

Eine Taxonomie für Kommunikationsserver im Krankenhaus

A classification of Communication Server in hospital

M. Lange¹, H.U. Prokosch¹, W. Hasselbring²

Zusammenfassung

In den Krankenhäusern breitet sich die EDV immer weiter aus. Krankenhäuser sind komplexe Organisationen, deren Teilsysteme sich relativ unabhängig und eigenbestimmt entwickeln, sich gleichzeitig aber auch aufeinander beziehen und einen gewissen Grad an Determiniertheit durch die übrigen Teilsysteme des Krankenhauses behalten. Durch die weitgehende Unabhängigkeit und die fehlende Verfügbarkeit von integrierten Software-Systemen, die die Bedürfnisse aller Teilsysteme berücksichtigen, entstehen viele isolierte EDV-Lösungen. Durch die gegenseitigen Abhängigkeiten – insbesondere auf Datenebene – entsteht andererseits die Anforderung zur Kopplung autonomer, heterogener Software-Systeme.

Um die Vielzahl unterschiedlicher Schnittstellen zwischen den einzelnen Software-Systemen zu managen, werden in zunehmendem Maße als technische Grundlage zur Kopplung Kommunikationsserver eingesetzt. Dieser Beitrag zeigt auf, welche Anforderungen dabei an einen Kommunikationsserver gestellt werden. Aus diesen Anforderungen werden Komponenten einer Kommunikationsserver-Software abgeleitet. Diese Komponenten bilden schließlich die Grundlage für die Entwicklung einer Taxonomie für Kommunikationsserver, sowie einen Vergleich einiger kommerzieller Kommunikationsserver auf Basis eines Anforderungskatalogs, der anhand der Taxonomie erstellt wurde.

Schlüsselwörter

Taxonomie, Kommunikationsserver, Krankenhaus-Informationssysteme

Summary

The subsystems of the organisation „hospital“ are developing relativ independent and self-determinating. On the other hand there are many relationships between and determinism by these subsystems. The comprehensive independence and the missing availability of integrated software systems wich consider the requirements of all subsystems promote the development of many isolated EDP systems. On the other hand the independence – especially on the data layer – entail the requirement to link these autonomous and heterogenous software systems.

¹ Institut für Medizinische Informatik und Biomathematik, WWU Münster

² INFOLAB, Tilburg University (KUB)

Increasingly Communication Server are used as technical basis to manage the various interfaces between the different software systems. Our paper describes the requirements of a Communication Server. From this requirements the necessarily components of a Communication Server are derived. This components build the basis for a classification and a comparison of commercial Communication Server.

Key words

Classification, Communication Server, Hospital Information Systems

Einleitung

Die aktuelle Situation an den deutschen Krankenhäusern ist durch eine sich immer weiter ausbreitende EDV geprägt. Neben den ursprünglich administrativen und funktionsorientierten Bereichen sind in zunehmendem Maße auch die stationären und ambulanten Abteilungen betroffen. Eine Kopplung dieser Systeme über einen Ausbau des Rechnerverbundes durch weitere Punkt-zu-Punkt-Verbindungen ist nicht mehr praktikabel. Jedes neue System bringt eine Vielzahl neuer Software-Schnittstellen mit sich.

Kommunikationsserver sollen hier eine Abhilfe schaffen. Die einzelnen EDV-Systeme in den Abteilungen werden nun nicht mehr direkt miteinander verbunden, sondern alle Kommunikationsverbindungen werden über den Kommunikationsserver geführt. Der Kommunikationsserver stellt eine einheitliche Implementierungsumgebung für die Schnittstellen zwischen den Systemen zur Verfügung.

Für die Realisierung eines Kommunikationsservers sind verschiedene Software-Architekturen denkbar. Neben einer zentralen Lösung, ist auch ein verteilter Ansatz denkbar. Die zur Zeit auf dem Markt befindlichen Produkte benutzen unterschiedliche Architekturen und unterstützen verschiedene Bandbreiten an Funktionen.

Viele Krankenhäuser stehen zur Zeit vor der Entscheidung, welche Bereiche der elektronischen Kommunikation zwischen den EDV-Systemen sie über einen Kommunikationsserver realisieren lassen [4]. Hierbei spielt eine Rolle, welche Architektur - zentral oder verteilt - sich im jeweiligen DV-Umfeld am besten einpassen läßt und die Ansprüche an die Performanz und die Sicherheit am ehesten erfüllt. Es geht also darum zu entscheiden, welches der auf dem Markt verfügbaren Produkte die eigenen Anforderungen am besten abdeckt.

In diesem Beitrag kann nur auf die wesentlichen Funktionen, Komponenten und Architekturansätze für Kommunikationsserver eingegangen werden. Eine vertiefende Betrachtung findet der interessierte Leser in [6].

1 Kommunikationsserver

Die Ursprünge der Entwicklung der Kommunikationsserver liegen in den USA. Dort sind Bezeichnungen wie *Integration Engine*, *Interface Engine* oder *Connection Server* gebräuchlich. Im deutschen Sprachgebrauch hat sich die Bezeichnung „*Kommunikationsserver*“ als Übersetzung der englischen Begriffe durchgesetzt.

Unklar ist bisher geblieben, was ein Kommunikationsserver zu leisten hat und welche Aufgaben er in der Kommunikation zwischen DV-Systemen zu erfüllen hat. Dies gilt insbesondere für das Einsatzgebiet „Krankenhaus“. Unumstritten sind die Basisdienste, wie beispielsweise das Empfangen und Weiterleiten der Nachrichten. Auch die Konvertierung der Kommunikationsprotokolle (Schicht 7 im ISO/OSI-Modell [10]) durch den Kommunikationsser-

ver gehört im allgemeinen zu seinen Standardaufgaben. Doch wie steht es um über diese Basisdienste hinausgehende Aufgaben? So zum Beispiel die Synchronisation des Nachrichtenaustausches oder die kontrollierte Steuerung von Kommunikationsprozessen im Sinne der Auftragskommunikation [8]. Im Krankenhaus werden überwiegend von Haus aus kommunizierunfähige DV-Systeme eingesetzt. Wo sind hier die Grenzen des Aufgabengebietes der Kommunikationsserver zu ziehen?

Unter einem *Kommunikationsserver* soll hier ein Software-System verstanden werden, das den Aufbau, den Betrieb und die Kontrolle von Kommunikationsverbindungen zwischen autonomen, heterogenen, in bezug auf Kommunikationsprozesse kommunizierfähigen DV-Systemen ermöglicht.

Nach dieser Definition gehört die Synchronisation der Nachrichtenübertragung zum Aufgabengebiet eines Kommunikationsservers. Die Zuständigkeitsgrenze zwischen Subsystem und Kommunikationsserver verläuft demnach zwischen der Bereitstellung und der Übertragung der Nachrichten. Ein DV-System soll hier als *kommunizierfähig* bezeichnet werden, wenn es in der Lage ist, die für den Kommunikationsprozeß notwendigen Nachrichten zu erzeugen bzw. entgegenzunehmen und die in der Nachricht enthaltenen Daten darzustellen.

1.1 Aufgaben eines Kommunikationsservers

Die Aufgaben eines Kommunikationsservers lassen sich folgendermaßen umreißen:

1. Nachrichten von einem Subsystem entgegennehmen

Die Nachrichten werden von dem Subsystem erzeugt und entweder dateibasiert oder dateilos durch ein entsprechendes Transportprotokoll vom Kommunikationsserver eingelesen. Nachrichten, die vom Kommunikationsserver empfangen werden, werden auch als *Inbound*-Nachrichten bezeichnet.

2. Empfänger der Nachrichten ermitteln

Im nächsten Schritt müssen die Empfänger der Nachricht ermittelt werden. Die Empfänger sind entweder in der Nachricht codiert oder werden implizit durch den Typ der Nachricht festgelegt (z.B.: Nachrichten vom Typ „Stammdaten eines neuaufgenommenen Patienten“ werden immer an das Laborsystem gesandt).

3. Übermitteln der Nachrichten an die Empfänger

Unter Umständen müssen die Nachrichtenstrukturen der Inbound-Nachrichten in die Kommunikationsprotokolle der Empfänger transformiert werden. Nachrichten, die den Kommunikationsserver verlassen, werden auch als *Outbound*-Nachrichten bezeichnet.

Um diesen Aufgaben gerecht zu werden, muß ein Kommunikationsserver einige grundlegende Funktionen bereitstellen. Diese sollen im folgenden kurz dargestellt werden. Die Reihenfolge stellt keine Bewertung der Funktionen in bezug auf ihre Relevanz für einen Kommunikationsserver dar, sondern spiegelt in etwa die Schritte bei der Konfiguration einer Kommunikationsverbindung bzw. dem Verarbeitungsprozeß einer Nachricht im Kommunikationsserver wider. Zum Teil sind diese Funktionen auch in [3] beschrieben.

F 1: Definition der logischen Kommunikationsbeziehungen

Bevor Kommunikationsverbindungen über einen Kommunikationsserver eingerichtet werden können, müssen die logischen Kommunikationsbeziehungen zwischen den Subsystemen des Krankenhauskommunikationssystems (KKS) definiert worden sein. Alle Kommunikationsprozesse mit den beteiligten Subsystemen und den ausgetauschten Nachrichten müssen identifiziert und definiert werden. Die Aufgabe des Kommunikationsservers ist es, geeignete Funktionen zur Abbildung der logischen Kommunikationsbeziehungen im Kommunikationsserver bereitzustellen. Dies geschieht in der Regel über die Definition von Routing-Tabellen, in denen festgelegt wird, welche Nachrichten zwischen welchen Subsystemen ausgetauscht werden.

F 2: Definition der Transportprotokolle

Um die Nachrichten zwischen den Subsystemen und dem Kommunikationsserver austauschen zu können, müssen Kommunikationsverbindungen zwischen den Subsystemen und dem Kommunikationsserver eingerichtet werden. Eine *Kommunikationsverbindung* besteht dabei aus der physikalischen Leitung über das Netz und einem Transportprotokoll, das den Nachrichtenaustausch über die Verbindung regelt. Während die physikalische Verbindung von dem KKS zugrundeliegenden Netzwerk erbracht wird, ist die Definition und Ausführung der Transportprotokolle Aufgabe des Kommunikationsservers.

F 3: Definition der Kommunikationsprotokolle

Neben der Definition und dem Aufbau der Kommunikationsverbindungen müssen die Strukturen der ausgetauschten Nachrichten definiert werden. Dies geschieht über die Definition eines Kommunikationsprotokolls für jede Kommunikationsverbindung. Die Kommunikationsprotokolle definieren nicht nur die Strukturen der Nachrichten, sondern durch sie wird auch die Semantik der Daten in den Nachrichten festgelegt. Dies ist für die spätere Definition der Transformationen der Nachrichten in andere Kommunikationsprotokolle erforderlich. Die Semantik der Datenfelder wird allerdings nicht in der Nachricht selbst kodiert, sondern sie wird lediglich in der Dokumentation der Kommunikationsprotokolle festgehalten.

Während sich diese ersten drei Funktionen mit der Konfigurierung der Kommunikationsverbindungen befassen, ergeben sich die folgenden Funktionen aus dem Betrieb und der Kontrolle der Kommunikationsprozesse durch den Kommunikationsserver:

F 4: Übertragung der Nachrichten zwischen den Subsystemen und dem Kommunikationsserver

Die Aufgaben des Kommunikationsservers beschränken sich hier auf die Übertragung der Nachrichten über die Ausführung der Transportprotokolle und den darin definierten Maßnahmen zur Übertragungssicherung und zur Synchronisation.

F 5: Nachrichtenidentifikation

Der erste Schritt der Nachrichtenverarbeitung im Kommunikationsserver besteht aus der Identifikation des Nachrichtentyps. In standardisierten Kommunikationsprotokollen sind die Nachrichtentypen bereits festgelegt (in HL7 [1] zum Beispiel über das MSH-Segment). Aber auch in proprietären Kommunikationsprotokollen müssen Nachrichtentypen und Identifikationsregeln für diese Typen vorgesehen sein. Der Kommunikationsserver muß also die

Definition und die Anwendung von Identifikationsregeln für die einzelnen Nachrichten aus den Kommunikationsprotokollen ermöglichen. Der Nachrichtentyp wird für die weitere Nachrichtenverarbeitung benötigt.

F 6: Ermittlung der Empfänger der Nachrichten (*Routing*)

Bei der Definition der logischen Kommunikationsbeziehungen wird festgelegt, welche Nachrichten zwischen welchen Subsystemen ausgetauscht werden. Somit ergeben sich meistens die Empfänger einer Nachricht implizit aus dem identifizierten Nachrichtentyp. Die Aufgabe des Kommunikationsservers beschränkt sich somit auf eine Auswertung der Definition der logischen Kommunikationsbeziehungen, die im allgemeinen in Form von Routing-Tabellen erfolgt. Für ein flexibles Multi- und Broadcasting ist aber auch eine dynamische Ermittlung der Empfänger aus dem Inhalt einer Nachricht, zum Beispiel aus speziellen Datenfeldern, denkbar.

F 7: Nachrichtentransformation

Da die einzelnen Subsysteme möglicherweise unterschiedliche Kommunikationsprotokolle verwenden, wird eine Transformierung der Nachrichten in die verschiedenen Protokolle innerhalb des Kommunikationsservers notwendig. Neben einer Transformierung der Nachrichtenstrukturen kann auch eine Werttransformation einzelner Datenfelder notwendig werden. So zum Beispiel die Transformierung der Darstellung eines Datums aus dem Format „JJMMTT“ in das Format „TT.MM.JJJJ“.

F 8: Fehlermanagement

Beim Fehlermanagement geht es um eine effektive Behandlung von inhaltlichen und strukturellen Fehlern der Nachrichten. Neben dem Erkennen der Fehler müssen Regeln definiert werden, die festlegen, was mit den falschen Nachrichten geschehen soll.

F 9: Persistenzsicherung

Nachrichten dürfen innerhalb des Kommunikationsservers nicht verloren gehen. Durch Zwischenspeichern von Nachrichten in persistenten Queues oder ähnlichen Speicherstrukturen ist die Persistenz der Nachrichten auch nach einem Systemabsturz vom Kommunikationsserver zu gewährleisten.

F 10: Überwachen der Kommunikationsverbindungen (*Monitoring*)

Eine wichtige Aufgabe des Kommunikationsservers ist neben der Verarbeitung der Nachrichten die Überwachung der Kommunikationsverbindungen. Darunter ist sowohl die Protokollierung der Vorgänge in Log-Dateien (*Logging*) als auch eine Kontrolle der Verbindungen zu verstehen. So sollte der Kommunikationsserver über Alarmierungsfunktionen verfügen, die zum Beispiel den Ausfall einer Verbindung zu einem Subsystem erkennen und einen entsprechenden Alarm auslösen können.

F 11: Testen der Konfigurationen

Aufgabe des Kommunikationsservers ist es, geeignete Funktionen zum Testen der konfigurierten Kommunikationsverbindungen bereitzustellen. Eine Möglichkeit wäre die Simulation einer Kommunikationsverbindung, so daß mit Hilfe von Testnachrichten der Verarbeitungsprozeß überprüft werden kann.

1.2 Komponenten eines Kommunikationsservers

Unter einer *Komponente* soll hier der Teil der Kommunikationsserver-Software verstanden werden, der für die Ausführung einer bestimmten Funktionalität zuständig ist. In einer Komponente sollen somit logisch zusammengehörende Funktionen zusammengefaßt werden. Für eine weitere Strukturierung können auch Teilkomponenten gebildet werden.

K 1: Serverkomponente

Die Serverkomponente umfaßt alle Funktionen, die für den Austausch der Nachrichten zwischen den Subsystemen und die Nachrichtenverarbeitung im Kommunikationsserver notwendig sind. Sie läßt sich weiter in eine *Verbindungskomponente* und eine *Verarbeitungskomponente* unterteilen. Die Verbindungskomponente leitet eine empfangene Nachricht zur Verarbeitung an die Verarbeitungskomponente weiter. Diese übermittelt die möglicherweise transformierte Nachricht an die Verbindungskomponente zurück. Die Verbindungskomponente ist somit für die Ausführung der Funktion F4, die Verarbeitungskomponente für die Ausführung der Funktionen F5 bis F7 zuständig.

K 2: Steuerungskomponente

Die Steuerungskomponente umfaßt alle Funktionen, die für die Steuerung des Nachrichtenflusses im Kommunikationsserver notwendig sind. Hierzu gehören auch die Funktionen F8 und F9.

K 3: Überwachungskomponente

Alle Funktionen, die die Überwachung der Kommunikationsverbindungen (*Monitoring F10*) ermöglichen, werden in dieser Komponente zusammengefaßt. Sie läßt sich in eine *Monitorkomponente* für die graphische Darstellung der Kommunikationsserver-Verbindungen, eine *Logging-Komponente* und eine *Archivierungskomponente* (Archivierung von Nachrichten zur Fehleranalyse) unterteilen.

K 4: Konfigurationskomponente

Alle bisher vorgestellten Komponenten müssen vom Administrator eines Kommunikationsservers konfiguriert werden. Dies gilt vor allem für die Serverkomponente. Aber auch das Logging und das Fehlermanagement bedürfen in der Regel einer Anpassung durch den Administrator. Alle Funktionen, die für die Konfigurierung notwendig sind, werden in dieser Komponente zusammengefaßt.

K 5: Testkomponente

Die Testkomponente stellt allgemein Funktionen zum Testen der über den Kommunikationsserver eingerichteten Kommunikationsverbindungen (Funktion F11) zur Verfügung. Diese Testfunktionen sollten unabhängig von dem eigentlichen Betrieb des Kommunikationsservers anwendbar sein.

Es sei nochmals darauf hingewiesen, daß die vorgestellten Komponenten lediglich eine logische Zusammenfassung der Funktionen eines Kommunikationsservers darstellen. Bei tatsächlich implementierten Kommunikationsservern werden die Funktionen in unterschiedlicher Weise auf Programm-Module verteilt. Eine mögliche Unterteilung eines Kommunikationsservers in einzelne Module wird in der Abbildung 1 dargestellt. In [4] wird ein weiteres Modell eines Kommunikationsservers in OMT-Notation [9] vorgestellt.

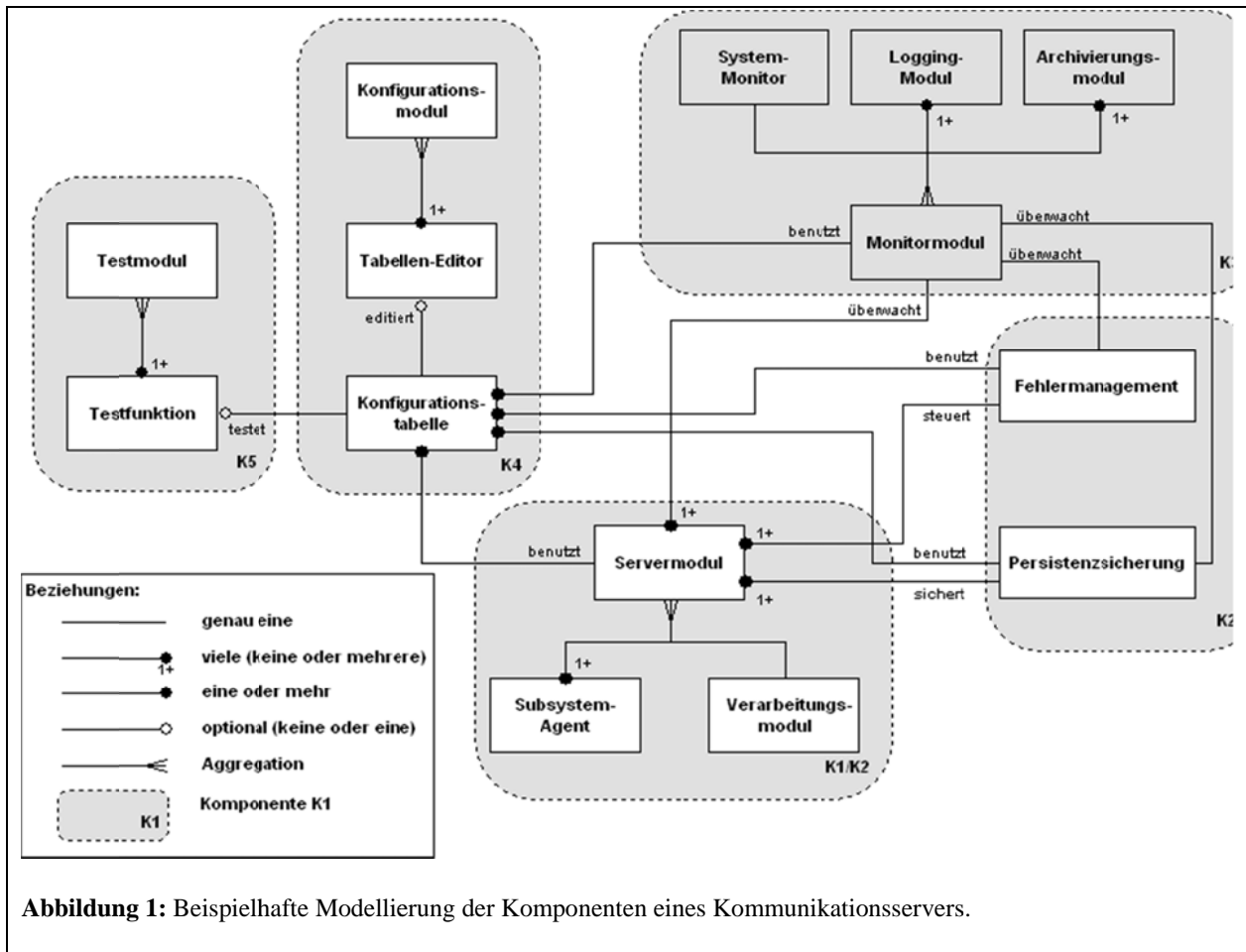


Abbildung 1: Beispielhafte Modellierung der Komponenten eines Kommunikationsservers.

2 Taxonomie

In diesem Abschnitt soll untersucht werden, wie die Funktionen der Komponenten auf Programm-Module verteilt werden können und welche System-Architekturen dabei für die Kommunikationsserver entstehen. Dabei soll hier nur die Serverkomponente (Komponente K1) betrachtet werden. Sie stellt die grundlegenden Funktionen eines Kommunikationsservers bereit, die Funktionen für den Nachrichtempfang, die Nachrichtenverarbeitung und -weiterleitung an die Subsysteme. Die Funktionen dieser Komponente und ihrer Teilkomponenten lassen sich auf sehr unterschiedliche Weise auf Programm-Module verteilen. Die dabei entstehenden Module beeinflussen als wichtigste Bestandteile entscheidend die Architektur eines Kommunikationsservers. Von der Architektur wiederum hängen die Performanz und die Verfügbarkeit des Kommunikationsservers ab. So dürfte ein Kommunikationsserver, der alle Funktionen der Serverkomponente in nur einem Modul vereinigt, kaum einen akzeptablen Nachrichtendurchsatz erreichen. Viele verschiedene Verarbeitungswege ermöglichen eine nebenläufige Verarbeitung der Nachrichten, die sich positiv auf die Performanz auswirken kann. Ein Kommunikationsserver, der für jedes Subsystem eigene, verteilbare Module mit den Funktionen der Verbindungs- und Verarbeitungskomponente bereitstellt, wird ein sehr hohes Maß an Verfügbarkeit bieten. Die Konsequenz hieraus ist aber ein hoher administrativer Aufwand, der sich negativ auf die Performanz auswirken kann.

Die Architektur, die sich aus der Aufteilung der Funktionen der Serverkomponente auf Programm-Module ergibt, liefert die Grundlage der Taxonomie für Kommunikationsserver, die im folgenden dargestellt werden soll.

2.1 Dimensionen der Taxonomie

In [2] und [7] werden Taxonomien entlang der Dimensionen *Verteiltheit*, *Heterogenität* und *Autonomie* für verteilte und föderative Datenbanksysteme vorgestellt. In Anlehnung daran unterteilt die hier dargestellte Taxonomie die Kommunikationsserver entlang der folgenden Dimensionen:

1. der *Verteiltheit* der Funktionen des Servermoduls,
2. der *Nebenläufigkeit* der Nachrichtenverarbeitung, sowie
3. der *Heterogenität* der Module des Kommunikationsservers.

Die drei Dimensionen der Taxonomie werden im folgenden näher spezifiziert.

2.1.1 Verteiltheit des Servermoduls

Die Motivation für diese Dimension wurde bereits in der Einleitung zu diesem Abschnitt dargestellt: Die Aufteilung der Funktionen des Servermoduls auf mehrere Programme ist eng mit der Performanz und der Verfügbarkeit des Kommunikationsservers verbunden. Die Funktionen der Verbindungskomponente werden von allen in [6] analysierten Kommunikationsservern auf verschiedene Module oder Programme verteilt. Dabei wird in der Regel für jedes an den Kommunikationsserver angeschlossene Subsystem mindestens ein Modul bereitgestellt, das die Übertragung der Nachrichten zu diesem Subsystem realisiert. Diese Programme werden auch als *Subsystem-Agenten* bezeichnet.

Auch die Funktionen der Verarbeitungskomponente lassen sich auf verschiedene Module aufteilen. Diesen Aspekt berücksichtigt die zweite Dimension der Taxonomie.

Die bei der Aufteilung der Module entstehenden Programme können nun auch im klassischen Sinne verteilt werden, also auf verschiedenen Rechnern ausgeführt werden. Somit läßt sich eine parallele Ausführung der Funktionen des Kommunikationsservers erreichen, die sich positiv auf die Performanz auswirken kann. Auch die Verfügbarkeit des Kommunikationsservers erhöht sich, wenn Kommunikationsverbindungen parallel betrieben werden können.

Bei der Verteiltheit der Funktionen der Serverkomponente sollen die folgenden Ausprägungen unterschieden werden:

1. **ungeteilt:**

Bei dieser Ausprägung werden alle Funktionen des Servermoduls von einem Programm realisiert. Die Performanz eines solchen Kommunikationsservers dürfte keine akzeptablen Werte erreichen, vor allem dann, wenn viele Subsysteme miteinander verbunden werden müssen, wie es in einem KKS der Fall ist.

2. **lokal-geteilt:**

Hier werden die Funktionen des Servermoduls auf verschiedene Programme aufgeteilt, aber alle Programme müssen auf einem Rechner lokalisiert sein. Dies ist zum Beispiel notwendig, wenn für die Programme *Threads* anstelle von vollwertigen Prozessen verwendet werden.

3. **verteilt:**

Bei dieser Ausprägung lassen sich die Programme, die die Funktionen des Servermoduls enthalten, im Netzwerk verteilen. So können zum Beispiel die Subsystem-Agenten auf die Rechner der Subsysteme ausgelagert werden. Eine Verteilung der Programme eines Kommunikationsservers setzt eine Interprozeßkommunikation voraus, die über Rechnergrenzen erfolgen kann.

Aus den drei dargestellten Ausprägungen lassen sich nun die folgenden drei Grade für diese erste Dimension der Taxonomie ableiten:



2.1.2 Nebenläufigkeit der Nachrichtenverarbeitung

Lassen sich die Funktionen des Servermoduls so auf Prozesse verteilen, daß mehrere getrennte Verarbeitungswege für Nachrichten entstehen, so kann eine nebenläufige Nachrichtenverarbeitung erreicht werden. Dabei soll erst dann von einer nebenläufigen Verarbeitung gesprochen werden, wenn mindestens zwei getrennte, vollständige Verarbeitungswege mit Nachrichtenidentifikation, Routing und Transformation für die Nachrichten existieren. Ist nur ein Verarbeitungsweg für alle Nachrichten vorgesehen, so liegt eine *serielle* Nachrichtenverarbeitung vor. Analog zum Kriterium der Geteiltheit werden hier drei Ausprägungen der Nebenläufigkeit unterschieden:

1. **seriell:**

Für alle Nachrichten existiert nur ein Verarbeitungsweg. Somit können die Nachrichten nur nacheinander abgearbeitet werden. Der Verarbeitungsweg selbst kann sich dabei aus mehreren Prozessen zusammensetzen, die hintereinander geschaltet sind. Bei einer solchen Architektur müssen Maßnahmen getroffen werden, damit sehr große Nachrichten die anderen Kommunikationsprozesse nicht zu lange blockieren. Bei einem sehr großen Nachrichtenaufkommen, kann sich der Verarbeitungsprozeß als Flaschenhals auswirken und Nachrichtenstaus entstehen lassen.

2. **lokal-nebenläufig:**

Existieren für die Nachrichten unterschiedliche, vollständige Verarbeitungswege, müssen alle an diesen Wegen beteiligten Prozesse auf einem Rechner lokalisiert sein, so soll von einer *lokal-nebenläufigen* Nachrichtenverarbeitung gesprochen werden. Zwar können Kommunikationsverbindungen mit einem sehr hohen Nachrichtenaufkommen auf mehrere Verarbeitungswege verteilt werden. Da alle Verarbeitungswege aber auf einem Rechner lokalisiert sein müssen, kann nur eine quasi-parallele Verarbeitung der Nachrichten stattfinden.

3. **verteilt-nebenläufig:**

Lassen sich nun die nebenläufigen Verarbeitungswege auch auf mehrere Rechner verteilen, so soll von einer *verteilt-nebenläufigen* Nachrichtenverarbeitung gesprochen werden. Mit einer solchen Architektur läßt sich eine hohe Verfügbarkeit einzelner Kommunikationsverbindungen erreichen, wenn diese über eigene, möglicherweise alternative Verarbeitungswege geführt werden können.

Auch für diese Dimension lassen sich drei Grade für die Taxonomie ableiten:



2.1.3 Heterogenität der Module

Liegt eine verteilte Architektur eines Kommunikationsservers vor, so läßt sich unterscheiden, ob die Module oder Programme des Kommunikationsservers auch auf unterschiedlichen Plattformen installiert sein können oder ob sie auf eine Plattform beschränkt sind. So spricht man von einem *homogenen* Kommunikationsserver, wenn alle Module auf der gleichen Plattform installiert sein müssen. Ein *heterogener* Kommunikationsserver liegt dann vor, wenn sich die Module auch auf verschiedenen Plattformen befinden können. Entsprechend lassen sich zwei Grade für die Dimension *Heterogenität* der Taxonomie unterscheiden:

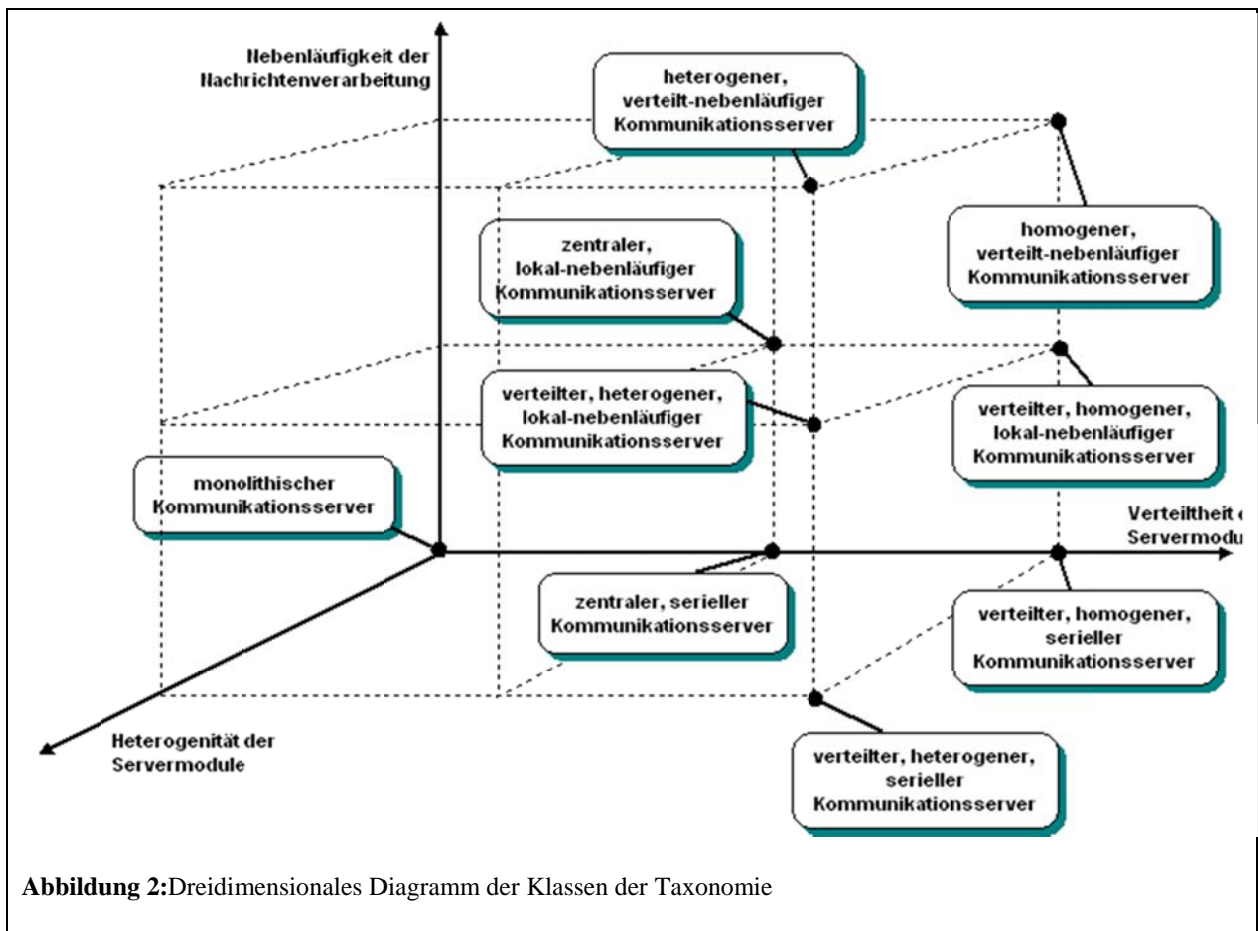


2.2 Klassen der Taxonomie

Die Klassen der Taxonomie werden nun durch die Zusammensetzung der drei Dimensionen *Verteiltheit*, *Nebenläufigkeit* und *Heterogenität* gewonnen. Sie werden in **Fehler! Verweisquelle konnte nicht gefunden werden.** dargestellt.

Dabei lassen sich in Bezug auf die Verteiltheit drei Ebenen unterscheiden (entspricht der horizontalen Achse in **Fehler! Verweisquelle konnte nicht gefunden werden.**):

1. In der ersten Ebene besitzen alle Dimensionen den niedrigsten Grad (ungeteilt, seriell, homogen). Es liegt ein **monolithischer Kommunikationsserver** vor, bei dem die Funktionen der Serverkomponente von nur einem Prozeß ausgeführt werden.
2. In der zweiten Ebene ist eine Aufteilung des Servermoduls auf verschiedene Prozesse möglich, die Prozesse müssen aber auf einem Rechner lokalisiert sein. Diese Kommunikationsserver sind natürlicherweise homogen. Es lassen sich zwei verschiedene Klassen unterscheiden:
 - a) **zentraler, serieller Kommunikationsserver**
Bei dieser Klasse besitzt die Dimension der Nebenläufigkeit den Grad 0 (seriell). Für alle Nachrichten wird nur ein Verarbeitungsweg bereitgestellt, der aber aus mehreren Prozessen bestehen kann.
 - b) **zentraler, nebenläufiger Kommunikationsserver**
Hier kann die Nachrichtenverarbeitung lokal-nebenläufig durchgeführt werden. Da aber alle Prozesse des Servermoduls auf einem Rechner lokalisiert sein müssen, handelt es sich um einen zentralen Kommunikationsserver.



3. In der dritten Ebene schließlich sind alle sechs möglichen Kombinationen der Grade der drei Dimensionen für Kommunikationsserver denkbar.

3 Anforderungskatalog

Aufbauend auf die im Abschnitt 1.2 dargestellten Komponenten eines Kommunikationsservers wurde in [6] ein umfassender Anforderungskatalog erstellt. Dieser Katalog besteht aus insgesamt 51 verschiedenen Anforderungen an die einzelnen Komponenten eines Kommunikationsservers. Neben den grundsätzlichen Anforderungen werden auch weitergehende Anforderungen beschrieben, die sich aus der zentralen Stellung des Kommunikationsservers im KKS ergeben. Die Anforderungen sind möglichst allgemein gehalten, da sie eine spezielle software-technische Realisierung nicht vorschreiben sollen.

Der Anforderungskatalog wurde für eine Analyse und einen Vergleich verschiedener Kommunikationsserver (siehe auch Abschnitt 4) um einen Fragenkatalog erweitert. Dazu wurden zu jeder Anforderung eine oder mehrere Fragen formuliert, die die Umsetzung der Anforderung in einem speziellen Kommunikationsserver betreffen. Aufgrund des Umfangs des Anforderungskataloges und des Fragebogens kann hier eine Wiedergabe nicht erfolgen.

4 Vergleich von Kommunikationsservern

Der Anforderungskatalog und die Taxonomie bieten die Basis für eine umfassende Analyse von Kommunikationsservern. In [6] wurden auf dieser Basis die Architektur, der Nachrichtenfluß sowie der Funktionsumfang verschiedener kommerziell verfügbarer Kommunikationsserver analysiert. Anschließend erfolgte eine Einordnung in die Taxonomie.

Analysiert wurden die folgenden Kommunikationsserver:

- *Cloverleaf* von HCI
- *DataGate* von STC
- *HCS (Hospital-Communication-System)* von SoftCon
- *HL7-Connection-Server* von prompt!
- *Pro7-System* von ICD e.V., ProDV und ROKD

Bei der Analyse der Kommunikationsserver wurden drei verschiedene Wege beschritten:

1. Von zwei der betrachteten Kommunikationsservern (*Cloverleaf* und *DataGate*) stand eine Installation zur Verfügung. Diese Kommunikationsserver konnten daher sehr ausführlich analysiert werden.
2. Ein Kommunikationsserver (*Pro7-System*) wurde Vorort beim Hersteller anhand eines Fragenkataloges analysiert.
3. An die Hersteller der übrigen Kommunikationsserver (*HL7-Connection-Server* und *HCS*) wurde der Fragenkatalog mit der Bitte um eine möglichst ausführliche Beantwortung versandt.

Der in der zweiten und dritten Methode verwendete Fragenkatalog setzte sich aus den Anforderungen des Anforderungskataloges zusammen. Jede Anforderung wurde dabei um eine Reihe von Fragen erweitert, die die genaue Umsetzung der Anforderung im Kommunikationsserver betreffen. Für eine Darstellung des Fragebogens sei auf [6] verwiesen. Dort werden auch die Ergebnisse der Analyse der einzelnen Kommunikationsserver ausführlich dargestellt.

Die Einordnung der Kommunikationsserver in die oben beschriebene Taxonomie stellt sich folgendermaßen dar (in alphabetischer Reihenfolge):

Cloverleaf:

Die Serverkomponente von *Cloverleaf* besteht aus einem oder mehreren Serverprozessen, die aber alle auf einem Rechner lokalisiert sein müssen. Jeder Serverprozeß stellt einen vollständigen Verarbeitungsweg für Nachrichten bereit, es handelt sich somit um einen **zentralen, lokal-nebenläufigen Kommunikationsserver**.

DataGate:

Hier lassen sich die Serverkomponente zwar auf mehrere Prozesse verteilen, durch den zentralen Serverprozeß existiert aber nur ein Verarbeitungsweg für alle Nachrichten. Daher handelt es sich hier um einen **zentralen, seriellen Kommunikationsserver**.

HCS:

Dieser Kommunikationsserver ist in die Klasse der **heterogenen, verteilt-nebenläufigen Kommunikationsserver** einzuordnen. Alle Module lassen sich auf die unterschiedlichsten Rechner verteilen. Im Prinzip läßt sich für jedes Subsystem auch ein eigener Nachrichtenverarbeitungsweg installieren. Eine parallele Nachrichtenverarbeitung ist somit möglich.

HL7-Connection-Server:

Bei diesem Kommunikationsserver handelt es sich wie bei Cloverleaf um einen **zentralen, lokal-nebenläufigen Kommunikationsserver**. Für jede Nachricht existiert ein eigener Verarbeitungsweg, die Serverprozesse müssen aber auf einem Rechner laufen.

Pro7-System:

Die Module des Pro7-Systems lassen sich auf verschiedene Rechner verteilen. Aufgrund der zentralen Stellung des Verarbeitungsmoduls ist aber nur eine serielle Verarbeitung der Nachrichten möglich. Daher handelt es sich hier um einen **verteilten, heterogenen, seriellen Kommunikationsserver**.

5 Ergebnisse

Bei dem Vergleich der Kommunikationsserver ist seit den Untersuchungen von Heitmann eine deutliche Marktkonzentrierung zu beobachten. Von denen in [3] untersuchten sechs verschiedenen Kommunikationsservern, sind heute nur noch vier auf dem Markt vertreten. Die Hersteller der übrigen beiden Produkte bieten nun Kommunikationsserver anderer Hersteller an. Zwar ist mit dem Pro7-System ein weiterer Kommunikationsserver hinzugekommen, doch bleibt abzuwarten, inwieweit er sich auf dem Markt durchsetzen wird. Zusammen mit dem Kommunikationsserver HCS ist das Pro7-System technisch als sehr fortschrittlich anzusehen. Allerdings existieren für beide Systeme bisher nur sehr wenige Routineinstallationen, so daß ein echter Praxistest noch nicht vorliegt. Als Marktführer sind zur Zeit die beiden aus den USA stammenden Kommunikationsserver Cloverleaf und DataGate anzusehen. Sie stellen wohl auch die am weitesten ausgereiften Produkte dar.

Bei der Auswahlentscheidung für einen bestimmten Kommunikationsserver kann keine generelle Empfehlung für eine bestimmte Kommunikationsserver-Architektur gegeben werden. Aufgrund einer umfassenden Analyse der jeweils vorliegenden Gesamtarchitektur des eigenen KKS, der Art und Anzahl der über den Kommunikationsserver abzuwickelnden Kommunikationsverbindungen, sowie den Anforderungen des lokalen Administrators an Konfigurations-, Monitor- und Managementfunktionen ergibt sich in der Regel ein für ein Krankenhaus spezielles Anforderungsprofil. Ein Abgleich dieses Profils mit der in dieser Arbeit vorgestellten Taxonomie kann für die jeweilige Auswahlentscheidung eine wichtige Hilfestellung sein.

6 Ausblick

Kommunikationsserver lassen sich im Krankenhaus nicht nur als zentrale Nachrichtenvermittlungsstelle eines KKS einsetzen. Neben einer Vernetzung mehrerer Krankenhäuser über einen Kommunikationsserver ist auch in großen Abteilungen eines Krankenhauses ein Einsatz von Kommunikationsservern denkbar. Diese „lokalen“ Kommunikationsserver steuern dann den Nachrichtenaustausch zwischen den Systemen in einer Abteilung. Über eine Kommuni-

kationsverbindung zum zentralen Kommunikationsserver des Krankenhauses besteht ein Anschluß an das KKS. So können zum Beispiel in einer Intensivstation eines Krankenhauses die verschiedenen Überwachungs- und Behandlungsgeräte über den Kommunikationsserver mit einem zentralen Arbeitsplatz verbunden werden, von dem aus alle Patienten überwacht werden können.

Auch innerhalb einzelner Geräte läßt sich ein - nur auf die notwendigen Komponenten reduzierter - Kommunikationsserver einsetzen. Auf diese Weise wird es möglich, die Schnittstelle des Gerätes ausgesprochen flexibel zu halten.

Kommunikationsserver ermöglichen es den einzelnen Teilsystemen, Nachrichten auszutauschen. Ein typisches Szenario stellt die Aufnahme eines Patienten in das Informationssystem der Verwaltung dar. Dieses System sendet die neuen Stammdaten an andere Informationssysteme, die diese Daten in ihren lokalen Datenbanken speichern. Das eigentliche Ziel ist also eine Replikation der Daten, die aber über das Senden und Empfangen von Nachrichten realisiert werden muß. Insbesondere die spätere Konsistenzsicherung der Replikat wird nicht direkt durch Kommunikationsserver unterstützt. Um auch auf Integrationsebene (also in der Kopplungssoftware) eine Unterstützung für eine konsistente Replikation bieten zu können, wurde basierend auf den Erfahrungen mit Kommunikationsservern ein Ansatz zur Integration von replizierten Informationen auf Basis der Datenbankschemata der einzelnen Informationssysteme entwickelt. Dieser Ansatz und eine entsprechende Implementierung werden in [2] beschrieben. Bei dieser schema-basierten Integration ist auf Ebene der Kopplungssoftware bekannt, welche Daten von wo nach wo repliziert werden sollen, so daß der Datenaustausch entsprechend optimiert und die Replikat konsistent gehalten werden können.

7 Literatur

- [1] HAMMOND, W.E., DE MOORE, G.J.E. (Hrsg) et al.: „HL7: A Protocol for the Interchange of Healthcare Data.“. Progress in Standardization in Health Care Informatics. IOS Press, 1993
- [2] HASSELBRING, W.: „Federated integration of replicated information within hospitals.“ International Journal on Digital Libraries, 1(3): 192-208, 1997
- [3] HEITMANN, K.U.: „Kommunikationsserver - Konzepte und Produkte.“ In HAAS, P., KÖHLER, C.O., KUHN, K., PIETRZYK, P., PROKOSCH, H.U. (Hrsg.), Praxis der Informationsverarbeitung im Krankenhaus. ecomed, 1996
- [4] JOSTES, C., PACZKOWSKI, J., SCHRÖDER, J.P.: „Integration im Krankenhaus - Offene Kommunikation mit dem standardisierten Anwendungsprotokoll HL7“. Informatik, Biometrie und Epidemiologie in Medizin und Biologie 26 (1), 47 - 55, 1995
- [5] KRÖBER, A.: „Anforderungsanalyse an einen konfigurierbaren HL7-Kommunikations-Server mittels OMT und ausführbarer Modelle.“. Diplomarbeit, Fachbereich Informatik der Universität Dortmund, 1996
- [6] LANGE, M.: „Eine Taxonomie und ein Anforderungskatalog für Kommunikationsserver im Krankenhaus.“ Diplomarbeit, Fachbereich Informatik der Universität Dortmund, 1997
Verfügbar im Internet: ftp://ls10-www.informatik.uni-dortmund.de/pub/Diplom-Arbeiten/da_Matthias_Lange.ps.gz
- [7] ÖZSU, M.T., VALDURIEZ, P.: „Distributed database systems: where are we now?“ IEEE Computer, 24(8):66-78, 1991
- [8] PROKOSCH, H.U.: „Ein Referenzmodell für den Einsatz wissensbasierter und wissenverarbeitender Funktionen innerhalb eines Krankenhaus-Informationssystems.“ Habilitationsschrift, Fachbereich Humanmedizin der Justus-Liebig-Universität Gießen, 1994

[9] RUMBAUGH, J., BLAHA, M., PREMERLANI, W., EDDY, F., LORENSEN, W.: „Object-Oriented Modeling and Design.“ Prentice Hall Englewood Cliffs, New Jersey 07632, 1991

[10] TANENBAUM, A.S.: „Computer Networks.“ Prentice Hall, 1996

Anschriften der Verfasser: Dipl. Inform. Matthias Lange, Institut für Medizinische Informatik und Biomathematik, Westfälische Wilhelms Universität, Domagkstraße 9, 48129 Münster; Prof. Dr. Hans-Ulrich Prokosch, Institut für Medizinische Informatik und Biomathematik, Westfälische Wilhelms Universität, Domagkstraße 9, 48129 Münster; Dr. Wilhelm Hasselbring, INFOLAB, Tilburg University (KUB), 5000 LE Tilburg, The Netherlands.