

Funkat, Gert; Kaeding, Anne-Kathrin; Detschew, Vesselin; Specht, Martin:

## Bettenzentriertes adaptierbares Stationsmanagement

---

<i>Zuerst erschienen in:</i>	Biomedizinische Technik = Biomedical Engineering. - Berlin [u.a.] : de Gruyter. - 46 (2001), S1, S. 454-455. Jahrestagung der Deutschen Gesellschaft für Biomedizinische Technik (DGBM) im VDE ; 35 (Bochum) : 2001.09.19-21
<i>Erstveröffentlichung:</i>	2001
<i>Datum Digitalisierung:</i>	2009-08-06
<i>ISSN (online):</i>	1862-278X
<i>ISSN(print)</i>	0013-5585
<i>DOI:</i>	<a href="https://doi.org/10.1515/bmte.2001.46.s1.454">10.1515/bmte.2001.46.s1.454</a>
<i>[Zuletzt gesehen:</i>	2019-12-06]

„Im Rahmen der hochschulweiten Open-Access-Strategie für die Zweitveröffentlichung identifiziert durch die Universitätsbibliothek Ilmenau.“

“Within the academic Open Access Strategy identified for deposition by Ilmenau University Library.”

„Dieser Beitrag ist mit Zustimmung des Rechteinhabers aufgrund einer (DFG-geförderten) Allianz- bzw. Nationallizenz frei zugänglich.“

„This publication is with permission of the rights owner freely accessible due to an Alliance licence and a national licence (funded by the DFG, German Research Foundation) respectively.“



## BETTENZENTRIERTES ADAPTIERBARES STATIONSMANAGEMENT

G. Funkat<sup>1</sup>, A.-K. Kaeding<sup>1</sup>, V. Detschew<sup>1</sup>, M. Specht<sup>2</sup>

<sup>1</sup>Institut f. Biomedizinische Technik und Informatik, TU Ilmenau, Deutschland

<sup>2</sup>Klinik für Anästhesiologie und Intensivtherapie, Klinikum der FSU Jena

gf@informatik.tu-ilmenau.de

### EINLEITUNG

Die aktuelle medizinische Versorgung im Krankenhaus ist eng an eine Informationsverarbeitung gekoppelt, die ständig eine große Anzahl von Daten und Informationen erzeugt und verarbeitet. Diese Daten und Informationen müssen archiviert werden, viel problematischer ist allerdings die situationsrelevante Lokalisierung und Repräsentation (graphische Benutzerschnittstelle). In vielen Häusern hat sich der Einsatz von elektronischen Patientenakten durchgesetzt. Diese Systeme fokussieren auf die Erfassung und Verwaltung der patientenbezogenen Verlaufsdokumentation.

### ANFORDERUNGEN

Für die Intensivversorgung der Klinik für Anästhesiologie und Intensivtherapie der FSU Jena bestand die

Solche Organisationsabläufe beinhalten unter anderem die Belegung, Sperrung und Aufnahmevorbereitung für ein Bett sowie die Entlassung und Verlegung eines Patient. Ist ein Bett als „belegt“ markiert, so sind darüber alle weiteren Patienteninformationen verfügbar (siehe Abbildung 1). Weiterhin sind die folgenden Anforderungen spezifiziert worden:

- Datenverwaltung von Patienten, die auf der Station behandelt werden oder behandelt wurden.
- Konfigurierbarkeit des Systems für die Verwaltung mehrerer Stationen mit veränderlicher Bettenzahl.
- Grafische Benutzeroberfläche entsprechend den Bedürfnissen des medizinischen Personals.
- Geeignete Visualisierung der Datenverwaltung. Zentraler Punkt für den Zugriff auf die Patienteninformationen soll das Bett sein, in dem der Patient liegt.

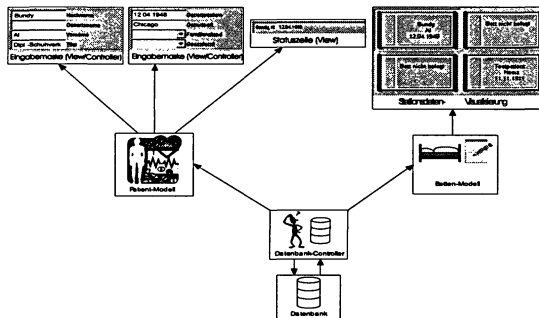


Abbildung 1: Visualisierung und Informationsfluss für das Stationsinformationssystem

Anforderung, ein adaptierbares Stationsmanagement zu entwickeln, um die Organisation der Patienten bettenzentriert zu realisieren. Diese Bettenzentrierung bedeutet, dass die zentrale Verwaltungssicht an der Struktur und aktuellen Situation der Intensivbetten orientiert ist. Damit soll ein direkter Zugang zu den jeweiligen Organisationsabläufen der Stationsbetten möglich sein.

- Zentrale Speicherung der Daten mit konkurrierendem Zugriff von mehreren Terminals aus.
- Gute Wartungseigenschaften des Systems hinsichtlich zukünftig notwendiger Erweiterungen, Minimierung des Einarbeitungsaufwandes.

## TECHNIKEN UND METHODEN

Ein Hauptziel der Entwicklung ist gewesen, ein Framework zu implementieren, welches es auch Nichtinformatikern ermöglicht, die endgültige Applikation aufzubauen. Dazu ist das Framework in einer Mehrschichten-Architektur entwickelt worden. Client-seitig bedeutet das den Aufbau mit vier Schichten: Graphische Benutzerschnittstelle (GUI), Cache, Datenbank-Adapter und CORBA-Schnittstelle. Die oberste Schicht, die GUI, dient zur Eingabe und Visualisierung der Daten. In der zweiten Schicht, dem Cache, werden die eingegebenen Daten verarbeitet bzw. zur Anzeige vorbereitet. Des Weiteren werden dort die lokalen Daten des Client verwaltet. Die dritte Schicht ist die clientseitige Schnittstelle zur Datenbank. Sie dient zur zentralen Datenverwaltung. Die vierte Schicht wird durch die client-seitige CORBA-Schnittstelle gebildet. Sie realisieren die Netzwerkkommunikation.

CORBA (Common Object Request Broker Architecture) ist ein Standard für verteilte Systeme. Dieser Standard wurde von der OMG (Object Management Group) festgelegt und wird ständig erweitert [1]. Im Rahmen dieses Projektes wurde eine Realisierung dieses Standards mit dem Namen ORBacus der Object Oriented Concepts, Inc. verwendet [2].

Die Kommunikation der Clients mit dem Server-Modul erfolgt CORBA. Der Server ist ebenfalls mehrschichtig aufgebaut. Die Kommunikation mit den Clients wird analog mit einer CORBA-Schnittstelle gesichert. Darunter liegen Schichten für das Nutzermanagement und die Client-Aktualisierung.

Für die hohe Flexibilität der gesamten Applikation ist auch die Gestaltung der Datenbankschnittstelle verantwortlich. Sie ist dergestalt konzipiert, dass auf dem serverseitigen Datenbank-Controller immer ein einheitliches Datenmodell verwendet wird und lediglich die Besonderheiten der verschiedenen Datenbanken implementiert werden müssen. Zum gegenwärtigen Entwicklungsstand ist die objektorientierte Datenbank POET angeschlossen. Der entscheidende Vorteil eines objektorientierten Datenbankmanagementsystems ist die Fortführung des objektorientierten Entwurfstiles. Die komplexen Objektstrukturen des Datenmodells müssen nicht auf relationale Strukturen aufgelöst werden. Die im Client eingegebenen Daten müssen nicht zur zentral verwaltet.

Ein großer Bestandteil der Entwicklungsarbeit nahm die Gestaltung der Benutzerverwaltung ein, die benutzerspezifisches Verhalten der Anwendung, Protokollierung und Sicherheit ermöglicht.

## DISKUSSION

Bei der Schaffung der vorgestellten Client-Server-Architektur für ein Stationsinformationssystem wurde besonderer Wert auf eine einfache Softwarestruktur gelegt, da das System leicht zu erweitern sein soll. Das Erscheinungsbild der Applikation wurde nach den

Vorstellungen des medizinischen Personals in Form eines Karteikastens gestaltet. Die Anwendung reagiert kontextbezogen, d. h. die Bedienmöglichkeiten zu einem Zeitpunkt sind vom dann aktuellen Zustand des Clients abhängig.

Für den Aufbau und die Erweiterung der gesamten Applikation muss der Anwender lediglich an Hand einer Handlungsanweisung die erforderlichen grafischen Elemente an das Datenmodell koppeln. Alle weiteren Client-Schichten sind für ihn transparent, was bedeutet, dass alle Mechanismen für die Client-Synchronisation, Zugriffs- und Nutzermanagement und Datenspeicherung automatisch angepasst werden.

Im Ergebnis des Projektes steht eine Lösung, die durch Nichtinformatiker ausgebaut und gepflegt werden kann und sich vor allem durch eine hohe Robustheit gegen Erweiterungen und Änderungen des Datenmodells auszeichnet.

## LITERATURHINWEISE

- [1] Baker, S.: *CORBA Distributed Objects: Using Orbix*. Addison-Wesley. ISBN 0-201-92475-7
- [2] ORBacus for C++ and Java. Version 4.0. Benutzerhandbuch. Object Oriented Concepts, Inc.