

Aus der Klinik für Frauenheilkunde und Geburtshilfe
des Universitätsklinikums Charité
der Humboldt-Universität zu Berlin
Campus Charité Mitte
Direktor: Prof. Dr. med. W. Lichtenegger



DISSERTATION

Konzept einer an semantischen Kriterien orientierten
Kommunikation für medizinische Informationssysteme

Zur Erlangung des akademischen Grades
doctor rerum medicinalium (Dr. rer. medic.)

vorgelegt der Medizinischen Fakultät Charité
der Humboldt-Universität zu Berlin

von

Herrn Dipl.-Ing. Trong-Nghia Nguyen-Dobinsky

geb. am 13.03.1953 in Thanh-Hoa, Viet-Nam

Dekan: Prof. Dr. med. M. Dietel

Gutachter: 1. PD Dr. rer. nat. J. Michel
2. Prof. Dr. med. R. Bollmann
3. Prof. emeritus of CS J. Weizenbaum

eingereicht: Juni 1997

Datum der Promotion: 3. April 1998

Zusammenfassung

Einleitung

In einem größeren Universitätsklinikum wie in der Charité sind EDV-gestützte Verfahren in verschiedenen Einrichtungen und für verschiedene Aufgaben im Einsatz: Verwaltung, Krankenversorgung, Forschung und Lehre. Diese Subsysteme sind in der Regel nicht in der Lage, Daten untereinander so auszutauschen, daß die in den Daten enthaltene Semantik nicht verlorengeht. Die Ursache liegt im wesentlichen in der Komplexität und in der Unschärfe der medizinischen Informationen. Medizinische Standards (HL7, DICOM, SNOMED, ICD, ICPM, ...) lassen sich für den Austausch von Daten verwenden, die gut formalisierbar und mit einer klaren Bedeutung behaftet sind. Nicht formalisierbare Daten, die z. B. in einem Befund oft vorkommen, lassen sich nicht ohne weiteres mit diesen Standards darstellen.

Ziel

Entwicklung eines Konzeptes für den Austausch medizinischer Daten, das die o. g. Probleme vermeidet.

Material und Methoden

Die Analyse der vorhandenen Subsysteme, Standards und Konzepte zeigt, daß das Konzept einerseits eine sehr einfache Syntax und eine simple Struktur aufweisen muß. Andererseits muß die medizinische Semantik voll erhalten bleiben. Als Vorbild kann die relationale Datenbank dienen, die mit einem Datentyp (Relation bzw. Tabelle) und einem einzigen Operator (SELECT) auf diesen Datentyp auskommt.

Ergebnisse

Das Konzept ist objektorientiert. Es enthält nur einen Datentyp. Das ist das AMICI-Objekt (AMICI: Architecture for Medical Information Exchange and Communication Interface). Über dieses AMICI-Objekt wird der gesamte Datenaustausch vorgenommen. Kann das Empfängersystem ein Objekt nicht oder nicht korrekt interpretieren, so wird die Interpretation vom Sendesystem übernommen. Ein Subsystem wird im Netzwerk über einen medizinischen Kontext angeschlossen, der das Interessengebiet und die Fähigkeit des Subsystems beschreibt. Das Subsystem kann an Hand der im Netz bekannten medizinischen Kontexte feststellen, welche weiteren Subsysteme für den eigenen Zweck interessant sein könnten. Alle AMICI-Objekte erhalten eine weltweit eindeutige Identifikation, so daß die Daten aus verschiedenen Institutionen, auch international, miteinander gemischt werden können.

Diskussion

Das Konzept kann als Basis für weitere Dienstleitungen in einem Klinikum bzw. einem Krankenhaus dienen. Namentlich zu nennen sind telemedizinische Anwendungen, bei denen nicht nur die Kommunikation zwischen Ärzten, sondern auch zwischen Patienten und Arzt möglich ist. Weiterhin betrifft dies den Einsatz von

Software-Agenten, die sich um den Informationsbedarf eines Arztes individuell kümmern.

Schlagworte

Medizinischer Kontext, Standard für den Austausch medizinischer Daten, Semantik medizinischer Daten, medizinische Informationssysteme.

Abstract

Introduction

Large hospitals like the University hospital Charité use in different units different information systems for recording patient and medical data. There are also different tasks: administration, healthcare, research and education. These medical information systems are often called subsystems. They are usually not able to exchange data without loss of semantic. The complexity and the variability of medical terminology cause this problem. Existing medical standards (e. g. HL7, DICOM, SNOMED, ICD, ICPM, ...) are helpful for well formalised terms. Non-formalised terms that are often used in diagnostic reports can not be represented by existing standards.

Aims

Development of a concept for medical information exchange which fulfills the requirements mentioned above.

Material and Methods

The system analysis that is performed based on existing subsystems, medical standards and concepts provides two essential requirements. On the one hand the syntax of such standard must be extremely simple. On the other hand the standard must be able to transfer extremely complex semantics. As an example relational databases (RDB) provide a good idea of such simple syntax and complex semantics. RDB's include only one data type. It is called relation or table. To manipulate tables one needs only one operation. That is the SELECT command in SQL.

Result

The concept is object oriented. It includes only one object called AMICI-object like RDB's (AMICI: Architecture for Medical Information Exchange and Communication Interface). Data exchange is completely performed by these AMICI-objects. If the receiving subsystem is not able to interpret and represent an object, the sending subsystem will take over this task. Within a network a subsystem uses a special AMICI-object called medical context to describe its features and its area of interest. A subsystem can inquire medical contexts to explore installed and running subsystems in the network. An international unique identifier identifies every AMICI object so that you can mix objects provided by different international institutions, e. g. to use them in multi-center-studies.

Discussion

This concept can also be used as a basic service for higher level applications in a hospital. Two of them are telemedicine and software agents. Telemedicine is not only a tool for physicians. It should be also a tool for communication and interaction between patient and physician.

Physicians can use personal software agents for information acquisition, which meets exactly his specific requirements.

Keywords

Medical context, standard for exchange of medical data, semantics of medical data, medical information systems.

Inhaltsverzeichnis

1	EINLEITUNG	4
1.1	Problemstellung	4
1.1.1	Medizinisch-wissenschaftliche Probleme	5
1.1.2	Patientenbezogene Probleme	8
1.1.3	Technische Probleme	9
1.1.4	Rechtlicher Aspekt	11
1.1.5	Ableitung aus der Problemanalyse	12
1.2	Stand der technologischen Forschung	13
1.2.1	Die objekt-orientierte Vorgehensweise (OO)	14
1.2.2	Firmenspezifische Realisierungen	17
1.2.3	Vergleich COM/OLE/ActiveX und Java	21
1.2.4	Das Netzwerkprotokoll RPC	22
1.2.5	Das Netzwerkprotokoll HTML/XML	22
1.2.6	Der HL7-Standard	24
1.2.7	Der DICOM-Standard	25
1.2.8	Das SNOMED-Konzept des College of American Pathologists	25
1.2.9	Das Semantic-Network	26
1.2.10	Vollständige Eigenentwicklung	28
1.2.11	Die Analyse von „freien“ Befundtexten	29
1.2.12	Weitere Arbeiten auf dem Gebiet der medizinischen Standards	29
1.3	Resumée	30
2	AUFGABENSTELLUNG	31
2.1	Systemabgrenzung	31
2.2	Anforderungskatalog	33
2.2.1	Maximale Semantik	34
2.2.2	Minimale Syntax	35
2.2.3	Praxisnähe	36
3	METHODIK	37
3.1	Lösungsrahmen	37
3.2	Systemphilosophie	38
3.3	Entwurfsphilosophie	39
3.4	Das allgemeine Objektkonzept	39
3.4.1	Motivation	39
3.4.2	Unterschied zum herkömmlichen Objektkonzept	40
3.5	Syntax der Objektbeschreibung	41
3.6	Der medizinische Kontext	42
3.6.1	Zweck des medizinischen Kontextes	42
3.6.2	Struktur des medizinischen Kontextes	43
3.6.3	Kontextwechsel	44
3.6.4	Eigenschaften medizinischer Kontexte	44
3.6.5	Kontext-Matching	45
3.6.6	Syntax der Kontextbeschreibung	46
3.6.7	Beispiel einer Transformation	47
3.7	Der virtuelle Namensraum	48
3.8	Das Transaktionskonzept	49

4	ERGEBNISSE	50
4.1	Systementwurf	51
4.1.1	Softwarestruktur	51
4.1.2	Programmablauf	53
4.2	Softwarekomponenten	55
4.2.1	Allgemeine Kommunikation zwischen den Komponenten	55
4.2.2	Das Treiber-Konzept	56
4.2.3	Der virtuelle Befundviewer (VRV)	56
4.2.4	External Communication-Layer (ECL)	57
4.2.5	Das Knoteninterface (NI)	57
4.2.6	Der Transaktionsverteiler (TAD)	58
4.2.7	Der Object-Handler (OH)	58
4.2.8	Die Attribute-Engine (AE)	58
4.2.9	Das Device-Independent-Interface (DII)	59
4.2.10	Das Device-Dependent-Interface (DDI)	59
4.2.11	Der virtuelle Befundtreiber (VRD)	59
4.3	Datenentwurf	61
4.4	Das spezielle Objektkonzept	62
4.4.1	Objekt-Identifikation	63
4.4.2	Objekt-Inhalt	65
4.4.3	Objektbeziehungen	66
4.4.4	Objektmethoden	66
4.5	Attributklassen	66
4.6	Objektklassen	67
4.6.1	Beispiel für Objektdefinition	71
4.7	Die Subsysteme	71
4.7.1	Der invasive Knoten	72
4.7.2	Der geburtshilfliche Knoten	72
4.7.3	Der Pränatal-Ultraschall-Knoten	73
4.7.4	Der Fetalautopsie-Knoten	75
4.7.5	Der HL7-Knoten	76
4.7.6	Der allgemeine Knoten	77
4.7.7	Der Referenzknoten	77
4.8	Testimplementierung	77
4.8.1	Zugriffe auf fremde Datenbanken	78
4.8.2	Abgleich zwischen verschiedenen Datenbanken	78
4.8.3	Antwortzeitverhalten der Zugriffe	79
5	DISKUSSION	81
5.1	Zielerreichung	82
5.2	Grenzen des Konzepts	84
5.3	Vergleiche mit anderen Konzepten	85
5.4	Nutzbarkeit des Konzepts für angrenzende Gebiete	89
5.5	Fazit	89
5.6	Ausblick	90
6	ZUSAMMENFASSUNG	93
7	LITERATURVERZEICHNIS	95

1 Einleitung

In einem größeren Krankenhaus wie beispielsweise der Charité wird die Informationstechnik auf verschiedenen Ebenen eingesetzt. Der Bedarf an automatischem Austausch von Informationen zwischen den eingesetzten EDV-Subsystemen ist in den vergangenen Jahren enorm gestiegen. Dieser Bedarf läßt sich z. Z. jedoch nur ungenügend befriedigen. Die Hindernisse für eine flächendeckende Integration von EDV-Subsystemen sind vornehmlich organisatorischer oder technischer Natur. Daneben gilt es finanzielle, personelle sowie politische Aspekte zu beachten.

Ein Konzept zur flächendeckenden Integration läßt sich nur erfolgreich realisieren, wenn alle Aspekte Berücksichtigung finden. Die vorliegende Arbeit befaßt sich jedoch ausschließlich mit dem technischen Aspekt des Problems, sowie mit dem medizinisch-wissenschaftlichen Teilaspekt, soweit dieser zur Analyse und Lösung des Problems erforderlich ist. Die Einbeziehung organisatorischer Aspekte würde den Rahmen dieser Arbeit sprengen. Ebenso wird der wirtschaftliche Aspekt nicht ausführlich behandelt. Bei der Konzeption wird lediglich darauf geachtet, daß erstens möglichst viele bereits vorhandene Ressourcen und Standards verwendet werden und zweitens die Realisierung keine besonderen wirtschaftlichen Anforderungen stellt.

Das Integrationskonzept wird an Hand des Beispiels Charité konkretisiert. Es läßt sich jedoch prinzipiell auf andere große Krankenhäuser übertragen. Bei der Anwendung auf städtische Krankenhäuser entfallen naturgemäß die Aspekte Forschung und Lehre.

Im einleitenden Kapitel werden zunächst die Probleme der Integration von EDV-Subsystemen beschrieben. Anschließend werden die aktuellen Ansätze in dem relevanten Bereich - anwendungsorientierte Kommunikationsprotokolle - untersucht. Schließlich wird an Hand der Problemanalyse das Ziel der vorliegenden Arbeit formuliert.

Die in dieser Arbeit verwendeten Begriffe sind meistens in der originären, englischen Fassung belassen, da sie im EDV-Sinne auch im Deutschen mit klar definierten Inhalten verknüpft sind, und ein deutsches Pendant nicht immer existiert.

1.1 Problemstellung

Versucht man die Informationen aus unterschiedlichen Anwendungen differenter medizinischer Disziplinen vor dem Hintergrund des Informationsflusses in einem Großkrankenhaus wie der Charité miteinander zu verknüpfen, so sieht man sich mit einer Reihe von Problemen konfrontiert.

Betrachtet man sie unter dem inhaltlichen Aspekt, so lassen sie sich in folgende Bereiche einordnen:

- medizinisch-wissenschaftliche,
- patientenbezogene,
- technische, sowie

- rechtliche Inhalte.

Prinzipiell sind andere Klassifikationen der Probleme möglich. In dieser Arbeit wird jedoch diese Einteilung gewählt, da sie auf den technischen Aspekt fokussiert ist. Abbildung 1-1 gibt einen Überblick über die auftretenden Probleme.

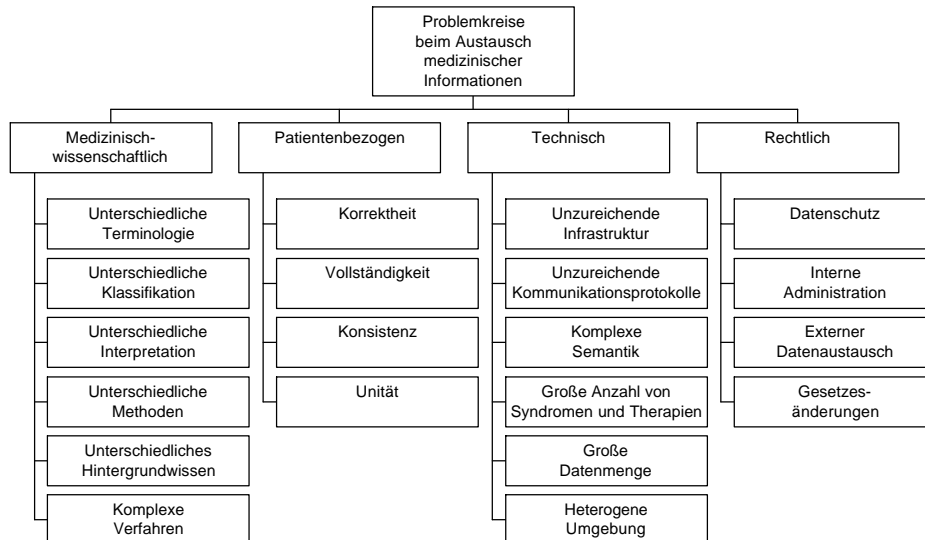


Abbildung 1-1: Strukturierung der Probleme

1.1.1 Medizinisch-wissenschaftliche Probleme

In der Charité besteht ein Bedarf an Informationssaustausch zwischen den Disziplinen. Z. Z. funktioniert der Datenaustausch weitgehend über Papier, sowie über gleichwertige Datenträger wie Photo, Film, Video, Dia. Ein papierloser, automatischer Informationsaustausch ist momentan nur in wenigen Fällen realisiert. So können z. B. Laborergebnisse aus der Pathobiochemie von Stationsarbeitsplätzen aus abgerufen werden.

Will ein Arzt einen Fall bearbeiten, so kann er in der Regel nur auf digitale Informationen der eigenen Einrichtung zurückgreifen. Daten aus anderen Disziplinen, die für diesen Fall notwendig sind, muß er sich auf konventionellen Datenträgern beschaffen. Dies ist dann besonders aufwendig, wenn ein Patient gleichzeitig von mehreren Disziplinen betreut wird. Eine Verknüpfung der Daten aus mehreren Disziplinen mit dem Ziel, alle relevanten patientenbezogenen Daten für die an der Behandlung beteiligten Ärzte übersichtlich darzustellen, ist meistens nicht möglich. Die Informationen sind nicht nach einem einheitlichen Schema strukturiert, so daß deren Vergleichbarkeit sehr stark darunter leidet. Es lassen sich beispielsweise Bildaufnahmen aus der Sonographie, Radiologie und Pathologie nur dann optimal miteinander vergleichen, wenn die Abbildungsebenen identisch gewählt sind.

Daten, die an sich nur einmal vorhanden sein müssen, jedoch von vielen Disziplinen benötigt werden, werden mehrfach erfaßt und auf papiernen Datenträgern ausgetauscht. Die Verfügbarkeit und die Aktualität dieser Informationen ist dementspre-

chend nicht hoch. Ein Beispiel hierfür ist die Anamnese. Sie wird bei vielen Diagnosen benötigt, liegt aber weder zentral noch online verfügbar für alle Beteiligten vor.

Eine interdisziplinäre Diagnostik und Therapie wird erschwert, weil die für die Diagnose eines Falles relevanten Informationen nicht zusammenhängend gespeichert und bereitgestellt werden können.

Ein repräsentatives Beispiel hierfür ist das Projekt CHILD, das ursprünglich innerhalb des vom BMFT geförderten Verbundprojekts „Risikoneugeborenes“ begonnen wurde. Dessen Ziel war die Verbesserung der pränatalen Diagnostik von Fehlbildungen. In diesem Projekt sollten Informationen über fetale Fehlbildungen so gespeichert werden, daß die beteiligten Disziplinen ohne weiteres auf diese Informationen zugreifen können. Primär gehören hierzu:

- die pränatale Diagnostik und Therapie,
- die klinische Genetik,
- die Zytogenetik,
- die Kinderradiologie,
- die Kinderchirurgie und
- die Kinderpathologie.

Hierbei treten Probleme auf, die aus dem unterschiedlichen Blickwinkel der Disziplinen, differenten Sprechweisen und sehr komplexen, disziplinspezifischen Verfahren resultieren:

- unterschiedliche Terminologie,
- unterschiedliche Klassifikation,
- unterschiedliche Interpretation,
- unterschiedliche Methoden,
- unterschiedliches Hintergrundwissen und
- komplexe Verfahren

Im folgenden Abschnitt werden an Hand von Beispielen die aufgeführten Probleme beschrieben.

Unterschiedliche Terminologie

Die in verschiedenen Disziplinen verwendete Terminologie bereitet für die automatisierte, formale Verarbeitung durch ein EDV-System große Probleme. Am Beispiel „Syndrom“ werden sie im folgenden erläutert.

Der Begriff Syndrom wird im allgemeinen als ein Muster von multiplen Anomalien oder Fehlfunktionen verwendet. In der Genetik wird der Begriff Syndrom gewöhnlich restriktiver als in anderen medizinischen Bereichen gehandhabt. Hier wird Syndrom als ein spezielles Muster von Anomalien verwendet, die auf die gleiche Ursache zurückzuführen sind. Dies wird auch genetische Anomalie genannt. Andere multiple Anomalien werden als nichtsyndromal bezeichnet. Sie werden auch Assoziation, Sequenzen oder Entwicklungsdefekt genannt. In anderen Disziplinen wird nicht selten unter Syndrom klassifiziert, was sich nicht eindeutig einer Krankheit zuordnen läßt, wobei die Definition von Krankheit auch nicht so eindeutig ist, daß der Entwurf eines einheitlichen Datenmodells möglich wäre. Ein Beispiel hierzu ist die unterschiedliche Betrachtungsweise von Krankheit bei Genetikern und Pathologen.

Dies führt dazu, daß ein automatisiertes EDV-System sein Datenmodell zunächst von den möglicherweise beteiligten Fachdisziplinen entkoppeln, bei der Präsentation der gespeicherten Information jedoch kontextgesteuert arbeiten muß. Es muß somit die Information in Abhängigkeit von der aktuellen Disziplin und ggf. vom Betrachter entsprechend aufbereitet werden.

Unterschiedliche Klassifikation

Zu der unterschiedlichen Terminologie kommt eine Reihe von unterschiedlichen Klassifikationen, z. B. von Organfehlentwicklungen. Viele Ärzte lassen diese unterschiedlichen Klassifikationen jedoch außer acht. Sie ziehen eine einfache Diagnose wie „zystische Nieren“ vor, obwohl diese Diagnose keine Hilfestellung für die Erkennung genetischer oder syndromaler Fehlfunktionen bietet. Sie kann zu einer Fehleinschätzung von Risiken und zu ineffektiven Diagnosemaßnahmen in der pränatalen Diagnostik verleiten. Die Benutzung von nicht mehr gültigen Klassifikationen wie der Potter-Klassifikation einer zystischen Niere kann ebenfalls zu einer Fehlinterpretation der Ätiologie führen.

Unterschiedliche Interpretation

Eine Verkleinerung eines Organs oder eine Reduktion der Zahl der Zellen kann durch viele Faktoren verursacht werden. Es kann eine Hypoplasie, eine Dysplasie, eine Dystrophie oder eine Atrophie vorliegen. Eine Fehlinterpretation hat hier natürlich Konsequenzen für das Verständnis der Natur dieser Anomalie (für die Einstufung: primäre oder sekundäre Diagnose, deren Zuordnung zu einem Syndrom..).

Eine automatisierte Verarbeitung solcher medizinischer Informationen erfordert hier eine gewisse Standardisierung als Vorarbeit. Sie ist z. Z. jedoch zumindest für den Bereich der pränatalen Diagnostik nicht in dem nötigen Umfang vorhanden. Das EDV-System muß momentan noch mit diesem Zustand „leben“.

Die unterschiedliche Interpretation medizinischer Begriffe, insbesondere der Syndrome und der Diagnose, ist von ähnlicher Natur wie die unterschiedlich verwendeten Klassifikationen.

Selbst innerhalb eines Fachgebiets variieren die Interpretationsmethoden von Klinik zu Klinik (Feigl, P. et al. [31]). Feigl untersucht die Vergleichbarkeit der medizinischen Daten mit Hilfe systematischer Codierung von 34 verschiedenen Angaben bei Krebsbefunden. Er fand, daß bei 25 Befunden 14% der Krankheitsangaben und noch immer 8% bei der einfachen Angabe des Datums der Diagnose nicht übereinstimmten.

Unterschiedliche Methoden

Selbst innerhalb einer Disziplin existieren Probleme, die aus der Tatsache resultieren, daß unterschiedliche Arbeitsgruppen mit unterschiedlichen Methoden und unterschiedlichen technischen Hilfsmitteln arbeiten. Dies führt zu unterschiedlicher Erfassung und Dokumentation von Untersuchungen.

Unterschiedliches Hintergrundwissen

Das Hintergrundwissen der künftigen Anwender bezüglich eines integrierten Systems ist sehr unterschiedlich. Verwaltungsangestellte, Krankenschwestern bzw. Pfleger, Hebammen, Studenten/Doktoranden, Ärzten und Wissenschaftlern werden es zwar für unterschiedliche Zwecke nutzen; sie sind jedoch alle gleichermaßen mit der Bedienung konfrontiert. Hieraus leiten sich bestimmte Anforderungen an die Bedienbarkeit des Systems und die Darstellung von medizinischen Sachverhalten ab. Jacobsen, T. J. , forderte ebenfalls eine einfache Bedienung für medizinische Informationssysteme [50].

Erstens muß die Bedienung so einfach gestaltet werden, daß ein mit nur wenig EDV-Hintergrundwissen ausgestatteter Anwender mit der Bedienoberfläche zurechtkommt. Am besten geeignet ist eine einheitliche, für alle Anwender bereits bekannte Oberfläche.

Zweitens muß diese Oberfläche den Anschluß an alle Informationsserver und Kommunikationshilfsmittel bereits mitbringen und zwar für alle in Frage kommenden Client-Plattformen.

Ein Beispiel hierfür kann die Oberfläche von Internet-Browsern liefern. Sie ist auf allen Client-Plattformen einheitlich und allen Anwendern, die bereits Erfahrung mit dem Internet haben, bekannt. Selbst bei geringer Erfahrung bieten solche Browser eine einfache und robuste Bedienung.

Komplexe Verfahren

Die in den verschiedenen Disziplinen der Medizin verwendeten Methoden wie Ultraschall, Röntgen, Computertomographie, Histologie, etc. sind so unterschiedlich, daß z. B. eine einheitliche Gestaltung der Erfassungsformulare praktisch nicht möglich ist. Selbst in einer Disziplin wie der pränatalen Diagnostik können verschiedene Ultraschallgeräte unterschiedliche Meßprotokolle liefern (2D/3D, mit/ohne Doppler, etc.).

Ein in annehmbarer Zeit realisierbares und einsetzbares System muß in der Praxis diesen Aspekt zwangsläufig eliminieren. Ein derartiges System darf sich nicht mit dem Inhalt eines zu integrierenden Subsystems beschäftigen. Alle Funktionen, die den Inhalt des jeweiligen Subsystems tangieren, müssen vom Subsystem bereitgestellt werden. Nur das Ergebnis solcher Funktionen läßt sich über die Grenzen der Subsysteme transferieren. Beispielsweise soll eine Messung an einem Gefäß vom Subsystem durchgeführt werden. Das Ergebnis der Messung - das ist eine einfache Zahl mit einer Dimensionsangabe - wird an das Empfängersystem geschickt.

1.1.2 *Patientenbezogene Probleme*

Da die Suche nach einem Befund meistens über den Patienten/die Patientin geschieht und alle weiteren Aktivitäten auf diese Person bezogen sind, stellt der Patient bzw. die Patientin einen wesentlichen Faktor in einem medizinischen System dar. Hier existieren jedoch wegen des niedrigen Kommunikationsgrades zwischen den Subsystemen noch immer Probleme durch:

- unkorrekte Daten,
- unvollständige Daten,
- inkonsistente Daten und
- mehrfach vorkommende Daten

Unkorrekte Daten

Die Daten über einen Patienten/eine Patientin werden meistens auf Papier und handschriftlich zwischen den Einrichtungen ausgetauscht. Sehr oft geschieht dies auch telefonisch. Diese Daten werden dann manuell in das empfangende System eingegeben (abgetippt). Dabei ist nicht immer gewährleistet, daß die Daten eines Patienten/einer Patientin in allen Subsystemen korrekt und einheitlich erfaßt werden. Hohnloser, J. H., et al. hatten bei einer Prüfung von 5 Befundparametern festgestellt, daß 3% bis 28% Daten falsch und 64% Daten überhaupt nicht codiert sind [46]. Fleege, J. C., versuchte, die Daten bereits bei der Eingabe auf die Korrektheit mit mehreren Methoden wie Datentyp, Wertebereich, Quercheck, Data Dictionary, etc. zu validieren, um die Brauchbarkeit der Eingabedaten für spätere Auswertung zu erhöhen [46]. Whates, P. D., et al. ermittelten eine signifikante Anzahl von falschen Eingaben beim Vergleichen von Programmausdrucken mit Patientendaten [85].

Unvollständige Daten

Da jede Einrichtung, sogar jeder Mitarbeiter, die Daten in unterschiedlichem Umfang und auf unterschiedliche Art und Weise erfaßt, sind die Daten nicht immer vollständig. Bei Bedarf stehen die benötigten Informationen nicht online zur Verfügung, und es müssen dann zeitraubend papierne Akten durchsucht werden.

Inkonsistente Daten

Ähnlich wie oben beschrieben, können falsche Eingaben inkonsistente Daten verursachen. Über ein und denselben Patienten existieren dann möglicherweise mehrere widersprüchliche Informationen.

Mehrfach vorkommende Daten

Durch unvollständige und/oder falsche Eingaben kann es zu mehrfach vorkommenden Daten kommen. Beispielsweise werden die Geburtsnamen bei der schwangeren Patientin nicht eingegeben, obwohl der Geburtsname noch immer ein gutes Merkmal für die Identifikation darstellt. Nach einer Heirat kann sich der Name der Patientin ändern. Bei einer folgenden Untersuchung wird nur der neue Name eingegeben. Für die eine Person werden dann mehrere voneinander unabhängige Datensätze gespeichert. Eine umfassende Information über diese Patientin ist somit nicht möglich. Eine darauf aufbauende statistische Auswertung liefert dann erwartungsgemäß falsche Ergebnisse.

1.1.3 Technische Probleme

Die technischen Probleme äußern sich in folgenden wesentlichen Punkten:

- unzureichende EDV-Infrastruktur,
- unzureichende Kommunikationsprotokolle,

- komplexe Semantik,
- große Anzahl von Syndromen und Therapien,
- große Datenmenge und
- heterogene Umgebung

Diese Probleme werden im folgenden Abschnitt beschrieben. Zunächst wird die aktuelle Struktur der EDV-Kommunikation der Charité kurz umrissen. Die daraus resultierenden Probleme für eine effektive Nutzung werden anschließend analysiert.

Unzureichende EDV-Infrastruktur

Es ist anzunehmen, daß in allen größeren Krankenversorgungseinrichtungen, ähnlich wie in der Charité, nicht in jeder Klinik, in jeder Abteilung eine eigene EDV-Infrastruktur zur Verfügung steht, die für eine moderne datenbank- und netzwerkorientierte EDV notwendig ist. Man kann andererseits nicht erwarten, daß eine solche Infrastruktur schlagartig in allen Kliniken/Abteilungen aufgebaut wird. Das geplante System muß anfänglich mit minimaler EDV-Infrastruktur auskommen.

Unzureichende Kommunikationsprotokolle

Wie bei der EDV-Ausstattung fehlt auch eine flächendeckende Infrastruktur für das „Networking“. Das geplante System muß mit dieser Tatsache klarkommen, d. h. es muß mit minimaler Kommunikationsstruktur einen funktionsfähigen Datenaustausch ermöglichen. Die Kommunikation muß einerseits auf einem weit verbreiteten und preiswerten Protokoll basieren. Andererseits müssen ebenfalls leistungsfähige und komplexe Protokolle wie FDDI oder ATM unterstützt werden, um einen hohen Datendurchsatz zu erzielen. Über diese Anforderung hinaus muß das System unabhängig von verfügbaren Kommunikationsprotokollen funktionieren.

Komplexe Semantik

Wie oben erläutert, wird eine künftige Kommunikation umfangreiche medizinische Inhalte bewegen müssen. Diese Inhalte dürfen bei einem Transport von einem medizinischen Subsystem zum anderen nicht verlorengehen. Das bedeutet, daß eine komplexe medizinische Semantik übertragen werden muß. Diese Komplexität ist in den für EDV-Verhältnisse unscharfen Definitionen von medizinischen Sachverhalten wie Krankheit, Syndrom, etc. begründet. Die Schnittstelle muß diese Semantik unterstützen, was eine essentielle Anforderung an das System bezüglich der Einsetzbarkeit bedeutet.

Große Anzahl von Syndromen und Therapien

Neben dem Problem der komplexen Semantik besteht auch noch das Problem der außerordentlich großen Anzahl von Syndromen und Therapien. Um die medizinische Semantik bei der Speicherung und Übertragung von Daten zu erhalten, muß der Syndrombegriff genügend klar und umfassend definiert werden. Dies ist jedoch eine große Anforderung, da z. Z. noch keine für den EDV-Einsatz allgemein gültigen Festlegungen des Syndrombegriffes sowie der damit zusammenhängenden Terminologie wie Assoziation, Sequenz, etc. vorhanden sind. Diese komplexe Semantik wirkt um so erschwerender, da die Anzahl der Krankheiten, Syndrome und Therapien in Bereichen von zigtausend liegt.

Große Datenmenge

Die große Datenmenge rührt einerseits von den sogenannten medizinischen Stammdaten (Patienten, Anatomie, Syndromologie, Methoden, ...), andererseits von den Bewegungsdaten (Untersuchung, Befunde, Diagnose, ...) her. Hinzu kommen noch Stand- und Bewegtbilddaten, so daß mit Datenmengen in Bereichen von Giga- und Terabytes zu rechnen ist. Mit den in naher Zukunft verfügbaren Speichermedien und der neuen Systemsoftware ist jedoch zu erwarten, daß die Datenmenge keine ernsthaften Probleme darstellen wird.

Heterogene Umgebung

Ebenfalls erschwerend kommt hinzu, daß diese Daten auf verschiedenen Subsystemen in verschiedenen Formaten verteilt sind. Die Systemumgebung ist in der Regel äußerst heterogen. Die Plattformen für die Serverseite reichen von Novell Netware, Apple Server, über Windows NT und Unix, bis DEC und Mainframe. Die Clientseite ist vertreten durch DOS-PC, Windows-PC in verschiedensten Ausführungen, Apple Macintosh ebenfalls in unterschiedlichsten Ausführungen, Next Step, Sun Microsystems, Silicon Graphics u. v. m.

1.1.4 Rechtlicher Aspekt

Bei der Konzeption und Realisierung eines solchen Systems spielt der rechtliche Aspekt eine nicht zu vernachlässigende Rolle. Neben dem an erster Stelle zu nennenden Datenschutz sind jedoch weitere Punkte zu berücksichtigen, die erst bei umfangreicher Kommunikation in Erscheinung treten. Es handelt sich z. B. um interne Administration, Datenaustausch mit externen Institutionen und die Berücksichtigung der sich ständig ändernden Gesetzeslage.

Datenschutz

Der Datenschutz dient im allgemeinen der Abwehr unerlaubter Zugriffe auf schutzwürdige Daten (Der Senat von Berlin [100]). Hier wird der Schutzaspekt von personenbezogenen Daten näher betrachtet. Der Angriff auf den Datenschutz - ob beabsichtigt oder nicht, ist zunächst unerheblich - kann an verschiedenen Stellen erfolgen. Zwei wichtige mögliche Schwachstellen liegen in der Anwendungs- und Kommunikationssoftware. Bei der Anwendungssoftware fehlt häufig ein Schutz vor Zugriffen Unbefugter gänzlich. Ist eine Anwendungssoftware von Stand-alone-Natur, also nicht kommunikationsfähig, wäre ausreichender Schutz durch einen gut gewählten abschließbaren Installationsort gegeben. Wird die Software dagegen im Netz betrieben oder gar mit Import-/Exportschnittstellen versehen, so bieten die Server und die Kommunikationsstrecke gute Angriffsflächen für unerlaubte Datenzugriffe. Neben dem altbewährten Paßwortmechanismus muß eine sichere Verschlüsselung der Daten zwecks Datenaustausch in allen Subsystemen eingebaut sein.

Interne Administration

Neben dem gesetzlichen Datenschutz sind weitere Schutzmechanismen notwendig, wie beispielsweise der Schutz der wissenschaftlichen, noch nicht veröffentlichten Daten oder der Schutz von noch nicht freigegebenen Befunden. Die meisten Subsysteme bieten hier bis jetzt überhaupt keinen Schutz an.

In einem Klinikum wie der Charité erfolgt die Speicherung und Verarbeitung medizinischer Daten vor dem Hintergrund der Krankenversorgung, Forschung und Lehre. Man kann für diese drei Aufgaben keine getrennten Netzwerke und Server installieren und betreiben. Die Trennung von Daten verschiedener Aspekte muß daher mit Hilfe intelligenter Zugriffsoftware und ggf. -hardware erfolgen.

Externer Datenaustausch

Im Prinzip unterliegt der externe Datenaustausch allen gesetzlichen Bestimmungen. Sie müssen im allgemeinen sogar noch schärfer ausgelegt werden, da man den Datenfluß nicht mehr verfolgen kann, wenn die Daten einmal auf den Weg geschickt werden. Hier liegt jedoch ein Dilemma. Da eine Klinik innerhalb des Klinikums in der Regel drei Aufgaben hat - Krankenversorgung, Forschung und Lehre -, gestaltet sich der Datenaustausch mit externen Einrichtungen, insbesondere mit externen Forschungseinrichtungen, äußerst kompliziert. Der Schutz der personenbezogenen Daten verlangt nach besonderen Schutzmechanismen wie einem ordentlich parametrisierten *Firewall*. Müssen einmal Daten innerhalb eines solchen *Firewalls* nach extern weitergegeben werden, so muß in der Regel für jede Verbindung eine gesonderte Genehmigung eingeholt, die Verbindung kontrolliert und ggf. protokolliert werden. Ein ungehinderter Datenaustausch mit anderen Forschungseinrichtungen ist somit nicht mehr möglich.

Gesetzesänderungen

Eine so offene medizinische Kommunikationsstruktur unterliegt naturgemäß Änderungen der hierfür in Frage kommenden Gesetze. Beispiele dafür sind Daten, die an die abrechnenden Kassen weitergeleitet werden, meldepflichtige Krankheiten/Epidemien und gesetzlich vorgeschriebene Statistiken. Es ist zu erwarten, daß mit zunehmender Automatisierung und zunehmender Regulierung seitens des Gesetzgebers der Umfang und die Form der Weitergabe der medizinischen Daten häufigeren Änderungen unterliegen. Ein Konzept muß an dieser Stelle den Aufwand der Softwareänderungen aufgrund einer Gesetzesänderung minimieren.

1.1.5 Ableitung aus der Problemanalyse

All die bereits beschriebenen Probleme führen dazu, daß sowohl Anwender die Anwendungen als auch die Anwendungen untereinander sich nicht verstehen. Dies bedeutet, daß eine interdisziplinäre Betrachtungsweise der gespeicherten medizinischen Informationen nicht ohne weiteres möglich ist. Eine interdisziplinäre Interpretation medizinischer Informationen stellt jedoch eine der wichtigsten Voraussetzungen für eine verbesserte Diagnoseunterstützung dar. Diese Probleme lassen sich jedoch nur dadurch lösen, daß eine Kommunikationsstruktur für die Kopplung der verschiedenen Subsysteme der medizinischen Disziplinen etabliert wird, die

- den Austausch (prinzipiell aller) patientenbezogener Daten ermöglicht,
- die Vollständigkeit und Integrität ausgetauschter Informationen inklusive semantischer Inhalte garantiert,
- durch klare syntaktische Strukturierung die Visualisierung und Interpretierbarkeit verbessert,

- die Integrität der Originaldaten erhält, sowie die allgemeine Datenredundanz minimiert und
- fallbezogene Recherchen erlaubt.

Der Schwerpunkt der Lösung darf somit nicht darin liegen, die vorhandenen Subsysteme zu verändern. Die Hauptstrategie muß sein, die z. Z. im Einsatz befindlichen Subsysteme zu ergänzen, ohne Änderungen an ihnen vornehmen zu müssen. Dies ist in vielen Fällen auch nicht möglich, da die Software-Hersteller nur äußerst ungern bzw. nur für teures Geld ihre Software verändern. Es ist für sie immer einfacher, die Software durch bestimmte Module, z. B. Import-/Export-Module, zu ergänzen, wobei ihre generelle Daten- und Programmstruktur nicht oder nur unwesentlich verändert werden muß. Mit anderen Worten, man verpacke diese Subsysteme mit einer Schicht Software, die dem Subsystem die Kommunikationsfunktionalität gemäß den gestellten Anforderungen verleiht. Im Kern arbeitet jedoch noch die ursprüngliche Softwaremaschine. Diese Technik wird als „*Verpackung*“ (Wrapping) bezeichnet.

Läßt sich das Subsystem nicht erweitern, so müßte man mit der Verpackungsoftware direkt auf die Daten zugreifen. Hier besteht jedoch das Problem, daß nicht jeder Softwarehersteller die Datenbeschreibung bereitwillig herausgibt.

1.2 Stand der technologischen Forschung

In diesem Abschnitt wird eine Übersicht über die aktuellen Entwicklungen auf dem Gebiet der Software-Technologie gegeben, die Einfluß auf das Konzept haben bzw. haben können. Der Schwerpunkt wird auf die softwaretechnologische Entwicklung des Datenaustausches und der medizinisch-informationstechnischen Standardisierung gelegt. Die netzwerktechnische Seite wird als gegeben angenommen, da in diesem Bereich die technische Entwicklung bereits so weit fortgeschritten ist, daß die Realisierung von Netzwerken - auch Mutliprotokollnetze - technisch keine Schwierigkeit bereitet. Man denke hierbei z. B. an Fast-Ethernet und ATM. An Ethernet im Gigabit-Bereich wird bereits experimentiert.

Im Bereich der Software-Technologie werden im wesentlichen folgende Konzepte untersucht:

- die Objektorientierung (OO),
- die firmenspezifischen Realisierungen der Objektorientierung,
- der Remote Procedure Call (RPC),
- das Hypertext-Konzept im Internet mit HTTP und HTML/XML und
- das Konzept autonomer und intelligenter Software.

Aus der medizinisch-informationstechnischen Standardisierung betrachten wir stellvertretend zwei Standards:

- HL7 Health Level Seven und
- DICOM Digital Imaging and Communications in Medicine.

Außerdem wird das SNOMED-Konzept des College of American Pathologists (Systematized Nomenclature of Human and Veterinary Medicine) näher betrachtet, das die in der Medizin verwendeten Begriffe in eine Struktur mit einheitlichen Bezeichnungen (Nomenklatur) einordnet.

An einer semantischen Kommunikation zwischen den medizinischen Subsystemen arbeitet beispielsweise z. Z. die Decision System Group am Brigham and Womens Hospital, Boston, deren Konzept mit dem Namen *Semantic Network* weiter unten erläutert wird.

1.2.1 Die objekt-orientierte Vorgehensweise (OO)

Für die vorliegende Arbeit ist das objekt-orientierte Konzept (OO) insofern interessant, als bei einem Datenaustausch gemäß dieses Konzepts nicht lediglich die „passiven“ Daten ausgetauscht werden, sondern auch sogenannte „aktive“ Objekte. Genau hier liegt der Vorteil dieses Konzepts für den Datenaustausch. Angenommen, wir wollen die Anamnese aus der Genetik in ein Subsystem der pränatalen Diagnostik holen und sie als Bestandteil eines Befundes betrachten. Da das Subsystem der pränatalen Diagnostik die interne Struktur der Genetik nicht kennt, was gewöhnlich der Fall ist, kann es mit den Anamnesedaten in der Regel wenig anfangen, wenn der Datenaustausch nicht objekt-orientiert erfolgt. Ist dagegen ein objekt-orientierter Datenaustausch realisiert, so wird ein Objekt des Typs Anamnese in den Befund integriert. Das Objekt erhält beispielsweise ein Rechteck oder einen beliebig geformten Bereich für die eigene Darstellung zugewiesen. Die Darstellung wird vom Objekt „Anamnese“ an Hand der auf der aktuellen Maschine zur Verfügung stehenden grafischen Ressourcen selbständig durchgeführt. Das Subsystem der pränatalen Diagnostik braucht sich nicht um die Darstellung zu kümmern. Selbst wenn der Befund auf einem externen Gerät wie einem Drucker ausgegeben werden soll, wird die Anamnese vom Objekt ausgedruckt. Das gleiche gilt für den weiteren Export des Befundes. In diesem Fall importiert das Subsystem der pränatalen Diagnostik nicht nur die Daten der Anamnese mit deren Darstellung, sondern alle Funktionalität, die es benötigt.

Im Prinzip beinhaltet das OO-Konzept folgende Merkmale:

1. Das Objekt modelliert einen Sachverhalt aus der realen Welt.
2. Die Eigenschaften der Objekte sind gekapselt und für die Umgebung des Objekts nicht direkt zugänglich.
3. Es gibt Daten- und Operationskapseln. Dies sind passive bzw. aktive Eigenschaften des Objekts.
4. Die Operationen in Objekten werden durch das Eintreffen von Nachrichten ausgelöst.

Beim Einsatz von OO-Komponenten muß jedoch zwischen den verschiedenen Varianten unterschieden werden. Sie lassen sich in folgende Kategorien unterteilen:

- Programmiersprachen,
- Datenbank-Management-Systeme und
- Kommunikationsprotokolle

Die Varianten der Programmiersprachen reichen von rein objekt-orientiert über objekt-orientiert-erweitert, wertorientiert bis logik/wissens/regel-orientiert. Für die Kommunikation kommen auf Grund der oben beschriebenen Eigenschaften von OO die reinen OO-Konzepte in Frage. Zu den OO-Programmiersprachen zählen:

- **SmallTalk:** Ausgangspunkt für SmallTalk ist das FLEX-System von A. Kay [116], [37], [57]. Weiterentwicklungen wurden von XEROX PARC (Xerox Palo Alto Research Center) betrieben. SmallTalk ist nicht nur eine Programmiersprache, sondern vielmehr eine Programmierumgebung [38], [96]. Jede Systemkomponente stellt ein Objekt dar. In SmallTalk-Sprache wird ein Objekt durch eine Klasse implementiert, wobei es zwei SmallTalk-Klassen gibt, die gleichberechtigt sind: System- und Benutzer-Klassen [38].

SmallTalk ist somit ein sehr strenges OO-Konzept. Trotzdem ist der Verbreitungsgrad von SmallTalk gering. SmallTalk existiert neuerdings auch für graphische Bedienoberflächen wie MS Windows in verschiedenen Varianten (95, NT).

SmallTalk ist wegen der Nichtkompatibilität der binären Codes nicht für die Implementierung des semantischen Kommunikationsstruktur-Konzepts geeignet. Die Eigenschaften von SmallTalk lassen sich jedoch als Vorlagen für den Systementwurf verwenden.

- **C++:** ursprünglich von Bjarne Stroustrup entwickelt [79], ist C++ jetzt von ANSI als Draft veröffentlicht [98]. Bei C++ handelt es sich um ein Superset von Standard-C. Diese Tatsache hilft, alte C-Programme in ein C++-Projekt ohne größere Umstellung zu übernehmen, was mit ein Grund für den Erfolg von C++ ist. C++ wird heutzutage von fast allen Compiler-Herstellern als integrierte Entwicklungsumgebung angeboten. Darunter ist eine Großzahl auf dem PC verfügbar (Borland, DEC, IBM, MetaWare, Microsoft, Openworks, Watcom, Zortech, ...). Ein Objekt wird in C++ durch eine Klasse implementiert. C++ enthält eine Vielzahl von sehr leistungsfähigen Konzepten wie Vererbung, auch Mehrfachvererbung (multiple inheritance), dynamische Bindung von Funktionen (dynamic binding), Polymorphie, Funktionsüberladen (overriding), Ausnahmebehandlung (exception) und Schablone (template). Damit lassen sich praktisch beliebige semantische Modelle abbilden. Ein Beispiel hierfür ist die Bibliothek MFC Microsoft Foundation Classes von Microsoft. Bis auf einige wenige Klassen entstammen alle darin enthaltenen Klassen der Basisklasse CObject [65].

Trotz des mittlerweile sehr hohen Verbreitungsgrads ist C++ nicht binär-kompatibel und lässt sich daher analog zu SmallTalk nicht ohne weiteres für die Implementierung der Kommunikationsstruktur verwenden.

- **Java:** ursprünglich für die Programmierung der Steuerung elektronischer Geräte wie Video-Recorder, Mikrowellenherde, etc. erfunden, erlebt Java mit der zunehmenden Nutzung des Internet eine Wiedergeburt [119]. Das Sprachkonzept von Java basiert auf dem Konzept von C++, wobei allerdings neue Eigenschaften definiert werden [83]:

A. Die erste Eigenschaft ist der Wegfall von Zeigern (pointers). Dies ist notwendig, um die Stabilität der Java-Programme zu steigern, was für ein Hostsystem ein Mehr an Sicherheit bedeutet. Diese Eigenschaft wirkt sich zwar positiv bezüglich der Laufstabilität der Software auf den Einsatz der semantischen Kommunikationsstruktur aus, ist jedoch nicht primär notwendig.

- B.** Die zweite Eigenschaft betrifft die Kapselung von Systemfunktionen. Ein Java-Programm kann nicht direkt auf die Systemfunktionen zugreifen. Hierfür bietet das Java-Laufzeitsystem einen Satz von Systemklassen. Diese Eigenschaft wirkt sich ebenfalls - mindestens theoretisch - positiv auf den Einsatz der semantischen Kommunikationsstruktur aus, ist jedoch nicht primär notwendig. In der Praxis verhindert diese Eigenschaft sehr oft die Realisierung von Anwendungen, deren Bedienoberfläche einem Standardverhalten auf dem betroffenen Betriebssystem gehorcht.
- C.** Eine dritte Eigenschaft ist die Plattformunabhängigkeit. Diese Eigenschaft rührt daher, daß Java-Programme von sogenannten Laufzeitsystemen ausgeführt werden, die auf der Zielmaschine installiert sind. Ist auf der Zielmaschine kein Laufzeitsystem vorhanden, so können Java-Programme nicht ausgeführt werden. Die Plattformunabhängigkeit reicht nicht aus, um ein beliebiges Java-Programm auf einer beliebigen Zielmaschine ausführen zu lassen. Um diese Eigenschaft zu verdeutlichen, wird ein Vergleich mit C++ im folgenden konstruiert. Wenn man für jede Zielmaschine zusätzlich zu dem C/C++-Laufzeitsystem einen C/C++-Compiler und -Linker installiert, die in der Lage sind, das C/C++-Programm dynamisch zu übersetzen und zu linken, so erreicht man die gleiche Plattformunabhängigkeit wie bei Java. Da für Java bereits eine Vielzahl von Laufzeitsystemen existiert, kann von einer (relativen) Plattformunabhängigkeit von Java ausgegangen werden. Diese Eigenschaft prädestiniert Java für die Implementierung von plattformunabhängigen Kommunikationsstrukturen. Sie ist auch eine der wichtigsten Voraussetzungen dafür.
- D.** Die vierte Eigenschaft betrifft die Struktur von Java-Programmen. Im Gegensatz zu einem C++-Programm wird eine Anwendung in Java durch eine Vielzahl von kleineren Anwendungsprogrammen implementiert (Applets). Sie werden in der Regel auf einem Server abgelegt und bei Bedarf auf den Arbeitsplatz geschickt und ausgeführt. Diese Eigenschaft stellt somit sicher, daß immer eine aktuelle Version eines Applets verwendet wird. In Java existieren demnach keine verschiedenen Versionen eines Applets. Verschiedene Versionen werden wie verschiedene Programme behandelt. Dieses Merkmal birgt jedoch ein Problem in sich. Durch das ständige Laden von Applets steigt die Netzbelastung drastisch. Dieser Nachteil läßt sich reduzieren, in dem der Arbeitsplatz die Applets auf die lokale Festplatte zwischenspeichert. Erst wenn eine Version auf dem zentralen Server neuer als die lokale ist, wird die neue Version geladen.

Die o. g. Eigenschaften von Java sind ein Grund dafür, daß Java oft als C++- bezeichnet wird.

Es existiert eine Reihe von Produkten der Klasse Datenbank-Management-System, die ebenfalls mehr oder weniger objekt-orientiert arbeiten (OODBMS). Da diese Produkte nicht die primäre Aufgabe haben, die Kommunikation zwischen den Anwendungen zu regeln, sondern die Daten zu verwalten, werden sie hier nicht weiter behandelt.

Zu dem OO-Konzept zählt das Component-Object-Model der Firma Microsoft mit den darauf aufbauenden Komponenten OLE/OCX/ActiveX, die im folgenden näher untersucht werden.

Im medizinischen Bereich wurde eine Reihe von Befundsystemen entwickelt, die auf dem objekt-orientierten Konzept basierten. Dore, L., et al. [27], verwendeten ein solches Modell [119]. Allerdings wurde auf Freitexte im Befund nicht verzichtet.

1.2.2 Firmenspezifische Realisierungen

Das OO-Konzept von Microsoft

Das Component-Object-Model (COM) wurde von Microsoft entwickelt [88], [57], [62] (s. *Abbildung 1-2 Das COM-Modell der Firma Microsoft*). Es wird wegen der Verbreitung stellvertretend für die Basiskonzepte anderer Firmen bzw. Institutionen untersucht.

Das COM definiert eine Reihe von Basiskonzepten und ist eine plattformübergreifende Spezifikation von Objekten, die in heterogenen Netzwerken eingesetzt werden können [28]. Das Anwendungsprogramm benötigt zur eigenen Nutzung eine Importdatei, die sogenannte Type Library, in der die Objektmethoden definiert sind. Diese Methoden werden Interfaces genannt. Deren Definition erfolgt in einer plattformübergreifenden Sprache, der Interface Description Language (IDL). Diese Sprache wurde ursprünglich von der Arbeitsgruppe „Distributed Computing Environment (DCE)“ der Open System Foundation (OSF) definiert. Sie wurde primär eingesetzt, um die Schnittstelle von Remote Procedure Call (RPC) zu definieren [26]. Microsoft erweiterte für COM diese Schnittstelle um die Möglichkeit, OLE- bzw. ActiveX-Funktionalitäten zu beschreiben und nannte sie MIDL.

Um ein COM-Objekt in eine C/C++-Anwendung zu integrieren, muß der Programmierer den MIDL-Compiler verwenden, um aus der MIDL-Datei die nötigen C/C++-abhängigen Definitionsdateien (header files für C/C++ und zusätzliche *.cpp files bei C++) zu erzeugen [63]. Für die Anbindung eines COM-Objekts an andere Programmiersprachen wie Object Cobol, Perl, etc. kann der entsprechende MIDL-Compiler diese notwendigen Dateien generieren.

Abbildung 1-2 zeigt die Arbeitsteilung zwischen Objekt-Programmierern, Anwendungsprogrammierern und Dienstleistung des Betriebssystems. Die grau hinterlegten Objekte werden vom Objekt-Programmierer zur Verfügung gestellt.

Weder der Programmier eines COM-Objekts noch der Anwendungsprogrammierer müssen sich darum kümmern, wie die Verbindung zustandekommt. Hierfür sorgt einerseits das COM selbst. Das COM generiert hierzu an Hand der MIDL die Platzhalterfunktionen (Proxy und Stub), die für die Umsetzung der Aufrufe sorgen. Diese Platzhalterfunktionen sorgen ferner dafür, daß die Verbindungen über Netzwerkgrenzen hinweg mit Hilfe von RPC bzw. LRPC zustandekommen. Für alle Programmierer sind diese Funktionen völlig transparent.

Mit der Version 4.0 des Betriebssystems Windows NT führte Microsoft das sogenannte Distributed COM (DCOM) ein. Für den Programm-Source-Code ändert sich im Prinzip nichts. DCOM wurde notwendig, um Werkzeuge für den Einsatz der COM-Objekte im Netzwerk und im neuen Threading-Modell von Windows NT 4.0 bereitzustellen. Hierbei sind die verschiedenen Threading-Modelle, die Sicherheit im Netzwerk und die „Remote Instantiation“ zu nennen [12].

Zusammengefaßt bietet COM/DCOM dem Anwender folgende Merkmale:

- binäre Kompatibilität zwischen den verschiedenen Applikationen, auf verschiedenen Plattformen,
- Unabhängigkeit von Programmiersprachen,
- gemeinsam benutzbarer Speicher und
- dynamisches Laden (load on call)

COM ist eine Basis für andere Dienste wie OLE Controls, OLE Automation, etc. (s. *Abbildung 1-3*). Das Basisobjekt von COM wird COM-Objekt genannt.

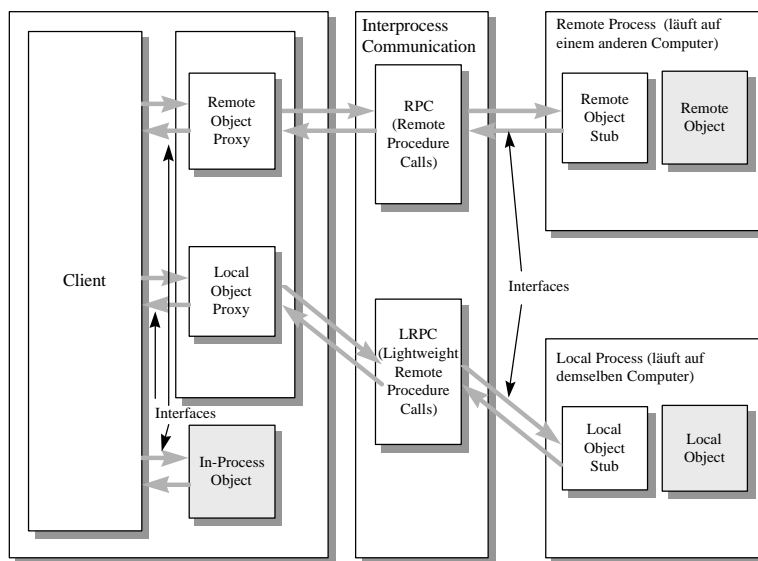


Abbildung 1-2 Das COM-Modell der Firma Microsoft

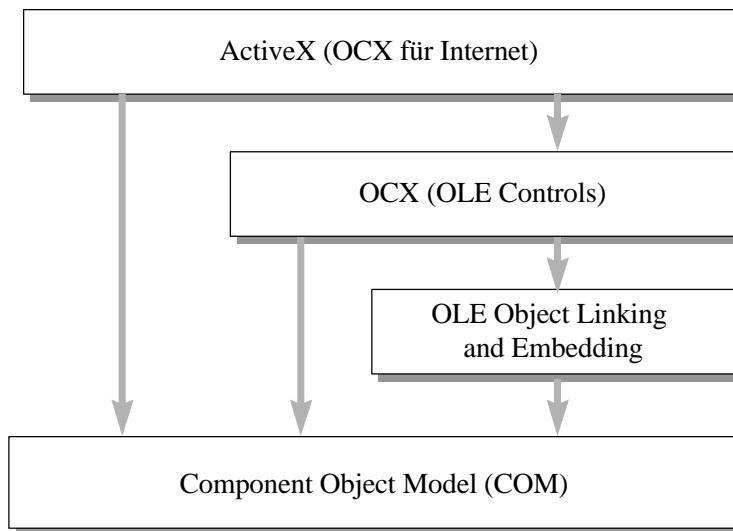


Abbildung 1-3 Das Microsoft-COM als Basis für weitere Konzepte [11]

Das OLE-Konzept von Microsoft basiert auf der COM-Spezifikation. Ein OLE-Programm kann entweder Client oder Server oder beides sein. Microsoft nennt das OLE-Konzept nicht object-oriented, sondern object-enabled [62]. Das OLE-Konzept beinhaltet folgende Merkmale (mehr Erläuterung siehe [11]):

- OLE Component Object Model,
- OLE Automation,
- OLE Controls,
- OLE Drag and Drop,
- OLE Component Management und
- OLE Documents.

Mit der Unterstützung der Microsoft-Software für den Internet-Einsatz hat Microsoft die Nachfolgeversion von OLE eingeführt. Diese ActiveX-Technologie ist nötig, um die OLE-Funktionalität Internet-fähig zu machen. Erstens muß der Umfang der zu übertragenden Codes und Daten gering sein. Zweitens müssen die Objekte, ob Controls oder Dokumente, in einem Browser einsetzbar sein (Johns, P. [52]). Dort wird ein Vergleich zwischen OLE/ActiveX-Control und Java-Applet vorgenommen. Diese ActiveX-Technologie stellt eine direkte Konkurrenz zum Java-Konzept dar (s. unten).

Das OO-Konzept von IBM (SOM/DSOM)

SOM/DSOM ist die Abkürzung von *System Object Model/Distributed System Object Model*. SOM und DSOM sind Objektmodelle, die von IBM für IBM OS/2® und IBM AIX definiert sind [117]. Diese Objekte bieten eingeschränkte Möglichkeiten, denn einige wichtige Eigenschaften fehlen. SOM selbst erlaubt Objekten nicht, miteinander zu kommunizieren. Sie können nicht von mehreren Anwendungen gemeinsam benutzt werden (share), auch zwischen Prozessen innerhalb einer Maschine nicht. Um dies zu erlauben, wird DSOM definiert. Im Gegensatz zu OLE behandeln SOM

und DSOM lokale und verteilte Objekte unterschiedlich. Für OLE ist es unerheblich, wo sich die Objekte befinden. Ferner versuchen SOM und DSOM, die Grenze zwischen OO-Programmierung und Objektmodell zu verwischen (siehe oben).

SOM und DSOM erlauben unkontrollierbare *implementation inheritance*: Objekte können Source-Code-Implementationen durch die *class hierarchies* vererben. Obwohl dies die Programmierung von SOM und DSOM Objekten schneller macht, steigert sich die Abhängigkeit zwischen den Objekten. Diese Abhängigkeit verhindert das automatische Upgrading. Weitere fehlende Merkmale sind Compound Document Support, Object Version Control, globally unique object identifiers und ein Mechanismus gegen die Dead-Lock-Situation.

SOM und DSOM sind außerdem nicht so weit verbreitet, daß sie sich für eine semantische Kommunikationsstruktur verwenden lassen. Neben SOM/DSOM entwickelt IBM ein neues Konzept mit dem Namen Component Broker, um traditionelle Anwendungen, die meistens noch auf Mainframes laufen, in das Internet/Intranet einzubinden. Das Konzept Component Broker weist Parallelen zu DCOM von Microsoft auf [22].

Das OO-Konzept von Apple (OpenDoc)

Bei OpenDoc handelt es sich um eine Spezifikation für zusammengesetzte Dokumente, die verschiedene Objekt-Technologien vereinigt: die Basis-OpenDoc-Architektur, das Bento-Dateisystem und die Open-Scripting-Architektur sowie das IBM SOM [93]. Die Entwicklung von OpenDoc wurde von einer Reihe von Computer-Herstellern übernommen, darunter Apple, IBM, WordPerfect und Novell. Bis heute sind wenig Anwendungen verfügbar, die OpenDoc unterstützen. Daher eignet sich OpenDoc z. Z. nicht für die Implementierung einer semantischen Kommunikationsstruktur. Neuerdings hat Apple die Entwicklung von OpenDoc eingestellt. Daher wird OpenDoc für die vorliegende Arbeit voraussichtlich keine entscheidende Rolle mehr spielen.

Das OO-Konzept CORBA

CORBA ist die Abkürzung für Common Object Request Broker Architecture (Brando, T. J. [94]).

CORBA ist eine Spezifikation der Object Management Group OMG. Bei dieser Gruppe handelt es sich um ein Konsortium, bestehend aus Softwareherstellern und -anwendern. Z. Z. umfaßt die OMG etwa 600 Mitglieder, darunter auch die Firma Microsoft, die das Konkurrenzkonzept COM/OLE entwickelt hat. CORBA ist ein Versuch, eine gemeinsame Basis für verschiedene Objektmodelle zu definieren. Die Funktionalität von CORBA ist begrenzt (kleinster gemeinsamer Nenner). Daher kann keine Garantie für die Interoperabilität zwischen verschiedenen Produkten erwartet werden. Laut Microsoft deckt CORBA nur etwa 15 % eines vollständigen Objektmodells ab [62]. Daher eignet sich CORBA z. Z. ebenfalls nicht für die Implementierung einer semantischen Kommunikationsstruktur. Nichtsdestotrotz wird CORBA von HL7 und ähnlichen Einrichtungen favorisiert. Es bestehen Bestrebungen von OMG und Microsoft, CORBA und COM/DCOM kompatibel zu machen. Das Konzept sollte daher CORBA als möglichen Bestandteil berücksichtigen.

1.2.3 Vergleich COM/OLE/ActiveX und Java

Das Java-Konzept von Sun ist bereits oben beschrieben worden (siehe dort). Hier wird ein Vergleich zwischen OLE- bzw. ActiveX-Controls von Microsoft und den Java-Applets von Sun angestellt.

Mit der Freigabe von Windows NT V4.0 wird DCOM ebenfalls freigegeben. Somit ist ein ActiveX-Control auch über ein Netzwerk einsetzbar. Man benötigt hierzu DCOM, das sowohl auf dem Server als auch auf dem Client installiert sein muß. Diese Netzwerkeigenschaft hat von Hause aus ein Java-Applet. Da Java sich an C++ anlehnt, kann Java die Vererbung (Inheritance) ohne weiteres anbieten (built-in-Eigenschaft). Durch die Embedding-Möglichkeit enthält ein ActiveX-Control auch die Vererbung, wenngleich eine Einsatzmöglichkeit anders ist als bei Java.

Im folgenden werden drei technische Eigenschaften beschrieben, die zumindest theoretisch eindeutig für den Einsatz von Java im Netz sprechen [113].

Erstens ist Java völlig plattformunabhängig. Dagegen ist ein ActiveX-Objekt immer an Windows gebunden.

Zweitens läuft ein Java-Applet in der geschützten Java-Umgebung. Ein ActiveX-Control hat die Möglichkeit, direkt auf die Hardware und auf das Betriebssystem zuzugreifen. Der Sicherheitsaspekt bei ActiveX kann daher als mangelhaft bewertet werden. Ein möglicher Einsatz von ActiveX macht gesonderte Schutzmaßnahmen für die Client-Maschine notwendig [111].

Drittens kann ein Java-Applet von einem Server aus nicht nur zu einem Client geschickt werden, was in der Regel der Fall sein wird. Ein Java-Applet kann auch an einen Server geschickt werden, um bestimmte Aufgaben selbständig auszuführen. In diesem Fall spricht man von einem Internet-Agenten bzw. einem Internet-Roboter. Ein ActiveX-Control kann z. Z. immer nur von einem Server an einen Client geschickt werden.

Die o. g. Ausführung gibt dem Java-Konzept einen deutlichen technischen Vorteil (Flynn, J., et al. [107]). Ob es sich in der Praxis durchsetzt, muß sich noch zeigen. Einige Tatsachen sprechen für das ActiveX-Konzept von Microsoft. Erstens hat Microsoft eine immense Werbekampagne für die eigene Internet-Technologie gestartet. Hierunter fallen auch die kostenlose Verfügbarkeit der Technologie, angefangen bei den Clients (Explorer) über die Server-Software (Internet Information Server), sowie die ActiveX-Controls, bis hin zum damit verbundenen Support wie Spezifikation, Testsoftware, etc. Zweitens bietet Microsoft eine Menge Werkzeuge wie Compiler, Netzwerk-, Datenbank- und Mail-Server und die Internet Domain Name Server (DNS) zu äußerst günstigen Konditionen an. Alle benötigte Software wird aus einer Hand angeboten. Es ist zu erwarten, daß die Komponenten gut miteinander kooperieren. Diese Aspekte dürfen bei der Entscheidung für oder gegen Java nicht außer acht gelassen werden.

1.2.4 Das Netzwerkprotokoll RPC

RPC (remote procedure call) ist ein Mechanismus, mit dem Anwendungen sich gegenseitig aufrufen können, auch über die Grenze eines Prozesses, eines Rechners und eines LAN hinweg.

RPC ist leicht zu benutzen und populär, da der Aufrufmechanismus wie ein lokaler Aufruf funktioniert. Er arbeitet für den Programmierer transparent. Er braucht sich nicht um das Netzwerk zu kümmern. Microsoft verwendet eine DCE-kompatible RPC für alle neueren Windows-Varianten (95, NT). Somit ist dieser RPC in der Lage, „Calls“ in einer heterogenen Umgebung abzusetzen. DCE steht für Distributed Computing Environment. Es beinhaltet eine Reihe von Spezifikationen, darunter den RPC. Abbildung 1-4 zeigt den RPC-Mechanismus in Form einer Übersicht [26].

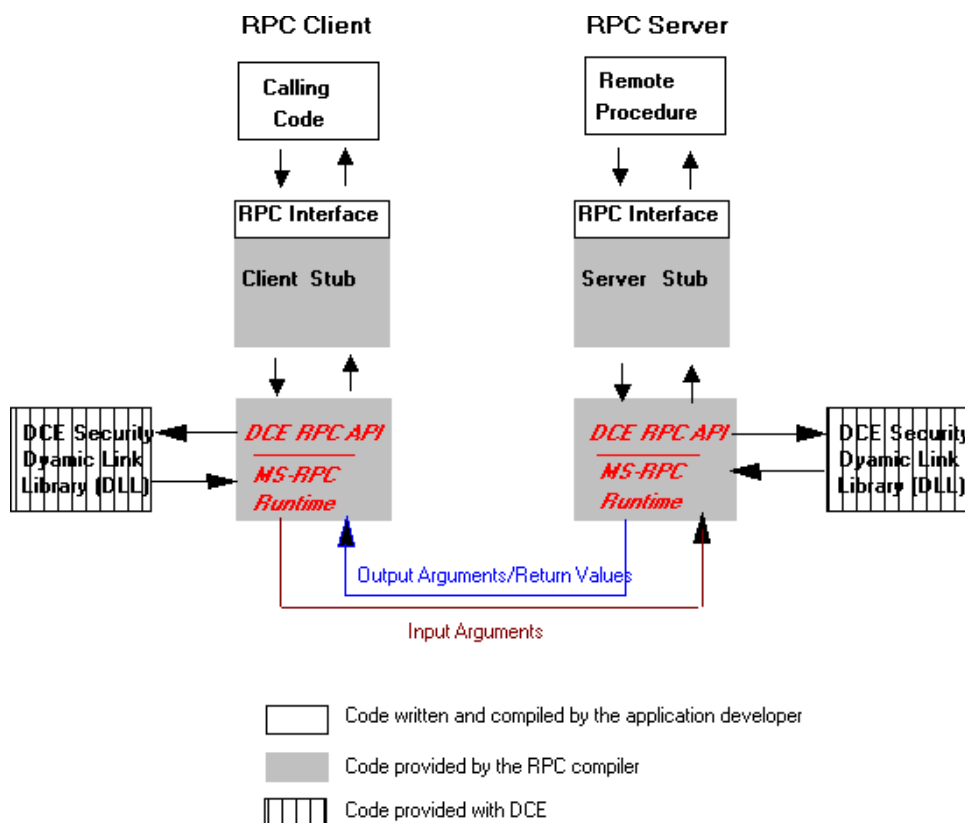


Abbildung 1-4 Übersicht über RPC-Mechanismus [26]

1.2.5 Das Netzwerkprotokoll HTML/XML

Das HTML-Protokoll ist von der SGML (Standard Generalized Markup Language) abgeleitet. SGML ist seit 1980 ISO-Norm. Sie definiert eine Dokument-Beschreibungs-Sprache (document description language). Diese Sprache ist dank ihrer Einfachheit sehr flexibel und leistungsfähig. Mit SGML lassen sich verschiedene Dokumenttypen definieren. Zu jedem Dokumenttyp existieren eine oder mehrere Dokument-Beschreibungs-Dateien. Sie werden an Hand einer sehr simplen Syntax fest-

gelegt. HTML ist ein Dokumenttyp von SGML. Dieser wird für die Internet-Browser eingesetzt.

HTML wurde ursprünglich in CERN entwickelt, um die Ergebnisse der Kernforschung in CERN rechnerunabhängig darzustellen. HTML steht für Hypertext Markup Language [95].

Die aktuelle freigegebene Version von HTML ist die Version 3.2 [137]. Sie hat jedoch für die Praxis keine Bedeutung mehr, da die zwei bedeutendsten Browser-Hersteller, NetScape und Microsoft, HTML in ihren Browsern erweitert haben. Dies betrifft im wesentlichen die Frames, die Tabellendarstellung und die Formulare.

In der Praxis läßt sich HTML als ein plattformunabhängiges Protokoll verwenden, um Information zu erfassen und darzustellen, da HTML-Dateien in reinem ASCII-Format, 7- bzw. 8bit codiert sind. Es ergibt sich somit kein Problem bei der Übertragung. Die Verarbeitung übernehmen HTML-Plug-Ins wie Java-Applets oder OLE-Controls (OCX/ActiveX). In einer HTML-Datei können Verweise auf eine weltweit gültige Adresse enthalten sein, die zusammen ein Netzwerk von unbegrenzter Komplexität bilden können. Diese Verweise können von verschiedenen Typen sein: eine weitere HTML-Datei, ein Sprungziel in einer HTML-Datei oder eine eingebettete Datei beliebigen Typs (Bilder, Audio, Video, 3D Animation, etc.). Der Browser führt diese eingebetteten Dateien aus, in dem er die sogenannten Plug-Ins bzw. Applets aufruft (Davidson, A. [99]).

Die Entwicklung bleibt hier jedoch nicht stehen. Die jetzige Version 3.x von HTML enthält vom Konsortium festgelegte Einträge, die vom Anwender nicht mehr modifiziert werden dürfen. In dem für die Festlegung von HTML zuständigen W3-Konsortium existieren bereits Vorschläge für die Erweiterung von HTML, so daß der Anwender dynamische Einträge (Entities) definieren kann. Dieses Vorhaben läuft unter dem Namen XML Extensible Markup Language (Cover, R. [97]). Die Version 1.0 ist im November 1996 zwecks öffentlicher Diskussion und Reviews bekannt gegeben worden. In XML wird nicht nur der Dokumenttyp, sondern auch das Verhalten von Softwareprogrammen, die die im Dokument enthaltenen Datentypen behandeln, festgelegt.

Die hier verwendete Strategie der standardisierten, flexiblen und minimalen Syntax kann als Vorbild für die Definition medizinischer Kontexte dienen, die medizinische Subsysteme beschreiben.

Zusammen mit Java bietet HTML/XML eine praktisch unbegrenzte Möglichkeit, plattformunabhängige semantische Kommunikationsstrukturen aufzubauen. Langfristig ist daher zu erwarten, daß HTML/XML die anderen Standards für den elektronischen Austausch von Dokumenten wie EDI/EDIFACT aus dem Feld medizinischer Netzwerke verdrängen wird. Vereinzelt finden sich jedoch nach wie vor darauf aufbauende medizinische Anwendungen wie es Branger, P., et al. für eine „Share-Care-Anwendung“ realisiert haben [10]. Detmer, W. M., stellte ein Modell für medizinische Informationssysteme auf der Basis von WWW-Server in Verbindung mit CGI auf [24].

1.2.6 Der HL7-Standard

Für den Austausch medizinischer Daten wurde die Schnittstelle HL7 (Health Level Seven) definiert [104], [42], [105]. Der Begriff „Level Seven“ verdeutlicht, daß sich der Standard ausschließlich mit der Ebene 7 des ISO-OSI befaßt. Das ISO-OSI-Modell (International Standardisation Organisation - Open System Interconnect) legt ein Modell mit sieben Ebenen fest. Die niedrigste Ebene regelt die Verkabelungsphysik des Netzes. Beispielsweise handelt es sich um eine Ethernet-, eine Arcnet- oder eine FDDI-Verkabelung. Es folgen dann die Verbindungs-, Transport-, Netzwerk-, Kommunikations-, Präsentations- und die Anwendungsebene. HL7 fokussiert auf die Anwendungsebene, macht daher keine Aussage über das zu verwendende Netzwerk. Es kann im Prinzip jedes Kommunikationsmittel eingesetzt werden. Eine serielle Verbindung von Rechner zu Rechner ist hier nicht ausgeschlossen. HL7 funktioniert nach dem Transaktionskonzept. Die Transaktionen werden in Form von HL7-Nachrichten verschickt bzw. empfangen. Die hierin enthaltenen Informationen sind im Prinzip lesbare Zeichen.

HL7 beinhaltet u. a. folgende Komponenten:

- Aufnahme, Verlegung, Entlassung (ADT Admission, Discharge and Transfer),
- Bestellung von Waren und Dienstleistungen,
- Laborergebnisse,
- Arzneimittel,
- Patientenverwaltung,
- Abrechnung und
- Krankenakte.

Die wichtigste davon ist die sogenannte ADT-Komponente. Diese Komponente hat die meisten Berührungspunkte mit der vorliegenden Arbeit. Hier sind bisher auch die meisten Inkonsistenzen zwischen den Subsystemen zu verzeichnen. Die ADT ist in der Regel ein verwaltungstechnischer Akt. Ein medizinisches Subsystem sollte eigentlich bei Verfügbarkeit eines zentralen Patientenverwaltungssystems keine eigene ADT-Funktion implementieren. Es sollte vielmehr alle ADT-Informationen über das zentrale System beziehen.

HL7 scheint der erste Standard in der medizinischen Versorgung zu sein, der weltweit Implementierungen erfährt. Hier hat der Implementator das erste Problem zu lösen. Der Standard berücksichtigt zwar andere Zeichensätze wie europäische oder japanische. Da er auf einem ASCII 7bit Code basiert, müssen ISO-Escape-Sequenzen verwendet werden, um zwischen ASCII 7bit, 8bit und Multibyte-Zeichensätzen umzuschalten. Da HL7 keine Aussagen über die Implementierung macht, existieren unterschiedliche Lösungen für die bei der Realisierung auftretenden Fragestellungen. Als Beispiel sei hier die Übertragung von binären Daten oder die Festlegung der maximalen Länge einer HL7-Nachricht genannt (Internet-Homepage der Duke University [105]).

Das Hauptproblem besteht jedoch darin, daß HL7 ferner keine Aussage über den obligatorischen Umfang einer Implementierung macht, sondern dies dem Anwender überläßt, in der Regel dem Krankenhaus und dem Implementator, in der Regel einem Softwarehaus. Diese müssen sich über den Umfang der auszutauschenden

Daten einigen. Eine andere Lösung wäre, daß die Software in der Handshake-Phase Verhandlungen über den Umfang des Datenaustausches führt. Die erste Lösung verhindert eine allgemeine Kompatibilität zwischen den Implementierungen. Die zweite Lösung macht die Software komplexer und daher fehleranfälliger und langsamer. HL7 ist somit nicht „*Plug and Play*“.

Ferner eignet sich HL7 z. Z. nicht für den Austausch von inhaltlichen medizinischen Daten wie Untersuchungsdaten, Befund und Diagnose. Hierzu bedient sich HL7 anderer Standards wie ICD, DICOM. ICD deckt nicht alle Möglichkeiten eines fein nuancierten Befundtextes ab. DICOM wird nicht von jedem HL7-fähigen System angeboten.

1.2.7 Der DICOM-Standard

DICOM (Digital Imaging and Communications in Medicine) ist ein Standard für den Austausch von Bildern, sowie den damit verbundenen Daten zwischen medizinischen Geräten. Ursprünglich wurde DICOM für den Datenaustausch im Bereich der Radiologie entwickelt (ACR American College of Radiology). Mittlerweile wird DICOM durch einen gemeinsamen Ausschuß von ACR und NEMA National Electrical Manufacturers Association bearbeitet [1].

Zwei der wichtigsten Datentypen in DICOM sind die „DICOM Information Objects“ und die „DICOM Service Classes“. „Information Objects“ sind beispielsweise Röntgen-, CT- und Ultraschallbilder, Patienten- und Untersuchungsinformationen, etc. „Service Classes“ sind Aktionen auf die „Information Objects“. Beispiele von „Service Classes“ sind Query/Retrieve und das Management von Patienten, Ergebnissen sowie Speichermedien [7].

Die „Information Objects“ werden syntaktisch und semantisch fest definiert. Jedes „Object“ erhält eine weltweit eindeutige Identifikation (ID). Die „Service Classes“ werden „Class“ für „Class“ ebenfalls syntaktisch und semantisch festgelegt. Eine „Service Class“ wird durch ein Paar von „Object IDs“ weltweit eindeutig identifiziert.

Der gedruckte Standard füllt in der Version 3.0 drei Aktenordner. Dies kann die Komplexität des Standards verdeutlichen. Es ist daher zu erwarten, daß nicht jedes Gerät den vollen Funktionsumfang des Standards zur Verfügung stellt. Die Verständigung zwischen den Geräten kann somit beeinträchtigt sein. Kimura, M., et al. veröffentlichte einen Standard für die 5,25 Zoll MO-Medien zwecks Übertragung von DICOM-Bildern als Zusatz zu DICOM 3.0 [56].

DICOM ist ein Standard für reine Bildspeicherung, -verarbeitung und reinen Bildaustausch. Es eignet sich daher nicht allein als allgemeiner Standard für den Austausch von medizinischen Daten.

1.2.8 Das SNOMED-Konzept des College of American Pathologists

SNOMED steht für Systematized Nomenclature of Human and Veterinary Medicine und wurde vom College of American Pathologists (CAP) entwickelt [20]. Der aktuelle Stand von SNOMED hat die Bezeichnung SNOMED International Version 3.3.

SNOMED unterteilt die in der Medizin verwendeten Begriffe in folgende Kategorien (es werden hier originale Bezeichnungen verwendet):

- Topography,
- Morphology,
- Function,
- Living Organisms,
- Chemicals, Drugs, and Biological Products,
- Physical Agents, Activities, and Forces,
- Occupations,
- Social Context,
- Diseases/Diagnoses,
- Procedures und
- General Linkages/Modifiers.

Mit diesen Unterteilungen sowie deren Kombinationen lassen sich fast alle in der Medizin verwendeten Begriffe erfassen, darunter auch Berufe, soziales Umfeld wie Sport, etc.

In Deutschland wird vielfach der sogenannte Wingert-SNOMED verwendet [89], [90], [91]. Diese Version ist keine bloße Übersetzung des amerikanischen Originals ins Deutsch. Sie weicht strukturell in vielen Punkten vom Original ab. Der Grund hierfür war, daß das amerikanische Original semantisch nicht fein genug spezifiziert wurde. Die nachfolgende Version des Originals hieß SNOMED II und wurde mehr in die Breite erweitert, in dem auch die Veterinär-Medizin hinzugenommen wurde. In die Tiefe - also mit mehr Semantik - wurde nichts grundlegendes mehr weiterentwickelt. Die aktuelle Version SNOMED International bringt wieder mehr Semantik. Z. Z. laufen Bestrebungen, die deutsche Wingert-SNOMED wieder mit der amerikanischen Version zusammenzuführen.

Nach Henry, S. B., et al. hat SNOMED das Potential zum Modellieren medizinischer Information [44], was Rothwell, D. J., et al. bestätigten [75]. Zelinger, J. et al. von Beth Israel Hospital, Harvard Medical School, Boston, MA, USA berichteten über ein Projekt, bei dem freie Befundtexte mit 80%iger Treffsicherheit auf SNOMED International abgebildet werden konnte [92].

Momentan laufen Arbeiten in allen Gremien von HL7, DICOM der NEMA und College of American Pathologists, um das Zusammenspiel aller drei Standards zu ermöglichen. DICOM soll in HL7 integriert werden (Keayes, R. [55]). Für DICOM wird eine sogenannte SNOMED-DICOM Microglossary definiert [20]. Ein Konzept für das Intranet muß somit diese Arbeiten beobachten und deren spätere Integration bereits jetzt vorsehen.

1.2.9 Das Semantic-Network

Um das ähnlich wie hier beschriebene Problem zu bewältigen, wurde vielfach das Konzept eines sogenannten Semantic-Network eingesetzt. Wir betrachten zunächst das von der DSG Decision System Group am Brigham and Womens Hospital von der Harvard Medical School in Boston entwickelte Semantic-Network mit dem Namen Thenetsys (Pattison-Gordon, E. [115]).

Das Network besteht aus *Nodes* und *Links*. Ein Node repräsentiert ein sogenanntes *Concept*. Beispielsweise „Femur“ ist ein *Concept*, das alle Femurs der Welt zusammenfaßt. Das linke Femur von einem bestimmten Patienten ist eine *Instance* (Pattison-Gordon, E. [115]).

Links zwischen Nodes stellen Beziehungen zwischen den Nodes dar. Links sind hier immer bidirektional. Z. B. das *Concept Femur* und das *Concept Bein* stehen in einer Relation mit dem Namen *ist der Ort*. Das *Bein* eines bestimmten Patienten *ist der Ort* des linken Femur. Die inverse Beziehung ist *hat den Ort*: Das linke Femur *hat den Ort* Bein (Pattison-Gordon, E. [115]).

Die DSG realisiert ein umfangreiches System, um das Konzept umzusetzen. Das zu realisierende System beinhaltet mehrere Funktionsbereiche wie Medizinisches Informationssystem, Tutorielsystem, Medikationssystem, Radiologisches Bildverwaltungssystem, etc.

Dieses Konzept basiert auf dem bekannten und verbreiteten Entity-Relationship-Modell. Die hier gewählte Vorgehensweise ergibt sich zwangsläufig aus den technischen Hilfsmitteln, die heutzutage zur Verfügung stehen: Multimedia, Internet/Intranet, relationale Datenbanken, etc. Es ist z. Z. nicht erkennbar, ob das Konzept auch eine Vereinfachung der Syntax und eine Reduzierung der Anzahl der Nodes und Links beinhaltet. Sollte dies nicht der Fall sein, so stellt es im Prinzip lediglich einen Satz von Werkzeugen zur Verfügung, die dem Anwender helfen, die eigenen, gespeicherten Informationen zu lesen und zu interpretieren. Der Zugriff auf fremde Datenbank wird hierdurch noch nicht möglich.

Weitere Anwendungen von „Semantic Network“ für verschiedene medizinische Fachbereiche fanden sich für Röntgendiagnostik [6], für die Modellierung der Inhalte von Visualisierungsboards (display) [15], für die Codierung von beruflichen Angaben im Gesundheitsbereich [41].

Eine in den USA oft verwendete Grundlage für Semantic Network ist das von der amerikanischen National Library of Medicine in Maryland, USA, entwickelte Unified Medical Language System (UMLS). UMLS enthält u. a. einen sogenannten Metathesaurus für Medizin und ist die Grundlage für Semantic-Network-Anwendungen wie natürlichsprachige Datenbankabfragen [17], für Hybridanwendung zwischen Expertensystem und Routine-Befundsystem [13], [82], für eine medizinische „Dictionary“ [53], für einen „Knowledge-Server“ [61], [77]. Rindfleisch, T. C., stellte fest, daß mit dem UMLS Metathesaurus ein 80%iges Extrahieren medizinischer Fakten aus Befundtexten möglich ist [74]. Weitere Projekte auf der Basis von Semantic Network wurden in vielen Veröffentlichungen beschrieben [48], [60], [71], [76] und [80].

1.2.10 Vollständige Eigenentwicklung

Das wohl größte bekannte Projekt mit dem Ziel, ein landesweites Gesundheitsnetzwerk zu installieren, wurde vom amerikanischen Department of Veterans Affairs (VA) durchgeführt.

Das VA hat 1991 mit einem Prototyp eines vollständigen Modells der medizinischen Informationsstruktur begonnen [19] und konnte 1996 das System mit dem Namen DHCP (Decentralized Hospital Computer Program) fertigstellen [121]. DHCP ist ein vollständiger Satz von in ANSI-Standard-M geschriebenen Computer-Programmen, die alle Funktionen eines medizinischen Netzwerks unterstützen sollen:

- Current Procedural Terminology (CPT) für die Codierung von ärztlichen Leistungen [122],
- Event Driven Reporting (EDR) für die Datenerfassung und -übertragung zwischen verschiedenen Computern [135],
- Health Level Seven (HL7), ein Standardprotokoll für den Austausch von Patientendaten zwischen verschiedenen Anwendungen,
- Kernel für die Standard-Software-Tools [136],
- Kernel ToolKit für die Entwicklung von Software [134],
- LetterMan für die Textverarbeitung [133],
- List Manager für das Handling von Listen [132], MailMan für die elektronische Korrespondenz [131],
- Messaging System für die Programmierung von Kommunikationsanwendungen [130],
- National Patch Module (NPM) für Versenden und Updates von Software [129],
- Patient Data Exchange (PDX) für den Austausch von Patientendaten [128],
- Text Integration Utilities (TIU) für das einheitliche Handling von klinischen Dokumenten [127],
- VA ClassMan für die multimediale Präsentation [126],
- VA FileMan für die Verwaltung von Daten mit Hilfe von Datenbanksystemen [125],
- VA Generic Code Sheet für das Handling von Codierungsblättern [124] und
- Visit Tracking für die Verknüpfung von Aktivitäten zwischen verschiedenen Einrichtungen [123].

Dieses Konzept machte gewaltige organisatorische, personelle und finanzielle Anstrengungen notwendig, die nicht in jeder Einrichtungen zur Verfügung stehen.

1.2.11 Die Analyse von „freien“ Befundtexten

Um die Semantik eines Befundtextes zu erhalten, wird oft eine Analyse des Befundtextes durchgeführt, der als *Freitext* vom Arzt eingegeben worden ist. Ash, J., be-

richtete 1995 über ein solches Projekt [4]. Gabrieli, E. R., analysierte Entlassungsbefunde, um medizinische Fakten herauszufinden [34].

1.2.12 Weitere Arbeiten auf dem Gebiet der medizinischen Standards

Chute, C. G., et al. berichteten 1996 über ein interessantes Projekt [16]. Ziel des Projekts war, zu klären, inwieweit die medizinischen Standards in der Lage sind, die medizinischen Begriffe abzubilden. Untersucht wurden hierbei ICD-9, ICD-10, CPT (Current Procedural Terminology [122]), SNOMED International, Read V2, UMLS V1.3 und NANDA. Chute versuchte, die aus vier klinischen Zentren erstellten Befundtexte mit Hilfe der Standards zu codieren. Die Bewertung eines Codiervorgangs erfolgte mit einer Note zwischen 0 für „nicht-möglich“ und 2 für „vollständig-codierbar“. Der Standard ICD-9 hatte eine Durchschnittsnote von 0,77, wobei ICD-9 in der Kategorie „Diagnose“ eine Note von 1,61 bekam. Dies war auch zu erwarten. Daß ICD-10 in der Sparte „Diagnose“ mit 1,60 und durchschnittlich mit 0,62 schlechter als ICD-9 abschnitt, war jedoch ziemlich überraschend. SNOMED International mit einer Durchschnittsnote von 1,74 und einer Note von 1,90 für „Diagnose“ stand ziemlich an der Spitze. Chute kam damit zu dem Schluß, daß z. Z. kein Allzweckstandard existiere, obwohl SNOMED in allen Bereichen führte.

Cimino, J. J., et al. versuchten, die Grundlage für den Austausch medizinischer Daten mit Hilfe von Software zu bilden, die die medizinischen Terme automatisch übersetzt [18]. Dielh, M., stellte 1995 ein Modell für die Abschätzung von Nutzen und Effektivität medizinischer Informationssysteme auf und warf die Frage des sogenannten „Systemreengineering“ im medizinischen Bereich auf [25]. Einen sogenannten „essential data set“ definierten Miodu, K., et al. [67]. Prokosch, H. U., et al. untersuchten die Wissensrepräsentation in der Pharmakologie an Hand der Arden-Syntax [72]. Die Universität Iowa entwickelte ein eigenes System für die Dokumentation mit dem Namen INFORMM (Prophet, C.M. [73]).

In Europa startete die Europäische Union 1994 ein umfangreiches Förderprogramm für Medizin, *Biomed* genannt, sowie das Programm AIM Advanced Informatics in Medicine. Ein gefördertes Projekt war NUCLEUS, das einen Prototyp für integrierte elektronische Krankenakten beinhaltet (Kanoui, H., et al. [54]). Thurin, A., et al. kombinierten die Arden-Syntax mit der Semantik von GALEN, einem Ergebnis des von der EU im Rahmen von AIM geförderten Projekts „Generalized Architecture for Languages, Encyclopedia and Nomenclature in Medicine“. Die Autoren stellten fest, daß SNOMED und ICD für viele Anwendungen nicht detailliert genug sind [81].

In der Bundesrepublik startete 1996 die Deutsche Telekom eine Werbekampagne für DOXX, ein auf ISDN basierendes Kommunikationssystem für die Anbindungen von Ärzten an DOXX-Server (Enge, J. [106]), (Deutsches Krebsforschungszentrum [102]). DOXX enthielt bereits bei der Bekanntmachung auf der MEDICA 96 zwei Module (Radiologie und Pathologie) und zwei Kooperationspartner [101].

1.3 Resumée

Nach der Analyse des aktuellen Forschungsstandes im Bereich der Software-Technologie und der medizinischen Standards kann folgendes festgestellt werden:

1. Ein integriertes medizinisches Informationssystem, das für alle Bereiche eingesetzt werden kann und damit das Problem des Datenaustausches unnötig macht, ist weiterhin nicht in Sicht (Jagannathan, V., et al. [51]). Standards für den Austausch medizinischer Informationen fehlen noch (Williams, M. H., et al. [87]).
2. Die technischen Hilfsmittel, um einen Standard für den Austausch von medizinischen Daten auf einem hohen semantischen Niveau zu ermöglichen, sind gegeben. Hier sind beispielsweise zu nennen: Java, HTML/XML. Mit der Implementation der „Healthcare Professional Workstation“ konnte Esterhay, R. J. Jr., feststellen, daß die Grenzen einer solchen Anwendung nicht technischer Natur sind, sondern in der Wirtschaftlichkeit, in den Datenschutzfragen und Standards liegen [30].
3. Die z. Z. verfügbaren Standards im medizinischen Bereich decken aus mehreren Gründen den Bedarf an einem allgemeinen Standard nicht. Einerseits ist ein Standard nur für einen speziellen Anwendungsbereich konzipiert; HL7 für die Verwaltung, DICOM für die Bildübertragung und SNOMED für eine einheitliche Benennung. Andererseits werden die Standards an Hand von fester Syntax spezifiziert. Sie sind umfangreich und wenig flexibel.
4. Will man einen allgemeinen Austausch medizinischer Daten verwirklichen, so muß ein allgemeiner, einfacher Standard definiert werden. Eine Integration auf dem herkömmlichen Weg ist und bleibt immer sehr komplex wie Cornet, R. aus einem Integrationsprojekt für eine einzige kardiographische Ambulanzklinik schlußfolgerte [19]. Alternativ dazu kann man sich vorstellen, daß für jede Disziplin ein Standard vom Umfang wie DICOM definiert werden müßte, und dies würde einen kaum vorstellbaren Aufwand für die Normung, Implementation, Verifikation und Pflege der Standards bedeuten. Die bereits sehr umfangreiche Spezifikation von ICD-10 ist für spezielle Anwendungen wie Fetalpathologie oder Hausmedizin noch immer nicht fein genug strukturiert. Für Hausmedizin wurde beispielsweise ICPC International Classification of Primary Care definiert (Hofmans Okkes, I. M. [45]).

Ein allgemeiner Standard muß daher so allgemein und einfach sein, daß der Normungsprozeß ebenso wie die Implementation und die Pflege mit minimalem Aufwand zu betreiben sind. Dies wurde u. a. vom Komitee für minimale Standards der ESGE vorgenommen [21]. Ein weiteres Beispiel war die Implementation von Essin, D. J., et al. die mit Hilfe eines *Wörterbuchs* versuchten, die Komplexität des Datenbankdesigns zu reduzieren [29].

2 Aufgabenstellung

In der vorliegenden Arbeit soll ein den im Abschnitt 2.2 *Anforderungskatalog* beschriebenen Anforderungen genügender Standard vorgeschlagen werden. An Hand vorhandener Anwendungen in der Universitätsfrauenklinik und im Institut für Pathologie der Charité wird untersucht, inwieweit ein solcher Standard aus technischer Sicht realisierbar ist und den medizinischen Anforderungen genügt.

In diesem Kapitel wird die Aufgabe - auf der obigen Problemstellung und dem Ergebnis der Recherche über den Stand der technologischen Forschung basierend - formuliert. Hierbei wird zunächst das Ist-System beschrieben, für das das Kommunikationskonzept entwickelt werden soll.

Im ersten Abschnitt „*Systemabgrenzung*“ wird der Definitionsbereich des Systems festgelegt. Hier wird beschrieben, was zum System gehört bzw. eine Schnittstelle mit dem System besitzt.

Anschließend wird der Anforderungskatalog zusammengestellt. Im Anforderungskatalog werden die zu lösenden Aufgaben sowie die Randbedingungen festgelegt, die bei der Lösung der Aufgaben zu beachten sind. Sie bestimmen somit die Lösungsmenge.

2.1 Systemabgrenzung

Im folgenden wird das Ist-System beschrieben, für das die Aufgabe formuliert wird. Hierbei spielen die folgenden Komponenten eine wichtige Rolle für die vorliegende Arbeit:

- die EDV-Kommunikationsstruktur mit der Server- und Netzwerkebene sowie
- die Anwendungsebene.

Die im folgenden am Beispiel der Charité gemachten Angaben sind typisch für große Universitätsklinika anzusehen.

Die Server-Ebene

In der Charité existiert ein heterogenes Netz von Servern aller Funktionen, angefangen von reinen File-Servern über Datenbankserver bis hin zu speziellen Anwendungsservern. Primär werden File-Server mit dem Betriebssystem Netware 3.x eingesetzt. Z. Z. sind über 40 Novell 3.x Server und etwa 10 Novell 4.x Server am Netz (Stand Dezember 1996). Dazu kommen etwa 40 Apple Macintosh in 20 Zonen. Eine geringe Anzahl von UNIX-Rechnern von HP und SUN, sowie Windows NT Server sind ebenfalls vorhanden. Im Rechenzentrum sind DEC-Rechner mit dem DecNet PathWork im Einsatz. Hierbei ist der Trend zu Windows NT als Server-Betriebssystem eindeutig erkennbar.

Die Netzwerkebene

Das Netzwerk der Charité wird in mehrere Subnetze unterteilt, die durch sehr unterschiedliche Verkabelungen über eine relativ homogene Router-Struktur verbunden sind. Die Verkabelungen reichen von Koax über Twisted Pair bis hin zu Glasfaser.

ISDN-Leitungen sind in Vorbereitung. Ein ISDN-Knoten mit 64 S₀ Anschlüssen wurde bereits in Betrieb genommen. Auf diesen Verkabelungen wird hauptsächlich Ethernet II für das Protokoll TCP/IP, IPX/SPX, DecNet und FDDI, sowie Ethernet Snap für AppleTalk geroutet. Das Microsoft LAN Manager Protokoll, das auf Broadcasting-Verfahren basiert, wird z. Z. nicht geroutet. Ferner wird das ATM-Protokoll eingesetzt, um die Verbindung mit dem UKRV, Universitätsklinikum Rudolph-Virchow, herzustellen. Da alle Server TCP/IP unterstützen, kann das System theoretisch mit einem Protokoll auskommen.

Die Anwendungsebene

Auf diesem relativ heterogenen Hintergrund laufen eine nicht ohne weiteres bezifferbare Anzahl von Anwendungen, die meist keine Verbindung miteinander haben. Hierbei stellen die DOS/Windows-Clients mit Abstand die Mehrheit dar (mehr als 700 Systeme). Abbildung 2-5 zeigt die Entwicklung der Computervernetzung in der Charité (Nguyen-Dobinsky, T. N., et al. [70])

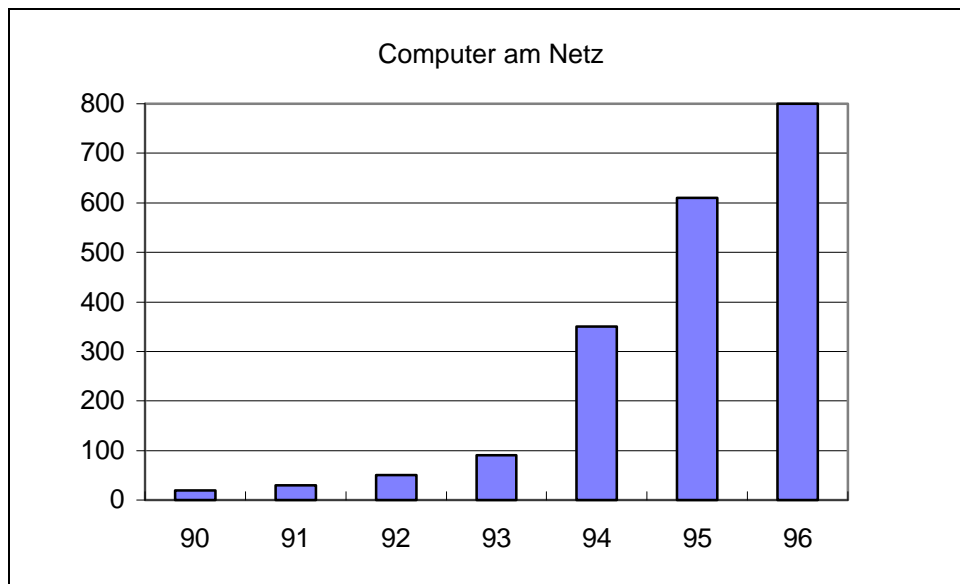


Abbildung 2-5 Entwicklung der Computervernetzung in der Charité

Diese Vielfaltigkeit der Systeme führt zwangsläufig zu einer Reihe von Problemen, die bereits im Kapitel Einleitung angeführt wurden.

In diesem Abschnitt wird die Systemabgrenzung am Beispiel der pränatalen Diagnostik konkreter vorgenommen. Für andere Kliniken bzw. Institute kann entsprechend vorgegangen werden.

Patientendaten: Das hier verwendete Subsystem speichert Daten über die schwangere Patientin, über ihre Anamnese, ihre Schwangerschaft und die Ultraschalluntersuchungen, die in der Abteilung durchgeführt werden. Eine direkte Kommunikation mit dem Identifikationssystem des Charité-Patientenverwaltungssystems existiert nicht.

Diagnose und Therapie: Die Untersuchungsdaten werden zum Teil über Bildschirmmasken, zum Teil über Freitext erfaßt. Diese Freitexte verlieren bei einer maschinellen Auswertung in der Regel ihre Semantik, ihre Bedeutung. Die Beschreibung der Diagnose bzw. der Therapie wird z. Z. weder vom Subsystem selbst noch mit Hilfe eines zentralen Systems codiert.

Fremdbefunde: Die Daten aus Fremdbefunden sollen nicht nur in der codierten Form zwischen den medizinischen Subsystemen übertragen werden. Auch die textlichen Daten sollen in ein Fremdsystem transferierbar sein, damit die persönliche, medizinische Beurteilung eines Arztes wiedergegeben werden kann.

Abbildung 2-6 zeigt die für die vorliegende Arbeit geltende Systemabgrenzung.

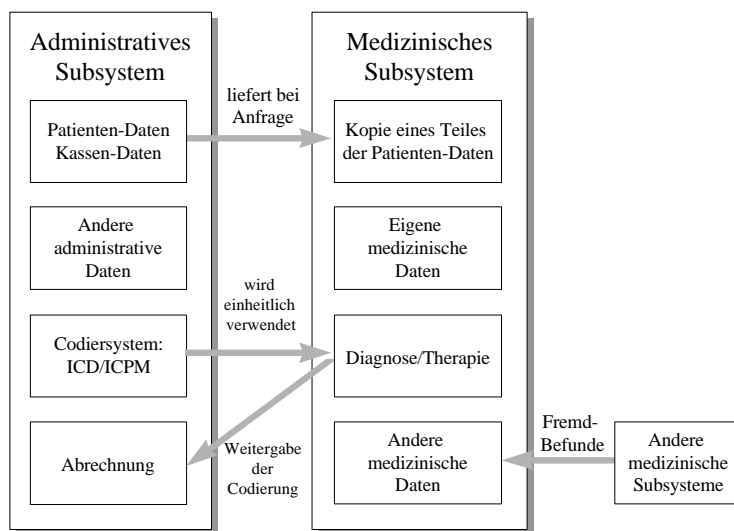


Abbildung 2-6 Systemabgrenzung

2.2 Anforderungskatalog

Wie oben bereits erwähnt, fehlt es an einer allgemeinen Möglichkeit, medizinische Informationen zwischen den medizinischen Subsystemen auszutauschen. Diese Möglichkeit kann durch einen Standard für Datenaustausch geschaffen werden. Als technische Voraussetzungen können Server, Netzwerk und Protokolle angesehen werden. In der Arbeit soll ein Standard für den Austausch von medizinischen Informationen vorgeschlagen werden, der folgende Hauptanforderungen erfüllen soll:

- Maximale Semantik
- Minimale Syntax und
- Praxisnähe.

Anhand von Beispielanwendungen in der Frauenklinik und im Institut für Pathologie wird untersucht, inwieweit ein solcher Standard aus technischer Sicht realisierbar ist, sowie den medizinischen Anforderungen genügt.

2.2.1 Maximale Semantik

Analog zu den relationalen Datenbanken darf die Semantik des Standards nicht anhand der Syntax definiert werden (s. DICOM). Mit anderen Worten muß eine neue

Bedeutung, ein neues Objekt oder ein neues Attribut durch einfaches Hinzufügen von Daten innerhalb des vorhandenen semantischen Rahmens erzeugt werden können. Man muß hierfür keinen neuen Datentyp definieren.

Die Anforderung nach maximaler Semantik steht an erster Stelle des Anforderungskatalogs. Um einen korrekten Informationsaustausch zu gewährleisten, muß die medizinische Semantik erhalten bleiben. Bereits 1980 plädiert Cerutti, S., für die Erhaltung der Semantik in medizinischen Datenbanken [14]. Williams, B. T., et al. fanden ebenfalls heraus, daß eine akkurate Semantik dem Anwender mehr Nutzen bringt [86].

Um festzustellen, ob die medizinische Semantik gewährleistet ist, wird folgende Definition formuliert:

Medizinische Identität

Zwei Datensätze sind medizinisch identisch, wenn: alle nachfolgenden Bedingungen (a-c) erfüllt sind:

- a. Sie das gleiche medizinische Objekt beschreiben.
- b. Jedes Attribut, das in dem einen Satz enthalten ist, auch in dem anderen enthalten sein muß. Hierbei müssen die Attributwerte jeweils identisch sein.
- c. Folgendes gilt: zwei Attributwerte sind medizinisch identisch, wenn ein beliebiger praktizierender Arzt nicht unterscheiden kann, ob der eine Wert durch einen maschinellen Transfer oder durch eine manuelle Übersetzung durch einen Mediziner zustande gekommen ist.

Diese Definition macht bezüglich der Darstellung keine Aussage. Sie überläßt sie den jeweiligen Subsystemen. Im wesentlichen kommt es darauf an, wie festgestellt werden kann, ob zwei Attributwerte medizinisch identisch sind. Hier bedient sich der Autor des sogenannten Turing-Testes. Dieser Test ist ein Gedankenspiel, das von dem englischen Mathematiker Alain Turing als Kriterium für die künstliche Intelligenz eingeführt wurde [103].

Die Prüfung, ob ein Austausch die Anforderungen erfüllt, geschieht durch folgende Tests:

a. Vollständigkeitstest

Ein System sendet einen beliebigen Datensatz an ein zweites System. Nach dem Transfer muß das Empfängersystem den Datensatz so darstellen können, daß der Inhalt des Datensatzes - medizinisch gesehen - vollständig erhalten bleibt, also mit dem ursprünglichen Datensatz medizinisch identisch ist. Es ist hierbei unerheblich, wie die Art der Darstellung gewählt wird (beispielsweise als Text, Zahlen-tupel oder Diagramm).

b. Reziprokttest

Ein System sendet einen beliebigen Datensatz an ein zweites System. Dieses sendet denselben Datensatz nach einer beliebigen Zeit wieder an den Sender zurück. Der jetzt vom Sendesystem empfangene Datensatz muß medizinisch identisch mit dem ursprünglichen Satz sein.

c. Looptest

Ein Austausch besteht den Looptest, wenn der Reziprokttest nach einer beliebigen Anzahl von Wiederholungen immer noch erfolgreich ist.

Darstellbarkeit

Nach einem Austausch kann sich die Darstellung ändern. Dies ist der Fall, wenn das Empfängersystem in der Lage ist, die erhaltenen Informationen zu präsentieren, ohne auf die Funktionalität des Sendesystems angewiesen zu sein. Beispielsweise wird im Sendesystem, einem pränatalen Diagnostiksystem, der Millimeter mit einer Dezimalstelle als Längenmaß verwendet. Im Empfängersystem wird dagegen Zentimeter, ebenfalls mit einer Dezimalstelle benutzt. Wird nun ein Wert von der Größe 0,2 mm an das Empfängersystem geschickt, so zeigt dieser Wert erwartungsgemäß im Empfängersystem 0,0 cm an. Somit gehen Informationen durch die unterschiedlichen Darstellungen verloren, was zu vermeiden gilt.

Fehlertoleranz

Es ist zu erwarten, daß ein Empfängersystem nicht alle Informationen interpretieren kann, die es vom Sendesystem erhält. In diesem Fall muß das Empfängersystem gemäß folgender abnehmender Priorität verfahren:

1. Übernahme der Information mit voller Semantik.
2. Übernahme der Information als Text, d. h. Semantik ist nur für menschliche Leser erhalten. Automatisierte Verarbeitung ist damit nicht mehr möglich.
3. Ablehnung der Übernahme und Versuch, die Funktionalität des Sendesystems in Anspruch zu nehmen.
4. Ablehnung der Übernahme und Protokollierung der Ablehnung.

Ist eine einzelne Information nicht übernehmbar, so muß das Empfängersystem versuchen, die restlichen Daten zu verarbeiten, es sei denn, der Datensatz wird ohne die betroffene Information sinnlos.

2.2.2 Minimale Syntax

Um den Umfang des Standards klein zu halten, sollten so wenig Datentypen (Objekte) wie möglich definiert werden. Optimal wäre es, wenn der Standard nur einen einzigen Datentyp und eine einzige Operation/Methode auf diesem Datentyp definiert. Ein Beispiel hierfür ist die relationale Datenbank. Sie kommt mit der Tabelle als einzigem Datentyp und mit der Operation SELECT als einziger Operation aus.

2.2.3 Praxisnähe

Folgende Detailanforderungen müssen erfüllt sein. Sie sind in der EDV allgemein bekannt und werden hier nicht mehr näher erläutert.

- Sicherheit der Übertragung,
- Sicherheit der Daten,
- Schutz des Urheberrechts,
- Schutz der forschungsrelevanten Daten,

- Plattformunabhängigkeit und Internationalisierbarkeit,
- Minimaler Realisierungs- und Pflegeaufwand und
- Minimaler Overhead

3 Methodik

Die Analyse der im Kapitel 2 *Aufgabenstellung* gestellten Anforderungen ergab, daß der Kern der Lösung in der Erhaltung der Semantik der Informationen liegen muß. Das ist die wichtigste Anforderung, die bisher wenig beachtet wurde. Die übrigen Anforderungen wie Sicherheit, Plattformunabhängigkeit, etc. sind in der Regel allgemein bekannt. Sie lassen sich mit einigem Bemühen unter Zuhilfenahme von vorhandenen technischen und/oder organisatorischen Hilfsmitteln ohne weiteres erfüllen.

Da die Semantik der Informationen in erster Linie vom Sende- und Empfängersystem, also den beteiligten Subsystemen, abhängt, müssen alle Anstrengungen darauf konzentriert sein, eine optimale Art und Weise der Kommunikation zu finden. Ausgehend von diesem Kerngedanken wird ein Lösungsrahmen festgelegt. Anschließend wird die Philosophie des Zielsystems definiert. Darauf aufbauend wird die beim Entwurf verwendete Philosophie spezifiziert. Dies bildet die Vorgabe für die Konzeptionsarbeiten, deren Ergebnisse sich im Kapitel 4 *Ergebnisse* in einer speziellen Lösung niederschlagen.

3.1 Lösungsrahmen

Da der Schwerpunkt der vorliegenden Arbeit im Austausch medizinischer Informationen liegt, spielen die medizinischen Subsysteme die Hauptrolle. Diese werden jedoch auf unterschiedlichste Art und Weise beschafft und eingesetzt. Man kann hier keine Vorschrift erlassen, wie ein Subsystem konstruiert werden muß. Ferner muß in einem großen Universitätsklinikum, in dem führende medizinische Forschungen betrieben werden, immer damit gerechnet werden, daß nicht nur neue, sondern auch neuartige medizinische Software eingesetzt wird, die nicht in den eventuell vom Klinikum vorgegebenen Rahmen paßt. Adelhard, K., et al. hat 1995 ein Konzept veröffentlicht, das eine Integration von neuen Methoden und Konzepten zuließ [2]. Walker, A. stellte ebenfalls fest, daß jedes Subsystem seine Eigenheiten besitzt [84].

Ein Standard für medizinischen Informationsaustausch muß die Existenz solcher Software berücksichtigen und sich daher auf den Informationsaustausch beschränken. Dieser Standard darf das innere Verhalten der Subsysteme nicht vorschreiben. Für die Funktionsfähigkeit eines auf diesem Standard basierenden Systems muß einzig die Fähigkeit, Informationen auf semantischer Ebene auszutauschen, entscheiden, ob ein Subsystem angeschlossen wird.

Die Hauptstrategie bei der Entwicklung eines solchen Standards muß aus diesen Gründen darin bestehen, daß die weiterhin autonom arbeitenden Subsysteme kommunikationsfähiger werden. Hierbei ist es unerheblich, wie ein Subsystem kommunikationsfähiger gemacht wird. Es kann von Hause aus, beispielsweise durch eine Neukonzeption (nativ) oder einfach durch den Einsatz eines Adapters (adaptiv) die Kommunikationsfähigkeit erhalten.

Weiterhin darf der Standard nicht versuchen, alles zu regeln. Er sollte lediglich einen Rahmen für die Kommunikation zur Verfügung stellen. Wie die Daten schließlich interpretiert werden, müssen die Subsysteme untereinander ausmachen. Somit kann

der Standard beispielsweise von der Entwicklung anderer Standards wie HL7, DICOM, SNOMED, etc. , die mit sehr hohem Aufwand von sehr vielen Fachleuten erarbeitet worden sind, abgekoppelt werden. Beispielsweise können sich zwei Subsysteme im Rahmen des Standards verständigen und reine HL7-Daten oder DICOM-Bilder austauschen.

Zusammengefaßt läßt sich die Lösungsstrategie durch folgende Punkte beschreiben:

- Die Subsysteme kommunizieren miteinander auf eine tolerantere Art und Weise. Wenn beispielsweise zwei Subsysteme miteinander Daten austauschen und dabei eventuelle Inkompatibilitäten feststellen, müssen sie versuchen, die möglicherweise vom Empfänger benötigten Informationen auf einer niedrigeren Semantikebene zu übertragen.
- Diese Toleranz in der Kommunikation muß über verschiedene Grenzen hinweg funktionieren. Es sind erstens die Grenze zwischen den medizinischen Fachbereichen wie Geburtshilfe/Pathologie oder Geburtshilfe/Neonatologie, zweitens die Diskrepanz zwischen den Systemplattformen und drittens die zwischen den verschiedenen Sprachen zu beachten.
- Im ersten Schritt ist damit zu rechnen, daß die meisten Subsysteme keine native Fähigkeit zu kommunizieren mit sich bringen. In diesem Fall wird das Subsystem einfach durch die sogenannte Verpackungstechnik (Wrapping) dazu fähig gemacht.

3.2 Systemphilosophie

Die Systemphilosophie bestimmt, in welche Richtung die Architektur des Systems und des Standards gehen sollen und ob das System z. B. monolithisch oder verteilt sein soll. Die Systemphilosophie basiert auf folgenden Prinzipien:

- Der Standard beschränkt sich auf den Austausch der medizinischen Daten zwischen den medizinischen Subsystemen bzw. zwischen einem medizinischen Subsystem und einem administrativen Subsystem.
- Der Standard legt ein Netz fest. An jedem Knoten des Netzes wird ein medizinisches Subsystem über einen sogenannten virtuellen Befundtreiber angeschlossen. Administrative Informationen werden über HL7 ausgetauscht.
- An vorhandenen Subsystemen wird, abgesehen vom Anschluß eines virtuellen Befundtreibers, nichts geändert. Jedes Subsystem kann sehr proprietär sein. Wenn es jedoch über den Treiber an das Netz angeschlossen werden kann, so ist es aus der Sicht des Netzes ein Lieferant von Informationen für das Netz, also für die restlichen Netzbenutzer. Es kann ebenfalls Informationen aus dem Netz beziehen und dem lokalen Anwender zur Verfügung stellen. Jedes Subsystem ist für den Teil verantwortlich, für den es konstruiert worden ist.
- Es entsteht somit ein Netz von kooperierenden Subsystemen, das sogenannte virtuelle Befundsystem.
- Die medizinische Semantik wird durch zwei Maßnahmen gewährleistet. Die erste besagt, daß der verantwortliche Knoten für die Interpretation und Darstellung des

Inhaltes sorgt, denn er ist der Knoten, der mit der Semantik des betroffenen Objekts am besten umgehen kann. Die zweite Maßnahme umfaßt die Definition eines medizinischen Kontextes. Jeder Knoten stellt seinen Kontext zur Verfügung. Wird ein Objekt aus einem bestimmten Kontext in einen anderen importiert, so kann an Hand der Kontexte geprüft werden, ob ein Austausch ohne Semantikverlust vorgenommen werden kann. Wenn nicht, werden die Informationen als reiner Text bzw. reine Graphik übertragen.

- Das vorhandene Netzwerk wird benutzt. Standardprotokoll ist HTML/XML in einer Internet-Struktur: TCP/IP, Domain Name Service.
- Das Konzept ist objekt-orientiert, so daß die Objekte eine Reihe von standardisierten Operationen selbst bieten, mit denen ein Client das Objekt erzeugen, manipulieren und wieder löschen kann. Jedes Objekt ist weltweit eindeutig identifizierbar. Hierfür wird das DNS-Konzept (Domain Name Service) erweitert und das allgemeine Objektkonzept entwickelt [3].
- Der größte Teil der notwendigen Software wird nur einmal entwickelt und kann dann in allen Einrichtungen durch Anpassung in Betrieb genommen werden. In den jeweiligen Einrichtungen muß lediglich die Anpassung in Form des virtuellen Befundtreibers für den Anschluß des Subsystems vorgenommen werden.
- Im ersten Schritt wird davon ausgegangen, daß die medizinischen Subsysteme Informationen für ein virtuelles Befundsystem liefern. Eine Veränderung der medizinischen Subsysteme durch das virtuelle Befundsystem wird erst im nächsten Schritt geplant und ist nicht Teil dieser Arbeit.

3.3 Entwurfsphilosophie

Die Entwurfsphilosophie bestimmt die Eckpunkte des Systementwurfs. Beispielsweise soll hier der Systementwurf objekt-orientiert erfolgen. Die Hauptphilosophie des Entwurfs ist die Objektorientierung. Bereits 1989 hat Barsalou, T., ein Konzept veröffentlicht, das die Modellierung von Objekten in relationalen Datenbanken beinhaltet [5]. Neben diesem objekt-orientierten Konzept umfaßt die Entwurfsphilosophie folgende Teilkonzepte:

- ein allgemeines Objektkonzept,
- einen medizinischen Kontext,
- einen virtuellen Namensraum und
- ein Transaktionskonzept.

3.4 Das allgemeine Objektkonzept

3.4.1 Motivation

Im vorangegangenen Abschnitt ist ein Überblick über die wichtigsten Objekte des Konzepts gegeben worden. Dort wurden ihre Merkmale - ihre Semantik - erläutert. Diese Objekte haben voraussichtlich sehr viele verschiedene Eigenschaften. Ein Versuch, diese Eigenschaften in einer Datenbank oder in einer Objektstruktur zu strukturieren, scheint kein gangbarer Weg zu sein. Ein Beleg hierfür sind die langjährigen und schwierigen Standardisierungsbemühungen in HL7, DICOM,

SNOMED, etc. Hinzu kommen die laufenden Änderungen, die in diesen langwierigen Prozessen integriert werden müssen.

Aus diesem Grund muß ein Objektkonzept entwickelt werden, das einen einfachen Mechanismus für die Handhabung und Änderungen von umfangreichen Merkmalen besitzt.

Das so definierte Objektkonzept bringt folgende Vorteile:

- Eine autonome Existenz von Objekten in den Subsystemen ist möglich. Dies erlaubt die getrennte Entwicklung von Knoten.
- Änderungen an einem Objekt im Netz können durchgeführt werden, ohne daß an benachbarten Subsystemen Update- bzw. Upgrade-Arbeiten gemacht werden müssen.
- Die heute vornehmlich verwendeten Programmierwerkzeuge unterstützen das Objektkonzept von Hause aus. Die auf der PC- und UNIX-Plattform verbreitete Programmiersprache C++ - und zunehmend auch Java - ist beispielsweise objektorientiert.

3.4.2 Unterschied zum herkömmlichen Objektkonzept

Der Kerngedanke hierbei ist der folgende. Alle Informationen, die im Rahmen des Konzepts verwendet werden sollen, sind entweder ein Objekt oder ein Attribut eines Objekts. Ein Objekt ist eine Menge, deren Elemente Attribute sind. Ein Attribut kann wiederum ein Objekt sein (s. unten).

Objekt
 Attribut 1
 Attribut 2
 Attribut x
 Attribut n
 (beliebig fortsetzbar)

Ein kleines Beispiel verdeutlicht den Unterschied zwischen diesem allgemeinen und dem herkömmlichen Objektkonzept. Wir betrachten im folgenden das Objekt *Patient*. Im herkömmlichen Konzept würde ein Entwurf wie folgt aussehen:

Patient
Familienname = Mustermann
Vorname = Angelika
Geburtsname = Testmann
Geburtsort = Deutschland
Geburtsdatum = 1.1.1900
Nationalität = DE

Das Objekt Patient im allgemeinen Objektkonzept würde wie folgt konstruiert sein:

Object: Class = Patient
Attribute: Name = Familienname; Type = Text; Value = Mustermann
Attribute: Name = Vorname; Type = Text; Value = Angelika
Attribute: Name = Geburtsname; Type = Text; Value = Testmann
Attribute: Name = Geburtsort; Type = Text; Value = Deutschland
Attribute: Name = Geburtsdatum; Type = ShortDate; Value = 1.1.1900
Attribute: Name = Nationalität; Type = NationalCode; Value = DE

Aus den 6 verschiedenen Attributen (Familiename bis Nationalität) sind 6 Einträge von *einem* Typ **Attribute** geworden. Ein Empfängersystem würde mit der zweiten Varianten weniger Probleme haben, selbst wenn aus Empfängersicht unbekannte bzw. unbrauchbare Attribute vorkommen. Das System kann beispielsweise ein Attribut leichter überlesen bzw. ignorieren.

Die Attribute eines Objekts werden auf eine einheitliche Art und Weise beschrieben. Somit hat das allgemeine Objektkonzept nur ein einziges Syntaxelement. Diese Konstruktion birgt folgende Vorteile:

- Sie erfüllt die Anforderung nach minimaler Syntax, da das Objekt nur aus einer Liste von Attributen besteht, gleichgültig, welchen Typs die Attribute sind.
- Die Tabellen von relationalen Datenbanken, die diese Objekte speichern sollen, können auf eine einheitliche Art und Weise aufgebaut werden. Sie haben immer die gleiche Anzahl von Spalten. In jeder Tabelle hat jede Spalte immer die gleiche Bedeutung. Die Software läßt sich damit erheblich vereinfachen. Sie kann einmal entwickelt und für alle Knoten eingesetzt werden. Die Software, die die Attribute formal bearbeitet, wird Attribute-Engine genannt. Sie stellt einen Rahmen, ein Frame-Work für das System dar.
- Die Semantik eines Objekts wird durch die Semantik der einzelnen Attribute bestimmt. Da die Liste der Attribute nicht begrenzt ist, kann hier jedes beliebige Attribut an das Objekt angehängt werden.
- Die Erweiterung eines Objekts um weitere Attribute erfolgt nur durch Hinzufügen einiger Zeilen in der Tabelle. Das bereitet der jetzigen Generation von Datenbanksystemen keinerlei Schwierigkeit.
- Somit ist die Forderung nach maximaler Semantik für den inneren Aufbau eines Objekts erfüllt. Die Forderung, daß die Semantik bei der Übertragung nicht verlorengelangen darf, wird durch die Einführung des medizinischen und des Darstellungskontextes gewährleistet.

3.5 Syntax der Objektbeschreibung

Die Syntax der Objektbeschreibung basiert auf dem HTML/SGML-Standard. Im folgenden wird sie an Hand der Metaspezifikation festgelegt, wobei der Zeilenumbruch keine syntaktische Bedeutung hat. Er dient lediglich der Übersichtlichkeit.

Bei den kursiv zwischen den Spitzklammern geschriebenen Texten handelt es sich um die zu beschreibenden Elemente. Die fettgedruckten, nicht-kursiv geschriebenen Texte sind Syntaxzeichen.

```
<object> ::=      <OBJECT
                   TYPE = <object_type>
                   INTERNAL-ID = <object_internal_ID>
                   EXTERNAL-ID = <object_external_ID>
                   CODEBASE = <code_base_URL>>
                   {<attribute>}
                   </OBJECT>

<attribute> ::=  <ATTR
                   NAME = <attribute_name>
```

```

TYPE = <attribute_type>
VALUE = <attribute_value>
[UNIT = <attribute_unit>]
>

```

Ein Objekt besteht syntaktisch aus zwei Teilen, dem Objekt-Header und dem Objekt-Body. Im Header ist die Identifikation des Objekts gespeichert. Im Body sind alle anderen Objektinhalte enthalten, die durch Attribute beschrieben werden. Streng genommen läßt sich die Identifikation im Objekt-Header auch als Attribut deklarieren. In der Praxis wird ein getrennter Header die Verarbeitung beschleunigen. Zugunsten der Performance wird hier ein „Stilbruch“ begangen, in dem die wichtigsten Informationen über das Objekt im Header angesiedelt und damit eine schnelle Identifikation ermöglicht, ohne daß alle Attribute durchsucht werden müssen.

Ein Objekt wird durch die Type-ID <object_type>, und die globale ID eindeutig identifiziert. Die globale ID wird durch die interne ID <object_internal_ID> und die externe ID <object_external_ID> beschrieben. Welches Applet das Objekt primär behandeln soll, wird durch die URL-Angabe **CODEBASE** festgelegt.

Jedes Objektattribut wird durch ein Tupel, bestehend aus *Name*, *Type*, *Value* und *Unit* festgelegt. Sie sind beliebig wiederholbar, wobei die Einheitsangabe optional ist. Objektattribute sind beliebig schachtelbar, so daß jede Konstruktion auch rekursiv möglich ist. Ein Attribut kann beispielsweise wieder ein Objekt sein.

Somit kann sowohl die Forderung nach minimaler Syntax als auch die Forderung nach maximaler Semantik erfüllt werden.

Beispiel

```

<OBJECT TYPE="Patient" INTERNAL-ID="x.x.x.x:3031/4711" EXTERNAL-
ID="4711@us.prenatal.ufk.charite.hu-berlin.de" CODEBASE="ufk-science.charite.hu-
berlin.de/prenatal-ultrasound/patient.exe"
  <ATTR NAME="FirstName" TYPE="Text" VALUE="Angelika" >
  <ATTR NAME="Name" TYPE="Text" VALUE="Mustermann">
  <ATTR NAME="Birthday" TYPE="ShortDate" VALUE="01.01.1953">
  <ATTR NAME="Diagnosis" TYPE="ICD-10" VALUE="P 27.0">
  <ATTR NAME="Diagnosis" TYPE="Text" VALUE="singuläre Umbilikalarterie">
  <ATTR NAME="Object" TYPE="Reference" VALUE="9876543421">
  <ATTR NAME="Histology" TYPE="Img" VALUE="http://www.ufk.prenatal.charite.de/prenatal-
db/images/1111111.tif">
</OBJECT>

```

3.6 Der medizinische Kontext

3.6.1 Zweck des medizinischen Kontextes

Jedes Subsystem behandelt ein anderes medizinisches Fachgebiet, hat in der Regel andere Ziele, nutzt deshalb andere Methoden und andere Terminologie. Diese Unterschiede müssen beim Datenaustausch berücksichtigt werden. Beispielsweise fokussiert der Geburtshelfer bei der Betrachtung des Ultraschallbildes einer Schwangeren auf mögliche Herz- und Lungenanomalien des Fetus. Der Urologe kann beim gleichen Bild seinen Blick auf die zusammengequetschte Harnblase konzentrieren. Die Untersuchungsverfahren der beiden Mediziner unterscheiden sich ebenfalls.

Beim Zusammenführen bzw. Austauschen dieser Informationen sorgt der medizinische Kontext verantwortlich für die korrekte Übertragung und Interpretation. Beim Versenden von Informationen von einem Subsystem zum anderen wird der medizinische Kontext des Sendesystems mitgeschickt. Beim Empfangen werden die Informationen an Hand des Quellkontextes und des Zielkontextes interpretiert. Man reißt somit die Daten nicht aus dem Zusammenhang und sie verlieren nicht ihre Bedeutung. Zudem wird ihre Bedeutung auch nicht verfälscht.

Das Konzept des medizinischen Kontextes basiert auf einem vielfach für hardware-unabhängige Gerätetreiber verwendeten Konzept. Hier wird ein sogenannter Device-Context verwendet, mit dessen Hilfe die Fähigkeiten eines Geräts auf einem hohen Abstraktionsniveau beschrieben werden.

Um den modifizierten Kontext zu bewahren, muß man für jedes Subsystem einen Kontext spezifizieren, der das Arbeitsgebiet des betroffenen Subsystems beschreibt. Der Kontext muß Informationen über die Patientengruppe enthalten, mit der das Subsystem arbeitet, über die Methoden, die im Subsystem benutzt werden, über die physikalischen Randbedingungen (constrains), die im Subsystem gelten, sowie weitere Parameter für die Systembeschreibung wie Organe, Funktionssysteme, etc. Ein Diagnosesystem für die pränatale Diagnostik kann beispielsweise mit folgendem Parametersatz versehen sein:

<i>Patient:</i>	<i>Weiblich</i>
<i>Alter:</i>	<i>zwischen 10 und 50</i>
<i>Organ:</i>	<i>Plazenta, Vagina, Fetus, etc.</i>
<i>Methoden:</i>	<i>2D-Sono, 3D-Farbdoppler.</i>

In der Fetalpathologie wird der Kontext z. B. folgende Einträge haben:

<i>Patient1:</i>	<i>Weiblich</i>
<i>Alter:</i>	<i>zwischen 10 und 50</i>
<i>Organe:</i>	<i>Plazenta</i>
<i>Methoden:</i>	<i>Makroskopie, Mikroskopie, etc.</i>

<i>Patient2:</i>	<i>Fetus</i>
<i>Alter:</i>	<i>zwischen 10 und 45 Schwangerschaftswochen</i>
<i>Organe:</i>	<i>Alle</i>
<i>Methoden:</i>	<i>Makroskopie, Mikroskopie, etc.</i>

Es ist hier ersichtlich, daß der Fetus bei der pränatalen Diagnostik nicht als Patient fungiert, bei der Fetalpathologie wird es jedoch als eigenständiger „Patient“ betrachtet. Ebenfalls wechselt die Methode von Sonographie in Makroskopie und Mikroskopie.

3.6.2 Struktur des medizinischen Kontextes

Die Struktur eines medizinischen Kontextes ist ähnlich wie die eines Objekts. Ein medizinischer Kontext ist ebenfalls eine Menge, deren Elemente Attribute sind. Diese beschreiben den medizinischen Kontext und somit das medizinische Subsystem. Hierbei lassen sich die Attribute in Gruppen zusammenfassen. Eine Gruppe von Gruppen ist ebenfalls möglich.

Einige Schlüsselwörter mit vordefinierter Bedeutung sind:

VALUE= „don't care“ Das Attribut muß vorhanden sein, dessen Wert spielt jedoch keine Rolle
MATCH=“yes“ Das Attribut wird zum Matchen benötigt
MATCH=“no“ Das Attribut wird zum Matchen nicht verwendet

3.6.3 Kontextwechsel

Der Übergang von einem Subsystem zum anderen wird durch den Kontextwechsel begleitet. Beim Kontextwechsel wissen die Subsysteme anhand der Parameter, wie das gleiche Attribut nun anders interpretiert und dargestellt werden muß, um im neuen Kontext einen Sinn zu geben. Diese Arbeitsweise ist semantik-orientiert.

Wird ein Objekt von einem Empfängersystem erneut weitergegeben, so wird der Originalkontext mitgegeben.

3.6.4 Eigenschaften medizinischer Kontexte

Es sei x ein Objekt aus dem Sendesystem mit dem medizinischen Kontext $md1$. Das im Empfängersystem mit dem medizinischen Kontext $md2$ ankommende Objekt sei x' . Die Transformation f wandelt Objekt x in x' um:

$$x' = f(x, md_1, md_2)$$

Die Transformation f soll folgende Eigenschaften aufweisen:

Identische Transformation

$$x = f(x, md_1, md_1)$$

Ein Objekt bleibt unverändert, wenn Sende- und Empfängersystem identisch sind ($md1 = md2$).

Verlustfreie und verlustbehaftete Transformation

Um festzustellen, ob eine Transformation verlustfrei ist, kann eine Loop-Transformation $L_n(x)$ vorgenommen werden.

$$x' = f(x, md_1, md_2)$$

$$L_1(x) := x'' = f'(x', md_2, md_1) = f' [f(x, md_1, md_2), md_2, md_1]$$

oder

$$L_1 = (f' \cdot f)^1(x) = f' \cdot f(x)$$

$$L_2 = (f' \cdot f)^2(x) = f' \cdot f \cdot f \cdot f(x)$$

...

$$L_n = (f' \cdot f)^n(x)$$

Eine Transformation ist verlustfrei, wenn nach einer beliebigen Anzahl von Loops folgendes gilt:

$$L_n(x) = x \quad \text{für } n \in N$$

Ist eine Transformation verlustbehaftet, so gilt:

$$L_n(x) \hat{=} x$$

Nach einer beliebigen Anzahl von Loops muß das Objekt vollständig im Ursprungsobjekt enthalten sein oder:

$$m[L_n(x)] \hat{=} m(x) \quad \text{für } n \in \mathbb{N}$$

wobei m die Mächtigkeit ist

Asymptotisches Verhalten

Nach einer beliebigen Anzahl von Loops darf das Objekt nicht verschwinden, sondern muß gegen eine Asymptote streben. Diese Asymptote ist die Untergrenze in der Erhaltung der Semantik. Ist eine Transformation verlustbehaftet, so müssen die Attribute in Primitiva (z. B. ASCII-Text) umgewandelt werden, falls das Empfängersystem diese nicht behandeln kann. Da jedes Subsystem die Primitiva behandeln können muß, dürfen sie nicht verlorengehen.

Da ein Objekt als eine Menge von Attributen definiert ist, kann für jedes Objekt die Mächtigkeit $m(x)$ bestimmt werden. Für die Transformation medizinischer Kontexte wird gefordert:

$$\lim_{n \rightarrow \infty} m[L_n(x)] = m_\infty > 0.$$

3.6.5 Kontext-Matching

Mit dem medizinischen Kontext und dem damit verbundenen Verhalten können die Subsysteme die von einem anderen Subsystem geschickten Informationen auf eine einheitliche Art und Weise filtern. Dies geschieht an Hand der oben beschriebenen Transformation. Sie bedient sich des folgenden Matching-Algorithmus'.

Das Matching wird einfach durch eine Durchschnittsbildung vorgenommen. Es seien md_1 der medizinische Kontext des Senders und md_2 der des Empfängers. Die Menge md wird durch die Operation:

$$md = md_1 \cap md_2$$

gewonnen. Ist md eine leere Menge, so hat der Empfänger keine Gemeinsamkeit mit dem Sender. Der Empfänger benötigt keine Information vom Sender. Ist md dagegen nicht leer, so kann der Empfänger md verwenden, um die in md enthaltenen Objekte und deren Attribute beim Anfordern der Information zu filtern.

Eine dynamische Generierung und/oder Anpassung von medizinischen Kontexten zur Laufzeit, beispielsweise für temporäre statistische Studien, ist ebenso denkbar wie erforderlich.

3.6.6 Syntax der Kontextbeschreibung

Die Syntax der Kontextbeschreibung basiert auf dem HTML/SGML-Standard. Im folgenden wird sie an Hand der Metaspezifikation festgelegt, wobei der Zeilenumbruch keine syntaktische Bedeutung hat. Er dient lediglich der Übersichtlichkeit.

Bei den kursiv zwischen den Spitzklammern geschriebenen Texten handelt es sich um die zu beschreibenden Elemente. Die fettgedruckten, nicht-kursiv geschriebenen Texte sind Syntaxzeichen.

Der medizinische Kontext ist eine einfache Liste von Attributen, die auch gruppiert werden können. Bis auf den Parameter MATCH sind alle anderen bereits in der Objektbeschreibung bekannt. MATCH gibt an, ob die Gruppe bzw. das Attribut beim Kontext-Matching verwendet werden soll. Beispielsweise wird das Attribut *Patient* immer, das Attribut *Organ* oft, das Attribut *Methode* in der Regel jedoch nicht zum Matchen verwendet.

```
<medical context ::= <MEDICALCONTEXT
    NAME = <object_type>
    INTERNAL-ID = <object_internal_ID>
    EXTERNAL-ID = <object_external_ID>
    CODEBASE = <code_base_URL>
    {<group>}
    {<attribute>}
</MEDICALCONTEXT >
```

```
<group> ::=
    <GROUP
    NAME = <object_type>]
    [INTERNAL-ID = <object_internal_ID>]
    [EXTERNAL-ID = <object_external_ID>]
    [CODEBASE = <code_base_URL>]
    [MATCH = „yes“ | „no“]
    {<group>}
    {<attribute>}
</GROUP>
```

```
<attribute> ::=
    <ATTR
    NAME = <attribute_name>
    TYPE = <attribute_type>
    VALUE = <attribute_value>
    [UNIT = <attribute_unit>]
    [MATCH = „yes“ | „no“]
    >
```

Beispiel

```
<MEDICALCONTEXT NAME="Prenatal-US" INTERNAL-ID = "x.x.x.x:3031" EXTERNAL-
ID="@us.prenatal.ufk.charite.hu-berlin.de" CODEBASE="ufk-science.charite.hu-
berlin.de/data/reference-server/prenatalcontext/md.exe>
  <GROUP NAME="Patient" MATCH="yes">
    <ATTR NAME="Sex" TYPE="Text" VALUE="female">
    <ATTR NAME="Sex" TYPE="SNOMED V3.3" VALUE="T-D0AA0">
    <ATTR NAME="MaxAge" TYPE="Integer" VALUE="50">
    <ATTR NAME="MinAge" TYPE="Integer" VALUE="10"> ....
  </GROUP>
  <GROUP NAME="ListOfOrganes">
    <ATTR NAME="Organe" TYPE="Text" VALUE="Placenta">
    <ATTR NAME="Organe" TYPE="SNOMED V3.3" VALUE="T-F1100">
    <ATTR NAME="Organe" TYPE="Text" VALUE="Fetus">
    <ATTR NAME="Organe" TYPE="SNOMED V3.3" VALUE="T-F5000">
    <ATTR NAME="Organe" TYPE="Text" VALUE="Uterus">
    <ATTR NAME="Organe" TYPE="SNOMED V3.3" VALUE="T-83000"> ...
  </GROUP>
```

```

<GROUP NAME="ListOfProcedures" MATCH=,no">
  <ATTR NAME="Procedure" TYPE="Text" VALUE="Ultrasonography, NOS">
  <ATTR NAME="Procedure" TYPE="SNOMED V3.3" VALUE="P5-B0000">
  <ATTR NAME="Procedure" TYPE="Text" VALUE="Diagnostic Doppler ultrasonography,
  NOS">
  <ATTR NAME="Procedure" TYPE="SNOMED V3.3" VALUE="P5-B0110">
  <ATTR NAME="Procedure" TYPE="Text" VALUE="Invasive medical procedure, NOS">
  <ATTR NAME="Procedure" TYPE="SNOMED V3.3" VALUE="P2-00020"> ...
</GROUP>
<GROUP NAME="ListOfDiagnosis" MATCH=,no">
  <ATTR NAME="Diagnosis" TYPE="Text" VALUE="CELLULAR AND SUBCELLULAR
  ABNORMALITIES">
  <ATTR NAME="Diagnosis" TYPE="SNOMED V3.3" VALUE="M-20000">
  <ATTR NAME="Diagnosis" TYPE="ICD-10" VALUE="759.9">
  <ATTR NAME="Diagnosis" TYPE="Text" VALUE="CONGENITAL ANOMALIES">
  <ATTR NAME="Diagnosis" TYPE="SNOMED V3.3" VALUE="M-60000">
  <ATTR NAME="Diagnosis" TYPE="ICD-10" VALUE="759.9"> ...
</GROUP>
</MEDICALCONTEXT>

```

3.6.7 Beispiel einer Transformation

Wir verwenden die in Abschnitt 4.7.4 und 4.7.3 angegebenen Beispiele für medizinische Kontexte der Fetalautopsie md_1 und des Pränatalultraschalls md_2 , wobei hier die Inhalte und die Schreibweise der Einfachheit halber nur verkürzt wiedergegeben werden.

$$\begin{aligned}
 md_1 &= ((Patient="female"; MaxAge="50"; MinAge="10") \\
 &\quad (Organe="Placenta"; Organe="Fetus"; Organe="Uterus")) \\
 &\quad ((Patient="Fetus"; MaxAge="45 WoG"; MinAge="10 WoG") \\
 &\quad (Organe="don't care")) \\
 md_2 &= ((Patient="female"; MaxAge="10"; MinAge="50") \\
 &\quad (Organe="Placenta"; Organe="Fetus"; Organe="Uterus"))
 \end{aligned}$$

Ein Befund x aus der Fetalautopsie soll in der pränatalen Diagnostik angezeigt werden:

$$\begin{aligned}
 x &= ((Patient="fetus"; Age="24 WoG") \\
 &\quad (Organe="Lung"; Diagnosis="Q33.6"))
 \end{aligned}$$

Es gilt nun, eine Transformation f durchzuführen:

$$x' = f(x, md_1, md_2)$$

Zunächst wird die Schnittmenge berechnet:

$$\begin{aligned}
 md = md_1 \cap md_2 &= ((Patient="female"; MaxAge="50"; MinAge="10") \\
 &\quad Organe="Fetus"; MaxAge="45 WoG"; MinAge="10 WoG")
 \end{aligned}$$

Da die Schnittmenge nicht leer ist, kann der Pränatalultraschallknoten davon ausgehen, daß die in md enthaltenen Objekte mit seinen eigenen Methoden behandelt werden können. Das sind die Einträge $Patient="female"$ und $Organe="Fetus"$. Anschließend wird x' aus x gefiltert:

$$\begin{aligned}
 x' &= ((Patient="fetus"; Age="24 WoG") \\
 &\quad (Organe="Lung"; Diagnosis="Q33.6"))
 \end{aligned}$$

Das Objekt x wird im Pränatalultraschallkontext vollständig übernommen. Eine Umkehrtransformation

$$x'' = f(x', md_2, md_1) = x$$

würde wieder x liefern. Dieses Beispiel funktioniert jedoch nur, wenn die implizite Festlegung gilt, daß beim Matching ein Objekt *Organe* mit einem Objekt *Patient* gematcht werden kann. Dies ist zwar semantisch korrekt, da ein zu behandelndes Organ der ganze Körper sein kann, ist formal ohne die implizite Festlegung jedoch falsch. In diesem Fall würde sich eine leere Schnittmenge ergeben und das Empfängersystem Pränatalultraschall kann sich entscheiden, ob die Daten jetzt als reiner Text dargestellt werden sollen, falls ja, würde die Transformation f als Ergebnis x' wie folgt liefern:

```
x' = (AttrType= „Text“; AttrValue=“Patient = Fetus“;  
AttrType= „Text“; AttrValue=“Age = 34 WoG“;  
AttrType= „Text“; AttrValue=“Organe = Lung“;  
AttrType= „Text“; AttrValue=“Diagnosis = Q33.6“)
```

Das Objekt x' wird dann mit den Standardattributen vom Typ *Text* dargestellt. Für eine maschinelle Weiterverarbeitung mag diese Reintextform nicht geeignet sein, für einen Arzt enthält dieser Text genauso viel Inhalt wie die codierte Form.

3.7 Der virtuelle Namensraum

Das Finden eines Objekts läuft im virtuellen Befundsystem an Hand der globalen ID ab (s. Abschnitt 4.4.1 *Objekt-Identifikation*). Der Referenzknoten verwaltet die im virtuellen Befundsystem vertretenen Subsysteme, die mit der System-ID eineindeutig identifiziert werden können. Der Referenzknoten verwendet das DNS-Konzept, um die Auflösung nach Netzwerkadresse durchzuführen bzw. den Namen eines Objekts an Hand der Adresse herauszufinden.

Der vom virtuellen Befundsystem bekannte Namensraum ist somit eine echte Unter-
menge des Internet-Namensraums. Sind im Internet mehrere virtuelle Befundsysteme
installiert, so braucht der Administrator nur bei dem jeweiligen Referenzknoten die
Adresse des Partners - also des Namenservers - eingetragen zu haben. Sofort steht
allen Anwendern eines Namensraums der gesamte Namensraum des Partners zur
Verfügung.

Die Schutzmechanismen für die virtuellen Namensräume basieren hierbei
vollständig auf den Schutzmechanismen des Internet, wie Firewall, sowie auf IPV6
basierende virtuelle Netze.

3.8 Das Transaktionskonzept

Das Transaktionskonzept wird sehr einfach gestaltet. Es ist nur eine Hierarchietiefe
von 1 erlaubt. Es gibt keine verschachtelte Transaktion. Eine Transaktion wird mit
BeginTransaction gestartet und mit EndTransaction beendet. Bei Fehlern wird Roll-
backTransaction aufgerufen. Bei jedem Start einer Transaktion wird festgelegt, ob
und wie lange der Sender auf die Antwort warten soll (synchron/asynchron und
Timeout). Bricht eine Verbindung während einer synchronen Transaktion ab, so wird

die Transaktion auch abgebrochen. Es wird dann an keinem Subsystem eine Veränderung vorgenommen.

4 Ergebnisse

Das Ergebnis der vorliegenden Arbeit ist ein Konzept für den Standard zum Austausch medizinischer Informationen, zunächst im Rahmen eines Intranet. Dieses Konzept wird im folgenden AMICI genannt (**A**rchitecture for **M**edical Information Exchange and **C**ommunication Interface).

Das Konzept wurde aus der im Kapitel 3 *Methodik* festgelegten Strategie bzw. Philosophie entwickelt. Das entwickelte Konzept wurde mit Hilfe ausgewählter Subsysteme überprüft. Die Auswahl erfolgt nach folgenden Kriterien:

- Die Subsysteme sollen auf so verschiedenen, in der Charité vertretenen Plattformen wie möglich laufen (z. B. DOS, Win95, Win NT, Novell, etc).
- Sie sollen so unterschiedliche Systemarchitekturen wie möglich aufweisen (Stand-alone, Client/Server, etc.).
- Es sollen so viele Informationen wie möglich über den internen Aufbau und die Arbeitsweise verfügbar sein.
- Die Ergebnisse der Arbeiten sollen sofort für den ersten Realisierungsschritt genutzt werden können.

Die Auswahl fiel auf folgende Subsysteme mit den in Tabelle 4-1 angegebenen Merkmalen:

Tabelle 4-1 Merkmale der untersuchten Subsysteme

Name	Inhalt	Anwender	Plattform	Architektur
Punktions-datei	invasive Untersuchungen in der pränatalen Diagnostik	Pränatale Diagnostik und Therapie	DOS	FoxBase-DB Single User
GebDat	geburtshilfliche Daten	Kreißsaal	Novell/DOS	dBase-DB Client/Server Multi User
PIA	Pränatal-Ultraschall, Diagnose, Bilder	Pränatale Diagnostik und Therapie	Novell/DOS	Watcom DB Client/Server Multi User
Autopsy	Autopsie-Protokolle, Bilder	Fetalpathologie	Win3.1/Win95	Access-DB Multi User
HL7-Server	Patienten-Stammdaten	Tumorzentrum	Win NT	SyBase-DB Client/Server Multi User HTML

Die Untersuchung dieser unterschiedlichen Subsysteme zeigte, daß ein Standard für die Kommunikation zwischen den Einrichtungen in einem größeren Krankenhaus wie der Charité vorgeschlagen werden kann. Er ist technisch ohne weiteres realisierbar. Organisatorisch jedoch muß jede medizinische Einrichtung einige Maßnahmen ergreifen. Wichtige Voraussetzungen hierfür sind erstens die Möglichkeit, auf die von

einem kommerziell erworbenen Subsystem verwalteten Daten zuzugreifen und zweitens müssen diese Daten in die vom Konzept vorgeschlagene Semantik umgesetzt werden. Dies kann erreicht werden, indem ein entsprechender virtueller Befundtreiber, ein sogenannter Wrapper, erstellt wird.

Im folgenden wird zunächst der Systementwurf mit der Software- und Datenstruktur beschrieben. Hierbei wird auf eine umfassende Darstellung verzichtet. Es wird sich vielmehr auf folgende Kernpunkte des Konzepts konzentriert:

- die allgemeine Struktur und der allgemeine Ablauf des Systems,
- die Anwendungen des allgemeinen Objektkonzepts,
- die Anwendungen des medizinischen Kontextes,
- die Behandlung der weltweit gültigen Identifikationen und
- die Behandlung von Attributen durch die Attribute-Engine.

Anschließend wird ein Überblick über die o. g. Subsysteme gegeben, wobei deren architektonischen Merkmale sowie die Art, die Objekte zwischen dem Subsystem und dem AMICI-System zu synchronisieren, im Vordergrund stehen. Im Anschluß hieran wird die Testimplementation beschrieben.

4.1 Systementwurf

Der Systementwurf wird durch folgende Punkte beschrieben:

- Softwarestruktur,
- Programmablauf und
- Datenentwurf.

4.1.1 Softwarestruktur

Virtuelles Befundsystem

Es wird ein Satz von Software-Werkzeugen entwickelt, die die *virtuellen Behandlungen* (siehe unten) gemäß den o. g. Anforderungen bearbeiten. Diese Werkzeuge bilden ein virtuelles Befundsystem, das den Charakter eines verteilten, multilateralen und dual-funktionalen Systems hat. Die Dualfunktionalität rührt daher, daß jeder Knoten den anderen sowohl als Client als auch als Server dienen kann. Das *virtuelle Befundsystem* läßt sich auf einer beliebigen Anzahl von Rechnern installieren. Jede Installation korrespondiert mit einem realen medizinischen Subsystem und liefert einen Beitrag zu einem gesamten virtuellen Befund. Alle Installationen kommunizieren miteinander über ein definiertes Protokoll.

Das System bildet somit ein Netz, bestehend aus Knoten. Jeder Knoten repräsentiert ein Subsystem. Der Knoten stellt ein Subsystem aus der Sicht des *virtuellen Befundsystems* (siehe unten) dar. Er stellt die Daten des Subsystems dem *virtuellen Befundsystem* zur Verfügung und gibt die Daten aus dem virtuellen Befundsystem an das angeschlossene Subsystem weiter. Der Knoten kommuniziert nicht direkt mit dem Subsystem. Er bedient sich eines *virtuellen Befundviewers* (siehe unten).

Die Knoten unterhalten Verbindungen untereinander. Die Subsysteme haben dagegen keine direkte Verbindung zueinander. Die Menge aller Knoten bildet das *virtuelle Befundsystem*. In Abbildung 4-1 wird das *virtuelle Befundsystem* grau hin-

terlegt. Die Knoten sind in der Regel auf einem Anwendungsserver (Unix bzw. Windows NT) installiert. Die Knotensoftware ist in der Regel untereinander identisch. Sie wird nur einmal entwickelt und mehrfach installiert. Lediglich durch die Einstellung der Parameter im medizinischen Kontext unterscheiden sie sich. Software-technisch besteht jeder Knoten aus einem WWW-Server, der einerseits Verbindung zum Browser unterhält, andererseits einen Satz von Applets aktiviert, die den virtuellen Befundviewer ansprechen. Abbildung 4-2 zeigt die schematische software-technische Konstruktion eines Knotens und den Ablauf einer Anfrage vom Browser.

Allgemeine *virtuelle Befundviewer*, die nicht an ein spezielles Subsystem angebunden sind, werden über einen allgemeinen Knoten bedient. Der allgemeine Knoten ist genauso wie jeder andere Knoten konstruiert. Allein der dazugehörige medizinische Kontext - in diesem Fall ist es ein allgemeiner medizinischer Kontext - macht ihn zu einem allgemeinen Knoten. Die Software für einen allgemeinen Knoten muß nicht neu entwickelt werden. Sie ist die Standardknotensoftware. Der virtuelle Befundviewer könnte dann überall die allgemein verfügbare Browser-Software mit einer allgemeinen Startseite sein.

In einem virtuellen Befundsystem existiert mindestens ein Knoten, der die Aufgaben eines Bezugsknotens übernimmt. Dieser Knoten heißt Referenzknoten. Ähnlich wie beim allgemeinen Knoten wird für den Referenzknoten die gleiche Software verwendet. Der Referenz-Client wird in der Regel benutzt, um das System zu installieren und administrieren. Für den Referenzknoten wird ein gesonderter medizinischer Kontext definiert.

Virtueller Befundviewer

Es wird ferner ein Darstellungsprotokoll festgelegt, das auf dem HTML oder gleichwertigem Protokoll basiert, so daß an jedem Arbeitsplatz ein *virtueller Befund* angezeigt werden kann, an dem ein HTML-fähiger Viewer verfügbar ist. In der Regel wird das ein kommerzieller Internet-Browser sein. In Abbildung 4-1 werden die *virtuellen Befundviewer* in der ersten Ebene dargestellt.

Virtueller Befundtreiber

Die medizinischen Subsysteme werden jeweils über einen sogenannten virtuellen Befundtreiber an das virtuelle Befundsystem angeschlossen. Die Differenzen zwischen dem realen medizinischen Subsystem - dem realen Befundsystem - und dem virtuellen Befundsystem werden durch den virtuellen Befundtreiber überbrückt. In Abbildung 4-1 werden die *virtuellen Befundtreiber* in der dritten Ebene dargestellt.

In der ersten Version besteht das System aus folgenden Knoten:

- der invasive Knoten,
- der Pränatal-Ultraschall-Knoten,
- der geburtshilfliche Knoten,
- der Fetalautopsie-Knoten,
- der HL7-Knoten,
- der allgemeine Knoten und
- der Referenzknoten

Jeder Knoten repräsentiert ein Subsystem. Die ersten fünf Knoten sind durch die vorhandenen Subsysteme begründet. Der allgemeine Knoten ist ein besonderer

Knoten, da er nicht auf vorhandenen Subsystemen basiert. Dieser Knoten wird geschaffen, um einen allgemeinen Viewer bieten zu können, ohne daß der Anwender sich auf einen medizinischen Kontext festlegen muß.

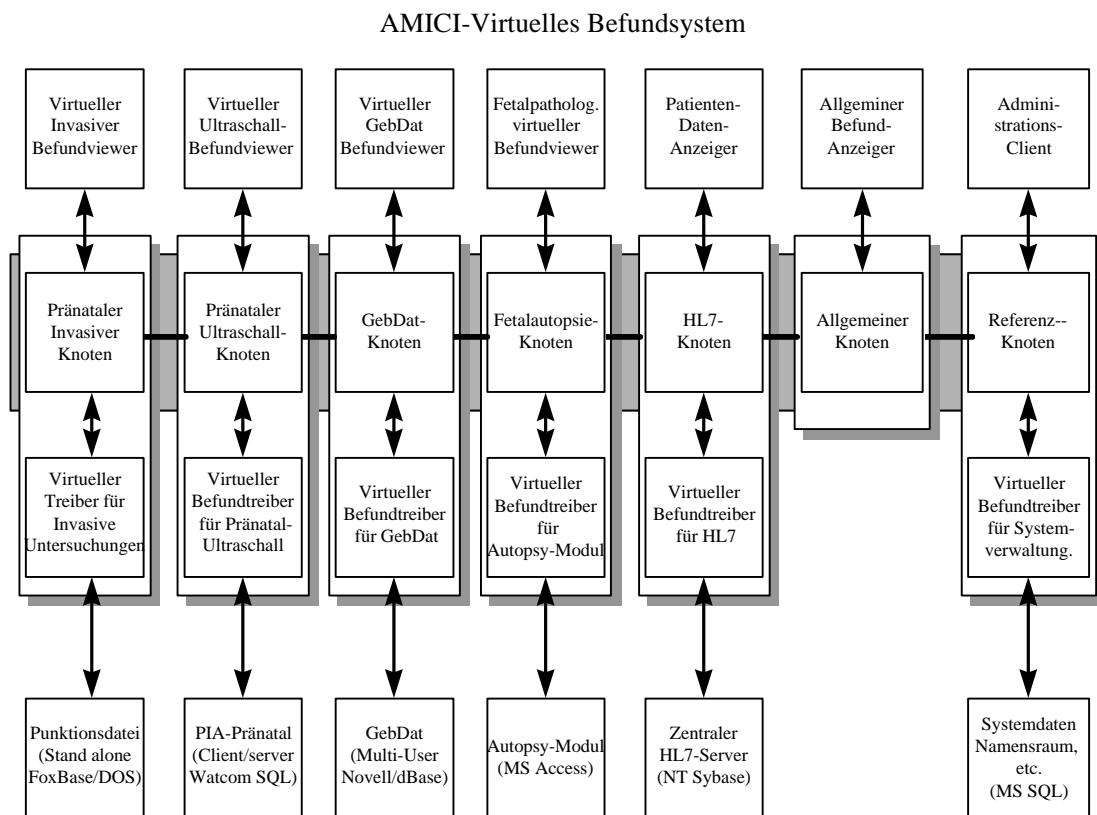


Abbildung 4-1 Softwarestruktur des Systems

Ein weiterer besonderer Knoten ist der Referenzknoten. Dieser Knoten wird geschaffen, um die Einrichtung, die Installation, den Betrieb, die Administration und die Wartung des Systems vornehmen zu können. Hier werden die Systemdaten wie der Namensraum, die medizinischen Kontexte, die Standardattribute mit den Standardmethoden für alle anderen Knoten bereitgestellt.

4.1.2 Programmablauf

Abbildung 4-2 zeigt den schematischen Programmablauf an einem Knoten.

Anfrage vom Client

Jeder Knoten hat in der Regel zwei Zugänge für die Clients.

Ein Standard-Client greift über den HTML/XML-Zugang auf einen WWW-Server zu, der je nach ausgewählter Startseite die gewünschte Funktionalität liefert. Spezielle Anwendungen, für die die jetzige Funktionalität von HTML bzw. die Geschwindigkeit von Standard-WWW-Servern und Internet-Browsern nicht ausreichen, können direkt auf einen Knoten zugreifen. Dies kann mit einem C/C++- oder einem beliebigen Client geschehen.

Die speziellen Clients greifen über die Communication-Layer auf das Knoteninterface zu, das die gesamte Funktionalität eines Knotens bietet, während die Standard-HTML-Clients indirekt über den WWW-Server mit den WWW-Applets auf das Knoteninterface zugreifen.

AMICI-Knoten

Die Anfragen beider Arten von Clients sind in Form von Transaktionen formuliert. Sie werden vom Knoteninterface bearbeitet, das die Aufgaben mittels eines Transaktions-Dispatchers verteilt. Der Dispatcher untersucht die Anfragen und ruft die entsprechenden Objektmethoden mittels eines Object-Handlers auf, um die Anfragen zu bearbeiten.

Treiber und interne Kommunikation

Der Object-Handler verteilt wiederum die ihm zugewiesenen Transaktionen auf den virtuellen Befundtreiber und/oder an die weiteren Knoten. Er sammelt auch deren Antworten und setzt die Attribute-Engine ein, um sie in die richtige Form zu bringen. Diese ruft wiederum das Device Independent Interface (DII) auf, um die von der Attribute-Engine im neutralen Format ausgegebenen Daten in das gewünschte Ausgabeformat zurückzugeben. Zu diesem Zweck bedient sich das DII entsprechender Ausgabetreiber. In diesem Fall ist es ein HTML/XML-Ausgabetreiber. Für andere Ausgabeformate wie HL7, DICOM, etc. lassen sich mit vertretbarem Aufwand entsprechende Treiber entwickeln.

Antwort an den Client

Um andere Formate zu erzeugen, muß das DII nur den richtigen Treiber laden. Beispielsweise kann dies eine TIFF-Ausgabe oder ein Report in RTF- bzw. WinWord-Format sein. Im Fall von HTML-Clients werden die Ergebnisse an den WWW-Server zurückgegeben, der die HTML-Seite an den Internet-Browser weitergibt.

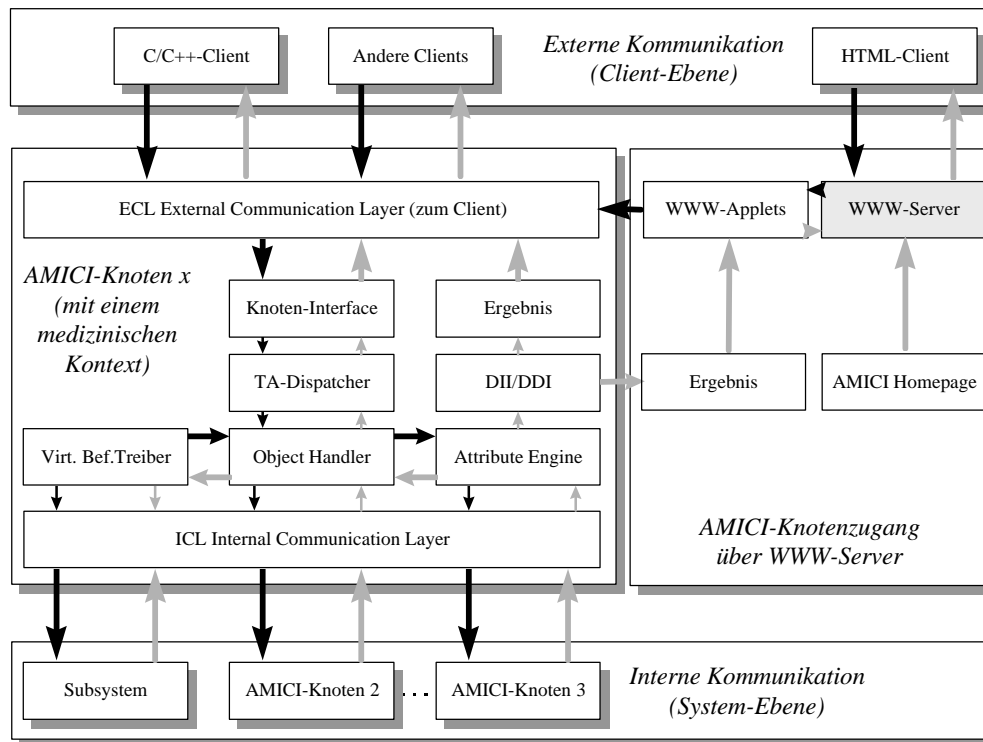


Abbildung 4-2 Schematische Darstellung des Ablaufes einer Anfrage

4.2 Softwarekomponenten

Im folgenden werden die Funktionalität und der wesentliche interne Ablauf der Softwarekomponenten beschrieben. Zunächst werden zwei Teilkonzepte für die Realisierung erläutert, die für alle Komponenten gelten. Das sind:

- die Art, wie die Komponenten miteinander kommunizieren und
- das Treiber-Konzept.

4.2.1 Allgemeine Kommunikation zwischen den Komponenten

AMICI besteht aus mehreren Komponenten, die ständig miteinander kommunizieren. Sie laufen in der Regel auf mehreren Rechnern in mehreren Prozessen und ggf. mehreren Threads. Der Mechanismus für diese Kommunikation wird auf RPC festgelegt. Die Semantik der Kommunikation erfolgt stets nach einem Schema. Eine Komponente bietet ihre Funktionalität über ein sogenanntes API: Application Programming Interface. Das o. g. Schema läuft im API wie folgt ab:

1. Erfragen der verfügbaren Anbieter.
2. Festlegen des gewünschten Kommunikationspartners.
3. Aufbauen einer Verbindung.
4. Erfragen und Festlegen der gewünschten Funktionalität.
5. Ausführen der Arbeiten.
6. Beenden der Verbindung.

Im Abschnitt 4.2.3 *Der virtuelle Befundviewer (VRV)* wird dieses Konzept beispielhaft beschrieben.

4.2.2 Das Treiber-Konzept

Um die Software plattformunabhängig zu gestalten, werden die Teile der Software in Treiber ausgelagert, die möglicherweise von der Hardware, von der Systemsoftware bzw. von bestimmten Standards abhängig sein können. Beispielsweise wird das Communication-Layer eingeführt, um die Abhängigkeit von einem bestimmten Protokoll zu eliminieren. Wird ein neues Protokoll benötigt, so wird einfach ein neuer Treiber erstellt, der dieses neue Protokoll an das Communication-Layer anpaßt. Diese Vorgehensweise ist bereits in der Softwareentwicklung erprobt und birgt daher kaum Risiko für die Realisierung.

4.2.3 Der virtuelle Befundviewer (VRV)

Der virtuelle Befundviewer (VRV: Virtual Report Viewer) ist ein Client-Programm, das in der Lage ist, die Anfrage vom Anwender entgegenzunehmen, sie in die virtuelle Anfrage umzuwandeln und die Antwort vom virtuellen Befundsystem in Form eines virtuellen Befundes korrekt anzuzeigen.

In der Regel ist der VRV ein regulärer Internet-Browser. In diesem Fall nimmt der VRV Kontakt mit einem WWW-Server auf und startet die Anwendung durch das Laden einer geeigneten Homepage. In dieser Homepage sind Verweise auf Applets enthalten, die als Java-Applets oder ActiveX-Elemente realisiert sind. Diese greifen wiederum über die AMICI-Client-API auf das External-Communication-Layer (ECL, siehe unten) zu, um die Funktionalität des Knoteninterfaces zu nutzen.

Der VRV kann ebenfalls ein gewöhnliches in C/C++ geschriebenes Client-Programm sein, das die Funktionalität des Knoteninterfaces nutzt, um die virtuellen Befunde aus einer virtuellen Behandlung zu holen und anzuzeigen. Es kann aber auch jedes beliebige Stück Software sein, das in der Lage ist, die o. g. Arbeiten durchzuführen. Diese Clients greifen direkt auf die AMICI-Client-API zu, ohne den Umweg über WWW-Server und Applets zu nehmen. Diese Hybridlösung bietet den Vorteil, daß alle Gelegenheitsanwender über die plattformunabhängige Browser-Technik Zugriff auf alle Anwendungen bekommen. Für Anwender, die AMICI in der Routine einsetzen, werden direkte C/C++-Lösungen angeboten, die z. Z. erhebliche Geschwindigkeitsvorteile bringen.

Die Bedienoberfläche eines solchen Clients läßt sich beispielsweise leichter an die Bedürfnisse der Anwender anpassen. Die meisten Browser lassen heutzutage zwar ihre Funktionalität durch sogenannte Plug-Ins erweitern, eine grundsätzlich andere Bedienoberfläche ist in der Regel jedoch nicht ohne weiteres möglich.

Abbildung 4-3 erläutert die Möglichkeiten eines virtuellen Befundviewers, auf das Knoteninterface zuzugreifen. Alle Clients bedienen sich der AMICI-Client-API. Der WWW-Server kann auf dem gleichen Server, auf dem die AMICI-Knotensoftware installiert ist oder auf einer getrennten Maschine laufen. Ein WWW-Server kann ferner durch die Installation mehrerer Homepages als Zugang für mehrere Subsysteme eingesetzt werden.

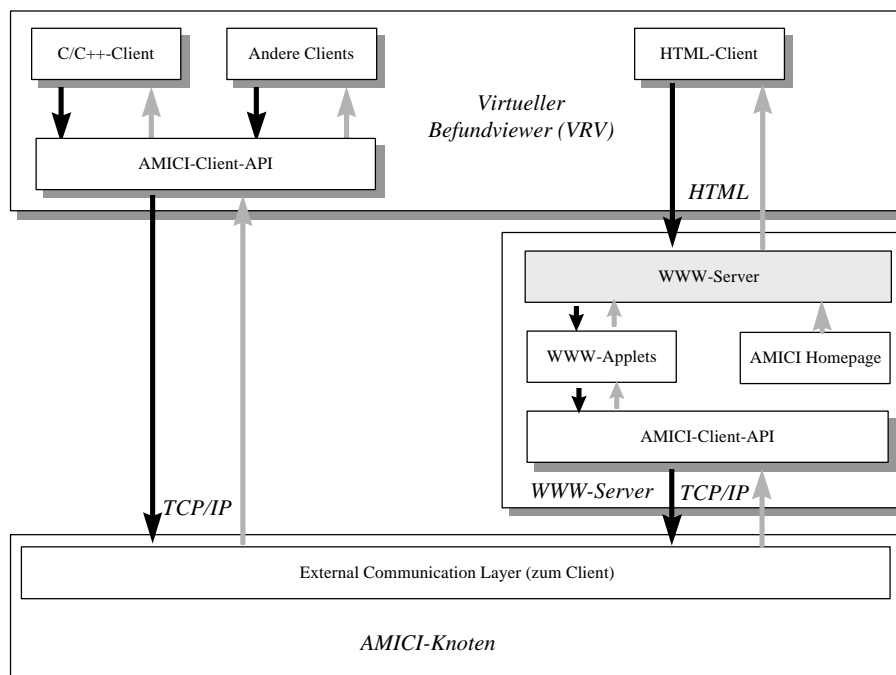


Abbildung 4-3 Zugang vom virtuellen Befundviewer zum AMICI-Knoten

4.2.4 External Communication-Layer (ECL)

Die Aufgabe des External Communication-Layer (ECL) ist es, wie der Name sagt, die Verbindung zwischen den Clients und dem Knoteninterface herzustellen und zu unterhalten.

Dieses Layer läuft auf einem Windows NT Server ab Version 4.0 und wird als Service automatisch mit einer für AMICI vordefinierten Servicenummer (z. B. 3010) gestartet. Es startet automatisch das Knoteninterface, das wiederum alle anderen notwendigen Komponenten nacheinander startet.

Dieses Layer bedient in erster Linie die TCP/IP-Anfragen von Clients und lässt sich später bei Bedarf auf andere Protokolle wie IPX/SPX, AppleTalk oder DecNet erweitern. Es ruft die hardware-unabhängige Win32bit-API für Netzwerk *WNETxx* auf, um die Netzwerkfunktionen zu benutzen. Somit ist das ECL nicht von bestimmter Hardware abhängig.

Bei einer Anfrage eines neuen Clients wird eine neue virtuelle Verbindung angelegt und eine neue Thread im Knoteninterface gestartet, die für diese Verbindung zuständig ist. Die Clients bedienen sich des ECL-API, um auf die Funktionalität eines Knotens zuzugreifen.

4.2.5 Das Knoteninterface (NI)

Das Knoteninterface (NI: Node Interface) unterhält die Verbindung zum External-Communication-Layer. Für jede Verbindung wird ein Thread vom Knoteninterface

gestartet. Das Knoteninterface arbeitet unabhängig von der Art, wie ein Client gebaut ist. Objekte, die zwischen dem ECL und dem NI ausgetauscht werden, sind Transaktionen der obersten Ebene. Das sind Transaktionen, die auf die „top-level“-Objekte vom Typ virtuelle Behandlung angewandt werden.

Die Hauptaufgabe des Knoteninterfaces ist die korrekte Handhabung dieser Transaktionen.

4.2.6 Der Transaktionsverteiler (TAD)

Der Transaktionsverteiler (TAD: Transaction Dispatcher) verteilt die vom Knoteninterface erhaltenen Transaktionen auf die Objekte der virtuellen Behandlungen, die durch den Object-Handler behandelt werden. Der TAD leitet bei jeder Anfrage eine *BeginTransaction* und bei Beendigung der Anfrage eine *EndTransaction* ein. Alle Aufrufe an den Object-Handler werden zunächst „probeweise“ ausgeführt. Ist die Transaktion erfolgreich zu Ende gelaufen, so wird durch *EndTransaction* dieser neue Zustand festgeschrieben. Ist dies nicht der Fall, so leitet der TAD ein *RollBack* ein. Hierbei kann beispielsweise der vor *BeginTransaction* gesicherte Zustand wiederhergestellt werden. Man kann dies auch durch sogenannte *Undos* realisieren. Ein *CreateObject*-Aufruf wird durch die Funktion *DeleteObject* wieder rückgängig gemacht.

4.2.7 Der Object-Handler (OH)

Der Object-Handler bietet in der OH-API einen Rahmen für alle Pflichtmethoden der Objekte. Auf dieser Ebene sind alle Funktionen nicht auf ein spezielles Objekt ausgerichtet. Intern splittet OH die in der API übergebenen Objektmethoden einerseits durch Aufrufe an Unterobjekte, die im „top-level“-Objekt enthalten sind, andererseits durch Aufrufe an die Attribute-Engine, die die im Objekt enthaltenen Attribute behandelt. Die Ergebnisse der Attribute-Engine werden entweder im Speicher oder in externen Dateien oder durch einen Datenstrom zurückgegeben.

4.2.8 Die Attribute-Engine (AE)

Die Attribute-Engine wird aufgerufen, wenn der Object-Handler (OH) ein Attribut behandeln muß; z. B. wenn OH die Aufgabe bekommt, ein Attribut darzustellen, ruft OH die Attribute-Engine auf. Diese untersucht, ob das Attribut ein Primitivum ist, wenn ja, ruft sie die Funktion *DisplayStdAttr* auf. Hierbei wird der Display-Context ausgewertet, um eine korrekte Darstellung des Attributs zu gewährleisten. Ist das Attribut kein Primitivum, so ruft die Attribute-Engine das betroffene Objekt mit der Objektmethode *DisplayObject* auf. Das Objekt kennt nun seine Struktur am besten und fängt wiederum an, seine eigenen Attribute darzustellen. Das Objekt ruft die Attribute-Engine wieder auf, falls es dabei auf die Primitiva trifft. Abbildung 4-4 zeigt das Zusammenspiel zwischen dem Object-Handler und der Attribute-Engine.

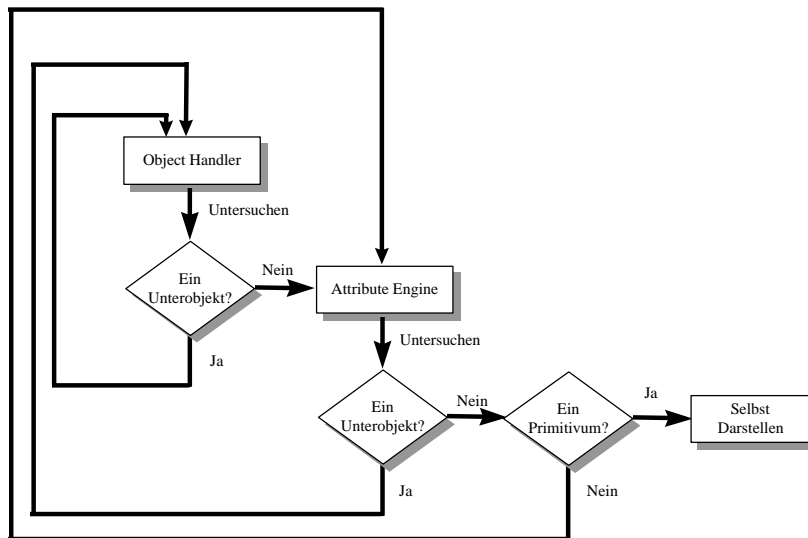


Abbildung 4-4 Zusammenspiel zwischen Object-Handler und Attribute-Engine

4.2.9 Das Device-Independent-Interface (DII)

Diese Schnittstelle sorgt für eine geräteunabhängige Ausgabe des Ergebnisses einer Transaktion. Dieses Interface arbeitet gemäß dem allgemeinen Treiber-Konzept (siehe Abschnitt 4.2.2 *Das Treiber-Konzept*). Ein Beispiel für die Realisierung findet sich im Abschnitt 4.2.11 *Der virtuelle Befundtreiber (VRD)*.

4.2.10 Das Device-Dependent-Interface (DDI)

Hier sind die Ausgabetreiber angesiedelt, die konkrete Ausgabeoperationen auf einem Gerät vornehmen. Diese Treiber arbeiten gemäß dem allgemeinen Treiber-Konzept (siehe Abschnitt 4.2.2 *Das Treiber-Konzept*). Ein Beispiel für die Realisierung findet sich im Abschnitt 4.2.11 *Der virtuelle Befundtreiber (VRD)*.

4.2.11 Der virtuelle Befundtreiber (VRD)

Ein virtueller Befundtreiber (VRD: Virtual Report Driver) ist das Bindeglied zwischen dem medizinischen Subsystem und dem virtuellen Befundsystem. Er muß in der Lage sein, die Anfrage vom virtuellen Befundsystem entgegenzunehmen, sie in das subsystem-spezifische Format umzuwandeln und an das Subsystem weiterzuleiten. Die Antwort vom Subsystem muß der virtuelle Befundtreiber wieder in das Format des virtuellen Befunds zurückverwandeln und an das virtuelle Befundsystem abliefern.

Der Einsatz des VRD erfolgt nach einem Layer-Modell, das aus folgenden Layern besteht:

- Das API-Layer

Dieses Layer stellt den VRD als ein logisches Gerat dar. Alle hier angebotenen Funktionen sind nicht gerateabhangig.

- Das Treiber-Layer
Dieses Layer ist das Verbindungsstuck zwischen dem gerate-unabhangigen API-Layer und dem Gerat bzw. den Lowlevel-Treibern. Es setzt die von dem API-Layer geforderten Funktionen in die gerateabhangigen Aufrufe um.
- Das VRD-Lowlevel-Layer
Die zu diesem Layer gehorenden Komponenten werden von Thirdparty-Herstellern geliefert. Es handelt sich um Datenbanktreiber (ODBC) und/oder Netzwerktreiber.

Im folgenden wird der Rumpf eines VRD-Treibers als Vorlage fur die Realisierung festgelegt. Die in der Vorlage noch fehlenden Funktionen werden nach diesem Muster realisiert.

```

THRESULT      vrdControl      (THDC      hDC,
                             int          iFunction,
                             int          iLenInData,
                             LPVOID*     lpInData,
                             LPVOID*     lpOutData)
{
    THRESULT      status;

    if (hDC < (THDC) NULL && iFunction != DRVFUNC_INIT)
        return (ER_UNDEFINEDERROR);

    status = 0;

    switch (iFunction)
    {
        case DRVFUNC_INIT:
        {
            LPCSTR  lpOutput;      // Output port or File name //
            LPCSTR  lpDriver;      // Driver name //
            int      iOpMode;      // Operation mode //

            lpDriver = lpInData [0];
            lpOutput = lpInData [1];
            iOpMode = (int) *((int *) lpInData [2]);
            status = vrdInitDevice (iOpMode, lpOutput);
            if (status < 0)
                return (ER_UNDEFINEDERROR);
        }
        break;
        case DRVFUNC_DELETE:
            vrdCloseDevice (hDC);
            break;
        case DRVFUNC_GETDEVCAPS:
            status = vrdInitDevCaps ((LPDOCIOCAPS) &CurrDocIOCaps);
            if (status < 0)
                return (ER_UNDEFINEDERROR);
            memcpy ((LPBYTE) lpOutData [0], (LPBYTE) &CurrDocIOCaps,
                sizeof (VRCDEVCAPS));
            break;
        case DRVFUNC_SETABORTPROC:
        {
            LPFNABORTPROC  lpfnAbortProc;
            lpfnAbortProc = lpOutData [0];
            status = vrdSetAbortProc (hDC, lpfnAbortProc);
            if (status < 0)
                return (ER_UNDEFINEDERROR);
        }
        break;
        case DRVFUNC_GETDEVICECAPINT:
        {
            int      iIndex;
            LPINT    lpiDevCaps;
            iIndex = (int) *((LPINT) lpInData [0]);

```



```

        lpDriver = lpOutData [0];
        status = vrdGetDeviceCapInt (hDC, iIndex, lpiDevCaps);
        if (status < 0)
            return (ER_UNDEFINDEDERROR);
    }
    break;
case DRVFUNC_GETDEVICECAPLONG:
    {
        int          iIndex;
        LPLONG       lpiDevCaps
        iIndex      = (LONG) *((LPLONG) lpInData [0]);
        lpDriver    = lpOutData [0];
        sttaus      = vrdGetDeviceCapLong (hDC, iIndex, lpiDevCaps);
        if (status < 0)
            return (ER_UNDEFINDEDERROR);
    }
    break;
case DRVFUNC_GETDEVICECAPDOUBLE:
    {
        int          iIndex;
        LPDOUBLE     lpiDevCaps
        iIndex      = (double) *((LPDOUBLE) lpInData [0]);
        lpDriver    = lpOutData [0];
        sttaus      = vrdGetDeviceCapDouble (hDC, iIndex, lpiDevCaps);
        if (status < 0)
            return (ER_UNDEFINDEDERROR);
    }
    break;
default:
    return (ER_UNDEFINDEDERROR);
    break;
}
return (status);
}

```

4.3 Datenentwurf

Im folgenden wird der logische Datenentwurf beschrieben. Auf die Darstellung der physikalischen Datenstruktur wird hier verzichtet, da sie nicht wesentlich zum Verständnis des Konzepts beiträgt.

Die AMICI-Daten werden im wesentlichen als relationale Datenbanken (RDB) abgelegt. Hier wird zwischen den AMICI-Daten selbst und den Daten der angeschlossenen medizinischen Subsysteme unterschieden. Die AMICI-Daten sind in erster Linie die Daten der virtuellen Behandlungen, die nur temporär existieren. Sie werden für die Dauer einer Transaktion aus den Daten der Subsysteme erzeugt und nach Beendigung der Aufgaben wieder gelöscht. Außerdem benötigt AMICI auch eigene Daten, die weder vom Subsystem noch vom Client geliefert werden können. Es sind dies beispielsweise die Systemdaten wie Anwender, Namensraum, Subsysteme, medizinische Kontexte, etc.

Da es möglich sein muß, von beliebigen Frontends über ODBC auf die AMICI-Datenbanken zuzugreifen und die Daten jederzeit in ein anderes Format zu überführen, können nur Datentypen verwendet werden, die in Standard SQL und ODBC verfügbar bzw. verlustfrei konvertierbar sind.

Alle Standbilddaten werden als einzelne TIFF- bzw. JPEG-Dateien abgelegt. Die Audio-Video-Daten werden als einzelne AVI- bzw. MPEG-Dateien gespeichert. Andere Formate sind durch einfaches Hinzufügen von Treibern jeder Zeit bearbeitbar.

4.4 Das spezielle Objektkonzept

Das im Kapitel 3 *Methodik* beschriebene allgemeine Objektkonzept gibt einen allgemeinen Rahmen, um die medizinischen Informationen zu strukturieren, die im Konzept zwischen den medizinischen Subsystemen ausgetauscht werden sollen. Ein wesentlicher Punkt des allgemeinen Objektkonzepts besagt, daß jedes Objekt eine Menge von Attributen besitzt.

Im Rahmen des allgemeinen Objektkonzepts lassen sich beliebige Objekte definieren. Der Begriff Objekt wird hier weit gefaßt. Ein Objekt kann reell (ein Organ) oder immateriell (ein Befund), sachlich (ein Fetus) oder zeitlich (eine Sektion) oder rein logisch (eine Bedingung) sein. Ein Objekt hat immer eine Objekt-Identifikation. Es kann, muß jedoch nicht weitere Objektattribute besitzen. Zwischen den Objekten können Beziehungen aufgebaut werden. Ein Objekt weiß selbst, was der Anwender mit dem Objekt tun kann. Hauptbestandteile eines Objekts sind:

- Objekt-Identifikation,
- Objektattribute,
- Objektbeziehungen und
- Objektmethoden

Abbildung 4-5 zeigt die allgemeine Struktur eines AMICI-Objekts.

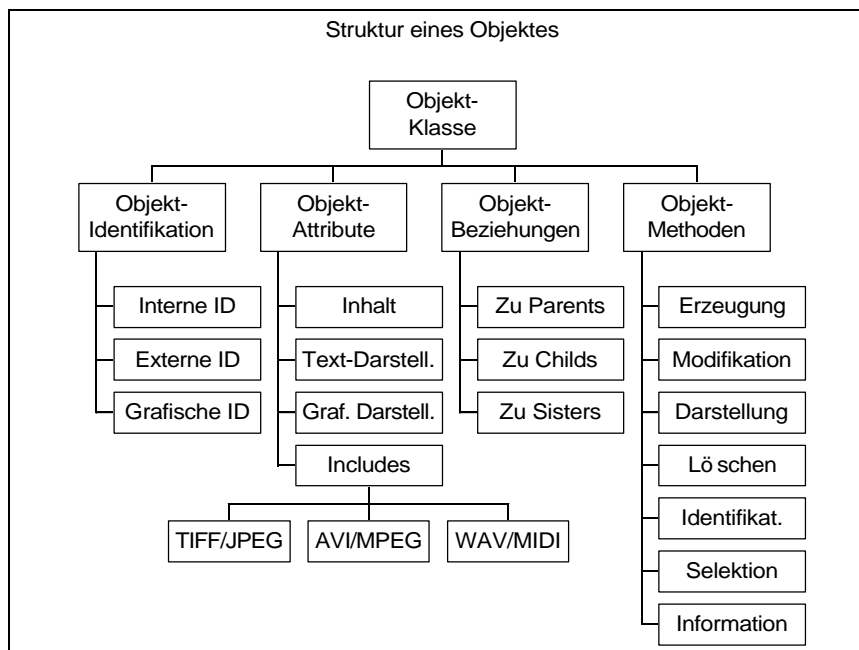


Abbildung 4-5 Allgemeine Struktur eines Objekts

Ferner hat ein Objekt immer einen eindeutigen Zustand. In der Regel befindet sich ein Objekt im Zustand *aktiv*. Ein Objekt läßt sich auch in einen Wartezustand versetzen. Es hat dann den Zustand *passiv*. Weitere Zustände, die diese beiden Hauptzustände näher beschreiben und sich beim Feindesign als notwendig erweisen, lassen sich bei Bedarf hinzufügen. Beispielsweise können für den Zustand *aktiv* folgende Klassifizierungen definiert werden: *unknown*, *initializing*, *intialized*, *closing*, *closed*, *destroying*, *destroyed*. Passive Zustände können sein: *waiting for input*, *waiting for message*. etc.

4.4.1 Objekt-Identifikation

Eine eindeutige Identifikation eines Objekts ist in einer Intranet/Internet-Umgebung essentiell wichtig für das Funktionieren der Anwendungen. Einerseits dürfen sich die ID's eines bekannten Objekts mit der Zeit nicht ändern. Andererseits muß jedes Subsystem die notwendigen Änderungen autark vornehmen können. Aus diesem Grund wird die Identifikation eines Objekts wie folgt realisiert. Jedes Objekt läßt sich auf folgende Art und Weise identifizieren:

- die weltweit eindeutige sogenannte externe ID (globale ID),
- die interne ID (lokale ID) und
- die graphische ID.

Globale Objekt-ID

Die globale Objekt-Identifikation ist ein zusammengesetzter Schlüssel. Sie ist weltweit gültig und besteht aus mehreren Teilen: einer sogenannten lokalen Objekt-ID und einer weltweit einmaligen ID des Subsystems. Die globale Objekt-ID besitzt eine interne und eine externe Darstellung.

Die globale Objekt-ID wird systemintern **global object ID** genannt. Ihre interne Darstellung hat die Form:

```
<internal global object ID> ::=      <internal subsystem ID>,  
                                     <service ID>,  
                                     <local object ID>  
  
<internal subsystem ID> ::=          <virtual IP address>  
<virtual IP address> ::=             byte.byte.byte.byte[.byte]  
<service ID> ::=                    {0|1|2|3|4|5|6|7|8|9}  
<local object ID> ::=                „beliebige Subsystem-bezogene Nummer“
```

Die externe Darstellung einer global Objekt-ID hat die Form:

```
<external global object ID> ::=      <locale object ID>@  
                                     <external subsystem ID>:  
                                     <service ID>  
  
<external subsystem ID> ::=          <external IP address>  
<external IP address > ::=          „IP Adresse gemäß Internet-Namenskonventionen“
```

Beispiele für externe globale Objekt-ID:

```
4711@pia-prenatal.ufk.charite.hu-berlin.de:3030  
47110@pia-patho.ifp.charite.hu-berlin.de:3030  
1174@hl7.rz.charite.hu-berlin.de:3001
```

Anmerkungen:

Die virtuelle IP-Adresse ist in der Regel die IP-Adresse des Servers, auf dem das Subsystem installiert ist. Sind auf einem Server mehrere Subsysteme verfügbar, so werden mehrere virtuelle IP-Adressen vergeben, je eine für ein Subsystem.

Die Service-ID identifiziert den jeweiligen Dienst. Beispielsweise wird festgelegt, daß der Dienst http die Service-ID 8080 zugewiesen bekommt. Für den virtuellen Befunddienst könnte eine noch nicht anderweitig vergebene Nummer verwendet werden, z. B. 4711.

System-ID

Die System-ID ist der Geltungsbereich der Objekt-ID. Die System-ID ist die weltweit eindeutige Identifikation eines installierten Subsystems, gleichgültig, ob es sich dabei um ein medizinisches oder ein administratives Subsystem handelt. Diese ID wird analog zu den IP-Nummern des Protokolls TCP/IP vergeben. Jedes Subsystem erhält eine solche ID. Besteht ein Subsystem physikalisch aus mehreren Komponenten, wobei jede Komponente logisch eine abgetrennte Aufgabe erfüllt, so kann für jede Komponente eine eigene ID zugewiesen werden.

Lokale Objekt-ID

Eine lokale Objekt-ID ist eine Identifikation, die innerhalb eines Subsystems eindeutig ist. Diese ID muß den Internet-Namenskonventionen genügen. Beispielsweise sind „blanks“, Kommata und „underlines“ verboten. Der virtuelle Befundtreiber sorgt für eine richtige Umwandlung zwischen der lokalen Objekt-ID des virtuellen Befundsystems und der internen ID des betroffenen Subsystems.

Die lokale Objekt-ID wird systemintern **local object ID** genannt. Sie wird vom jeweiligen Subsystem vergeben und gilt nur innerhalb des betrachteten Subsystems. Mit dieser Konstruktion wird jede lokale ID zu einer globalen, wenn zusätzlich der Geltungsbereich, also die System-ID, angegeben ist.

Beispielsweise kann dem in der pränatalen Diagnostik eingesetzte Befundsystem PIA eine ID wie *pia-prenatal.ufk.charite.hu-berlin.de* zugewiesen werden. Eine in diesem System gespeicherte Behandlung hat die lokale ID 4711. Die globale ID der Behandlung wäre:

4711 @ pia-prenatal.ufk.charite.hu-berlin.de

Natürlich muß systemintern nicht der lange Text „*pia-prenatal.ufk.charite.hu-berlin.de*“ als Schlüssel verwendet werden. Eine dynamisch verwaltete Tabelle à la Hosts-Datei kann diesen langen Text durch eine 4 Byte lange Nummer ersetzen.

Die Vergabe der System-ID basiert auf der Internet-Struktur der Domains. Innerhalb der „parent“-Domain „*charite.hu-berlin.de*“ muß ein ID-System als Referenz deklariert werden. Sinnvollerweise wird für die Charité das Schlüsselsystem der zentralen Patientenverwaltung ausgewählt, das mit der Patienten-ID, Aufnahme-ID und Stations-Nummer eine Behandlung identifizieren kann. Z. Z. wird die Patientenverwaltung durch das Modul ISH der Firma SAP übernommen. Die Umrechnung von einem Referenzschlüssel zu einer subsystem-bezogenen, globalen ID erfolgt durch das Subsystem.

Graphische ID

Die graphische ID existiert nur temporär und ist immer abhängig von der gerade verwendeten Darstellung. Die graphische ID wird benutzt, damit der Anwender am virtuellen Befundviewer ein Objekt mit Hilfe eines Zeigergeräts, z. B. einer Maus, identifizieren kann. Sie wird vom Viewer generiert.

4.4.2 Objekt-Inhalt

Strenggenommen lassen sich alle Daten eines Objekts einheitlich als Attribut realisieren, inklusive der Schlüssel. Da die Schlüssel eine besondere Rolle spielen, werden sie wie im obigen Abschnitt gesondert behandelt.

Ein Attribut kann einfach sein. Alle einfachen Attribute müssen vom RPC-Protokoll unterstützt sein, um Daten über die Systemgrenzen hinweg und online austauschen zu können.

Die Attribute werden von der sogenannten Attribute-Engine behandelt. Die Attribute-Engine hat zwei primäre Aufgaben. Erstens sorgt sie für die richtige Interpretation der Attribute in Abhängigkeit vom medizinischen und Darstellungskontext. Zweitens führt sie bei der Integration eines einzelnen Attributes in ein Objekt die Integritätsprüfung durch. Die Attribute-Engine und die Kontexte sind näher im Abschnitt *4.2.8 Die Attribute-Engine (AE)* beschrieben.

Diese Attribute sind die primären Eigenschaften des Objekts. Sie sind unabhängig von der Umwelt. Z. B. ist der Geburtstag eine Ureigenschaft eines Patienten/einer Patientin - gleichgültig, in welcher Systemumgebung diese Eigenschaft betrachtet wird. Lediglich die Darstellung eines Datums kann von der Ländereinstellung und der bevorzugten Form abhängen. Nichtsdestotrotz hängt oft die Entscheidung, ob ein Attribut den Inhalt des Objekts darstellt oder nicht, von der jeweiligen Sicht des Anwenders bzw. der Anwendung ab.

Textdarstellungsattribut (Text Attributes):

Diese Attribute sind keine Ureigenschaft eines Objekts. Sie steuern lediglich die textliche Darstellung eines Attributs. Sie sind nicht zu verwechseln mit dem Textattribut eines graphischen Systems wie CorelDraw! oder Microsoft PowerPoint. Hier legen sie - systemunabhängig - die Vorschrift für die Textdarstellung des Objekts fest, falls es als Text dargestellt werden soll. Sie sind ähnlich wie die Textattribute bei HTML. Es wird nichts über die tatsächliche Größe des Textes ausgesagt. Sie legen vielmehr nur die Wichtigkeit der Darstellung fest.

Grafisches Darstellungsattribut (Graphic Attributes):

Analog zum Textdarstellungsattribut legen die graphischen Attribute die gewünschte graphische und bildliche Darstellungsart eines Objekts fest. Die tatsächliche Darstellung hängt vom jeweiligen medizinischen und Darstellungskontext ab.

Includes

Includes sind Objekte, die auf der Betriebssystemebene existieren. Sie können Teil eines Objekts sein oder aber andere Objekte enthalten, die auf einem Massenspeicher gespeichert sind. Includes werden verwendet, um größere binäre Daten abzuliegen. Beispielsweise kann eine TIFF-Datei als Include in einem Objekt verwendet werden.

4.4.3 Objektbeziehungen

Zwischen den Objekten sind Beziehungen möglich. Die Objektrelationen sind in den Objektattributen beschrieben und werden in der Regel auch wie Attribute behandelt. Somit lassen sich die Objektrelationen mit den Standardwerkzeugen der Attribute-Engine behandeln. Es bedarf keiner gesonderten Softwaremodule. Die Software wird damit klein gehalten.

Es sind zwei Arten von Relationen zu unterscheiden. Die strukturellen Relationen legen die formalen Beziehungen zwischen *Parent-Child* und *Sister-Sister* fest. Sie werden durch die Methoden *Get/SetParent*, *Get/SetChild* und *ToFirst/ToPrev/ToNext/ToLastObject* geregelt.

Die inhaltlichen Relationen werden dagegen durch die Methode *SelectObject* und *Get/SetRelation* abgedeckt.

4.4.4 Objektmethoden

Die aktiven Eigenschaften eines Objekts sind ausführbar und werden auch *Methoden* genannt. Jedes Objekt muß mindestens einen Satz von Pflichtmethoden für die Standardbehandlung von Objekten anbieten wie *CreateObject*, *DisplayObject*, etc.

4.5 Attributklassen

Die Attribute sind unterteilt in folgende Kategorien:

- Basisdatentypen,
- einfache Datentypen (Primitiva) und
- zusammengesetzte Datentypen (Structure).

Basisdatentypen sind Standarddatentypen, die von RPC unterstützt werden. Sie sind ohne Dimension und bilden die Basis für die Primitiva. Jedes Primitivum besteht aus einer Struktur, die den Attributnamen, den Typ des Attributwertes, den Wertebereich und die Dimension des Attributes enthält.

```
<primitivum> ::= <name>
                  <type-ID>
                  <value type>
                  [<value range>]
                  [<unit>]
                  [<scope>]
```

Der Name des Attributs *<name>* wird vom System vergeben. Die Treiber setzen diese Systemnamen in die Namen um, die vom Subsystem interpretiert werden können. Jedes Attribut muß einem Typ zugeordnet sein. Ein Attributtyp kann z. B. Durchmesser, Länge, Gewicht, Diagnose, Therapie, etc. sein. Der Attributtyp *<type-ID>* wird ebenfalls vom System vergeben. Der Typ des Attributwertes *<value type>* gibt an, welchen Basistyp der Attributwert besitzt (s. oben). Der Wertebereich *<value range>* gibt an, in welchem Bereich der Attributwert liegen darf. Der Geltungsbereich *<scope>* gibt das System an, in dem die Bezeichnung bzw. die Einheit verwendet wird. Beispielsweise kann der Scope „SNOMED V3.3“ oder „local“

sein. Die Dimension *<unit>* gibt schließlich die Einheit des Attributwertes an. *<value range>*, *<unit>* und *<scope>* sind optional.

Beim Austausch können die beteiligten Softwarekomponenten an Hand dieser Angaben und ggf. vom medizinischen sowie Darstellungskontext entscheiden, wie ein Attribut umgerechnet und dargestellt werden soll. Die Attribute Engine ist immer in der Lage, Primitiva zu behandeln. Dies garantiert, daß ein virtueller Befund immer angezeigt bzw. ausgedruckt werden kann.

Zusammengesetzte Attribute sind Strukturen aus Primitiva. Zusätzlich sind die Zusammenhänge zwischen den Attributen mitgespeichert, so daß eine Darstellungs-routine online prüfen kann, ob die Eingabe noch gültig ist, ohne den Server ansprechen zu müssen. Ein zusammengesetztes Attribut kann wieder ein Objekt sein. Alle Objekte stammen aus einer Klasse und haben die gleiche Struktur. Im folgenden wird die allgemeine Objektstruktur beschrieben.

4.6 Objektklassen

Objekte gleichartiger Eigenschaften bilden eine Klasse. Eine bestimmte Menge von Objekten einer Klasse bildet ein *ObjectSet*, analog zum *RecordSet* der relationalen Datenbanken. Hier wird ein hybrides Konzept gewählt. Einerseits sind durch das *ObjectSet* Operationen wie bei einer relationalen Datenbank möglich und leicht abbildbar. Andererseits wird durch das Objektkonzept die OO-Verarbeitung ermöglicht.

Das Objekt, das zwischen dem medizinischen Subsystem und dem virtuellen Befundsystem ausgetauscht werden soll, ist die virtuelle Behandlung. Sie enthält u. a. zahlreiche Verweise auf andere Objekte, die nicht von medizinischen Systemen verwaltet werden. Das sind die Stammdaten (Anatomie, Syndrome, Normwerte, IDC/ICPM-Codes), die personenbezogenen Daten (Patienten, medizinisches und administratives Personal), Daten über die Einrichtungen (Klinik, Station, Institut, Verwaltungsstelle, Behörde, Praxis). Alle diese Daten werden auch als Objekte gehandhabt (siehe auch unten).

Im folgenden werden die Objektklassen aufgelistet, die zum Nucleus des virtuellen Befundsystems gehören und die Merkmale des Konzepts ausprägen. Hierbei handelt es sich beispielsweise um die Klassen *Behandlung*, *Untersuchung*, *Patient*, *Diagnose*, etc. Andere Objektklassen, die auch wichtig für das Funktionieren eines Systems sind, jedoch nicht zum Wesen des Konzepts gehören, werden nicht zum Nucleus gezählt. Nicht zum Nucleus gehört z. B. die Verwaltung der Systemkomponenten, Hilptexte, etc.

Der Nucleus wird wiederum in 2 Bereiche unterteilt. Der erste wird Anwendungs- der zweite System-Nucleus genannt. Der Anwendungs-Nucleus enthält Objektklassen, die für die Anwendungen relevant sind. Das sind z. B. *Behandlung*, *Untersuchung*, *Diagnose*, *Therapie*, etc. Zum System-Nucleus gehören Objektklassen, die für das Funktionieren des Systems lebensnotwendig sind, die jedoch nicht direkt mit den Anwendungen zu tun haben, z. B. *Netzwerk-Anschluß*, *Transaktionen*, etc..

Abbildung 4-6 zeigt die Objektklassen, die zum Anwendungs-Nucleus gehören. Anschließend werden die wesentlichen Bestandteile der Datenstruktur kurz erläutert.

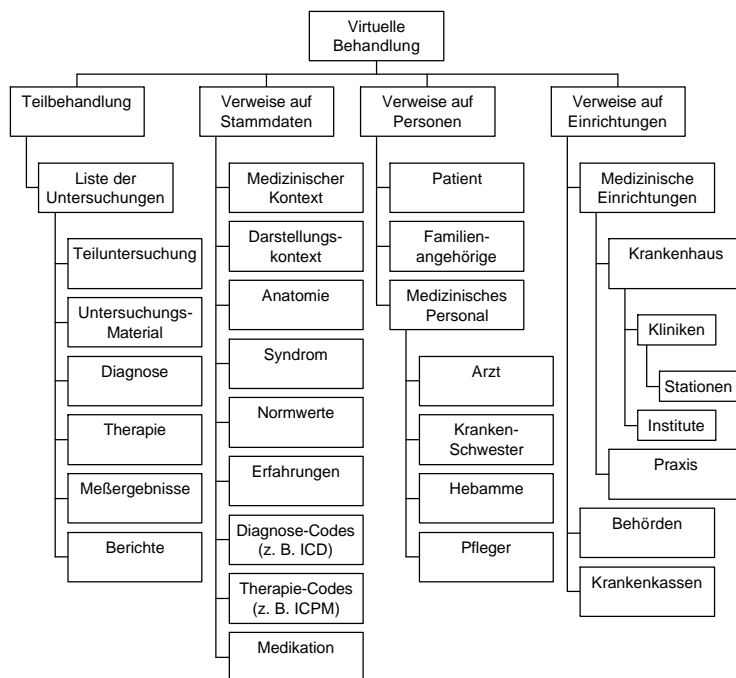


Abbildung 4-6 Schematische Darstellung der Datenstruktur

Einrichtung

Eine Einrichtung ist jede Organisationseinheit im Sinne des X500-Standards. Beispielsweise bezeichnet „de“ eine Einrichtung. „ufk.charite.hu-berlin.de“ ist ebenfalls eine Bezeichnung der Einrichtung *Universitätsfrauenklinik*. Die externe Bezeichnung der Einrichtung muß den Internet-Namenskonventionen genügen.

Subsystem

Ein Subsystem im Sinne des Standards ist jedes Softwaresystem, das über einen *virtuellen Befundtreiber* Informationen an das System liefert und/oder Informationen vom System bezieht.

Medizinisches Subsystem

Ein medizinisches Subsystem ist ein Subsystem, das für einen Teil der medizinischen Informationen im virtuellen Befundsystem verantwortlich ist. Medizinische Informationen beinhalten auch die Codierung ICD, ICPM der Diagnose bzw. Therapie. In der Regel liefert ein medizinisches Subsystem medizinische Informationen an das System und bezieht administrative Informationen vom System.

Administratives Subsystem

Ein administratives Subsystem ist ein Subsystem, das verantwortlich für einen Teil der administrativen Informationen im virtuellen Befundsystem ist. Administrative Informationen beinhalten die Patientendaten, Krankenkassen, Einweisungsdaten und Daten über die Aufnahme, Verlegung und Entlassung des Patienten/der Patientinnen. Abrechnungsdaten werden hier nicht betrachtet.

In der Regel liefert ein medizinisches Subsystem medizinische Informationen an das System und bezieht administrative Informationen vom System.

Virtuelle Behandlung

Es wird ein Hauptdatentyp festgelegt. Das ist die „*virtuelle Behandlung*“. Eine *virtuelle Behandlung*, oder einfach *Behandlung* genannt, ist jeder Vorgang in einem Subsystem, der als Ergebnis einen Befund liefert. Eine Behandlung ist entweder eine „top-level“- oder Teilbehandlung. Eine top-level-Behandlung hat eine globale Objekt-ID der höchsten Organisationseinheit im virtuellen Befundsystem. Eine Teilbehandlung ist ein Teil einer übergeordneten Behandlung, einer sogenannten parent-Behandlung. Die übergeordnete Behandlung kann eine top-level- oder aber eine Teilbehandlung sein.

Alle anderen Informationen wie Patienten, Anamnese, Diagnose, Therapie, etc. sind Attribute der Behandlung. Die Behandlung wird gemäß der Objektorientierung mit einem Satz von Operationen festgelegt.

Beispielsweise wird eine top-level-Behandlung in der Charité durch eine eindeutige z. Z. 12stellige Aufnahme-Nummer identifiziert. Im virtuellen Befundsystem wird eine solche Behandlung z. B. durch folgende globale Objekt-ID bezeichnet:

`123456789012@h17.charite.hu-berlin.de`

Die virtuelle Behandlung wird durch einen virtuellen Befund dargestellt. Eine virtuelle Behandlung beginnt mit der Aufnahme und endet mit der Entlassung des Patienten/der Patientin. Alle Vorgänge und Ergebnisse während dieser Zeit lassen sich als Attribute des Objekts *virtuelle Behandlung* integrieren. Hierzu siehe auch Abbildung 4-6.

Untersuchungsmaterial

Das Untersuchungsmaterial ist das Objekt einer Behandlung. Es bildet immer eine anatomische Einheit.

Anmerkungen

In den meisten Fällen ist das Untersuchungsmaterial ein Teil des Patienten/der Patientin. Es kann jedoch von einer anderen Person stammen, z. B. einem Fet oder einem Verwandten des Patienten/der Patientin. Es kann auch der ganze Körper eines Menschen sein. Beim Kontextmatching wird diese Tatsache berücksichtigt.

Person

Alle Personen, die irgendwie mit dem virtuellen Befundsystem zu tun haben, werden durch ein Objekt dargestellt. Sie werden lediglich durch ihre Rolle voneinander unterschieden. Beispielsweise kann ein und dieselbe Person einmal die Rolle des medizinischen Personals, einmal die Rolle eines Patienten übernehmen.

Technisch gesehen läßt sich ein Patient optimal durch den zusammengesetzten Schlüssel

`<Personalausweis-Nummer>@<Land>`

weltweit eindeutig identifizieren. Diese Art steht allerdings heute nicht zur Verfügung. So läßt sich z. B. die Patient-ID durch

`<Charité-Patient-Nummer>@charite.hu-berlin.de`

ebenfalls weltweit eindeutig identifizieren. Dadurch wird eine mehrfache Speicherung einer Person vermieden. Durch den Abgleich der AMICI-Personendatenbank mit den in den Subsystemen gespeicherten Patientendaten könnten Eingabefehler gefunden und korrigiert werden.

Personen werden in folgende Gruppen eingeteilt:

- Patient/Patientin,
- Familienangehörige und
- Medizinisches Personal
- Administratives Personal

Virtueller Befund

Eine *virtuelle Behandlung* wird durch einen *virtuellen Befund* dargestellt. Der Anwender sieht die virtuelle Behandlung durch den *virtuellen Befund*. In Datenbanksprache stellt der *virtuelle Befund* ein View auf die *virtuelle Behandlung* dar. Welche Bestandteile der virtuellen Behandlung im virtuellen Befund angezeigt werden sollen, wird in Vorlagen definiert. Jeder Anwender kann eigene Vorlagen zum Anzeigen von *virtuellen Befunden* definieren.

Ein *virtueller Befund* muß nicht real existieren. Es wird bei Bedarf von den Informationen der Subsysteme gebildet. Der Inhalt und das Layout eines virtuellen Befunds läßt sich vom Anwender definieren. Der Inhalt wird von den gewünschten Subsystemen geliefert. Die Darstellung wird durch den medizinischen und den Darstellungskontext gesteuert. Beides kann vom Anwender festgelegt werden. Die Darstellung übernimmt der *virtuelle Befundviewer*.

Anmerkungen:

Der Begriff *virtueller Befund* wird hier daher sehr weit gefaßt. Alles, was die Subsysteme liefern, läßt sich in einen virtuellen Befund integrieren. Durch die Definition eines medizinischen Kontextes kann der Anwender quasi die Krankenakte eines Patienten/einer Patientin anzeigen lassen. Alle Behandlungen, die durch die Subsysteme verfügbar gemacht werden können, lassen sich anzeigen.

Darstellungskontext

Bei der Anzeigefunktion wird der Darstellungskontext benötigt, in dem die vom Sendesystem gewünschte Darstellung gespeichert ist. Im Darstellungskontext sind die Darstellungsattribute abstrakt definiert. Hier wird keine physikalische Angabe gemacht. Das Empfängersystem wertet den Darstellungskontext aus und vergleicht ihn mit dem jeweiligen Device Context, um Darstellungsvorschriften für die Darstellungsroutinen zu formulieren.

4.6.1 Beispiel für Objektdefinition

```
<OBJECT TYPE="MedicalContext" INTERNAL-ID="141.11.11.11" EXTERNAL-
ID="prenatal.ufk.charite.hu-berlin.de" CODEBASE="ufkscience.prenatal.ufk.charite.hu-
berlin.de/data/reference-server/prenatalcontext.exe>
  <ATTR NAME="Scope" TYPE="Text" VALUE="SNOMED V3.3" >
  <ATTR NAME="Language" TYPE="Text" VALUE="EN" >
  <ATTR NAME="VirtualOrgane" TYPE="Text" VALUE="Fetus">
  <ATTR NAME="VirtualOrgane" TYPE="SNOMED V3.3" VALUE="T-F5000">
  <ATTR NAME="VirtualProcedure" TYPE="Text" VALUE="Ultrasonography">
  <ATTR NAME="VirtualProcedure" TYPE="SNOMED V3.3" VALUE="P5-B0000">
  <ATTR NAME="VirtualProcedure" TYPE="Text" VALUE="Diagnostic Doppler ultrasonography">
  <ATTR NAME="VirtualProcedure" TYPE="SNOMED V3.3" VALUE="P5-B0110">
  <ATTR NAME="Diagnosis" TYPE="ICD-10" VALUE="Q33.6">
  <ATTR NAME="Diagnosis" TYPE="Text" VALUE="secundäre Lungenhypoplasie">
  <ATTR NAME="Object" TYPE="Reference" VALUE="9876543421">
  <ATTR NAME="Histology" TYPE="Img" VALUE="http://www.ufk.prenatal.charite.de/prenatal-
db/images/1111111.tif">
</OBJECT>
```

4.7 Die Subsysteme

Im ersten Schritt ist der Anschluß folgender Subsysteme vorgesehen:

- die Punktionsdatei,
- die geburtshilfliche Datenbank,
- das Pränatal-Ultraschall-Befund- und -Archivierungssystem,
- das Fetalautopsie-Protokoll-System und
- die Patienten-Verwaltung über HL7.

Jedes Subsystem wird zu einem Knoten, wobei ein Windows-NT-Server mehrere Knoten aufnehmen kann. Zu den den Subsystemen zugehörigen Knoten kommen zwei weitere Knoten dazu. Der erste ist der Referenzknoten und hat den medizinischen Kontext „System“. Der zweite ist ein Knoten für nicht subsystemgebundene, allgemeine Anfragen. Beispielsweise können hier Studenten oder interdisziplinäre Forschungsgruppen nicht-patienten- und nicht-personengebundene Studien vornehmen. Somit sind folgende Knoten notwendig:

- der invasive Knoten,
- der Pränatal-Ultraschall-Knoten,
- der geburtshilfliche Knoten,
- der Fetalautopsie-Knoten,
- der HL7-Knoten,
- der allgemeine Knoten und
- der Referenzknoten.

Im folgenden werden diese Knoten näher beschrieben.

4.7.1 Der invasive Knoten

Die hier zu behandelnde Punktionsdatei enthält primär Informationen zu invasiven Maßnahmen, die in der pränatalen Diagnostik vorgenommen wurden. Diese Maßnahmen werden Zeile für Zeile eingegeben. Das Merkmal der Punktionsdatei ist es, daß die Datenbank keine eigentlichen Schlüssel besitzt. Die Informationen über die

Patientin werden in vollem Umfang mit Namen, Vornamen, Geburtsort/Geburtsdatum und Adresse gespeichert.

Der virtuelle Befundtreiber für diesen Knoten muß daher eine eigene Tabelle für die Verwaltung der lokalen Identifikationen sowie für deren Umrechnung in das globale System besitzen.

Der zugehörige medizinische Kontext lautet:

```
<MEDICALCONTEXT NAME="Prenatal-Invasive" INTERNAL-ID="x.x.x.x:3031/4712" EXTERNAL-
ID="invasive@prenatal.ufk.charite.hu-berlin.de" CODEBASE="ufk-science.charite.hu-
berlin.de/data/reference-server/prenatalcontext/md.exe>
  <ATTR NAME="Scope" TYPE="Text" VALUE="SNOMED V3.3" >
  <ATTR NAME="Language" TYPE="Text" VALUE="EN" >
  <GROUP NAME="Patient">
    <ATTR NAME="Sex" TYPE="Text" VALUE="female">
    <ATTR NAME="MaxAge" TYPE="Integer" VALUE="50">
    <ATTR NAME="MinAge" TYPE="Integer" VALUE="10"> ....
  </GROUP>
  <GROUP NAME="ListOfOrgane">
    <ATTR NAME="Organe" TYPE="Text" VALUE="Placenta">
    <ATTR NAME="Organe" TYPE="Text" VALUE="Fetus">
    <ATTR NAME="Organe" TYPE="Text" VALUE="Uterus"> ...
  </GROUP>
  <GROUP NAME="ListOfProcedures">
    <ATTR NAME="Procedure" TYPE="Text" VALUE="CVS">
    <ATTR NAME="Procedure" TYPE="SNOMED V3.3" VALUE="P2-00020"> ...
  </GROUP>
  <GROUP NAME="ListOfDiagnosis">
    <ATTR NAME="Diagnosis" TYPE="ICD-10" VALUE="Q33.6">
    <ATTR NAME="Diagnosis" TYPE="Text" VALUE="secundäre Lungenhypoplasie">
    <ATTR NAME="Diagnosis" TYPE="ICD-10" VALUE="P27.0">
    <ATTR NAME="Diagnosis" TYPE="Text" VALUE="hypoplastische Nabelschnurarterie
links"> ...
  </GROUP>
</MEDICALCONTEXT>
```

4.7.2 Der geburtshilfliche Knoten

Die im Kreißsaal verwendete Datenbank erfaßt die geburtshilflichen Informationen. Primär sind es Daten über die Geburten selbst mit Angaben über die Schwangerschaft, die Entbindung und das Kind bzw. die Kinder. Diese Datenbank wird wie ein Geburtenbuch organisiert. Ihr Merkmal ist es, daß die Daten jahresweise organisiert werden, wobei die Geburten innerhalb eines Jahres mit einer Geburtennummer als Schlüssel versehen werden. Ähnlich wie bei der Punktionsdatei muß hier der virtuelle Befundtreiber eine eigene Verwaltung und Umrechnung der Schlüssel für die Patientinnen und für die „Behandlungen“ (Entbindungen bzw. Operationen) besitzen.

Der zugehörige medizinische Kontext lautet:

```
<MEDICALCONTEXT NAME="Obstetrics" INTERNAL-ID="x.x.x.x:3031/4713" EXTERNAL-
ID="xx@obstetrics.ufk.charite.hu-berlin.de" CODEBASE="ufk-science.charite.hu-
berlin.de/data/reference-server/obstetricscontext/md.exe>
  <ATTR NAME="Scope" TYPE="Text" VALUE="SNOMED V3.3" >
  <ATTR NAME="Language" TYPE="Text" VALUE="EN" >
```

```

<GROUP NAME="Patient">
  <ATTR NAME="Sex" TYPE="Text" VALUE="female">
  <ATTR NAME="MaxAge" TYPE="Integer" VALUE="50">
  <ATTR NAME="MinAge" TYPE="Integer" VALUE="10"> ....
</GROUP>
<GROUP NAME="ListOfOrgane">
  <ATTR NAME="Organe" TYPE="Text" VALUE="Placenta">
  <ATTR NAME="Organe" TYPE="Text" VALUE="Fetus">
  <ATTR NAME="Organe" TYPE="Text" VALUE="Uterus"> ...
</GROUP>
<GROUP NAME="ListOfProcedures">
  <ATTR NAME="Procedure" TYPE="Text" VALUE="OBSTETRICS: EXCISIONS">
  <ATTR NAME="Procedure" TYPE="SNOMED V3.3" VALUE="P1-86300">
  <ATTR NAME="Procedure" TYPE="Text" VALUE="Classical cesarean section">
  <ATTR NAME="Procedure" TYPE="SNOMED V3.3" VALUE="P1-86311"> ...
</GROUP>
<GROUP NAME="ListOfDiagnosis">
  <ATTR NAME="Diagnosis" TYPE="ICD-10" VALUE=" P 27.0">
  <ATTR NAME="Diagnosis" TYPE="Text" VALUE=" hypoplastische Nabelschnurarterie">
  <ATTR NAME="Diagnosis" TYPE="ICD-10" VALUE=" P 02">
  <ATTR NAME="Diagnosis" TYPE="Text" VALUE=" Minderwuchs"> ...
</GROUP>
</MEDICALCONTEXT>

```

4.7.3 Der Pränatal-Ultraschall-Knoten

Der Pränatale Knoten basiert auf einem Ultraschall-Befund- und -Archivierungssystem. Dieses ist ein System aus mehreren Servern, wobei der Anwendungs-Client sich immer an den Anwendungs-Server wendet, um die Anwendung nutzen zu können. Die Datenbank dieses Subsystems hat sowohl Schlüssel für die Untersuchungen als auch für die Patientinnen. Die Umsetzung dieser Identifikationen erfolgt gemäß Abschnitt 4.4.1 *Objekt-Identifikation*.

Die interne Struktur dieses Subsystems ist zwar recht komplex (siehe Abbildung 4-8). Diese Komplexität bleibt dem AMICI-Knoten jedoch verborgen. Er nimmt lediglich Kontakt mit seinem Ansprechpartner - dem virtuellen Befundtreiber - auf.

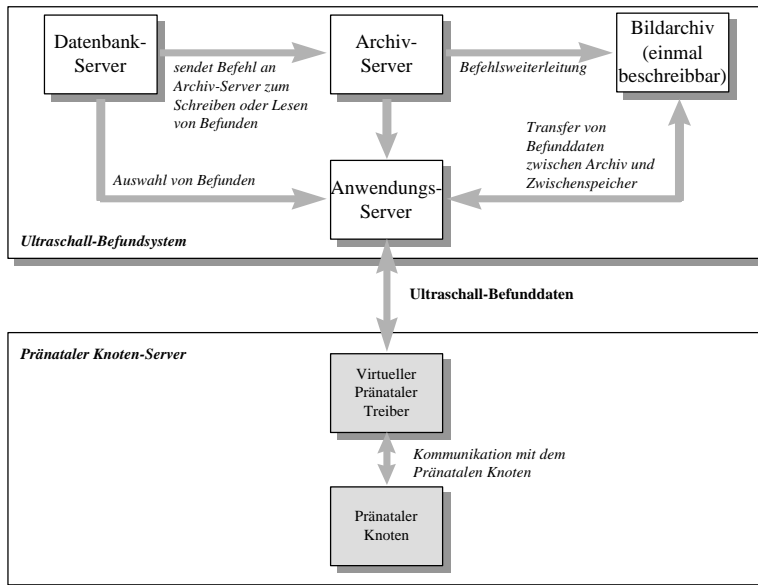


Abbildung 4-7: Struktur des Ultraschall-Befund- und -Archivierungssystems

Dem Befund- und Archivierungssystem erscheint der Pränatal-Ultraschall-Knoten wiederum als ein gewöhnlicher Client. Dieser führt lediglich besondere Funktionen aus, was dem Subsystem jedoch nicht bekannt sein muß. Die Kommunikation zwischen dem Befundsystem und dem Pränatal-Knoten erfolgt gemäß AMICI-Konzept über den virtuellen Befundtreiber. Der Pränatal-Ultraschall-Knoten-Server ist ein Windows NT-Server mit der Konfiguration gemäß Abbildung 4-2. Der zugehörige medizinische Kontext lautet:

```
<MEDICALCONTEXT NAME="Prenatal-US" INTERNAL-ID="x.x.x.x:3031/4711" EXTERNAL-
ID="us@prenatal.ufk.charite.hu-berlin.de" CODEBASE="ufk-science.charite.hu-
berlin.de/data/reference-server/prenatalcontext/md.exe>
  <ATTR NAME="Scope" TYPE="Text" VALUE="SNOMED V3.3" >
  <ATTR NAME="Language" TYPE="Text" VALUE="EN" >
  <GROUP NAME="Patient">
    <ATTR NAME="Sex" TYPE="Text" VALUE="female">
    <ATTR NAME="MaxAge" TYPE="Integer" VALUE="50">
    <ATTR NAME="MinAge" TYPE="Integer" VALUE="10"> ...
  </GROUP>
  <GROUP NAME="ListOfOrgane">
    <ATTR NAME="Organe" TYPE="Text" VALUE="Placenta">
    <ATTR NAME="Organe" TYPE="Text" VALUE="Fetus">
    <ATTR NAME="Organe" TYPE="Text" VALUE="Uterus"> ...
  </GROUP>
  <GROUP NAME="ListOfProcedures">
    <ATTR NAME="Procedure" TYPE="Text" VALUE="Ultrasound">
    <ATTR NAME="Procedure" TYPE="Text" VALUE="Color-Doppler">
    <ATTR NAME="Procedure" TYPE="Text" VALUE="3D-Color-Doppler"> ...
  </GROUP>
  <GROUP NAME="ListOfDiagnosis">
    <ATTR NAME="Diagnosis" TYPE="ICD-10" VALUE="Q0">
    <ATTR NAME="Diagnosis" TYPE="Text" VALUE="xxxx">
    <ATTR NAME="Diagnosis" TYPE="ICD-10" VALUE="P0">
    <ATTR NAME="Diagnosis" TYPE="Text" VALUE="xxxx"> ...
  </GROUP>
</MEDICALCONTEXT>
```

4.7.4 Der Fetalautopsie-Knoten

Der Fetalautopsie-Knoten ist eine Eigenentwicklung. Die Basisdatenbank ist in MS Access realisiert. Die Struktur dieser Datenbank ist schon nach dem allgemeinen Objekt-konzept ausgerichtet. Ein Merkmal dieser Datenbank ist, daß alle Objekte ein eigenes Schlüssel-system haben: Untersuchungen, Patienten (Fetus), Organe, Methoden, Bilder, etc. Der zugehörige medizinische Kontext lautet:

```
<MEDICALCONTEXT NAME="Fetalautopsy" INTERNAL-ID ="x.x.x.x:3031/4714" EXTERNAL-
ID="autopsy@fetal.ifp.charite.hu-berlin.de" CODEBASE="ifp-science.charite.hu-
berlin.de/data/reference-server/pathologycontext/md.exe"
  <ATTR NAME="Scope" TYPE="Text" VALUE="SNOMED V3.3" >
  <ATTR NAME="Language" TYPE="Text" VALUE="EN" >
  <GROUP NAME="Context">
    <GROUP NAME="Patient">
      <ATTR NAME="Sex" TYPE="Text" VALUE="female">
      <ATTR NAME="MaxAge" TYPE="Integer" VALUE="50">
      <ATTR NAME="MinAge" TYPE="Integer" VALUE="10"> ....
    </GROUP>
    <GROUP NAME="ListOfOrgane">
      <ATTR NAME="Organe" TYPE="Text" VALUE="Placenta">
      <ATTR NAME="Organe" TYPE="Text" VALUE="Uterus"> ...
    </GROUP>
    <GROUP NAME="ListOfProcedures">
      <ATTR NAME="Procedure" TYPE="Text" VALUE="Autopsy">
      <ATTR NAME="Procedure" TYPE="Text" VALUE="Macroscopy">
      <ATTR NAME="Procedure" TYPE="Text" VALUE="Microscopy"> ...
    </GROUP>
    <GROUP NAME="ListOfDiagnosis">
      <ATTR NAME="Diagnosis" TYPE="ICD-10" VALUE="P0">
      <ATTR NAME="Diagnosis" TYPE="Text" VALUE="xxxx">
      <ATTR NAME="Diagnosis" TYPE="ICD-10" VALUE="P1">
      <ATTR NAME="Diagnosis" TYPE="Text" VALUE="xxxx"> ...
    </GROUP>
  </GROUP>
  <GROUP NAME="Context">
    <GROUP NAME="Patient">
      <ATTR NAME="Name" TYPE="Text" VALUE="Fetus">
      <ATTR NAME="Sex" TYPE="Text" VALUE="don't care">
      <ATTR NAME="MaxAge" TYPE="Integer" VALUE="45" UNIT="WoG">
      <ATTR NAME="MinAge" TYPE="Integer" VALUE="0" UNIT="WoG"> ....
    </GROUP>
    <GROUP NAME="ListOfOrgane">
      <ATTR NAME="Organe" TYPE="Text" VALUE="don't care">
    </GROUP>
    <GROUP NAME="ListOfProcedures">
      <ATTR NAME="Procedure" TYPE="Text" VALUE="Autopsy">
      <ATTR NAME="Procedure" TYPE="Text" VALUE="Macroscopy">
      <ATTR NAME="Procedure" TYPE="Text" VALUE="Microscopy"> ...
    </GROUP>
    <GROUP NAME="ListOfDiagnosis">
      <ATTR NAME="Diagnosis" TYPE="ICD-10" VALUE="Q0">
      <ATTR NAME="Diagnosis" TYPE="Text" VALUE="xxxx">
      <ATTR NAME="Diagnosis" TYPE="ICD-10" VALUE="Q1">
      <ATTR NAME="Diagnosis" TYPE="Text" VALUE="xxxx"> ...
    </GROUP>
  </GROUP>
</MEDICALCONTEXT>
```

Erläuterung

Hier werden im Gegensatz zum Pränatal-Ultraschall-Kontext 2 Patiententypen eingetragen: die schwangere Frau und das Fet. Für jeden Patiententyp wird ein eigener Kontext angelegt.

4.7.5 Der HL7-Knoten

Der HL7-Knoten basiert auf dem HL7-Server, der ursprünglich für die Erfassung der Patientenstammdaten für das Charité-Tumorzentrum konzipiert wurde.

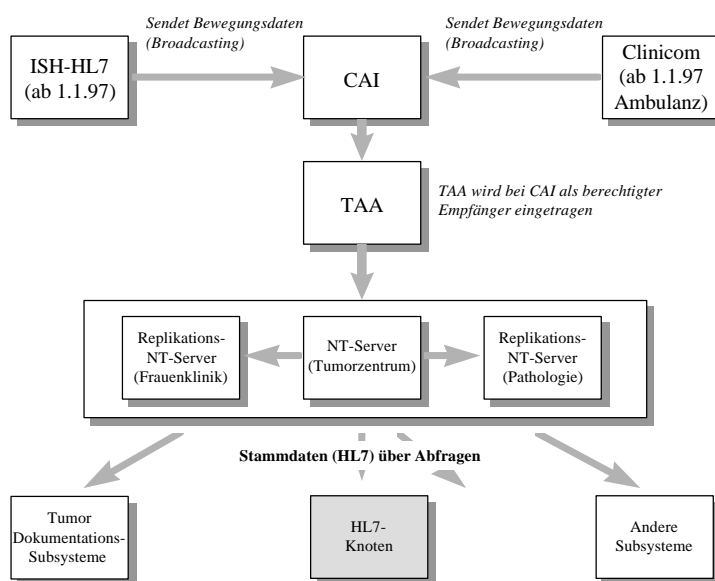


Abbildung 4-8: Struktur des HL7-Servers

Der HL7-Server läuft aus Sicherheitsgründen auf einem System von verteilten und replizierten auf SyBase basierten Datenbankservern, das die Patienten-Stammdaten zwischenspeichert. Diese Server werden vom Kommunikationsserver des Charité-Rechenzentrums CAI über eine Transaktionsanwendung TAA (Modul ISH der Firma SAP bzw. System Clinicom der Fa. SMS) gespeist. Der HL7-Server bereitet die Daten auf und bietet den anfragenden Subsystemen durch Abfrage nur die Stammdaten an, die das Subsystem auch benötigt. Somit muß nicht jedes Subsystem die Stammdaten speichern bzw. zwischenspeichern. Im Gegensatz zu anderen HL7-Servern, die nach dem Broadcasting-Prinzip arbeiten, in dem die in der Patientenverwaltung stattfindenden Transaktionen wie Aufnahme, Verlegung und Entlassung als Nachrichten an alle Interessenten automatisch und ständig verschickt werden, bietet dieser HL7-Server die Daten per Abfrage an. Sie werden nur bei Bedarf vom HL7-Server an den anfragenden Client geschickt. Der HL7-Server übernimmt ferner die Verwaltung von Zugangsberechtigungen, um den Datenschutz und die Datensicherheit zu gewährleisten. Dies betrifft auch den Schutz klinikinterner Daten.

Der AMICI-HL7-Knoten bedient sich dieses HL7-Servers und ist verantwortlich für die einheitliche Versorgung des AMICI-Netzes mit Patienteninformationen. Ein virtueller HL7-Treiber verpackt den HL7-Server mit der AMICI-Funktionalität, so daß der HL7-Anschluß gewährleistet ist.

Der zugehörige medizinische Kontext lautet:

```
<MEDICALCONTEXT NAME="Admin" INTERNAL-ID="x.x.x.x:3031/4715" EXTERNAL-
ID="patient@Admin.charite.hu-berlin.de" CODEBASE="admin.charite.hu-berlin.de/data/reference-
server/admincontext/md.exe">
  <ATTR NAME="Scope" TYPE="Text" VALUE="HL7 V2.2" >
  <ATTR NAME="Language" TYPE="Text" VALUE="DE" >
  <GROUP NAME="Patient" MATCH=„yes“>
    <ATTR NAME="Sex" TYPE="Text" VALUE="don't care">
    <ATTR NAME="MaxAge" TYPE="Text" VALUE=" don't care">
    <ATTR NAME="MinAge" TYPE="Integer" VALUE=" don't care"> ....
  </GROUP>
</MEDICALCONTEXT>
```

4.7.6 Der allgemeine Knoten

Der allgemeine Knoten hat kein eigentliches medizinisches Subsystem als Basis. Er fragt je nach Anfrage vom Client die Knoten im Netz ab. Sein medizinischer Kontext ist standardmäßig leer. Hier lassen sich jedoch beliebige medizinische Kontexte, beispielsweise für Studienzwecke dynamisch generieren.

4.7.7 Der Referenzknoten

Der Referenzknoten hat ebenfalls kein eigentliches medizinisches Subsystem als Basis. Hier werden die Systemdaten wie alle medizinischen Kontexte, der Namensraum, die Liste der gültigen Primitiva, die URL-Codebases für die Objekte und die Benutzer mit ihren Rechten und Autorisierungen, etc. verwaltet.

Obwohl dieser Knoten kein medizinisches Subsystem im Hintergrund besitzt, hat er jedoch einen virtuellen Befundtreiber, der die dort verwalteten Daten weitergibt.

Der zugehörige medizinische Kontext lautet:

```
<MEDICALCONTEXT NAME="System" INTERNAL-ID="x.x.x.x:3031/4710" EXTERNAL-
ID="system@amici.charite.hu-berlin.de" CODEBASE="amici.charite.hu-berlin.de/data/reference-
server/systemcontext/md.exe">
  <ATTR NAME="Scope" TYPE="Text" VALUE=„SNOMED V3.3“ MATCH=„no“>
  <ATTR NAME="Language" TYPE="Text" VALUE="EN" MATCH=„no“>
</MEDICALCONTEXT>
```

4.8 Testimplementations

Die Untersuchung umfaßte ebenfalls eine Testimplementations, die an folgenden Subsystemen vorgenommen wurde:

- Pränatalultraschall,
- GebDat und
- Punktionsdatei.

Die Testimplementation diente folgenden Zwecken:

- Testen der Möglichkeit, auf fremde Datenbanken zuzugreifen,
- Testen der gleichzeitigen Zugriffe auf mehrere Datenbanken zwecks Abgleich und
- Messen des Antwortzeitverhaltens beim Zugriff mit eigener Software.

4.8.1 Zugriffe auf fremde Datenbanken

Bis auf das Subsystem Pränatalultraschall, bei dem die Befunddaten verschlüsselt in der Datenbank abgelegt werden, konnte die Semantik der restlichen Daten mit Hilfe von Abgleichen zwischen Datenbankinhalt und herkömmlichen Befunden festgestellt werden. Für den Zugriff auf die Pränatalultraschalldaten lieferte der Hersteller die Software für die Konvertierung der verschlüsselten Daten in eine SyBase-Datenbank im Klartext. Somit konnte auf alle Fremddaten ungehindert zugegriffen werden. Der Zugriff erfolgte über verschiedene ODBC-Treiber.

Mit eigener Software konnten ferner Zugriffseinschränkungen ohne weiteres aufgehoben werden. Beispielsweise konnte bei der Punktionsdatei mit der Originalsoftware nur jahresweise gearbeitet werden. Die eigene Software verarbeitet alle Jahrgänge von 1986 bis heute.

4.8.2 Abgleich zwischen verschiedenen Datenbanken

Ist der gleichzeitige Zugriffe auf mehrere Datenbestände möglich, so lassen sich viele Auswertungen vornehmen, die bisher nicht möglich waren. Beispielsweise kann der Anwender alle relevanten Informationen aus mehreren Datenbanken verwenden, die einen Patienten/eine Patientin betreffen. Somit lassen sich Krankengeschichten besser konstruieren. Eine andere Möglichkeit besteht der Abgleich verschiedener Datenbanken.

Tabelle 4-2 Verteilung eines Diagnosetextes in verschiedenen Schreibweisen

Diagnose	Anzahl	Diagnose	Anzahl	Diagnose	Anzahl
Zyogenetik	9	Zytogenetikj	1	Zytogenwetik	1
Zytgenetik	3	ZytogenetikKary	1	Zytogernetik	1
Zytoenetiik	1	Zytogenetilk	1	Zytoggenetik	1
Zytoenetik	8	Zytogenetk	7	Zytognetik	1
Zytogebnetik	1	Zytogenezik	1	Zytonetik	1
Zytogeneik	13	Zytogeneztik	1	Zytoogenetik	1
Zytogenetik	1719	Zytogenik	1	Zytotogenetik	1
Zytogenetik 2	1	Zytogenmetik	1	Zytpogenetik	1
ZytogenetikAZ	1	Zytogennetik	1	Zytzogenetik	1
ZytogenetikCV	1	Zytogentik	3	Zyzogenetik	1

In der Testimplementation wurden die GebDat-Datenbank mit 6919 Datensätzen und die Punktionsdatei mit 25605 Datensätzen miteinander abgeglichen, um zu überprüfen, ob die Adressen der Patientinnen in beiden Datenbanken noch übereinstimmen. Von den 6919 GebDat-Datensätzen sind 5248 Patientinnen auch in der Punktionsdatei enthalten, davon sind 58 Datensätze (1%), bei denen die Postleitzahlen übereinstimmen. Beim Abgleich der Straßennamen fanden sich 680 gleiche Einträge (13%) . Der Grund für die schlechte Übereinstimmung von 1% war, daß

die eine Datenbank schon und die andere noch nicht auf die 5stellige PLZ umgestellt worden ist. Bei den Straßennamen lag der Grund in den verschiedenen Schreibweisen. Wie die Schreibweisen die Ergebnisse einer Auswertung verfälschen können, zeigt die Tabelle 4-2, die die Verteilung der Diagnoseeingabe „Zytogenetik“ auf 30 verschiedenen Schreibweisen in einer Datenbank wiedergibt. Mit dieser Art der Abfrage ließe sich die Aktualisierung bzw. die Korrektur einer Datenbank mit Hilfe anderer, korrekterer Datenbestände ohne weiteres durchführen.

Der Abgleich der Daten aus mehreren Datenbeständen bereitete erwartungsgemäß Probleme bei der Identifikation der Patientinnen und der Untersuchungen, die naturgemäß in den Subsystemen unterschiedlich verwaltet werden.

4.8.3 Antwortzeitverhalten der Zugriffe

In der Testimplementierung wurden Antwortzeiten für verschiedene Zugriffsarten in der Abendzeit zwischen 17.00 und 20.00 Uhr gemessen. Zu dieser Zeit war kein nennenswerter Routinebetrieb mehr zu erwarten, der Einflüsse auf die Messung haben konnte. Es wurde die Zugriffszeit für verschiedene Zugriffsarten an drei Systemen - jeweils mit der Originalversion und der eigenen Testimplementierung - gemessen. Jede Messung wurde 10 mal wiederholt. Die eigenen Versionen wurden mit Microsoft Visual C++ Version 5.0 bzw. Microsoft Access Version 5.0 und den entsprechenden ODBC-Treibern entwickelt.

Tabelle 4-3 Antwortzeitverhalten von Originalversion und Testimplementation

Zugriffsart	Durchschnittliche Zugriffszeit [sek], [%]									Bemerkung Notiz
	System 1			System 2			System 3			
	Ori	E	D%	Ori	E	D%	Ori	E	D%	
Suche eines Datensatzes	< 1	< 1	0	< 1	< 1	0	< 1	< 1	0	1)
Laden Befund mit Bildern	50,6	29,5	-42	-	-	-	-	-	-	2)
Starten der Anwendung	12,1	9,5	-21	2,0	3,8	+90	8,0	2,4	-70	selten verwendet
Liste aller Datensätze	-	50,6	-	-	9,7	-	-	4,9	-	3)
Statistik für einen Monat	-	-	-	10,6	19,3	+82	-	-	-	2)
Suche mit Suchmaske	-	< 1	-	-	< 1	-	-	< 1	-	3)

Legende: Ori: Originalsystem
E: Eigene Testimplementierung
D%: Differenz in %

Notiz: 1) Antwortzeit ist unterhalb von Meßgenauigkeiten
2) Zugriffsart nur bei einem System vorhanden
3) Zugriffsart nicht im Originalsystem verfügbar

Tabelle 4-3 faßt die Ergebnisse der Messungen zusammen. Die am häufigsten benutzte Zugriffsart ist der Zugriff auf einen ausgewählten Datensatz. Hierbei konnte wegen der bei beiden Versionen erzielten sehr geringen Zugriffszeiten kein Unterschied festgestellt werden. Die Zeit für den Start einer Anwendung variierte sehr stark: zwischen -70% und +90%. Diese Zugriffsart spielt jedoch keine wesentliche Rolle, da sie nur selten verwendet wird. Die anderen, häufig benötigten Zugriffsarten wie *Liste aller Datensätze*, *Statistik für einen Monat* und *Suche mit Suchmaske* werden leider von den meisten Originalversionen nicht angeboten.

5 Diskussion

In diesem Kapitel wird das Ergebnis der Arbeit unter folgenden Gesichtspunkten diskutiert:

- Zielerreichung,
- Grenze des Konzepts,
- Vergleich mit anderen Konzepten und
- Nutzbarkeit des Konzepts für angrenzende Gebiete.

Anschließend wird ein Resümée gezogen und ein Ausblick auf die Weiterentwicklung des Konzepts gegeben.

Die Diskussion wird vor dem Hintergrund des gesteckten Zieles, des erreichten Ergebnisses und der derzeitigen technologischen Entwicklung geführt. Zur Erleichterung der Diskussion werden an dieser Stelle die wichtigsten Punkte aus den o. g. Bereichen in kurzer Form in Erinnerung gerufen.

Zunächst zum Ziel der Arbeit; die vorliegende Arbeit soll:

- einen allgemeinen Standard für die Kommunikation zwischen medizinischen Informationssystemen vorschlagen und
- eine Überprüfung dieses Standards an Hand von in der Universitätsfrauenklinik und im Institut für Pathologie eingesetzten Befundsystemen vornehmen.

Der vorzuschlagende Standard soll folgende Hauptanforderungen erfüllen:

- maximale Semantik,
- minimale Syntax und
- Praxisnähe.

Das vorgeschlagene Konzept basiert auf einigen Voraussetzungen. Die wichtigsten sind:

- der Zugang zu den Daten eines Subsystems inklusive Dateninhaltsbeschreibung ist möglich und
- die Software-Entwicklungs-Ressourcen für einen virtuellen Befundtreiber sind vorhanden.

Das Konzept hat folgende Philosophie:

- An vorhandenen Subsystemen wird nichts geändert. Es wird lediglich für jedes beteiligte Subsystem ein sogenannter virtueller Befundtreiber hinzugefügt.
- Jedes Subsystem soll für den Teil verantwortlich sein, für den es konstruiert worden ist. Es entsteht ein Netz von kooperierenden Subsystemen, das sogenannte virtuelle Befundsystem.
- Der größte Teil der notwendigen Software wird nur einmal entwickelt und kann dann in allen Einrichtungen durch Anpassung in Betrieb genommen werden. In der jeweiligen Einrichtung muß lediglich die Anpassung in Form des virtuellen Befundtreibers vorgenommen werden.

5.1 Zielerreichung

Die im Rahmen dieser Arbeit gestellte Aufgabe ist mit drei qualitativen Zielen verbunden. Dies sind: *maximale Semantik*, *minimale Syntax* und *Praxisnähe*. Nachdem das Ergebnis der Arbeit vorliegt, wird jetzt überprüft, inwieweit diese Ziele erreicht worden sind.

Das erste Ziel ist „*maximale Semantik*“. Wie bereits erwähnt, ist dies eine qualitative und mehr programmatische Formulierung. Um festzustellen, inwieweit das Ziel erreicht wird, wurde der Begriff „*medizinische Identität*“ postuliert. Dieser Begriff stützt sich auf ein Gedankenspiel. Es ist daher äußerst schwierig, den Grad der Erfüllung dieses Zieles mit einem Meßverfahren direkt zu untersuchen. Man muß hier entweder auf dem Gedankenspiel basierend weitermachen oder eine indirekte, ggf. praktische Methode anwenden. Eine Kombination von beiden erwähnten Verfahren ist ebenfalls denkbar. Im folgenden wird die Erfüllung des Zieles „*maximale Semantik*“ auf beiden Wegen untersucht.

Wir betrachten zunächst das Gedankenspiel. In der Testimplementation wurden die Datensätze bei der Anzeigefunktion nicht vom Sende- in das Empfängersystem transformiert. Das Empfängersystem verwendete die Darstellungsfunktion des Sendesystems, um die Informationen aus dem Sendesystem anzuzeigen. Da die Datensätze nicht transformiert wurden, ist der Datensatz im Empfängersystem - gemäß dem *Postulat der medizinischen Identität* - identisch mit dem Datensatz im Sendesystem. Die Forderung nach maximaler Semantik ist somit zu 100% erreicht. Dies gilt jedoch nur für die Darstellungsfunktion, bei der das Sendesystem voll in Anspruch genommen werden kann. Für andere Funktionen wie Statistiken, bei denen mehrere Datenbestände gegeneinander abgeglichen werden müssen, genügt diese Art der Beweisführung nicht mehr, da die Datensätze von einem System in das andere konvertiert werden müssen, andernfalls würde eine gemeinsame Statistik unmöglich sein. Für solche Funktionen hängt die Güte der Erhaltung der medizinischen Semantik von der Konvertierung der Attribute eines Objekts von einem System in das andere ab. Die Qualität der Konvertierung ist nach dem Konzept nach oben durch die 100%ige Identität, nach unten durch eine untere Schranke begrenzt. Hier ist nur die untere Schranke interessant. Sie wird durch die Kompatibilität der medizinischen Kontexte der beteiligten Systeme bestimmt. Die Qualität der Konvertierung hängt daher von der Qualität der Definition der Kompatibilität der Kontexte ab. In der ersten Näherung könnte man die medizinischen Kontexte untereinander als 100%ig **nicht**-kompatibel deklarieren. Der Befundtreiber wird während der Konvertierung sofort auf die semantikle Transformation - das ist die Konvertierung in ein reines Textformat - umschalten, sobald er eine 100%ige Nicht-Kompatibilität feststellt. In diesem Fall erreicht man zwar keine automatisierte Weiterverarbeitung durch das Empfängersystem, aber die medizinische Identität ist wiederum zu 100% gewährleistet. Durch schrittweise Erhöhung der Kompatibilität der medizinischen Kontexte läßt sich nach und nach eine Verbesserung der Semantik erreichen.

Es ist in der Praxis auch zu erwarten, daß Semantikverluste vorkommen. Dadurch, daß die Systeme die Attribute als Primitiva darstellen, sobald sie mit deren Semantik nicht klarkommen, wird sich ein Semantikverlust gemäß dem asymptotischen Ver-

halten des medizinischen Kontextes in Grenzen halten. Es ist weiterhin damit zu rechnen, daß die Subsysteme wegen eigenen mangelhaften Datenbankdesigns keine genügenden semantikbehafteten Informationen liefern können. Noch 1995 berichtete Hammond, K. W. über solche fehlerhaften Entwürfe [40].

Das zweite Ziel ist die Forderung nach minimaler Syntax. Die Syntax des Konzepts basiert auf dem OO-Konzept. Hierbei wurde lediglich eine Untermenge eines allgemeinen OO-Konzepts ausgenützt. Die Menge der syntaktischen Elemente des Konzepts ist durch das Objektkonzept begrenzt, das folgende Elemente enthält: *Klasse, Objekt, ID, Attribut, Methode, Relation, Nachricht*. Durch rekursive Konstruktion erhält man ein unbegrenztes Netzwerk von semantischen Objekten. Die Syntax kann aus diesem Grund als minimal angesehen werden.

Die Praxisnähe ist das dritte Ziel der Arbeit. Hier bieten sich zwei Teilbereiche an. Der erste betrifft die Entwicklung, Pflege und Administration der Kommunikationsstruktur, der zweite die Anwendung beim Endanwender.

Die Entwicklung, Pflege und Administration des Systems setzt - wie oben erwähnt - zwei Punkte voraus. Zum einen muß ein ungehinderter Zugang zu den Daten eines Subsystems gegeben sein, der eine genügende Dokumentation der Datensätze mit einschließt. Zum anderen müssen Programmierleistungen für die Entwicklung des virtuellen Befundsystems inklusive virtueller Befundtreiber bereitgestellt werden. Für die Testimplementierung waren beide Voraussetzungen erfüllt. Beim späteren Einsatz könnten hier gewisse Probleme auftreten. Eine fehlende Programmierleistung könnte durch eine gute Zusammenarbeit der Einrichtungen der Charité oder durch Einkauf von fremden Leistungen ohne weiteres kompensiert werden. Die Bereitschaft von Herstellern, den Zugang zu den Daten zu ermöglichen, ist in der Regel sehr gering, insbesondere wenn das Subsystem schon gekauft worden ist. Wenn diese Voraussetzung nicht von vornherein vorhanden ist, läßt sie sich nachträglich kaum noch verbessern. Eine Einrichtung, die dieses Konzept nutzen will, müßte daher bei Kaufverhandlungen auf den Zugang zu den Daten bestehen, und zwar nicht nur für die erste Version. Bei jeder weiteren Version muß der Hersteller unaufgefordert die Änderungen bekannt machen, in der Regel schon vor der Installation der neuen Version, damit die Einrichtung genügend Zeit bekommt, den virtuellen Befundtreiber zu aktualisieren. Eine Lösung könnte sein, daß die Einrichtung den virtuellen Treiber als Bestandteil des Kauf- und Pflegevertrags vereinbart. Sie erhält damit automatisch immer den aktualisierten Befundtreiber vom Hersteller.

Dadurch, daß ein Internet-Browser als Client eingesetzt wurde, ergibt sich ein unschätzbare Vorteil. Das System kann an jedem Client eingesetzt werden und der Endanwender hat keinerlei Probleme mit der Bedienung. Der Internet-Browser hat jedoch eine Schwachstelle. Einerseits bietet ein Internet-Browser die Funktion „*Formular*“ mit den Eingabemöglichkeiten wie bei einer Dialogbox oder einem Fenster des Betriebssystems: Editierfelder, Auswahlknöpfe, etc. Man kann praktisch jedes Formular damit konstruieren. Anders als bei einer nativen Fenster- und Dialogbox-Funktion, die unter dem Betriebssystem ausgeführt wird, kennt ein Internet-Browser keine Fensterfunktion auf der Clientseite. Sie verbleiben auf dem Internet-Server. Jedes Formular kann somit auf Korrektheit erst geprüft werden, nachdem es der Endanwender an den Server geschickt hat. Eine interaktive Prüfung z. B. nach

der Eingabe jedes Zeichens ist wegen der sinkenden Performance und der möglichen Netzbelastung nicht opportun. Sie entfällt völlig.

Das Antwortzeitverhalten wurde an Hand einer Testimplementierung am Pränatal-Ultraschall-Subsystem untersucht, indem die Antwortzeit des virtuellen Systems mit der des originalen Subsystems verglichen wurde. Die Tabelle 4-3 zeigt einen Vergleich von 20 Versuchen, die in der Zeit zwischen 17.00 und 20.00 Uhr stattfanden, also in der Zeit, in der der Routinebetrieb am wenigsten abgewickelt wurde. Eine Verschlechterung des Verhaltens konnte weder bei reinem Text noch bei Bildübertragung festgestellt werden. Allerdings wurde bei der Testimplementierung weder eine Datenverschlüsselung noch eine Übertragung über einen WWW-Server vorgenommen. Beim Hinzuschalten der Verschlüsselung und beim zusätzlichen Einsatz eines WWW-Servers muß mit einer Verschlechterung gerechnet werden. Sie läßt sich bei den weiterhin stark sinkenden Hardware-Preisen durch den Einsatz von leistungsfähigeren Rechnern und durch Optimierung der Softwarealgorithmen leicht kompensieren bzw. verbessern.

5.2 Grenzen des Konzepts

Das hier vorgestellte Konzept eignet sich hervorragend für die Anzeige von interdisziplinären, medizinischen Informationen (s. oben). Die flächendeckende Umsetzung des Konzepts ist ohne weiteres möglich, wenn die zwei o. g. Voraussetzungen erfüllt werden können.

Eine Erweiterung des Konzepts um die Möglichkeit, Daten in ein Subsystem einzuspeisen, ist nicht ohne weiteres machbar, allerdings in der Regel auch nicht gewollt. Der Grund hierfür ist simpel. Bei der Anzeige von Befunden liefert das Subsystem Informationen über den virtuellen Befundtreiber an das virtuelle Befundsystem. Der Zugriff auf den Datenbestand des Subsystems erfolgt also nur lesend. Die Gefahr, die Integrität des Datenbestandes zu zerstören, besteht hier nicht. Das wird anders, wenn man Daten aus dem virtuellen Befundsystem in ein Subsystem einspeisen will. Hier ist die Gefahr groß, die Integrität des Datenbestandes des Subsystems durch Programmierfehler zu zerstören. Viele Subsysteme sind so komplex strukturiert, daß es nicht möglich ist, für die Programmier- und Testphase ein Test-Subsystem nachzubauen. Ein Beispiel hierfür ist das System PIA in der Abteilung Pränatale Diagnostik und Therapie der Frauenklinik. Auf der Server-Ebene umfaßt das System drei Server: einen Novell-Fileserver, einen Watcom-SQL-Datenbankserver und einen DOS-Server für die Steuerung des Archivs mit einer Jukebox, die die WORM-Medien enthält. Es sieht natürlich anders aus, wenn der Hersteller des Subsystems die Entwicklung und die Pflege des virtuellen Befundtreibers übernimmt. Da dies nicht immer der Fall sein wird, bleibt die Einspeisung der Subsysteme mit den Daten aus dem virtuellen Befundsystem nur den Subsystemen vorenthalten, über die die jeweilige Einrichtung die volle Kontrolle hat. Das ist z. B. der Fall, wenn das Subsystem inklusive dem Befundtreiber von der Einrichtung selbst entwickelt ist oder der Hersteller diese Arbeit vollständig verantwortlich übernimmt.

Das nächste Problem betrifft die Identifikation der Objekte im virtuellen Befundsystem. Das Konzept sieht auf jeder Ebene ein Schlüsselssystem vor, das die weltweit eindeutige Identifikation ermöglicht. Dies ist jedoch nur möglich, wenn alle beteiligten Einrichtungen sich auf eine einheitliche Organisationsform der Schlüsselvergabe

einigen können. Man könnte sich eine Organisation analog zu den Internet Domain Name Services vorstellen. Technisch stellt das Vorhaben also kein Problem dar. Organisatorisch ist zu erwarten, daß die Realisierung mit vielen Schwierigkeiten verbunden sein wird. Wenn man die Daten nicht nur im Intranet-, sondern auch im Internet-Bereich verarbeiten will, so werden die nationalen Datenschutzgesetze eine große Rolle spielen, die leider von Land zu Land sehr verschieden sind. In der Bundesrepublik darf z. Z. beispielsweise kein Schlüsselsystem eingesetzt werden, das Rückschlüsse auf die betroffene Person erlaubt. Dagegen wird in Schweden so vorgegangen, daß jede Person durch eine landesweit eindeutige Personalnummer erfaßt wird. Diese Nummer kann und wird auch dann bei allen Institutionen wie Behörden, Banken, Versicherungen, etc. zur Identifizierung eingesetzt.

Das Konzept sieht vor, daß die Subsysteme sich nicht um die administrativen Patientendaten kümmern. Sie müssen sich auf die zentrale Patientenverwaltung der übergeordneten Einrichtung, in diesem Fall der Charité, verlassen. All die Probleme, die im Abschnitt *1.1.2 Patientenbezogene Probleme* beschrieben sind, müssen in der zentralen Patientenverwaltung also gelöst sein. Hierzu kann z. Z. für die Charité keine Aussage gemacht werden, da eine Umstellung der zentralen Patientenverwaltung gerade erst erfolgt ist.

Das Problem des Urheberrechts der forschungsrelevanten Daten wurde bei der Testimplementierung nicht gelöst. Es scheint z. Z. keine technisch geeignete Lösung zu geben. Hier kann daher nur empfohlen werden, eine organisatorische Lösung zu suchen, beispielsweise durch inter-institutionelle Vereinbarungen.

Last, but not least muß das Problem des Datenschutzes angesprochen werden. Im Konzept wurde dieses Problem ebenfalls nicht behandelt. Es würde den Rahmen der vorliegenden Arbeit sprengen. Hinzu kommen noch z. Z. ungelöste juristische Probleme. Man braucht sich nur vorzustellen, daß alle medizinischen Einrichtungen wie Krankenhäuser, Krankenkassen, Ärzte, Apotheker, etc. miteinander verbunden wären und der Informationsaustausch auf einem hohen semantischen Niveau ohne weiteres funktionieren würde. Ein solches Netz bietet eine unbegrenzte Anzahl von Angriffsflächen für einen potentiellen Datenmißbrauch. Durch geschickt formulierte Abfragen kann ein Angreifer nahezu alle Informationen und alle möglichen Schlußfolgerungen daraus aus dem Netz beziehen. Bevor ein solches Konzept flächendeckend realisiert werden soll, müssen sich die Verantwortlichen, sowohl medizinisch als auch politisch, sehr gründlich Gedanken über den Datenschutzaspekt medizinischer Informationen machen, da die hier zu lösenden Probleme zum größten Teil nicht mehr technischer Natur sind (Iversen, K. R. [49]). Dadurch, daß dieses Problem relativ konflikträchtig ist, müssen alle Beteiligten zusammenarbeiten, um eine für alle Seiten gangbare Lösung zu finden (Lawrence, L.M. [58]).

5.3 Vergleiche mit anderen Konzepten

Die z. Z. bekannten Konzepte zur Verbesserung der Kommunikation zwischen medizinischen Subsystemen lassen sich prinzipiell in zwei Kategorien einordnen. Nachfolgend wird der Vergleich an Hand eines konkreten Beispiels vorgenommen. Wir betrachten hierzu die Geburtshilfe in der Universitätsfrauenklinik der Charité, in der Zugriffe auf Informationen aus folgenden Systemen ständig benötigt werden:

- das Ultraschallbefund- und -archivierungssystem (PIA),
- die geburtshilfliche Datenbank (GebDat),
- die Punktionsdatei und
- das Fetalautopsie-Modul (Autopsy-Modul).

Sollten beispielsweise Komplikationen bei einer Geburt zu erwarten sein, so werden aus der Sicht des Geburtshelfers häufig folgende Informationen benötigt:

- Befunde aus vorangegangenen Geburten, falls vorhanden (GebDat),
- Befunde aus der Pränatalen Diagnostik, ggf. für vorangegangene Schwangerschaften (PIA),
- Befunde aus invasiven Maßnahmen (Punktionsdatei) und
- Vergleichbare Befunde aus der Fetalautopsie, falls vorhanden (Autopsy).

Andere Informationen wie aus der Genetik, der Neonatologie oder anderen Laboratorien können ebenfalls notwendig sein. Einige Fragenstellungen könnten beispielsweise wie folgt aussehen:

Liegen bei dieser Patientin bereits Komplikationen aus vorangegangenen Schwangerschaften und Geburten vor? Sind Schwangerschaftsabbrüche vorgenommen worden, falls ja, welche Ergebnisse liegen in der Fetalpathologie vor? Was wurde in der pränatalen Diagnostik festgestellt, was wurde bei der Punktion gefunden? Hat die Autopsie die Verdachtsdiagnose bestätigt oder nicht?

Bei diesen Fragestellungen müssen die Inhalte der vier Datenbanken miteinander verknüpft werden. Die erste Verknüpfung wird z. B. über die Patientin, die zweite Verknüpfung über die Schwangerschaft und die dritte über die verschiedenen Untersuchungen hergestellt.

Kategorie1: Alles aus einer Hand

Bei diesem Konzept wird versucht, alle Subsysteme in den Einrichtungen einheitlich zu definieren und zu realisieren. Das System kann durch einen Auftrag an einen großen Systemanbieter oder durch eine interne Software-Entwicklungs-Gruppe realisiert werden. Den ersten Weg ging das Donauspital in Österreich [110], die zweite Lösung wählte das amerikanische Department of Veterans Affairs und das Brigham and Womens Hospital in Boston.

Das Donauspital vergab einen Großauftrag an Firma Siemens, die alle Subsysteme für die Patientenverwaltung und die Handhabung von bildgebenden Verfahren in der Radiologie realisiert [110]. Der Austausch medizinischer Informationen mit anderen Systemen ist hier jedoch ausgeklammert. Diese sind jedoch der Schwerpunkt des vorliegenden Konzepts. Somit ist ein Vergleich zwischen den beiden Konzepten nur bedingt zulässig. Hierfür wird der fiktive Fall angenommen, daß das Spital nicht nur die Handhabung der Bilddaten, sondern auch die Erfassung und Bearbeitung von inhaltlichen medizinischen Daten in Auftrag vergäbe. So ist anzunehmen, daß das Konzept für eine gewisse Zeit sehr gut funktioniert - so lange kein Bedarf an fremder Software bestünde. Sobald ein Stück fremde Software eingesetzt wird, ließe sich das Konzept nicht mehr aufrechterhalten. Es ist bei der rasanten Entwicklung der medizinischen Hard- und Software zu erwarten, daß diese Karenzzeit nicht sehr lange dauern wird. Denn es existiert kein Software-Hersteller, der in allen

medizinisch-informatischen Bereichen gleichermaßen hochspezialisiert ist. Ein gut geführtes Krankenhaus ist immer gezwungen, die neuen Techniken einzusetzen, um eine zeitgemäße Krankenversorgung zu gewährleisten. Das Krankenhaus muß also fremde Software über kurz oder lang beziehen.

Setzt man dagegen eigene Software-Entwickler ein, wie es das Brigham and Womens Hospital, Boston, praktiziert, so entstehen andere Probleme, wobei das Problem weiter besteht, fremde Software einzusetzen, denn die eigenen Entwickler können auch nicht in allen medizinischen Gebieten erstklassig sein. Dazu kommt das organisatorische Problem beim Einsatz eigener Entwickler. Dieses Problem besteht in der Frage, wie die Entwickler organisiert werden. Werden sie zentral zu einer Abteilung zusammengefaßt, so verlieren sie mit der Zeit den Kontakt zu Anwendern und damit ein Gefühl für deren Probleme. Werden sie der jeweiligen Einrichtung zugeordnet, so kann bei größeren Projekten keine effiziente Software-Entwicklungs-Gruppe organisiert werden. Eine ausgewogene organisatorische Form hängt in bedeutendem Maße von den beteiligten Persönlichkeiten ab. Dies kann sich jedoch schlagartig bei Personalwechsel ändern. Eine Studie der Universität Iowa zeigte, daß eine durch Anwender vorangetriebene Eigenentwicklung kosteneffektiver ist und den speziellen Anwenderanforderungen besser gerecht wird (Flanagan, J. R., et al. [32]).

Eine Kombination aus eigenen Entwicklern und virtuellem Befundsystem erscheint am meisten erfolversprechend zu sein. Die Softwareentwicklung kann klein gehalten werden. Sie muß sich lediglich um die Pflege des virtuellen Befundsystems und die Entwicklung neuer virtueller Befundtreiber kümmern. Mit einer kleingehaltenen, aber auf die Integration von Fremdsoftware hochspezialisierten Software-Entwicklungs-Gruppe läßt sich auch viel besser mit den Softwareherstellern über den Zugang zu den Daten und zu den Dokumentationen verhandeln.

Wären die im obigen Beispiel angegebenen Subsysteme aus einer Hand, so bestünde die Möglichkeit, mit einer Hardware- bzw. Systemsoftware alle Informationen anzuzeigen. Ob der Anwender hierbei nur eine Anwendung benutzen muß, um auf diese Informationen zuzugreifen und nicht vier Programme gleichzeitig aufrufen und die gleiche Anfrage stellen muß, ist damit noch nicht gewährleistet. In der Praxis wird es so aussehen, daß ihm bestimmte vorgefertigte Anfragen angeboten werden. Falls er eine neue Fragenstellung zu bearbeiten hat, dann versagt ein solches Konzept. Dieses Konzept versagt ebenfalls, wenn ein fremdes System neu hinzukommt.

Kategorie 2: „Intranet-Lösungen“

Seitdem das Internet zunehmend beliebter wird, läßt sich eine „Aufbruchsstimmung“, ja sogar „Hysterie“, in Richtung Intranet verzeichnen. Betrachtet man diese sogenannten Intranet-Lösungen genauer, so stellt sich folgendes heraus. Diese Intranet-Lösungen verwenden die Internet-Technologie, um den herkömmlichen Client zu ersetzen. Dahinter steht in der Regel ein Datenbankserver, der über eine Verbindung z. B. durch das Produkt dbWeb der Firma Microsoft mit einem WWW-Server verfügt. Somit bieten diese Lösungen dem Anwender den Vorteil, von überall auf die Daten zugreifen zu können, wenn er nur einen Internet-fähigen Rechner besitzt.

Es entstünde dann eine Anzahl von Internet-fähigen Befundviewern, die jedoch nichts miteinander zu tun hätten. Der Anwender könnte dann beispielsweise einen Internet- Browser viermal mit vier verschiedenen Homepages aufrufen, um die vier obigen Programme nacheinander anzusprechen und das Ergebnis einer Anfrage als Eingabe für die nächste manuell zu formulieren. Dadurch, daß die einzelnen Subsysteme trotz gemeinsamer Bedienoberfläche keine gemeinsamen Schlüssel für eine Patientin und keine gemeinsame Definition einer Untersuchung haben, wird der Anwender Schwierigkeiten bei Nutzung der Befundviewer bekommen. Er muß beispielsweise wissen, daß der in der Geburtshilfe geläufige Begriff *Schwangerschaft* in der Fetalpathologie keine Entsprechung findet. Dort werden alle Untersuchungen als *Fälle* mit einer Eingangsnummer abgespeichert.

Das Problem der unterschiedlichen Semantik in den Subsystemen wird dadurch nicht gelöst. Läßt man den Vorteil eines Internet-Clients außer acht, so bleibt alles „beim alten“. Semantisch wird mit diesen Lösungen kein Problem gelöst.

Mit dem AMICI-Konzept wird beispielsweise der erste Teil solcher Anfrage wie folgt formuliert (an SQL angelehnt):

Frage: Hatte diese Patientin (lokale ID: 1234567) bereits Komplikationen bei vorangegangenen Schwangerschaften und Geburten?

```
SELECT ALL OBJECTS FROM prae natal, gebdat, punktion, autopsy
WHERE
    VALUE (gebdat.treatmentset.local-id) = „1234567“
AND
    VALUE (prae natal.treatmentset.pat-id) = VALUE (gebdat.treatmentset.pat-id) = VALUE
    (punktion.treatmentset.pat-id) = VALUE (autopsy.treatmentset.pat-id)
AND
    (VALUE (prae natal.treatmentset.finding) NOT NULL OR
    VALUE (gebdat.treatmentset.finding) NOT NULL OR
    VALUE (punktion.treatmentset.finding) NOT NULL OR
    VALUE (autopsy.treatmentset.finding) NOT NULL);
```

Die Wörter *prae natal*, *gebdat*, *punktion*, *autopsy* bezeichnen die medizinischen Kontexte der Subsysteme. Das Wort *treatmentset* bezeichnet die Menge aller virtuellen Behandlungen in einem Subsystem. *pat-id* ist ein Attribut von *treatment* und der Operator *VALUE* liefert den Wert eines Attributs zurück. Die Wörter *SELECT*, *ALL*, *OBJECTS*, *FROM*, *WHERE*, *VALUE*, *AND*, *OR*, *NOT*, *NULL* und Sonderzeichen wie Gleichheitszeichen, Komma, Semikolon, Klammern sind reservierte Wörter bzw. Trennzeichen.

Diese Abfrage muß der Anwender nicht formulieren. Ihm wird eine einheitliche graphisch geführte Bedienung angeboten, die erst durch die minimale Syntax möglich wird, ohne daß der Anwender die interne Struktur der Daten kennen muß. Die graphische Bedienung setzt die Anfrage intern in die obige Formulierung um. Einen Scanner und einen Parser hierfür zu entwickeln, stellt keine besondere Herausforderung dar.

Wird die Abfrage genauer betrachtet, dann können wir feststellen, daß der Anwender die Patientenummer in einem Fremdsystem nicht kennen muß. Er gibt die lokale Nummer ein. Die Übersetzung der lokalen Nummern in das globale System übernimmt das System. Auch die in verschiedenen Subsystemen

unterschiedlich organisierten Datenbanken werden durch das System vereinheitlicht. Es existiert nur noch eine kleine Anzahl von vereinheitlichten Objekten wie Behandlung (*treatment*), Befund (*finding*) sowie die Menge aller Behandlungen (*treatmentset*).

5.4 Nutzbarkeit des Konzepts für angrenzende Gebiete

Der Schwerpunkt der Anwendung des vorgeschlagenen Konzepts liegt im sogenannten virtuellen Befund, bei dem Informationen aus allen angeschlossenen Subsystemen an einer Stelle zur Verfügung gestellt werden. Dieses Merkmal bietet sich daher für interdisziplinäre Veranstaltungen an. Solche Veranstaltungen haben in der Charité Tradition. Beispiele hierfür sind die Tumorkonferenz und die interdisziplinären Konsiliarsitzungen. Die Tumorkonferenz findet wöchentlich statt und behandelt interdisziplinäre Fragestellungen, die die Tumore betreffen. Die pränatale Konsiliarsitzung findet ebenfalls wöchentlich unter der Federführung der Frauenklinik statt, an der eine Reihe von Kliniken und Instituten teilnehmen: Frauenklinik, klinische und Zytogenetik, Radiologie, Neonatologie, Kinderklinik und Pathologie. Hier werden Problemfälle in der Geburtshilfe interdisziplinär besprochen und Maßnahmen getroffen. Ähnliche Veranstaltungen finden in anderen Kliniken und Instituten statt.

Auf solchen Veranstaltungen könnte sich der Einsatz des Systems mit dem virtuellen Befundviewer als fruchtbar erweisen.

5.5 Fazit

Für die Kommunikation innerhalb einer Einrichtung wie der Charité muß ein Standard geschaffen werden, der auf einem sehr hohen Abstraktionsniveau angesiedelt sein muß. Dies ist notwendig, um den Standard von einer bestimmten Plattform bzw. bestimmten Entwicklung unabhängig zu machen. Andernfalls kann der Standard von neuen Entwicklungen überholt werden. Er kann u. U. technisch bereits veraltet sein, bevor irgendeine Realisierung abgeschlossen ist.

Der Standard beschränkt sich auf den Austausch medizinischer Daten zwischen den medizinischen Subsystemen bzw. zwischen einem medizinischen Subsystem und einem administrativen Subsystem. Die Philosophie des Standards basiert auf folgenden Prinzipien:

- Der Standard legt ein Netz fest. An jedem Knoten des Netzes wird ein medizinisches Subsystem über einen sogenannten virtuellen Befundtreiber angeschlossen. Zwischen den Knoten werden die Informationen ausgetauscht.
- Jedes Subsystem kann sehr proprietär sein. Wenn es jedoch über den Treiber an das Netz angeschlossen werden kann, so ist es aus der Sicht des Netzes ein Lieferant von Informationen für das Netz, also für die restlichen Netzbenutzer. Es kann ebenfalls Informationen aus dem Netz beziehen und dem lokalen Anwender zur Verfügung stellen.
- Die medizinische Semantik wird durch zwei Maßnahmen gewährleistet. Die erste besagt, daß der verantwortliche Knoten für die Interpretation und Darstellung des

Inhaltes sorgt, denn er ist der Knoten, der mit der Semantik des betroffenen Objekts am besten umgehen kann. Die zweite Maßnahme umfaßt die Definition eines medizinischen Kontextes. Jeder Knoten stellt seinen Kontext zur Verfügung. Wird ein Objekt aus einem bestimmten Kontext in einen anderen importiert, so kann an Hand der Kontexte geprüft werden, ob ein Austausch ohne Semantikverlust vorgenommen werden kann. Wenn nicht, werden die Informationen als reiner Text bzw. reine Graphik übertragen.

Der hier vorgeschlagene Standard läßt sich ohne weiteres mit den vorhandenen technischen Hilfsmitteln der Charité implementieren.

Es sind jedoch einige organisatorische Voraussetzungen erforderlich. Die erste Voraussetzung betrifft den Zugriff auf und die Interpretationsfähigkeit der Daten, die meist in kommerziell erworbenen medizinischen Subsystemen abgelegt sind. Für jedes Subsystem müssen ggf. Verhandlungen mit dem Hersteller geführt werden, damit die Beschreibung der Daten und evtl. die Zugriffssoftware zur Verfügung gestellt werden können. Desweiteren wird vorausgesetzt, daß für jedes Subsystem, sofern die o. g. Dokumente vorliegen, ein virtueller Befundtreiber erstellt wird. Dies ist nur möglich, wenn das medizinische und das informatische Know-How in der jeweiligen Einrichtung vorhanden sind.

5.6 Ausblick

Der Datenaustausch zwischen medizinischen Einrichtungen ist ein wichtiger Schritt zur Verbesserung der medizinischen Informationsverarbeitung. Er stellt jedoch nur den ersten Schritt dar. Hierbei werden die interdisziplinären Informationen zusammengefaßt, aufbereitet und angezeigt. Sie zu verarbeiten, obliegt weiterhin ausschließlich dem Arzt mit seinen Fähigkeiten und Erfahrungen. Der zweite Schritt in die Richtung der automatisierten Verarbeitung von medizinischen Informationen führt ohne Zweifel in die Richtung Entscheidungshilfe. Sie beinhaltet die wissensbasierte Verarbeitung und die Bildverarbeitung. Während die Bildverarbeitung in der Charité sehr früh und erfolgreich praktiziert wird, beispielsweise durch die Arbeitsgruppe AMBA Automatisierte Mikroskopbildanalyse im Institut für Pathologie der Charité und die Arbeitsgruppe AG Imaging in der Charité, an der Wissenschaftler aus verschiedenen Einrichtungen teilnehmen, wird das Fachgebiet wissensbasierte Verarbeitung (künstliche Intelligenz) noch vernachlässigt. Hier und da existieren vereinzelt Ansätze in dieser Richtung.

Nachdem das Intranet zur Verfügung steht, eröffnen sich hier einige neue und neuartige Möglichkeiten für die medizinische Informatik. Zum einen ist dies die Telemedizin, zum anderen der Einsatz intelligenter und autonomer Software.

Die Telemedizin wird in den USA in vielen Fällen bereits in der Routine eingesetzt. Das Hauptmotiv hierfür war rein ökonomischer Natur. Dort findet sich - im Gegensatz zur Bundesrepublik - quasi der Nährboden für Telemedizin als Sonderfall von WAN-Anwendungen. Die Telekommunikationskosten sind in der Bundesrepublik noch immer eine oft unüberwindbare Hürde. Dazu kommen noch ungeklärte rechtliche Verhältnisse bezüglich der Verschlüsselung, der Autorisierung und der Netzsicherheit [108]. Häufig fehlen auch noch Anwendungen, die sich nahtlos in den Arbeitsablauf eines Arztes integrieren lassen. Beispielsweise können die meisten Telemedizin-

Programme auf die Datenbanken der anwendenden Einrichtung nicht zugreifen, um die bereits vorhandenen Patientendaten und Vorbefunde in eine telemedizinische Sitzung automatisch einzubauen. Der Anwender müßte alles noch einmal mühsam eingeben. Diese Probleme werden in einigen Jahren jedoch gelöst sein, so daß die Telemedizin auf der Basis eines funktionsfähigen Intranets flächendeckend eingesetzt werden kann. In der Universitätsfrauenklinik wird Teleultraschall bereits in der Routine verwendet, allerdings nur innerhalb des Klinikums (Hufnagl, P. [47]).

Die intelligente und autonome Software stammt aus den Forschungslaboren der künstlichen Intelligenz (KI), die heute noch wegen der anfangs zu hoch gesteckten Ziele an einem schlechten Ruf leidet. Die KI ist heute jedoch besser als ihr Ruf. Durch die Schaffung kleinerer Projekte mit praktischer Zielsetzung entsteht eine Reihe nutzbarer Anwendungen, auch in der Medizin. Carenini, G., benutzte ein UMLS-Semantic-Network in Verbindung mit der KI und kam zu dem Schluß, daß die KI für einen Einsatz in der Medizin Vorteile bringen würde, obwohl KI noch schwierig zu handhaben sei [13]. Hybridkonzepte, bei denen KI und herkömmliche Methoden sich gegenseitig ergänzen, wurden an vielen Stellen entwickelt (Moser, W., et al. [68]), (van Ginneken, A.M., et al. [35]). Ferner wurden im Zusammenhang mit dem Internet neue Ansätze entwickelt. Der Net-Agent, auch Net-Robot genannt, ist einer dieser neuen Ansätze, der in seiner Agentenwelt „lebt“, autonom „agiert“ und einen bestimmten Auftrag im Netz ausführt. Die Assistenten von MS Office 97 stellen beispielsweise den ersten Schritt in Richtung Software-Assistenten dar. Es ist daher wahrscheinlich, daß diese Gattung intelligenter Software auch in die Medizin Einzug halten wird. Man denke nur an die Überinformationen in einem Intranet/Internet oder an die Schwierigkeiten, eine akkurate Anfrage zu formulieren, um die gewünschte Antwort aus dem Intranet/Internet zu bekommen. Man bekommt mitunter mehrere Hunderttausend sogenannte Hits.

Ungeachtet der noch fehlenden Ergebnisse in Richtung semantischer Bild- und wissensbasierter Verarbeitung läßt sich der medizinische Datenbestand, der durch das virtuelle Befundsystem entsteht, ohne weiteres mit Hilfe der heutigen Multimedia-Technik zu wirkungsvollen Lehrmitteln aufbereiten. Die Werkzeuge hierfür sind heute bereits ausgereift, finanziell erschwinglich und für den nicht-versierten Anwender so leicht zu bedienen, daß die einzigen Hindernisse nur noch in einer fehlenden pädagogischen Aufbereitung des Materials liegen.

Die von vielen medizinischen Informatikern diskutierte elektronische Krankenakte, die im englischsprachigen Raum oft als EPR Electronic Patient Record oder CPR Computerized/Computer-based Patient Record sind mit den medizinischen Standards wie ICD, ICPM, HL7, DICOM, SNOMED sowie mit deren Integrationskonzepten in naher Zukunft keine bloße Vision mehr. Bomba, B., et al. verlangten 1995 für Australien eine landesweite integrierte medizinische Informationsinfrastruktur [8].

Heller, E. E., propagierte CPR sogar als Vision im Gegensatz zu herkömmlichen Krankenhaus- (HIS) oder medizinischen Informationssystemen (MIS) [43].

6 Zusammenfassung

In der Charité werden in verschiedenen Kliniken und Instituten zahlreiche EDV-Systeme für die Erfassung und Auswertung von Befunden, inklusive Patienten-, Diagnose- und Therapie- sowie Anamnese-Daten eingesetzt. Diese hier als medizinische Subsysteme bezeichneten, meist kommerziell erworbenen EDV-Systeme lassen bisher keinen bzw. nur einen sehr beschränkten Datenaustausch zu. Die Beschränkung rührt von fehlenden Import-/Export-Werkzeugen und Standards her. Die fehlenden Standards betreffen sowohl die Formatfestlegung (Syntax) als auch die Inhaltsbeschreibung (Semantik).

Die im Rahmen der vorliegenden Arbeit durchgeführte Recherche ergab, daß die technologische Entwicklung bereits einen Satz von Werkzeugen für die Integration unterschiedlicher Systeme, auf unterschiedlichen Plattformen zur Verfügung stellt. Als Beispiel werden hier die Internet-Technologie mit den entsprechenden Protokollen und die objekt-orientierten Technologien wie C++, Java, COM, CORBA genannt. Die Anforderungen spezieller Anwendungen können durch die medizinischen Standards in vielen Fällen erfüllt werden. Für die Integration von getrennten, kommerziell erworbenen Subsystemen lassen sie sich nicht ohne weiteres einsetzen. Der wichtigste Grund hierfür ist die sehr komplexe Semantik in der Medizin. Hinzu kommt, daß die Standards durchweg komplexe Syntax verwenden, so daß eine software-technische Umsetzung zu kostspielig wäre.

Ein Standard für den inhaltlichen Austausch von medizinischen Informationen muß daher folgende Anforderungen erfüllen: minimale Syntax, maximale Semantik und Praxisnähe. Der Standard muß auf einem hohen Niveau definiert sein. Ähnlich wie HL7 muß der Standard auf der Schicht 7 der ISO-OSI angesiedelt sein. Er muß plattform- und netzwerkprotokoll-unabhängig sein.

In der vorliegenden Arbeit wird ein solcher Standard vorgeschlagen. Die Philosophie des Standards basiert auf folgenden Prinzipien:

- Der Standard legt ein Netz fest. An jedem Knoten des Netzes wird ein medizinisches Subsystem über einen sogenannten virtuellen Befundtreiber angeschlossen.
- Jedes Subsystem kann sehr proprietär sein. Wenn es jedoch über den Treiber an das Netz angeschlossen werden kann, so ist es aus der Sicht des Netzes ein Lieferant von Informationen für das Netz, also für die restlichen Netzbenutzer.
- Die medizinische Semantik wird durch zwei Maßnahmen gewährleistet. Die erste besagt, daß der verantwortliche Knoten für die Interpretation und Darstellung des Inhaltes sorgt. Die zweite Maßnahme umfaßt die Definition eines medizinischen Kontextes. Jeder Knoten stellt seinen Kontext zur Verfügung.

Der vorgeschlagene Standard verwendet zahlreiche andere verbreitete Standards, zu nennen sind: HL7, DICOM, SNOMED und HTML/XML.

Anhand von Testimplementationen in der Frauenklinik, in der Strahlenklinik und im Institut für Pathologie wurde untersucht, inwieweit ein solcher Standard aus technischer Sicht realisierbar ist und den medizinischen Anforderungen genügen kann.

Aus der Frauenklinik wurden das Befund- und Bildarchivierungssystem PIA der Fa. ViewPoint GmbH, Gilching, die geburtshilfliche Datenbank und die Punktionsdatei, aus dem Institut für Pathologie wurde der Modul Fetal Autopsy, eine Eigenentwicklung des Instituts und aus dem Berliner Tumorzentrum Charité der HL7-Server als Beispielanwendung eingesetzt.

Die Forderung nach maximaler Semantik konnte an Hand der Testimplementationen als erfüllt bezeichnet werden, solange das Sendesystem (Server) alle Operationen übernimmt. Die Einführung einer neuen Semantik gestaltet sich durch die Entwicklung eines virtuellen Befundtreibers relativ einfach und verursacht keine unüberwindbare finanzielle Hürde. Die vorhandenen Subsysteme lassen sich weiter einsetzen. Eine komplette Neuentwicklung ist nicht erforderlich.

Die Syntax ist mit der Definition von wenigen Elementen wie Objekten, Klassen, ID, Relationen, Attributen als einfach zu bezeichnen. Die Rekursion erlaubt unbegrenzte Kombinationsmöglichkeiten.

Durch die Verwendung eines Standard-Internet-Browsers läßt sich das System plattformunabhängig einsetzen. Der Routinebetrieb wird durch die Einführung des Systems nicht gestört. Die Forschungsaktivitäten wurden mit Hilfe der noch sehr rudimentären Testimplementationen verstärkt. Der hauptsächliche Nutzen lag in der Möglichkeit, Daten aus verschiedenen Disziplinen zusammenzuführen und so abzugleichen. So ließen sich viele neue Erkenntnisse gewinnen, die erst durch den Datenabgleich von mehreren Datenbeständen sichtbar werden konnten.

Da der vorgeschlagene Standard die heute und wahrscheinlich in Zukunft verwendeten Standards einschließt und nur geringen Entwicklungsaufwand verursacht, läßt er sich auf weitere Einsatzgebiete ohne weiteres übertragen.

Es wurde ebenfalls deutlich, daß der Routineeinsatz eines solchen Systems die Lösung einiger noch nicht gelöster Probleme voraussetzt; zum einen ist dies der Schutzaspekt der Daten. Hierbei muß sowohl der Schutz der personenbezogenen Daten als auch der Schutz des geistigen Eigentums berücksichtigt werden. Zum anderen macht die Entwicklung eines virtuellen Befundtreibers einiges an Entwicklungsressourcen erforderlich, die nicht in jeder medizinischen Einrichtung vorhanden sind.

7 Literaturverzeichnis

Im folgenden wird die in der Arbeit verwendete Literatur zusammengestellt. Die rasche Entwicklung auf dem Gebiet der Kommunikation, Netzwerke, Integration, medizinische Standards machte die Recherche nicht nur in den herkömmlichen Medien wie in Büchern, Zeitschriften, sondern auch in neuen Medien wie Internet und Microsoft Developer Network notwendig.

Da die im Internet gefundene Literatur nicht nach den üblichen bibliographischen Konventionen aufgebaut ist, werden hierfür eigene Konventionen festgelegt. Jeder Eintrag in der Internet-Literatur wird eindeutig durch die sogenannten URL (Universal Resource Location), das Datum und die Uhrzeit des Downloads angegeben. Falls die Internet-Seite keinen Autor enthält, wird der Name des WWW-Servers bzw. der Einrichtung verwendet.

Um die Handhabung zu erleichtern, wird das Literaturverzeichnis in zwei Teile unterteilt: Print-Medien/CD-ROM und Internet-Seiten.

A. Print-Medien/CD-ROM

- [1] ACR-NEMA. (1994). ACR-NEMA Digital Imaging and Communications in Medicine (DICOM) Standard V3.0. American College of Radiology, Network Communications Support for Message Exchange.
- [2] Adelhard, K., Eckel, R., Holzel, D., Tretter, W. (1995). A prototype of a computerized patient record. *Comput Methods Programs Biomed*, 48, 115-119.
- [3] Albitz, P., Liu, C. (1994). *DNS and BIND in a Nutshell*. O'Reilly & Associates, Inc. Sebastopol, CA, USA.
- [4] Ash, J. (1995). Cross-site study of the implementation of information technology innovations in health sciences centers. *Proc Annu Symp Comput Appl Med Care*, 795-799.
- [5] Barsalou, T. (1989). An object-based architecture for biomedical expert database systems. *Comput Methods Programs Biomed*, 30, 157-168.
- [6] Bell, D.S., Pattison Gordon, E., Greenes, R.A. (1994). Experiments in concept modeling for radiographic image reports. *J Am Med Inform Assoc*, 1, 249-262.
- [7] Bidgood, W.D., Jr., Horii, S.C. (1992). Introduction to the ACR-NEMA DICOM standard. *Radiographics*, 12, 345-355.
- [8] Blackwell Verlag. (1992). *ICPM*. Berlin Wien: Blackwell Verlag. ICPM. 267.
- [9] Bomba, B., Cooper, J., Miller, M. (1995). Working towards a national health information system in Australia. *Medinfo*, 8 Pt 2, 1633.
- [10] Branger, P., van't Hooft, A., van der Wouden, H.C. (1995). Coordinating shared care using electronic data interchange. *Medinfo*, 8 Pt 2, 1669.
- [11] Brockschmidt, K. 1997. *Developing Applications with OLE 2.0*. Microsoft Developer Network CD 97.
- [12] Brown, N., Kindel, C. 1996. *Distributed Component Object Model (DCOM) Binary Protocol*. Microsoft Developer Network CD 97.
- [13] Carenini, G., Moore, J.D. (1993). Using the UMLS Semantic Network as a basis for constructing a terminological knowledge base: a preliminary report. *Proc Annu Symp Comput Appl Med Care*, 725-729.
- [14] Cerutti, S. (1980). Semantic models in medical record data-bases. *Med Inf Lond*, 5, 215-226.
- [15] Chechile, R.A., Eggleston, R.G., Fleischman, R.N., Sasseville, A.M. (1989). Modeling the cognitive content of displays. *Hum Factors*, 31, 31-43.

- [16] Chute, C.G., Cohn, S.P., Campbell, K.E., Oliver, D.E., Campbell, J.R. (1996). The content coverage of clinical classifications. For The Computer-Based Patient Record Institute's Work Group on Codes & Structures. *J Am Med Inform Assoc*, 3, 224-233.
- [17] Cimino, J.J., Aguirre, A., Johnson, S.B., Peng, P. (1993). Generic queries for meeting clinical information needs. *Bull Med Libr Assoc*, 81, 195-206.
- [18] Cimino, J.J., Barnett, G.O. (1990). Automated translation between medical terminologies using semantic definitions [published erratum appears in *MD Comput* 1990 Jul-Aug;7(4):268]. *MD Comput*, 7, 104-109.
- [19] Cornet, R., van Mulligen, E.M., Timmers, T. (1994). A cooperative model for integration in a cardiology outpatient clinic. *Proc Annu Symp Comput Appl Med Care*, 590-594.
- [20] Cramer, P. CAP. 1996; The Systematized Nomenclature of Human and Veterinary Medicine, SNOMED International Version 3.3. Illinois: College of American Pathologists. 234.
- [21] Crespi, M., Delvaux, M., Schapiro, M., Venables, C., Zwiebel, F. (1994). Minimal standards for a computerized endoscopic database. Ad hoc Task Force of the Committee for Minimal Standards of Digestive Endoscopy of the European Society of Gastrointestinal Endoscopy (ESGE). *Am J Gastroenterol*, -53.
- [22] Computer Zeitung. (1997). Component Broker liefert Infrastruktur und Entwicklungsumgebung zugleich. Nr. 22/30. Mai 1997 12-12.
- [23] Dayhoff, R.E., Maloney, D.L., Kuzmak, P.M., Shepard, B.M. (1991). Integrating medical images into hospital information systems. *J Digit Imaging*, 4, 87-93.
- [24] Detmer, W.M., Shortliffe, E.H. (1995). A model of clinical query management that supports integration of biomedical information over the World Wide Web. *Proc Annu Symp Comput Appl Med Care*, 898-902.
- [25] Diehl, M. (1995). Health system re-engineering: a CPRS economic decision model. *Proc Annu Symp Comput Appl Med Care*, 688-692.
- [26] Digital Equipment Corporation. (1994). Digital Distributed Computing Environment (DCE) for Windows NT.
- [27] Dore, L., Lavril, M., Jean, F.C., Degoulet, P. (1995). An object oriented computer-based patient record reference model. *Proc Annu Symp Comput Appl Med Care*, 377-381.
- [28] Edstrom, W., Berardinelli, R. (1993). Component Objects, Technology Overview. Redmond, WA, USA. Microsoft Corporation.
- [29] Essin, D.J., Lincoln, T.L. (1995). Implementing a low-cost computer-based patient record: a controlled vocabulary reduces data base design complexity. *Proc Annu Symp Comput Appl Med Care*, 431-435.
- [30] Esterhay, R.J., Jr. (1994). User metaphors for health care professional workstations. *Int J Biomed Comput*, 34, 95-113.
- [31] Feigl, P., Glaefke, G., Ford, L., Diehr, P., Chu, J. (1988). Studying patterns of cancer care: How useful is the medical record? *AM J PUBLIC HEALTH*, -533).
- [32] Flanagan, J.R., Olson, J.D., Wagner, J.R., Nuehring, N.E., Krumbholz, S.D. (1995). Cost-effective health information systems: user-driven internal development at the University of Iowa. *Proc Annu Symp Comput Appl Med Care*, 703-707.
- [33] Fleege, J.C., van Diest, P.J., Baak, J.P. (1992). QUALITY CONTROL METHODS FOR DATA ENTRY IN PATHOLOGY USING A COMPUTERIZED DATA MANAGEMENT SYSTEM BASED ON AN EXTENDED DATA DICTIONARY [SEE COMMENTS]. *Hum Pathol*, -7.
- [34] Gabrieli, E.R. (1988). Computer-assisted assessment of patient care in the hospital. *J Med Syst*, 12, 135-146.
- [35] van Ginneken, A.M., Liem, E.B., Moorman, P.W. (1993). Integrating QMR with a computer-based patient record. *Proc Annu Symp Comput Appl Med Care*, 98-102.
- [36] van Ginneken, A.M., van der Lei, J., Moorman, P.W. (1992). Towards unambiguous representation of patient data. *Proc Annu Symp Comput Appl Med Care*, 69-73.

- [37] Goldberg, A., Robson, D. (1983). SmallTalk-80 The Language and its Implementation, Addison-Wesley Publ. Co.
- [38] Goldberg, A. (1984). SmallTalk-80 The Interactive Programming Environment, Addison-Wesley.
- [39] Habertheuer, K.H., Kier, P., Ruckser, R., Scherz, M., Honiger, S., Sebesta, C., Tiefengraber, E., Schmid, A., Mandl, A., Sterz, M., et al. (1995). EDV-unterstützte Organisation der Hochdosistherapie mit Stammzelltransplantation im Donauspital. *Wien Med Wochenschr*, 145, 64-65.
- [40] Hammond, K.W. (1995). Treatment planning: implications for structure of the CPR. *Proc Annu Symp Comput Appl Med Care*, 362-366.
- [41] Harber, P., Miller, G., Smitherman, J. (1991). Work coding: beyond SIC and SOC, BOC and DOT [see comments]. *J Occup Med*, 33, 1274-1280.
- [42] Health Level Seven. (1994). Health Level Seven Version 2.2. Health Level Seven.
- [43] Heller, E.E. (1995). The computer-based patient record vision contrasted with HIS/MIS. *Int J Biomed Comput*, 39, 19-23.
- [44] Henry, S.B., Holzemer, W.L. (1994). Can SNOMED International represent patients' perceptions of health-related problems for the computer-based patient record? *Proc Annu Symp Comput Appl Med Care*, 184-187.
- [45] Hofmans Okkes, I.M., Lamberts, H. (1996). The International Classification of Primary Care (ICPC): new applications in research and computer-based patient records in family practice. *Fam Pract*, 13, 294-302.
- [46] Hohnloser, J.H., Fischer, M.R., König, A., Emmerich, B. (1994). Data quality in computerized patient records. Analysis of a haematology biopsy report database. *Int J Clin Monit Comput*, 11, 233-240.
- [47] Hufnagl, P., Nguyen-Dobinsky, T.N. (1997). Telemedizin - Der virtuelle Arzt im Cyberspace - oder viel Lärm um nichts? *Humboldt Spektrum*, 2/97.
- [48] Inoue, Y., Takahashi, T., Iwai, S. (1991). [Laboratory interpretive reporting system. A prototypical approach for knowledge representation in acid-base and electrolyte disorders]. *Rinsho Byori*, 39, 1035-1043.
- [49] Iversen, K.R., Heimly, V., Lundgren, T.I. (1995). Implementing security in computer based patient records clinical experiences. *Medinfo*, 8 Pt 1, 657-660.
- [50] Jacobsen, T.J. (1994). Workstations as enabling technologies for the computer-based patient record (CPR): point of care approaches across the patient care continuum-nursing's perspective. *Int J Biomed Comput*, 34, 123-129.
- [51] Jagannathan, V., Reddy, Y.V., Srinivas, K., Karinithi, R., Shank, R., Reddy, S., Almasi, G., Davis, T., Raman, R., Qiu, S., et al. (1995). An overview of the CERC ARTEMIS project. *Proc Annu Symp Comput Appl Med Care*, 12-16.
- [52] Johns, P. 1996. The ABCs of MFC ActiveX Controls. Microsoft Developer Network CD 97, 246.
- [53] Joubert, M., Miton, F., Fieschi, M., Robert, J.J. (1995). A conceptual graphs modeling of UMLS components. *Medinfo*, 8 Pt 1, 90-94.
- [54] Kanoui, H., Joubert, M., Favard, R., Maury, G., Pelletier, M. (1995). Acts and knowledge management in the NUCLEUS hospital information system. *Proc Annu Symp Comput Appl Med Care*, 338-342.
- [55] Keayes, R. (1995). White Paper on HL7-DICOM Interoperability: HIS/RIS-PACS Intercommunication in a Mixed DICOM and HL7 Environment. Richmond, BC, Canada. A.L.I. Technologies Inc.
- [56] Kimura, M., Ohyama, N., Inamura, K., Ando, Y., Shigemura, N., Shima, Y., Saito, T. (1995). IS&C (Image Save & Carry) standard: standardized exchange media for medical information. *Medinfo*, 8 Pt 1, 207-211.
- [57] Krasner G: SmallTalk-80 Bits of History, Words of Advice, Addison-Wesley Publ. Co., 1983

- [58] Lawrence, L.M. (1994). Safeguarding the confidentiality of automated medical information. *Jt Comm J Qual Improv*, 20, 639-646.
- [59] Liaw, S.T. (1994). General practice patient records. *Aust Fam Physician*, 23, 209-13.
- [60] Lunt, P.K. (1991). The perceived causal structure of loneliness. *J Pers Soc Psychol*, 61, 26-34.
- [61] McCray, A.T., Razi, A. (1995). The UMLS Knowledge Source server. *Medinfo*, 8 Pt 1, 144-147.
- [62] Microsoft Corporation. 1994. The Microsoft Object Technology Strategy. Microsoft Developer Network Library April 1996, 1-31.
- [63] Microsoft Corporation. Microsoft. 1997; MIDL Programmer's Guide and Reference. MSDN CD 97. Microsoft Corporation.
- [64] Microsoft Corporation. 1997. COM Interfaces and Interface Implementations. Microsoft Developer Network Library April 1996. Microsoft Corporation.
- [65] Microsoft Corporation. 1995. Microsoft Visual C++ Development System for Windows 95 and Windows NT Version 4. Microsoft Press. 6 Bände.
- [66] Moehr, J.R. (1994). Privacy and security requirements of distributed computer based patient records. *Int J Biomed Comput*, 35 Suppl, 57-64.
- [67] Moidu, K., Falsone, J.J., Nair, S. (1994). Computer-based patient record: the essential data set approach. *Proc Annu Symp Comput Appl Med Care*, 462-466.
- [68] Moser, W., Bohm, V., Bohmer, K., Engelbrecht, R., Brenner, H.H. (1995). Integrated development of a knowledge-based CPR system for quality assurance in diabetes outpatient clinics. *Medinfo*, 8 Pt 1, 236-239.
- [69] Murray, W. H., Pappas, C. (1995). *The Visual J++ Handbook*. Academic Press Inc. Boston San Diego New York London Sydney Tokyo Toronto.
- [70] Nguyen-Dobinsky, T.N., Hufnagl, P. (1997). Wie hoch ist der Turm von Babel in den Großkrankenhäusern? *Humboldt Spektrum*, 1/97.
- [71] Prokosch, H.U., Amiri, F., Krause, D., Neek, G., Dudeck, J. (1995). A semantic network model for the medical record of a rheumatology clinic. *Medinfo*, 8 Pt 1, 240-244.
- [72] Prokosch, H.U., Kamm, S., Wiczorek, D., Dudeck, J. (1991). Knowledge representation in pharmacology. A possible application area for the Arden Syntax? *Proc Annu Symp Comput Appl Med Care*, 243-247.
- [73] Prophet, C.M. (1993). The patient problem/nursing diagnosis form: a computer-generated chart document. *Proc Annu Symp Comput Appl Med Care*, 326-330.
- [74] Rindfleisch, T.C., Aronson, A.R. (1994). Ambiguity resolution while mapping free text to the UMLS Metathesaurus. *Proc Annu Symp Comput Appl Med Care*, 240-244.
- [75] Rothwell, D.J., Cote, R.A., Cordeau, J.P., Boisvert, M.A. (1993). Developing a standard data structure for medical language--the SNOMED proposal. *Proc Annu Symp Comput Appl Med Care*, 695-699.
- [76] Schramm, C., Goldberg, M., Pagurek, B. (1989). Multimedia radiological reports: creation and playback [published erratum appears in *J Digit Imaging* 1990 Feb 3(1):2]. *J Digit Imaging*, 2, 106-113.
- [77] Sherertz, D.D., Tuttle, M.S., Olson, N.E., Hsu, G.T., Carlson, R.W., Fagan, L.M., Acuff, R.D., Cole, W.G., Nelson, S.J. (1995). Accessing oncology information at the point of care: experience using speech, pen, and 3-D interfaces with a knowledge server. *Medinfo*, 8 Pt 1, 792-795.
- [78] Springer Verlag. (1992). *ICD-10*. Berlin, Heidelberg, New York: Springer Verlag. ICD-10.
- [79] Stroustrup, Bjarne: *The C++ Programming Language*, 2. Ausgabe, Addison-Wesley, Reading, MA, 1991.
- [80] Taylor, J.G. (1996). Breakthrough to awareness: a preliminary neural network model of conscious and unconscious perception in word processing. *Biol Cybern*, 75, 59-72.

- [81] Thurin, A., Carlsson, M., Gill, H., Wigertz, O. (1995). Arden syntax and GALEN terminology support: a powerful combination to represent medical knowledge. *Medinfo*, 8 Pt 1, 110.
- [82] Tuttle, M.S., Sherertz, D.D., Fagan, L.M., Carlson, R.W., Cole, W.G., Schipma, P.B., Nelson, S.J. (1993). Toward an interim standard for patient-centered knowledge-access. *Proc Annu Symp Comput Appl Med Care*, -8.
- [83] Vanderburg, G. (1996). *Tricks of the Java Programming Gurus*. Sams.net Publishing. Indianapolis, USA.
- [84] Walker, A. (1995). Communication standards: impact on nursing practice. *Medinfo*, 8 Pt 2, 1372.
- [85] Whates, P.D., Birzgalis, A.R., Irving, M. (1982). Accuracy of hospital activity analysis operation codes. *Br Med J Clin Res Ed*, 284, 1857-1858.
- [86] Williams, B.T., Imrey, H., Williams, R.G. (1991). The lifespan personal health record. *Med Decis Making*, 11, S74-6.
- [87] Williams, M.H., Hu, J. (1994). Making heterogeneous medical databases interoperable. *Comput Methods Programs Biomed*, 43, 275-281.
- [88] Williams, S., Kindel, C. 1994. The Component Object Model: A Technical Overview. Microsoft Developer Network Library April 1996.
- [89] Wingert, F. (1984). *SNOMED - Systematische Nomenklatur der Medizin*, 2 Bände. New York: Springer.
- [90] Wingert, F. (1985). *SNOMED Manual*. New York: Springer.
- [91] Wingert, F. (1985). Reduction of redundancy in a categorized nomenclature. In: Coté, R.A. et al. (1985), 191-202.
- [92] Zelingher, J., Rind, D.M., Caraballo, E., Tuttle, M.S., Olson, N.E., Safran, C. (1995). Categorization of free-text problem lists: an effective method of capturing clinical data. *Proc Annu Symp Comput Appl Med Care*, 416-420.

B. Internet-Seiten

- [93] Apple. (1997 June 10. 16:53). About OpenDoc. <http://www.opendoc.apple.com/dev/about.html>.
- [94] Brando, T.J. (1997 June 10. 16:50). Comparing DCE and CORBA. <http://www.mitre.org/research/domis/reports/DCEvCORBA.html>.
- [95] CERN. (1997 June 08. 12:21). A Little History. <http://www.cern.ch/CERN/WorldWideWeb/History/All.html>.
- [96] CERN. (1997 June 08. 12:25). VisualWorks, Object Oriented environment for development in SmallTalk. <http://www1.cern.ch/PTTOOL/VisualWorks/Tool.html>.
- [97] Cover, R. (1997 June 08. 12:32). Extensible Markup Language (XML). <http://www.sil.org/sgml/xml.html>.
- [98] Cygnus. (1997 June 08. 12:08). The ISO/ANSI C++ Draft. <http://www.cygnus.com/misc/wp/dec97pub/>
- [99] Davidson, A. (1997 June 10. 16:55). Programming with the HTTP Protocol, <http://www.webtechniques.com/features/jul96davis.html>.
- [100] Der Senat von Berlin. Der Senat von Berlin. (1996 November 01. 17:59). Berliner Datenschutzgesetz (BlnDSG), Gesetz zum Schutz personenbezogener Daten in der Berliner Verwaltung, in der Fassung vom 17.12.90, zuletzt geändert durch Gesetz vom 03.07.1995. <http://www.rewi.hu-berlin.de/Datenschutz/Gesetze/blndsg.html>.
- [101] Deutsche Telekom. (1996 November 20.). Deutsche Telekom AG at MEDICA 96, launch of DOXX.
- [102] DKFZ. (1996 October 20. 16:38). Medicus-2, Radiologie-Server für DOXX Klienten. <http://mbi.dkfz-heidelberg.de/mbi/medicus/doxx/pinciple.html>.
- [103] Drexler, K. E. (1997 June 08. 13:35). Thinking Machine. http://www.asiapac.com/EnginesOfCreation/EOC_Chapter_5.html.

- [104] Duke University. (1997 June 10. 16:55). Table of Contents (HL7 V2.3). <http://www.mcis.duke.edu/standards/HL7/pubs/version2.3/html/httoc.htm#J101>. Duke University.
- [105] Duke University. (1997 June 10. 16:55). HL7 Frequently Answered Questions Version 1.1, 12/10/95. <http://www.mcis.duke.edu/standards/HL7/faq/HL7FAQ10.HTM>. Duke University.
- [106] Enge, J. (1997 June 10. 16:55). DOXX - ISDN communications for practioners. <http://www.deteberkom.de/projekte/texte/Spirite.html>.
- [107] Flynn, J., Clarke, B. (1997 June 10. 16:37). Is Java Better than OLE? <http://www.datamation.com/PlugIn/issues/1996/march1/PG1.html>.
- [108] Heise Verlag. (1997 June 10. 16:37). Multimediagesetz regelt digitale Signaturen. <http://www.heise.de/ct/Artikel/96/11/krypto.htm>.
- [109] Horii, S.C., Prior, F.W., Bidgood, W.D.J., Parisot, C., Claeys, G. (1996 September 09. 21:32). DICOM: An Introduction to the Standard. http://www.xray.hmc.psu.edu/dicom/dicom_intro/DICOMIntro.html: Department of Radiology, Hospital of the University of Pennsylvania.
- [110] Köttl, F. (1997 June 10. 18:35). Wiens Donaospital als Vorreiter, Die digitale Radiologie setzte weitere IT-Integration in Gang. <http://www.computerwoche.de/archiv/1995/47/9547c074.html>
- [111] Lemay, L., Perkins, C. (1997 June 10. 16:35). Yes; Java's Secure. Here's Why. <http://www.datamation.com/PlugIn/issues/1996/march1/03ajava4.html>
- [112] Mauney, J. OSF. (1996 October 18. 18:06). DCE Frequently Asked Questions. <http://www.osf.org/dec/faq-mauney.html>.
- [113] McCarthy, V., Gosling, J. (1997 June 10. 16:40). Gosling On. <http://www.datamation.com/PlugIn/issues/1996/march1/03ajava2.html>. Datamation.
- [114] McDougall, M., Adams, J. Health Level Seven. (1996 July 30. 19:03). Health Level Seven. <http://www.va.gov/publ/standard/health/hl7.html>.
- [115] Pattison-Gordon, E. (1996 July 13. 22:53). Knowledge Representation for Medical Concepts. <http://www.dsg.harvard.edu/public/general/DSGKR.html>.
- [116] Rheingold, H. (1997 June 10. 16:47). Tools For Thought: The People and Ideas of the Next Computer Revolution. <http://www.well.com/user/hlr/texts/tft11.html>.
- [117] Schwarz, M. (1997 June 11. 09:05). Übersicht über CORBA, SOM und DSOM. http://wwwendres.informatik.tu-muenchen.de/da_fa_archiv/fopra/schwarma/corba3.html.
- [118] Sun Microsystems, Inc. (1997 June 08. 12:29). JDK 1.1.2 Documentation. <http://java.sun.com/products/jdk/1.1/docs/index.html> und weitere Links.
- [119] Sun Microsystems, Inc. (1997 June 08. 12:28). What is Java? <http://java.sun.com/nav/whatis/index.html> und weitere Links.
- [120] NLM. (1997 Jule 08. 14:52). UMLS Metathesaurus. <http://wwwkss.nlm.nih.gov/Docs/meta.fact.htm>: Office of Public Information, National Library of Medicine. MD, USA.
- [121] VA. (1997 June 10. 17:05). DHCP Infrastructure/Database Management Modules. <http://www.va.gov/monogrph/infmods.htm>: The Department of Veterans Affairs.
- [122] VA. (1997 June 10. 17:05). Current Procedural Terminology (CPT). <http://www.va.gov/monogrph/infrastr.uct/cpt>: The Department of Veterans Affairs.
- [123] VA. (1997 June 10. 17:05). Visit Tracking. <http://www.va.gov/monogrph/infrastr.uct/visitr.htm>: The Department of Veterans Affairs.
- [124] VA. (1997 June 10. 17:07). VA Generic Code Sheet. <http://www.va.gov/monogrph/infrastr.uct/gcodesht.htm>: The Department of Veterans Affairs.
- [125] VA. (1997 June 10. 17:07). VA FileMan. <http://www.va.gov/monogrph/infrastr.uct/fileman.htm>: The Department of Veterans Affairs.
- [126] VA. (1997 June 10. 17:07). VA ClassMan. <http://www.va.gov/monogrph/infrastr.uct/classman.htm>: The Department of Veterans Affairs.

- [127] VA. (1997 June 10. 17:08). Text Integration Utilities (TIU).
<http://www.va.gov/monogrph/infrastr.uct/tiu.htm>: The Department of Veterans Affairs.
- [128] VA. (1997 June 10. 17:10). Patient Data Exchange (PDX).
<http://www.va.gov/monogrph/infrastr.uct/edr.htm>: The Department of Veterans Affairs.
- [129] VA. (1997 June 10. 17:10). National Patch Module (NPM).
<http://www.va.gov/monogrph/infrastr.uct/npm.htm>: The Department of Veterans Affairs.
- [130] VA. (1997 June 10. 17:11). Messaging System.
<http://www.va.gov/monogrph/infrastr.uct/message.htm>: The Department of Veterans Affairs.
- [131] VA. (1997 June 10. 17:12). MailMan. <http://www.va.gov/monogrph/infrastr.uct/mailman.htm>:
The Department of Veterans Affairs.
- [132] VA. (1997 June 10. 17:12). List Manager. <http://www.va.gov/monogrph/infrastr.uct/list.htm>:
The Department of Veterans Affairs.
- [133] VA. (1997 June 10. 17:15). LetterMan. <http://www.va.gov/monogrph/infrastr.uct/ltrman.htm>:
The Department of Veterans Affairs.
- [134] VA. (1997 June 10. 17:15). Kernel ToolKit.
<http://www.va.gov/monogrph/infrastr.uct/toolkit.htm>: The Department of Veterans Affairs.
- [135] VA. (1997 June 10. 17:16). Event Driven Reporting (EDR).
<http://www.va.gov/monogrph/infrastr.uct/edr.htm>: The Department of Veterans Affairs.
- [136] VA. (1997 June 10. 17:17). Kernel. <http://www.va.gov/monogrph/infrastr.uct/kernel.htm>: The
Department of Veterans Affairs.
- [137] W3C. (1996 July 30.). Introducing HTML 3.2. <http://www.w3.org/pub/WWW/Markup/Wilbur/>

Lebenslauf

Allgemeine Angabe

Name Nguyen-Dobinsky, geb. Nguyen
Vorname Trong-Nghia
Geburtsdatum 13.03.1953
Geburtsort Thanh-Hoa, Vietnam
Wohnort Grunewaldstraße 55, 10825 Berlin
Familienstand verheiratet seit 1980, 1 Kind
Ehefrau Britta Dobinsky
Geburt unseres Sohnes Momme Vu Dobinsky 23.03.1978
Staatsang. deutsch seit 1980

Schulbildung

1959-1964 Grundschulen
1964-1966 Humanistisches naturwissenschaftliches Gymnasium Chu-Van-An, Saigon
1966-1968 Technisches Gymnasium Nguyen-Truong-To, Saigon, Grundstufe
1968-1971 Technisches Gymnasium Cao-Thang, Saigon, Mittel- und Oberstufen
1971 Baccalaureat Technique II (Technisches Abitur)
1972 Goethe-Institut, Berlin und Arolsen, Erlernen der deutschen Sprache

Studium

1972-1977 Technische Universität, Berlin, Studium der Schiffstechnik
1975-1977 Tutor, Fachbereich 11, Konstruktionslehre und Thermische Maschinen
1977 Diplom-Hauptprüfung

Berufstätigkeit

04.1977-09.1980 TU Berlin, wissenschaftlicher Assistent
10.1980-07.1983 IKO Software Service GmbH, Stuttgart, Gruppenleiter Technischer Informationssysteme
08.1983-03.1984 sietec Siemens Systemtechnik GmbH und Co KG, Berlin, Projektleiter
04.1984-09.1984 VW-Gesellschaft für technische Datenverarbeitungssysteme mbH, Berlin, Gruppenleiter Knowledge Engineering und DB-Systeme
10.1984-12.1987 sietec Siemens Systemtechnik GmbH und Co KG, Berlin, Hauptgruppenleiter Technische IS, Stellvertretender Abteilungsleiter
01.1988-08.1994 TRONY Gesellschaft für Beratung und Entwicklung von Software mbH Geschäftsführer
09.1994-04.1995 Institut für Pathologie, Charité, wissenschaftlicher Mitarbeiter
05.1995-jetzt Universitätsfrauenklinik, Charité, wissenschaftlicher Mitarbeiter

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, daß die eingereichte Dissertationsschrift mit dem Thema

*Konzept einer an semantischen Kriterien
orientierten Kommunikation für medizinische
Informationssysteme*

von mir selbst und ohne unzulässige Hilfe Dritter verfaßt wurde, auch in Teilen keine Kopie anderer Arbeiten darstellt und die benutzten Hilfsmittel sowie die Literatur vollständig angegeben sind.

Berlin, den _____

Dipl.-Ing. Trong-Nghia Nguyen-Dobinsky

Danksagung

An dieser Stelle möchte ich Herrn PD Dr. J. Michel, Institut für medizinische Informatik und Biometrie, für die Vergabe des Themas und die freundschaftliche Betreuung der Dissertation danken.

Herrn Prof. R. Bollmann, Universitätsfrauenklinik, danke ich für die Überlassung der technischen Hilfsmittel, insbesondere der EDV-Ressourcen, des Datenmaterials. Für den Zugriff auf die geburtshilfliche Datenbank danke ich Herrn Prof. H. Halle, ebenfalls Universitätsfrauenklinik. Meinen Dank aussprechen möchte ich auch den Mitgliedern der Pränatalen Forschungsgruppe PFAG der Universitätsfrauenklinik: Herrn PD Dr. R. Chaoui, Herrn Dr. K. Kalache, Herrn Dr. J. Hartung und Herrn Dr. K.-S. Heling für die wertvollen inhaltlichen Hinweise. Ebenfalls danke ich den Mitarbeitern und Mitarbeiterinnen der Abteilung Pränatale Diagnostik und Therapie der Universitätsfrauenklinik für die Unterstützung und das angenehme Forschungsklima. Ohne all diese Unterstützungen wäre meine Dissertation nicht zustande gekommen.

Herrn Dr. P. Hufnagl, Institut für Pathologie, möchte ich hier einen sehr herzlichen Dank für die zahlreichen inhaltlichen und sehr fruchtbaren Diskussionen und Gespräche aussprechen, die nicht mit der Abgabe meiner Dissertation beendet werden, sondern in weitere Forschungsarbeiten einfließen sollen. Frau Dr. Tennstedt, Institut für Pathologie, danke ich für die Nutzung des Fetal-Autopsy-Moduls und die inhaltlichen Unterstützungen.

Last but not least bedanke ich mich bei meiner Frau, Britta Dobinsky, für die unendliche Geduld und die „germanistischen“ Beratungen während meiner Dissertationszeit.