

Kopplungsstrategien für Anwendungssysteme im Krankenhaus

Die Autoren

Heiko Niemann
 Wilhelm Hasselbring
 Thomas Wendt
 Alfred Winter
 Matthias Meierhofer

Dipl.-Inf. Heiko Niemann, OFFIS,
 Escherweg 2,
 26121 Oldenburg,
 E-Mail: heiko.niemann@offis.de;
 Prof. Dr. Wilhelm Hasselbring,
 Carl von Ossietzky Universität
 Oldenburg, Fachbereich Informatik,
 Abteilung Software Engineering,
 26111 Oldenburg,
 E-Mail: hasselbring@informatik.uni-
 oldenburg.de;
 Dipl.-Inf. med. Thomas Wendt,
 Prof. Dr. Alfred Winter,
 Universität Leipzig,
 Institut für Medizinische Informatik,
 Statistik und Epidemiologie,
 Liebigstraße 27,
 04103 Leipzig,
 E-Mail:
 {wendt|winter}@imise.uni-leipzig.de;
 Dipl.-Inf. Matthias Meierhofer,
 Meierhofer AG,
 Wamslerstraße 2,
 81829 München,
 E-Mail: m.meierhofer@meierhofer.de

■ 1 Einleitung

Krankenhausinformationssysteme wie das des Universitätsklinikums Leipzig (UKL) bestehen oft aus vielen unterschiedlichen Anwendungssystemen, die auf Produkten verschiedener Hersteller basieren. Für die Integration dieser Anwendungssysteme werden Lösungen auf der Basis unterschiedlicher Integrationstechniken angeboten. Bei der Auswahl von Anwendungssystemen für die medizinische Informationsverarbeitung sind die Vor- und Nachteile der angebotenen Lösungen oft nicht umfassend bekannt.

Im Kooperationsprojekt „EKKIS“ (Enge Kopplung klinischer Informationssysteme [Offi01]) werden in Zusammenarbeit des UKL mit dem OFFIS (Oldenburger Forschungs- und Entwicklungsinstitut für Informatik-Werkzeuge und Systeme) und der Meierhofer AG [Meie02] Kopplungsstrategien für KIS-Komponenten entwickelt. Der Artikel beschreibt Strategien und Techniken für die enge Kopplung zur Integration von autonomen Anwendungssystemen beispielhaft für das UKL. Darüber hinaus werden Lösungsmöglichkeiten für ausgewählte Probleme der Integration beschrieben. Dabei werden konzeptionelle Grundlagen wie Transaktionskonzepte und Replikationsverfahren berücksichtigt.

Bei der Vorstellung von Problemen mit Integrationslösungen wird unterschieden zwischen Problemen, die sich unmittelbar aus den gewählten Kopplungsstrategien und Integrationstechniken ableiten lassen und solchen, die sich aus organisatorischen

Randbedingungen ergeben. Die behandelten Kopplungsstrategien sind die synchrone Datenkommunikation, die asynchrone Datenkommunikation und die zentrale Nummernvergabe. Die behandelten Integrationstechniken sind Remote Function Calls (RFC) des SAP R/3 Systems und der Nachrichtentransfer über einen Kommunikationsserver. Die behandelten Grundlagen für Transaktionskonzepte beinhalten Queued Transactions, Sagas und das 2-Phasen-Commit.

In Abschnitt 2 wird eine Vorgehensweise zur Integration von Anwendungssystemen erläutert. Die einzelnen Schritte und Ergebnisse der Integration anhand dieser Vorgehensweise wird im Abschnitt 3 beschrieben, wobei in Unterabschnitten die Analyse der Geschäftsprozesse vorgenommen, die Integration der Anwendungen betrachtet und die technische Realisierung durch Einsatz einer Middleware vorgestellt wird. Abschließend werden in Abschnitt 4 die Lösungen diskutiert und in Abschnitt 5 ein Ausblick auf weitere Arbeitspunkte gegeben.

■ 2 Integration von Anwendungssystemen

Zur systematischen Behandlung der Integrationsthematik wurden verschiedene Modelle geschaffen. In [Hass00] wird die Systemintegration anhand eines 3-Ebenen-Modells diskutiert, in dem Organisationseinheiten durch drei Architekturebenen beschrieben werden (vgl. Bild 1):

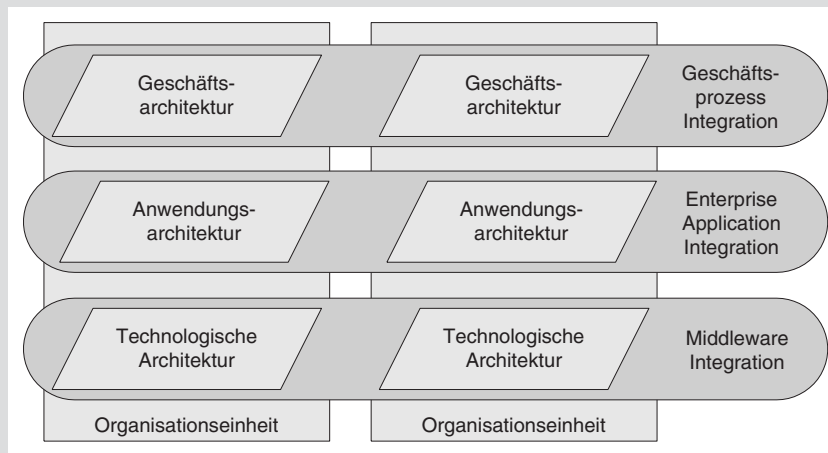


Bild 1 Horizontale Integration zur Unterstützung der Geschäftsprozesse

1. Geschäftsarchitektur: In dieser Ebene werden die Organisationsstruktur und die Arbeitsabläufe für die Geschäftsregeln und -prozesse definiert.
2. Anwendungsarchitektur: Diese Ebene beschreibt die Implementierung des Geschäftskonzepts durch die Unternehmensanwendungen.
3. Technologische Architektur: Die Informations- und Kommunikationsinfrastruktur wird auf dieser Ebene dargestellt.

Eine möglichst effektive Unterstützung der übergreifenden Geschäftsprozesse erfordert nun eine horizontale Integration auf den verschiedenen Ebenen. Die Integration auf der Anwendungsebene ist auch als EAI (Enterprise Application Integration) bekannt. Auf der technologischen Ebene setzt Middleware dabei die Integration technisch um, z. B. durch CORBA Object Request Broker [Corb02].

Es können zwei grundlegende Strategien zur Herstellung von Integration unterschieden werden: bottom-up oder top-down. Bei der Bottom-up-Strategie wird auf der technologischen Ebene begonnen:

1. Installation der technischen Integrationsplattform
2. Verknüpfung der Anwendungssysteme
3. Anpassung der Geschäftsprozesse an die neuen bzw. erweiterten Möglichkeiten

Die Top-down-Strategie beginnt auf der Geschäftsebene:

1. Analyse der Geschäftsprozesse und Erstellung eines Anforderungskataloges
2. Klärung der beteiligten Anwendungssysteme und deren (benötigter) Interaktion
3. Entscheidung für eine Integrationstechnik

In der Praxis wird man sich üblicherweise nicht nur auf eine Strategie beschränken, sondern eine angemessene Kombination anstreben, bei der aber top-down gestartet werden sollte [Hass02]. Für die Integrationsarbeiten im hier behandelten Integrationszenario wurde eine Mischform gewählt, bei der primär top-down vorgegangen wird, jedoch unter Berücksichtigung der am UKL vorhandenen Integrationslösungen.

■ 3 Vorgehen und Ergebnisse des EKKIS-Projektes

3.1 Analyse der Geschäftsprozesse

Das gewählte Top-down-Vorgehen sieht als ersten Schritt eine Analyse der für das Projekt relevanten Geschäftsprozesse vor. Für das hier betrachtete Integrationszenario wurden zwei Geschäftsprozesse identifiziert: Der Prozess der Patientenaufnahme und der Prozess der Diagnosen- und Prozedurendokumentierung für Operationen. Die Bilder 2 und 3 zeigen beide Prozesse als Aktivitätsdiagramme auf der Basis der UML [RuJB99].

Aufnahmen werden am UKL hauptsächlich durch die Mitarbeiter der zentralen Patientenaufnahmestelle durchgeführt. Sie haben im Vergleich zu den Klinikmitarbeitern wesentlich bessere Kenntnis von administrativen Aspekten, z. B. der Einstufung in Behandlungskategorien. Zusätzlich sollen auch in den Kliniken selbst Aufnahmen durchgeführt werden. Diese werden vorab als sogenannte Plan-Aufnahmen erfasst, um Patienten in die OP-Planung einbeziehen zu können, auch wenn sie noch nicht vor Ort im Klinikum sind. Bereits bei der OP-Planung werden einige medizinische Informationen dokumentiert. Deshalb müssen die Plan-Aufnahmen mit den späteren „echten“ Aufnahmen zusammengeführt werden.

Diagnosen und Prozeduren (= medizinische Maßnahmen), die in den Kliniken dokumentiert werden, müssen für die Fallabrechnung in der Kostensicherungsstelle zur Verfügung stehen. Diagnosen und Prozeduren zu einer Operation oder einer Untersuchung werden oft nicht zusammenhängend dokumentiert. Viele OP-Diagnosen können erst mehrere Tage nach der Operation gestellt und dokumentiert werden, z. B. wenn Gewebeproben aufwändig zu untersuchen sind. Für Fallabrechnungen ist die Vollständigkeit der Dokumentation wichtig. Für eine Kommunikation von Diagnosen und Prozeduren muss also im Geschäftsprozess ein Abschluss der Dokumentation vorgesehen werden. Die Entlassung aus dem Krankenhaus reicht dafür nicht aus.

3.2 Anwendungsintegration

Die Probleme, zu deren Lösung das EKKIS-Projekt beitragen soll, ergeben sich daraus, dass die im Patientenmanagement und in den Kliniken verwendeten Anwendungssysteme und die bestehenden Schnittstellen teilweise nicht für die geforderten Geschäftsprozesse geeignet sind.

Wir beschränken uns hier auf die Betrachtung

- des Patientenmanagementsystems auf der Basis des IS-H-Moduls im R/3-System und
- des Operationsplanungs- und -dokumentationssystems auf der Basis des MCC-Systems [Meie02].

3.2.1 Ein Modell des Integrationsszenarios

Bild 4 zeigt ein einfaches Modell der Kopplung von IS-H und MCC. Es erlaubt, die beteiligten Anwendungssysteme im Zusammenhang mit Aufgaben bzw. Aktivitäten der Geschäftsprozessmodellierung und mit genutzten physischen Werkzeugen zu betrachten. Das Bild wurde auf der Basis des Metamodells 3LGM erstellt (3-Level Graph-Based Model², [WiBW01], Nachfolger des 3LGM [WiHa95]). Das 3LGM sieht ebenfalls drei Ebenen vor, macht aber detailliertere Modellierungsvorgaben als in Abschnitt 2 diskutiert. Es unterscheidet

- eine fachliche Ebene mit Aufgaben der Informationsverarbeitung und den interpretierten bzw. bearbeiteten Objekttypen,
- eine logische Werkzeugebene mit Anwendungssystemen und Angaben zu Schnittstellen und Datenbanken,
- eine physische Werkzeugebene mit Angaben zu Geräten, Netzverbindungen usw.

Die Anwendungsebene und die technologische Ebene aus dem Abschnitt 2 sind hier zur logischen Werkzeugebene vereint. Die fachliche Ebene entspricht der Geschäftsmodellierungsebene.

Zwischen IS-H und MCC sind zwei verschiedenartige elektronische Kommunikationsverbindungen realisiert:

1. Über einen Kommunikationsserver werden Daten zu Aufnahmen, Verlegungen und Entlassungen von Patienten (ADT-Daten) von IS-H an MCC asynchron übermittelt. Dazu werden von IS-H ADT-Nachrichten periodisch in einer Datei zur Verfügung gestellt. Der Kommunikationsserver liest die Nachrichtendatei periodisch, löscht die gelesenen Nachrichten und leitet sie in eine andere Nachrichtendatei für MCC weiter. Die Nachrichten werden nicht konvertiert, aber über einen Filter werden einige Nachrichtentypen, z. B. Nachrichten zum Anlegen von Operationen, von der Weiterleitung ausgeschlossen (vgl. Bild 5). Die MCC-seitige Empfangskomponente liest und löscht den Inhalt der Nachrichtendatei periodisch. In Log-Dateien wird festgehalten, welche Nachrichten erfolgreich verarbeitet werden konnten und welche nicht.

2. Über eine RFC-Schnittstelle (Remote Function Call, [Sap02]) werden aus MCC an IS-H OP-Daten, insbesondere Diagnosen und Prozeduren, und Daten zu geplanten Aufnahmen synchron übermittelt. Aufnahmedatensätze werden in MCC erfasst, aber nicht unmittelbar gespeichert. Die Speicherung erfolgt erst nach Übermittlung an IS-H, da von IS-H als Rückgabewerte die Fallnummer zur Aufnahme bereitgestellt wird. Über die Fallnummer können nachträgliche Aufnahmeänderungen und auch OP-Daten direkt dem Behandlungsfall zugeordnet werden. Das Übermitteln von Operationsdatensätzen ist nicht an die Speicherebene geknüpft, sondern an das Abschließen von Operationsdokumentationen. In MCC werden über eine Abschlussfunktion Operationsdokumentationen für eine weitere Bearbeitung gesperrt. Beim Abschließen einer Operation wird diese zunächst per RFC in IS-H gespeichert. IS-H gibt eine Bewegungsnummer zurück, die in der MCC-Datenbank gespeichert wird. Falls die Funktionalität zum Öffnen, Ändern und wiederholten Abschließen einer bereits abgeschlossenen Operation genutzt wird, kann durch die Kenntnis der Bewegungsnummer auch in IS-H eine Korrektur vorgenommen werden.

Bild 5 zeigt die logische Werkzeugebene aus Bild 4 mit den Kommunikationsverbindungen. Das Patientenmanagementsystem und das OP-Planungs- und Dokumentationssystem sind eingebettet in ein komplexes Informationssystem mit vielen weiteren Anwendungssystemen. Die logische Werkzeugebene in Bild 6 zeigt beide Systeme in einem größeren Ausschnitt dieses Informationssystems. Das Bild zeigt zusätzlich

- den Kommunikationsserver eGate, der Nachrichten filtert, transformiert und weiterleitet,
- Anwendungssysteme sogenannter Leistungsstellen wie der Radiologie (MEDOS), des Zentrallabors (C-LAB), der Pathologie (DC-PATHOS) und der Endoskopie (PIA),
- ein Spezialesystem zur intensivmedizinischen Überwachung (COPRA) und
- ein System zur Erfassung und Auswertung innerbetrieblicher Leistungsdaten (LEDIS).

Da die verschiedenen Systeme gemeinsame Daten, wie Stamm- oder Dokumentationsdaten, eines Patienten verarbeiten, wird eine Kopplung dieser Systeme benötigt, um z. B. Doppelerfassung und Inkonsistenz zu vermeiden. Auch der Zeitfaktor spielt eine gewichtige Rolle. So birgt der Austausch von Daten per Beleg, per Diskette oder

Kernpunkte für das Management

Krankenhausinformationssysteme bestehen in ihrem rechnerunterstützten Teil auf Grund unterschiedlichster Anforderungen der verschiedenen Bereiche im Krankenhaus aus heterogenen und autonomen Anwendungssystemen. Eine Kopplung dieser Systeme bedeutet im Allgemeinen eine Replikation von Patientendaten.

- Die lose Kopplung über einen Kommunikationsserver ist Stand der Technik in klinischen Informationssystemen.
- Eine engere Kopplung der Systeme wird durch eine synchrone Kommunikation mit Transaktionskontrolle erreicht, um eine höhere Konsistenz zu erreichen.
- Die Anforderungen an eine Replikationsstrategie im Krankenhaus bez. Konsistenz, Verfügbarkeit und Performanz können nur über eine Kombination verschiedener Verfahren erreicht werden.
- Die Systemintegration sollte beginnend auf Ebene der Geschäftsprozesse über die Anwendungslogik hin zur technischen Middleware idealerweise top-down erfolgen.

Stichworte: Krankenhausinformationssystem (KIS), synchrone/asynchrone Kommunikation, Kopplungsstrategie, Kommunikationsserver, Applikationsintegration, Schemaintegration, Datenreplikation

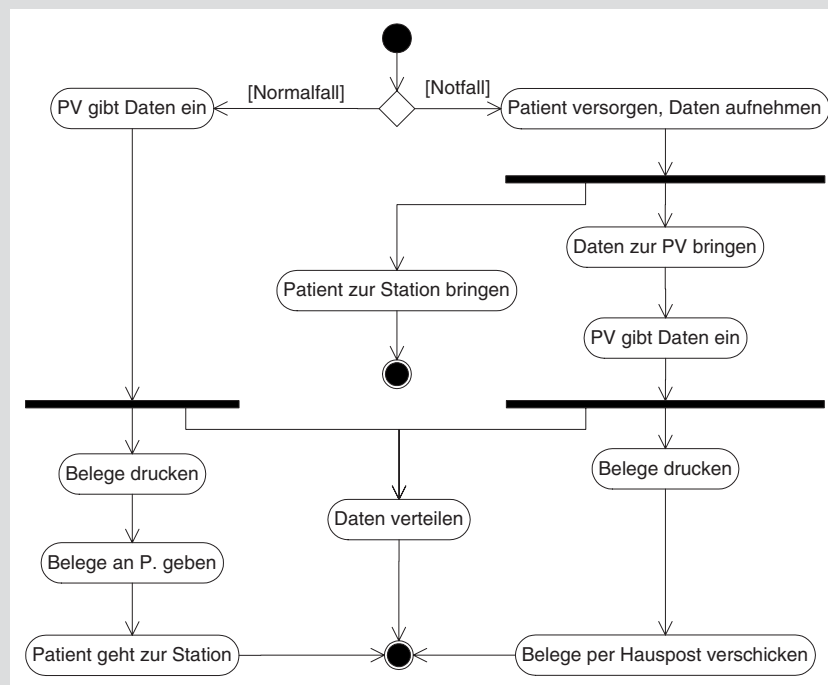


Bild 2 Aufnahme eines Patienten im UKL

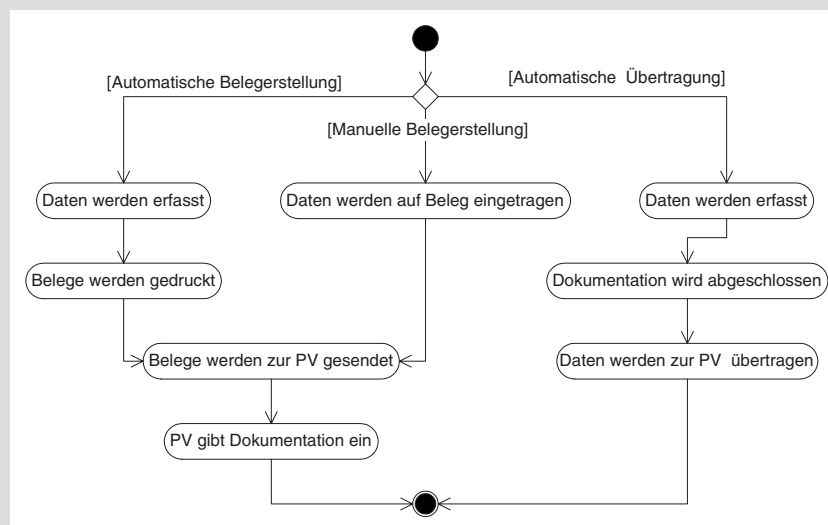


Bild 3 Dokumentation von Diagnosen, Prozeduren und OPs

selbst über einen langsamen Kommunikationskanal die Gefahr einer verspäteten Aktualisierung.

Wesentlich ist der Aspekt der zentralen Nummernvergabe: Um die medizinischen und administrativen Patientendaten zuzuordnen und austauschen zu können, werden einheitliche Identifikationsnummern

benötigt. IS-H zeichnet für die Vergabe der Identifikationsnummern verantwortlich. Wenn andere Anwendungssysteme z. B. die Funktionalität Patientenaufnahme durchführen sollen, so wird eine Patienten-ID (bzw. eine Fall-ID) von IS-H benötigt, was eine synchrone Kopplung sinnvoll macht.

In Bild 3 zur Prozessanalyse sind sowohl die älteren Abläufe „Automatische Belegerstellung“ und „Manuelle Belegerstellung“ als auch der neue Prozess „Automatische Übertragung“ zu sehen. Die automatische Übertragung wird zurzeit eingeführt und vom EKKIS-Projekt begleitet.

3.2.2 Probleme auf der Anwendungsebene

Mit der Integration von MCC und IS-H sind u. a. folgende Probleme verbunden:

1. Organisatorische Probleme:
 - Die Fallnummer wird zur Zuordnung von Daten benötigt und ist daher übergreifendes Organisationsmittel. Sie darf nicht von mehreren unabhängigen Stellen bzw. Anwendungssystemen vergeben werden.
 - Die elektronische Übermittlung erfolgt zunächst unsichtbar für die Anwender. Wenn OP-Daten an IS-H übermittelt wurden, gehen die IS-H-Anwender davon aus, dass die abrechnungsrelevanten Daten vorliegen und weiterverarbeitet werden können. Für nachträgliche Änderungen über die RFC-Schnittstelle, die wiederum Änderungen an Rechnungen nach sich ziehen, wird also ein Benachrichtigungsmechanismus benötigt. Nur dann können unter Umständen Änderungsmeldungen an die Kostenträger, i. d. R. die Krankenkassen, geschickt werden.
2. Technische Probleme:
 - Eine sehr enge Kopplung von Anwendungssystemen hat zur Folge, dass der Ausfall eines der beteiligten Anwendungssysteme unmittelbar die Arbeit mit dem Partnersystem behindert. Ein IS-H-Ausfall könnte also die Durchführung von Plan-Aufnahmen mit MCC unmöglich machen.
 - Die asynchrone Nachrichtenübermittlung über den Kommunikationsserver ist mit einem Zeitversatz (zurzeit etwa sechs Minuten) verbunden. Wenn Aufnahmedaten aus IS-H sehr schnell benötigt werden, z. B. bei der OP-Planung, ist diese Zeit zu lang.
 - Bei einer starken Belastung des SAP-Systems, z. B. wegen laufender Auswertungen des Controlling, ist das MCC-System bei Freigabe einer Operation über die RFC-Schnittstelle längere Zeit (etwa 30–40 Sekunden) blockiert.

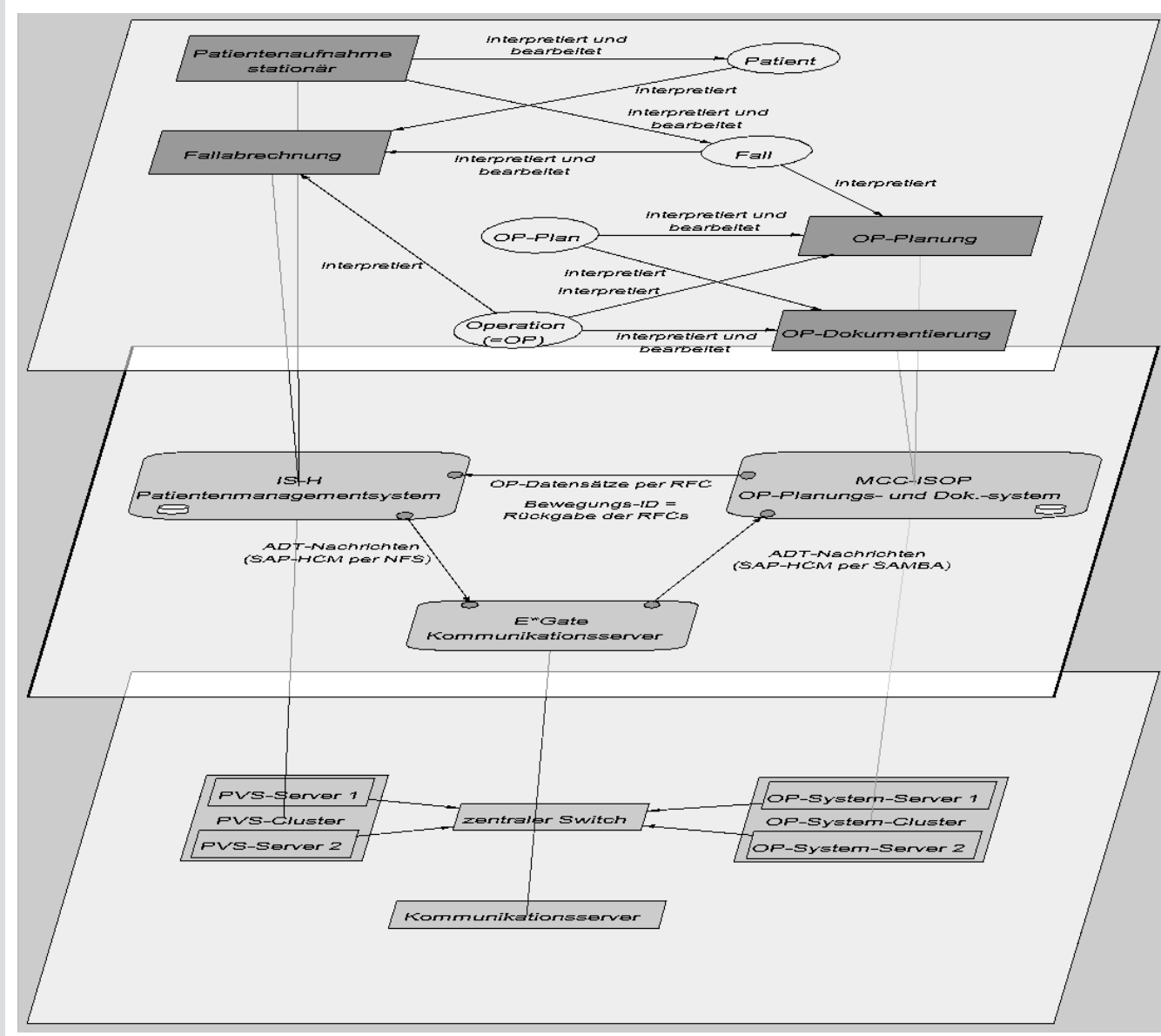


Bild 4 Einfaches Drei-Ebenen-Modell der Kopplung von IS-H und MCC

3.2.3 Schemaintegration

Im betrachteten Szenario bedeutet Anwendungsintegration im Wesentlichen ein Replizieren von Daten. In diesem Abschnitt soll zunächst die Abbildung der Daten der beteiligten Systeme, d. h. die Schemaintegration untersucht werden. Anschließend werden Replikationsstrategien und damit einhergehende Transaktionskonzepte diskutiert.

In Kliniken ist die Kopplung der Informationssysteme über einen Kommunikationsserver Stand der Technik, so auch im UKL (vgl. Bild 6). Der asynchrone Austausch der Daten erfolgt hier über speziell für Kliniken entwickelte Systeme, die neben der Funktionalität eines Kommunikationsservers (siehe z. B. [LaPH99]) insbesondere die Transformation auf das Standardformat für Krankenhäuser HL7 (Health Level Seven) bzw. der verschiedenen Versionen von HL7 bietet. HL7 ist ein Kommunikationsstandard, der von der HL7-Organisa-

tion [Heal01] definiert wurde. Der Entwicklung des HL7-Standards wird ein Reference Information Model (RIM) zugrunde gelegt, in dem die meisten der im Gesundheitswesen zu kommunizierenden Informationen als Klassen und Klassenbeziehungen modelliert sind.

Auch bei der Schemaintegration gibt es grundsätzlich zwei Strategien, die Integration durchzuführen (vgl. [Hass02]): Bei der Bottom-up-Strategie wird ausgehend von den lokalen Datenmodellen ein globa-

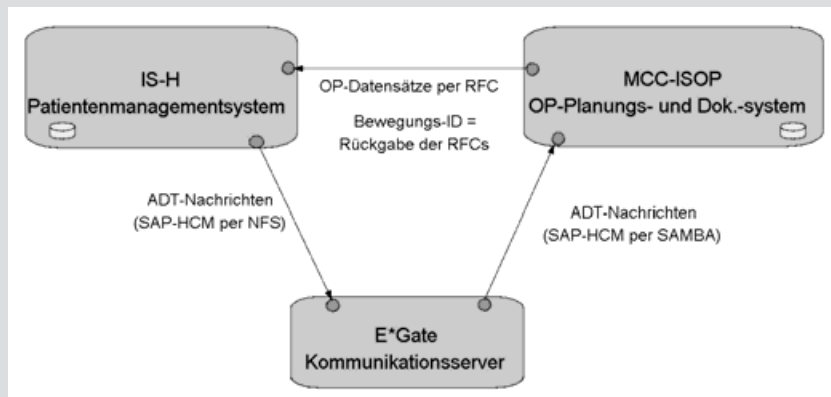


Bild 5 Kommunikationsverbindung zur Übermittlung von ADT-Daten aus IS-H an MCC und von Operationsdaten aus MCC an IS-H (Übersicht)

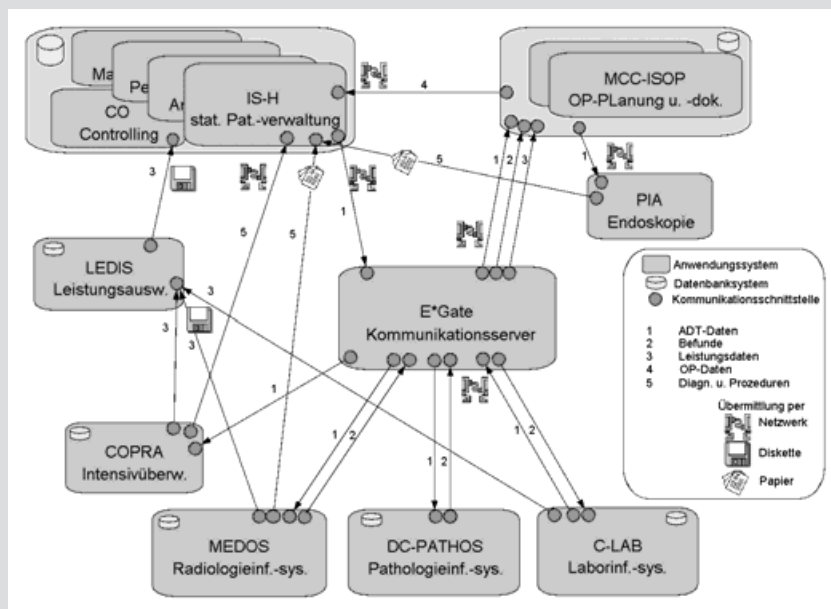


Bild 6 Ausschnitt aus der elektronischen ADT- und Leistungsdatenkommunikation am UKL

les Datenmodell entwickelt, während bei der Top-down-Strategie ausgehend von einem globalen Datenmodell, am besten einem standardisierten Referenzmodell, eine Abbildung zu den lokalen Datenmodellen erstellt wird. Das Top-down-Verfahren bietet im Allgemeinen den Vorteil einer leichteren Erweiterbarkeit auf mehrere Systeme. Außerdem ist bei Vorhandensein eines geeigneten Referenzmodells die Abbildung einfacher zu erstellen, insbesondere dann, wenn die Systemhersteller neue Versionen ihrer Systeme an diesen Standard anpassen.

Da im medizinischen Bereich mit HL7-RIM ein Referenzmodell vorliegt, soll die Abbildung über dieses Standardmodell durchgeführt werden. Wie in Bild 7 gezeigt, wird hierfür eine Abbildung von HL7-RIM auf IS-H bzw. MCC gebildet. Die in Bild 7 dargestellten Datenmodelle stellen nur einen groben Ausschnitt der tatsächlichen Datenmodelle dar. Sind die Abbildungen von dem Referenzmodell zu den lokalen Modellen gefunden, so ergeben sich indirekt auch die Abbildungen zwischen den lokalen Datenmodellen. Bei einer Erweiterung um ein neues lokales Daten-

modell wird dann nur die Abbildung zum Referenzmodell benötigt und die Abbildung zu den anderen lokalen Modellen ergibt sich automatisch. Auf Details der jeweiligen Datenmodelle und der Abbildungen können wir hier aus Platzgründen nicht eingehen.

3.2.4 Replikationsstrategien

Bei der Datenreplikation bestehen Zielkonflikte bezüglich der Gewährleistung von: Konsistenz, Verfügbarkeit und Performanz (vgl. Bild 8). Eine Verbesserung bei einem dieser Kriterien zieht im Allgemeinen eine Verschlechterung bei den anderen Kriterien nach sich. Beispielsweise bedeutet eine Erhöhung der Konsistenzanforderung eine verminderte Verfügbarkeit der Systeme, da bei jeder Schreiboperation alle beteiligten Systeme verfügbar sein müssen. Im UKL ist einerseits Konsistenz gefordert, andererseits darf aber die Autonomie der verschiedenen Anwendungssysteme nicht eingeschränkt werden: Bei Absturz eines Systems müssen die anderen autonom weiterarbeiten können.

Somit unterscheiden wir analog zu den Kommunikationsverfahren zwischen synchronen und asynchronen Replikationsverfahren, d. h. im synchronen Fall werden alle Kopien zeitgleich aktualisiert, im asynchronen Fall wird die Propagierung zeitversetzt durchgeführt. Für beide Varianten sind in der Literatur verschiedenste Verfahren zu finden, z. B. in [Dada96]. In der Tabelle 1 sind die Speicherkonzepte hinsichtlich der oben genannten Kriterien bewertet. Zusätzlich werden als Grenzfälle Unikate, d. h., im Gesamtsystem tritt jedes Objekt nur einmalig auf, und Duplikate, d. h., es werden Kopien erstellt, die aber im Gegensatz zu Replikaten nicht automatisch abgeglichen werden, aufgeführt.

Aus Sicht der Kopplung im UKL kommt für die synchrone Replikation das Primary-Copy-Verfahren und für die asynchrone Replikation das Peer-To-Peer-Verfahren in Frage. Da im UKL das administrative System IS-H das zentrale System darstellt, bietet es sich als das System an, das beim Primary-Copy-Verfahren die Primärkopie hält. Somit erfolgen Datenmanipulationen im ersten Schritt der Primary-Copy-Replikation in IS-H, die Propagierung als zweiter Schritt dieses Verfahrens kann dann asynchron über den Kommunikationsserver erfolgen. Da bei klinischen Systemen Einfüge-Operationen überwiegen

[Hass97] und somit Änderungskonflikte eher selten sind, sollen gegenüber dem Standard des Primary-Copy-Verfahrens folgende Erweiterungen im UKL zulässig sein:

1. Inkonsistentes Lesen wird toleriert, d. h., ein lokales System darf die lokalen Daten lesen und muss nicht beim Primärsystem die Aktualität prüfen.
2. Das initiierende System ändert sowohl auf der Primärkopie als auch auf dem lokalen Datenbestand. Damit muss auf diesem System nicht auf die Aktualisierung über den Kommunikationsserver gewartet werden, sondern die Verarbeitung kann sofort fortgesetzt werden, d. h. die Performanz bleibt weitgehend erhalten.
3. Ist das Primärsystem, also IS-H, nicht verfügbar, so wird trotzdem lokal geändert und mit der Verarbeitung fortgesetzt. Die Änderung am Primärsystem muss später nachgeholt werden.

Der dritte Punkt dieser Erweiterungen bedeutet einen dynamischen Wechsel zur asynchronen Replikation, d. h., die Kopien werden nicht zeitgleich aktualisiert. Auf Grund der geforderten Autonomie der klinischen Systeme muss eine Änderung an jedem beteiligten System möglich sein, was z. B. beim Peer-To-Peer-Verfahren gegeben ist. Hier darf jedes Replikat geändert werden, die Propagierung erfolgt asynchron und Konflikte werden in Kauf genommen. Die durch dieses Verfahren gegebenenfalls entstehenden Konflikte müssen bei der Zusammenführung behandelt werden.

3.2.5 Transaktionskonzepte

Replikation von Daten bedeutet Manipulation in verteilten Systemen. Um eine ordnungsgemäße Verarbeitung in verteilten

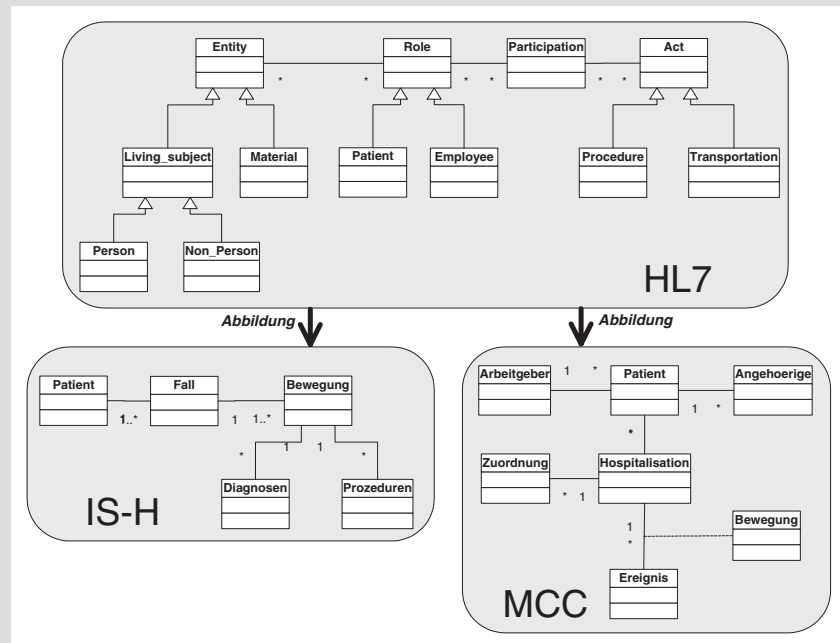


Bild 7 Top-down-Integration von HL7-RIM auf IS-H bzw. MCC (grober Ausschnitt der jeweiligen Datenmodelle)

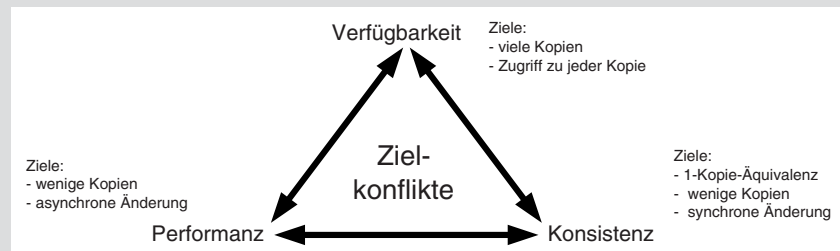


Bild 8 Zielkonflikte der Datenreplikation

Tabelle 1 Bewertung der Speicherkonzepte bezüglich der Zielkonflikte bei Replikation					
	Konsistenz	Verfügbarkeit		Performanz	
		Lesezugriff	Schreibzugriff	Lesezugriff	Schreibzugriff
Duplikate	Gering	Hoch	Hoch	Hoch	Hoch
Asynchrone Replikation	reduziert, wird automatisch hergestellt	Hoch	Hoch, geringe Einschränkung	Hoch	Hoch, geringe Einschränkung
Synchrone Replikation	Hoch, aber Probleme bei Knotenausfall	Hoch	Reduziert	Hoch	Reduziert
Unikate	Hoch	Reduziert, abhängig von der Lage der Unikate	Reduziert, abhängig von der Lage der Unikate	Reduziert, abhängig von der Lage der Unikate	Reduziert, abhängig von der Lage der Unikate

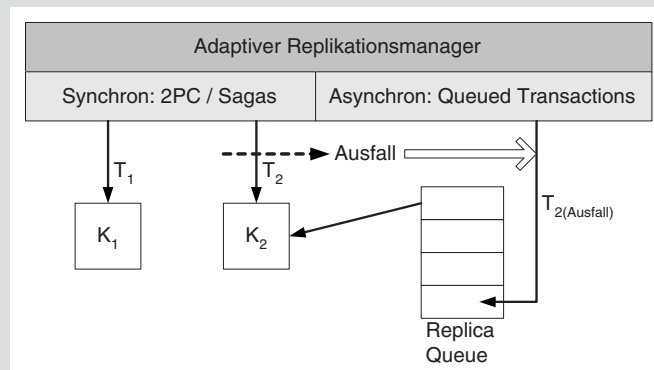


Bild 9 Adaptiver Replikationsmanager als Kombination von 2PC/Sagas und Queued Transactions

Systemen zu gewährleisten, werden Transaktionskonzepte benötigt (vgl. [GrRe93]). Je nach eingesetztem Replikationsverfahren können nun verschiedene Transaktionssysteme zum Einsatz kommen. Im Folgenden soll kurz skizziert werden, wie die im vorhergehenden Unterabschnitt diskutierten Replikationsstrategien transaktionsorientiert umgesetzt werden können.

Das Standardverfahren für die Replikation im UKL sollte das Primary-Copy-Verfahren sein, wobei hier dahingehend erweitert wird, dass zusätzlich das lokale System ebenfalls zeitgleich die Änderung durchführt. Somit wird an mehreren verteilten Systemen eine Aktion durchgeführt, die durch ein Transaktionssystem gesichert werden muss. Das Zwei-Phasen-Commit-Protokoll (2PC-Protokoll) als Transaktionskonzept garantiert konsistenzwahrende Änderung im verteilten Fall. Dieses Konzept wird von vielen Datenbanklieferanten in Form des XA-Protokolls umgesetzt. Das Anwendungssystem SAP R/3 setzt zwar im Allgemeinen auf XA-fähige Datenbanken auf, gibt diese Funktionalität aber nicht als Schnittstelle weiter. Somit muss an dieser Stelle ein erweitertes Transaktionskonzept eingesetzt werden, das offenen geschachtelte Transaktionen zulässt. Da in der ersten Aktion auf IS-H geändert wird, danach auf dem lokalen System, kann das Konzept der verketteten Transaktionen (sogenannte Sagas [GaSa87]) dieses Problem lösen. Auf Kompensationstransaktionen, die bei Sagas hinsichtlich des Backward Recovery von Interesse sind, soll hier verzichtet werden, sondern gegebenenfalls wird die Transaktion neu aufgesetzt, was dem Forward Recovery entspricht.

Für die asynchrone Replikation kann das Konzept der „Queued Transactions“ [BeNe97] verwendet werden. Transaktionsanforderungen werden hier in eine Warteschlange gestellt, die dann von einem Server abgeholt und bearbeitet werden. Anschließend wird ein Ergebnis wiederum in eine Warteschlange gestellt, von wo das initiiierende System die Antwort entgegen nimmt. Diese Funktionalität wird im Wesentlichen beim Peer-To-Peer-Verfahren benötigt, wobei gegebenenfalls auf die Antwort verzichtet werden kann. Die Realisierung kann z. B. über den Kommunikationsserver geschehen.

In Bild 9 ist ein Replikationsmanager dargestellt, der eine adaptive Replikationsstrategie durch Kombination der Transaktionskonzepte 2PC, Sagas und Queued Transactions umsetzt. Dieser globale Manager kann zur Realisierung der Replikationsziele am UKL beitragen: Im Normalfall werden die Daten der beteiligten Systeme über Sagas geändert. Steht ein System wegen Ausfall nicht zur Verfügung, wird über die Queued Transactions zur asynchronen Replikation gewechselt. Am UKL ist IS-H führendes Anwendungssystem für die Nummernvergabe. Daher muss bei Nichtverfügbarkeit von IS-H die Nummernvergabe anders geregelt werden, z. B. kann in IS-H ein Nummernbereich gesperrt werden und dieser Bereich wird dann MCC zur Verfügung gestellt. Der dynamische Wechsel von synchroner zu asynchroner Replikation kann auch dann erfolgen, wenn z. B. die Performanz des Systems IS-H nicht mehr akzeptabel ist.

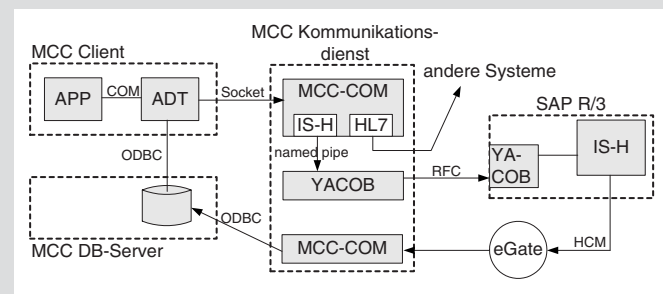


Bild 10 Architektur der engen Kopplung MCC → IS-H

3.3 Middleware Integration

Eine grobe Klassifizierung von Middleware ist in [RuMB00] zu finden, wo zwischen Präsentationsintegration, Datenbankintegration und Applikationsintegration unterschieden wird:

- Die Präsentationsintegration ist die einfachste Form der Integration. Hier geht es im Wesentlichen darum, den Benutzern eine einheitliche Oberfläche für den Zugriff auf verschiedene Anwendungen zur Verfügung zu stellen.
- Bei der Datenbankintegration greifen die globalen Anwendungen über eine Middleware direkt auf die Daten bzw. Datenbanksysteme der Anwendungen zu, z. B. über ODBC. Die Komplexität der Integration bei dieser Methode hängt von den Zugriffsmöglichkeiten auf die zugrundeliegenden Datenquellen ab. Im Falle von relationalen Datenbanksystemen ist die Integration z. B. relativ einfach. Ein Nachteil besteht darin, dass bei Änderung des Datenmodells selbst nur einer Altanwendung die Integrationslösung angepasst werden muss. Ein weiterer Nachteil ist, dass die Logik der Altanwendungen nicht berücksichtigt wird bzw. redundant implementiert werden muss und diese Altanwendungen konkurrierend auf die Daten zugreifen.
- Die Applikationsintegration setzt nun direkt auf den Altanwendungen auf. Middleware wie MOM (Message-oriented Middleware) oder DOT (Distributed Object Technology) nutzen dabei die von den Altanwendungen zur Verfügung gestellten Schnittstellen in Form

von Funktionen und Prozeduren (API). Damit ist also auch die Logik der Altanwendungen integriert und die globalen Anwendungen treten nicht in Konkurrenz zu den Altanwendungen, sondern bedienen sich deren Funktionalität. Der Nachteil besteht in der größeren Komplexität und in der Abhängigkeit von den APIs, die die Altanwendungen zur Verfügung stellen.

Da auf die Daten von SAP R/3 nicht direkt zugegriffen werden sollte, sondern ausschließlich über Schnittstellen wie z. B. RFC oder BAPI, handelt es sich im betrachteten Szenario um eine Applikationsintegration. Also kann für die nachrichten-orientierte Integration eine lose Kopplung, z. B. ein Kommunikationsserver, oder für die Objekt- bzw. Komponententechnologie eine enge Kopplung, Produkte der CORBA- bzw. EJB-Spezifikation [Corb02; Java02] verwendet werden. Auch die Microsoft-Komponententechnik COM+ [Plat99] (vormals DCOM bzw. COM) zählt zu den Varianten einer Applikationsintegration.

Technisch gesehen stellt die Kopplung über einen Kommunikationsserver eine eher einfache Integration dar. Sicherlich ist zu beachten, wie die ausgetauschten Nachrichten verarbeitet werden, insbesondere aus der Sicht transaktionsorientierter Verarbeitung. An dieser Stelle sei auf [ScSS00] verwiesen. Im Folgenden soll die enge Kopplung über die Komponententechnologie vertieft werden. Da MCC bereits intern COM+ verwendet und seitens SAP entsprechende Schnittstellen zur Verfügung stehen, bot sich die Microsoft-Technologie als Middleware zur Integration der Systeme MCC und IS-H an. Weiterhin war es möglich, die „fertige“ Komponente YACOB (Yet Another Communication Box [Unio02]) wiederzuverwenden.

In Bild 10 ist die Architektur der engen Kopplung von MCC zu IS-H dargestellt. Wird im MCC-Client die ADT-Komponente angesprochen, so wird per Socket-Verbindung der MCC-Kommunikationsdienst aufgerufen. An diesen Dienst konnte die YACOB-Komponente angebunden werden, die nun ihrerseits über RFC auf ein YACOB-spezifisches Modul in SAP zugreift, das letztendlich IS-H anbindet. Falls für die angesprochene Operation seitens IS-H Identifikationsnummern vergeben werden, so werden diese IDs neben dem Return-Code auf demselben Weg an die ADT-Komponente zurückgeliefert.

Damit kann nun auch vom MCC-Client bei erfolgreicher Durchführung auf IS-H die Operation auf dem lokalen Datenbanksystem durchgeführt werden (vgl. die Anmerkungen zum erweiterten Primary-Copy-Verfahren im Abschnitt 3.2.4).

Die Propagierung einer Änderung in IS-H wird über das SAP-spezifische Format HCM an den Kommunikationsserver eGate gesendet, der diese Nachricht an die entsprechenden Systeme verteilt, also auch an MCC selbst. Falls die Transaktion vorher schon in MCC lokal durchgeführt wurde, weil z. B. von MCC initiiert, so wird die Nachricht verworfen, andernfalls wird die Transaktion auf dem lokalen Datenbanksystem durchgeführt.

Bei einer von MCC angestoßenen Transaktion handelt es sich um eine Saga (vgl. Abschnitt 3.2.4), die in der ersten Teiltransaktion eine Änderung auf IS-H und in der zweiten Teiltransaktion eine Änderung auf MCC durchführt. Falls nun während der ersten Teiltransaktion ein Fehler auftritt, so wird die komplette Transaktion neu aufgesetzt. Wegen der geforderten Autonomie ist denkbar, aber derzeit nicht realisiert, diese Teiltransaktion später nachzuziehen und mit der lokalen Verarbeitung trotzdem fortzusetzen. Ein Fehlschlagen der zweiten Teiltransaktion wird dadurch neu aufgesetzt, dass in diesem Fall über den Kommunikationsserver die Änderung erneut mitgeteilt wird, d. h., es wird ein Forward Recovery gemäß dem Sagas-Konzept durchgeführt, was evtl. eine Konfliktlösung impliziert.

■ 4 Diskussion

In diesem Artikel werden die Integrationsmöglichkeiten von Anwendungssystemen im Universitätsklinikum Leipzig diskutiert, wobei sowohl die eher lose Kopplung über einen Kommunikationsserver als auch die eher enge Kopplung über die Komponententechnologie COM+ dargestellt werden. Es wird eine Vorgehensweise anhand eines Drei-Ebenen-Modells zur Informationssystemintegration vorgeschlagen. Weiterhin werden Konzepte für Replikationsstrategien und Transaktionsverfahren vorgestellt.

Für das UKL haben die Ausführungen der vorhergehenden Abschnitte folgenden Nutzen:

1. Die aktuell zur Kopplung von IS-H und MCC verwendeten Integrationstechniken können besser bewertet werden. Es ist jetzt bekannt, welchen theoretischen Integrationsansätzen sie zugeordnet werden können und welche Vor- und Nachteile diese Ansätze haben.
2. Es ist besser bekannt, welche Probleme nicht mit vorhandenen theoretischen/technischen Lösungsansätzen gelöst werden können, also organisatorisch gelöst werden müssen. Beispielsweise kann das Problem der Benachrichtigung über Änderungen mit den behandelten Ansätzen nicht gelöst werden.
3. Es ist besser bekannt, welche Lösungen für die benannten Probleme angestrebt werden können, z. B. durch Anforderungen bez. der Implementierung von Sagas zur Umsetzung des Primary-Copy-Verfahrens an die Hersteller/Lieferanten der Anwendungssysteme.

Bei der Kopplung über den Kommunikationsserver wird mit asynchronen Replikaten gearbeitet, bei der Kopplung über die RFC-Schnittstelle mit synchronen Replikaten. Die in Abschnitt 3.2.2 genannten Integrationsprobleme können mit den beschriebenen Ansätzen wie folgt gelöst werden:

1. Zu den organisatorischen Problemen:
 - Beim Anlegen von Aufnahmen in MCC kann über die RFC-Schnittstelle die synchrone Replikation der Fallnummer und anderer Falldaten erreicht werden. Mit dem aktuellen Entwicklungsstand der RFC-Schnittstelle ist die Transaktionssicherheit jedoch nicht vollständig gewährleistet.
 - Das Fehlen eines Benachrichtigungsmechanismus über automatische Datenänderungen im Hintergrund beeinflusst die Datensicherheit nicht unmittelbar. Da ein Benachrichtigungsmechanismus aber als Organisationshilfe unverzichtbar ist, wären also zusätzlich zu Transaktionskonzepten auch Benachrichtigungskonzepte umzusetzen, die im EKKIS-Projekt nicht behandelt werden. Theoretisch könnte über das Sperren von Datensätzen eine nachträgliche Änderung verhindert werden. Praktisch ist ein solches Vorgehen nicht akzeptabel.
2. Zu den technischen Problemen:
 - Die Forderung nach zentraler Vergabe der Fallnummer erfordert eine synchrone Replikation, wenn außer-

halb des führenden Anwendungssystems (hier: SAP R/3 IS-H) neue Aufnahmen angelegt werden. Bei Ausfall von IS-H muss das Fehlen der zentralen Nummernvergabe in Kauf genommen werden. Kompromissregelungen, die ein nachträgliches Zusammenführen von Daten auf die korrekte Fallnummer vorsehen, sind meist aufwändig und fehleranfällig.

- Da die OP-Planung in MCC durchgeführt wird, liegt es nahe, Plan-Aufnahmen unmittelbar in MCC durchzuführen, wobei über die RFC-Schnittstelle eine schnelle synchrone Replikation erreicht wird.
- Um bei Performance-Problemen eines Anwendungssystems die anderen möglichst wenig zu behindern, sollten nur diejenigen Daten synchron repliziert werden, bei denen es unbedingt notwendig ist. Bei asynchroner Kommunikation werden unter Umständen keine Identifikationsnummern abgeglichen, sodass nachträgliche Änderungen von Datensätzen schwierig sind. Adaptive Replikationstechniken sollen hier Abhilfe schaffen, befinden sich allerdings noch in der Entwicklung.

5 Ausblick

Generell konnten wir für den Anwendungsbereich Krankenhaus aufzeigen, wie für spezifische Anforderungen geeignete Kopplungsstrategien auf verschiedenen Ebenen der Systemintegration entwickelt werden können. Zusammenfassend kann festgestellt werden, dass im Kontext eines großen Universitätsklinikums keine hundertprozentige Datenkonsistenz erreicht werden kann, um die notwendige Autonomie der Teilsysteme zu wahren. Einen optimalen Mittelweg bieten die vorgestellten adaptiven Replikationsverfahren.

Als zukünftiger Arbeitspunkt kann der Einsatz eines verbesserten Integrationservers untersucht werden. Hier könnten Basissysteme verwendet werden, die der J2EE-Spezifikation [JNBL01] genügen. Eine enge Kopplung kann dann über den Transaktionsdienst JTA, eine lose Kopplung über den Nachrichtendienst JMS realisiert werden. Möglicherweise kann bei Benutzung eines EJB-Servers [Java02] die YACOB-Komponente durch die EJB-Konnektoren-Technologie [ShSN02] er-

setzt werden. Für die Anbindung von MCC erscheint auch die neue Microsoft .NET Technologie [Plat01] sinnvoll.

Literatur

- [BeNe97] *Bernstein, Philip; Newcomer, Eric*: Principles of Transaction Processing. Morgan Kaufmann Publishers, 1997.
- [Corb02] *CORBA*: The OMG's CORBA Website. <http://www.corba.org>, Abruf am 2002-03-01.
- [Dada96] *Dadam, Peter*: Verteilte Datenbanken und Client/Server-Systeme. Springer-Verlag, 1996.
- [GaSa87] *Garcia-Molina, Hector; Salem, Kenneth*: Sagas. In: ACM SIGMOD, San Francisco (1987) S. 249–260.
- [GrRe93] *Gray, Jim; Reuter, Andreas*: Transaction Processing: Concepts and Techniques. Morgan Kaufmann, 1993.
- [Hass97] *Hasselbring, Wilhelm*: Federated integration of replicated information within hospitals. In: International Journal on Digital Libraries 1 (1997) 3, S. 192–208.
- [Hass00] *Hasselbring, Wilhelm*: Information System Integration. In: Communications of the ACM 43 (2000) 6, S. 32–36.
- [Hass02] *Hasselbring, Wilhelm*: Web Data Integration for E-Commerce Applications. In: IEEE Multimedia 9 (2002) 1, S. 16–25.
- [Heal01] *HL7-Organisation*: Health Level Seven, HL7. <http://www.hl7.org>, Abruf am 2001-03-01.
- [Java02] *Java*: Java 2 Platform, Enterprise Edition. <http://java.sun.com/j2ee>, Abruf am 2002-03-20.
- [JNBL01] *Juric, Matjaz; Nagappan, Ramesh; Basha, Jeelani; Leander, Rick; Long, Peter*: Professional J2EE EAI. WROX, 2001.
- [LaPH99] *Lange, Matthias; Prokosch, Hans-Ulrich; Hasselbring, Wilhelm*: Eine Taxonomie für Kommunikationsserver im Krankenhaus. In: Informatik, Biometrie und Epidemiologie in Medizin und Biologie 30 (1999) 1, S. 21–43.
- [Meie02] *Meierhofer AG*: Projekte und Produkte der Meierhofer AG. <http://www.meierhofer.de>, Abruf am 2002-03-15.
- [Offi01] *OFFIS*: EKKIS: Enge Kopplung klinischer Informationssysteme. http://www.offis.de/projekte/ekki/projekt_ekki.htm, Abruf am 2002-03-01.
- [Plat99] *Platt, David*: COM+ verstehen. Microsoft Press, 1999.
- [Plat01] *Platt, David*: Die Microsoft .Net Plattform. Eine Einführung. Microsoft Press, 2001.
- [RuMB00] *Rub, William; Maginnis, Francis; Brown, William*: Enterprise Application Integration: A Wiley Tech Brief. John Wiley & Sons, 2000.
- [RuJB99] *Rumbaugh, James; Jacobson, Ivar; Booch, Grady*: The Unified Modeling Language Reference Manual. Addison-Wesley, 1999.
- [Sap02] *SAP*: SAP Help Portal. <http://help.sap.com>, Abruf am 2002-03-01.
- [ScSS00] *Schuler, Christoph; Schuldt, Heiko; Schek, Hans-Joerg*: Transactional Execution Guarantees for Data-Intensive Process in Medical Information Systems. In: Proceedings of the 1st European Workshop on Computer-based Support for Clinical Guidelines and Portals, Leipzig (2000), S. 89–103.
- [ShSN02] *Sharma, Rahul; Stearns, Beth; Ng, Tony*: J2EE Connector Architecture and Enterprise Application Integration. Addison Wesley, 2002.
- [Unio02] *UNIORG*: External Transaction Services. http://www.uniorg.de/www/pdf/ETS_Handbuch.pdf, Abruf am 2002-03-01.
- [WiBW01] *Winter, Alfred; Brigl, Birgit; Wendt, Thomas*: A UML-based ontology for describing hospital information system architectures. In: Medinfo, Netherlands 10 (2001) 1, S. 778–782.
- [WiHa95] *Winter, Alfred; Haux, Reinhold*: A Three-Level Graph-Based Model for the Management of Hospital Information Systems. In: Methods of Information in Medicine 34 (1995) 4, S. 378–396.

Abstract

Strategies for coupling enterprise application systems in hospitals

The realization of specific coupling strategies and integration techniques is discussed exemplarily for a university hospital, the Universitätsklinikum Leipzig (UKL) in Germany. The integration of the central patient management system serving the administrative department with the clinical workplace systems serving hospital wards is presented. We analyse the integration with respect to the underlying theoretical conceptions and discuss information management issues for deployment and maintenance of the overall IT infrastructure.

This paper presents the pros and cons of synchronous and asynchronous patient data replication, the practical experience with the deployment of a communication server, and the experience with using an RFC interface, a proprietary interface to the enterprise resource planning system SAP R/3. Both, organizational and technical issues are addressed in this context.

Keywords: hospital information system (HIS), synchronous/asynchronous communication, coupling strategies, communication server, enterprise application integration, schema integration, data replication