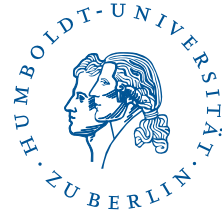


HUMBOLDT-UNIVERSITÄT ZU BERLIN



Studienarbeit zur Integration von Krankenhausinformationssystemen am Beispiel von SAP und FileMaker

Daniel Neumann¹

Dezember 2008
Institut für Informatik

Betreuung durch:
Petra Kriegel, SAP[®], Helios Klinikum Berlin-Buch²
Ulrich Hertweck, FileMaker[®], system.de System&Project GmbH³
Prof. Dr. Ulf Leser, Institut für Informatik, Humboldt-Universität zu
Berlin

¹dneumann (at) informatik.hu-berlin.de

²petra.kriegel (at) helios-kliniken.de

³uli (at) system.de

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	2
1.2	Umfeld	3
2	Das Krankenhaus-Informationssystem	4
2.1	Funktionen eines KIS	6
2.2	KIS-Architekturen	7
2.3	Krankenhauskommunikation	10
2.3.1	Kommunikationsserver	11
2.3.2	HL7	13
2.4	SAP im Krankenhaus: IS-H und IS-H*Med	16
2.4.1	Die ADT-Schnittstelle	18
2.4.2	BAPI	21
2.4.2.1	Interner und externer Zugriff	22
2.4.2.2	SAP Java Connector	24
3	KIS im HELIOS Klinikum Berlin-Buch	29
3.1	SAP, IS-H und IS-H*Med im HELIOS Klinikum Berlin-Buch	31
3.2	Weitere Medizinische Informationssysteme	32
3.2.1	Heartware: Patientendokumentation in der Kardiologie	33
3.2.1.1	Funktionalitäten	36
3.2.1.2	Skriptgesteuerte Ausführung	37
3.2.1.3	Import von XML-strukturierten Daten	38
3.2.2	Überblick über weitere medizinische Informationssysteme	39
4	Unidirektionaler Datenabgleich zwischen IS-H*Med und Heartware	44
4.1	Vorgehen	45
4.2	Implementierung und Einbindung in den Betrieb	47
4.2.1	SAP Connector	47
4.2.2	Heartware	60

4.3 Probleme und Fehlerverhalten	64
5 Fazit	66

Abbildungsverzeichnis

2.1	Kommunikationsserver fördern die Bus- oder Sternstruktur eines KIS (rechts), aus [Lan03].	11
2.2	Socket- oder dateibasierte Kommunikation, aus [Lan03].	12
2.3	Gewöhnliches 3-Schichten-Architekturmodell, aus [Mos99].	16
2.4	SAP R/3, IS-H, IS-H*Med und IS-H*Med-Module, aus [GSP ⁺ 03].	17
2.5	Aufbau der datei-basierten Nachrichten, aus [SAP06].	20
2.6	Business Objekt Patient, Bildschirmfoto aus dem SAP BOR.	27
2.7	Struktur der Business Objekte, aus [SAP07].	28
3.1	Übersicht über die eingesetzten Informationssysteme und deren Kommunikation	30
3.2	Das Hauptmenü von Heartware 1.1	41
3.3	Das Hauptmenü von Heartware 2	42
3.4	Import: Aktualisierung von passenden Datensätzen, aus [Fil04]	42
3.5	Skript „Drucken“. Die während der Skripterstellung gültigen Druckereinstellungen werden für die Druckereinrichtung zur Laufzeit genutzt.	43
4.1	Modulare Aufgabenverteilung	46
4.2	Definition einer gerichteten SAP-Schnittstelle mit dem Patientendokumentationssystem Heartware	65

Tabellenverzeichnis

2.1	ADT-Ereignistypen in IS-H	19
2.2	Beispiel Dokumentenverweis	26
2.3	Mapping von ABAP, Java und JCo, aus [Sch02].	28
4.1	Fehlercodes von BO Patient	49
4.2	Filterattribute von BO Patient	50

Kapitel 1

Einleitung

Die Integration von Informationstechnologie (IT) in Krankenhäusern gilt noch immer als eine der großen Herausforderungen der IT. Ein Blick zurück in die Geschichte der medizinischen Datenverarbeitung soll dies begründen.

Im Jahre 1935 wurden erstmals Krankenblattdaten in einem Krankenhaus in Rochester, USA auf eine Lochkarte gespeichert [Sch00], also bereits vor der elektronischen Datenverarbeitung (EDV). Ein erstes Expertensystem - HELP [Dud86], welches entscheidungsunterstützendes, zeitgemäßes, medizinisches Wissen bietet, entstand in den USA in den 1960er Jahren. Besonders Krankenhäuser in ländlichen Gebieten profitierten davon.

Software aus der Wirtschaft wird für die Administration in Krankenhäusern ab den frühen 1970er Jahren adaptiert. Seitdem ist laut Bourke [Bou94] die Gesetzgebung ein Motor für den weiteren Einzug der IT in das Gesundheitswesen.

Mit den Personalcomputern (PCs) in den 1980er Jahren begann eine nebenläufige Entwicklung weg von den großen monolithischen Systemen in den Rechenzentren hin zu angepasster Abteilungssoftware, die teilweise nur auf einem Arbeitsplatzrechner ausgeführt wurde. Als Gründe für diese Entwicklung ist einerseits zu nennen, dass dies überhaupt erst einmal möglich wurde, und andererseits, dass die Wünsche der Anwender auf Großrechnern in Rechenzentren nicht so einfach und vor Allem nicht schnell genug umgesetzt werden können. Dieser Anwendungsstau ist bekannt als Softwarekrise: Häufig ist die eingesetzte Software nicht mehr veränderbar ohne nicht gewollte Nebeneffekte zu produzieren, da sie einer veralteten Softwaretechnologie folgt (schlecht strukturiert, systemnah, siehe auch [Han93]) und dafür zu

komplex geworden ist. Die Anwender lösten diesen Anwendungsstau für sich durch den Einsatz von Personalcomputer.

Auch immer mehr Geräte aus der Medizintechnik werden zu eigenständigen Computern und sind somit prinzipiell in der Lage, ihre erhobenen Daten weiterzureichen. Wie Schadow [Sch00] 1999 feststellt, verhinderten zu dieser Zeit häufig proprietäre Schnittstellen aus rein wirtschaftlichen Gründen, dem Fernhalten der Konkurrenz, die produktive Nutzung.

Diese Entwicklung brachte einige Probleme mit sich. Gerade in der Medizin muss spezialisiertes Wissen des medizinischen Personals im gesamten Krankenhaus zur Verfügung stehen. Dies stellt ein Ziel der Integration dar. Die Situation zu der Zeit war dagegen „IT-Wildwuchs“ [Han93], der Insellösungen förderte:

- Medienbrüche zwischen Abteilungen bis hin zu Medienbrüchen zwischen Anwendungen auf einem PC
- Keine einheitliche Netzwerktechnologie
- Prozessredundanz
- Keine (standardisierten) Schnittstellen
- Keine zeitnahe Verfügbarkeit der Information

1.1 Motivation

Heute sind all diese Punkte keinesfalls gelöst, oft ist die Entwicklung der medizinischen EDV der letzten 40 Jahre immer noch zu erkennen. Daraus resultiert meine Motivation, im Rahmen einer Studienarbeit diesen Zustand zu verbessern.

Ein Medienbruch [Sch00] kennzeichnet sich durch die Isolation von Prozessen, die zum Teil redundant ausgeführt werden. Das bedeutet einen aufwendigeren Kommunikationsweg. Indem versucht wird, den Medienbruch auszugleichen, werden verschiedene Kommunikationswege gekoppelt, um die Information vom Sender zum Empfänger zu übertragen. Dabei muss immer auf die Eigenschaften der einzelnen Kommunikationsteilnehmer Rücksicht genommen werden. Dies ergibt nicht nur eine zusätzliche Belastung der daran beteiligten Kommunikationspartner, sondern auch eine höhere Fehleranfälligkeit, verringerte Zuverlässigkeit, indirekter, asynchroner Zugriff auf die Daten. Die Gewährleistung des Datenschutzes und die Klärung der

Verantwortlichkeit bei falscher Information sind weitere Probleme. Immerhin sind dies meist die „persönlichsten“ Informationen eines Menschen und falsche Eingaben können lebensbedrohend sein.

1.2 Umfeld

Das im Helios Klinikum Berlin-Buch eingesetzte kardiologische Befunddokumentationssystem Heartware (siehe Kapitel 3.2.1 kann in seiner aktuellen Form nicht den Anspruch als Teil eines integrativen Krankenhausinformationssystems gerecht werden. Heartware besitzt zwar Schnittstellen zum importieren und exportieren, diese werden allerdings nicht genutzt. Somit finden wir hier den oben beschriebenen Medienbruch. Nur für Benutzer des kardiologischen Systems sind Befunde direkt und sofort verfügbar. Mangels einer Schnittstelle für Aufnahme, Entlassung und Verlegung (ADT) müssen diese Informationen in beteiligten Systemen redundant eingegeben werden. Dies kann zu falschen oder fehlenden Daten führen, besonders wenn der Mensch als Kommunikationsteilnehmer eintreten muss.

Das führende System im HELIOS Klinikum Berlin ist SAP[®]IS-H*Med (siehe Kapitel 3.1). Um Daten der Aufnahme, Verlegung und Entlassung (ADT) in diesem und in Heartware konsistent zu halten, wird eine unidirektionale Schnittstelle von SAP nach Heartware benötigt. Dadurch übernimmt Heartware die Rolle eines Sub-Systems. Welche Möglichkeiten sich anbieten und wie eine Beispielimplementierung dieser Schnittstelle aussieht, ist Teil dieser Studienarbeit.

Kapitel 2

Das

Krankenhaus-Informationssystem

Krankenhausinformationssysteme (KIS) sind nach Haas [Haa05] komplexe medizinische Informationssysteme, die aus einem Patientendatenverwaltungssystem, einem administrativen Informationssystem und einem Medizinischen Informationssystem bestehen. Die Patientendatenverwaltung beinhaltet alle Funktionen für die Verwaltung der Patientendaten, die notwendig sind zur Abrechnung und Erfüllung der gesetzlichen Nachweispflichten. Dazu gehören Funktionen wie

- Aufnahme, Verlegung, Entlassung
- Leistungserfassung (Diagnosen (ICD¹), Leistungen (OPS² bzw. DRG³))
- Bettenbelegungsplanung und
- Abrechnung.

¹ICD steht für International Classification of Diseases and Related Health Problems, zu Deutsch die Internationale Klassifikation der Krankheiten und verwandter Gesundheitsprobleme. Jede Diagnose wird demnach durch einen ICD-Schlüssel repräsentiert. Aktuell wird der ICD-10-Katalog verwendet.

²OPS steht für Operationen- und Prozedurenschlüssel. Die OPS-Klassifikation wird jährlich von der DIMDI (Deutsche Gesellschaft für medizinische Dokumentation und Information) aktualisiert.

³DRG steht für Diagnosis Related Groups, zu Deutsch Diagnosebezogene Fallgruppen. Diagnosen, OPS-Codes und demographische Daten eines Patienten dienen dazu, den Patienten in Fallgruppen einzuordnen. Die Fallgruppen werden über Fallpauschalen abgerechnet.

Das administrative Informationssystem ist für alle Anwendungen der Verwaltung (außer Patientenverwaltung) wie Buchhaltung, Kostenrechnung, Personalwesen und Controlling zuständig. Außerdem bilden logistische Bereiche wie Materialwirtschaft, Technik, Apotheke oder Küche das Aufgabenfeld.

Unter dem Begriff Medizinisches Informationssystem versteht man eine Anwendung, die die Dokumentation und Organisation der medizinischen Abteilungen unterstützt. Dazu zählen

- Fachabteilungsinformationssysteme (z.B. Radiologie)
- Leistungsstelleninformationssysteme (z.B. Labor)
- Ambulanzinformationssysteme
- Leistungskommunikationssysteme
- Pflegeinformationssystem
- Kommunikationsserver und
- Querschnittsanwendungen.

Weitere Definitionen sollen zeigen, dass unterschiedliche Ansichten über die Funktion eines KIS existieren.

Nach Prokosch [Pro94] ist ein „Krankenhaus-Informationssystem ein Krankenhauskommunikationssystem, dessen Funktionalität um informationsverarbeitende und wissensverarbeitende Funktionen erweitert wurde“. Als Beispiele für Informations- und Wissensverarbeitung führt er an:

1. leistungsfähige Informationsretrievalfunktionen, die über die jeweiligen Informationen einer einzelnen Abteilung hinausgehen
2. vollständig integrierte Informationsbanken
3. Funktionen des Entscheidungsmonitoring und
4. Funktionen der Entscheidungsunterstützung

Dugas und Schmidt [DS03] berufen sich 2003 auf die Definition von Haux 1998: „Ein Krankenhausinformationssystem ist das Teilsystem eines Krankenhauses, welches alle informationsverarbeitenden (und -speichernden) Prozesse und die an ihnen beteiligten menschlichen und maschinellen Handlungsträger in ihrer informationsverarbeitenden Rolle umfasst“ und enthalten schon nicht mehr die Funktionen des Entscheidungsmonitoring und der Entscheidungsunterstützung.

Zusammenfassend lässt sich sagen, dass ein KIS nicht gekauft werden kann, sondern aus der Gesamtheit der Informationssysteme und deren integrativer Kopplung ein einzigartiges, einem evolutionärem Prozess [Lan03] unterworfenes Krankenhausinformationssystem entsteht.

2.1 Funktionen eines KIS

Die Deutsche Forschungsgemeinschaft (DFG) hat in einem Anforderungskatalog [Hau01] die Hauptaufgaben für die Informationsverarbeitung im Krankenhaus zusammengefasst. Danach umfasst die Informationsverarbeitung die folgenden Funktionen:

- Patientenbehandlung
- Führen der Krankenakte
- Arbeitsorganisation und Ressourcenplanung
- Krankenhausmanagement
- Forschung und Lehre

Die Patientenbehandlung wird durch ein Patientenmanagementsystem unterstützt, das alle Teilschritte von Aufnahme bis Entlassung beinhaltet, bereitgestellt auf einem klinischen Arbeitsplatzsystem (KAS). Für alle Beteiligten an der Behandlungsplanung (Ärzte, Pflege, Patient) müssen die erforderlichen Informationen gesammelt und elektronisch vorgehalten werden. Ebenfalls erforderlich für die Leistungsabrechnung ist die Leistungsdokumentation.

Die Krankenakte umfasst die Basisdokumentation, die einheitlich im Krankenhaus gestaltet sein soll. Des Weiteren werden in der Fachdokumentation an die jeweilige Fachabteilung angepasste Formulare verwendet. Für spezielle Sichten wie klinische Studien enthält die Krankenakte eine Spezialdokumentation. Die Aufgabe der Informationsverarbeitung in der Arbeitsorganisation und Ressourcenplanung ist die Koordinierung der Mitarbeiter und die Zuteilung zu Räumen, OP-Sälen, Betten oder Geräte. Außerdem ist die Terminplanung sowie die Logistik wie u. a. Bestellsysteme (Medikamente, Speisen) Gegenstand.

Das Krankenhausmanagement kann durch das Controlling unterstützt werden. So werden Statistiken und Berichte bereitgestellt, die auf dem Datenbestand der Informationsverarbeitung beruhen.

Dieser Datenbestand ist auch für die Forschung und Lehre wichtig. Er kann auch für die Durchführung von klinischen Studien genutzt werden.

2.2 KIS-Architekturen

Generell ist bei der Planung eines KIS anstatt der funktionsorientierten auf die date-norientierte Sichtweise zu setzen: Heute bestimmt der Informationsinhalt den Wert eines Informationssystems. Die Daten sollen beständig gegen die (relativ) kurzzeitig eingesetzten Rechnersysteme und Anwendungsfunktionen sein, die zudem noch stärkeren Veränderungen unterworfen sein können [Sch00].

Außerdem sollte in der Designphase keine zu starke Simplifizierung der Arbeitsabläufe und Kommunikationsprozesse stattfinden. Die Gesamtheit der (eingegrenzten) Sichtweisen der Mitarbeiter muss berücksichtigt werden, denn diese Sichtweisen ergeben schließlich das Gesamtbild des KIS.

Nach Prokosch [Pro01] können drei Konzepte von Architekturen für ein Krankenhausinformationssystem (KIS) unterschieden werden:

1. Das monolithische System: Ein kommerzieller Anbieter stellt ein System bereit, welches möglichst umfassend die Funktionalität eines KIS liefert. Ein solches System existiert nicht, auch wenn Anbieter immer mehr administrative und medizinische/klinische Funktionalitäten in ihre Produkte integrieren.
2. So kann das Konzept der heterogen verteilten Systeme genutzt werden, die Schwäche der fehlenden vollständigen Integration durch zugeschnittene medizinische Fachabteilungsinformationssysteme („Best-of-Breed“-Systeme) auszugleichen. Best-of-Breed-Systeme stellen ausgereifte Produkte zu angemessenen Kosten dar, die sich in einem heterogen verteilten System einfach gegen neue bessere Systeme austauschen lassen. Ein Kommunikationsserver übernimmt die Aufgabe, zwischen inkompatiblen Schnittstellen Nachrichten zu transformieren. Will man bei diesem Konzept die Synchronisierung der aktuellen Da-

ten erreichen, wird eine Integrationsplattform mit einem Hintergrundspeicher benötigt, der die Inhalte aller kommunizierten Nachrichten vorhält [Wen05].

3. Der dritte Ansatz ist es, ein KIS basierend auf Komponenten aufzubauen. Diese Komponenten werden aus einer Art Baukastensystem von abgrenzbaren wieder verwendbaren Einzelkomponenten zusammengesetzt. Welche Grenzen das sind, kann in der technischen Anforderung, in der wirtschaftlichen Sicht oder aus der medizinischen Sicht begründet sein. Die Funktionalität der Einzelkomponenten wird über standardisierte APIs bereitgestellt.

Allen Konzepten gemein ist die per Definition des Krankenhausinformationssystems geforderte Integration. Ob diese sich nur über die patientenbezogenen Daten erstreckt oder wie von Lenz und Schmücker [LS03] gefordert auch die funktionale Integration, die Präsentationsintegration und die Ablaufintegration umfassen, ist abhängig vom Konzept.

Funktionale Integration bedeutet, dass für eine „Reihe redundant entwickelter Querschnittsfunktionen (z.B. Autorisierung, Patientenidentifikation)“ [LS03] klare Funktionsgrenzen abgestimmt werden müssen. Dies kann nur im Rahmen einer umfassenden Systemarchitektur geschehen und ist daher bei dem Konzept der heterogen verteilten Systeme schwerer anzuwenden. Aber auch dafür sind Lösungen wie die CORBAMED-Initiative entwickelt worden. Standardisierte Dienste sollen die funktionale Überschneidung reduzieren. In der Praxis werden Dienste wie „patient identification service“ zwar angewendet, aber eher im Umfeld des Komponentenkonzepts.

Die Präsentationsintegration bedeutet, dass der Benutzer gefühlt nur mit einem System im KIS arbeitet. Dafür ist es notwendig, dass der Benutzer sich nur einmal anmelden braucht, und die Präsentation ein einheitliches „look & feel“ besitzt, auch wenn im Hintergrund mit verschiedenen Komponenten interagiert wird. In einer verteilten Umgebung, wie es die beiden letzten Konzepte skizzieren, muss bei einem Kontextwechsel zwischen gleichzeitig genutzten Komponenten ein Kontextmanagement die Synchronisation der Daten vollziehen.

Als Ablaufintegration wird die Vorgehensweise bezeichnet, die Ausrichtung der Informationssystemkomponenten den Anforderungen der Behandlungs- oder Geschäftsprozessen unterzuordnen. Optimal wäre eine Trennung von Ablaufspezifika-

tion und Anwendungslogik, um schneller und genauer auf Veränderungen im Ablauf zu reagieren.

Hilfreich ist es, wenn Einfluss auf die strategische Planung des Gesamtsystems genommen werden kann. Dann ist es möglich, ein führendes System zu bestimmen, weiterhin können mehrfach vorhandene Funktionen in verschiedenen Informationssystemen vermieden werden. Als Ergebnis wird die Datenredundanz vermindert und die Notwendigkeit zu Synchronisierung minimiert. Damit ist die Datenkonsistenz gewährleistet [LS03]. Kann der Arbeitsablauf in einem Krankenhaus, also die Behandlungs- und Geschäftsprozesse gut in die IT-Struktur integriert werden, fördert das, genauso wie die einheitliche Präsentation des KIS, den Bedienungskomfort des Benutzers.

Wenn man die Integration in den Griff bekommt, sind die Verteilung und die Heterogenität durchaus wünschenswert. Durch Verteilung wird eine bessere Ressourcenauslastung erreicht. Als Beispiele sind Netzwerkdrucker oder Datei-Server zu nennen. Wenn im Integrationsschema berücksichtigt, können auch Netzwerkdatenbanken für die zentrale Datenhaltung existieren.

Die Verteilung unterstützt die Anpassung an die Organisationsstruktur durch Rightsizing, den Abbau von Überkomplexität. Verantwortung kann in die betreffende Abteilung abgegeben (Downsizing) oder Spezialaufgaben an abteilungsübergreifender Stelle gebündelt werden (Upsizing) [Sch00].

Die Heterogenität ermöglicht die Anschaffung von neuen Komponenten nach selbst gewählten Kriterien, wie geeignetst oder günstigst. Wie oben in der Konzeptvorstellung erwähnt, kann diese Komponente später einfach ausgetauscht werden, sollte sie veränderten Anforderungen nicht mehr gerecht werden.

Diese Anforderungsanpassungen können für neue diagnostische und therapeutische Verfahren, organisatorische Veränderungen, Krankenhauszusammenschlüsse oder für sich ändernde gesetzliche Rahmenbedingungen erforderlich sein. Daher sind als Kriterien für die Qualität einer Integrationslösung zum einen die Fähigkeit zur Weiterentwicklung und zum anderen das Einfügen in eine übergeordnete KIS-Architektur zu nennen.

Da es in einem Krankenhaus um die Gesundheit des Menschen geht und dies oft von einer schnellen Verfügbarkeit von Informationen abhängt, hat die Kommunika-

tion in einem Krankenhaus einen sehr hohen Stellenwert. Im nächsten Kapitel steht sie daher im Mittelpunkt.

2.3 Krankenhauskommunikation

Krankenhauskommunikation heißt nicht rechnergestützte Kommunikation. Rechnergestützte Kommunikation über ein Netzwerk stellt lediglich einen möglichen Kommunikationskanal dar. Es werden einige weitere Kommunikationsformen neben der rechnergestützten in einem Krankenhaus genutzt, teils traditionell, teils um Mängel in anderen Kommunikationsformen ausgleichen zu können. Dies sei nur hier erwähnt, Gegenstand ist im Folgenden die medizinische Datenkommunikation.

Grundsätzlich können Signale oder Zeichen in ihrer Syntax nicht die Bedeutung, also die Semantik übertragen, über diese müssen sich Sender und Empfänger einig sein. In der medizinischen Kommunikation ist daher der Code in den Klassifikationen z.B. für Krankheiten (ICD(-10)) oder Leistungen/Prozeduren in der Medizin (ICPM) festgehalten.

Um medizinische Informationssysteme zu koppeln, müssen diese sich also verstehen. Umso mehr Systeme miteinander kommunizieren müssen, desto wichtiger ist ein ausgereifter Kommunikationsstandard, der die Wartung und individuelle Anpassung der Schnittstellen der einzelnen Systeme minimiert. Nach Schadow [Sch00] existieren zwar verschiedene Kommunikationsstandards, diese decken aber nicht unbedingt alle Bereiche des medizinischen Feldes ab. DICOM zum Beispiel wird nur in der Diagnostik verwendet, EDIFACT nur in der Administration. Außerdem muss der Standard bereits von der Phase der Definition zur Phase der Anwendung überschritten sein. Ein proprietärer „Standard“ (z.B. SAP HCM) kann auch nicht der Kommunikationsstandard der Wahl sein, da nicht erwartet werden kann, dass alle Hersteller diesen in ihr Produkt integrieren.

So schreibt Schadow: „Der einzige Standard, der den genannten Kriterien zu entsprechen scheint, ist HL7.“

2.3.1 Kommunikationsserver

In einem Krankenhaus existieren häufig historisch gewachsene IT-Strukturen neben Neuanschaffungen. Es ist notwendig, ein heterogen verteiltes Gesamtsystem zu bilden, so dass die Informationssysteme in organisierter Weise miteinander kommunizieren und ein KIS ausbilden können.

Krankenhauskommunikation in organisierter Weise bedeutet, dass diese hilft, die möglichst optimale Integration der patientenbezogenen Daten zu erreichen, auch wenn das von der Kommunikation alleine nicht abhängt (siehe Kapitel 2.2).

Eine Lösung, elektronische Kommunikation zwischen den verschiedenen Informationssystemen (IS) in einem Krankenhaus zu ermöglichen, besteht darin, die Kommunikation wenigstens zu einem, dem Kommunikationsserver (KS) zu ermöglichen. Er dient also dem Umstand, dass keine vollständig standardisierten Kopplungsprotokolle die direkte Kommunikation durch einen gemeinsamen Kanal ermöglichen. In einem heterogen verteilten KIS können durch den Einsatz eines KS einfacher einzelne Informationssysteme ersetzt werden, da nur die Kommunikation zu der Kommunikationsdrehscheibe neu implementiert werden muss. Die so entstehende Bus- oder Sternstruktur ist in der Grafik 2.1 abgebildet und zeigt, dass der KS dem Konzept der Enterprise Application Integration (EAI) folgt.

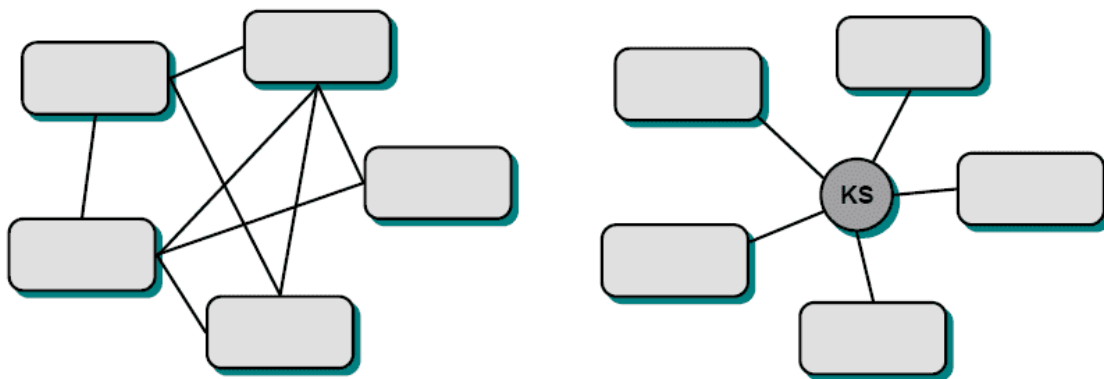


Abbildung 2.1: Kommunikationsserver fördern die Bus- oder Sternstruktur eines KIS (rechts), aus [Lan03].

Um auf Schnittstellen der IS zugreifen zu können, stellt der KS Kommunikationsagenten bereit, die die Kopplung auf Transport- (z.B. TCP oder Samba) und

Anwendungsschicht (z.B. HL7) ermöglichen. Wie in Abbildung 2.2 zu sehen, können so auch IS miteinander kommunizieren, die zum einen dateilose socketbasierte Schnittstellen anbieten und zum anderen über den Austausch von Dateien in Verbindung treten. Der KS übernimmt die Transformation der Protokolle zwischen den Agenten.

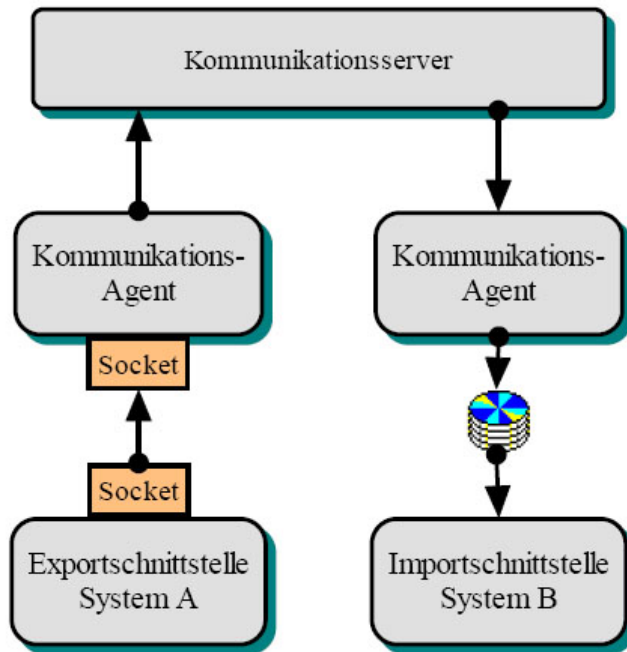


Abbildung 2.2: Socket- oder dateibasierte Kommunikation, aus [Lan03].

Eine Erweiterung des Kommunikationsservers stellt die Vorhaltung der Patientendaten als zentraler Befundserver [Pro01] dar, in diesem Sinne kann er so als Informationsserver benannt werden.

Prokoschs dritte These, „Kommunikationsserver erleichtern die elektronische Kommunikation, sie lösen nicht das Problem der Integration“ [Pro01] will aber auch die Nachteile eines Kommunikationsservers gegenüber der immer anzustrebenden vollständigen Integration der zu koppelnden Informationssysteme anmahnen. Denn durch ein KS entsteht kein Krankenhausinformationssystem, „lediglich“ ein Krankenhauskommunikationssystem. Es fehlt die semantische Integration [LS03], also eine Einigung darüber, wie die semantische Heterogenität der zu kommunizierenden

Teilsysteme aufgelöst wird. Des Weiteren ist ein Kommunikationsserver nicht von sich aus für die Synchronisation redundanter Datenbestände in der Lage. Ein großer Schritt in der Reduzierung der semantischen Heterogenität stellt der nachrichtenbasierte Standard Health Level 7 (HL7) dar.

2.3.2 HL7

Der Name HL7 („Health Level Seven“) leitet sich aus dem ISO/OSI-Referenzmodell („International Standards Organization/Open System Interconnection“) ab, indem es sich als ein medizinisches Datenkommunikationsprotokoll in der Anwendungsschicht einordnet. Tatsächlich wurde nie eine enge Bindung an das Modell angestrebt, denn die OSI-Forderung, welche aussagt, dass Aufgaben des Protokolls in Module getrennt werden müssen, wird nur teilweise berücksichtigt. Auch die Tatsache, dass sich in der Praxis das TCP/IP-Referenzmodell durchgesetzt hat, ist für HL7 uninteressant, da es nur die (in beiden vorhandene) Anwendungsschicht betrifft⁴.

Tatsächlich handelt es sich bei HL7 um eine Organisation aus Kliniken, Softwarehersteller, Beratungsunternehmen und privaten Mitgliedern. Ihren Ursprung hat das Nachrichtenprotokoll in den USA. Drei bis vier Treffen sind jährlich angesetzt. Das erklärte Ziel von HL7 ist es: „to provide standards for the exchange of data among healthcare computer applications that eliminates or substantially reduces the custom interface programming and program maintenance that may otherwise be required“⁵ (Vgl. [HL788][HL790][HL794][HL797]).

Die HL7-Organisation unterteilt sich in Komitees. Diese Komitees sind für einen Funktionsbereich („functional area“) zuständig und erarbeiten „ihr“ Kapitel („chapter“) der Standard-Dokumentation. Da die Komitees sich durch ihre Kapitel, also

⁴Der Name HL4 wurde scheinbar nie in Erwägung gezogen.

⁵Das erklärte Ziele von HL7 ist es: einen Kommunikationsstandard für das Gesundheitswesen anzubieten, der es ermöglicht, die sonst nötigen Arbeiten an eigenen Schnittstellenentwicklungen und Programmwartungen zu eliminieren oder diese wesentlich zu verringern.

ihrem Spezialgebiet, unterscheiden, stehen die Kapitel stellvertretend für das dahinter stehende Komitee.

Kapitel 1 entspricht keinem Komitee und beinhaltet die Einleitung in Grundsätze und Ziele von HL7.

Kapitel 2 wird von dem „control“-Komitee bearbeitet. Hier werden die Spezialgebiete festgelegt, auch über neue Kapitel wird entschieden.

Im Jahr 1987 erschien die Version 1.0 mit dem ursprünglichen Stamm von 5 Kapiteln. Dieser stellt neben den ersten beiden Kapiteln die Administration von Stamm- und Bewegungsdaten um Aufnahme („admission“), Entlassung („discharge“) und Verlegung („transfer“), kurz ADT (**Kapitel 3**) dar. Ferner findet man Aufträge („order entry“) (**Kapitel 4**) und Anfragen an Informationssysteme („query“), (**Kapitel 5**).

Bereits ein Jahr später, 1988, kamen mit Version 2.0 folgende, als notwendig erachtete Gebiete noch hinzu: Die Abrechnung („finance“) (**Kapitel 6**) und die Befundberichterstattung („observation reporting“), kurz ORU (**Kapitel 7**).

Version 2.1, erschienen am 31.12.1992, enthält keine neuen Gebiete, wohl aber Ergänzungen und substantielle Änderungen.

In Version 2.2, Erscheinungsdatum 01.12.1994, wurde das Kapitel für Codesysteme („master files“) eingeführt (**Kapitel 8**). Das Query-Chapter wurde nach Kapitel 2 verschoben.

Version 2.3 (07.04.1997) enthält vier neue Kapitel sowie Erweiterungen der bestehenden. Dazu zählen:

- Dokumentenverwaltung („medical records/information management“) (**Kapitel 9**),
- Termin- und Ressourcenplanung („scheduling“) (**Kapitel 10**),
- Patientenüberweisung („patient referral“) (**Kapitel 11**) und
- Therapie/Pflege und deren Planung („patient care“) (**Kapitel 12**).

Am 17.05.1999 wurde Version 2.3.1 veröffentlicht, eineinhalb Jahre später, am 01.11.2000 bereits Version 2.4 mit u. a. drei neuen Kapiteln:

- „Laboratory Automation“ (**Kapitel 13**)
- „Application Management“ (**Kapitel 14**)

- „Personnel Management“ (**Kapitel 15**)

Version 2.5 vom 30.08.2003, enthält eine Erweiterung des zweiten Kapitels („control“) (Vgl. [HLS08]).

2.3 sollte die letzte Version vor Version 3 sein, um Ressourcen für den Versionsprung zu bündeln, denn bei Version 3 handelt es sich um einen kompletten Neuentwurf auf objektorientierter Basis. Dieser ist allerdings immer noch nicht abgeschlossen. Heute sind vor allem Version 2.1 und 2.2 verbreitet im Einsatz.

Die Idee hinter dem Nachrichtenformat HL7 ist die lose Kopplung der kommunizierenden Systeme. Das ermöglicht eine Heterogenität der Systeme. Ein Nachteil ist jedoch die geringe Informationsverteilung. Die Nachrichtenanzahl kann aber erweitert werden, was ausgleichend wirkt.

HL7 ist ereignisgesteuert. Tritt eine neue Situation ein, so kann diese interrogativ, indikativisch oder imperativ durch HL7 behandelt werden.

Interrogativ bedeutet, dass eine Situation Information benötigt, die über eine HL7-Anfrage erhalten werden kann (engl. *query*). Indikativ steht für ein eingetretenes Ereignis, dessen Information an anderer Stelle verfügbar gemacht werden soll (engl. *unsolicited update*). Der Imperativ fordert mit der Übermittlung der Information eine bestimmte Handlung auszuführen (engl. *order*) [Sch00].

HL7-Nachrichten folgen dem EDI (electronic data interchange)-Prinzip: Sie sind dem Formular nachempfunden, von Maschinen bearbeitbar und vom Menschen dennoch lesbar. Außerdem wird ein begrenzter Zeichensatz verwendet, wie ihn Fernschreiber benutzten.

HL7 ermöglicht durch die große Anzahl an Ereignistypen, die es untersützt, dass die Semantik der auszutauschenden Information erhalten bleibt. Codierungsregeln sind für die Syntax der Nachricht verantwortlich, die eventgesteuerte Ausführung erfüllt die Pragmatik⁶.

⁶Eine Nachricht wird definiert als eine Einheit ausgetauschter Information in einer bestimmten Form (Syntax), einer bestimmten Bedeutung (Semantik) und zu einem bestimmten Zweck (Pragmatik).

Ein Informationssystem, welches HL7 einsetzt, ist SAP mit der Branchenkomponente Krankenhaus. Dieses soll im nächsten Kapitel vorgestellt werden.

2.4 SAP im Krankenhaus: IS-H und IS-H*Med

SAP wurde im Jahr 1972 in Walldorf, Deutschland gegründet [Mos99]. Am Anfang standen die drei Buchstaben für Systemanalyse und Programmentwicklung, der Fokus war auf die rechnergestützte Finanzbuchhaltung gerichtet. 1988 ging die Firma als *SAP Aktiengesellschaft Systeme, Anwendungen und Produkte in der Datenverarbeitung* an die Börse, seit 1979 war mit dem Produkt R/2 der Funktionsbereich erweitert worden. Im Mai 2005 beschloss das Unternehmen, allein das Kürzel SAP als Firmenname und Logo zu führen. Der heute viertgrößte Softwareentwickler weltweit und Marktführer im Bereich Unternehmenssoftware [Mos99] brachte 1993 sein Produkt R/3 heraus. R/3 unterstützt sowohl zweischichtige als auch dreischichtige Architekturen. Die drei Schichten Datenbank, Applikationsserver und Präsentation sind in Abbildung 2.3 dargestellt. SAP ist damit in der Lage, durch Skalierung der einzelnen Schichten sein Produkt R/3 an die Größe und das Wachstum eines Unternehmens anzupassen.

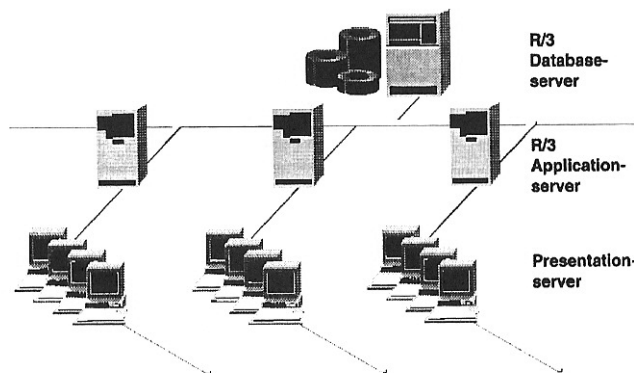


Abbildung 2.3: Gewöhnliches 3-Schichten-Architekturmodell, aus [Mos99].

Die R/3-Funktionen erweiterte SAP um Branchenlösungen, englisch Industry Solution (IS), an dessen Namenssuffix die Branche unterschieden werden kann. Eine Auflistung findet sich auf [help.sap.com]. So gibt es neben dem hier näher betrachte-

ten SAP for Healthcare IS-H für das Krankenhaus bzw. IS-H*Med für das Klinische System auch Lösungen für die Versorgungswirtschaft (IS-U für Utilities) oder auch für das „Campus Management“ einer Hochschule (IS-HER-CM steht für Higher Education & Research Campus Management). Diese Lösungen enthalten zusätzliche für den jeweiligen Einsatzzweck entwickelte Komponenten.

IS-H ist eine „krankenhausspezifische Erweiterung des Standard-R/3-Systems, die das Patientenmanagement, die medizinische und pflegerische Dokumentation und die Patientenabrechnung unterstützt.“ [SAP06]

Des Weiteren umfasst die Branchenkomponente Krankenhaus die Integration in die Standardfunktionen Finanzbuchhaltung, Controlling und Materialwirtschaft sowie einer Kommunikationskomponente für krankenhauserne und -externe Datenübermittlung. Außerdem wird die Auswertung der Patienten- und Krankenhausdaten ermöglicht.

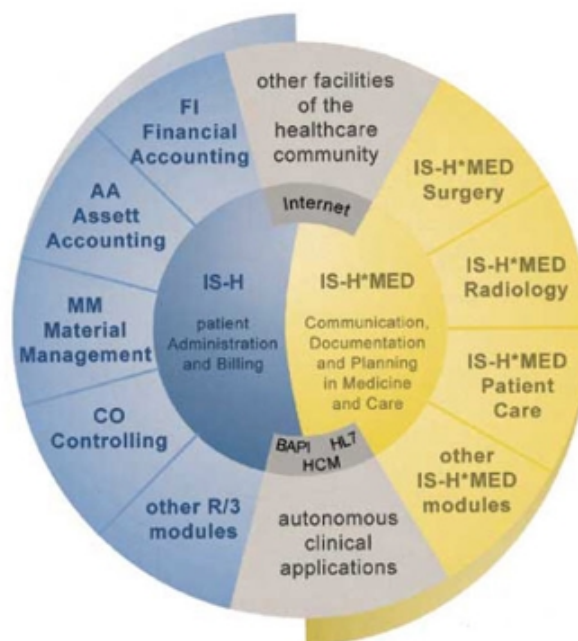


Abbildung 2.4: SAP R/3, IS-H, IS-H*Med und IS-H*Med-Module, aus [GSP⁺03].

Die Abbildung 2.4 zeigt die Beziehungen zwischen SAP R/3-Funktionen (linker Aussenring), IS-H (linke Hälfte des Kerns), IS-H*Med (rechte Hälfte des Kerns) und IS-H*Med-Module (rechter Aussenring).

IS-H*Med wird von T-Systems Österreich und der Berliner Firma GSD, nun Siemens Medical Solutions GSD GmbH [GSD08], seit 1994 entwickelt. Mittlerweile wird in 272 lizenzierten Standorten weltweit IS-H*Med eingesetzt [GSD08]. Die vollständige Integration einer Software von Fremdanbietern in R/3 wird von Seiten der SAP gefördert, so stellt [SAP06] die Dokumentation nicht nur für IS-H, sondern auch für IS-H*med bereit.

Das Basismodul von IS-H*med, in der Abbildung als rechte Hälfte des Kerns dargestellt, erfüllt folgende Funktionalität:

- klinischer Arbeitsplatz mit anpassbaren Sichten
- medizinische patientenbezogene Dokumentation
- Kommunikation von Aufträgen und Befunden zwischen den Funktionsstellen und den Stationen
- Terminplanung für Patient und Abteilung

Die Kommunikationskomponente von IS-H ermöglicht die datei-basierte Nachrichtenübertragung, die im nachfolgenden Kapitel als ADT-Schnittstelle vorgestellt wird. Die datei-lose Kommunikation über Remote Function Call-fähige Funktionsbausteine (engl. RFC-enabled Function Module, RFM) und im Speziellen über die im Kapitel 2.4.2 beschriebene BAPI-Schnittstelle ist bereits Teil des SAP Business Framework. Dieses dient der technischen Integration und dem betriebswirtschaftlichen Datenaustausch zwischen SAP-Komponenten untereinander und zwischen SAP und Nicht-SAP-Komponenten.

2.4.1 Die ADT-Schnittstelle

Mit IS-H wird eine Schnittstelle in SAP eingeführt, welche die Administration von Stamm- und Bewegungsdaten der Patienten ermöglicht. Durch die Schnittstelle können Informationen zur Aufnahme („admission“), Entlassung („discharge“) und Verlegung („transfer“), kurz ADT genannt, eines Patienten mit Subsystemen ausgetauscht werden.

Für die krankenhausinterne Kommunikation wird zum einen der medizinische Nachrichtenstandard HL7 in Version 2.1 unterstützt, zum anderen IS-H Communication Module (IS-HCM) in Version 1.1 und 1.2. IS-HCM ist ein SAP-eigener

Nachrichtentyp [SAP06], dessen Struktur ähnlich aufgebaut ist wie die von HL7-Nachrichten. Zusätzlich wird der Dateitransfer unterstützt [GSP⁺03]. Mit dem Blick auf die Standardisierung von Schnittstellen zur Schaffung eines ganzheitlichen Krankenhausinformationssystems unabhängig vom Hersteller wird daher IS-HCM in dieser Arbeit nicht berücksichtigt und der Fokus auf die HL7-Implementierung gelegt.

Nur für die SAP-interne Kommunikation regelt IS-H den Transport der Nachrichtendatei, für die Kommunikation mit einem Partnersystem innerhalb des Krankenhauses wird gefordert, dass dieses System die in einem definierten Verzeichnis abgelegte Nachrichtendatei selbst abholt.

Health Level 7 (HL7) wurde im Allgemeinen bereits im Kapitel 2.3.2 vorgestellt. Hier soll die Einbindung in IS-H dargestellt werden. Das Nachrichtenformat von HL7 wird auch in IS-H ereignisbasiert eingesetzt, das heißt, ein Ereignis in der realen Welt macht eine Kommunikation nötig. Entweder erfolgt die Kommunikation als Interrogativ, als Indikativ oder als Imperativ. Die von IS-H unterschiedenen Ereignistypen sind in der Tabelle 2.1 aufgeführt.

Ereignistyp	Codierter Ereignistyp
Aufnahme des Patienten	A01
Verlegen des Patienten	A02
Entlassen des Patienten	A03
Änderungen der Daten des Patienten	A08
Ankündigung Patientenaufnahme	A14
Ankündigung Verlegung	A15
Ankündigung Entlassung	A16

Tabelle 2.1: ADT-Ereignistypen in IS-H

In der rechten Spalte stehen die nach Kapitel 3 der HL7-Spezifikation benannten zugehörigen ADT-Formate. Weitere von HL7 definierte Ereignisse werden in IS-H selbst nicht benutzt.

IS-H*Med erweitert die rudimentäre HL7-Unterstützung von IS-H. Nun ist es z.B. möglich, mittels HL7 Befund- und Labordaten (HL7 Kapitel 4) sowie Dokumentverweise zu übernehmen.

Wie die HL7-Nachrichten in eine Datei eingebettet werden, zeigt die Grafik 2.5.

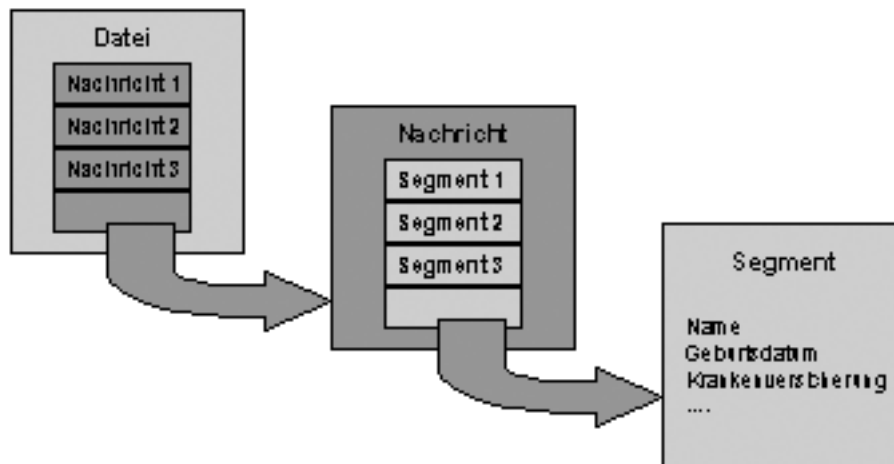


Abbildung 2.5: Aufbau der datei-basierten Nachrichten, aus [SAP06].

In einer Datei können bis zu 1000 Nachrichten enthalten sein, jede Nachricht besteht aus Segmenten wie

- Nachrichtenkopf (engl. Message Header, MSH)
- Patientendaten (engl. Patient identification segment, PID)
- allgemeine Auftragsdaten (engl. Common Order Segment, ORC)
- Ereignistyp (engl. event type, EVN)
- Behandlungsfall (engl. Patient Visit, PV1)
- Bezugspersonen (engl. next of kin, NK1)
- usw.

Welche Segmente in einer Nachricht vorkommen, ist abhängig vom Ereignistyp [HL790].

Das in der Tabelle 2.2 auf Seite 26 vorgestellte Beispiel wird verwendet, um einen Dokumentenverweis bekannt zu geben. Dieser Nachrichtentyp verwendet die Segmente MSH, PID und ORC [SAP06] [HL798].

Die einzelnen Felder innerhalb eines Segments werden durch | getrennt. Enthält ein Feld mehrere Komponenten, werden diese durch ^ geteilt, bzw. durch weitere Zeichen, die im Nachrichtenkopf als Trennzeichen definiert werden. In diesem Fall

sind das die Zeichen "~", "\" und "&". Die Segmente einer Nachricht werden durch CR LF (Wagenrücklauf und Zeilenvorschub) gegliedert.

2.4.2 BAPI

Die Business Application Programming Interfaces (BAPIs) der Kommunikationskomponente von R/3 stellen die dateilose Kommunikation von, nach und innerhalb R/3 dar. 1996 begann SAP, mit dem Business Framework eine objektorientierte Strukturierung der R/3-Funktionalität in Anwendungskomponenten vorzunehmen. Objektorientiert deshalb, da die R/3-Prozesse und Daten in Form von SAP Business Objects (BO) verfügbar gemacht werden sollen [SAP07]. Auf die SAP Business Objects kann über API-Methoden zugegriffen werden. Diese API-Methoden werden im SAP-Jargon Business Application Programming Interfaces genannt. Innerhalb von R/3 werden diese Business Objects im BAPI-Explorer dokumentiert und im Business Object Repository (BOR) angeordnet und definiert. Die Gründe für die Einführung waren [Mos99]:

- Umstrukturierung von R/3 in nicht mehr nur funktionell, sondern auch architektonisch getrennte Komponenten
- BAPIs als Voraussetzung der Komponentenbildung von R/3
- BAPIs ermöglichen Zugang zu R/3 für Mitbewerber, Partner und kundeneigene Programme
- SAPs Versprechen, eine offene, objektorientierte Schnittstellenunterstützung mit langfristiger Pflege der eingeführten BAPIs anzubieten
- Implementierung und Änderungen innerhalb des BO sind transparent
- Möglichkeit der Kopplung der Komponenten auf betriebswirtschaftlicher Ebene

In der Sprache der Objektorientierung werden SAP-Business-Objekttypen als Klasse bezeichnet und deren SAP-Business-Objekte als Instanzen der Klasse. BAPIs stellen als Gesamtheit der Methoden des Objekttyps seine Eigenschaften dar. Attribute sind auch in SAP die durch Methoden zu manipulierende Beschreibung eines SAP-Business-Objekts. Schlüsselfelder stellen die Einmaligkeit der Instanz sicher.

Die Abbildung 2.6 zeigt das Business Objekt Patient, wie es in der Business Object Repository (BOR) hinterlegt ist. Zu jedem Blatt sind detaillierte Informationen verfügbar. Die grün markierten Methoden stellen BAPI-Methoden dar.

BAPI-Methoden stellen also nur einen Teil der RFM dar (jene mit einem Haken), diese wiederum nur einen Teil der gesamten ABAP-Funktionsbausteine (alle Methoden).

Struktur der BOs

SAP-Business-Objekte sind als Entität in Schichten angelegt, wie die nachfolgende Grafik zeigt. Dadurch können Struktur- und Implementierungsdetails verborgen bleiben. Abbildung 2.7 soll dies verdeutlichen.

Der Kern als innerste Schicht stellt die eigentlichen Daten dar.

Die zweite Schicht wird als Integritätsschicht bezeichnet, weil sie die Integrität des Methodenaufrufs prüft. Dies „umfasst Geschäftsregeln („Business Rules“) zur konsistenten Einbettung in das Umfeld und Einschränkungen („Constraints“) bezüglich Werten und Domänen, die für den Business-Objekttypen gelten“ [SAP07]. Zum Beispiel kann nicht an einen Kunden verkauft werden, zu dem keine Informationen zur Geschäftsbeziehung vorliegen.

In der nach außen hin folgenden Schnittstellenschicht wird die Struktur und Implementierung der BOs beschrieben und die Methoden, die BAPIs, für den Zugriff definiert. Der Zugriff kann sowohl von einem Fremdsystem wie auch von einer Komponente erfolgen.

In der äußersten Schicht, der Zugriffsschicht, werden die für einen externen Zugriff einsetzbaren Technologien definiert.

Durch diese Schichtung ermöglichen BAPIs eine Integration auf betriebswirtschaftlicher und nicht auf technischer Ebene. Es entsteht eine größere Stabilität der Koppelung, unabhängig von der eingesetzten Kommunikationstechnologie.

2.4.2.1 Interner und externer Zugriff

Der interne Zugriff bedeutet, dass Komponenten innerhalb des SAP-Systems BAPIs nutzen. Für den externen Zugriff auf BAPIs werden in der Zugriffsschicht mehrere Kommunikationstechnologien definiert. Neben Microsofts Component Object

Model/Distributed Component Object Model (COM/DCOM) ist auch eine Anbindung mittels CORBA (Common Object Request Broker Architecture) möglich. Für den Zugriff aus Java heraus werden entweder Business-Objekte als BAPI-Klassenbibliotheken zur Verfügung gestellt, oder es werden in der Entwicklungsumgebung Proxy-Klassen der BAPI-Methoden angelegt [Sch02]. Metadaten und Datentypen dazu liefert das BOR. Der folgende Code stellt einen Auszug aus dem ABAP-Quelltext der Funktion BAPI_PATIENT_SEARCH dar.

Listing 2.1: Auszug aus dem ABAP-Quelltext der Funktion BAPI_PATIENT_SEARCH

```

1 FUNCTION BAPI_PATIENT_SEARCH.
2 * "
3 * " " Lokale Schnittstelle:
4     IMPORTING
5         VALUE(INSTITUTION) LIKE BAPI1084SRCH-INSTITUTION
6         VALUE(FILTER_PATIENTID) LIKE BAPI1084SRCH-PATIENTID
7         OPTIONAL
8         VALUE(FILTER_PATCASEID) LIKE BAPI2099DATA-PATCASEID
9         OPTIONAL
10        VALUE(FILTER_LAST_NAME_PAT) LIKE BAPI1084SRCH-
11        LAST_NAME_PAT OPTIONAL
12        VALUE(FILTER_FRST_NAME_PAT) LIKE BAPI1084SRCH-
13        FRST_NAME_PAT OPTIONAL
14        VALUE(FILTER_BIRTH_NAME) LIKE BAPI1084SRCH-BIRTH_NAME
15        OPTIONAL
16        VALUE(FILTER_DOB_FROM) LIKE BAPI1084SRCH-DOB OPTIONAL
17        VALUE(FILTER_DOB_TO) LIKE BAPI1084SRCH-DOB OPTIONAL
18        VALUE(FILTER_SEX_EXT) LIKE BAPI1084SRCH-SEX_EXT OPTIONAL
19        VALUE(FILTER_SSN) LIKE BAPI1084SRCH-SSN OPTIONAL
20        VALUE(FILTER_EXT_PAT_ID) LIKE BAPI1084SRCH-EXT_PAT_ID
21        OPTIONAL
22        VALUE(FILTER_DOC_NO) LIKE BAPI1084SRCH-DOC_NO OPTIONAL
23        VALUE(SEARCH_PHONETIC) LIKE BAPINALL-XIND OPTIONAL
24        VALUE(MAXCNT) LIKE BAPINALL-MAXCNT DEFAULT -1
25     EXPORTING
26         VALUE(WORST_RETURNED_MSGTY) LIKE BAPINALL-WORSTREMSG
27     TABLES
28         PATIENTS STRUCTURE BAPI1084SRCH

```

```

23          RETURN STRUCTURE BAPIRET2 OPTIONAL
24 * //
25 * ****
26 * 07.06.99 4.03 new.
27 * ****

```

Im Fall des Business Objekt Patient mit der BAPI Patient.Search sind dies auf der Exportseite die Tabellen Patients und Return, sowie ein einzelner Rückgabeparameter, der den schlimmsten aufgetretenen Nachrichtentyp (WORST_RETURNED_MSGTY) beinhaltet. Die für die Abfrage in SAP zu importierenden Filterparameter können ebenfalls in einer Proxy-Klasse vorgehalten werden.

2.4.2.2 SAP Java Connector

Für Kunden und Partner der SAP ist es möglich, eine Programmbibliothek zu nutzen, die den Zugriff auf SAPs Remote Function Call-Protokoll bzw. den BAPIs ermöglicht. Der SAP Java Connector (JCo) basiert auf dem Java Native Interface (JNI). Damit trotzdem eine Art Java-übliche Plattformunabhängigkeit gewahrt bleibt, wird JCo für viele Plattformen wie Win32, Linux, Solaris, AIX, HP-UX, Tru64, OS/390 und Linuxs390x angeboten. Die Dokumentation von JCo befindet sich auf dem FTP-Server <ftp://ftp.sap.com/pub/sapjco/>.

JCo bietet Middlewarefunktionen für die transparente Kommunikation zwischen einer externen Anwendung und dem R/3. Es verwendet das RFC-Schnittstellensystem, um RFC-enabled Function Modules (RFM) aufzurufen. Unter die RFMs fallen auch die BAPIs. Der Java Connector unterstützt drei Arten von RFCs: synchron, transaktional oder „queued“ [Sch02].

RFMs können drei Arten von Parametern besitzen. Import-Parameter (der Client sendet diese an die RFM), Export-Parameter (RFM sendet diese zurück zum Client) und bidirektionale Tabellen. Import und Export können als einfache Felder oder auch als Strukturen (mehrere einfache Felder) vorliegen. Tabellen sind im Prinzip nur erweiterte Felder mit zusätzlichen Spaltenbezeichnungen. Im BOR und im BAPI-Explorer ist dokumentiert, welche Parameter optional und welche Pflicht sind. Exportparameter sind immer optional. Eine BAPI-Methode wird durch das JCo.Function-Objekt repräsentiert. Um mehrere BAPI-Methoden zu verwalten, wird das JCo.Repository-Objekt verwendet. Es repräsentiert das BOR in JCo.

Das `JCO.CreateClient`-Objekt stellt die Verbindung zum SAP-System her und ruft die BAPI-Methode auf.

Nach erfolgreichem Aufruf können die Parameterlisten des `JCO.Function`-Objekts (`import`, `export`, `tables`) über die Methoden

- `getImportParameterList`
- `getExportParameterList`
- `getTableParameterList`

geholt werden.

Innerhalb der Listen wird auf die einzelnen Parameter mit ihrem Namen und ihrem Datentyp zugegriffen.

Strukturen werden durch ein Objekt `JCO.Structure` repräsentiert. Die Methode `getStructure` ermöglicht den Zugriff auf alle mit Namen und Datentyp bekannten Strukturparameter.

Mit der Methode `JCO.Table.getTable` wird ein `JCO.Table`-Objekt erzeugt. Einzelne Werte werden über `String getString(„Name“)` angesprochen.

`RETURN` ist ein optionaler Parameter, der als Struktur oder Tabelle Meldungen übergibt, die anstelle von `Exceptions` über den Erfolgsstatus des BAPI-Aufrufs informieren. Als Struktur wird er über `getExportParameterList` angesprochen, als Tabelle über `getTableParameterList`.

Da SAPs Programmiersprache ABAP andere Datentypen verwendet, muss eine Abbildung auf Java-Datentypen stattfinden. Tabelle 2.3 zeigt die unterschiedlichen Datentypen und das zugehörige Mapping.

In Kapitel 4 wird der SAP Java Connector für das Ausführen einer BAPI-Methode genutzt.

Nachricht	Stelle	Typ	Länge	Bezeichnung	Pflichtfeld	Wert
MSH						
	1	ST	1	Field-Separator	Ja	= Feldtrennzeichen
^^\&\	2	ST	4	Encoding characters	Ja	^^\& = Weitere Trennzeichen
KISALT	3	ST	8	Sending application	Ja	sendendes System
<SAP-MANDANT>	5	ST	3	Receiving application	Ja	SAP-Mandant
<IS-H Einrichtung>	6	ST	4	Receiving facility	Ja	ISH-Einrichtung
	10	ST	20	Message control ID		Meldungsnummer
PID						
	3	CM	10	Patient ID (internal)	(Entw.)	IS-H Patienten ID
<IS-H Fallnummer>	4	ST	10	Alternate Patient ID	(Oder)	IS-H Fallnummer
ORC						
DT	1	ID	2	Order control	Ja	DT (Kennzeichen Dokumentverweis)
	2	CM	14	Placer order number		eindeutige Anforderungsnummer <Anforderung(8)> ^ <Anforderungstyp(6)>
	3	CM	14	Filler order number		Auftragsnummer des externen Systems
	5	ID	2	Order Status		Status des Dokuments (s. Statusverarbeitung)
<dokumentierendeOE>	10	CN	8	Entered by	Ja	IS-H*MED dokumentierende OE
<Mitarbeiter>	11	CN	10	Verified by	Ja	IS-H*MED verantwortlicher Mitarbeiter
	12	CN	8	Ordering provider		IS-H*MED anfordernde pfleg. OE
<Dokumentdatum>	15	TS	14	Order effective date/time	Ja	Dokumentendatum, Format: YYYYMMDDHHMMSS
<Dateiname>	16	CE	70	Order control code	Ja	Dokumentenpfad, variabler Teil
<DokTyp-Name>^<Version>	18	CE	13	Entering device	Ja	Dokumenttyp(10) ^ DokumenttypVersion(3)
<Betreff>	19	XCN	40	Action by		Beschreibung (Betreff) des Dokuments

Legende der benutzten Datentypen:

ST - Zeichenkette (String); CM - Gemischt (Composite); ID - Identifikator für HL7-Tabelle

CN - Gemischte ID (Composite ID number and name); TS - Zeitstempel (Time Stamp)

CE - kodiertes Element (Coded element)

XCN - Erweiterte gemischte ID (Extended composite ID number and name)

Tabelle 2.2: Beispiel Dokumentenverweis

Business Objekt Patient		✓ HC Patient
Interfaces		
IFSAP	SAP Standard Interface	
Schlüsselfelder		
Patient.PatientId		✓ IS-H: Patientennummer
Attribute		
Patient.ObjectType	Objekttyp	
Patient.Sex		IS-H: Geschlechtskennzeichen - intern
Patient.StandLastName		✓ IS-H: standardisierter Nachname
Patient.StandardFirstName		✓ IS-H: standardisierter Vorname
Patient.DateOfBirth		✓ IS-H: Geburtsdatum
Patient.BirthName		✓ IS-H: Geburtsname
Patient.Nationality		✓ IS-H: Staatsangehörigkeit
Patient.PostalCode		✓ IS-H: Postleitzahl Patient
Patient.City		✓ IS-H: Wohnort Patient
Patient.VIP		✓ IS-H: VIP-Kennzeichen
Patient.ExtPatientID		✓ IS-H: Externe Patientenidentifikation
Patient.LastName		✓ IS-H: Nachname Patient
Patient.FirstName		✓ IS-H: Vorname Patient
Patient.Institution		✓ IS-H: Einrichtung
Methoden		
Patient.ExistenceCheck	Existenz des Objekts prüfen	
Patient.Search		✓ IS-H BAPI Patient.Search - Patient suchen
Patient.Enqueue		✓ IS-H BAPI Patient.Enqueue - Patient sperren
Patient.Dequeue		✓ IS-H BAPI Patient.Dequeue - Patient entsperren
Patient.Create		✓ IS-H BAPI Patient.Create - Patient anlegen
Patient.GetDetail		✓ IS-H BAPI Patient.GetDetail - Patient Details anzeigen
Patient.Change		✓ IS-H BAPI Patient.Change - Patient ändern
Patient.Cancel		✓ IS-H BAPI Patient.Cancel - Patient stornieren
Patient.GetCaseList		✓ IS-H BAPI PatientCase.GetList - Fälle zum Patient
Patient.Display		✓ Patientenstamm in Transaktion anzeigen
Patient.ChangeTransaction		✓ Patientenstamm in Transaktion ändern
Patient.DisplayCaseOverview		✓ Fallübersicht des Patienten anzeigen
Patient.DisplayBillingInformation		✓ Rechnungen des Patienten anzeigen
Ereignisse		
Patient.Create_Patient		✓ Patientenstammsatz angelegt
Patient.Change_Patient		✓ Patientenstammsatz geändert
Patient.Cancel_Patient		✓ Patientenstammsatz storniert
Patient.DoChange		✓ Patientendaten sind zu ändern

Abbildung 2.6: Business Objekt Patient, Bildschirmfoto aus dem SAP BOR.

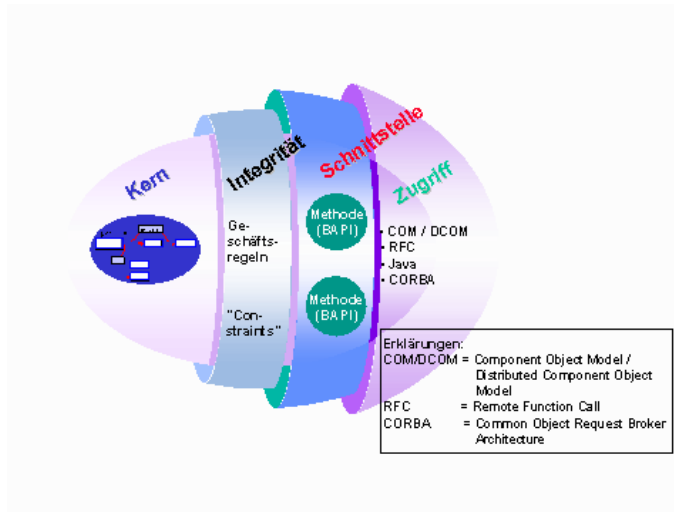


Abbildung 2.7: Struktur der Business Objekte, aus [SAP07].

ABAP Type	Description	Java Data Type	JCo Type Code
b	1-byte integer	int	JCO.TYPE_INT1
s	2-byte integer	int	JCO.TYPE_INT2
I	4-byte integer	int	JCO.TYPE_INT
C	Character	String	JCO.TYPE_CHAR
N	Numerical Character	String	JCO.TYPE_NUM
P	Binary Coded Decimal	BigDecimal	JCO.TYPE_BCD
D	Date	Date	JCO.TYPE_DATE
T	Time	Date	JCO.TYPE_TIME
F	Float	double	JCO.TYPE_FLOAT
X	Raw	data byte[]	JCO.TYPE_BYTE
g	String (variable-length)	String	JCO.TYPE_STRING
y	Raw data (variable-length)	byte[]	JCO.TYPE_XSTRING

Tabelle 2.3: Mapping von ABAP, Java und JCo, aus [Sch02].

Kapitel 3

KIS im HELIOS Klinikum

Berlin-Buch

Wie im Kapitel 2 ausführlich beschrieben, sind Krankenhausinformationssysteme (KIS) komplexe medizinische Informationssysteme, die aus einem Patientendatenverwaltungssystem, einem administrativen Informationssystem und einem medizinischen Informationssystem bestehen.

Im Klinikum Buch wird SAP eingesetzt, welches die administrativen Aufgaben eines Krankenhauses übernimmt. Die Branchenkomponente IS-H erlaubt die Patientendatenverwaltung und das damit verknüpfte Modul IS-H*Med ermöglicht die medizinische Dokumentation. SAP übernimmt dabei die führende Rolle, der sich alle anderen medizinischen Informationssysteme unterordnen.

Wird in dieser Arbeit nicht weiter differenziert von SAP gesprochen, so ist damit die Gesamtheit der im Klinikum Buch eingesetzten SAP-Komponenten, inklusive IS-H und IS-H*Med gemeint. Die in Buch eingesetzten medizinischen Informationssysteme und deren Einbindung in das KIS sind in der Grafik 3.1 abgebildet.

Ist keine (direkte) Kommunikation zum führenden System möglich, wird eine der folgenden Möglichkeiten realisiert:

- ein Kommunikationsserver in seiner Zwischenspeicherfunktion dient als Übersetzer der Nachrichten zwischen den Systemen
- Ablösung des alten Systems und bis dahin
 - Weiternutzung und kompensieren des Medienbruchs durch den Mensch

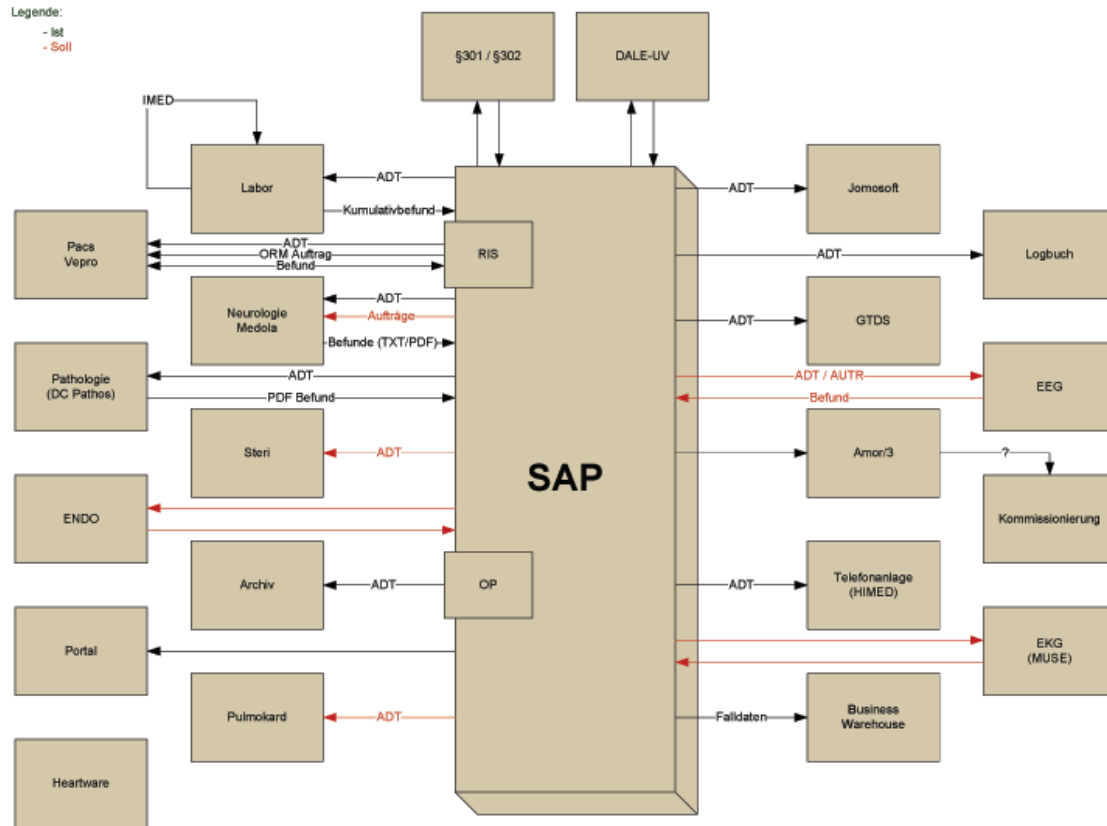


Abbildung 3.1: Übersicht über die eingesetzten Informationssysteme und deren Kommunikation

– proprietäre oder eigene Schnittstellenentwicklung

Helios hat mit der Übernahme des Krankenhausbetriebs der Charité im Jahre 2001 unter anderem auch einige bis dahin eingesetzte medizinische Informationssysteme erhalten. Auf diese Systeme wurden mit der Frage nach Integration die oben genannten Möglichkeiten angewendet, da SAP und IS-H vor der Übernahme durch Helios nur für die Abrechnung in Buch genutzt wurden. Bis heute ist das einzig übriggebliebene Charité-Informationssystem das von der Kardiologie eingesetzte Heartware. Es handelt sich um ein monolithisches System, das keine rechnergestützte Kommunikation mit SAP besitzt. Der Medienbruch wird durch den Anwender kompensiert. Die Komplexität und Funktionsvielfalt gestatteten bisher keine Ablösung. Eine Schnittstellenentwicklung scheiterte bisher unter anderem an den unterschied-

lichen vorhandenen Schnittstellen (siehe dazu auch Kapitel 3.2.1.1). Alle weiteren Systeme wurden entweder in ihrer Funktion in SAP abgebildet oder es wurden neue Informationssysteme eingekauft, welche die Kommunikation mit SAP von Anfang an realisiert haben.

Die einzelnen Komponenten, die als Gesamtheit das KIS im Klinikum Buch darstellen, werden in den nächsten Unterkapiteln vorgestellt.

3.1 SAP, IS-H und IS-HMed* im HELIOS Klinikum Berlin-Buch**

SAP, IS-H und IS-H**Med* werden als führende Informationssysteme des KIS im Klinikum eingesetzt. Das bedeutet, dass im Falle der Datenredundanz in diesen Systemen die aktuellen Versionen der Daten gespeichert sind. SAP selbst wird für die administrativen Aufgaben im Klinikum genutzt. Außerdem wird das SAP-GUI (Graphical User Interface) für eine einheitliche Präsentation verwendet.

SAP for Healthcare (IS-H) und IS-H**Med* lassen sich in Patientendatenverwaltungssystem und medizinisches Informationssystem anhand ihrer Aufgabenbereiche unterscheiden. Aber Zusatzmodule, „Add ons“ genannt, wie der OPS-Collector¹ von IS-H**Med* ermöglichen auch eine verbesserte Abrechnung. Die Abrechnung stellt allerdings eine Funktion des Patientendatenverwaltungssystems dar, so dass eine genaue Trennung nicht gemacht werden kann und auch nicht notwendig ist, da IS-H**Med* ein vollständig integriertes klinisches Arbeitsplatzsystem für SAP for Healthcare [GSD08] darstellt. Im Klinikum werden nicht alle angebotenen Module von IS-H**Med* genutzt. Mögliche Gründe sind, dass Produkte anderer Hersteller geeigneter wirken, oder die Module erscheinen einfach unnötig. Außerdem existierten, wie bei der Kardiologie, bereits medizinische Informationssysteme im Klinikum, so dass bisher zum Beispiel auf das IS-H**Med*-Modul Kardiologie nicht zurückgegriffen wurde.

¹Der OPS-Collector wurde eingeführt, um auch einfache Ereignisse in der Intensivmedizin, wie die Implantation einer Medikamentenpumpe oder Medikamentengaben, zu dokumentieren. Aus diesen Einzelleistungen kann sich summiert wieder ein OPS-Code ableiten, der zusätzlich abgerechnet wird (vgl. [GSD08]).

Das Modul „Basis“ stellt grundlegende Funktionen Arztbriefschreibung, Leistungserfassung, Auswertungen (Reports) und den klinischen Arbeitsplatz bereit. Der klinische Arbeitsplatz bietet angepasste Sichten, Plantafeln für Tagesarbeitslisten und weitere Möglichkeiten wie der „Klinische Auftrag“ für die Steuerung und Überwachung der Patientenbehandlung. Klinikweit eingesetzt sind so die gespeicherten Daten der über 400'000 in SAP erfassten Patienten und deren 1,2 Millionen Fälle von jedem klinischen Arbeitsplatz zu jeder Zeit aus einsehbar. „Klinische Aufträge“ sind Abbildungen realer Aufträge an Funktionsstellen im SAP-System, die in den jeweiligen Planungslisten wiederzufinden sind. Damit auch ein komplexer Auftrag, wie eine Operation, im System genau dokumentiert werden kann, wird das Modul „OP-System“ genutzt, das neben einer Raumplanung für die OP-Säle auch mehrstufige OP-Planung und einen OP-Monitor bereitstellt, der das aktuelle Geschehen übersichtlich zusammenfasst.

Bei dieser Menge an sensiblen Daten wird die Forderung nach einem komplexen Berechtigungsmanagement von SAP durch Rollen bedient. Damit erhält jeder Nutzer nur die Informationen, die er für seine Arbeit benötigt.

Weitere eingesetzte Module sind Medikation, Ambulanz und Radiologie. Das Radiologie-Modul ist als RIS (Radiologie-Informationssystem) in der Übersichtsabbildung 3.1 genannt. Die Radiologie erhält eine spezielle Auftrags- und Terminplanung, die mit externen Bildarchivsystemen wie PACS abgeglichen wird. An das PACS² werden Auftrags- und ADT-Daten gesandt, RIS erhält einen Dokumentenverweis auf die abgelegten Bilder zurück.

Da IS-H*Med nur einen Teil der medizinischen Informationssysteme abdeckt, übernehmen Fremdsysteme die weiteren Funktionen. Diese werden im nächsten Kapitel vorgestellt.

3.2 Weitere Medizinische Informationssysteme

Im Klinikum Buch wird bei medizinischen Informationssystemen nur zwischen Fachabteilungs- und Leistungsstelleninformationssystemen unterschieden. Ambulanz- und

²PACS steht für Picture Archiving and Communication System, zu Deutsch Bildarchivierungs- und Kommunikationssystem. Bildgebende Geräte liefern digitale Bilder, alte Aufnahmen werden mit Hilfe von Scannern digitalisiert und stehen dann ebenfalls im PACS zur Verfügung.

Pflegeinformationssystem sowie das Leistungskommunikationssystem sind bereits in den SAP-Komponenten realisiert. Ein Kommunikationsserver ist nicht vorhanden, Planungen für die Einführung sind allerdings im Gange. Die Schnittstellenwartung und -einrichtung im SAP-System selbst wird als zu aufwendig angesehen. Der Kommunikationsserver soll daher die Kommunikation zwischen SAP und externen Systemen übernehmen.

Diagnostische Geräte sind heute genauso Leistungsstelleninformationssysteme wie die schon klassischen Herzkatheter oder MRT. Ein EKG oder ein Ergometer können Auftrags- und Patientendaten vom führenden System erhalten und speisen ihre Ergebnisse wiederum in das KIS ein. Andere Leistungsstellen sind bereits in SAP integriert, wie die Radiologie und das OP-System.

Die verwendeten Fachabteilungsinformationssysteme, insbesondere das kardiologische Patientendokumentationssystem Heartware, sind nun Gegenstand der Betrachtung.

3.2.1 Heartware: Patientendokumentation in der Kardiologie

Wie schon genannt, ist Heartware das einzige von der Charité übernommene Informationssystem, das sich derzeit noch im Einsatz befindet. Es wurde von der IT-Abteilung der Franz-Volhard-Klinik für Kardiologie, namentlich Rainer Stein und Dankward Hähnlein, ab 1993 entwickelt. Heartware basiert auf der Datenbanksoftware FileMaker. Nach der deutschen Einheit wurde die Klinik ab 1992 mit Macintosh-Computern der Firmen Apple und Umax ausgestattet, die als Präsentationsplätze für das überall in der Klinik einzusetzende Fachabteilungsinformationssystem Heartware dienen sollten. Diese eigenständige Entwicklung entstand unter der eingeschränkten Sicht „nur“ für die kardiologische Klinik eine elektronisch verteilte Patientendokumentation anbieten zu können. Die Möglichkeit, die sich dadurch allerdings bot, war ein genau auf die Klinik, die Ärzte und den Arbeitsablauf abgestimmtes Informationssystem. Durch die Nähe zu den Entwicklern konnten schnell Verbesserungen oder Veränderungen in einem bestimmten Rahmen eingebunden werden. Die durch Befunddokumentation unterstützten Leistungsstellen waren und sind zum großen Teil noch heute:

- Angiographie

- Sonographie
- Ergometrie
- Echokardiographie
- Kardio-MRT
- Herzkatheter (HK-Dokumentation, Koronarbefund, PTCA-Bericht)
- EKG und Langzeit-EKG
- Myokard-Thallium-Szintigraphie
- Röntgen
- Poliklinik (Sprechstundenbögen)

Dazu kommen Funktionen wie Patienteneinbestellung, Bettenbelegungskalender, Terminplanung, ADT, Arztbriefschreibung, Labordatenabruf oder Stationslisten für den direkten Zugriff auf die Daten der auf den Stationen liegenden Patienten. In der Abbildung 3.2 auf Seite 41 ist das Hauptmenü von Heartware zu sehen.

Von Anfang an wurde das System immer auch als Grundlage für Forschungsstudien genutzt. Der Datenbestand beziffert sich heute auf rund 150 000 Fälle³ von 87 000 verschiedenen Patienten. FileMaker erlaubt das Suchen im Eingabelayout. Einfache Fragestellungen können so mit dem Arzt zusammen meist ad hoc beantwortet werden. Abfragen, die über einzelne Befundgrenzen hinweg gehen, sind nur durch Erweiterung des Suchlayouts um die abzufragenden Felder möglich. Für wiederkehrende Studien sind die Abfrageparameter in der Datenbank als Recherchen gesichert und können von den Ärzten unabhängig von der IT gesichtet werden. 2004 wurde beschlossen, dass der Betrieb von Heartware weitergeführt wird, da mehrere Gründe dafür sprachen:

- Großer Funktionsumfang (für die Kardiologie)
- Angepasste formularähnliche Befunddokumentation
- Geringe Latenzen in der Datenkommunikation zwischen Client und Server
- Akzeptiertes System
- Gute Recherchemöglichkeit

³Ein Fall eines Patienten umfasst bei stationären Patienten alle Befunde, die während seines Aufenthaltes angelegt wurden, bei einem ambulanten oder nur für den Besuch von Funktionsstellen angereisten Patienten werden die Befunde in einer Fallsammlung zusammengefasst.

- Adaption der Funktionalitäten in SAP zu kostenaufwendig

Laut Aussage der kardiologischen Ärzte gelingt, besonders in den ambulanten Episoden⁴, durch die angepasste Befunddokumentation eine schnellere Befunderfassung. Der Arzt hat somit mehr Zeit für das Gespräch mit dem Patienten, oder er kann die Patientenfrequenz erhöhen und so die Wartezeit verringern.

Diese Gründe wogen schwerer als die mit den Jahren und der Entwicklung in der medizinischen Informatik immer stärker in den Vordergrund tretenden Nachteile:

- keine Patientenidentifikationsnummer, Zuordnung über Name, Vorname und Geburtsdatum
- keine Schnittstellenkommunikation zu externen Systemen, dadurch Doppelerfassung
- keine Abrechnungsfunktionalitäten
- kein Berechtigungsmanagement
- abgeschlossene Episoden nicht vor späterer Änderung geschützt
- keine Versionierung
- keine Codierung von Diagnosen oder Prozeduren
- kein Einblick von Rechnern ohne Heartware-Präsentation möglich (andere Abteilungen)
- A4-Portrait-Orientierung nicht mehr zeitgemäß

Die für die weitere Nutzung im Helios-Netz nötige Portierung Heartwares zur x86-Plattform unter Microsoft Windows wurde von der Firma „System&Project“ vorgenommen. Die Macintosh-Systeme wurden sukzessiv durch Fat-PC-Clients mit dem Betriebssystem Windows 2000 und Windows XP sowie Thin-Clients unter Citrix mit dem Betriebssystem Windows Server 2003 abgelöst.

Um einige der Nachteile der alten Heartware-Version auszuräumen, wurde die Firma „System&Project“ mit Heartware Version 2 beauftragt. In Kapitel 4 werden diese benannt. Die Abbildung 3.3 zeigt das Hauptmenü dieser Version.

Zurzeit findet die Evaluierung der neuen Fassung (siehe Abbildung 3.3) durch das medizinische Personal der Kardiologie, durch die Krankenhausverwaltung und

⁴In der Poliklinik der kardiologischen Klinik werden Sprechstunden von Kardiologie-Ärzten angeboten, die von externen Patienten besucht werden.

durch die IT statt. Weiterhin soll hier die in der Praxis eingesetzte Heartware-Version betrachtet werden.

3.2.1.1 Funktionalitäten

Heartware ist ein aus 50 einzelnen Datenbanken (im Sinne von FileMaker) aufgebautes relationales Datenbanksystem, unter dem Dach des Datenbankprogramms FileMaker 5. Die zweischichtige Client/Server-Architektur besteht aus dem FileMaker Server, der die einzelnen Datenbanken vorhält, sowie den FileMaker Pro Clients, die der Präsentation dienen. Ein proprietäres Protokoll übernimmt die Kommunikation zwischen Client und Server. Auf Netzwerkebene kommt entweder TCP/IP oder Appletalk für die Apple-Version von FileMaker zum Einsatz.

Jede einzelne Heartware-Datenbank bietet verschiedene Sichten, Layouts genannt, um die Relationen darzustellen. Jede FileMaker-Datei enthält genau eine Datenbank mit genau einer Tabelle von Relationen. Die Datenbankstruktur ändern, als auch Datenfelder definieren oder löschen, konnte seit Beginn der Heartware-Entwicklung während der Laufzeit durchgeführt werden.

Die Abfragefunktionalität in Heartware wird über die GUI angeboten. Die Suchkriterien werden in Felder der Suchabfrage eingegeben. Die Suchabfrage sieht wie ein leerer Datensatz aus, da dasselbe Layout dafür verwendet wird und folgt damit dem „Query by Example“-Prinzip (QBE) [Zlo77]. Ausschließende Kriterien sind ebenfalls möglich. Die Ergebnismenge kann zur Weiterverarbeitung exportiert werden. Die unterstützten Formate, auch für den Import, sind laut [Fil04]:

- Microsoft Excel
- DBF (nur dBase III und IV)
- DIF
- ClarisWorks
- Text mit Tab
- Text mit Semikolon
- BASIC
- FileMaker Pro
- Serienbrief

- SYLK
- WK1/WKS (Lotus 1-2-3)
- HTML-Tabelle (nur Export) und
- XML (nur FileMaker eigene Dokumenttypdefinition (DTD) unterstützt)

Wenn Daten in eine bestehende Datei importiert werden sollen, können

- der Datei neue Datensätze hinzufügt,
- vorhandene Datensätze in der Datei aktualisiert oder
- passende Datensätze in der Datei aktualisiert werden.

Bei Punkt 2 werden Datensätze aus der Quelldatei in der Zieldatei durch die gleiche Anzahl an Datensätzen ersetzt, ohne Vergleich der Feldinhalte. Punkt 3 aktualisiert nur jene Datensätze, bei denen vorher festgelegte Felder identisch sind. Die Abbildung 3.4 soll dies verdeutlichen.

Sollen bestehende Patientenstammdaten durch Importierung der Daten aus dem führenden System aktualisiert werden, ist die dritte Methode die richtige Wahl.

Heartware wird über Tasten bedient. Den beschrifteten Tasten sind Skripte hinterlegt, die Aufgaben automatisiert ausführen. Was skriptgesteuerte Ausführung in Heartware bedeutet, ist Thema des folgenden Kapitels.

3.2.1.2 Skriptgesteuerte Ausführung

Ein Skript ist eine Zusammenfassung von mehreren Anweisungen mit automatischem Ablauf. Skripte werden innerhalb von FileMaker Pro erstellt. Um eine Aufgabe zu automatisieren, wird durch Aneinanderreihen von in einer Liste vordefinierten Skriptschritten und Angeben der zugehörigen Optionen ein Skript erstellt [Fil04]. Das Skript wird in der formularbasierten Ansicht mit einer geeignet beschrifteten Taste verknüpft.

Ein einfaches Beispiel ist das Ausdrucken eines Datensatzes in einem speziell zum Drucken angelegten Layout, wie es in Abbildung 3.5 zu sehen ist.

Neben diesem einfachen Ablauf können Verzweigungen, Schleifen und Teilskripte eingebunden werden, um auch komplexere Aufgaben „zu skripten“. Als komplexe Aufgabe ist auch das automatisierte Importieren von Daten in die Datenbank zu sehen, welches im nächsten Abschnitt vorgestellt wird.

3.2.1.3 Import von XML-strukturierten Daten

Es ist möglich, durch ein Skript einen Import zu steuern. Die Importdaten können XML-strukturiert vorliegen. Leider wird nur eine einzige⁵, von FileMaker Inc. erstellte, Dokumenttypdefinition (DTD) *FMPXMLResult* unterstützt. Sollen Daten im XML-Format in Heartware importiert werden, muss die Datei, wie von *FMPXMLResult* gefordert, aufgebaut sein:

```

1 <!DOCTYPE FMPXMLRESULT [
2   <!ELEMENT FMPXMLRESULT (ERRORCODE, PRODUCT, DATABASE, METADATA,
3     RESULTSET)>
4     <!ATTLIST FMPXMLRESULT
5       xmlns CDATA #REQUIRED>
6     <!ELEMENT ERRORCODE (#PCDATA)>
7     <!ELEMENT PRODUCT EMPTY>
8       <!ATTLIST PRODUCT
9         NAME CDATA #REQUIRED
10        VERSION CDATA #REQUIRED
11        BUILD CDATA #REQUIRED>
12     <!ELEMENT DATABASE EMPTY>
13       <!ATTLIST DATABASE
14         NAME CDATA #REQUIRED
15         RECORDS CDATA #REQUIRED
16         DATEFORMAT CDATA #REQUIRED
17         TIMEFORMAT CDATA #REQUIRED
18         LAYOUT CDATA #REQUIRED>
19     <!ELEMENT METADATA (FIELD)*>
20       <!ELEMENT FIELD EMPTY>
21       <!ATTLIST FIELD
22         NAME CDATA #REQUIRED
23         TYPE (TEXT | NUMBER | DATE | TIME | CONTAINER) #
24           REQUIRED
25         EMPTYOK (YES | NO) #REQUIRED
26         MAXREPEAT CDATA #REQUIRED>
27     <!ELEMENT RESULTSET (ROW)*>
28       <!ATTLIST RESULTSET
29         FOUND CDATA #REQUIRED>

```

⁵Tatsächlich existiert mit *FMPDSOResult* noch eine zweite DTD, diese besitzt allerdings keine für die Semantik wichtigen Metadaten und wird daher nicht in Erwägung gezogen.

```

28     <!ELEMENT ROW (COL) *>
29         <!ATTLIST ROW
30             RECORDID CDATA #REQUIRED
31             MODID CDATA #REQUIRED>
32     <!ELEMENT COL (DATA) *>
33         <!ELEMENT DATA (#PCDATA)>
34 ]>

```

Das Metadata-Element erlaubt es, die für die Semantik wichtigen Feldbezeichnungen in die XML-Datei aufzunehmen. Dies ist eine Voraussetzung, um nur passende Datensätze abzugleichen, da so nicht nur die Syntax, d. h. die Reihenfolge der Daten, zur Kontrolle genutzt wird.

Es ist möglich, eine anders geformte XML-Datei durch ein zugehöriges Extensible Stylesheet Language for Transformations (XSLT) in die korrekte Struktur zu überführen.

Soll die Importierung automatisch ablaufen, so muss sichergestellt sein, dass tatsächlich der richtige Datensatz aktualisiert wird. Dafür werden in Quell- und Zieldatenbank eindeutig zuzuordnende Schlüsselfelder benötigt. Sind diese nicht vorhanden, kann eine automatische Aktualisierung nur verantwortet werden, wenn das Skript einer menschlichen Kontrollinstanz die zu aktualisierenden Daten präsentiert und auf die Auswahl des korrekten Datensatzes wartet. Dieses Problem wird weiter in Kapitel 4 anhand der Beispielimplementation thematisiert.

3.2.2 Überblick über weitere medizinische Informationssysteme

Sind diagnostische Geräte heute zu Leistungsstelleninformationssystemen gereift, so kann der angebotene Funktionsumfang dieser Leistungsstelleninformationssysteme sogar der Definition eines Fachabteilungsinformationssystems entsprechen. Damit ist die Unterteilung der medizinischen Informationssysteme zum Teil aufgehoben. Einige Beispiele aus Buch sollen dies verdeutlichen.

Seit kurzem wird für das Ruhe-Elektrokardiogramm (EKG), das Langzeit-EKG und das Belastungs-EKG das Leistungsstelleninformationssystem MUSE von GE Healthcare verwendet. Muse kommuniziert mit SAP mittels HL7. ADT und Klinischer Auftrag werden in Muse aufgenommen. Damit besitzt der Untersucher mit die-

sem System alle aktuellen demographischen Daten zu einem Patienten. Aufzeichnungen aus früheren Fällen können mit der aktuellen verglichen werden. Abschließend wird ein möglicher Diagnosebericht und Kurvenformergebnisse an SAP gesendet.

Unter Anderem wurde für das Herzkatheterlabor Pulmokard der Firma Siemens Medical Solutions angeschafft. Geplant ist hier ebenfalls, dass Patientendaten durch HL7 in das System gelangen. Momentan liegt die Schnittstelle noch brach und der Medienbruch wird durch den Anwender durch manuelle Eingabe der Daten kompensiert. Bei einer Herzkatheteruntersuchung wird eine ständige Durchleuchtung der Herzkranzgefäße mit Röntgenstrahlen benötigt. Die erfassten Bilder ergeben durch die hohe Aufnahmefrequenz von ca. 15fps⁶ Bewegtbilder. Die Speicherung der Bewegtbilder übernimmt Pulmokard. Der Untersucher kann diese kurzen Filme eines Falles anschließend mit einem Untersuchungsbericht kommentieren.

Die Patientendokumentation für diese beiden Systeme wird allerdings in Heartware vollzogen. Dies hat historische Gründe, da die Vorgänger der beiden vorgestellten Systeme reine Leistungsstelleninformationssysteme waren. Durch die erweiterte Befundungsfunktion und die verbesserte Integrationsmöglichkeit in ein KIS sind diese Systeme zu Fachabteilungsinformationssystemen gereift, deren Funktionalität allerdings nicht ausgereizt wird.

⁶„fps“ steht für frames per second, zu Deutsch Bilder pro Sekunde.

FileMaker Pro - [Aufnahme+Deckblatt.fp5]

Datei Bearbeiten Ansicht Einfügen Format Datensätze Scripts Fenster Hilfe

HAUPTMENÜ Franz-Volhard-Klinik
HEARTWARE 1.1e1 Berlin-Buch

ERST-Aufnahme eines Patienten NOT-Aufnahme Station N [CHIPKARTE]

Namenssuche Einbestellung Arztadressen Rotation How 2 Stationslisten

Aktueller Patient: 114950

Nachname, Vorname, Geburtsdatum:
TEST Willy, geb.: 01.01.1981

Gewählter Aufenthalt von...bis: 22.01.2004 Geburtsdatum: 01.01.1981

Für obigen Patienten :

Neues Befundblatt erstellen (führt zu Auswahlmenü)
Wiederaufnahme dieses Patienten

Zu einem existierenden Blatt gehen :

Aufnahmeblatt EKG Arztbrief Labor Röntgen HK-Doku HK-Etik.
NOTAUFNAHME Belastungs-EKG Sonographie Koronarbefund
Deckblatt Langzeit-EKG Angiologie Urinbogen
WU-Poliklinik/St.T HK-Elektrophys. Echo MRT PTCA-Bericht
Aufn.-Schein Verlaufs-Doku. Nuklearmedizin Herzteam
Entl.-Schein HK-Anmeldung HK-Programm OP-Warteliste

Beenden

Für diesen Patienten sind bei diesem Aufenthalt bisher folgende Befundbögen angelegt worden:
- angezeigt wird das Erstellungsdatum -

EPISODEN:		*2201 2004: Extern -> BEFUNDE:
28.10.1989 - 21.12.1999	T	19.11.2003: EKG
07/02/2001 - 08/02/2001	T	13.04.2004: HK-Elektrophys.
2201 2004	Extern	26.07.2004: Angio
		26.07.2004: Angio
		26.10.2004: HK-Doku.
		08.03.2005: Not-A.
		bis 2005: 1 * HK-Anmeldung
		15.09.2005: WU-Poli/T
		13.04.2006: Not-A.
		10.07.2006: HK-Anmeldung
		20.02.2007: MRT
		23.02.2007: HK-Anmeldung
		14.08.2007: MRT
		21.08.2007: Nuklearmed.
		06.05.2008: WU-Poli/T

ACHTUNG!!!
Die Befund-Anzeige (rechts ->) funktioniert für stationäre Aufenthalte nur noch für >= 1996!
Die Befunde sind aber recherchierbar.

CorAg- & Interventions-FollowUp

Recherche: Aktuelle Herzkatheter-Untersuchungen der letzten Woche

Hypertonie-Recherche NYHA-Recherche EFES-Recherche SUPERVISOR

100 Blättern

Abbildung 3.2: Das Hauptmenü von Heartware 1.1

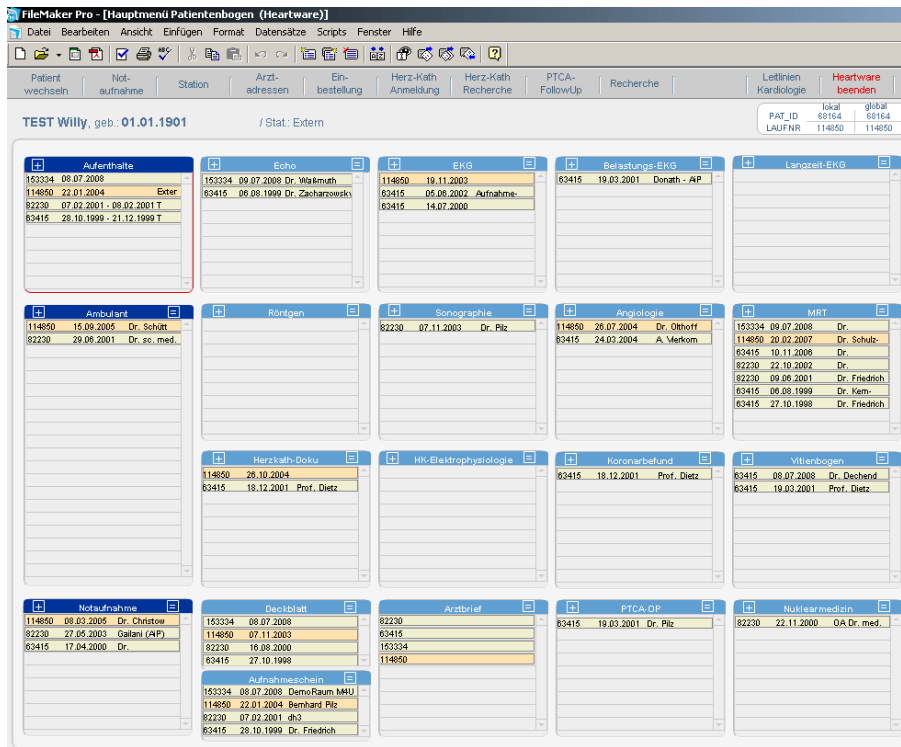


Abbildung 3.3: Das Hauptmenü von Heartware 2

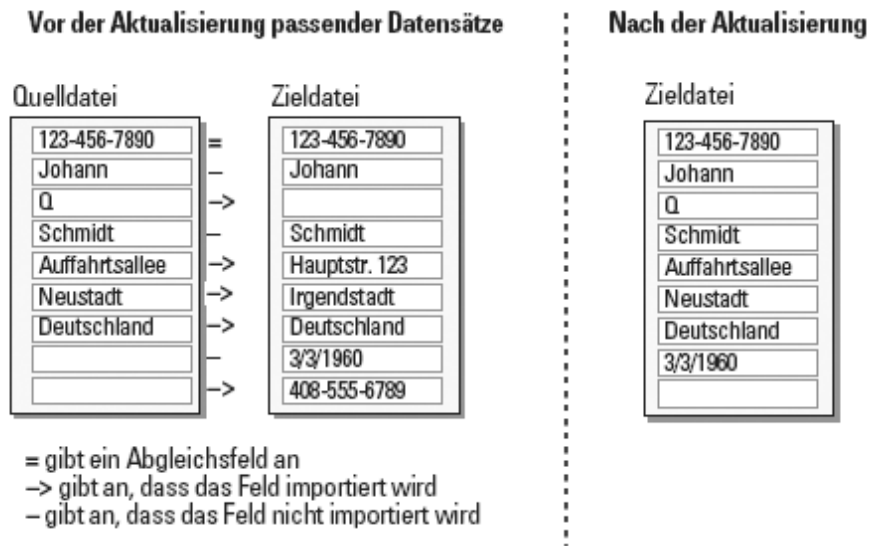


Abbildung 3.4: Import: Aktualisierung von passenden Datensätzen, aus [Fil04]

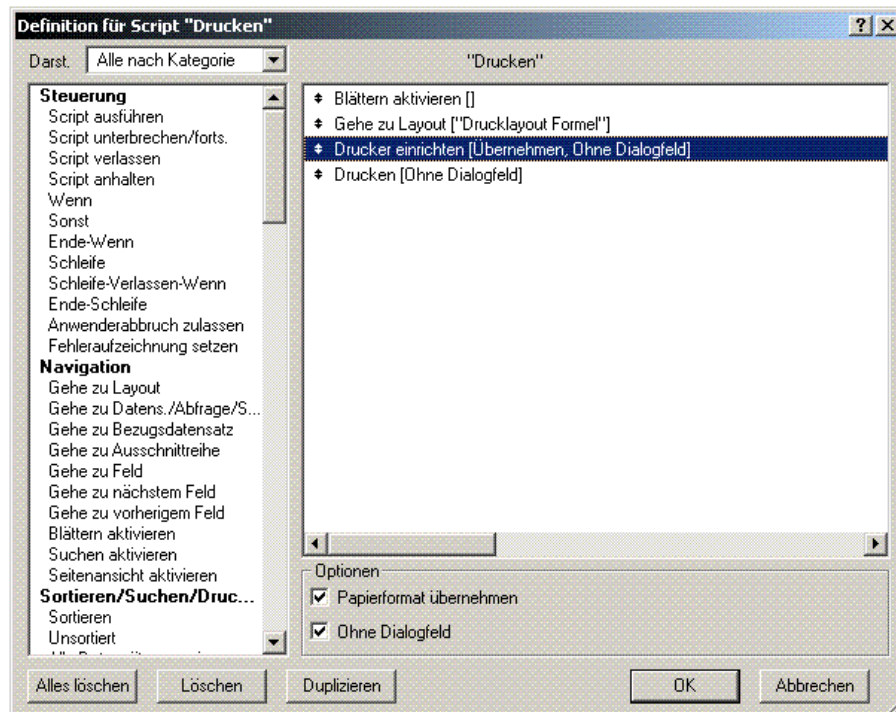


Abbildung 3.5: Skript „Drucken“. Die während der Skripterstellung gültigen Druckereinstellungen werden für die Druckereinrichtung zur Laufzeit genutzt.

Kapitel 4

Unidirektionaler Datenabgleich zwischen IS-H*Med und Heartware

In Kapitel 3.2.1 wurden die Nachteile von Heartware aus heutiger Sicht aufgezeigt. Dieser zunächst unfair erscheinende Vergleich zwischen aktuellen Anforderungen an ein medizinisches Informationssystem und einem 15 Jahre alten System ist realistisch, da erst 2008 mit der neuen Version einige der Nachteile angegangen wurden. Diese sind:

- Patientenidentifikationsnummer
- Berechtigungsmanagement
- lesender Zugriff auf Heartware durch ein Web-Interface und
- abgeschlossene Episoden können gegen Überschreiben gesperrt werden.

Am Portrait-Format wird festgehalten, da dies dem Format des auszudruckenden Formulars entspricht. Auch gegen die Abrechnungsfunktionalität hat man sich bewusst entschieden, da diese Aufgabe in SAP erledigt wird. Damit wird die für die Leistungserfassung notwendige Codierung der Diagnosen und Prozeduren ebenfalls nicht benötigt. Ein medizinisches Informationssystem sollte heute nur in einem KIS bestehen, wenn es darin integriert ist. Das bedeutet, dass es eine Kommunikation zwischen Heartware und dem führenden System, SAP, geben sollte, um seine weitere Existenz zu sichern. In der Agenda für die neue Heartware-Version ist die Kommunikation mit SAP ebenfalls aufgeführt, in dem Sinne, dass Patientenstammdaten von

SAP nach Heartware übernommen werden sollen. Umgesetzt werden konnte dieses Vorhaben allerdings bisher nicht. Gründe dafür sind:

- Heterogenität in der Kommunikation SAP - Heartware
- Zuordnung der Patienten ist nur über Name, Vorname, Geburtsdatum möglich
- SAP-Patientenidentifikationsnummer (PatID) ist in Heartware nicht bekannt
- Heartware 2-PatID ist in SAP nicht bekannt

Die SAP-PatID ist eindeutig einem Patienten zugeordnet, genauso die PatID der neuen Heartware-Version. Es existiert aber keine Zuordnung zwischen diesen beiden Identifikationsnummern, zwischen den beiden Systemen.

Im folgenden stellt sich somit die Aufgabe, wie in Heartware einem Patienten die passende SAP-PatID zugeordnet werden kann. Die doppelt zu pflegenden Patientenstammdaten können in diesem Zuge ebenfalls in Heartware aktualisiert werden.

4.1 Vorgehen

Die Patienten müssen in beiden Systemen möglichst eindeutig identifiziert werden. Davon ausgehend, dass von jedem Patienten wenigstens der Name, der Vorname und das Geburtsdatum korrekt in beiden Systemen vorhanden sind, kann dies als schwache eindeutige Zuordnung genutzt werden. Schwach deshalb, da nicht ausgeschlossen werden kann, dass in einem Datenbestand von mehreren Hunderttausend Patienten die Suchkriterien auf mehrere Patienten zutreffen. Ist die Abfrage tatsächlich mit mehreren Treffern von SAP beantwortet worden, muss aus der Trefferliste der richtige Patient ausgewählt werden. Eine automatisierte Lösung, die in Heartware jedem Patienten die zugehörige SAP-PatID vergibt, ist somit nicht realisierbar. Aber es ist abzusehen, dass die Menge an Patienten mit gleichen Namen, Vornamen und Geburtsdatum nicht groß ist und mit einem vertretbaren Aufwand bei diesen Fällen die SAP-PatID manuell eingetragen werden kann. Dieser Vorgang ist einmalig für die Initialisierung des SAP-PatID-Feldes in Heartware als auch wiederholt möglich. Dem Arzt sollte außerdem die Möglichkeit gegeben werden, auf Anforderung aktuelle Patientenstammdaten seines Patienten aus SAP zu importieren. Diese Aufgabe

soll in dieser Arbeit angegangen werden zeigen, ob und wie die Kommunikation zwischen den heterogenen Systemen aussehen kann. Aufgrund der Verschiedenheit in den Kommunikationsmöglichkeiten ist ein modulares Vorgehen notwendig, welche in der Abbildung 4.1 beschrieben wird.



Abbildung 4.1: Modulare Aufgabenverteilung

Nun lässt sich die gerichtete Schnittstelle wie in Abbildung 4.2 dargestellt definieren. Das Verhalten bei möglichen Fehlern wird in Kapitel 4.3 untersucht, im Folgenden wird der Weg der funktionierenden Kommunikation vorgestellt. Der erste Schritt ist die vom Heartware-Anwender zu startende Anforderung, die Patientenstammdaten zu aktualisieren. Der Name, der Vorname und das Geburtsdatum aus dem aktuell aufgerufenen Patientendatensatz sind die Suchkriterien für die BAPI-Methode Patient.Search im nächsten Schritt. Die Kommunikation geschieht datei-los über RFC.

4.2 Implementierung und Einbindung in den Betrieb

Die Aufgabenteilung sieht drei Gebiete für die Implementierung vor: SAP, SAP Connector und Heartware. Zunächst wird in SAP ein Zugang eingerichtet. Das BO Patient bietet auch BAPI-Methoden an, die Änderungen in den Patientenstammdaten von SAP ermöglichen. Dies ist allerdings hier nicht vorgesehen. Daher sind bei der Berechtigungsvergabe sorgfältig alle Schreibrechte zu entfernen. Siehe dazu die Abbildung 2.6 in Kapitel 2.4.2. Die für die Durchführung der Methode benötigte Berechtigung lautet N_PATI_SRC.

Zunächst wird das Projekt für eine Testumgebung entwickelt. Damit wird sichergestellt, dass der Produktivbetrieb während der Entwicklung und dem Testen zu keiner Zeit gefährdet wird. R/3 besitzt für diesen Zweck ein Testsystem, welches autark vom Entwicklungs- sowie Produktivsystem läuft. Es enthält in regelmäßigen Abständen aktualisierte gültige Patientendaten. Auch der Funktionsumfang aller Komponenten ist identisch zum Produktivsystem.

Um den Produktivbetrieb von Heartware nicht zu gefährden, wurde für diesen Zweck ein neuer Filemaker Server aufgesetzt. Eine Kopie der Heartware-Datenbank wurde aufgespielt, die nicht nur die Datenbasis liefert, sondern durch die Nutzung derselben Layouts auch das Testen realitätsnäher gestaltet. Bei Erfolg des Projektes können die Anpassungen an das sich im Einsatz befindende Heartware so einfacher vollzogen werden.

Das im folgenden Kapitel im Mittelpunkt stehende Programm SAP Connector ermöglicht die indirekte Kommunikation zwischen SAP und Heartware.

4.2.1 SAP Connector

Die verwendete Programmiersprache ist Java in der Version 1.5, als Entwicklungsumgebung wird Eclipse Europa Winter (Version 3.3.2) eingesetzt. Die Gründe dafür sind die für das Projekt verfügbaren Hilfsmittel: Für die Kommunikation mit SAP bietet der SAP Java Connector (JCo) in Version 2 bereits Datentypen und Funktionen, die verwendet werden können. JCo wurde in Kapitel 2.4.2.2 vorgestellt. Ein weiteres Hilfsmittel ist ein Eclipse-Plug-In: „sap2java“ erstellt lokal Proxyklassen

der gewünschten BAPI-Methode und erleichtert damit die Konstruktion des Abfragevektors. Das Sourceforge-Projekt ist zu finden unter <http://eclipsesap2java.sourceforge.net>. Das Plug-In wird als Assistent (Wizard) gestartet und erwartet folgende Eingaben:

Eingabefelder	Erläuterung
Container: /SAP Connector	Name des Eclipse-Projekts
R/3 Host Name: blnsap09.blnsapdom.de	Name oder Adresse des Servers
Client: 400	Mandant/System ID
System no: 01	Systemnummer
Language: DE	Sprache
Username: kardbapi	Benutzername
Password: ***	Passwort
BAPI Name: BAPI_PATIENT_SEARCH	BAPI-Methode
Package Name: com.sap.bapipatient	Package-Name

Nach Durchlaufen des Assistenten wurde ein Package `com.sap.bapipatient` generiert, welches die Klassen `Bapi1084srch`, `Bapiret2`, `InputBapi_patient_search` und `OutputBapi_patient_search` enthält.

Die Klasse `Bapi1084srch` wird aus den möglichen Export- und Importparametern der BAPI-Methode gebildet. Das Mapping auf die entsprechenden Java-Datentypen (siehe Kapitel 2.4.2.2) wurde dafür bereits vorgenommen. Des Weiteren sind Getter- und Setter-Methoden der einzelnen Attribute generiert worden.

In der Klasse `Bapiret2` werden die Attribute für die Tabelle von aufgetretenen Fehlern vorgehalten. Mögliche Fehlermeldungen sind in Tabelle 4.1 aufgelistet.

Die Fehlermeldungen wurden nur während der Entwicklungsphase ausgewertet, im Produktivbetrieb werden sie nicht an den Anwender weitergeleitet. `InputBapi_patient_search` enthält Attribute sowie Getter- und Setter-Methoden der Abfrageparameter. Die zur Verfügung stehenden Filterattribute sind in Tabelle 4.2 beschrieben.

Für die Implementierung wird `FILTER_LAST_NAME_PAT`, `FILTER_FRST_NAME_PAT` und `FILTER_DOB_FROM` verwendet. Außerdem muss die Variable `INSTITUTION` mit dem Wert 124 belegt sein. `OutputBapi_patient_search`

Fehlercode	Beschreibung
N1034	Fall <PATCASEID> <CHECKDIGIT> nicht vorhanden
N1036	Bitte gültige Einrichtung eingeben
N1280	Geschlecht <SEX> ist nicht vorgesehen (Bitte Eingabe prüfen)
N1541	Fehler bei der phonetischen Suche
N1600	Berechtigungsfehler
N1830	Keine Patienten mit angegebenen Merkmalen gefunden
N1831	Bitte geben Sie eine genauere Selektion an
N1832	„Geburtsdatum bis“ wurde durch Tagesdatum ersetzt
N1833	„Geburtsdatum von“ ist größer Tagesdatum (Bitte Eingabe prüfen)
N1836	Es wurden mehr als <MAXCNT> Sätze selektiert, bitte einschränken

Tabelle 4.1: Fehlercodes von BO Patient

wird nicht eingesetzt. Es hält mögliche Exportparameter vor, die nach SAP exportiert werden können.

Um den SAP Java Connector zu nutzen, wird die Bibliothek `sapjco.jar` im Betriebssystem durch setzen der Systemvariablen `CLASSPATH = {sapjco-install-path}\sapjco.jar` und `PATH = {sapjco-install-path}` bekannt gemacht. Außerdem befindet sich in dem Verzeichnis `sapjco-install-path` die JNI-Laufzeitbibliothek `sapjcorfc.dll` sowie die native RFC-Bibliothek `librfc32.dll`. Nativ bedeutet, dass diese beiden Bibliotheken plattformabhängig nur auf Windows-Systemen einsetzbar sind. Das Marshalling zwischen den Java-Klassen und den nativen RFC-Datentypen ist die Hauptaufgabe von JCo's JNI-Laufzeitbibliothek. In der beiliegenden Dokumentation wird die Installation auf den weiteren unterstützten Plattformen beschrieben.

Im nächsten Schritt ist das Archiv `sapjco.jar` mit den JCo-Java-Klassen in das Projekt einzubinden. Dabei ist zu beachten, dass dieses Archiv während der Laufzeit benötigt wird und somit zentral auf dem Heartware-Server abzulegen ist. Außerdem wird auf diesem Server ein Verzeichnis freigegeben, welches das Anlegen einer temporären xml-Datei und das Schreiben in diese Datei erlaubt.

Im Manifest des Projektes wird festgelegt, dass ein Programmaufruf an die main-

Filterattribut	Beschreibung
FILTER_BIRTH_NAME	Filter Geburtsname Patient
FILTER_DOB_FROM	Filter Geburtsdatum von
FILTER_DOB_TO	Filter Geburtsdatum bis
FILTER_DOC_NO	Filter Ausweisdokumentnummer
FILTER_EXT_PAT_ID	Filter Externe Patientenidentifikation
FILTER_FRST_NAME_PAT	Filter Vorname Patient
FILTER_LAST_NAME_PAT	Filter Nachname Patient
FILTER_PATCASEID	Filter Fallnummer
FILTER_PATIENTID	Filter Patientennummer
FILTER_SEX_EXT	Filter Geschlecht (externer Schlüssel)
FILTER_SSN	Filter Sozialversicherungsnummer
INSTITUTION	Einrichtung
MAXCNT	Maximale Anzahl Patienten lesen aus Datenbank
SEARCH_PHONETIC	Phonetische Suche

Tabelle 4.2: Filterattribute von BO Patient

Methode der Klasse Patient gelehrt wird. Main-Methode der Klasse Patient.java und globale Variablendefinition:

```
1  public class Patient extends Object{
2      JCO.Repository repository;
3      JCO.Client klient;
4      JCO.Table tabelle;
5      static String NachName;
6      static String VorName;
7      static Date GebDatum;
8
9      Document dom;
10 (... )
11     /**
12      * Hier starten wir. Erwartet Aufruf mit Parameterübergabe in
13      * der Form
14      * <pre>sapconnector.jar Nachname Vorname Geburtsdatum</pre>
15      * Geburtsdatum in der Form "YYYYMMDD"
16      */
17     public static void main (String [] args) {
18
19         if (args.length>0) {
20             NachName = args[0];
21             if (args.length>1) {
22                 VorName = args[1];
23                 if (args.length>2) {
24                     GebDatum = toDate(args[2]);
25                 }
26             }
27         }
28
29         //neue Patientenabfrage instanziiieren
30         Patient app = new Patient();
31
32         //XML-Datei schreiben
33         app.createDocument();
34         app.createDOMTree();
35         app.printToFile();
36     }
```

Wie in Kapitel 4.2.2 beschrieben wird, kann von einem korrekten Programmaufruf ausgegangen werden, wenn dieser durch das Skript in Heartware gestartet wurde. Daher ist keine Überprüfung der übergebenen Argumente vorgesehen. Der String Geburtsdatum wird in der Methode toDate(String Geburtsdatum) zu dem Datentyp Date konvertiert.

Die neue Objektinstanz Patient wird durch `Patient app = new Patient()` gebildet. Die Zugangsdaten zu SAP sind in dem Programm fest hinterlegt und werden in einem Repository gespeichert.

```

1  public Patient() {
2
3      klient = JCO.createClient(
4          "400",          // Mandant/System ID
5          "kardbapi",    // Benutzername
6          "***",        // Passwort
7          "DE",          // Sprache
8          "blnsap09.blnsapdom.de", //Name/Adr. d. Servers
9          "01" );        // Systemnummer
10     try {
11         //Mit SAP verbinden
12         klient.connect();
13
14         repository = new JCO.Repository("KARD", klient);

```

Die BAPI-Methode, auf die zugegriffen werden soll („BAPI_PATIENT_SEARCH“), wird angegeben. Die zu verwendenden Filterparameter werden definiert.

```

15     JCO.Function function = null;
16
17     //SAP BAPI-Methode, auf die zugegriffen werden soll ,
18     //benennen
19     function = this.createFunction("BAPI_PATIENT_SEARCH");
20
21     // Abfragefilter erstellen
22     JCO.Metadata filter_meta = new JCO.Metadata("FILTER");
23     filter_meta.addInfo("FILTER_LAST_NAME_PAT", JCO.
24         TYPE_CHAR, 30, 255, 0, 0, null, null, 0, null, null
25     );

```

```
23         filter_meta.addInfo("FILTER_FRST_NAME_PAT", JCO.  
                TYPE_CHAR, 30, 255, 30, 0, null, null, 0, null,  
                null);  
24         filter_meta.addInfo("FILTER_DOB_FROM", JCO.TYPE_DATE,  
                8, 60, 0);  
25         filter_meta.addInfo("INSTITUTION", JCO.TYPE_CHAR, 4,  
                255, 78, 0, null, null, 0, null, null);
```

Diese Filterparameter werden mit Werten belegt und in einer Parameterliste gespeichert.

```
26         JCO.ParameterList filter= JCO.  
                createParameterList(filter_meta);  
27  
28         // Abfragefilter füllen  
29         filter.setValue(NachName, "FILTER_LAST_NAME_PAT");  
30         filter.setValue(VorName, "FILTER_FRST_NAME_PAT");  
31         filter.setValue(GebDatum, "FILTER_DOB_FROM");  
32         filter.setValue("0124", "INSTITUTION");
```

Die Parameterliste wird an die Abfragefunktion übergeben und die Abfrage ausgeführt. Die als Antwort zu erwartende Tabelle „PATIENTS“ wird in einer Tabelle des Datentyps JCO.table gespeichert.

```
33         //Abfragefilter an Abfragefunktion  
                übergeben  
34         function.setImportParameterList(filter);  
35  
36         //Führe die Abfrage aus!  
37         klient.execute(function);  
38  
39         //Speichere Tabelle "Patients" in Table tabelle  
40         tabelle = function.getTableParameterList().getTable("PATIENTS")  
                ;
```

Die Verbindung zu SAP wird wieder getrennt. Ist im Laufe der Abfrage ein Fehler aufgetreten, beendet sich das Programm. Eine Fehlerausschrift für den Anwender ist nicht vorgesehen.

```
41         catch (Exception ex) {  
42             ex.printStackTrace();  
43             System.exit(1);
```

```

44     }
45     //SAP BAPI-Verbindung trennen
46     klient.disconnect();
47 }

```

Für das Schreiben der xml-Datei wird eine neue Instanz des Document Object Model (DOM)-Baums erzeugt:

```

1     /**
2      * neue Instanz zum Aufbau des DOM-Baums
3      */
4     private void createDocument() {
5
6         DocumentBuilderFactory dbf = DocumentBuilderFactory.
7             newInstance();
8         try {
9             DocumentBuilder db = dbf.newDocumentBuilder();
10
11            dom = db.newDocument();
12
13            }catch(ParserConfigurationException ex) {
14                System.out.println("Error while trying to
15                instantiate DocumentBuilder" + ex);
16                System.exit(1);
17            }
18        }
19    }

```

Darin werden die benötigten Elemente eingehängt, beginnend mit dem Wurzelelement:

```

1     /**
2      * Hier wird der Baum aufgebaut, Helperklassen benötigt!
3      * <pre>
4      * <ERRORCODE></ERRORCODE>
5      * <PRODUCT BUILD="" NAME="" VERSION="" />
6      * <DATABASE DATEFORMAT="" LAYOUT="" NAME="" RECORDS=""
7      *     TIMEFORMAT="" />
8      * <METADATA>

```

```

8      *      <FIELD EMPTYOK="NO" MAXREPEAT="1" NAME="PatNachname"
          TYPE="TEXT" />
9      *      <FIELD EMPTYOK="NO" MAXREPEAT="1" NAME="PatVorname"
          TYPE="TEXT" />
10     *      <FIELD EMPTYOK="NO" MAXREPEAT="1" NAME="GeburtsDatum"
          TYPE="DATE" />
11     *      <FIELD EMPTYOK="NO" MAXREPEAT="1" NAME="PAT_ID" TYPE="
          NUMBER" />
12     *      <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="PatStrasse ,
          Hausnr."
13           TYPE="TEXT" />
14     *      <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="PatPLZ" TYPE
          ="TEXT" />
15     *      <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="PatOrt" TYPE
          ="TEXT" />
16     *      <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="PatTelefon"
          TYPE="TEXT" />
17     * </METADATA>
18     * </pre>
19     */
20     private void createDOMTree(){
21
22         /* Wurzelement erstellen
23         <FMPXMLRESULT xmlns="http://www.filemaker.com/
          fmpxmlresult">
24         */
25         Element fmpxmlEle = dom.createElement("FMPXMLRESULT");
26         fmpxmlEle.setAttribute("xmlns", "http://www.filemaker.
          com/fmpxmlresult");
27         dom.appendChild(fmpxmlEle);

```

Die weiteren Elemente sind ERRORCODE, PRODUCT und DATABASE mit den zugehörigen Attributen.

```

28         Element errorEle = dom.createElement("ERRORCODE");
29         fmpxmlEle.appendChild(errorEle);
30
31         Element productEle = dom.createElement("PRODUCT");
32         productEle.setAttribute("BUILD", "");
33         productEle.setAttribute("NAME", "");

```

```
34     productEle.setAttribute("VERSION", "");
35     fmpxmlEle.appendChild(productEle);
36
37     Element databaseEle = dom.createElement("DATABASE");
38     databaseEle.setAttribute("DATEFORMAT", "");
39     databaseEle.setAttribute("LAYOUT", "");
40     databaseEle.setAttribute("NAME", "");
41     databaseEle.setAttribute("RECORDS", "");
42     databaseEle.setAttribute("TIMEFORMAT", "");
43     fmpxmlEle.appendChild(databaseEle);
```

Das Element METADATA wird definiert. Das Unterelement FIELD wird durch die Hilfsmethode createFIELD aufgebaut. Anschließend wird das METADATA an den Wurzelknoten angehängen.

```
44     Element metadataEle = dom.createElement("METADATA");
45
46     metadataEle.appendChild(createFIELD("NO", "1", "
47         PatNachname", "TEXT"));
48     metadataEle.appendChild(createFIELD("NO", "1", "
49         PatVorname", "TEXT"));
50     metadataEle.appendChild(createFIELD("NO", "1", "
51         GeburtsDatum", "DATE"));
52     metadataEle.appendChild(createFIELD("NO", "1", "PAT_ID", "
53         NUMBER"));
54     metadataEle.appendChild(createFIELD("YES", "1", "
55         PatStrasse, Hausnr.", "TEXT"));
56     metadataEle.appendChild(createFIELD("YES", "1", "PatPLZ",
57         "TEXT"));
58     metadataEle.appendChild(createFIELD("YES", "1", "PatOrt",
59         "TEXT"));
60     metadataEle.appendChild(createFIELD("YES", "1", "
61         PatTelefon", "TEXT"));
62
63     fmpxmlEle.appendChild(metadataEle);
```

Die übergebenen Argumente von createFIELD werden zum Anlegen des Elements FIELD und dessen Attribute benutzt.

```
1     /**
2     * Hilfsmethode zum Aufbau des Elements FIELD
```

```

3      * <pre>
4      * <FIELD EMPTYOK="NO" MAXREPEAT="1" NAME="PatNachname" TYPE="
        TEXT"/>
5      * </pre>
6      */
7      private Element createFIELD(String emptyok, String maxrepeat,
        String name, String type){
8          Element fieldEle = dom.createElement("FIELD");
9          fieldEle.setAttribute("EMPTYOK", emptyok);
10         fieldEle.setAttribute("MAXREPEAT", maxrepeat);
11         fieldEle.setAttribute("NAME", name);
12         fieldEle.setAttribute("TYPE", type);
13         return fieldEle;
14     }

```

Das letzte an die Wurzel anzuhängende Element ist RESULTSET. Es speichert die Daten aus der Tabelle Patients.

```

1      /*<RESULTSET FOUND="">
2      */
3      Element resultsetEle = dom.createElement("RESULTSET");
4      resultsetEle.setAttribute("FOUND", "");

```

In einer Schleife werden die einzelnen Zeilen der Tabelle abgearbeitet. Existiert für einen Patienten ein Stornokennzeichen (CANCEL_IND="X"), wird diese Zeile ignoriert.

```

1      for (int i = 0; i < tabelle.getNumRows(); i++) {
2          tabelle.setRow(i);
3          if (tabelle.getString("CANCEL_IND").equals("X")
4              ) {
5              }
6          else {
7              Element rowEle=createROW(tabelle.getString("
                LAST_NAME_PAT"),
8                  tabelle.getString("FRST_NAME_PAT"),
9                  tabelle.getString("DOB"),
10                 tabelle.getString("PATIENTID"),
11                 tabelle.getString("STR_NO"),
12                 tabelle.getString("PCD"),

```

```

13         tabelle.getString("CITY"),
14         tabelle.getString("PHONENO"));
15     resultSetEle.appendChild(rowEle);
16     }
17 }

```

Durch den Aufruf der Hilfsmethode createROW wird ein neues ROW-Element erstellt. Dieses wird unter RESULTSET in den Baum eingehangen. Da die BAPIs die interne Repräsentation der Daten in SAP ausgeben, ist eine Umformung des Geburtsdatums in dessen deutsche Form von Nöten.

```

1     /**
2     * Hilfsmethode zum Aufbau des Elements ROW
3     * DATEFORMAT="D.m. yyyy"
4     * <pre>
5     * <ROW MODID="" RECORDID="">
6     * </pre>
7     */
8     private Element createROW(String patnachname, String patvorname
9
10        String geburtsdatum, String pat_id, String
11           strasse ,
12        String plz, String ort, String telefon) {
13
14        //Umformatieren: von yyyy-dd-mm zu dd.mm.yyyy
15        String geb_edit = "dd.mm.yyyy";
16        geb_edit = geburtsdatum.substring(8, 10) + "." +
17        geburtsdatum.substring(5,7) + "." + geburtsdatum.substring
18        (0,4);
19
20        Element rowEle = dom.createElement("ROW");
21        rowEle.setAttribute("MODID", "");
22        rowEle.setAttribute("RECORDID", "");
23
24        rowEle.appendChild(createCOL(patnachname));
25        rowEle.appendChild(createCOL(patvorname));
26        rowEle.appendChild(createCOL(geb_edit));
27        rowEle.appendChild(createCOL(pat_id));
28        rowEle.appendChild(createCOL(strasse));
29        rowEle.appendChild(createCOL(plz));

```

```

27         rowEle.appendChild(createCOL(ort));
28         rowEle.appendChild(createCOL(telefon));
29
30         return rowEle;
31     }

```

Jede Reihe der Tabelle besitzt Spalten. Für die Abbildung in XML erhält jedes ROW-Element mit Hilfe der Methode createCOL „Spaltenelemente“.

```

1     /**
2     * Hilfsmethode zum Aufbau des Elements COL
3     * <pre>
4     * <COL>
5     *     <DATA>Test</DATA>
6     * </COL>
7     * </pre>
8     */
9     private Element createCOL(String data){
10
11         Element colEle = dom.createElement("COL");
12         Element dataEle = dom.createElement("DATA");
13         Text dataText = dom.createTextNode(data);
14         dataEle.appendChild(dataText);
15         colEle.appendChild(dataEle);
16         return colEle;
17     }

```

Nachdem RESULTSET vollständig aufgebaut wurde, wird es unter dem Wurzelknoten eingehangen:

```

1         fmpxmlEle.appendChild(resultSetEle);
2     }

```

Die aufgebaute Instanz des DOM wird mit der Methode printToFile auf die Festplatte geschrieben. Der Name der Datei wird aus den beim Programmstart übergebenen Argumenten gebildet. Vom Nachnamen und Vornamen werden jeweils die ersten beiden Buchstaben genutzt. Bei einer zeitnahen mehrfachen Ausführung der Abfrage wird so vermieden, dass die angelegte Datei keine falschen Patientenstammdaten enthält.

```

1     /**

```

```

2      * XML-Datei mit einzigartigem Namen auf Festplatte schreiben
3      */
4      private void printToFile(){
5
6          try
7          {
8              OutputFormat format = new OutputFormat(dom);
9              format.setIndenting(true);
10
11             XMLSerializer serializer = new XMLSerializer(
12                 //Ausgabedatei einzigartig benennen
13                 new FileOutputStream(new
14                 File("\\\\hbwxpfm9\\Patients\\patient" + NachName.
15                 substring(0,2) + VorName.substring(0,2) +
16                 tabelle.getString("DOB").substring(0, 4) + ".xml"
17                 )), format);
18
19             serializer.serialize(dom);
20
21         } catch(IOException ie) {
22             ie.printStackTrace();
23         }
24     }

```

Mit dem Speichern der Patientenstammdaten hat SAP Connector seine Aufgabe erfüllt. Im nächsten Kapitel wird die Abfrage aus Heartware-Sicht betrachtet.

4.2.2 Heartware

In Heartware wird die Abfrage initialisiert und in Heartware werden die Ergebnisse derer importiert. Die einzelnen Schritte, die dafür notwendig sind, werden in einem Skript „Ausführung“ zusammengefasst. Um die Komplexität zu verbergen sind aus dem Skript einzelne Aufgaben in Unterskripte ausgelagert worden. Dieses Skript soll nun im Detail vorgestellt werden, jeder einzelne Schritt erhält eine Beschreibung.

```

1 Wenn [
2   IstLeer ( ${sap-bapi}::
3     GeburtsDatum )
4   ODER IstLeer ( ${sap-bapi}::
5     PatNachname )
6   ODER IstLeer ( ${sap-bapi}::
7     PatVorname )
8   ODER ( NICHT IstGültig ( ${sap-bapi
9     }::GeburtsDatum )
10    ODER NICHT IstGültig ( ${sap-
11    bapi}::PatNachname)
12    ODER NICHT IstGültig ( ${sap-
13    bapi}::PatVorname ) )
14 ]
15 Eigenes Dialogfeld anzeigen["
16   Patientenstammdaten fehlen";
17 "Für die Abfrage müssen Name, Vorname
18   und Geburtsdatum eingetragen und
19   gültig sein"
20 Aktuelles Script verlassen []
21 Ende (Wenn)

22 Eigenes Dialogfeld anzeigen["SAP-
23   Abfrage";
24 "Diese Aktion importiert Daten des
25   Patienten aus SAP ISH-Med in den
26   aktuellen Datensatz. Sie werden um
27   die Auswahl des richtigen
28   Patienten gebeten. Fortfahren?"

29 Wenn [
30   Falls ( Hole ( LetzteMeldungswahl
31     )=2; 2 )]
32   Script ausführen["SAP-Abfrage"]
33   Scriptpause setzen[Dauer (Sekunden
34     ): 1]

```

Wenn Felder leer sind oder nicht gültig bzgl. Felddefinition sind, dann Dialogfeld mit Hinweis anzeigen. Beenden des Skripts.

Ein Hinweis vor der bevorstehenden Aktion wird in einem Dialogfeld eingeblendet. Abbruch möglich.

Wenn Dialogfeld bestätigt wurde, Skript „SAP-Abfrage“ ausführen, anschließend 1 Sekunde warten.

19 Datensätze importieren [Ohne Dialogfeld; 20 "\\\"&\"\\\"&\"hbwxp9m9\Patients\patient " 21 & ZeichenLinks (\${sap-bapi}:: PatNachname; 2) 22 & ZeichenLinks (\${sap-bapi}:: PatVorname; 2) 23 & Jahreszahl (\${sap-bapi}:: GeburtsDatum) 24 & ".xml"	Pfad zur neu erstellten xml-Datei, Zusammen- setzen des Dateinamens nach dem Muster „patientNNVVJJJJ.xml“, wobei NN und VV jeweils die ersten beiden Buchstaben des Namens dar- stellen; Nur passende importieren, Rest verwerfen. Keine neuen Datensätze an- hängen.
25 Sonst 26 Eigenes Dialogfeld anzeigen[" Abbruch"; 27 "es wurde keine Verbindung aufgebaut" 28 Ende (Wenn)	Aktion wurde vom Anwender abgebro- chen, wird mit einem Dialogfeld bestä- tigt.
29 Script ausführen[" patient.xml löschen "]	Löschen der xml-Datei

Das Skript „SAP-Abfrage“ startet das Programm SAP Connector, die zu übergebenden Argumente werden zusammengesetzt. Die am Ende des Skripts gesetzte Pause ist empirisch ermittelt und soll SAP Connector Zeit geben, die Abfrage durchzuführen und das Ergebnis auf die Festplatte zu schreiben.

```

1  "java -jar " &"\"&"\"&"hbwxpfm9\
    sapjco\sapconnector.jar "
2  & ${sap-bapi}::PatNachname & " "
3  & ${sap-bapi}::PatVorname & " "
4  & LiesAlsText ( Jahreszahl (
    LiesAlsDatum ( ${sap-bapi}::
    GeburtsDatum) ) )
5  & Falls ( MonatZahl ( LiesAlsDatum (
    ${sap-bapi}::GeburtsDatum) ) <
    10; "0" )
6  & LiesAlsText ( MonatZahl (
    LiesAlsDatum ( ${sap-bapi}::
    GeburtsDatum) ) )
7  & Falls ( KalendertagZahl (
    LiesAlsDatum ( ${sap-bapi}::
    GeburtsDatum) ) < 10; "0" )
8  & LiesAlsText ( KalendertagZahl (
    LiesAlsDatum ( ${sap-bapi}::
    GeburtsDatum) ) )
9  Scriptpause setzen[Dauer (Sekunden):
    2]

```

Pfad zur auszuführenden Datei, Zusammensetzen der Parameter nach dem Muster „Name Vorname JJJJMMTT“; 2 Sekunden Skriptpause während der Abfrage

Ein Beispielaufruf für die Patientin Anna Test, geboren am 1. Januar 1911 sähe so aus:

```
java -jar \\hbwxpfm9\sapjco\sapconnector.jar Test Anna 19110101
```

Das Skript „patient.xml löschen“ kümmert sich um die Entfernung der für die Abfrage geschriebenen Datei. Die Nomenklatur des Dateinamens ist hier genauso zu wählen wie im Programm SAP Connector.

```

1  "\"&"\"&"hbwxpfm9\sapjco\delxml.bat
    "
2  & ZeichenLinks ( ${sap-bapi}::
    PatNachname; 2)
3  & ZeichenLinks ( ${sap-bapi}::
    PatVorname; 2)
4  & Jahreszahl ( ${sap-bapi}::
    GeburtsDatum )

```

Pfad zur Batchdatei, Zusammensetzen des Parameters nach dem Muster "NNVVJJJJ", wobei NN und VV jeweils die ersten beiden Buchstaben des Namens darstellen

Der Beispielaufruf für die Testpatientin lautet

\\hbwxp9\sapjco\delxml.bat TeAn1911.

Das eigentliche Löschen der Datei erledigt die Stapelverarbeitung delxml.bat.

4.3 Probleme und Fehlerverhalten

Die indirekte Kommunikation, die zum einen dateibasiert, zum anderen datei-los realisiert wurde, ist keine elegante Lösung, um ein Fachabteilungsinformationssystem in ein KIS zu integrieren. Neben den Problemen, die während der Entwicklung auftraten und gelöst wurden, wurde auch das Fehlerverhalten zur Laufzeit im Produktivbetrieb betrachtet. In einem verteilten System ist auch die Verfügbarkeit der beteiligten Komponenten der unidirektionalen Schnittstelle zu berücksichtigen. Wie Fehler verhindert dem Anwender kommuniziert werden, soll nachfolgende Auflistung klären:

Fehler	Vorgehen
- kein Datensatz in SAP für Suchkriterien vorhanden	→ nicht möglich, da Aufnahme über SAP
- Fehler im abfragenden Programm, keine xml-Datei geschrieben	→ Meldung in Heartware über Fehler, keine Importierung
- Zu viele Entsprechungen, ab 100 sperrt SAP die Ausgabe	→ (nicht möglich, da erwartet werden kann, dass wenige Patienten mit identischen Kriterien im Datenbestand existieren), Meldung in Heartware über Fehler
- Nicht legale Werte für Abfrage (z.B. 32.10.2001)	→ Skript mit Plausibilitätsprüfung vor Ausführung der Abfrage
- Doppelaufruf des Skripts bei gleichem Patienten	→ Doppelaufruf an einem Arbeitsplatz durch Skriptpause nicht möglich; Bei einem Doppelaufruf an verschiedenen Arbeitsplätzen sperrt FileMaker Server die gleichzeitige Änderung eines Datensatzes.

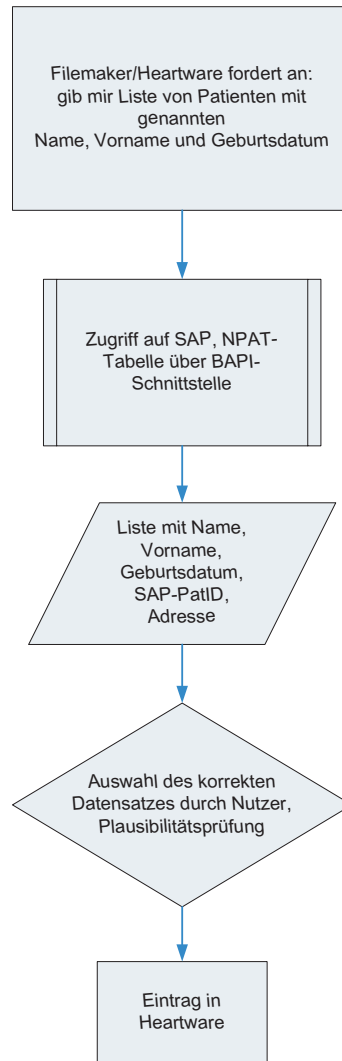


Abbildung 4.2: Definition einer gerichteten SAP-Schnittstelle mit dem Patientendokumentationssystem Heartware

Kapitel 5

Fazit

Ein Krankenhausinformationssystem stellt hohe Anforderungen an seine konzeptionelle Gestaltung. Das Ziel ist die optimale Integration aller zum KIS zugehörigen Informationssysteme, um zeitnah und von jedem Arbeitsplatz im Krankenhaus den Zugriff auf Patientendaten zu erhalten. Die Vorteile sind für den Patienten eine bestmögliche Betreuung sowie kürzere Aufenthaltszeiten. Für das Krankenhaus liegen die Vorteile bei einer wirtschaftlicheren Ausnutzung der Ressourcen und damit einer verbesserten Wettbewerbsfähigkeit. Außerdem transportieren zufriedene Patienten ein positives Bild des Krankenhauses. Für die Forschung ist ein großer Datenbestand, der aus vielen verschiedenen Informationssystemen gespeist wird, eine große Bereicherung. Das Data-Warehouse KIS ermöglicht es, Fragestellungen zu beantworten, die ohne die Datenintegration nur durch manuelles Zusammentragen von Treffern aus Abfragen verschiedener Informationssysteme ermittelt werden konnten. Selten wird ein KIS komplett neu entworfen, oft bestehen ältere gewachsene IT-Strukturen neben Neuanschaffungen, die den neueren technologischen Entwicklungen, wie der Unterstützung von Kommunikationsstandards á la HL7, bereits Rechnung tragen. Der erste Schritt zur Integration ist daher das Herstellen der Kommunikation untereinander, möglicherweise vereinfacht durch einen Kommunikationsserver. Das bloße Bereitstellen der Inhalte, zum Beispiel über eine browserbasierte Präsentation, ist keine Integration von Informationssystemen, nicht einmal eine Kommunikation zwischen ihnen. Die Problematik der nebeneinander auf einem klinischen Arbeitssystem existierenden Anwendungen wurde auch im Rahmen dieser Arbeit angegan-

gen, indem die Insellösung Heartware um eine unidirektionale Schnittstelle zu SAP erweitert wurde. Eine vollständige Integration im Sinne eines KIS ist diese Lösung immer noch nicht. Dafür bedarf es zum Einen einer bidirektionalen Schnittstelle, die es erlaubt, innerhalb von SAP die Heartware-Befunde eines Patienten aufzurufen. Zum Anderen unterstützt die entwickelte Kommunikationslösung nicht den Anspruch einer standardisierten Schnittstelle, die auf einem Nachrichtenstandard wie HL7 aufbaut. Begründet liegt das darin, dass das Heartware-basierende Filemaker nicht für den Gesundheitssektor im Speziellen entwickelt wurde und damit entsprechende Schnittstellen für die HL7-geführte Kommunikation nicht vorgesehen sind. Wenn es diese in naher Zukunft gäbe, könnte es ein integriertes medizinisches Informationssystem Heartware geben, wenn auch einige der Nachteile weiter bestehen bleiben. Die Entscheidung für oder gegen einen gesicherten Weiterbetrieb ist dann aber möglicherweise weniger wirtschaftlicher Natur.

Die Priorität liegt heute eindeutig bei der Integration aller in einem Krankenhaus erfassten Daten eines Patienten in ein gemeinsam genutztes KIS. Daher muss heute darüber nachgedacht werden, wie die Vorteile, die das kardiologische Patientendokumentationssystem Heartware zweifellos bietet, in Anforderungen für ein neues integriertes Fachabteilungsinformationssystem gerettet werden können.

Literaturverzeichnis

- [Bou94] BOURKE, Michael K.: *Strategy and Architecture of Health Care Information Systems*. New York : Springer-Verlag, 1994
- [DS03] DUGAS, Martin ; SCHMIDT, Karin: *Medizinische Informatik und Bioinformatik*. Heidelberg : Springer, 2003
- [Dud86] DUDECK, Joachim: HELP - ein medizinisches Verwaltungs- und Informationssystem. In: *GI Jahrestagung 2* (1986), S. 530–540
- [Fil04] FILEMAKER, Inc.: *FileMaker Pro 7 Benutzerhandbuch*. Santa Clara, CA, USA, 2004. http://www.filemaker.de/downloads/pdf/documentation/fm7_Benutzerhandbuch.pdf
- [GSD08] GSD, T-Systems A.: *i.s.h.med.* <http://www.ishmed.de>. Version: 10 2008
- [GSP⁺03] GELL, G. ; SCHMUCKER, P. ; PEDEVILLA, M. ; LEITNER, H. ; NAUMANN, J. ; FUCHS, H. ; PITZ, H. ; KOLE, W.: SAP and partners: IS-H and IS-H* MED. In: *Methods Inf Med* 42 (2003), Nr. 1, S. 16–24
- [Haa05] HAAS, Peter: *Medizinische Informationssysteme und elektronische Krankenakten*. Berlin : Springer Verlag, 2005. – 756 S. <http://dx.doi.org/10.1007/3-540-26855-3>
- [Han93] HANSEN, Wolf R.: *Client-Server-Architektur: Grundlagen und Herstellerkonzepte für Downsizing und Rightsizing*. Reading (MA) : Addison-Wesley, 1993

- [Hau01] HAUX, Reinhold et a.: Anforderungskatalog für die Informationsverarbeitung im Krankenhaus / Universität Heidelberg, Institut für Medizinische Biometrie und Informatik. Version: Januar 2001. <http://iig.umat.at/dokumente/anforderungskatalog.pdf>. 2001. – Forschungsbericht
- [HL788] HL7: *Health Level Seven Standards, Version 2.0: An Application Protocol for Electronic Data Exchange in Health Care Environment*. Philadelphia : Health Level-7, Inc, 1988
- [HL790] HL7: *Health Industry Level Seven Interface Standards, Version 2.1*. Chicago : Health Level-7, Inc, 1990
- [HL794] HL7: *Health Level Seven: an application protocol for electronic data exchange in health care environments, version 2.2*. Ann Arbor : Health Level-7, Inc, 1994
- [HL797] HL7: *Health Level Seven, Version 2.3*. Ann Arbor : Health Level-7, Inc, 1997
- [HL798] HL7: *Health Level Seven Implementation Support Guide for HL7 Standard Version 2.3*. Ann Arbor : Health Level-7, Inc, 1998 <http://www.hl7.org/Special/IG/final.pdf>
- [HLS08] HEALTH LEVEL SEVEN, Inc.: *Health Level 7*. www.hl7.org. Version: 12 2008
- [Lan03] LANGE, M.: *Semantische und syntaktische Integration von konkurrierenden Informationssystemen in der Medizin*, Münster (Westfalen), Diss., 2003
- [LS03] LENZ, R. ; SCHMÜCKER, P: Stete Weiterentwicklung nötig - Integration heterogener IT-Systeme im Krankenhaus. In: *Der Klinikarzt* 32 (2003), November, Nr. 11, 385-390. <http://www.thieme-connect.de/ejournals/pdf/klinikarzt/doi/10.1055/s-2003-44532.pdf>
- [Mos99] MOSER, Gerd: *SAP R/3 interfacing using BAPIs*. Braunschweig : Vieweg, 1999. – 219 S.

- [Pro94] PROKOSCH, Hans U.: *Ein Referenzmodell für den Einsatz wissensbasierter und wissensverarbeitender Funktionen innerhalb eines Krankenhaus-Informationssystems*. Gießen : Justus-Liebig-Universität Gießen, 1994. – 200 S. – Gießen, Univ., Habil.-Schr.
- [Pro01] PROKOSCH, Hans U.: KAS, KIS, EKA, EPA, EGA, E-Health: Ein Plädoyer gegen die babylonische Begriffsverwirrung in der Medizinischen Informatik. In: *Informatik, Biometrie und Epidemiologie in Medizin und Biologie* 32 (2001), Nr. 4, S. 371–382
- [SAP06] SAP®: *IS-H/IS-H*MED*. http://help.sap.com/saphelp_ish472/helpdata/DE/59/1c86353e13d558e10000009b38f889/frameset.htm.
Version: April 2006
- [SAP07] SAP®: *BAPI-Benutzerhandbuch (CA-BFA)*. http://help.sap.com/saphelp_46c/helpdata/de/7e/5e141a4a1611d1894c0000e829fbbd/frameset.htm. Version: Juli 2007
- [Sch00] SCHADOW, Gunther: Krankenhauskommunikation mit HL7. In: *Berichte aus der medizinischen Informatik und Bioinformatik*. Aachen : Shaker, 2000, S. 360. – Zugl.: Berlin, Freie Univ., Diss., 1999
- [Sch02] SCHUESSLER, Thomas G. ; ARASOFT GMBH (Hrsg.): *Developing Applications with the SAP Java Connector (JCo)*. 0.8.1. Arasoft GmbH, 2002
- [Wen05] WENTZ, Bernhard et a.: 10 Jahre Erlanger Kommunikationsdrehscheibe - der Weg zu einer zukunftssicheren Integrationsplattform. In: *50. Jahrestagung der Deutschen Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie e.V. (gmds)*. Freiburg im Breisgau : GMDS, September 2005
- [Zlo77] ZLOOF, M.M.: Query-by-Example: A Data Base Language. In: *IBM Systems Journal* 16 (1977), Nr. 4, S. 324–343

Selbständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit mit dem Titel *“Studienarbeit zur Integration von Krankenhausinformationssystemen am Beispiel von SAP und FileMaker”* selbständig, nur unter Zuhilfenahme der angegebenen Quellen und ohne unerlaubte Hilfe verfasst habe.

Berlin, 18. Januar 2009

Daniel Neumann

Einverständniserklärung

Ich bin damit einverstanden, dass die vorliegende Arbeit mit dem Titel *“Studienarbeit zur Integration von Krankenhausinformationssystemen am Beispiel von SAP und FileMaker”* in der Bibliothek ausgelegt wird.

Berlin, 18. Januar 2009

Daniel Neumann