

# Erfahrungen mit dem Einsatz anwendungsspezifischer Piktogramme zur partizipativen Anforderungsanalyse

Wilhelm Hasselbring

Universität Tilburg, Infolab, PO Box 90153, NL-5000 LE Tilburg, Niederlande (e-mail: hasselbring@acm.org)

Eingegangen am 16. Januar 1998 / Angenommen am 6. Oktober 1998

**Zusammenfassung.** Die Erfahrungen aus einem Projekt im Bereich Krankenhausinformationssysteme werden in diesem Papier diskutiert. Insbesondere konzentrieren wir uns dabei auf den Einsatz von Techniken zur partizipativen Anforderungsanalyse und die dabei für die Anwender erreichte Nachvollziehbarkeit von den ermittelten Anforderungen zum entwickelten System. Ein zentraler Aspekt ist hierbei der durchgängige Einsatz von *anwendungsspezifischen* Piktogrammen in der Modellierung der Anforderungen und in der Implementierung der graphischen Benutzungsschnittstellen, wobei auch das frühzeitige Prototyping zur Überprüfung der ermittelten Anforderungen eine wichtige Rolle spielt.

**Schlüsselwörter:** Partizipative Anforderungsanalyse, Piktogramme, Nachvollziehbarkeit, Prototyping, Erfahrungsbericht

**Abstract.** The experience gained in a project in the domain of hospital information systems is discussed in the present paper. Particularly, we emphasize the use of techniques for participatory requirements analysis and the achieved user's traceability from the elicited requirements towards the realized system. A central concern is the general use of application-specific pictograms in requirements modeling and in the implementation of the graphical user interfaces, whereby early prototyping for evaluating the elicited requirements plays an important role.

**Key words:** Participatory requirements analysis, pictograms, traceability, prototyping, experience report

**CR Subject Classification:** D.2.1, D.2.2, H.1.2, H.5.2, J.3

## 1 Einleitung

### 1.1 Entwicklung von Software für den Krankenhausbereich

Bei der Entwicklung von Software für den Krankenhausbereich nimmt die Anforderungsanalyse eine zentrale Stellung ein, um die *tatsächlichen* Anforderungen der Anwender

zu erfüllen. Der Anwendungsbereich Krankenhaus zeichnet sich besonders durch ein hohes Maß an unterschiedlichen Aufgaben und teilweise gegensätzlichen Anforderungen unterschiedlicher Anwendergruppen, die sich grob den verschiedenen Berufsgruppen im Krankenhaus zuordnen lassen, aber auch durch deren Zusammenarbeit aus.

Ein Krankenhausinformationssystem ist als das gesamte informationsverarbeitende und informationsspeichernde Teilsystem eines Krankenhauses zu betrachten [33]. Der rechnerunterstützte Teil eines Krankenhausinformationssystems ist der Teil, der Rechner und Rechnernetze einsetzt. Bei der Entwicklung eines rechnerunterstützten Krankenhausinformationssystems ist eine Berücksichtigung aller Informationsflüsse wichtig. Eine isolierte Betrachtung des rechnerunterstützten Teils ist nicht ausreichend.

Die Erfahrungen aus einem interdisziplinären Kooperationsprojekt zwischen der Universität Dortmund (Fachbereich Informatik und Fachbereich Statistik) und dem Herzenzentrum im Klinikum Wuppertal mit dem Einsatz von Piktogrammen in der Modellierung bei der Neuentwicklung eines Informationssystems für die Kardiologie werden insbesondere in Bezug auf die Partizipation der Anwender diskutiert. Dieses Informationssystem dient der Unterstützung der medizinischen Forschung durch die statistische Auswertung von Diagnose- und Therapiedaten.

Als zentrale Technik für die Anforderungsanalyse wurden im vorgestellten Projekt Kooperationsdiagramme eingesetzt, die es erlauben, zusammen mit den Anwendern die Anforderungen an ein zu entwickelndes Informationssystem auf der Ebene von Informationsflüssen in der Klinik zu analysieren, wobei auch der nicht durch Rechner unterstützte Informationsaustausch berücksichtigt wird. Die resultierenden Kooperationsdiagramme bieten dann die Grundlage für die Strukturierung der Daten in der Datenbank, die Spezifikation des Verhaltens der Applikation und die Gestaltung der Benutzungsschnittstellen.

### 1.2 Einsatz von Piktogrammen

Ein Piktogramm ist eine stilisierte bildliche Darstellung zur Informationsvermittlung. Solche graphischen Darstellungen haben i.a. eine spezifische Bedeutung für verschiede-

dene Anwendungsbereiche (z.B. ein Halteverbotsschild für den Straßenverkehr). In graphischen Benutzungsschnittstellen für Software-Systeme werden häufig *Ikons* [4, 13] verwendet, die Piktogramme darstellen, um mit den Ikons eine bestimmte Metapher verbinden zu können. Softwaretechnisch können Piktogramme als Vektor- oder Rastergraphiken repräsentiert werden.

Piktogramme sind zur Unterstützung der Nachvollziehbarkeit für die Anwender besser geeignet als die ausschließliche Verwendung rein textueller Begriffe. Ein Piktogramm repräsentiert eine graphische Metapher, die sich mit reinen Texten (bzw. Akronymen) so nicht erreichen läßt [13]. Piktogramme sind aber nicht als Ersatz für textuelle Beschreibungen anzusehen, sondern als Ergänzung. Ein Modellierungswerkzeug sollte z.B. das Umschalten (bzw. die gemeinsame Darstellung) zwischen dem Namen und dem Symbol eines Piktogramms sowohl im Gesamtmodell als auch für ausgewählte Bereiche erlauben. Eine empirische Studie, über die in [16] berichtet wird, zeigte, daß durch die (Möglichkeit zur) Kombination von verschiedenen Modalitäten (hier Piktogramm und Text) eine optimale Informationsvermittlung erreicht wird. Diese Studie hat auch ergeben, daß es nicht ausreicht, sich auf vorgegebene (standardisierte) Symbole zu beschränken. Trotzdem ist es natürlich nicht sinnvoll, für *jedes* Modellierungselement unbedingt ein eigenes Piktogramm einzuführen.

Der Einsatz von benutzerdefinierten Piktogrammen in Datenflußdiagrammen zur Modellierung der Anforderungen durch verschiedene am Entwicklungsprozeß beteiligte Interessenvertreter wird z.B. in [26, 27] vorgeschlagen. Wir diskutieren den Einsatz von Piktogrammen während der Entwicklung einer Komponente eines Krankenhausinformationssystems, wobei insbesondere für die Anwender ein nachvollziehbarer Entwicklungsprozeß von den Anforderungen zum Anwendungssystem erreicht werden konnte.

### 1.3 Übersicht

In Abschnitt 2 diskutieren wir zunächst einige Aspekte der partizipativen Anforderungsanalyse. Abschnitt 3 stellt die Technik der Kooperationsdiagramme vor, bevor in Abschnitt 4 die Erfahrungen aus dem vorgestellten Projekt diskutiert werden. Abschnitt 5 liefert eine Zusammenfassung und einen Ausblick.

## 2 Techniken für die partizipative Anforderungsanalyse

Für die partizipative Ermittlung von Anforderungen an Krankenhausinformationssysteme hat sich die Kombination von Fragebögen, Interviews und Prototyping bewährt, was z.B. auch durch die Studie, über die in [32] berichtet wird, bestätigt wurde. In [6] wurden mehrere Studien zur partizipativen Software-Entwicklung ausgewertet. Eine wichtige Erkenntnis ist, daß das organisatorische Umfeld eine interdisziplinäre Zusammenarbeit erlauben muß und daß Prototyping eine geeignete Technik zur Einbindung der Anwender darstellt.

Insbesondere in Skandinavien hat der *partizipative Entwurf* [6] eine lange Tradition, wobei soziale Aspekte, wie

die Mitbestimmung bei der Gestaltung des Arbeitsplatzes, eine wichtige Rolle spielen. Der amerikanische Ansatz zur *gemeinschaftlichen Anwendungsentwicklung* [34] betont dagegen primär die partizipative Definition der funktionalen Anforderungen, was auch im vorgestellten Projekt ein vorrangiges Ziel ist. Beide Ansätze betonen die Einbeziehung der Anwender in die Gestaltung von Anwendungssystemen.

Für die Partizipation der Anwender ist es u.a. wichtig zu betrachten, wie die Anforderungen spezifiziert werden (Abschnitt 2.1), was Nachvollziehbarkeit in diesem Zusammenhang bedeutet (Abschnitt 2.2) und welche Rolle das Prototyping spielt (Abschnitt 2.3).

### 2.1 Spezifikationstechniken

In der traditionellen strukturierten Analyse [9] werden üblicherweise Datenflußdiagramme zur Beschreibung der (funktionalen) Anforderungen eingesetzt. In einigen älteren Ansätzen zur objektorientierten Modellierung, wie z.B. OMT [31], werden Datenflußdiagramme auch mit Klassendiagrammen (Beschreibung von Struktur) und Statecharts (Beschreibung von Dynamik) kombiniert. Häufig wird vorgeschlagen, Klassendiagramme auch zur initialen Anforderungsanalyse einzusetzen [2]. Wir sind allerdings nicht der Ansicht, daß sich Klassendiagramme als guter, *erster* Ausgangspunkt zur gemeinsamen Analyse der Anforderungen mit den Anwendern an ein neues Informationssystem eignen, bevor ein funktionales Modell konstruiert wird: auch wenn die zukünftigen Anwender ein Klassendiagramm (glauben zu) verstehen, ist damit die *Funktionalität* eines neuen Systems i.a. noch nicht ermittelt. Anwender erwarten üblicherweise eine gewisse Funktionalität von einem Informationssystem und nicht eine bestimmte Struktur, in der relevante Daten zu speichern sind oder ein System strukturiert wird.

Trotzdem ist es sehr sinnvoll, Datenmodelle (spezifiziert durch Klassendiagramme) für die Anforderungsanalyse mit den zukünftigen Anwendern zu diskutieren, um z.B. zu überprüfen, ob die Modellierer konzeptionelle Fehler gemacht haben. Anwender verstehen Datenmodelle, in denen auf hohem Niveau Begriffe aus ihrem Arbeitsbereich verwendet werden, wenn sie durch einen Modellierer erklärt werden. Zur Modellierung der Funktionalität sind jedoch andere Modelle notwendig.

Inzwischen ist es etabliert, *Anwendungsfallbeschreibungen* (use cases) zur Beschreibung der funktionalen Anforderungen in objektorientierten Modellierungsnotationen, wie z.B. in der UML [12], einzusetzen. Diese Technik hat ihren Ursprung im OOSE-Softwareentwicklungsprozeß von Jacobson [22]. OOSE ist ein Ansatz zur objektorientierten Entwicklung, der sehr gut durch das CASE-Werkzeug Objectory unterstützt wird [19]. Anwendungsfallbeschreibungen werden mit Akteuren in Beziehung gestellt. Die *Akteure* interagieren mit dem zu entwickelnden System. Sie sind nicht Teil des zu entwickelnden Systems. Die Akteure können Menschen in bestimmten Rollen oder technische Gerätschaften sein, die nicht genau beschrieben werden müssen. Die Anwendungsfallbeschreibungen beschreiben, *was* das zu entwickelnde System können soll. Anwendungsfälle sind sequentielle Folgen von Transaktionen, die zunächst informell beschrieben werden. Zusätzlich gehören

zum Anforderungsmodell *Schnittstellenbeschreibungen* (z.B. Prototypen für graphische Benutzungsschnittstellen).

Klassendiagramme eignen sich gut zur Problembereichsanalyse (domain analysis) [3], worauf wir in diesem Beitrag allerdings nicht genauer eingehen. Entsprechend dem OOSE-Softwareentwicklungsprozeß von Jacobson [22] sind Problembereichs- und Analysemodelle, zusätzlich zu den Anwendungsfallbeschreibungen, die zentralen Modelle, die im Analyseprozeß zur Anforderungsbeschreibung erstellt werden sollten. Ein *Problembereichsmodell* enthält Objekte mit Gegenstück im realen Problembereich bzw. Konzepte, mit denen ein zu entwickelndes System umgehen soll. Das Problembereichsmodell dient auch als Grundlage für das *Analysemodell*, das eine erste Strukturierung des zu entwickelnden Systems auf hoher Abstraktionsebene liefert [22]. Implementierungsaspekte sollen im Analysemodell noch vernachlässigt werden; das Analysemodell ist aber als ein erster, grober Entwurf anzusehen. Das *Entwurfsmodell* dient in OOSE zur Verfeinerung und Anpassung der Anforderungs- und Analysemodelle an die Systemumgebung (Programmiersprache, Datenbank, Netzwerk etc.).

Hier sei angemerkt, daß Kommunikation, Nebenläufigkeit und Konkurrenz zwischen Anwendungsfällen nicht graphisch modelliert werden können. Anwendungsfälle kommunizieren mit Akteuren, nicht mit anderen Anwendungsfällen. Beziehungen zwischen Anwendungsfällen dienen nur zur inkrementellen Spezifikation der Anwendungsfälle (Benutzt- und Erweiterungsbeziehungen). Mit der im vorgestellten Projekt eingesetzten Technik der Kooperationsdiagramme wird insbesondere Kommunikation modelliert, wobei die Notation an die traditionellen Datenflußdiagramme erinnert (siehe Abschnitt 3).

## 2.2 Nachvollziehbarkeit

Nachvollziehbarkeit (traceability) in der Software-Entwicklung kann als die Möglichkeit zur Beschreibung und zum Verfolgen des Lebenszyklus eines Artefakts sowohl in Vorwärts- als auch in Rückwärtsrichtung definiert werden [7, 14, 30]. Ein (Modellierungs-) Element aus einer Anforderungsbeschreibung sollte dann z.B. von seinem 'Ursprung' (wo kommt diese Anforderung her) bis hin zur Implementierung (wie wurde diese Anforderung erfüllt) verfolgt werden können; und umgekehrt. Nachvollziehbarkeit in beide Richtungen ist notwendig. Hypertext-Techniken stellen eine Möglichkeit dar, Nachvollziehbarkeit in Modellierungswerkzeugen zu unterstützen. In diesem Unterabschnitt soll die Nachvollziehbarkeit in der Software-Entwicklung zur Einordnung und Abgrenzung der vorgestellten Arbeit kurz diskutiert werden.

Nachvollziehbarkeit soll insbesondere unterstützen, Änderungen in einzelnen Modellen — z.B. in den Anforderungen, im Entwurf, im Programm etc. — in anderen Modellen nachvollziehen zu können. Das ist auch dann besonders wichtig, wenn große Softwaresysteme im Team entwickelt werden [29]. Zwischen den verschiedenen Sichten bzw. Sichtweisen [25] der unterschiedlichen Beteiligten, die bestimmte Rollen im Entwicklungsprozeß einnehmen (insbesondere auch die Anwender) gibt es i.a. Abhängigkeiten, die möglichst konsistent gehalten werden sollten, auch wenn

temporäre Inkonsistenzen zugelassen werden müssen [11]. Die in einem bestimmten Kontext zu unterstützende Nachvollziehbarkeit hängt auch von den jeweiligen Beteiligten bzw. Interessenvertretern und der aktuellen Phase im Entwicklungsprozeß ab.

Der Ursprung einer Anforderung (requirements pre-traceability) [14, 28] bezieht sich z.B. auf solche Aspekte, die zur Beschreibung dieser Anforderung geführt haben. Diese *Prä-Nachvollziehbarkeit* ist wichtig und hilfreich, um Änderungen der Anforderungen, die aus dem Anwendungsbereich kommen, vernünftig handhaben zu können. Ein wichtiges Ziel ist, im nachhinein Anforderungen verstehen und die Entstehung von Anforderungen nachvollziehen zu können. Die Prä-Nachvollziehbarkeit kann z.B. die Verantwortlichkeiten für bestimmte Aspekte einer Anforderung liefern [15]. Die Grundlagen und Ursachen, die zur Ermittlung der Anforderungen geführt haben, sollten damit erkennbar sein.

Andererseits ist die *Post-Nachvollziehbarkeit* von den Anforderungen hin zur implementierten Anwendung eine wichtige Voraussetzung für eine gute Akzeptanz von Software-Systemen durch die Anwender. Dieser Beitrag konzentriert sich auf Aspekte der Partizipation der Anwender und damit auf die Post-Nachvollziehbarkeit. Diese Nachvollziehbarkeit von der Anforderungsbeschreibung zum entwickelten System soll die Partizipation unterstützen. Aspekte der Nachvollziehbarkeit für die Entwickler, insbesondere im Kontext von Softwareentwicklungsumgebungen, spielen dabei eine untergeordnete Rolle, so daß wir diese Aspekte hier nicht genauer diskutieren werden.

## 2.3 Prototyping

Eine frühzeitige Überprüfung der Ergebnisse der Anforderungsanalyse durch Ausführung und Bewertung von Prototypen ist sinnvoll, um eine inkrementelle und evolutionäre Entwicklung zu ermöglichen [5]. Für das Prototyping müssen natürlich bestimmte technische und organisatorische Voraussetzungen erfüllt sein. Wichtig ist, daß beim Prototyping nicht nur Oberflächenmasken (sogenannte Mock-Up-Prototypen) zur Verfügung stehen, sondern daß auch ein gewisses Maß an Funktionalität vorhanden ist, damit die Arbeitsprozesse am Prototypen nachvollzogen werden können.

Um den Anwendern frühzeitig die Möglichkeit zur Überprüfung der ermittelten Anforderungen zu geben, sind ausführbare Prototypen, mit denen die Anwender wichtige Arbeitsschritte durchspielen können, sehr hilfreich. Bekannterweise haben Anwender große Schwierigkeiten damit, ihre Wünsche den Entwicklern gegenüber verständlich zu machen. Häufig wissen sie auch nicht, was sie eigentlich wollen bzw. benötigen. Das kann dann oft erst durch die Arbeit mit einem (prototypischen) Anwendungssystem ermittelt werden. Zur systematischen Bewertung der Prototypen ist es sinnvoll, strukturierte Fragebögen zu entwerfen, die gemeinsam von den Anwendern und den Entwicklern ausgewertet werden.

Ein weiterer Aspekt für die Überprüfung der Ergebnisse der Anforderungsanalyse ist eine frühzeitige Simulation von Modellen. Für Kooperationsdiagramme ist das jedoch nur eingeschränkt möglich, da hier noch kein Kontrollfluß explizit modelliert wird. Zustandsautomaten (bzw. Statecharts)


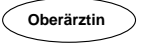



Konzept	Notation
Organisationsbereich	
Funktionelle Rolle	
Informationsfluß	
Art der Informationsweitergabe	
Medien als Informationsquelle und -senke	






Abb. 1. Vorgegebene, *problembereichsspezifische* Elemente in den eingesetzten Kooperationsdiagrammen für Krankenhausinformationssysteme

können durch Simulation ausgeführt werden [18] (das gilt z.B. auch für Petri-Netze); dieses konnten wir im vorgestellten Projekt allerdings nicht einsetzen, weil uns auf der Zielplattform keine entsprechenden Werkzeuge zur Verfügung standen.

### 3 Kooperationsdiagramme




Kooperationsdiagramme modellieren den Informationsfluß zur Kooperation von Organisationsbereichen und Personen in funktionellen Rollen. Die Technik der Kooperationsdiagramme wurde z.B. in [24] für die Definition von Kernsystem und Ausbaustufen zur Auswahl eines integrierten Krankenhausinformationssystems eingesetzt (dort *Kooperationsbilder* genannt).

Im Rahmen des hier vorgestellten Projekts sind die Kooperationsdiagramme in modifizierter und erweiterter Form für die *Neuentwicklung* eines Informationssystems zum Einsatz gekommen. Ein wichtiges Ziel für die Erweiterungen war, einen auch für die Anwender nachvollziehbaren schrittweisen Übergang von den Ergebnissen der Analyse hin zum entwickelten System zu erreichen. Eine wesentliche Erweiterung zum Ansatz aus [23, 24] ist die Ergänzung um *anwendungsspezifische* Piktogramme, die auch in den graphischen Benutzungsschnittstellen des zu entwickelnden Systems verwendet werden.

Unsere vordefinierten, *problembereichsspezifischen* Elemente in Kooperationsdiagrammen für Krankenhausinformationssysteme sind in Abb. 1 dargestellt. In Kooperationsdiagrammen wird der Informationsaustausch zwischen Organisationsbereichen und funktionellen Rollen, die Personen einnehmen können, dargestellt. Die Art der Informationsweitergabe wird an den gerichteten Kanten, die den Informationsfluß darstellen, durch vorgegebene Piktogramme für Telefon () , direktes Gespräch () , rechnergestützte Kommunikation () , Hauspost () oder Briefpost () dargestellt. So können alle relevanten Informationsflüsse im Krankenhaus explizit in der Analyse berücksichtigt werden, um eine isolierte Betrachtung der rechnergestützten Kommunikation zu vermeiden.

In [23, 24] werden nur *problembereichsspezifische* Piktogramme eingesetzt. Der allgemeine *Problembereich* ist

hier das Krankenhaus. Im vorgestellten Projekt ist der *Anwendungsbereich* ein konkretes kardiologisches Informationssystem. Die problembereichsspezifische Notation wurde schrittweise zu einer anwendungsspezifischen Notation erweitert. Dazu wurden die folgenden Änderungen bzw. Erweiterungen gegenüber dem Ansatz aus [23, 24] durchgeführt:

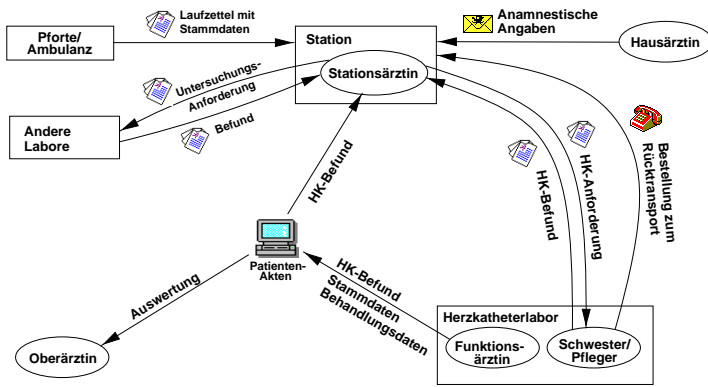
- Die vorgegebenen, problembereichsspezifischen Piktogramme zur Kennzeichnung der unterschiedlichen Kommunikationsarten werden schrittweise um spezielle, *anwendungsspezifische* Piktogramme aus dem jeweiligen Anwendungsbereich ergänzt, die für die Anwender einen Rückschluß auf die Bedeutung bzw. auf die Art der übermittelten Information zulassen (z.B.  $\Omega$  als Symbol für Herzkatheterbefunde im vorgestellten Projekt).
- Der Rechner kann in den Kooperationsdiagrammen nicht nur als Medium zur Informationsweitergabe eingesetzt werden () , sondern auch als Informationsquelle und -senke () . Das gleiche gilt für traditionell archivierte Informationen () . So kann auch Information, die nicht direkt zwischen Personen oder Organisationseinheiten ausgetauscht wird, explizit berücksichtigt werden. Das gilt insbesondere für Informationen, die inkrementell erhoben und nur bei Bedarf benötigt werden.

Um bei komplexeren Modellen die Übersicht zu wahren, ist es möglich, die Informationsquellen und -senken logisch auf mehrere Modellierungs-Komponenten aufzuteilen, die dann jeweils für eine Datenmenge zuständig sind. Technisch können mehrere solcher logischen Komponenten in einem System realisiert werden. Umgekehrt kann eine logische Komponente natürlich auch durch mehrere physische Systeme realisiert werden.

- Rollen, die direkt Organisationsbereichen zugeordnet werden können, werden *in* die entsprechenden Rechtecke gezeichnet. In [23, 24] werden solche Rollen *neben* den zugehörigen Organisationsbereichen angeordnet.

Es hat sich bewährt, die Kooperationsdiagramme zunächst an einer großflächigen Tafel (bzw. Flip-Chart) gemeinsam mit den Anwendern zu konstruieren und dann später mit einem Text- und Graphikprogramm zu dokumentieren. Die Entwickler übernehmen bei der Modellierung anfangs nur die Moderation. Die durch die Entwickler dokumentierten Ergebnisse werden später durch die Anwender begutachtet.

Für eine Unterstützung des Arbeitsflusses durch ein Workflow-Management-System wäre eine Modellierung der *Reihenfolge*, in der die einzelnen Arbeitsschritte ausgeführt werden (sollen), sinnvoll. Das wird durch die hier vorgestellten Kooperationsdiagramme nicht explizit unterstützt. Für diesen Zweck wären z.B. FUNSOFT-Netze [8], die eine abstrakte Petri-Netz-Variante darstellen, besser geeignet als Kooperationsdiagramme. Im vorgestellten Projekt waren die zeitlichen Abläufe für die Behandlungen selbst jedoch fest vorgegeben, so daß auf deren explizite Modellierung verzichtet wurde. Die Daten werden immer dann eingegeben, wenn sie anfallen. Daher reichte es im vorgestellten Kontext aus, für den Informationsfluß die Art der Weitergabe und die übertragene Information zu spezifizieren.



**Abb. 2.** Ein Ausschnitt aus dem Kooperationsdiagramm zur Ist-Analyse. Rechtecke stellen Organisationsbereiche dar, Ovale stehen für funktionelle Rollen und gerichtete Kanten stehen für die Informationsweitergabe. HK steht für Herzkatheter

#### 4 Partizipative Anforderungsanalyse im Krankenhaus-Projekt

Die Grundlage und Motivation für das im folgenden vorgestellte Projekt war ein bestehendes Informationssystem zur Erfassung und Auswertung von Herzkatheter-Untersuchungen. Die Hauptkritikpunkte am alten System waren:


- Die graphische Nutzungsoberfläche war sehr schlecht strukturiert.
- Das Datenmodell war für wissenschaftliche Auswertungen kaum geeignet.

Um die Anforderungen der Anwender besser zu erfüllen, wurde ein besonderer Schwerpunkt auf einen partizipativen Entwicklungsprozeß gelegt, wobei die Mitwirkung der Anwender eine zentrale Rolle spielt. Der Einsatz von Piktogrammen in der Analyse (Abschnitt 4.1) und in der prototypischen Implementierung der graphischen Benutzungsschnittstellen (Abschnitt 4.2) wird im Kontext des vorgestellten Projekts diskutiert.

##### 4.1 Analyse des Ist-Zustands und der Anforderungen

Um eine genaue Einschätzung der Eigenschaften des in der Klinik existierenden, alten Systems zu gewinnen, wurde zunächst gemeinsam mit den Anwendern der Ist-Zustand durch Kooperationsdiagramme untersucht (Abschnitt 4.1.1), bevor die Anforderungen an das neu zu entwickelnde System mit Hilfe von weiteren Kooperationsdiagrammen ermittelt wurden (Abschnitt 4.1.2). In Abschnitt 3 wurde die eingesetzte Technik der Kooperationsdiagramme eingeführt.

##### 4.1.1 Analyse des Ist-Zustands mit dem alten System

Ein Ausschnitt des Kooperationsdiagramms als Ergebnis der Ist-Analyse ist in Abb. 2 dargestellt. Die Annotationen an den Kanten geben die Art der Informationsweitergabe durch Piktogramme und den übertragenen Inhalt textuell an. Die Interaktion mit dem Rechner als Informationsquelle und -senke erfolgt implizit rechnerunterstützt, so daß hier die Art der Informationsweitergabe nicht angegeben werden muß (die direkte rechnerunterstützte Informationsweitergabe wird durch das Piktogramm  modelliert). Wie in Abb. 2 zu erkennen ist, fand bisher keine direkte Kommunikation über

den Rechner statt; er diente nur dazu, Behandlungsdaten im Herzkatheterlabor einzugeben und bei Bedarf auf der Station einzusehen oder statistisch auszuwerten.

Zur Erläuterung des Modells: Patienten kommen i.a. mit einem Laufzettel von der Pforte oder aus der Ambulanz auf die Station und werden evtl. im Herzkatheterlabor operiert. Die Herzkatheteranforderungen und -befunde werden durch die Hauspost zwischen Station und Herzkatheterlabor transportiert. Untersuchungsanforderungen und -befunde werden mit den anderen Laboren des Klinikums ebenfalls per Hauspost ausgetauscht. Die anamnestischen Angaben (Krankenvorgeschichte) werden per Briefpost in das Herzzentrum transportiert und dort als Teil der Behandlungsdaten in das existierende Informationssystem eingegeben.

Mit dem existierenden System mußten alle Behandlungsdaten durch den behandelnden Funktionsarzt im Herzkatheterlabor eingegeben werden. Die Daten wurden durch den Oberarzt für Auswertungen und gelegentlich durch den Stationsarzt für Herzkatheterbefunde verwendet. Die Qualität der eingegebenen Daten war nicht sehr hoch, weil die Eingabe mit den schlecht strukturierten Masken für den behandelnden Arzt eine erhebliche Mehrbelastung darstellte. Auch waren die in der Datenbank verwendeten Datenstrukturen kaum für aussagekräftige Auswertungen geeignet, so daß eine Neuentwicklung notwendig wurde.

Der Transport von Patienten aus der Station in die Labore wurde hier nicht explizit modelliert, weil hierbei keine 'Information' ausgetauscht wird. Die Anwesenheit eines Patienten ist für die Durchführung von Behandlungen natürlich notwendig. Um die konkreten zeitlichen Abläufe in der Klinik angemessen zu modellieren, müßten diese Aspekte auch berücksichtigt werden. Da im vorgestellten Projekt die zeitlichen Abläufe für die Behandlungen selbst jedoch fest vorgegeben waren und es nicht das Ziel war, ein Workflow-Management-System zu realisieren, wurde auf eine explizite Modellierung der zeitlichen Abläufe verzichtet. Die Daten werden immer dann eingegeben, wenn sie anfallen.

##### 4.1.2 Analyse der Anforderungen an das neu zu entwickelnde System

Auf der Basis der Analyse des Ist-Zustands wurden dann die Anforderungen an das neu zu entwickelnde System gemeinsam mit den Anwendern mit Hilfe von weiteren Kooperationsdiagrammen untersucht. Eine zentrale Frage

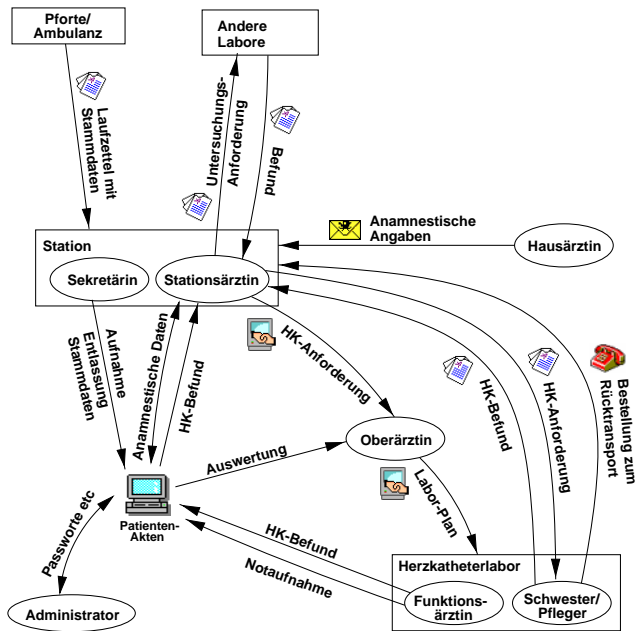


Abb. 3. Ein Ausschnitt des ersten Kooperationsdiagramms zur Anforderungsanalyse, in dem nur die vordefinierten Elemente in Kooperationsdiagrammen für Krankenhausinformationssysteme verwendet werden (siehe Abb. 1)

der Anforderungsanalyse war: *Welche Daten werden wo durch wen erhoben und von wem benötigt?* Ein Ausschnitt des Ergebnisses ist in Abb. 3 dargestellt, wobei zunächst nur die vordefinierten, problembereichsspezifischen Elemente für Krankenhausinformationssysteme verwendet werden (siehe Abb. 1).

Ein wesentlicher Unterschied zum alten System ist die Tatsache, daß die Eingabe der Daten jetzt durch verschiedene Personen bzw. Rollen durchgeführt wird. Die Daten werden im neuen System eingegeben, wenn sie anfallen. Z.B. gibt die Sekretärin die Stammdaten ein, sobald sich die Patienten in der Abteilung melden. Die Sekretärin spielte im Kontext des alten Systems noch keine Rolle. Als weitere neue Rolle ist die des Administrators zur Vergabe von Zugriffsrechten hinzu gekommen. Die anamnестischen Daten werden nicht mehr im Herzkatheterlabor, sondern auf der Station eingegeben. Nur bei Notoperationen müssen die Stammdaten eines neuen Patienten im Herzkatheterlabor eingegeben werden.

Die Arbeitsorganisation wird durch das neue System unterstützt, indem der Stationsarzt Herzkatheteranforderungen für den Oberarzt zusammenstellt. Der Oberarzt stellt dann aus den Anforderungen den Labor-Plan zusammen. Aus organisatorischen Gründen werden die Herzkatheteranforderungen und -befunde wie zuvor auch auf Papier zwischen Station und Herzkatheterlabor ausgetauscht.

Insbesondere die Eingabe, Speicherung und Bearbeitung der medizinischen Daten durch die Ärzte wurde in Zusammenarbeit mit Statistikern der Universität Dortmund optimiert, um die Qualität der geplanten Auswertungen zu erhöhen. Die Strukturierung der Eingabemasken in der späteren Implementierung orientiert sich auch an der Zuordnung der Information zu den einzelnen Piktogrammen (siehe auch Abschnitt 4.2).

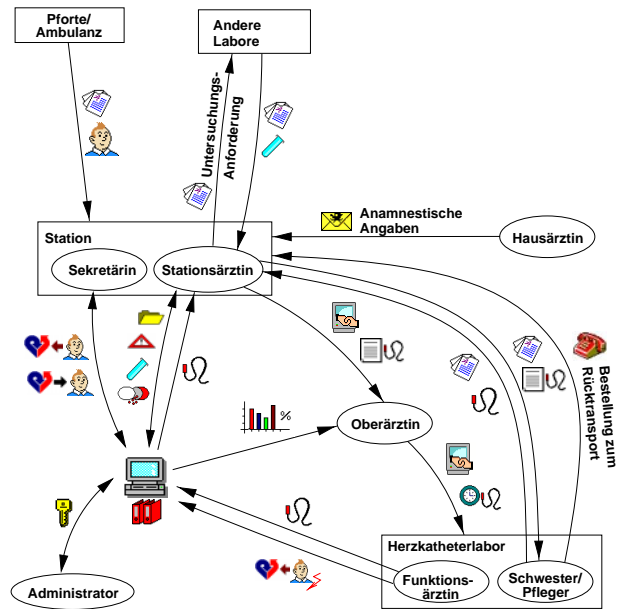


Abb. 4. Das modifizierte Kooperationsdiagramm zur Anforderungsanalyse. An den Kanten sind die übermittelten, einzugebenden bzw. gelesenen Informationen anschaulich durch anwendungsspezifische Piktogramme dargestellt: steht für die Patientendaten, steht für die allgemeinen anamnестischen Daten, steht für das Risikoprofil, steht für die Laborwerte, steht für die Medikation, ist das Logo des Herzzentrums im Klinikum Wuppertal, etc. (siehe Abb. 7 für eine vollständige Liste der anwendungsspezifischen Piktogramme). Diese Symbole stammen aus dem Anwendungsbereich und haben für die Anwender eine klare Bedeutung. Als naheliegende Erweiterung bietet es sich an, z.B. auch für die Rollen Piktogramme einzuführen, was im vorgestellten Projekt jedoch noch nicht gemacht wurde

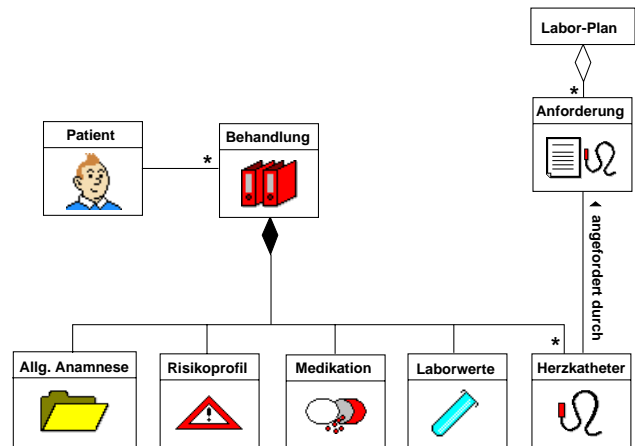
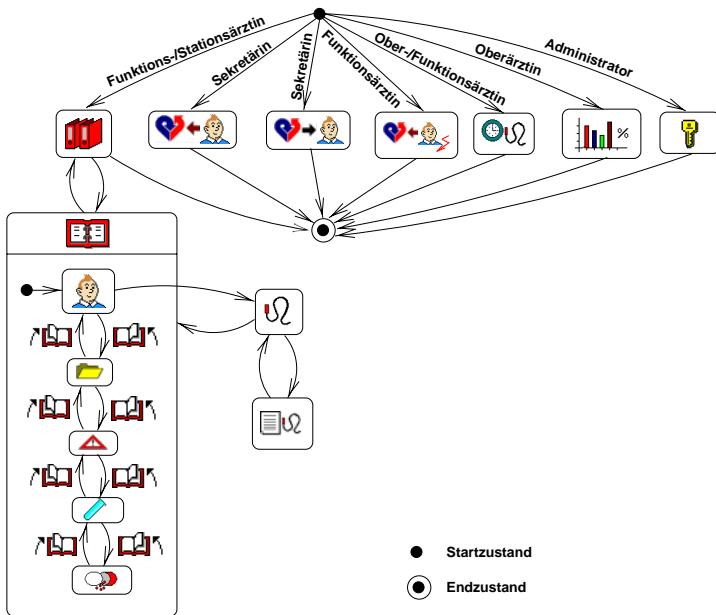


Abb. 5. Ein Ausschnitt des Analysemodells für die Patientenakten

In Abb. 4 ist ein modifiziertes Kooperationsdiagramm zur Anforderungsanalyse dargestellt, in dem an den Kanten die übermittelten, einzugebenden bzw. gelesenen Informationen anschaulich durch weitere Piktogramme dargestellt sind. Bei den neuen Piktogrammen handelt es sich um anwendungsspezifische Erweiterungen für das Herzzentrum des Klinikums Wuppertal, die von den Anwendern selbst mit ausgewählt wurden. Selbstverständlich müssen nicht für alle Modellierungselemente Piktogramme eingeführt werden.



**Abb. 6.** Die Dynamik der Benutzungsschnittstelle modelliert als Statechart. Teilweise werden Piktogramme zur Beschreibung der Zustände und der Zustandsübergänge eingesetzt. Ausgehend vom Startzustand können sich die Anwender in einer bestimmten Rolle anmelden und erhalten dann eine zu dieser funktionellen Rolle gehörende Benutzungsschnittstelle. Die Piktogramme, die diese Benutzungsschnittstellen im Statechart identifizieren, stammen aus den Kooperationsdiagrammen und dem Datenmodell. Die Navigation durch die Patientenakte erfolgt in mehreren verknüpften Masken, wie im geschachtelten Zustand für die geöffnete Patientenakte dargestellt. Für die Navigation wurden neue Piktogramme zum Vor- und Zurückblättern eingeführt. Wegen der besonderen Bedeutung der Herzkatheter-Befunde und -Anforderungen wurden die entsprechenden Masken aus dem geschachtelten Zustand herausgenommen

In Abb. 5 ist ein Ausschnitt des Analysemodells (im Sinne des OOSE-Softwareentwicklungsprozesses von Jacobson; siehe Abschnitt 2.1) für die Patientenakte dargestellt. Einem Patienten können mehrere Behandlungen zugeordnet werden, wobei jede Behandlungsakte mehrere (angeforderte) Herzkatheter-Untersuchungen enthalten kann. Die anamnestischen Daten werden je Behandlung genau einmal erfasst. Die Behandlungsdaten setzen sich hier aus einem allgemeinen anamnestischen Teil, dem spezifischen Risikoprofil, der Medikation, den Werten aus anderen Laboren und den bisherigen Herzkatheterbefunden zusammen. Der Labor-Plan ordnet die Anforderungen.

Zur Modellierung dieses Analysemodells wird die UML-Notation für Klassendiagramme verwendet [12]. Der \* an einer Assoziation spezifiziert die Kardinalität  $\geq 0$  (die Grundeinstellung ist die Kardinalität 1). Der kleine Pfeil an Assoziationsnamen deutet die Leserichtung an. Die gefüllte Raute modelliert *ungeteilte* Aggregation (Komposition): die aggregierten Teile gehören zu genau einem Aggregat, wobei die Existenz der Teile an die Existenz des Aggregats gebunden ist [12]. Teile mit der Kardinalität 1 dürfen nicht entfernt werden. Die UML erlaubt auch die Spezifikation der *geteilten* Aggregation, wobei Teile zu mehreren Aggregaten gehören dürfen und auch ohne enthaltene Aggregate existieren können [12]. Herzkatheter-Anforderungen können auch existieren, ohne einem Labor-Plan zugeordnet zu sein.

Auch im Analysemodell werden einige der in den Kooperationsdiagrammen verwendeten Piktogramme benutzt. So erhalten wir eine nachvollziehbare *Zuordnung* von der funktionsorientierten Darstellung der Kooperationsdiagramme hin zur objektorientierten Modellierung. Das Analysemodell wurde mit den zukünftigen Anwendern diskutiert, um zu überprüfen, ob bei der Modellierung konzeptionelle Fehler gemacht wurden. Dieses Modell beschreibt die zu speichernden Daten auf relativ hohem Abstraktionsniveau.

In Abb. 6 ist die Dynamik der Benutzungsschnittstelle als Statechart [17] modelliert, wobei auch hier teilweise Piktogramme zur Beschreibung der Zustände und der Zu-

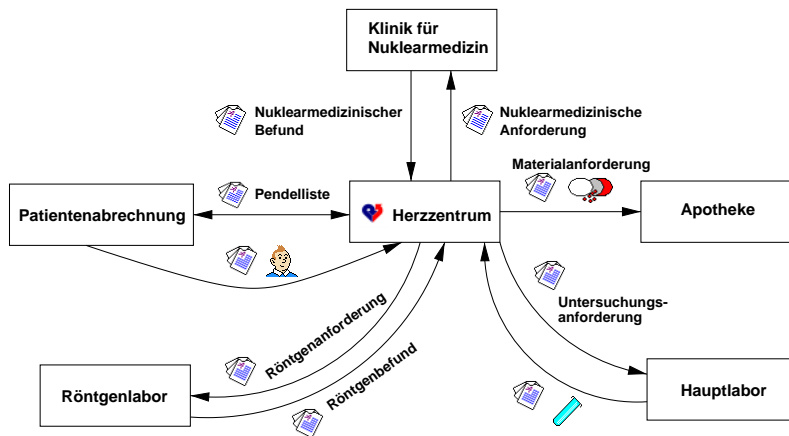
Piktogramm	Kommentar	Piktogramm	Kommentar
	Logo des Herzzentrums in Wuppertal		Notfallaufnahme in das Herzzentrum
	Entlassung aus dem Herzzentrum		Medikation
	Aufnahme in das Herzzentrum		Risikoprofil
	Herzkatheterlaborplan		Herzkatheterbefund
	Herzkatheteranforderung		Auswertungstatistik
	Laborwerte		Stammdaten
	Allgemeine Anamnese		Patientenakten
	Vorblättern in den Akten		Geöffnete Akte
	Zurückblättern in den Akten		

**Abb. 7.** Die anwendungsspezifischen Piktogramme für das Herzzentrum Wuppertal

standsübergänge eingesetzt werden. Die Zustandsübergänge entsprechen dem Öffnen/Schließen von Eingabemasken und die Zustände entsprechen dem Bearbeiten der entsprechenden Masken. Die Navigation durch die Patientenakte wurde hierarchisch durch einen geschachtelten Zustand modelliert.

Abbildung 7 bietet einen Überblick mit kurzen Erläuterungen zu den anwendungsspezifischen Piktogrammen.

Im vorgestellten Projekt wurde in der ersten Phase nur der Informationsfluß innerhalb einer Abteilung (dem Herzzentrum) genauer untersucht. Für eine angemessene Unterstützung des Informationsflusses in der gesamten Klinik



**Abb. 8.** Ein kleiner Ausschnitt des Kooperationsdiagramms zur abteilungsübergreifenden Anforderungsanalyse. Hier werden noch keine zusätzlichen anwendungsspezifischen Piktogramme verwendet

ist es notwendig, auch den Informationsaustausch zwischen verschiedenen Abteilungen zu betrachten, was zur Zeit in einer weiteren Projektphase geschieht. Abbildung 8 stellt einen kleinen Ausschnitt des Kooperationsdiagramms zur übergreifenden Anforderungsanalyse zwischen den verschiedenen Abteilungen des gesamten Klinikums dar. Hier wird der Ist-Zustand modelliert, wobei noch keine zusätzlichen anwendungsspezifischen Piktogramme eingeführt wurden. Als neue Anforderung hat sich z.B. unmittelbar ergeben, daß die Laborwerte (✍) aus dem Hauptlabor direkt elektronisch übertragen werden sollten, statt sie in der kardiologischen Abteilung von den erhaltenen Papierunterlagen nochmals zu erfassen. Das gleiche gilt für die Stammdaten, die bereits in die Rechner der Verwaltung (Patientenabrechnung) eingegeben wurden (sofern kein Notfall vorliegt), sowie für die Daten aus den anderen Abteilungen. Für eine genauere Diskussion der sich aus diesen Anforderungen ergebenden Probleme und Lösungsansätze für die Integration der verteilten, heterogenen Informationssysteme der einzelnen Abteilungen sei auf [20] verwiesen. Dieses Papier konzentriert sich auf die Aspekte der partizipativen Anforderungsanalyse.

#### 4.2 Prototypische Implementierung und Evaluation

Im vorgestellten Projekt wurden den Anwendern frühzeitig Prototypen zur Unterstützung der einzelnen Arbeitsschritte zur Verfügung gestellt, um die Ergebnisse der Anforderungsanalyse zu überprüfen und zu verfeinern. Zur systematischen Bewertung der Prototypen wurde für jede Maske ein strukturierter Fragebogen entworfen, der zusammen mit den Anwendern ausgewertet wurde. Für das Prototyping müssen bestimmte technische und organisatorische Voraussetzungen erfüllt sein, was im vorgestellten Projekt der Fall war.

Abbildung ?? zeigt eine Eingabemaske der Benutzungsschnittstelle des implementierten Systems: der Einstieg in die Behandlungsakte. Auf den Eingabemasken der Benutzungsschnittstelle wurden die zugehörigen Piktogramme zur einfachen Wiedererkennung dargestellt. Die Piktogramme dienen in der gesamten Anwendung zur Kennzeichnung und zur Navigation zwischen den einzelnen Masken der Benutzungsschnittstelle. Zur Identifikation der Maske in Abb. ?? dient das Piktogramm (geöffnete Akte) und zur Navigation werden auf den Knöpfen der unteren Leiste einige Piktogramme aus dem Kooperationsdiagramm der Anforderungs-

Das Screenshot zeigt die Eingabemaske für eine Behandlungsakte. Die Maske ist in mehrere Bereiche unterteilt. Oben ist der Patient Name, Geschlecht und Geburtsdatum angegeben. Darunter sind Aufnahme- und Entlassungsdaten, die Dauer der Behandlung und die Station. Es folgen Felder für Gewicht, Größe, KOF, BMI, Hausarzt und Studienstatus. Die untere Leiste enthält Navigations- und Suchfunktionen.

Behandlungsakte einsehen: 18501	
Behandlungsakte 18501	
Von Mustermann, Hans ♂ männlich 31.01.90 ( 7 ) 11533	
Aufnahme:	14.03.1997 Aufnahmegrund: Z.n. Myokardinfarkt
Entlassung:	00.00.00 Behandlungsart: Stationär
Dauer:	0 Tage Station: M4
Gewicht:	78 kg Hausarzt:
Größe:	183 cm
KOF:	1,99 m <sup>2</sup> Üw.-Klinik:
BMI:	23,29
Zst. n. MI:	Ja Studien: keine
Zst. n. ACB:	Nein keine
Ändern ... Schließen	

**Abb. 9.** Eine Eingabemaske der entwickelten Anwendung: Einstieg in die Behandlungsakte. Die Piktogramme dienen zur Kennzeichnung und zur Navigation. Zur Kennzeichnung dient hier das Symbol für die geöffnete Patientenakte (links oben). In der unteren Leiste befinden sich Symbole zur Navigation entsprechend dem Dynamikmodell aus Abb. 6. Zusätzlich gibt es noch Suchfunktionen, die durch Fragezeichen an den entsprechenden Piktogrammen dargestellt werden. Die eigentlichen Daten werden im mittleren Hauptbereich der Maske eingegeben

analyse in Abb. 4, dem Analysemodell in Abb. 5 bzw. der Beschreibung der Dynamik in Abb. 6 verwendet.

Wie in Abschnitt 2.3 diskutiert wurde, ist es wichtig, daß beim Prototyping nicht nur Oberflächenmasken zur Verfügung stehen, sondern daß auch ein gewisses Maß an Funktionalität vorhanden ist, damit die Arbeitsprozesse am Prototypen nachvollzogen werden können. Die Implementierung erfolgte im vorgestellten Projekt auf einem Netzwerk von Apple Macintosh Rechnern auf Basis der Client/Server-Version der Datenbank 4D [1], so daß im Prototypen schon Datenbankfunktionalität verfügbar war. Die Strukturierung der Daten basiert auf dem Analysemodell aus Abb. 5 und wurde in Zusammenarbeit mit Statistikern der Universität Dortmund für die spätere statistische Auswertung hin optimiert. Für ad-hoc Auswertungen wurden einfache statistische Verfahren direkt im Informationssystem implementiert (Tortendiagramme, verschiedene Mittelwertberechnungen, etc). Für komplexe Auswertungen steht eine Export-Schnittstelle zum Statistiksystem SAS [10] zur Verfügung. Aus dem alten System konnten ca. 90% der Behandlungsda-



ten übernommen werden. Diese Altdaten wurden mit einem entsprechenden Vermerk versehen, um später differenzierte Auswertungen zu ermöglichen.

Auch im Handbuch werden die Piktogramme verwendet, um für die Anwender einen einfachen Wiedererkennungseffekt zu erreichen. Das Kooperationsdiagramm aus der Anforderungsanalyse (Abb. 4) und das Statechart für die Spezifikation der Dynamik in der Benutzungsschnittstelle (Abb. 6) dienen hier als Überblick über die Funktionalität der Applikation.

## 5 Zusammenfassung und Ausblick

Dieses Papier berichtet über die Erfahrungen mit dem Einsatz von Piktogrammen bei der Neuentwicklung eines medizinischen Informationssystems in einem interdisziplinären Kooperationsprojekt zwischen Informatikern, Statistikern und Mediziner. Die anwendungsspezifischen Piktogramme unterstützen dabei für die Anwender eine Nachvollziehbarkeit von der Anforderungsanalyse hin zum entwickelten System. Die Partizipation bei der Modellierung und das Wiedererkennen der Anforderungsspezifikation im Anwendungssystem (in der Form von anwendungsspezifischen Piktogrammen) hatte einen erheblichen Einfluß auf die Akzeptanz des neuen Informationssystems. Die frühzeitige Überprüfung der Ergebnisse der Anforderungsanalyse durch Ausführung und Bewertung von Prototypen durch strukturierte Fragebögen hat sich ebenfalls sehr positiv auf die Akzeptanz ausgewirkt.

Die Kooperationsdiagramme haben es uns auf einfache Art erlaubt, zusammen mit den Anwendern die Anforderungen an das neue Informationssystem auf der Ebene des Informationsflusses in der Klinik zu analysieren. Die Kooperationsdiagramme boten die Grundlage für die Strukturierung der Daten im Analysemodell, sowie die Beschreibung der Dynamik und die graphische Gestaltung der Benutzungsschnittstellen.

Die Erkenntnis, daß Prototyping eine geeignete Technik zur Einbindung der Anwender darstellt und daß das organisatorische Umfeld dazu eine interdisziplinäre Zusammenarbeit erlauben muß, um die Anwender angemessen an der Anwendungsentwicklung teilhaben zu lassen, hat sich im hier vorgestellten Projekt bestätigt. Zusätzlich konnten wir die Partizipation der Anwender durch den Einsatz von anwendungsspezifischen Piktogrammen in der Modellierung und in der prototypischen Implementierung unterstützen. Es ist natürlich nicht sinnvoll, für *jedes* Modellierungselement unbedingt ein eigenes Piktogramm einzuführen. In den in Abschnitt 4 präsentierten Modellausschnitten wurden teilweise relativ viele Piktogramme verwendet, um die Ideen zu illustrieren.

Aus Platzgründen wurde in diesem Papier nicht auf eine spezifische Werkzeugunterstützung für die Modellierung mit Piktogrammen und die Nachvollziehbarkeit eingegangen. Dazu verweisen wir auf [21]. Nachvollziehbarkeit in Entwicklungswerkzeugen dient in erster Linie den Entwicklern. Dieses Papier konzentriert sich auf Fragen der Nachvollziehbarkeit für die Anwender.

Das neu entwickelte Informationssystem befindet sich seit Februar 1997 im Einsatz und ist, auch aufgrund der parti-

zipativen Entwicklung, bei den Anwendern auf sehr gute Akzeptanz gestoßen. Die vorgestellten Analysetechniken sollen auch im größeren Rahmen in anderen Bereichen der Klinik eingesetzt werden, um weitere Erfahrungen damit zu sammeln.

*Danksagung.* An dieser Stelle sei Oliver Alsbach, Andreas Christmann, Klaus Emmerich und Ulf Radmacher für die sehr gute Zusammenarbeit in diesem interdisziplinären Projekt gedankt. Die Anmerkungen der Gutachter gaben wertvolle Hinweise zur besseren Strukturierung und Fokussierung des Beitrags. Das vorgestellte Projekt wurde durchgeführt während der Autor an der Universität Dortmund tätig war.

## Literatur

1. ACI Software Vertriebs GmbH: 4D Dokumentation. Neufahrn 1996
2. Balzert, H.: Methoden der objektorientierten Systemanalyse. Mannheim: BI-Wissenschafts-Verlag 1995
3. Bodemann, J., Hasselbring, W., Mehlstäubler, D., Jahnke, T., Röser, A.: Eine objektorientierte Problembereichsanalyse für die elektronische Patientenakte. In: Abstracts zur 41. GMDS-Jahrestagung, Bonn, 1996
4. Brami, R.: Icons: a unique form of painting. *ACM Interactions* **4**(5), 15–28 (1997)
5. Budde, R., Kautz, K., Kuhlenkamp, K., Züllighoven, H.: Prototyping — An Approach to Evolutionary System Development. Berlin: Springer 1992
6. Clement, A., den Besselaar, P. V.: A retrospective look at PD projects. *Commun. ACM* **36**(4), 29–37 (1993)
7. Davis, A.: Software requirements analysis and specification. Englewood Cliffs, NJ: Prentice Hall 1990
8. Deiters, W., Gruhn, V.: The FUNSOFT net approach to software process management. *International Journal of Software Engineering and Knowledge Engineering* **4**(2), 229–256 (1994)
9. DeMarco, T.: Structured Analysis and System Specification. Englewood Cliffs, NJ: Prentice Hall 1985
10. Dufner, J., Jensen, U., Schumacher, E.: Statistik mit SAS. Stuttgart: Teubner 1992
11. Finkelstein, A., Gabbay, D., Hunter, A., Kramer, J., Nuseibeh, B.: Inconsistency handling in multi-perspective specifications. *IEEE Trans. Softw. Eng.* **20**(8), 569–578 (1993)
12. Fowler, M., Scott, K.: UML Distilled: Applying the Standard Object Modeling Language. Object Technology Series. Reading, MA: Addison-Wesley 1997
13. Gittins, D.: Icon-based human-computer interaction. *International Journal of Man-Machine Studies* **24**(6), 519–543 (1986)
14. Gotel, O., Finkelstein, A.: An analysis of the requirements traceability problem. In: Proc. 1st International Conference on Requirements Engineering, pp. 94–101. Piscataway, NJ: IEEE Computer Society Press 1994
15. Gotel, O., Finkelstein, A.: Contribution structures. In: Proc. Second IEEE International Symposium on Requirements Engineering, pp. 100–10. Piscataway, NJ: IEEE Computer Society Press 1995
16. Guastello, S., Traut, M., Korienek, G.: Verbal versus pictorial representations of objects in a human-computer interface. *International Journal of Man-Machine Studies* **31**(1), 99–120 (1989)
17. Harel, D.: Statecharts: A visual formalism for complex systems. *Science of Computer Programming* **8**(3), 231–274 (1987)
18. Harel, D., Gery, E.: Executable object modeling with Statecharts. *Commun. ACM* **30**(7), 31–42 (1997)
19. Hasselbring, W.: Erfahrungen mit dem Einsatz von Objectory für vorlesungsbegleitende Übungen in der Softwaretechnik-Ausbildung. *Softwaretechnik-Trends* **16**(2), 10–15 (1996)
20. Hasselbring, W.: Federated integration of replicated information within hospitals. *International Journal on Digital Libraries* **1**(3), 192–208 (1997)
21. Hasselbring, W.: Nachvollziehbarkeit von den Anforderungen zum entwickelten System durch den Einsatz von Piktogrammen. *Software-Technik Memo Nr. 97*, Universität Dortmund, 1997

22. Jacobson, I., Christerson, M., Jonsson, P., Övergaard, G.: Object-Oriented Software Engineering – A Use Case Driven Approach. Reading, MA: Addison-Wesley 1992
23. Krabbel, A., Wetzel, I., Ratuski, S.: Objektorientierte Analysetechniken für übergreifende Aufgaben. In: Proc. GI-Fachtagung Softwaretechnik 96, Koblenz, 1996, Softwaretechnik-Trends 16/3, pp. 65–72
24. Krabbel, A., Wetzel, I., Ratuski, S.: Anforderungsanalyse für Krankenhausinformationssysteme: Definition von Kernsystem und Ausbaustufen. In: Hasselbring, W. (ed.), Erfolgsfaktor Softwaretechnik für die Entwicklung von Krankenhausinformationssystemen. Münster: Krehl 1997
25. Nissen, H., Jeusfeld, M., Jarke, M., Zemanek, G., Huber, H.: Managing multiple requirements perspectives with metamodels. IEEE Software **13**(3), 37–48 (1996)
26. Ohnishi, A.: Visual software requirements language based on communication model. In: Proc. Third Int'l Workshop on Requirements Engineering: Foundation of Software Quality, REFSQ'97, Barcelona, Spanien, 1997
27. Ohnishi, A.: VRDL: A Visual Software Requirements Language. In: Proc. Third Biennial World Conference on Integrated Design & Process Technology (IDPT'98): Design, Software, Process Development and Management, Berlin, 1998, pp. 257–264
28. Pohl, K.: PRO-ART: Enabling Requirements Pre-Traceability. In: Proc. Second IEEE International Symposium on Requirements Engineering, pp. 76–84. Piscataway, NJ: IEEE Computer Society Press 1995
29. Pohl, K., Jacobs, S.: Traceability between cross-functional teams. NATURE Report Series 94-13, RWTH Aachen, 1994. (Published at 1st Intl. Conf. on Concurrent Engineering, Pittsburgh, USA, August 1994)
30. Ramesh, B., Stubbs, C., Powers, T., Edwards, M.: Requirements traceability: theory and practice. Annals of Software Engineering **3**, 397–415 (1997)
31. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorenzen, W.: Object-Oriented Modelling and Design. Englewood Cliffs, NJ: Prentice Hall 1991
32. Symon, G., Fitter, M., Radstone, C., Kunkler, I., Hancock, B.: The process of deriving requirements for a hospital information system. Behaviour and Information Technology **11**(3), 131–140 (1992)
33. Winter, A., Zimmerling, R., Bott, O., Gräber, S., Haas, P., Hasselbring, W., Haux, R., Heinrich, A., Jaeger, R., Kock, I., Möller, D., Penger, O.-S., Prokosch, H.-U., Ritter, J., Terstappen, A., Winter, A.: Das Management von Krankenhausinformationssystemen: Eine Begriffsdefinition. Informatik, Biometrie und Epidemiologie in Medizin und Biologie **29**(2), 93–105 (1998)
34. Wood, J., Silver, D.: Joint Application Development, 2nd ed. New York: Wiley 1995



*Wilhelm Hasselbring* ist Assistenzprofessor am Infolab der Universität Tilburg, Niederlande. Nach Abschluß seines Informatikstudiums an der Technischen Universität Braunschweig arbeitete er von 1989 bis 1994 als wissenschaftlicher Mitarbeiter an der Universität GH Essen und der Universität Dortmund im Bereich Softwaretechnik. Er promovierte 1994 an der Universität Dortmund, wo er dann bis August 1998 als wissenschaftlicher Assistent (C1) tätig war. Seine aktuellen Arbeitsschwerpunkte liegen im Bereich der Softwaretechnik für verteilte und kooperative Informationssysteme. In diesem

Zusammenhang arbeitet er mit Krankenhäusern und Kliniken bei der Entwicklung von Krankenhausinformationssystemen zusammen, wobei die systematische Anforderungsermittlung und die softwaretechnische Realisierung von flexiblen Architekturen für verteilte Informationssysteme zentrale Aufgaben darstellen.