wurden ausgewählt, weil beide Benutzergruppen mit Audiometer arbeiten.

3.2.1 Funktionale Anforderungen

Die Funktionalen Anforderungen sind in drei Gruppen aufgeteilt: „allgemeine Funktionen“, „CI-Technik“ und „Audiologie und Pädaudiologie“. Die „allgemeine Funktionen“ stehen „CI-Technik“ und auch „Audiologie und Pädaudiologie“ zur Verfügung. In diesem Abschnitt werden nur für das Verständnis relevante Produktdaten aufgezählt, eine Beschreibung der Datenbankstruktur folgt im Entwurf und das Datenbankschema ist im Anhang A zu finden.

3.2.1.1 Allgemeine Funktionen

Es kommt vor, dass Patienten mit Hilfe einer Bezugsperson kommunizieren. Betroffene Patienten bekommen erst mit einem Implantat ein Hörvermögen und haben bisher keine Lautsprache lernen können. Mit „allgemeine Funktionen“ werden Funktionen zu einer Benutzergruppe zusammen gefasst. Die Benutzergruppe Wissenschaft stehen die Funktionen der „allgemeine Funktionen“ jedoch nicht zur Verfügung, da hier andere datenschutzrechtlichen Anforderungen vorliegen. Es wird ein einheitliches Design der Patientenstammdaten, Verwaltung Bezugspersonen, Bemerkungsfeld und Informationsfelder in dieser Gruppe gebündelt.

Wichtige Felder der Ansicht Patientenstammdaten sind die Patientennummer (Klinik-ID), Fallnummer, Behandlungstage, Bezugsperson und Therapeut, Bemerkungsfeld und der Status der Versorgung. Patientennummer und Fallnummer dienen im Uniklinikum Köln zur Identifikation des Patienten und eines Falles. Die Berechnung der Behandlungstage zum Fall kann nicht in anderen Anwendungen der Uniklinik Köln verarbeitet werden. Bisher wird dies vom Sekretariat mit Hilfe einer Tabellen-

² Herr Ahlering untersucht in seiner IP-Arbeit [AHL10] die Gruppen Sekretariat und Wissenschaft.

kalkulation erledigt, sollte aber nach Möglichkeit von der Benutzergruppe Sekretariat automatisch verarbeitet werden³.

3.2.1.2 CI-Technik

Die „CI-Technik“ berät, bestellt und konfiguriert CIs. Vor einer OP gibt es Vorgespräche, und der Patient entscheidet welches CI er nutzen will. Nach der Implantation wird der Patient technisch betreut, hierzu gehören auch regelmäßige Feineinstellungen. Bei der Erstanpassung wird das CI aktiviert und der Patient kann zum ersten Mal nach der Implantation das Gerät nutzen.

Es werden Daten über die CIs benötigt, wie z.B. CI-Seriennummer, Sprachprozessor, Datum der OP und Datum der Erstanpassung.

3.2.1.3 Audiologie und Pädaudiologie

Die „Audiologie und Pädaudiologie“ stellt die Hörfähigkeit ohne Hörhilfe, mit Hörgerät oder mit CI fest. Hierzu werden subjektive und objektive Hörtests durchgeführt und Ergebnisse gespeichert, z.B. zum Feststellen der Hörfähigkeit vor bzw. nach einer Operation. In bestimmten Zeitabschnitten wird die Hörfähigkeit des Patienten kontrolliert und die Ergebnisse dokumentiert. Hierzu werden Audiometer eingesetzt, welche die Ergebnisse grafisch aufbereiten. Der Vergleich mit vorherigen Ergebnissen ist grundlegend im Einsatz der weiteren Therapie. Diese Gruppe arbeitet nicht nur mit CI-Patienten zusammen.

Es werden Daten, wie die Zeit seit letzter Untersuchung und Ergebnisse der letzten Untersuchung benötigt. Die Ergebnisse der Untersuchungen müssen persistent gespeichert werden.

3.2.2 Nichtfunktionale Anforderungen

Durch die Speicherung und Verarbeitung von Patientendaten sind zum Betrieb des Systems hohe Sicherheitsanforderungen zum Schutz dieser Daten zu gewährleisten. Datensätze dürfen nicht gelöscht werden, statt dessen werden diese als gelöscht markiert. Todesfälle von Patienten müssen als solche erkennbar sein. Zuerst kommen die Qualitätsanforderungen gefolgt von den technischen Anforderungen Pflege und Wartbarkeit.

3.2.2.1 Benutzbarkeit

Neue Mitarbeiter und Vertretungen sollten sich in ihren jeweiligen Fachgebieten mit ihren jeweils zu erwartenden Computerkenntnissen gut einarbeiten können. Hierzu sollte eine Schulung von insgesamt zwei Stunden aller Systemfunktionen ausreichen. Zur Gestaltung der Benutzerschnittstellen sollen die Formulare der MS Access DB vom CI-Zentrum Essen als Grundlage dienen.

3.2.2.2 Zuverlässigkeit

Das Softwaresystem muss in angemessener Zeit Patientendaten liefern können. Es dürfen keine Patientendaten verloren gehen. Sollten Netzwerkprobleme auftauchen, so wäre es wünschenswert, wenn die bisherigen Daten weiterhin abrufbar sind.

³ Wird im Rahmen der IP-Arbeit [AHL10] von Herrn Ahlering untersucht.

Bei einer Unterbrechung der Daten werden die Daten zwischengespeichert. Diese müssen dann synchronisiert werden. Bei Konflikten gibt es eine Meldung und der Benutzer muss die Daten selbst abgleichen. Zum Abgleichen sollte der letzte Benutzer, welcher Änderungen am Datensatz gespeichert hat, mit Namen und Kontaktdaten angezeigt werden.

3.2.2.3 Sicherheit

Anforderungen und das Lesen von Daten sind nur durch eine Authentifizierung nutzbar. Verbindungen zu anderen Rechnern dürfen nur verschlüsselt erfolgen.

Als Sicherung gegen Datenverlust durch Hard- oder Softwarefehler müssen Serverseitiger Datenbank-Dump, Synchronisation des Datenbank-Dump auf externen Servern möglich sein. Dabei soll es tägliche, wöchentliche und monatliche Backups geben. Auf der Seite der Clients wird der Verlust bei Fehlern von der Arbeitssitzung als mögliches Risiko akzeptiert.

3.2.2.4 Flexibilität

Die Software benötigt für den laufenden Betrieb eine gewisse Flexibilität. Formulare sollen die Möglichkeit enthalten, Zusatzinformationen über hinzufügbare Felder vergeben zu können. Die Kliniken arbeiten mit unterschiedlichen oder abgeänderten Tests, wobei neue Tests oder Klinik interne Erweiterungen hinzu kommen. Im Umfeld einer Klinik werden Studien durchgeführt, interne und auch von außen beauftragte.

1. Zusatzfelder mit Bezeichnung, Einheit (% , Zahl, dB, Ja oder Nein und Auswahlfeld) und Inhaltsfeld
2. Individuelle oder neue Tests
3. Aufnahme Studien

3.2.2.5 Übertragbarkeit

Die Daten sollen über Schnittstellen zur weiteren Verarbeitung in gängigen Formaten exportiert werden können. Die Formate sind csv und xml.

3.2.2.6 Datenschutz

Folgende Datengruppen werden innerhalb der Software verwaltet:

1. Patientendaten
2. Verwaltungsdaten
3. Wissenschaftliche Daten
4. Klinische Daten

Es muss gewährleistet sein, dass die Patientendaten nicht in die Hände Dritter geraten können, auch nicht der Verwaltung, was rein klinische Ergebnisse anbelangt und auch nicht von anderen CI-Zentren und hier auch nicht in anonymisierter Form.

Lokale Daten müssen sicher gespeichert werden, z.B. durch eine einfache Kopie der Datenbank darf kein Zugriff auf Patientendaten möglich sein.

3.2.2.7 Änderbarkeit (Pflege)

Der Einsatz in verschiedenen Universitätskliniken und Kooperationen mit externen Dritter sollte eine geeignete Software-Architektur aufweisen, bei der sich Änderungen individuell zuschneiden lassen. Änderungen im Lebenszyklus der Software sind auf jeden Fall durch neue Geräte und neue Anforderungen oder durch die geforderten Flexibilität vorhersehbar.

3.2.2.8 Wartbarkeit

Der Einsatz von verschiedenen Betriebssystemen, Rechnerarchitekturen und wechselnder Einsatz von Hardware (z.B. Einlesegeräte) erfordert von vornherein eine Planung der Wartung des zu entwickelnden Softwaresystems.

3.2.2.9 Finanzielle Mittel

Zur Entwicklung des Systems und für den Betrieb sind die finanziellen Mittel begrenzt. Die erste Version der Software kann daher nicht alle Anforderungen erfüllen, sollte jedoch in der Lage sein, die vorhandenen MS Access DBen abzulösen. Die Wartungskosten im Monat müssen für ein CI-Zentrum überschaubar sein. Die offenen Anforderungen sollten durch neue Versionen oder wenn möglich, innerhalb der Pflege entwickelt werden. Den offenen Anforderungen sollten von den CI-Zentren Prioritäten zugeordnet werden und dementsprechend von Entwicklern beachtet werden.

3.3 Untersuchung verschiedener Lösungsansätze

Nach der Betrachtung der Umwelt und der Auflistung der funktionalen und nichtfunktionalen Anforderungen werden nun mögliche Lösungsansätze gegenüber gestellt. Da die Software von mehreren Benutzergruppen gleichzeitig bedient werden muss, werden zuerst die Möglichkeiten der verteilten Anwendungen beschrieben. Daran anschließend werden wichtige Eigenschaften von Thin Clients und Rich Clients miteinander verglichen. Die Entscheidung auf welchem Datenmodell die Software basiert, hat langfristige Auswirkungen auf die Softwareentwicklung. Die Entwicklung einer Umgebung zur Verarbeitung von HL7-Nachrichten ist durch die Komplexität von HL7 aufwendig, eine Alternative zur Eigenentwicklung besteht in der Nutzung von fertiger Software.

3.3.1 Verteilte Anwendungen

Das zu entwickelnde System wird von vielen Benutzern über verschiedene Rechnern benutzt, deshalb wird ein Client benötigt. Ein Client sollte keine Abhängigkeiten zu anderen Clients besitzen, es ist nicht gewährleistet, dass ein Zusammenschluss von Clients zu einem Netzwerk immer erreichbar ist. Deshalb kann das Datenmodell nicht auf Seiten der Clients verteilt werden. Das Datenmodell muss auf einem Server liegen.

Die Verarbeitung von HL7 Nachrichten kann nur serverseitig verarbeitet werden. Eine clientseitige Verarbeitung von HL7 Nachrichten ist nicht sinnvoll, so ist z.B. die Zustellung zum Client eine Belastung innerhalb des KIS, da ein Client nicht immer zur Verfügung steht und die Anzahl der verwendeten Clients Schwankungen unterliegen. Es reicht zudem aus, HL7 Nachrichten von einem Empfänger

zu verarbeiten und über der Datenbank (DB) allen Clients aktualisierte Informationen zugänglich zu machen. Folgende verteilte Anwendungsarchitekturen stehen zur Auswahl:

1. Client-Server-Architektur
2. 3-Schichten-Architektur (3-Tier-Architektur)
3. Mehrschichten-Architektur (N-Tier-Architektur)

Bei der Client-Server-Architektur dient ein DBMS als Server. Die Präsentationsschicht und Geschäftslogik ist auf dem Client implementiert, welcher direkt auf das DBMS zugreift. Hierdurch gibt es eine hohe Zahl von gleichzeitigen Verbindungen, die Anzahl der Verbindungen zu einem DBMS ist begrenzt und der Aufbau der Verbindung ressourcenaufwendig. Durch den direkten Zugriff muss die Firewall auf das JDBC Protokoll hin freigeschaltet werden. Bei der 3-Schichten-Architektur befindet sich die Geschäftslogik auf einem eigenen Tier, dem Application-Server. Der Client enthält die Präsentationsschicht und Routinen zur Eingabevalidierung. Die Geschäftslogik wird über Services angesprochen. Der Schutz des DBMS und Geschäftslogik wird hierdurch deutlich erhöht [REI09], es findet kein Zugriff außerhalb des Servers statt. Die Anzahl der Verbindungen zur DBMS ist steuerbar, Connection Pooling ist nutzbar. Problematisch kann die Validierung auf Seiten des Clients und des Application-Servers sein, da mehrfache Validierungen an unterschiedlichen Stellen fehleranfällig und damit wartungsintensiv sind.

Client-Server-Architektur	
Vorteile	Nachteile
Lazy Loading kann genutzt werden	direkter Zugriff auf die Datenbank
Kein Application-Server notwendig	kein Connection Pooling möglich
	Tunneling der Firewall erforderlich
3-Tier-Architektur	
Vorteile	Nachteile
Saubere Bildung von Schichten	Lazy Loading nicht im Client möglich
Anbindung unterschiedlicher Clienttypen möglich (z.B. RCP, Mobile Device, Webapplikation)	Datenvalidierung muss gegebenenfalls doppelt implementiert werden
Connection Pooling kann genutzt werden	
Tunneling der Firewall nicht erforderlich	
Definition sauberer Interfaces möglich	
N-Tier-Architektur	
Vorteile	Nachteile
Kapselung des Zugriffs auf bestehende Systeme möglich	Latenzzeiten durch verteilte Kommunikation
Geschäftslogik wird zentral	

Tabelle 3.1: Vor- und Nachteile Verteilter Anwendungen [SR09]

Bei der Mehrschichten-Architektur kann auch auf andere Systeme zugegriffen oder Services zur Verfügung gestellt werden. Bei der Kommunikation mehrerer Systeme untereinander kann der Application-Server die Datenmodelle harmonisieren und Nachrichten durchstellen.

In Tabelle 3.1 sind die Vor- und Nachteile der verteilten Anwendungen aufgeführt. Für das zu entwickelnde System wird ein zentraler Ort für das Datenmodell, z.B. ein DBMS, benötigt. Des Weiteren wird auf mehrere Systeme zugegriffen, auf medizinische Geräte und das KIS.

Zur Ablösung der MS Access-DB reicht eine Client-Server-Architektur. Durch die Anforderungen der Datenintegration (KOMMSERV, medizinischen Geräte) wird eine N-Tier-Architektur entwickelt werden müssen. Es ist zu erwarten, dass langfristig mehr Systeme hinzu kommen. Der Zugriff einer größeren Benutzeranzahl ist mit einem Application-Server unter Einsatz von Connection Pooling steuerbar.

3.3.2 Thin Client vs. Rich Client

Die Ablösung der MS Access-DB kann durch folgende Client-Architekturen erfolgen:

1. Thin Client, z.B. eine Webapplikation auf einem Webserver
2. Rich Client, z.B. Eclipse Rich Client Plattform

Bei einem Thin Client ist die funktionale Ausstattung auf die Ein- und Ausgabe einer Anwendung oder einen Rechner als Endgerät beschränkt. Der Thin Client braucht eine zuverlässige Netzwerkverbindung zum Server. Ein Vertreter einer Thin Client Anwendung ist ein Browser zum Aufruf von Webapplikationen. Eine Webapplikation ist ein Programm, das auf einem Webserver ausgeführt wird, das Datenmodell und die Geschäftslogik werden auf dem Server verarbeitet. Benutzer greifen mit Hilfe des Browsers auf die Webapplikation zu, dabei werden Daten in Sessions (serverseitig) und Cookies (clientseitig) gespeichert. Dies ist notwendig, da der Aufruf mit Einsatz vom HTTP-Protokoll geschieht und HTTP zustandslos ist. Zustandslos bedeutet hierbei, dass der Webapplikation nicht bekannt ist, ob es sich um ein Erstaufruf oder das Fortsetzen einer Interaktion mit dem Benutzer ist. Mit dem Einsatz von Session, Cookie und versteckten Feldern werden Zustände und Daten zwischen gespeichert. Es ist jedoch bei jedem Aufruf der Webapplikation das erneute Auslesen dieser Datenquellen und deren Verarbeitung notwendig. Mit einer Webapplikation kann nicht an das Verhalten von klassischen Softwareprodukten vollständig umgesetzt werden. Softwareeigenschaften werden zwar zunehmend bei einer Webapplikation mit Hilfe von z.B. JavaScript (onMouseOver etc.) simuliert, jedoch ist ein vollständiger Ersatz wie Listener unter SWT/JFace zur Zeit noch nicht möglich. So wird z.B. bei einer Veränderungen eines Datenfeldes eine Intervallaktualisierung zum Server geschickt, gegen das sofortige Aufrufen eines Listeners ist dies jedoch nur eine Annäherung mit erhöhter Belastung des Webserver.

Bei dem Rich Client wird die Verarbeitung der Daten vor Ort erledigt, ein Server dient hier zur Bereitstellung weiterer Dienste und zur Nutzung eines persistenten Datenmodells. Kommt es beim Rich Client zu Netzwerkproblemen, können Daten bei einer erneuten Verbindung synchronisiert werden. Es kann bei Netzwerkproblemen oder Ausfall des Server im Gegensatz zum Thin Client weiter gearbeitet werden. Der Rich Client ist meist ein Framework und basiert auf einen Fat Client. Rich Client Frameworks sind durch Module und Plug-ins erweiterbar und bieten reichhaltige Module. So gibt es oft bei Rich Client Frameworks Updatemechanismen, Hilfebereiche und eine große Auswahl an fertigen Plug-ins. Bei den Benutzeroberflächen werden Verarbeitungen z.B. bei SWT/JFace mit Listener angestoßen. Validierung, Verifizierung und Datenverarbeitung können auf dem Client bearbeitet werden. Die ERCP hat durch SWT/JFace ein Look&Feel wie das vorhandene Betriebssystem, es werden die nativen Elemente des Betriebssystem genutzt.

3.3.3 Datenhaltung

Aufgrund der vorhandenen und zu erwartenden Datenmenge im GB-Bereich kommt zur Zeit nur ein Datenbankmanagementsystem (DBMS) in Frage. Eine DB und ein DBMS bilden zusammen ein Datenbanksystem. Der Einsatz einer DB unterscheidet sich von der traditionellen Dateiverarbeitung durch [SBN05]:

- Existenz eines Katalogs
- Programm/Daten- und Programm/Operation-Unabhängigkeit
- Datenabstraktion
- Unterstützung mehrerer Benutzersichten (Views)
- Gemeinsame Nutzung von Daten durch mehrere Transaktionen
- Bessere Möglichkeiten für die Umsetzung von Standards
- Kürzere Anwendungsentwicklungszeiten
- Höhere Flexibilität
- Größere Verfügbarkeit aktueller Daten für alle Benutzer
- Erhöhte Wirtschaftlichkeit

DBMS werden in der Verwendung des zugrunde liegenden Datenmodells unterschieden. Aktuelle Vertreter von verwendeten Datenmodelle in DBMS sind:

1. Relationale Datenmodell
2. Semistruktuierte Datenmodell
3. Objektdatenmodell
4. Objektrelationale Datenmodell

Die Grundlages des Relationalen Datenmodells ist die mathematische Relation. Operationen auf das Relationale Datenmodell wird durch die Relationale Algebra bestimmt, welche als theoretische Grundlage der Abfragesprache SQL dient. Das Relationale Datenmodell gilt allgemein als ausgereift und stabil.

Ein Vertreter der Semistruktuierten Datenmodelle sind die XML-DB. XML besitzt einige Möglichkeiten, wie z.B. eine Reihenfolge der Datensätze, die im Relationalen Datenmodell nur umständlich modelliert werden können⁴. Eine bekannte Abfragesprache der XML-DBen ist XPath.

Das Objektdatenmodell nutzt den Aufbau der Objektorientierten Programmiersprachen wie Eigenschaften und Methoden auf Objekte, welche in Klassen zusammengefasst werden. Erweitert ein relationales DBMS das Modell um die Objektorientierung, handelt es sich um ein objektrelationales Datenmodell.

Einige relationale DBMS können mit einem XML-Datentyp umgehen. Auf diese Weise können Vorteile von XML, wie z.B. das Speichern von Attributen in Baumstrukturen, auch im relationalen Datenmodell verwendet werden. Relationale DBMS gelten allgemein als ausgereift, stabil und performant, zudem sollten die meisten Entwickler Erfahrungen mit dem Umgang einer Relationalen DB besitzen.

In Abbildung 3.3 ist der Aufbau des relationalen Datenmodells abgebildet. An dieser Stelle steht die

⁴ Eine Reihenfolge von Datensätzen im Relationalen Datenmodell kann z.B. mit Hilfe eines Zeitstempels pro Datensatz hergestellt werden.

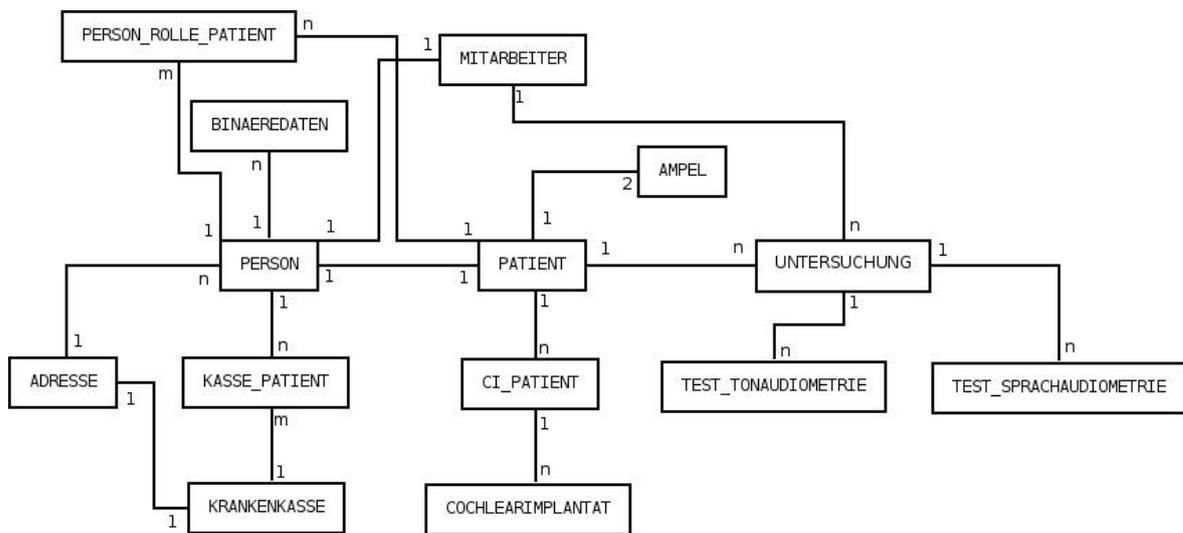


Abbildung 3.3: Datenbankschema ohne Attribute

Person im Mittelpunkt, zu jeder Person können Binärdaten gespeichert werden. Eine Person kann entweder Mitarbeiter, Patient oder eine Bezugsperson sein. Dieselbe Adresse können mehrere Personen referenzieren, z.B. bei Familienmitgliedern. Patienten können mehrere CIs zugeordnet sein oder kein CI-Träger sein. Dies ist u.a. bei der Voruntersuchung der Fall. Zu einer Untersuchung muss es immer einen Patienten und einen verantwortlichen Mitarbeiter geben. Die Untersuchungen zum Patienten sammeln sich in der UNTERSUCHUNG, welche dann auf die jeweiligen Untersuchungsergebnisse referenziert. Zu einem Patienten gehört eine Ampel, die den Mitarbeitern den Status des Patienten auf einen Blick ermöglicht. Die Ampel wird aber auch zu statistischen Zwecken und zur Identifikation von einzuladenden Patienten genutzt. Wenn Patienten z.B. länger als ein Jahr nicht zur CI-Kontrolle kamen, soll diese Gruppe von Patienten zum Zwecke einer Einladung aufgelistet werden. Das Schema ist erweiterbar, neben der PERSON ist UNTERSUCHUNG ausbaubar. In den CI-Zentren existieren viele verschiedene Untersuchungsarten und dementsprechende Untersuchungsergebnisse. Diese sind Grundlage einer digitalen Patientenakte. Die UNTERSUCHUNG muss um weitere Untersuchungsarten ergänzt werden können.

3.3.4 Programmiersprache

Das zu entwickelnde System sollte auf den gängigen Betriebssystemen⁵ lauffähig sein. Die zwei häufigsten Vertreter für betriebssystem unabhängige Entwicklung ist die Umsetzung als Webapplikation mit PHP oder die Nutzung einer plattformunabhängigen Programmiersprache wie z.B. Java. PHP ist eine serverbasierte Skriptsprache und seit der fünften Version auch objektorientiert. Die Benutzeroberflächen werden mit XHTML/CSS für den Browser aufbereitet. Anbindungen an DB wie MySQL und PostgreSQL sind vorhanden, die Verarbeitung von XML kann SAX und DOM unterstützt erfolgen, PHP hat auch eigene Entwicklungen wie SimpleXML. SimpleXML verarbeitet XML-Dokumente als Arrays, jedoch schränkt dies auch die Nutzung ein. Im Datenmodell XML ist es kein Problem, mehrere identische Werte zu besitzen. Wird diese Eigenschaft auf Arraykeys übertragen, gehen Infor-

⁵ Microsoft Windows, Mac OS/X und GNU/Linux

mationen verloren, da es bei Arrays jeweils nur einen Key geben kann.

Java-Programme werden in einen plattformunabhängigen Bytecode überführt und von einem Java-Runtime-Environment (JRE) interpretiert und ausgeführt. Für das jeweilige Betriebssystem muss ein JRE zur Verfügung stehen und die genutzten Bibliotheken sollten ebenfalls plattformunabhängig sein. Wo dies nicht der Fall sein kann, werden die von der Bibliothek bereit gestellten Methoden nicht verwendet werden können.

Benutzeroberflächen können mit SWING oder SWT aufgebaut werden. Beim Look&Feel ist SWT am jeweiligen Betriebssystem angelehnt, da die nativen Elemente der Betriebssysteme genutzt werden. Dagegen sind die Widgets bei SWING in Java geschrieben und es werden keine nativen Elemente des Betriebssystem benutzt. Anbindungen an den gängigen DBMS stellt auch für Java kein Problem dar.

3.3.5 Einsatz Open-Source-Software

Eine Eigenentwicklung zur Verarbeitung von HL7 ist sehr aufwendig. Deshalb wird überprüft, ob es möglich ist, hier Software aus dem Open-Source Bereich zu nutzen. Vorteile der Open-Source Software sind:

- vor Einsatz kann das Produkt ohne Kosten überprüft werden,
- keine Lizenzkosten oder andere Anschaffungskosten,
- Eigenentwicklung innerhalb des Produkts durch offenen Code möglich,
- keine Abhängigkeit von einem Hersteller.

Dagegen sind die Nachteile durch den Einsatz einer Open-Source Software:

- kein Support,
- keine Garantien für lange Laufzeiten, da ein Vertrag hier rüber nicht vorhanden ist,
- keine Gewährleistungen,
- keine garantierten Ansprechpartner,
- hohe Veränderungsrate in den APIs.

Bei Sourceforge.net [SOU09] werden 58 Open-Source Tools unter dem Suchbegriff HL7 aufgelistet. Davon sind acht mit der Programmiersprache Java geschrieben. Nach der Beschreibung werden folgende Projekte betrachtet:

1. HAPI: ein Parser von HL7 2.x Nachrichten
2. jL7: eine HL7 Bibliothek
3. HL7 Java API: eine HL7 Bibliothek

Das Open-Source Tool HAPI (1) ist mit zwei auswählbaren Lizenzen ausgestattet und darf kommerziell, wie auch in Open-Source-Produkten, verwendet werden. Da HAPI ein Parser ist, kann dieser in einen ETL-Prozess integriert werden, die Transformation bleibt Aufgabe in der eigenen Entwicklung. Es kann damit gerechnet werden, dass ein Parser in zukünftigen Versionen, im Gegensatz zu den APIs, weniger Änderungen unterliegt.

3.4 Zusammenfassung

Durch die vorhandenen MS Access-DBen, einem KOMMSERV im KIS und die notwendige Einbindung eines Audiometer ist das Umfeld des neuen Systems festgelegt. Durch das Hinzufügen der Gruppe allgemeine Benutzergruppe werden globale Funktionen gebündelt. Anforderungen wurden getrennt nach funktionalen und nichtfunktionalen aufgelistet. Die Untersuchung von verschiedenen Lösungsansätze bietet für das nachfolgende Kapitel die Basis der Entwurfsentscheidungen. Dabei wurde auch ein Datenbankschema vorgestellt, die Datenbank verwaltet Patienten und Ergebnisse aus Untersuchungen.

4 Entwurf und Implementierung

Im Entwurf werden grundlegende Entscheidungen getroffen und bei der Implementierung mit Hilfe von Prototypen untersucht und umgesetzt.

4.1 Entwurf

Die MS Access-DBen werden durch ein neues System abgelöst. Hinzu kommen neue Anforderungen an das zu erstellende System, wie z.B. Daten aus anderen Softwaresystemen integrieren zu können. Zur Umsetzung existieren verschiedene Lösungsansätze, in diesem Abschnitt wird der Entwurf des neuen Systems vorgestellt. Der vorgestellte Entwurf und auch die darauf folgende Implementierung ist als Prototyp zu verstehen, welcher in weiteren Arbeiten erweitert und verfeinert werden kann. Die grundlegenden Bestandteile, wie z.B. das Datenbankschema und die Gestaltung der Benutzeroberflächen, sollten keinen gravierenden Änderungen mehr unterliegen.

4.1.1 Architektur

Das System sollte als N-Tier-Architektur, wie in Abbildung 4.1 dargestellt, fertig gestellt werden. Die Anbindung an den KOMMSERV des KIS muss auf dem Server verarbeitet werden. Die Entwicklung einer Webapplikation zur weiteren Anbindung von Benutzergruppen oder Thin Clients ist langfristig nicht auszuschließen und kann in der N-Tier-Architektur hinzugefügt werden. In den CI-Zentren wird die Entwicklung einer Rich Client Plattform bevorzugt, der Prototyp wird mit der ERCP umgesetzt. Zu der Geschäftslogik auf dem Client gehört die Fähigkeit Daten nach Netzwerkproblemen zu synchronisieren. Dies ist z.B. eine Anforderung an das neue System, welche nicht mit einer Webapplikation umgesetzt werden kann. Ein Application-Server sollte zum Connection Pooling verwendet werden, eine Umsetzung ist jedoch kein Bestandteil des Prototypen. Hierdurch kann die N-Tier-Architektur umgesetzt werden und bei späteren Erweiterungen muss der Rich Client nicht angepasst werden. Eine Möglichkeit zur Umsetzung des Application-Server bietet das Framework Spring.

Die Datenintegration wird mit eigenen Systemen umgesetzt, jeweils für HL7 und GDT in Java. Die Steuerung der ERCP findet über die Benutzeroberflächen statt, welche nativ aufgebaut sind. Der Benutzer bekommt hierdurch das Erscheinungsbild des benutzten Betriebssystems.

4.1.2 Datenbank

Als Datenmodell wird ein relationales DBMS mit PostgreSQL ausgewählt. Das DBMS ist leistungsfähig und als Open Source Datenbank fallen keine Lizenzkosten an. Im Gegensatz zu MySQL liegt der Einsatzschwerpunkt bei einer PostgreSQL Datenbank im GB-Bereich, wogegen MySQL spezialisiert auf das Ausliefern von kleinen Datenmengen ist. Deshalb wird das DBMS MySQL überwiegend bei Webapplikationen eingesetzt. Die maximale Datenbankgröße bei PostgreSQL ist nicht limitiert, pro Tabelle liegt diese bei 32 TB, wobei einer Zeile 1,6 TB und ein einzelnes Feld 1 GB Speicher belegen kann [POS09]. Die PostgreSQL-DB ist der zentrale Speicherort der Daten, auch für binäre Bilder der Patienten. Da PostgreSQL den Datentyp XML unterstützt, können die Vorteile vom XML-Datenmodell in ausgewählte Szenarien (z.B. Bemerkungsfelder) verwendet werden.

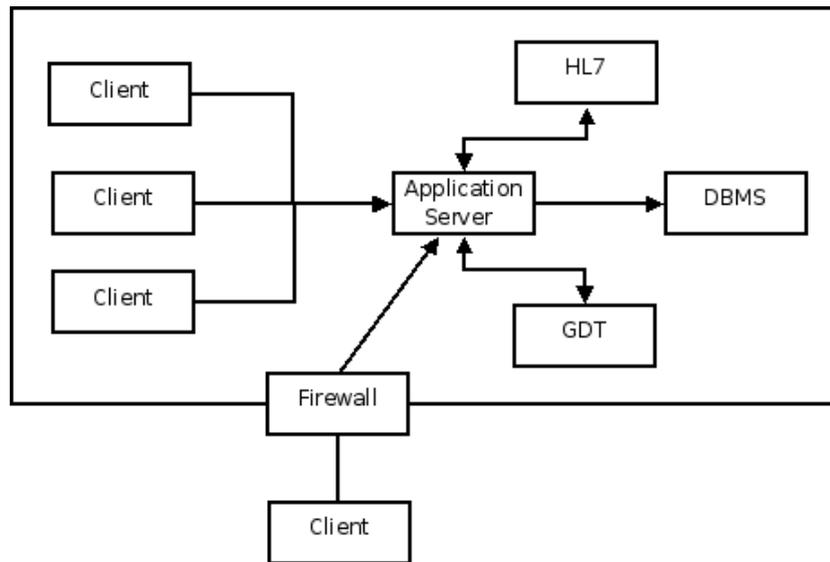


Abbildung 4.1: System als N-Tier-Architektur

In Abbildung 3.3 ist das Datenbankschema skizziert und im Abschnitt 3.3.3 sind Details zum Aufbau beschrieben. Die Attribute und einzelne Entitäten, Relationen, Ableitungen und Business Roles sind im Anhang A durch Tabellen beschrieben.

4.1.3 HL7

Wie schon in den Grundlagen (3.1) erwähnt, gibt es drei Möglichkeiten HL7-Nachrichten auf dem Server entgegen zu nehmen:

1. Datei: Informationen werden in einer Datei angefügt
2. Datei-Schnittstelle: Die Kommunikation erfolgt über Dateien, die in einem bestimmten Verzeichnis abgelegt werden.
3. Programm-Programm Schnittstelle: z.B. Unix-Pipes

Sollte es Probleme bei der Verarbeitung der Möglichkeiten Datei (1) oder Datei-Schnittstelle (2) kommen, wird die Datei umfangreich oder im Verzeichnis sammeln sich Dateien an. Beide Situationen sind mit serverseitigen Skripten gut zu überwachen. Da die Nachrichten im Fehlerfall noch vorhanden sind, können diese nach Behebung des Fehlers weiter verarbeitet werden. Treten dagegen Fehler bei der Programm-Programm Schnittstelle (3) auf, werden Fehler durch den KOMMSERV gemeldet. Das Wartungs-Team des KOMMSERV muss bei Fehlern das Software-Wartungsteam der Anwendung benachrichtigen. Die Varianten eins und zwei sind deshalb der dritten Variante vorzuziehen, da hier das Software-Wartungsteam ohne Mitarbeit des Rechenzentrums Fehler erkennen und beheben kann.

In Abbildung 4.2 ist der Prozess zur Verarbeitung der HL7-Nachrichten mittels der Datei-Schnittstelle (2) dargestellt. Das Verzeichnis der Datei-Schnittstelle (2) wird in Intervallen auf Dateien hin überprüft. Sind Dateien im Verzeichnis vorhanden, werden diese zur weiteren Verarbeitung im FIFO-Verfahren verschoben und die Daten extrahiert. Zur Extraktion der HL7-Nachrichten wird der Open-

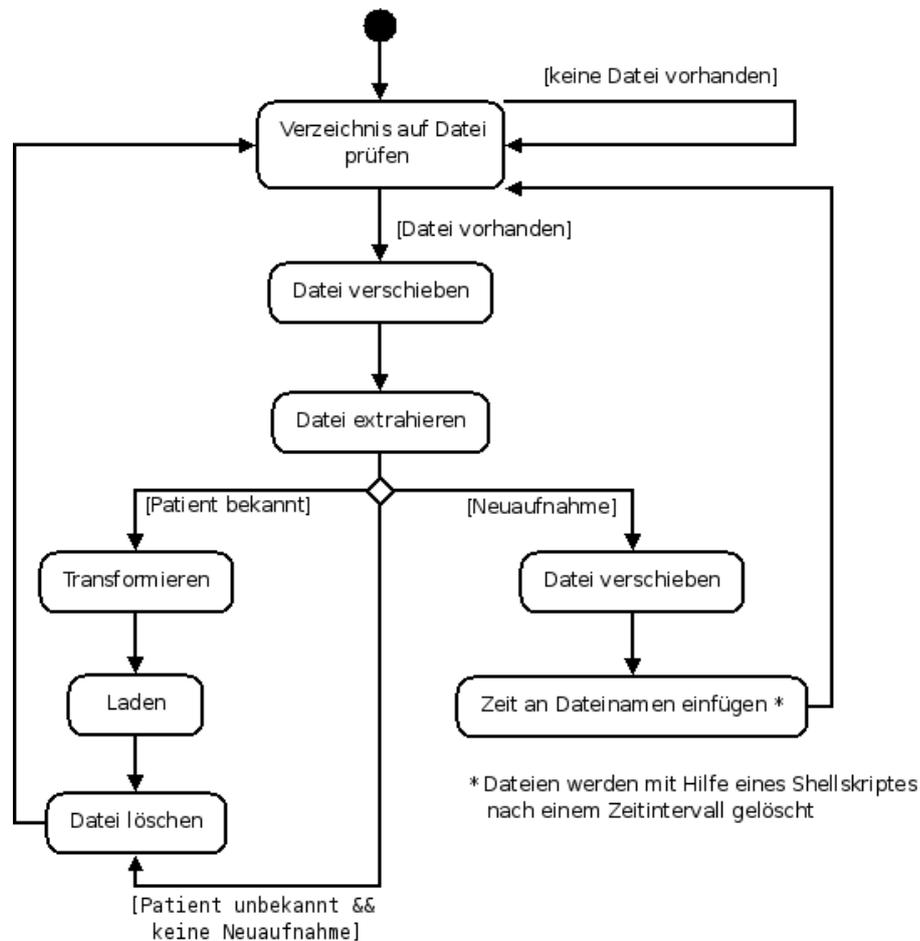


Abbildung 4.2: ETL-Prozess mit HL7-Nachrichten

Source Parser HAPI benutzt. Mit Hilfe dieses Parsers werden die HL7-Nachrichten eingelesen (extrahiert), bewertet und weiter verarbeitet. Ist der Patient bekannt, werden die Daten transformiert und geladen. Anschließend wird die Datei gelöscht und das Verzeichnis wird erneut auf Dateien hin geprüft. Ist der Patient unbekannt und handelt es sich um eine Neuaufnahme, wird die Datei in ein dafür vorgesehenes Verzeichnis verschoben und mit einem Zeitstempel versehen. Ein Shellskript löscht regelmäßig ältere Dateien, die Zeitdauer (z.B. 48 Stunden) kann im Shellskript festgelegt werden. Kommt nun der Patient zum Sekretariat des CI-Zentrums, werden neue Patienten zuerst in diesem Verzeichnis gesucht. Sollten Daten zum neuen Patienten vorliegen, werden diese transformiert und geladen (load). Ist der Patient unbekannt und handelt es sich um keine Neuaufnahme, wird die Datei gelöscht. Die HL7-Nachricht wird somit als unwichtig verworfen. Im Rahmen dieser Arbeit ist es nicht möglich, die Schnittstelle HL7 vollständig zu implementieren. Deshalb gibt es im Prototypen nur die Möglichkeit HL7-Nachrichten zu lesen. Sollten Fehler bei der Verarbeitung auftauchen, werden diese in einer Logdatei gespeichert und das Wartungsteam mit einer E-Mail benachrichtigt.

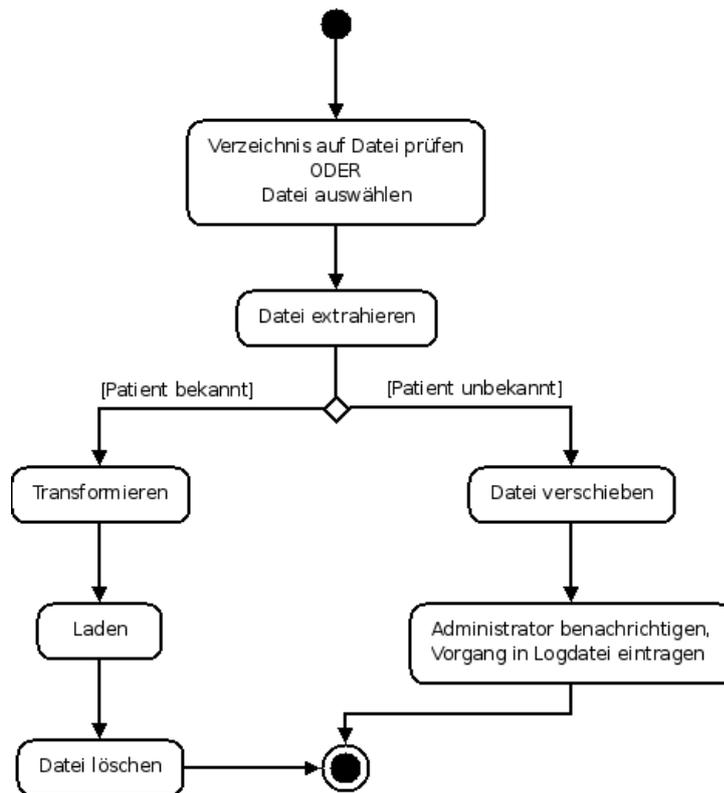


Abbildung 4.3: ETL-Prozess mit GDT-Nachrichten

4.1.4 Klinisches Audiometer AT900

Bei den GDT-Informationen ist von einem bekannten Patienten auszugehen. Sollte dennoch der Fall auftauchen, dass ein Patient nicht zugeordnet werden kann, muss der Administrator die Entscheidung der weiteren Verarbeitung treffen. Auch die Kommunikation mit GDT erfolgt über Verzeichnisse. Diese macht eine Überwachung einfacher und bei Störungen reicht das Eingreifen des Software-Wartungsteams. Die Datenintegration sollte auf zwei Arten angeboten werden:

1. Überwachung eines Verzeichnisses pro Audiometer
2. Einlesen einer ausgewählten GDT-Datei

Bei der Überwachung eines Verzeichnisses pro Audiometer (1) wird der Prozess als ein eigenes Tier in der N-Tier-Architektur eingebettet. Dagegen wird eine GDT-Datei (2) als ein Plug-in entwickelt und z.B. die Nutzung über das Menü mit einem Wizard angeboten. Bei beiden Prozessen ist der Ablauf derselbe und in Abbildung 4.3 dargestellt. Sollte ein Verzeichnis (1) überwacht werden, ist das Tier regelmäßig aufzurufen. Ist eine Datei vorhanden, wird diese extrahiert. Wenn der Patient bekannt ist, werden die Daten transformiert und in die DB geladen. Sollte der Patient nicht bekannt sein, muss ein Fehler vorliegen. In diesem Fall wird die Datei in ein eigenes Verzeichnis verschoben und das Software-Wartungsteam benachrichtigt.

4.1.5 Rollen/Aufgaben

Die Anwendung wird von mehreren Benutzergruppen verwendet, welche in unterschiedlichen Disziplinen tätig sind. Hierzu muss eine Nutzer- und Rechteverwaltung entworfen und implementiert werden. Ein Benutzer kann mehreren Benutzergruppen zugeordnet werden.

Die Anmeldung erfolgt beim Starten der Anwendung, wobei die Gültigkeit der Zugangsdaten überprüft wird. Der Benutzer wird dann in einen allgemeinen Benutzerbereich geleitet, von welchem aus er Oberflächen nach Berechtigungen aufrufen kann. Beim Beenden werden die aktuellen Einstellungen gespeichert und beim erneuten Starten wieder hergestellt. Die allgemeinen Funktionen stehen allen Mitarbeitern zur Verfügung.

Die im Prototypen betrachteten Benutzergruppen sind „CI-Technik“ und „Audiologie und Pädaudiologie“. Die „Allgemeinen Funktionen“ können zudem als eine eigene Benutzergruppe betrachtet werden, die allen Mitarbeitern zur Verfügung steht. Die Benutzergruppen werden im Folgenden mit den ihnen vorgestellten Funktionen und einer eigenen Seitenstruktur übersichtlich dargestellt.

4.1.5.1 Allgemeine Funktionen

Die Ansicht der „Allgemeinen Funktionen“ (Abb. 4.4) steht jedem Mitarbeiter zur Verfügung. Sind keine Einstellungen aus vorherigen Sitzungen gespeichert, wird als Standard die Sicht der „Allgemeinen Funktionen“ aufgerufen. Im Wesentlichen werden hier die Patientenstammdaten verwaltet und allgemeine Informationen dargestellt. Dazu gehören die Felder *Aktuelles*, *Studien*, *Statistiken* und *Raumbelegung*. Im Feld *Aktuelles* können allgemeine Informationen hinterlegt werden, *Studien* dient zur Übersicht aller aktuellen Studien, in *Statistiken* werden allgemeine Zahlen, wie z.B. Anzahl der Operationen im aktuellen Jahr, veröffentlicht. In *Raumbelegung* können die tagesaktuellen Raumbelegungen der gemeinsamen Räume eingesehen werden.

Allgemeine Kommentare zum Patienten verfassen

Mitarbeiter können hier Kommentare zum Patienten hinterlassen, welche allen Benutzergruppen zugänglich sind.

Suche Patient

Allgemeine Suchmaske, um Patienten nach auswählbaren Kriterien zu suchen und einzuordnen. Dabei werden nicht nur die Patienten in der DB berücksichtigt, sondern in einer eigenen Tabelle auch die neuen Patienten, die im temporären Speicher des KIS liegen.

Patient neu

Maske um neue Patienten anzulegen, wobei diese zuerst über die Suche in die DB geladen werden sollten. Nur wenn der Patient nicht gefunden wird, kann dieser neu angelegt werden.

Patient Bezugsperson verwalten

Bezugspersonen sind z.B. Eltern, Übersetzer, Sorgeberechtigte, Therapeuten usw. Diese können eingesehen und verwaltet werden.

Versorgungsstatus Patient (Ampel)

Jeder Patient befindet sich vor und nach einer OP in der Behandlung. In welchem Status sich der Patient befindet, wird mit Hilfe von zwei farbigen Bildern dargestellt. Diese befinden sich neben dem

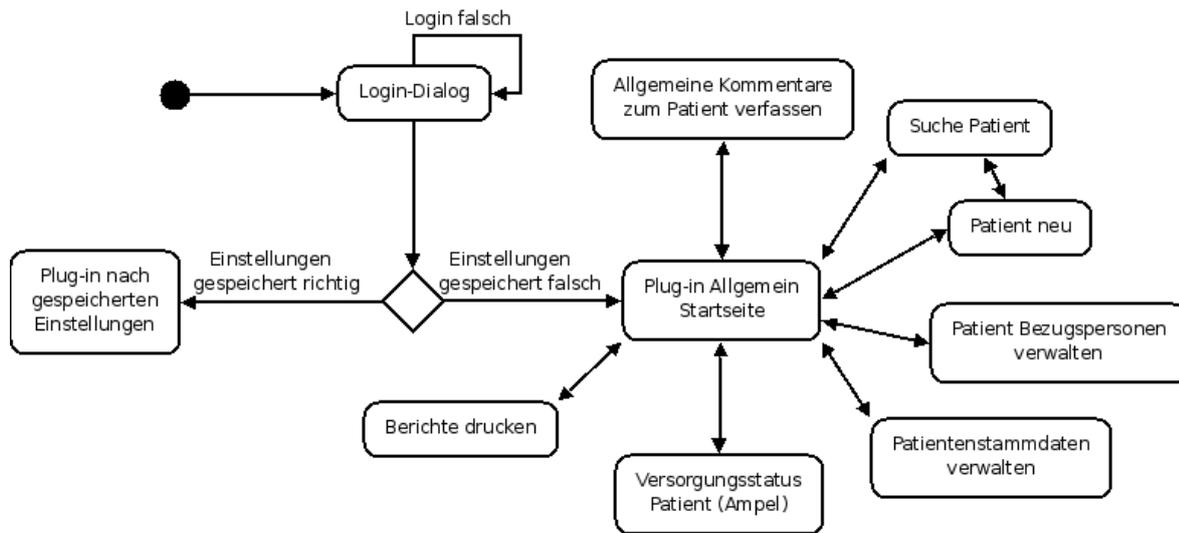


Abbildung 4.4: Allgemeine Funktionen

Foto für jedes Patientenohr. Sind alle Voraussetzungen erfüllt, geht der Patient in eine nächste Phase über. Mögliche Behandlungs-Stadien eines Patienten sind:

1. Präoperativ: Daten werden noch erfasst (grau)
2. Präoperativ: Befunderhebung abgeschlossen (blau)
3. Präoperativ: CI-Kandidat (weiß)
4. CI abgelehnt, bzw. nicht wahrgenommen (blau, mit X)
5. Stationär (z.B. Medizinische Befunde, Messungen Schwindel) (schwarz)
6. Postoperativ: Rehastatus (grün)
7. Postoperativ: Rehastatus läuft aus (gelb)
8. Postoperativ: nicht mehr in Reha (rot)

Berichte drucken

Hier können Berichte gedruckt werden, je nach Benutzerberechtigung. Es können alle allgemeinen Daten bzw. nur ausgewählte Teile gedruckt werden.

4.1.5.2 CI-Technik

Die Benutzergruppe CI-Technik kann die Benutzeransicht nach dem Starten der Allgemeinen Funktionen erreichen. In Abb. 4.5 sind die wesentlichen Funktionen aufgelistet.

Kommentare zum Patienten verfassen

Diese Kommentare sind nur innerhalb der Ansicht dieser Benutzergruppe sichtbar und bearbeitbar.

Suche zum Patient

Die Suche umfasst mehr Möglichkeiten als die der Allgemeinen Funktionen, wie z.B. nach bestimmten CI-Komponenten zu suchen.

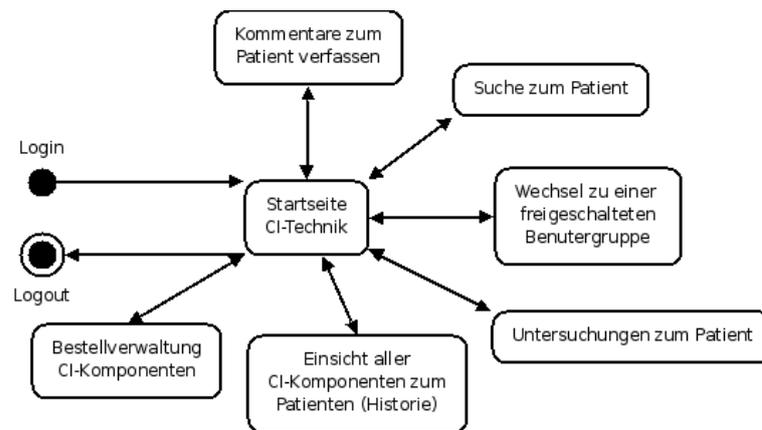


Abbildung 4.5: Benutzergruppe CI-Technik

Wechsel zu einer freigeschalteten Benutzergruppe

Wechsel zu anderen Benutzergruppen und den dazugehörigen Ansichten und Berechtigungen sind möglich, soweit die Rechte vorhanden sind.

Untersuchungen zum Patienten

Maske zur Eingabe der Merkmale einer Patientenuntersuchungen, z.B. bei der Erstanpassung wird die Farbe der CI-Komponenten gespeichert.

Einsicht aller CI-Komponenten zum Patienten (Historie)

Tabellarische Auflistung aller CI-Komponenten, inkl. der Implantate mit Seriennummer usw..

Bestellverwaltung CI-Komponenten

Bestellverwaltung der CI-Komponenten, welche bestellt werden und rechtzeitig vor einer Operation oder Anpassung zur Verfügung stehen müssen.

4.1.5.3 Audiologie und Pädaudiologie

Die Aufgabe der „Audiologie und Pädaudiologie“ besteht u.a. darin das Hörvermögen des Patienten zu messen. Beim Frageninventar werden standardisierte Fragebögen ausgefüllt, wobei Parameter in bekannte Beziehungen zueinander stehen. Im besten Fall können hierdurch Effekte vorhergesagt werden und es dient der Erfolgskontrolle [UHR09]. Die Funktionen sind in der Abbildung 4.6 dargestellt.

Kommentare zum Patienten verfassen

Diese Kommentare sind nur innerhalb der Ansicht dieser Benutzergruppe sichtbar und bearbeitbar.

Suche zum Patienten

Die Suche umfaßt weitere Möglichkeiten als die der Allgemeinen Funktionen, wie z.B. nach Datum sortiert Untersuchungen der Patienten mit einem Audiometer aufzulisten.

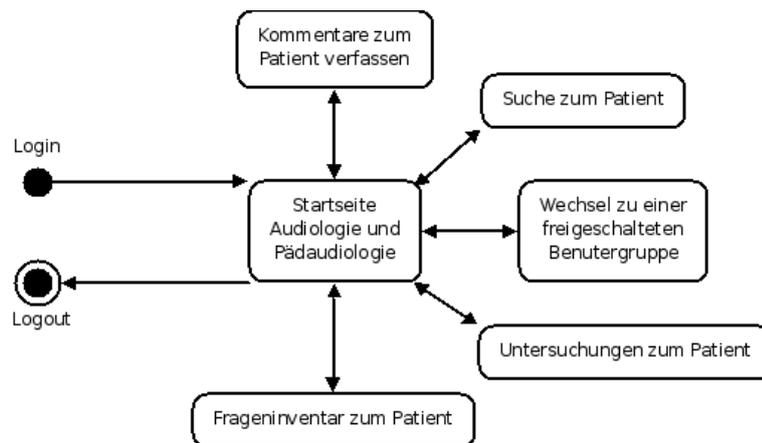


Abbildung 4.6: Benutzergruppe Audiologie + Pädaudiologie

Wechsel zu einer freigeschalteten Benutzergruppe

Wechsel zu anderen Benutzergruppen und den dazugehörigen Ansichten und Berechtigungen sind möglich, soweit die Rechte vorhanden sind.

Untersuchungen zum Patienten

Hier können Ergebnisse der Untersuchungen eingegeben oder aus einem Verzeichnis ausgelesen werden¹.

Frageninventar zum Patient

Eingabe zusätzlicher Informationen zum Frageninventar, welche nicht automatisch aus anderen Feldern ausgelesen werden können.

4.2 Implementierung

In diesem Abschnitt wird der vorausgegangene Entwurf implementiert und interessante Aspekte der Umsetzung näher beschrieben.

4.2.1 Anwendung HL7

HL7-Nachrichten werden mit der Datei-Schnittstelle ausgetauscht. Im Fehlerfall können die HL7-Nachrichten nicht mehr weiter verarbeitet werden und es ist nicht auszuschließen, dass der gesamten Speicherplatz der Partition belegt wird. Dies kann andere Serverdienste und das Betriebssystem zum Absturz bringen. Deshalb wird die Datei-Schnittstelle auf einer eigenen Partition eingerichtet, wodurch andere Serverdienste und das Betriebssystem nicht abstürzen werden.

Zur Überwachung der Datei-Schnittstelle wird ein Skript (siehe Abb. 4.7) eingesetzt, welches jede halbe Stunde aufgerufen wird. Das Skript überprüft die Anzahl der Dateien und legt die Anzahl in einer Logdatei ab. Beim Überschreiten des Limit wird eine E-Mail an das Software-Wartungsteam gesendet. Am Tag werden im KIS der Uniklinik Köln ca. 8000 HL7-Nachrichten gesendet. Im Durch-

¹ Beim Audiometer muss hierzu vorher eine GDT-Datei mit den Untersuchungsergebnissen abgelegt werden.

```

1  #!/bin/sh
2  #
3  # Skript:      observation_count_files.sh
4  # Version:    0.01
5  # Author:     Sascha-Ulf Habenicht <YYYY@Habenicht.biz>
6  #
7  ### Constanten ###
8  USER=root

9  LIMIT='500'
10 DIRECTORY=/var/spool/hl7_2_5/

11 DATE=$(date +"%Y-%m-%d-%H:%M")
12 LOG=/var/log/observation_count_files.log

13 ### watcher ###
14 ( sleep 25 ; kill -15 -$$ ) >/dev/null 2>&1 ) &

15 ### Count Files ###
16 COUNT_FILES=`ls -l ${DIRECTORY} | wc -l`

17 if [ ${COUNT_FILES} -ge $LIMIT ] ; then
18     # Error
19     WARNING=${HOSTNAME}': '${DATE}' | '${DIRECTORY}' is over Limit ('${LIMIT}') with '${COUNT_FILES}' Files.'
20     echo ${WARNING} >> ${LOG}
21     echo ${WARNING} | mail -s "Directory over Limit (HL7)" ${USER}
22 else
23     OK=${HOSTNAME}': '${DATE}' | '${DIRECTORY}' has '${COUNT_FILES}' Files.'
24     echo ${OK} >> ${LOG}
25 fi

```

Abbildung 4.7: Bash-Skript zur Überwachung Datei-Schnittstelle

schnitt sind dies 350 Nachrichten in der Stunde, ein guter Ausgangswert der Überwachung liegt bei schätzungsweise 500 Nachrichten. Das Limit läßt sich empirisch im Betrieb überwachen, da die Anzahl der vorhandenen Nachrichten in einer Logdatei geschrieben werden.

Die HL7-Nachrichten werden mit dem Parser HAPI unter Java verarbeitet. HL7-Nachrichten werden vom KOMMSERV in der Version 2.3 zugestellt. Die Nachrichten werden als Dateien eingelesen und dem Parser übergeben.

4.2.2 Prototyp Plug-in GDT

GDT-Dateien werden auf zwei Arten eingelesen. Es kann für jedes Audiometer eine Datei-Schnittstelle eingerichtet werden. Des Weiteren können GDT-Dateien über die Menüleiste der ERCP importiert werden.

Die Datei-Schnittstelle kann auch mit Hilfe des Skriptes (siehe Abb.4.7) überwacht werden, das Limit sollte bei schätzungsweise 10 Nachrichten liegen.

4.2.3 DBMS PostgreSQL

Die Konfiguration wurde auf einem GNU/Linux Debian Lenny System durchgeführt und sollte bei anderen Betriebssysteme ähnlich ablaufen. Folgenden Dateien werden für die Konfiguration des DBMS verwendet:

- **/etc/postgresql/8.3/main/postgresql.conf**: Konfigurationsdatei,
- **/etc/postgresql/8.3/main/pg_hba.conf**: Authentifizierung der Clients,
- **/var/log/postgresql**: Logdatei.

Folgende Einstellungen wurden gesetzt:

```
listen_addresses = 'localhost,IP-NUMMER' // Lauscht auf localhost und der Server-IP-Adresse  
ssl = true  
fsync = on
```

Bei der Übertragung von Patientendaten sollte die Verschlüsselung SSL zum Schutz der Patientendaten bei der Übertragung im Netzwerk aktiviert sein. Das DBMS soll Transaktionen sofort persistent speichern, aktiviert mit fsync auf on.

Clientauthentifizierung wird durch die Datei pg_hba.conf kontrolliert, HBA steht für host-based authentifizierung. Um den Zugang aus allen IP-Räumen zu erlauben, muss folgender Eintrag mit einer Datenbank und User eingefügt werden:

```
hostssl Datenbank User 0.0.0.0/0 md5
```

Der Benutzer und die Datenbank müssen noch im DBMS hinzugefügt werden. Dies kann mit folgenden SQL-Befehle ausgeführt werden:

```
CREATE USER User;  
ALTER USER User WITH PASSWORD 'geheim';  
CREATE DATABASE Datenbank;
```

Die Logdateien liegen im Verzeichnis /var/log/postgresql und geben bei Fehlern wertvolle Hinweise. Der Zugriff aus JAVA heraus auf PostgreSQL erfolgt mittels JDBC 4 Treiber. JDBC 4 Treiber sind in Java geschrieben und müssen deshalb nicht für jedes Betriebssystem einzeln gepflegt werden.

Auf der Konsole wurde der Benutzer vom Packetmanager postgres für das Client-Programm psql eingerichtet. Eine Alternative ist die Weboberfläche phpPgAdmin, es gibt weitere Anwendungen wie z.B. pgAdmin3. Die Installation von phpPgAdmin kann ebenfalls durch den Packetmanager erfolgen. Es ist ratsam, mit Hilfe von Apache Basic die Benutzer zum Zugriff auf phpPgAdmin einzuschränken.

4.2.4 Aufbau Benutzeroberflächen

Die Benutzeroberflächen werden mit Hilfe des SWT Layouts GridDATA aufgebaut. Aus diesem Grund ist es möglich, weitere Widgets gezielt einzusetzen. Durch den tabellarischen Aufbau von GridDATA ist es außerdem möglich, ein Feld weiter zu unterteilen. Die Auflösung der Benutzeroberflächen ist 1280x1024, dies entspricht den Möglichkeiten der vorhandenen Monitore in den CI-Zentren.

Ein Drittel der Benutzeroberfläche ist den allgemeinen Funktionen zugeordnet, zwei Drittel wiederum der jeweiligen Benutzergruppe.

4.3 Zusammenfassung

Im Entwurf wird die Grundlage für die Implementierung geschaffen. Es wird die Implementierung als N-Tier-Architektur empfohlen, als DBMS wurde PostgreSQL ausgewählt. Der Kommunikationsprozess mit dem KOMMSERV wurde mit der Datei-Schnittstelle entwickelt und zur Verarbeitung der HL7-Nachrichten ein Open-Source Parser verwendet. Zur Kommunikation mit dem Audiometer wird ebenfalls eine Datei-Schnittstelle und eine Import-Funktion vorgeschlagen. Die Funktionen der Benutzeroberflächen wurden festgelegt.

Bei der Implementierung wurden nur interessante Aspekte aufgezeigt, die Überwachung der Datei-Schnittstellen ist notwendig und kann mit Hilfe eines Bash-Skriptes (siehe Abb. 4.7) erfolgen. Beim DBMS PostgreSQL muss der IP-Raum zum jeweiligen erlaubten Zugriff eingerichtet werden. Dies stellte bei der Verwendung von dynamischen IP-Adressen ein Problem dar. Die Benutzeroberflächen werden mit GridLayout implementiert, womit Veränderungen oder Erweiterungen von Widgets in der Entwicklung und Pflege erleichtert werden.

5 Fazit

In der Zusammenfassung werden die in der Einleitung genannten Ziele noch einmal aufgenommen und die Ergebnisse zusammengefasst. Anschließend werden jene Ergebnisse aufgezählt, die weiter in der Softwareentwicklung verwertet werden können.

Im Abschnitt über die Einführung im Arbeitsalltag werden Empfehlungen für die Startphase des neuen Softwaresystems gegeben. Der Abschnitt Projektstand gibt einen aktuellen Stand der Umsetzung in den CI-Zentren Köln und Essen wieder.

Zum Schluss gibt es einen Ausblick über weitere mögliche Entwicklungsschritte des neuen Softwaresystems.

5.1 Zusammenfassung

Die Grundprobleme der Datenintegration Verteilung, Autonomie und Heterogenität können dazu dienen, den Aufwand von anderen Softwaresystemen zur Integration zu erfassen. Mit dem ETL-Prozess wird aufgezeigt, wie aus einer Datenquelle in einem Zielsystem Daten übernommen werden können. Die Daten in einem KIS liegen verteilt in mehreren Softwaresystemen. Durch den Einsatz eines KOMMSERV gibt es nur ein Softwaresystem, über welches mit dem KIS kommuniziert wird. Die Spezialisierung des KOMMSERV auf den Datenaustausch erlaubt die Festlegung auf ein Austauschformat und Version und ist bei der Transformation der verschiedenen Datenmodelle hilfreich. Mit HL7 steht ein ausgereifter Kommunikationsstandard zur Verfügung.

Dagegen liegen Daten vieler medizinischer Geräte in einem Softwaresystem des Herstellers oder eines Partners des Herstellers. Hier kann oft der Kommunikationsstandard GDT genutzt werden. Die Extraktion der GDT-Nachrichten kann z.B. am klinischen Audiometer AT-900 nur durch Benutzeraktionen zustande kommen. Es können hier zur jeweiligen Sitzung des Patienten die Testergebnisse der Untersuchung als GDT-Datei abgelegt werden. Nach Gesprächen mit der Firma Auritech ist eine Anbindung an die Paradox-DB möglich oder die Entwicklung einer HL7-Schnittstelle, welche sich in Planung befindet, abzuwarten. Die Anbindung an der Paradox-DB ist unter Java mit Hilfe eines ODBC-Treibers umsetzbar, jedoch ist dies eine individuelle Lösung zum Integrationsproblem des Audiometers und Bedarf Einsicht in die Unterlagen zur Paradox DB.

Mit Perspektiven, Tabs, einheitlichen Views sowie einer Menüleiste stehen den Entwicklern gute Voraussetzungen zur Verfügung mit der ERCP übersichtliche Benutzeroberflächen anzubieten. SWT und JFace machen einen ausgereiften Eindruck und sind nach einiger Übung gut zu bedienen. Der Einsatz auf mehreren Betriebssystemen¹ stellt keine Probleme da. Die ERCP ist schwer zu erlernen und verlangt eine intensive Auseinandersetzung mit dem Framework. Die Einarbeitung wird mit einer Softwareumgebung belohnt, wo sich auch andere Entwickler zurecht finden können. Die Wartung und Pflege wird zudem unterstützt, so ist z.B. ein Updatemechanismus im Framework nutzbar.

5.2 Ergebnisse

Der Kommunikationsstandard HL7 wird im Prototypen nur lesend benutzt, es werden keine Daten an das KIS gesendet. Wenn sich im Praxisalltag der Open-Source Parser HAPI und der gesamte Prozess des Prototypen zur Verarbeitung der HL7-Nachrichten bewährt, ist eine Umsetzung auch schreibend

¹ Entwickelt wurde auf Mac OSX Snow, Microsoft Windows XP und GNU/Linux Debian Lenny.

mit HL7 Informationen im KIS zur Verfügung zu stellen.

Mit GDT können die Untersuchungsergebnisse gespeichert werden, hier sind jedoch Aktionen der Benutzer notwendig (wie z.B. speichern). Die Möglichkeiten der Verarbeitung von GDT-Nachrichten muss fester Bestandteil des neuen Systems bleiben. Bei dem klinischen Audiometer AT-900 ist jedoch eine eigene Softwareumgebung zur Synchronisation mit der Paradox-DB sinnvoll. Grafische Auswertungen werden zur Laufzeit erzeugt und nicht in der Paradox-DB persistent gespeichert. Zur Zeit gibt es keinen Service von Avantgarde für andere Softwaresysteme, um Untersuchungsergebnisse grafisch aufzuwerten.

Das Datenbankschema ist mehrfach getestet worden und kann für den Praxisalltag weiter ausgebaut werden. Eine Erweiterung der Datenbank sollte problemlos möglich sein, z.B. bei der Einführung neuer Untersuchungen. Ausgangspunkt der Datenbank ist die Person, der Mitarbeiter, Patient oder jemanden eine Rolle gegenüber einem bestimmten Patienten einnimmt. Untersuchungen und deren Ergebnisse können, bezogen auf einen Patienten, über eine Entität UNTERSUCHUNG hin in das Datenbankschema eingebaut werden.

Die grafischen Oberflächen können in der weiteren Softwareentwicklung übernommen werden. Mit Hilfe des GridLayouts lassen sich diese beliebig erweitern und in weitere Felder unterteilen. Diese Oberflächen sollten nun den Benutzergruppen in einer Testphase zur Verfügung gestellt werden, damit weitere Ideen und Anregungen durch diese vor dem Abschluss der Softwareentwicklung einfließen können.

5.3 Einführung in den Arbeitsalltag

Die vorhandenen MS Access Datenbanken wurden von einigen Mitarbeitern nie richtig eingesetzt. Dies ist auch mit der neuen Software zu befürchten. Deshalb müssen im Rahmen eines Change-Management die Mitarbeiter zum Einsatz der Software sensibilisiert werden. Das Trainings-Programm sollte in Rollen aufgeteilt werden und zielgerichtet auf den einzelnen Mitarbeiter abgestimmt werden. Eine klare Kommunikation, wie z.B. Konsequenzen der CI-Zentren beim Scheitern des Einsatzes, sollte jedem Einzelnen eine Verantwortung klar machen. Nur, wenn dem Mitarbeiter deutlich wird, was von ihm erwartet wird, wird sich dieser regelkonform verhalten. Leitfäden, eine Hotline und offene Kommunikation sind unentbehrlich für den Erfolg der Software.

5.4 Projektstand

Das ZIK im Universitätsklinikum Köln unterstützt das CI-Zentrum bei einem Projektantrag zur Anbindung an HL7-Nachrichten. Zur Zeit ist ein Projektantrag in Arbeit und in Absprache mit dem ZIK wird dieser demnächst bei der Verwaltung der Uniklinik Köln eingereicht. Ein GNU/Linux Server wurde dem CI-Zentrum schon zur Verfügung gestellt, auf welchem auch die Datenbank installiert wurde. Es ist geplant, die HL7-Nachrichten durch den KOMMSERV zu filtern und in einer eigenen Partition des Servers als Dateien ab zu legen. Diese müssen dort weiter verarbeitet werden, im FIFO-Verfahren. Im CI-Zentrum Köln werden einige Thin-Clients gegen neue Rechner ausgetauscht, da diese vorraussichtlich nicht genug Leistung besitzen. Das neue System wird zukünftig im Medlan P statt Medlan W betrieben.

Das ZIT im Universitätsklinikum Essen hält eine Softwarelösung außerhalb der bisherigen Systeme für nicht sinnvoll. Hier ist vereinbart worden, die Ergebnisse dieser Arbeit gemeinsam auszuwerten und auf eine Implementierung mit einer Portal-Lösung, durch das ZIT, zusammen zu arbeiten. Die

Ablösung der bisherigen MS Access-DB ist hiervon unbetroffen, da die Arbeiten am Portal von den jetzigen Projektteilnehmern als langfristig angesehen werden.

5.5 Ausblick

Mit der HL7-Anbindung wird ohne Einsatz des Client serverseitig externe Systeme eingebunden. Dieses Software-Modul kommuniziert mit dem Datenmodell des KOMMSERV und transferiert neue Informationen in die DB. Es kann sinnvoll sein, weitere Komponenten, wie u.a. eine Webapplikation einzubinden. Damit auf die DB nicht unzählige Systeme unkontrolliert zugreifen und dementsprechend direkte Verbindungen benötigen, ist die Entwicklung eines Application Server langfristig unumgänglich. Es sollte somit eine N-Tier-Architektur statt Client-Server-Architektur umgesetzt werden.

Die Untersuchungsergebnisse bestehen in den CI-Zentren oft aus Zahlenreihen, welche grafisch aufbereitet werden müssen. Vorhandene Softwareprodukte zur grafischen Aufbereitung sind kostspielig und besitzen oft nur eine Einzelplatzlizenz. Die Entwicklung eines Plug-in zur grafischen Aufbereitung bietet weiteres Ausbaupotential, neben grafischen Bibliotheken müssen hier auch vorhandene Din-Normen beachtet werden.

Durch den Gestaltungsspielraum der CI-Zentren und Anbindung der medizinischen Gerätschaften können die Patientendaten fachgerecht aufbereitet werden und z.B. im Speicherformat PDF/A dem KIS für eine vollständige Patientenmappe zur Verfügung gestellt werden.

Nachdem die MS Access-DBen abgelöst sind, sollte die weitere Entwicklung der Software eine dynamische Workflow-Umgebung bekommen. Zur Gestaltung von Workflow-Prozessen müssen zuerst die vorhandenen Arbeitsprozesse aufgelistet und untersucht werden. Anschließend sollte die Umgebung speziell auf die Bedürfnisse der CI-Zentren entwickelt werden. Harmonisierung, Einfachheit und gegebenenfalls selbst gestalterischen Work-Flow ergeben ein interessantes und aktuelles Softwareprodukt.

Die Umsetzung der N-Tier-Architektur mit Java, ERCP und gegenfalls das Framework Spring als Application Server bietet eine zeitgemäße technologische Umgebung zum weiteren Ausbau des zu entwickelnden Systems. Durch die Einarbeitung einer Work-Flow Verwaltung würden neben den Technologien auch die Prozesse in den CI-Zentren eingebunden. Der Faktor Technologie, Prozess und Mensch sind im Zusammenspiel die Methode für ein modernes und gelungenes Softwareprodukt. Die Mitarbeiterschulungen und die Motivation der Mitarbeiter muss auch von den Entwicklern gezielt unterstützt werden.

A Datenbankschema

Die Beschreibung des Datenbankschema ist in Entitäten, Relationen, Ableitungen und Einschränkungen unterteilt. In den Tabellen werden Attribute und verwendete Datentypen beschrieben, Beziehungen der Tabellen untereinander werden im letzten Feld der jeweiligen Tabelle gezeigt.

Entitäten

Ausgangspunkt ist die Person. Eine Person kann ein Patient selbst sein oder eine bestimmte Rolle gegenüber einem Patienten haben. Mitarbeiter werden in einer eigenen Tabelle persistent gespeichert, um die Zugriffszeiten im vornherein bei der Anmeldung zu reduzieren ¹.

PERSON Personenbezogene Datenerhebung.		
Attribut	Typ	Beschreibung
ID	VARCHAR(20)	ID der Person
NAME	VARCHAR(30)	Nachname
VORNAME	VARCHAR(30)	Vorname
ADRESSE_ID	VARCHAR(20)	Adressenzuordnung
GEB_DATUM	DATE	Geburtsdatum
GESTORBEN_AM	DATE	Im Todesfall Sterbedatum
GEB_ORT	VARCHAR(30)	Geburtsort
GESCHLECHT	CHAR	Geschlecht
STAATSANGEHOERIGKEIT	VARCHAR(30)	Staatsangehörigkeit
MUTTERSPRACHE	VARCHAR(30)	Muttersprache
SPRACHE	VARCHAR(30)	Gesprochene Sprache
RELIGION	VARCHAR(20)	Religionsangehörigkeit
TEL_PRIVAT	VARCHAR(20)	private Telefonnummer
TEL_MOBIL	VARCHAR(20)	mobile Telefonnummer
FAX_PRIVAT ²	VARCHAR(20)	Faxnummer
TEL_ARBEIT	VARCHAR(20)	berufliche Telefonnummer
TEL_MOBIL	VARCHAR(20)	Faxnummer
FAX_ARBEIT	VARCHAR(20)	Faxnummer
EMAIL	VARCHAR(50)	E-Mail Adresse
INFO	XMLTYPE	XML-Dokument zur Erweiterung von Informationen
Constraints		
Primary Key (ID)		
Foreign Key (ADRESSE_ID) references ADRESSE(ID)		

Tabelle A.1: Entität Bezugsperson

¹ Die Tabellen Patienten und Personen sind im Betrieb umfangreich, wogegen die Mitarbeiteranzahl gering ist (meist weniger als 50 Mitarbeiter).

² Schwerhörige Menschen besitzen oft ein privates Faxgerät.

ADRESSE		
Adresse zu Personen (mit Rolle), Mitarbeiter, Patienten oder Krankenkasse.		
Attribut	Typ	Beschreibung
ID	VARCHAR(20)	Identifikation der Adresse ³
ORT	VARCHAR(50)	Ort
PLZ	VARCHAR(50)	Postleitzahl
STRASSE	VARCHAR(50)	Strasse
NR	VARCHAR(50)	Hausnummer
INFO	XMLTYPE	XML-Dokument zur Erweiterung von Informationen
Constraints		
Primary Key (ID)		

Tabelle A.2: Entität Adresse

MITARBEITER		
Liste beinhaltet die Berechtigten zur Benutzung der Software. Zur Zugriffsberechtigung gibt es auch Informationen zu den erlaubten Benutzergruppen.		
Attribut	Typ	Beschreibung
ID	VARCHAR(20)	Identifikation des Mitarbeiters
PASSWORT	VARCHAR(10)	Passwort
PERSON_ID	VARCHAR(20)	Personenbezogene Daten
ADMIN	BOOL	Mitarbeiter darf Benutzer mit Berechtigungen verwalten
TECHNIK	BOOL	Nutzung „CI-Technik“ erlaubt
AUDIOLOGIE	BOOL	Nutzung „Audiologie und Pädaudiologie“ erlaubt
Constraints		
Primary Key (ID)		
Foreign Key (PERSON_ID) references PERSON(ID)		

Tabelle A.3: Entität MITARBEITER

³ Identifikation notwendig, so kann z.B. zwei Krankenkassen dieselbe Adresse haben. Zieht eine Krankenkasse um, muss dies nicht für alle Kassen mit derselben vorherigen Anschrift gelten.

PATIENT		
Daten des Patienten.		
Attribut	Typ	Beschreibung
ID	VARCHAR(20)	ID des Patienten
PERSONEN_ID	VARCHAR(20)	Personendaten, inkl. Adresse
PATIENTENNUMMER	VARCHAR(20)	Patientennummer, oft als Klinik-ID bezeichnet
FALLNUMMER	VARCHAR(20)	Fallnummer
ERSTKONTAKT	DATE	Aufnahme der Betreuung durch das Zentrum
BEHANDLUNGSTAGE	INTEGER(3)	genehmigte Behandlungstage der Krankenkasse
HOERALTER_LINKS	VARCHAR(3)	Muss aus Ableitung berechnet werden
HOERALTER_RECHTS	VARCHAR(3)	Muss aus Ableitung berechnet werden
AMPEL_LINKS	VARCHAR(20)	Versorgungstatus linke Ohrseite
AMPEL_RECHTS	VARCHAR(20)	Versorgungstatus rechte Ohrseite
INFO	XMLTYPE	XML-Dokument zur Erweiterung von Informationen
Constraints		
Primary Key (ID)		
Foreign Key (PERSONEN_ID) references PERSON(ID)		
Foreign Key (AMPEL_LINKS) references AMPEL(ID)		
Foreign Key (AMPEL_RECHTS) references AMPEL(ID)		

Tabelle A.4: Entität Patient

KRANKENKASSE		
Daten und Anschrift der Krankenkassen, Mitarbeiterinformationen ins XML-Feld.		
Attribut	Typ	Beschreibung
ID	VARCHAR(30)	ID der Krankenkasse
NAME	VARCHAR(40)	Name
NUMMER	VARCHAR(20)	Nummer der Kasse
ADRESSE_ID	VARCHAR(20)	Adresse
TEL	VARCHAR(20)	Telefonnummer
FAX	VARCHAR(20)	Faxnummer
EMAIL	VARCHAR(50)	E-Mail Adresse
INFO	XMLTYPE	XML-Dokument zur Erweiterung von Informationen
Constraints		
Primary Key (ID)		
Foreign Key (ADRESSE_ID) references ADRESSE(ID)		

Tabelle A.5: Entität Krankenkasse

BINAEREDATEN		
Binäre Daten wie Bilder. Ein zukünftiger Ausbau zur Speicherung von Dokumenten ist interessant.		
Attribut	Typ	Beschreibung
OID	VARCHAR(20)	ID des binären Datenstrom
PERSON_ID	VARCHAR(20)	Zuordnung zur Person
DATUM	DATE	Datum der Speicherung des Bildes
BILD	BYTEA	binärer Datenstrom
Constraints		
Primary Key (OID)		
Foreign Key (PERSON_ID) references PERSON(ID)		

Tabelle A.6: Entität binärer Datenstrom

COCHLEARIMPLANTAT		
Identifikation des Cochlear Implantat.		
Attribut	Typ	Beschreibung
ID	VARCHAR(20)	ID des Gerätes
SPRACHPROZESSOR	VARCHAR(30)	Typ des Sprachprozessors
SPRACHPROZESSORNR	VARCHAR(30)	Sprachprozessornummer
SERIENNUMMER	VARCHAR(50)	Seriennummer
HERSTELLER	VARCHAR(20)	Firma des CI
TYP	VARCHAR(20)	Typ des Implants
INFO	XMLTYPE	XML-Dokument zur Erweiterung von Informationen
Constraints		
Primary Key (ID)		

Tabelle A.7: Entität Cochlear Implantat

UNTERSUCHUNG		
Grunddaten zur Untersuchung eines Patienten.		
Attribut	Typ	Beschreibung
ID	VARCHAR(20)	ID der Untersuchung
MITARBEITER_ID	VARCHAR(20)	Mitarbeiter der Klinik
PATIENT_ID	VARCHAR(20)	Patient, welcher untersucht wird
DATUM	DATE	Datum
ART	VARCHAR(20)	Kontrolle oder Anpassung
INFO	XMLTYPE	XML-Dokument zur Erweiterung von Informationen
Constraints		
Primary Key (ID)		
Foreign Key (MITARBEITER_ID) references MITARBEITER(ID)		
Foreign Key (PATIENT_ID) references PATIENT(ID)		

Tabelle A.8: Entität Untersuchung

AMPEL		
Patient durchläuft von der Erstuntersuchung bis zu einem Jahr Nachversorgung nach der OP. Durch den Datentyp DATE kann die Zeitspanne nachvollzogen werden.		
Attribut	Typ	Beschreibung
ID	VARCHAR(20)	Identifikation
A1	DATE	Präoperativ: Daten werden noch erfasst
A2	DATE	Präoperativ: Befunderhebung abgeschlossen
A3	DATE	CI abgelehnt
A4	DATE	CI Kandidat
A5	DATE	Stationär
A6	DATE	Postoperativ: Rehastatus
A7	DATE	Postoperativ: Rehastatus läuft aus
A8	DATE	Postoperativ: nicht mehr in Reha
Constraints		
Primary Key (ID)		

Tabelle A.9: Entität Ampelfunktion

UNTERSUCHUNGTONAUDIOMETRIE		
Daten aus der Untersuchung Tonaudiometrie		
Attribut	Typ	Beschreibung
ID	VARCHAR(20)	ID
UNTERSUCHUNG_ID	VARCHAR(20)	Zuordnung zur Untersuchung
KURVE	VARCHAR(20)	Luftleitung, Knochenleitung, Freifeld, Blähkurve
ART	VARCHAR(10)	Rechts, Links oder Variabel
F0250	VARCHAR(20)	250 Hz
F0500	VARCHAR(20)	500 Hz
F1000	VARCHAR(20)	1000 Hz
F2000	VARCHAR(20)	2000 Hz
F4000	VARCHAR(20)	4000 Hz
F6000	VARCHAR(20)	6000 Hz
Constraints		
Primary Key (ID)		
Foreign Key (UNTERSUCHUNG_ID) references UNTERSUCHUNG(ID)		

Tabelle A.10: Entität Untersuchung

UNTERSUCHUNGSPRACHAUDIOMETRIE		
Daten aus der Untersuchung Sprachaudiometrie.		
Attribut	Typ	Beschreibung
ID	VARCHAR(20)	ID der Untersuchung
UNTERSUCHUNG_ID	VARCHAR(20)	Zuordnung zur Untersuchung
NUTZSCHALL	VARCHAR(20)	Nutzschall
LUFTLEITUNG	VARCHAR(20)	Luftleitung
LKONFIGURATION	VARCHAR(20)	Lautsprecherkonfiguration: links, rechts, hinten, vorne
KKONFIGURATION	VARCHAR(20)	Kopfhörerkonfiguration: links, rechts, beide
STOERSCHALL	CHAR	Störschall: an oder aus
STOERSCHALL_ART	VARCHAR(20)	Art des Störschall
F000	VARCHAR(3)	Wert in dB
F005	VARCHAR(3)	Wert in dB
F010	VARCHAR(3)	Wert in dB
F015	VARCHAR(3)	Wert in dB
F020	VARCHAR(3)	Wert in dB
F025	VARCHAR(3)	Wert in dB
F030	VARCHAR(3)	Wert in dB
F035	VARCHAR(3)	Wert in dB
F040	VARCHAR(3)	Wert in dB
F045	VARCHAR(3)	Wert in dB
F050	VARCHAR(3)	Wert in dB
F055	VARCHAR(3)	Wert in dB
F060	VARCHAR(3)	Wert in dB
F065	VARCHAR(3)	Wert in dB
F070	VARCHAR(3)	Wert in dB
F075	VARCHAR(3)	Wert in dB
F080	VARCHAR(3)	Wert in dB
F085	VARCHAR(3)	Wert in dB
F090	VARCHAR(3)	Wert in dB
F095	VARCHAR(3)	Wert in dB

UNTERSUCHUNGSPRACHAUDIOMETRIE		
Daten aus der Untersuchung Sprachaudiometrie.		
Attribut	Typ	Beschreibung
F100	VARCHAR(3)	Wert in dB
Constraints		
Primary Key (ID)		
Foreign Key (UNTERSUCHUNG_ID) references UNTERSUCHUNG(ID)		

Tabelle A.11: Entität Untersuchung

Relationen

Nach den Entitäten folgt nun die Beschreibung der Relationen.

PERSON_ROLLE_PATIENT		
Eine Person kann Elternteil, Bezugsperson, Sorgerechter, Amtsvormund, Therapeut gegenüber einem Patient sein.		
Attribut	Typ	Beschreibung
PATIENT_ID	VARCHAR(20)	ID des Patienten
ROLLE	VARCHAR(20)	Rolle gegenüber des Patienten
PERSON_ID	VARCHAR(20)	ID der Person
Constraints		
Primary Key (PATIENT_ID, ROLLE, PERSON_ID)		
Foreign Key (PATIENT_ID) references PATIENT(ID)		
Foreign Key (PERSON_ID) references PERSON(ID)		

Tabelle A.12: Relation Personen in Rolle zum Patienten

KASSE_PATIENT		
Zuordnung des Patienten zu der Krankenkasse.		
Attribut	Typ	Beschreibung
PATIENT_ID	VARCHAR(20)	ID des Patienten
KASSEN_ID	VARCHAR(20)	ID der Krankenkasse
STATUS	VARCHAR(20)	Selbstzahler, Privat oder Kasse
VERSICHERUNGSNUMMER	VARCHAR(30)	Versicherungsnummer des Patienten
VERSICHERUNGSSTATUS	VARCHAR(20)	Versicherungsstatus des Patienten
VON	DATE	Datum ab der Versicherung
BIS	DATE	Datum, falls Versicherung nicht aktuell ist
Constraints		
Foreign Key (PATIENT_ID) references PATIENT(ID)		
Foreign Key (KASSEN_ID) references KRANKENKASSE(ID)		

Tabelle A.13: Relation Patient zur Krankenkasse

CI_PATIENT		
Eine Person kann links oder rechts oder beide Hörseiten Implantate tragen. Durch Reimplikation kann es mehrere Geräte zum Patienten geben.		
Attribut	Typ	Beschreibung
PATIENT_ID	VARCHAR(20)	ID des Patienten
COCHLEAR_ID	VARCHAR(20)	ID des implantierten CI
OHR_SEITE	CHAR	Seite des Ohres von vorne aus betrachtet, links oder rechts
OP	DATE	Datum der OP ⁴
ALTER	INTEGER(3)	Alter des Patienten bei der OP in Monaten
ERSTANPASSUNG	DATE	Datum der ersten Anpassung des CI nach der OP
ZUGANG	VARCHAR(20)	Implationszugang ⁵
INFO	XMLTYPE	XML-Dokument zur Erweiterung von Informationen
Constraints		
Primary Key (PATIENT_ID, COCHLEAR_ID)		
Foreign Key (PATIENT_ID) references PATIENT(ID)		
Foreign Key (COCHLEAR_ID) references COCHLEARIMPLANTAT(ID)		

Tabelle A.14: Relation CI zu Patient

Ableitungen

Ableitungen (Derivations)
(CA1) Das Höralter eines Patienten berechnet sich pro Ohrseite durch aktuelles Datum minus Versorgung (Erstanpassung) der betrachteten Ohrseite. ⁶
(CA2) Bei Kassenpatienten werden Versorgungen von den genehmigten Behandlungstagen in Minuteneinheiten verrechnet.
(CA3) Lebensalter wird in Monaten gespeichert und bei unter 5 Jährigen mit zusätzlicher Lebensmonatszahl angegeben.

Tabelle A.15: Ableitungen

Einschränkungen

Einschränkungen (Constraints)
(BR1) Ein Administrator muss auch Mitarbeiter der Abteilung sein oder berechtigt sein, Patientendaten zu administrativen Zwecke einzusehen.
(BR2) Ein Mitarbeiter darf nur Patientendaten verarbeiten oder einsehen die zu einer seiner Rollen gehört.
(BR3) Behandlungstage müssen bei Kassenpatienten genehmigt sein, sonst kann der Patient nicht abgerechnet werden.
(BR4) Ein Patient darf nur angeschrieben oder untersucht werden, wenn kein Sterbedatum vorliegt.

Tabelle A.16: Einschränkungen

⁴ Aktuelle Implantate werden mit Hilfe des OP-Datums ermittelt. History der Implantate darf nicht verloren gehen.

⁵ Standardcochleostony, Round Window Insertion, Sonstiges

⁶ Kann nicht allein durch eine Formel abgebildet werden, dient nur der Orientierung und muss korrigierbar sein.

Tonaudiometrie

Der GDT-Datensatz beinhaltet die Identifikation des Patienten und Testergebniss von einem Audiometertest.

01380006310<CR><LF>
014810000187<CR><LF>
014921802.00<CR><LF>
01030003<CR><LF>
0193101Mustermann<CR><LF>
0133102Paul<CR><LF>
017310318041962<CR><LF>
0158402AUDI00<CR><LF>
017620005111993<CR><LF>
0238410Ton, Spr, MaTe<CR><LF>
017843205111993<CR><LF>
0158439115814<CR><LF>
0468800/21/20/16/17/20/39/77/76/73/60/41/35/<CR><LF>
0448801/8/8/9/255/12/17/14/15/20/19/23/19/<CR><LF>
0348820/0/0/0/0/0/0/0/0/0/0/<CR><LF>
0348821/0/0/0/0/0/0/0/0/0/0/<CR><LF>
0498802/255/16/13/14/16/33/68/60/42/34/255/255/<CR><LF>
0438803/255/2/3/4/6/11/11/11/9/6/255/255/<CR><LF>
0128830/2/<CR><LF>
0128831/1/<CR><LF>
0128829/2/<CR><LF>

Glossar

Nachfolgend sind noch einmal wesentliche Begriffe dieser Arbeit zusammengefasst und erläutert. Eine ausführliche Erklärung findet sich jeweils in den einführenden Abschnitten sowie der jeweils darin angegebenen Literatur. Das im Folgenden im Rahmen der Erläuterung verwendete Symbol ~ bezieht sich jeweils auf den im Einzelnen vorgestellten Begriff, das Symbol ↑ verweist auf einen ebenfalls innerhalb dieses Glossars erklärten Begriff.

Abrechnungsdatentransfer Der ~ dient zur Übermittlung von relevanten Daten zur Abrechnung mit den Krankenkassen und gehört zu der Gruppe der ↑xDT ↑Datenaustauschformate.

Application-Server Ein ~ (Anwendungsserver) ist ein Server, auf dem Anwendungsprogramme ausgeführt werden. Der ~ stellt Dienste zur Verfügung.

Algorithmus Der ~ ist eine Handlungsvorschrift zur Lösung eines oder einer bestimmten Art von Problemen in endlich vielen Schritten.

Audiologie Diese Gruppe stellt die Hörfähigkeit ohne Hörhilfe, mit Hörgerät oder ↑Cochlea Implantat fest. Hierzu werden subjektive und objektive Hörtests durchgeführt und Ergebnisse gespeichert.

Audiometer Medizinisches Gerät zur Ermittlung der Hörfähigkeit eines Patienten.

Avantgarde Software zur Visualisierung der Messdaten, die mit einem ↑Audiometer erzeugt wurden.

Behandlungsdatentransfer ~ ist ein ↑Datenaustauschformat für Behandlungsdaten und ermöglicht den Export und Import von kompletten Datensätzen verschiedener Hersteller. ~ gehört zur Gruppe der ↑xDT ↑Datenaustauschformate.

Bibliothek ~ ist ein Synonyme für Software-Bibliothek, welche zur Wiederverwendung Funktionalitäten bündelt und mehrfach verwendet werden kann. Reduziert und vereinfacht die Software-Entwicklung durch Wiederverwendung.

CI-Technik Die ~ ist für die Versorgung und Kontrolle des Patienten mit einem ↑Cochlea Implantat zuständig.

Cochlea ~ , zu deutsch Hörschnecke, ist das Innenohr. Die ~ stellt das Rezeptorfeld für die Hörwahrnehmung dar.

Cochlea Implantat (engl. cochlear implant) Ein ~ ist eine Hörprothese, die hochgradig schwerhörigen und ertaubten Menschen das Hören ermöglicht und die Hörsinneszellen des Innenohres ersetzt.

Datenaustauschformat Ein ~ ist ein spezielles ↑Datenformat zum Austausch zwischen zwei verschiedenen Systemen.

Datenbank Menge der verwaltenden Daten in einer festen Struktur, die durch das ↑Datenbankmanagementsystem verwaltet wird. Die feste Struktur ist abhängig vom verwendeten Datenmodell.

Datenbankmanagementsystem ~ ist eine Verwaltungssoftware, die ein Datenmodell nutzt und darauf u.a. lesende und schreibende Zugriffe kontrolliert.

Datenformat Ein ~ regelt den strukturellen Aufbau der Daten u.a. zum Austausch oder persistenten Speicherung.

Datenquelle Eine ~ liegt vor, wenn aus einem anderen System Daten erhalten oder abgefragt werden kann.

Data-Warehouse Das ~ ist eine zentrale Datensammlung, deren Inhalt sich aus Daten unterschiedlicher Quellen zusammensetzt. Das ~ dient der Analyse dieser Daten und hilft bei der Aufbereitung zur betriebswirtschaftlichen Analyse.

Eclipse ~ ist ein Programmierwerkzeug zur Entwicklung von Software und ist als Entwicklungsumgebung für die Programmierung in Java bekannt. Durch das ↑Plug-in Konzept kann Eclipse als ↑Framework zur Entwicklung eigener Software eingesetzt werden.

Framework Ein ~ ist ein Programmiergerüst und kann zur Entwicklung eigener Software eingesetzt werden. Das ~ zeichnet sich dadurch aus, dass viele bewährte Entwurfsentscheidungen vorgegeben werden und dabei helfen, die eigene Entwicklung zu verkürzen.

e*Gate ~ ist ein ↑Kommunikationsserver der Firma SUN und kann als Datenquelle dienen.

Gerätedatentransfer ~ soll eine systemunabhängige Datenübertragung zwischen medizinischen Geräten und einer Software gewährleisten. ~ gehört zu der Gruppe der ↑xDT Datenaustauschformate.

Health Level Seven ~ ist ein klinischer Standard für den Austausch von Daten und legt den Aufbau des ↑Datenaustauschformat fest.

Kommunikationsserver Ein ~ nimmt Nachrichten entgegen und sendet diese an eine Liste von Adressaten weiter. Der ~ ist in der Lage Nachrichten zu verstehen und in andere Datenformate, jeweils für die Adressaten, umzuwandeln.

Informationssystem Ein ~ stellt Werkzeuge zur persistenten Speicherung und Verarbeitung von Informationen zur Verfügung.

Labordatentransfer Der ~ dient zur Anforderung von Laboruntersuchungen und zur Übermittlung der Ergebnisse dieser Untersuchungen. ~ gehört zu der Gruppe der ↑xDT Datenaustauschformate.

Pädaudiologie ~ ist ein Teilgebiet der ↑Audiologie, welches für Hörtests von Patienten im Kindesalter zuständig ist.

Plug-in Ein ~ ist ein Softwaremodul, das in ein anderes Softwareprodukt eingebunden wird und damit dessen Funktionalität erweitert.

Prototyp Ein ~ ist eine erste Umsetzung eines Entwurfes zur Software, welches zielgerichtet Anforderungen untersucht und dabei hilft, Details zur Softwareentwicklung im Voraus zu klären.

Software Development Kit Sammlung von Werkzeugen und Anwendungen, um eine Software zu erstellen.

SWT ~ ist eine Bibliothek für die Erstellung grafischer Oberflächen mit Java. SWT ist Bestandteil der Entwicklungsumgebung ↑Eclipse und nutzt native grafische Elemente des Betriebssystems. Mit ~ können Programme erstellt werden, die eine Optik vergleichbar mit nativen Programmen aufweisen.

xDT ~ ist eine Gruppe von ↑Datenaustauschformaten, hierzu gehören ↑Abrechnungsdatentransfer, ↑Behandlungsdatentransfer, ↑Gerätedatentransfer und ↑Labordatentransfer. ~ wurde im Auftrag der Kassenärztlichen Bundesvereinigung erstellt.

Widget Ein ~ ist eine Komponente eines grafischen Fenstersystems.

Abkürzungen

3LGM ²	3-Ebenen-Meta-Model
ADT	Abrechnungsdatentransfer
API	Application Programming Interface
AVS	Archivverwaltungssystem
BDT	Behandlungsdatentransfer
CI	Cochlea Implantat
CI-Zentrum	Cochlear-Implant-Zentrum
DB	Datenbank
DBMS	Datenbank Management System
DWH	Data-Warehouse
DWHS	Data-Warehouse-System
ERCP	Eclipse Rich Client Plattform
EVN	Event (engl.): Ereignis
FIFO	First-In-First-Out
GDT	Gerätedatentransfer
GUI	Graphical User Interface (Software-Komponente)
HL7	Health Level Seven
HNO	Hals-Nasen-Ohrenklinik
JDBC	Java Database Connectivity
JDT	Java Development Tooling
JRE	Java Runtime Environment
KDMS	Krankenhausdatenmanagementsystem
KIS	Krankenhausinformationssystem
KOMMSERV	Kommunikationsserver
LDT	Labordatentransfer
LIS	Laborinformationssystem
MED-DB	Medizinische Datenbank
MedLan-P	Patientennetzwerk
MedLan-W	Wissenschaftsnetzwerk
MSH	Message Header
ODBC	Open Database Connectivity
OP	Operativer Eingriff
PDE	Plug-in Developer Environment
PID	Patient Identification
PVS	Patientenverwaltungssystem
RMI	Reference Information Model
SDK	Software Development Kit
SWT	Standard Widget Toolkit
UI	User Interface
Abb.	Abbildung
etc.	et cetera (lat.): „und so weiter“
i.d.R.	in der Regel
u.a.	unter anderem

vgl.	vergleiche
z.B.	zum Beispiel

Abbildungen

1.1	Fachlandkarte HNO-Uniklinik (Köln)	2
2.1	Klassifikation von Anfragesystemen nach [KRD99]	6
2.2	Das 3-Ebenen-Meta-Modell (3LGM ²) [LEH05]	8
2.3	Aufbau einer HL7-Nachricht [KOC03]	10
2.4	Audiometriemodul AT900	11
2.5	Grafische Auswertung Tonaudiometrie ¹	12
2.6	Eclipse SDK [ERC09]	14
2.7	Workbench Lebenszyklus [JM06]	16
3.1	Ansicht Patientenstammdaten der MS Access-DB Uniklinikum Köln	20
3.2	KIS im 3LGM ² -Modell (Universitätsklinikum Köln)	21
3.3	Datenbankschema ohne Attribute	29
4.1	System als N-Tier-Architektur	34
4.2	ETL-Prozess mit HL7-Nachrichten	35
4.3	ETL-Prozess mit GDT-Nachrichten	36
4.4	Allgemeine Funktionen	38
4.5	Benutzergruppe CI-Technik	39
4.6	Benutzergruppe Audiologie + Pädaudiologie	40
4.7	Bash-Skript zur Überwachung Datei-Schnittstelle	41

Literatur

- [AHL10] AHLERING, BERND: *Entwicklung eines Patientendaten Verwaltungs-Prototypen mit dem besonderen Fokus: Auswertungsunterstützungen von Patientenstammdaten*. 2010. – IP-Arbeit an der Universität Oldenburg
- [AUR07] AURITECH GMBH (Hrsg.): *Bedienungsanleitung Audiometriesystem AT900*. Hamburg: AURITECH GmbH, 2007
- [DAU08] DAUM, BERTHOLD: *Java-Entwicklung mit Eclipse 3.3*. Heidelberg : dpunkt.verlag, 2008
- [ERC09] *ECLIPSE SDK*. THE ECLIPSE FOUNDATION. <http://help.eclipse.org/ganymede/topic/org.eclipse.platform.doc.isv/guide/images/sdk-arch.jpg>. Version: 2009. – Besucht am 3. Dezember 2009; 10:00 Uhr
- [FN07] FELIX NAUMANN, ULF L.: *Informationsintegration: Architekturen und Methoden zur Integration verteilter und heterogener Datenquellen*. Heidelberg : dpunkt.verlag, 2007
- [HG09] HOLGER GÜNZEL, ANDREAS B.: *Data-Warehouse-Systeme*. Heidelberg : dpunkt.verlag, 2009
- [JB08] JENS BENDISPOSTO, HEIKO S. u.: *Eclipse Rich Client Platform: Entwicklung von erweiterbaren Anwendungen mit RCP*. München, 2008
- [JM06] JEFF MCAFFER, JEAN-MICHEL L.: *Rich Client Application Development*. IBM Rational Software. http://wiki.eclipse.org/images/d/d9/EclipseCon_RCP_Tutorial_2006.pdf. Version: 2006
- [KAS06] *SCHNITTSTELLEN*. KASSENÄRZTLICHE BUNDESVEREINIGUNG (KBV). <http://www.kbv.de/ita/4274.html>. Version: 2006. – Besucht am 19. November 2009; 14:00 Uhr
- [KOC03] KOCH, SASCHA: *IuK-Systeme im Gesundheitswesen*. OFFIS. http://www.mh-hannover.de/fileadmin/zentrale_einrichtungen/tumorzentrum/Klinisches_Krebsregister/Arbeitsgruppe_TuDaSch/Dokumentensammlung/weitere_Themen/KRG_AGS_DOK_HL7200302.pdf. Version: 2003
- [KRD99] KLAUS R. DITTRICH, RUXANDRA D.: An Overview and Classification of Mediated Query Systems. In: *SIGMOD Rec.* 28 (1999), 63–72. <http://doi.acm.org/10.1145/333607.333615>. – ISSN 0163–5808
- [LEH05] LEHMANN, THOMAS M.: *Handbuch der Medizinischen Informatik*. München Wien : Carl Hanser Verlag, 2005
- [LEN02] LENZERINI, MAURIZIO: Data Integration: A Theoretical Perspective. (2002), S. 233–246
- [NUE05] *INFORMATIONEN GDT/BDT-SCHNITTSTELLE UND TEST-DATEI VOM AT900*. ING.-BÜRO JOACHIM NÜSS, 2005. – Besucht am 2. Dezember 2009; 7:00 Uhr

- [POS09] *ABOUT. POSTGRESQL* Global Development Group. <http://www.postgresql.org/about/>. Version: 2009. – Besucht am 1. Dezember 2009; 15:00 Uhr
- [QUA99] QUALITÄTSRING MEDIZINISCHE SOFTWARE (Hrsg.): *Anbindung von medizinischen Meßgeräten (GDT - GERÄTEDATEN-TRÄGER)*. Revidierte Fassung Dienstag, 4. März 1999. Darmstadt: QUALITÄTSRING MEDIZINISCHE SOFTWARE, 1999. – Schnittstellenbeschreibung zum systemabhängigen Datentransfer zwischen Praxis-EDV und Meßgeräten
- [REI73] REICHERTZ, PETER L.: Medizinische Informatik - Aufgaben, Wege und Bedeutung. In: *IBM Nachrichten 23* (1973), S. 567–576
- [REI09] REICHERT, STEFAN: *Eclipse RCP im Unternemenseinsatz*. Heidelberg : dpunkt.verlag, 2009
- [RL09] ROLAND LASZIG, ERNST L. ; 9 (Hrsg.): *Praxis der Audiometrie*. Georg Thieme Verlag, 2009
- [SBN05] SHAMKANT B. NAVATHE, RAMEZ E. ; 3 (Hrsg.): *Grundlagen von Datenbanksystemen*. München : Pearson Studium, 2005
- [SCH00] SCHADOW, GUNTHER: *Krankenhauskommunikation mit HL7: Analyse, Implementation und Anwendung eines Protokollstandards für medizinische Datenkommunikation*. Aachen : Shaker Verlag, 2000
- [SOU09] *FIND AND DEVELOP OPEN SOURCE SOFTWARE*. GEEKNET INC. <http://sourceforge.net>. Version: 2009. – Besucht am 17. Dezember 2009; 21:00 Uhr
- [SR09] STEFAN REICHERT, PETER F.: *Enterprise Eclipse RCP*. <http://it-republik.de/jaxenter/artikel/Enterprise-Eclipse-RCP-2163.html>. Version: 2009. – Besucht am 20. Dezember 2009; 18:00 Uhr
- [UHR09] *FRAGENINVENTARE*. Jean-Uhrmacher-Institut für klinische HNO-Forschung. <http://www.uhrmacherinstitut.de/index.php?id=621>. Version: 2009. – Besucht am 28. Dezember 2009; 16:00 Uhr
- [W3L09] *DREI-EBENEN-META-MODELL*. Institut für Medizinische Informatik, Statistik und Epidemiologie (IMISE). <http://www.3lgm2.de>. Version: 2009. – Besucht am 15. Dezember 2009; 21:30 Uhr

Index

A		SWT 17
Anforderungen 22		
Funktionale 22		
Nichtfunktionale 23		
Audiometer		
Abbildung 11		
AT900 21		
Gerätedatentransfer 21		
Grafische Auswertung 12		
C		
Cochlear Implantat 1		
Zentrum 1		
D		
Datei-Schnittstelle		
Gerätedatentransfer 12		
Helth Level Seven 34		
Datenhaltung 28		
Datenintegration 5		
Abbildung Anfragesysteme 6		
Anfragesysteme 6		
Autonomie 6		
Heterogenität 6		
Prozess 7		
Verteilung 6		
E		
Eclipse 13		
Benutzerschnittstellen 17		
Konfiguration 14		
Lebenszyklus Plug-in 15		
Rich Client Plattform 14		
Ergebnisse 45		
H		
Health Level Seven 9		
Aufbau 10		
HNO-Klinik		
Fachlandkarte 2		
J		
Java 29		
JFace 17		
SWING 17		
K		
Krankenhausinformationssystem 7		
3LGM ² 8		
Abbildung Architektur im 3LGM ² -Modell 21		
Architektur 8		
Kommunikationsserver 9		
Uniklinik Köln 20		
M		
MS Access-DBen 19		
Bestandsaufnahme 19		
O		
Open Source 30		
HAPI 30		
HL7 Java API 30		
jL7 30		
P		
PHP 29		
PostgreSQL 41		
Projektstand 46		
Prototyp 1		
Allgemeine Funktionen 37		
Architektur 33		
Audiologie und Pädaudiologie 39		
CI-Technik 38		
Datenbank 33		
Datenbankmanagementsystem 41		
Gerätedatentransfer 36		
Health Level Seven 34		
Layout GridData 42		
Rollen 37		
R		
Rich Client 27		
T		
Thin Client 27		
V		
Verteilte Anwendungen 25		
3-Schichten-Architektur 25		

Client-Server-Architektur	25
Mehrschichten-Architektur	25

W

Webapplikation	27
----------------------	----

X

xDT	12
Abrechnungsdatentransfer	12
Behandlungsdatentransfer	12
Gerätedatentransfer	12
Labordatentransfer	12

Z

Ziele	2
-------------	---

Versicherung

Hiermit versichere ich, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Außerdem versichere ich, dass ich die allgemeinen Prinzipien wissenschaftlicher Arbeit und Veröffentlichung, wie sie in den Leitlinien guter wissenschaftlicher Praxis der Carl von Ossietzky Universität Oldenburg festgelegt sind, befolgt habe.

Oldenburg, 3. Januar 2010

Sascha-Ulf Habenicht