

Serviceorientierte Integration medizinischer Geräte – ganzheitliche IT-Unterstützung klinischer Prozesse

Christian Mauro · Jan Marco Leimeister
Helmut Krcmar

**Deutsche Krankenhäuser
leiden oftmals unter
großen informations-
logistischen Defiziten.
Das Konzept der
serviceorientierten
Geräteintegration
(SODA) kann helfen,
eine medienbruchfreie
und durchgängige
IT-Unterstützung
klinischer Prozesse
zu ermöglichen.**

Einleitung

Ein Szenario, wie es tagtäglich in deutschen Kliniken anzutreffen ist: Herr Müller kommt mit der Verdachtsdiagnose Kolonkarzinom in die Notaufnahme eines Krankenhauses. Routinemäßig werden ein EKG (Elektrokardiogramm), eine Blutuntersuchung sowie

ein Röntgen-Thorax durchgeführt. Das EKG wird in Papierform erstellt und in der Patientenakte abgeheftet. Das Röntgenbild wird in einer separaten Röntgentüte des Patienten aufbewahrt. Die Ergebnisse der Blutuntersuchung sind nach Fertigstellung meist in der elektronischen Patientenakte (ePA) gespeichert. Um alle Befunde des Patienten zu sichten, muss der Arzt somit drei verschiedene Dokumentenordner (die ePA, die Papierakte und die Röntgentüte) kontrollieren. Durch diese umständliche und zeitaufwendige Prozedur bleibt dem behandelnden Arzt oft nur wenig Zeit für den Patientenkontakt. Da sich nach weiteren Untersuchungen die Verdachtsdiagnose bestätigt, wird Herr Müller in die Chirurgie verlegt und ein OP-Termin geplant. Alle benötigten Ressourcen (Personal, Räume, Geräte etc.) werden hierfür über einen Case Manager manuell gebucht. Für die OP relevante Daten stehen teils in der Papierakte, teils über die ePA zur Verfügung. Ob bereits alle Daten zur Verfügung stehen, hängt davon ab, ob zum Zeitpunkt der Verlegung alle Papierbelege auf der Station ein-

gegangen sind. Wenn nicht, müssen die fehlenden Papierbefunde auf die chirurgische Station nachgeschickt werden, was meist ein hoher logistischer Aufwand ist. Während der Operation werden die von den verschiedensten Geräten erfassten Daten vom Anästhesisten dokumentiert und in der Papierakte abgelegt. Nach der Operation wird Herr Müller in der Intensivstation an Überwachungsgeräte sowie Pumpen zur Verabreichung von Medikamenten und Nahrung angeschlossen. Die erfassten Daten werden wiederum in Überwachungsprotokollen von den Schwestern und Ärzten handschriftlich und in Papierform dokumentiert. Jeder Tag hat einen einzelnen Bogen. Medikamente müssen auf jedem Bogen neu erfasst werden. Die Übertragung erfolgt meist durch den Nachtdienst, der die Kurve zwischen 3 und 6 Uhr morgens überträgt. Dies ist zum einen fehleranfällig und beschäftigt zudem eine Arbeitskraft pro Überwachungsstation für mindestens eine Stunde nur für die manuelle Übertragung der Kurven. Allergien oder Wechselwirkungen müssen vom Arzt anhand der vorliegenden Dokumente geprüft werden. Bei unvollständiger Dokumentation kann es auch hier leicht zu Fehlern kommen, die im schlimmsten Fall Menschenleben gefährden. Herr Müller hingegen hat Glück und kann nach erfolg-

DOI 10.1007/s00287-010-0436-4
© Springer-Verlag 2010

Christian Mauro · Helmut Krcmar
Technische Universität München,
Lehrstuhl für Wirtschaftsinformatik,
Boltzmannstr. 3, 85748 Garching
E-Mail: {mauro, krcmar}@in.tum.de

Jan Marco Leimeister
Universität Kassel, Fachbereich Wirtschaftsinformatik,
Nora-Platiel-Straße 4, 34127 Kassel
E-Mail: leimeister@uni-kassel.de

Zusammenfassung

Deutsche Krankenhäuser leiden oftmals unter großen informationslogistischen Defiziten, verursacht durch Medienbrüche, fehlende Interoperabilität von Systemen und organisationale Friktionen oder Restriktionen. Dies führt zu Effizienz- und Effektivitätsverlusten der medizinischen und administrativen Prozesse sowie der Behandlungsqualität.

Ein aktueller Trend im Bereich der Software-Systeme im Gesundheitswesen sind serviceorientierte Architekturen (SOA), um den meist historisch gewachsenen heterogenen IT-Strukturen zu begegnen. Medizinische Geräte als wesentlicher Bestandteil medizinischer Prozesse bleiben jedoch meist unberücksichtigt. Vielversprechend erscheint daher die Anwendung des SOA-Paradigmas auf das Problemfeld der Integration medizinischer Geräte. Dies soll eine ganzheitliche Integration aller an den klinischen Prozessen beteiligten Informationsquellen und damit eine medienbruchfreie und durchgängige IT-Unterstützung von Prozessen ermöglichen.

Dieser Beitrag gibt einen Überblick über das Konzept der serviceorientierten Integration medizinischer Geräte (SODA) und fokussiert insb. auf Herausforderungen beim Design entsprechender Integrationsarchitekturen sowie Design Patterns, die diese Probleme adressieren.

reichem Behandlungsverlauf das Krankenhaus verlassen.

Das Szenario beschreibt beispielhaft typische informationslogistische Defizite in deutschen Krankenhäusern. Dem entgegen steht die Vision des „Seamless Healthcare“, bei der vertikal und horizontal durchgängige Prozesse, Daten und Informationstechnologien es ermöglichen, dass jeder Akteur die richtige Information, zum richtigen Zeitpunkt, am richtigen Ort in der richtigen Qualität und Quantität erhält [33]. Zur Erreichung dieser Vision müssen alle am Prozess beteiligten IT-Systeme integriert werden. Aufgrund der vorherrschenden heterogenen IT-Strukturen in Krankenhäusern ist die Realisierung einer solchen Gesamtintegration jedoch sehr komplex [35]. Ein aktueller Trend im Bereich der Softwaresysteme im Gesundheitswesen zur

Erreichung dieses Ziels sind serviceorientierte Architekturen (SOA), die als Ansatz zur Begegnung der gewachsenen heterogenen IT-Strukturen dargestellt werden [5, 29, 30, 34]. Unberücksichtigt bleiben hierbei meist medizinische Geräte, die – wie im Szenario angedeutet – jedoch ein wichtiger Datenlieferant sowie wesentlicher Bestandteil medizinischer Prozesse sind. Dieser Beitrag fokussiert daher auf die Erweiterung der Anwendung des SOA-Paradigmas auf das Problemfeld der Integration medizinischer Geräte. Dies soll eine ganzheitliche Integration aller an den klinischen Prozessen beteiligten Informationsquellen (wie z. B. Geräte) und damit eine optimale durchgängige IT-Unterstützung von Prozessen ermöglichen. Durch Anwendung serviceorientierter Konzepte sollen zudem die Anpassung der IT an sich ändernde Prozesse sowie die Entwicklung neuer Applikationen mit Beteiligung von Geräten schneller und einfacher möglich sein. Darüber hinaus sollen typische Herausforderungen der Geräteintegration (bspw. Plug-and-Play-Mechanismen) adressiert werden, um den Anwender bestmöglich bei seiner Arbeit zu unterstützen.

Dieser Beitrag gibt einen Überblick über das Konzept der serviceorientierten Integration medizinischer Geräte und fokussiert insbesondere auf Herausforderungen beim Design entsprechender Integrationsarchitekturen sowie Design Patterns, die diese Probleme adressieren. Allgemeine Fragestellungen bezüglich des Einsatzes von SOA im Krankenhaus können wegen des begrenzten Umfangs nicht thematisiert werden. Der Beitrag gliedert sich wie folgt: Im Abschnitt „IT-Unterstützung klinischer Prozesse“ führen wir in die Thematik integrierter Behandlungspfade sowie deren IT-Unterstützung ein. Dazu wird auch auf die Integration medizinischer Geräte sowie serviceorientierte Architekturen eingegangen. Der Abschnitt „Geräteintegration mit SODA“ erläutert zunächst das Konzept der Service Oriented Device Architecture (SODA) und geht dann über zu den spezifischen Herausforderungen bei der Anwendung serviceorientierter Konzepte auf Geräte. Anschließend werden ausgewählte, bereits existierende und neue Design Patterns vorgestellt, die Lösungen für Herausforderungen serviceorientierter Geräteintegration aufzeigen. Der Abschnitt „Verwandte Arbeiten“ gibt einen kurzen Einblick in weitere relevante Forschungsprojekte in diesem Bereich. Der Abschnitt „Fazit und offene Fragestellungen“

Abstract

German hospitals often face major shortcomings in their information logistics. This is caused by media frictions, isolated systems, lacking interoperability of systems and organizational limitations. In consequence, efficacy and efficiency losses of medical and administrative processes as well as the quality of treatment have to be stated.

A current trend in the field of software systems in healthcare are service oriented architectures (SOA) to address historically grown heterogeneous IT infrastructures. Medical devices are an essential part of medical processes but they remain unconsidered by most approaches. Applying the concept of SOA to the integration of medical devices seems to be very promising. This could enable a concise and comprehensive integration of all sources of information and therefore a seamless IT support of clinical processes.

This article provides an overview of the concept of service oriented device integration (SODA). It especially focuses on challenges when designing solution architectures based on this concept as well as design patterns that try to solve the specific design problems.

fasst die Ergebnisse des Beitrags zusammen und gibt einen kritischen Ausblick auf offene Fragestellungen und weitergehende Forschungsbedarfe.

IT-Unterstützung klinischer Prozesse

Integrierte Behandlungspfade

Das Geschäftsprozessmanagement gewinnt in Krankenhäusern zunehmend an Bedeutung. Spätestens seit der Einführung von DRGs (Diagnosis Related Groups) im Jahre 2004, die eine Abrechnung auf Basis von Diagnosen statt der Verweildauer vorsieht [8], ist neben administrativen Prozessen auch die Optimierung medizinischer Prozesse ein wesentlicher Schritt zur Kostensenkung. Integrierte Behandlungspfade erscheinen als ein geeignetes Steuerungsinstrument zum Management medizinischer Prozesse. Sie beschreiben „den optimalen Weg eines speziellen Patiententyps mit seinen entscheidenden diagnostischen und therapeutischen

Leistungen und seiner zeitlichen Abfolge“ [9, S. 9]. Fünf Zielkategorien werden hierbei verfolgt, für die jeweils entsprechende Kennzahlen bzw. Qualitätsindikatoren definiert werden müssen [2]: Kundenzufriedenheit, Qualität, Zeit, Termintreue und Kosten.

IT kann wesentliche Unterstützungsfunktionen bei der Anwendung von integrierten Behandlungspfaden leisten. Bezogen auf medizinische Prozesse sind dies insbesondere [21]

- das einfache und schnelle Auffinden benötigter Informationen, die dem Nutzer zusammen mit dem Pfad durch eine angemessene Visualisierung dargestellt werden, sowie
- das Anzeigen kontextabhängiger Programmelemente. Abhängig vom aktuellen Patienten und dessen Status im Behandlungspfad könnten beispielsweise im Pfad vorgesehene nächste Behandlungsschritte angezeigt werden, deren zugehörige Aktivitäten sich direkt anstoßen lassen.

Auch administrative Prozesse können durch IT unterstützt werden, indem Kennzahlen (z. B. die Auslastung hochpreisiger medizinischer Geräte) automatisiert erfasst und analysiert werden. Diese Optimierung der Geschäftsprozesse kann als Wertbeitrag der eingesetzten IT verstanden werden [20]. Insbesondere im Krankenhaus lassen sich hierzu vielfältige Potenziale identifizieren [12].

Serviceorientierte Architekturen

Im Krankenhaus setzt sich die Gesamtarchitektur der IT aus einer Vielzahl spezialisierter Systeme (Laborsystem, Radiologieinformationssystem etc.) mit oftmals lokaler Datenhaltung zusammen (Abb. 1). Die so entstehenden heterogenen Strukturen lassen sich nur mit großen Aufwänden zu einem interoperablen Gesamtsystem integrieren, da die Schaffung der benötigten Schnittstellen zeit- und kostenintensiv ist [35]. Dies führt zu nicht integrierten Subsystemen und damit zu redundanten Datenhaltungen sowie Medienbrüchen bzw. nicht verfügbaren Informationen. Eine komplette IT-Unterstützung integrierter Behandlungspfade benötigt jedoch genau diese Gesamtintegration aller beteiligten IT-Systeme, da die einzelnen Systeme jeweils nur einen Teil des Gesamtprozesses abbilden [37]. Zwar existieren im Gesundheitswesen eine ganze Reihe an IT-Standards (vgl. [16, 36]), jedoch

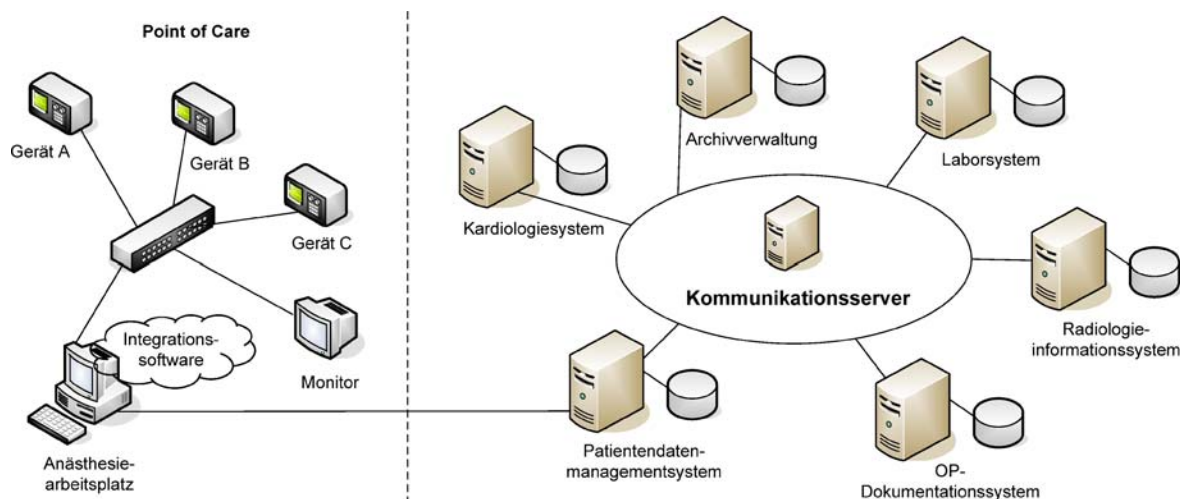


Abb. 1 Komponenten einer typischen Krankenhaus-IT-Infrastruktur (in Anlehnung an Haas [15] und Khetepal [18])

werden diese von den Herstellern nur vereinzelt umgesetzt. Zum einen ist das Einhalten solcher Standards kostenintensiv und zum anderen wird durch proprietäre Insellösungen eine Herstellerbindung erzeugt, da das Ablösen solcher Produkte aufwendig ist [37].

Im Gesundheitswesen ist daher ein Trend hin zu serviceorientierten Architekturen zu beobachten, um den Herausforderungen heterogener IT-Architekturen zu begegnen. Das wohl derzeit prominenteste Beispiel ist die Einführung der elektronischen Gesundheitskarte. Die zugehörige Telematikinfrastruktur ist als serviceorientierte Architektur konzipiert [14]. Doch nicht nur institutionsübergreifend, sondern auch innerhalb eines einzelnen Krankenhauses können serviceorientierte Konzepte einen Mehrwert leisten. Durch die Kapselung der verschiedenen proprietären Altsysteme als Services werden die funktionalen Verantwortlichkeiten klar definiert. Beispielsweise ist genau ein Service für die Verwaltung der administrativen Patientendaten zuständig. Eine redundante und damit potenziell inkonsistente Datenhaltung wird somit vermieden. Darüber hinaus lassen sich Services zu komplexen Abläufen kombinieren [23]. Dies soll eine nahe Ausrichtung der IT an den Geschäftsprozessen sowie Flexibilität im Falle sich ändernder Prozesse ermöglichen. Die nahe Ausrichtung der IT an den Prozessen kann im Krankenhaus helfen, das Personal mit allen aktuell benötigten medizinischen Informationen zu versorgen und damit die Qualität der Behandlung zu erhöhen. Aufgrund dessen be-

ginnen erste Krankenhäuser, wie z. B. das Klinikum München rechts der Isar, mit der Umsetzung von Pilotprojekten zur Realisierung serviceorientierter Architekturen im Krankenhaus [32]. Weitere Forschung und Praxisprojekte werden zeigen, ob sich die gegenwärtigen Probleme durch den Einsatz serviceorientierter Konzepte dezimieren lassen. In der Folge fokussiert der Beitrag auf ein noch junges Forschungsgebiet, das sich mit der serviceorientierten Integration medizinischer Geräte befasst und daher einen Teilaspekt serviceorientierter Architekturen im Krankenhaus adressiert.

Integration medizinischer Geräte

Neben der Integration von Softwaresystemen ist auch die Einbindung medizinischer Geräte ein wesentlicher Aspekt für eine ganzheitliche IT-Unterstützung klinischer Prozesse, da Geräte ein wesentlicher Bestandteil der medizinischen Behandlung und damit eine wichtige Datenquelle sind. Wie auch bei den Softwaresystemen, gestaltet sich die Integration schwierig, da eine Interoperabilität medizinischer Geräte nicht gegeben ist [22]. Existierende Standards werden von den Herstellern nur selten umgesetzt, um die Bindung des Kunden zu stärken. Darüber hinaus ist kein Standard umfangreich genug, um Interoperabilität auf allen Ebenen zu leisten. Beispielsweise arbeitet der Standard HL7 nur auf der Applikationsebene und kann daher keine technische Interoperabilität sicherstellen [16]. Die Art des Übertragungsprotokolls wird z. B. von HL7 nicht definiert. Umfänglichere Standards wie

die ISO/IEEE 11073 Serie adressieren Interoperabilität auf allen Ebenen, sind jedoch sehr komplex und berücksichtigen zudem aktuelle Anforderungen nicht [31]. Beispielsweise beschränken sich die einzigen verfügbaren Kommunikationsprofile (ISO/IEEE 11073-30200 und ISO/IEEE 11073-30300) auf eine Übertragung auf IrDA-Basis (Infrared Data Association). Für den Austausch von interpretierbaren Informationen zur Unterstützung klinischer Prozesse ist jedoch eine Interoperabilität auf allen Ebenen nötig [15]. Ein Schritt in diese Richtung ist der Zusammenschluss verschiedener Standards zu Profilen, getrieben z. B. von Initiativen wie IHE (Integrating the Healthcare Enterprise¹).

Die potenziellen Vorteile interoperabler medizinischer Geräte sind vielfältig, z. B. eine bessere Behandlungsqualität durch bessere Informationsqualität bei der Behandlung, verbesserte Workflows, Möglichkeiten zur Gerätefernwartung etc. [22]. Medizinische Geräte sind nicht nur Bestandteil medizinischer Prozesse, sondern auch für IT-Prozesse und administrative Prozesse relevant. Im Kontext von IT-Prozessen ermöglichen integrierte Geräte eine lückenlose Überwachung des Gerätelebenszyklus. Beispielsweise könnten für Wartungsprozesse alle benötigten Informationen (wie Alter, Nutzungshäufigkeit, Hersteller, Modell, Version, Standort etc.) automatisiert bereitgestellt werden. Für administrative Prozesse sind ebenfalls spezifische Daten von Interesse. Beispielhaft erwähnt sei im Kontext von Abrechnungsprozessen die Beatmungsdauer im Falle der künstlichen Beatmung eines Patienten. Diese ist abrechnungsrelevant [6] und eine automatisierte Erfassung daher sinnvoll, um alle getätigten Leistungen effizient abrechnen zu können. Daher überrascht es nicht, dass Initiativen wie IHE versuchen, die Standardisierung auf allen Interoperabilitätsebenen voranzutreiben. Dennoch stellt sich die aktuelle Situation wie folgt dar:

„Interoperability is an almost non-existent feature of medical devices.“ [22, S. 3]

Geräteintegration mit SODA

Allgemeines Konzept

Die Idee der serviceorientierten Geräteintegration ist es, ein Gerät als Service zu kapseln, analog zu Softwareservices einer serviceorientierten Archi-

tektur, siehe Abb. 2. Dieses Konzept ist in der Literatur auch unter der Bezeichnung SODA (Service Oriented Device Architecture) bekannt [7]. Ein Service ist eine Softwarekomponente, die eine bestimmte Funktionalität zur Verfügung stellt und deren Schnittstelle in standardisierter Form (z. B. bei Verwendung der Webservicetechnologie mittels der Webservice Description Language – WSDL) spezifiziert ist [19]. Wir unterscheiden Softwareservices und Geräteservices. Softwareservices kommen zwar auf einem Server zur Ausführung, sind jedoch ansonsten unabhängig von Hardwarekomponenten. Geräteservices hingegen sind zusätzlich zur Ablaufplattform an ein bestimmtes Gerät gebunden. Welche Besonderheiten sich hieraus ergeben, wird im nächsten Abschnitt diskutiert. Ein Geräteservice könnte z. B. in der medizinischen Domäne eine Funktionalität anbieten, die den aktuellen Blutdruck eines Patienten zurückgibt. Darauf aufbauend können höherwertige Services implementiert werden, die z. B. ein Patientenmonitoring umsetzen.

Die Vorteile von SODA sind insbesondere in der losen Kopplung sowie der Komponierbarkeit von Services zu sehen. Die Kapselung als Service bewirkt, dass bei der Implementierung einer Funktionalität, die einen Geräteservice aufruft, die herstellerspezifische Schnittstelle des Geräts nicht bekannt sein muss. Diese wurde durch die serviceorientierte Integration bereits abstrahiert. Die Entwicklung von Anwendungen mit Gerätezugriff wird somit erleichtert, da keine manuelle Programmierung der Schnittstelle zum Gerät erfolgen muss. Die zum Zugriff auf das Gerät notwendigen Softwarekomponenten lassen sich z. B. bei der Verwendung von Webservicetechnologien innerhalb der gängigen Entwicklungsumgebungen (wie Eclipse oder Microsoft Visual Studio) automatisch generieren. Der Aufruf des Geräteservices erfolgt durch ein standardisiertes Protokoll (z. B. SOAP) auf semantischer Ebene (z. B. „getBloodPressure“). Durch eine lose Kopplung müssen Anwendungen, die ein bestimmtes Gerät nutzen, nach einem Gerätewechsel (anderer Hersteller oder anderes Modell) nicht angepasst werden, da die Kommunikation über den Geräteservice erfolgt. Lediglich die Komponente, die das Gerät integriert, muss gegebenenfalls modifiziert und bei Bedarf erweitert werden. Die vorherige Serviceschnittstelle kann jedoch unverändert bzw. abwärtskompatibel bleiben. Dies ermöglicht die Komponierbarkeit

¹ <http://www.ihe.net>

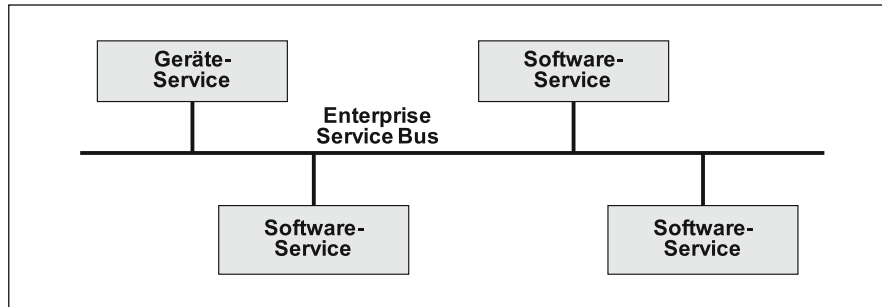


Abb. 2 Geräteservices in Analogie zu Softwareservices (in Anlehnung an de Deugd et al. [7])

von Services, durch die die IT-Unterstützung der Prozesse auch auf Geräte erweitert werden kann, z. B. unter Verwendung der Business Process Execution Language (BPEL, vgl. auch [4]). Auch institutionsübergreifende Szenarien sind denkbar.

Ein weiterer Vorteil der Kapselung als Service ist, dass einem Gerät Funktionalitäten logisch hinzugefügt werden können, die es ursprünglich nicht besitzt. Beispielsweise könnte eine Funktion „getLocation“ angeboten werden, die den aktuellen Standort des Geräts angibt. Die Implementierung der Funktion könnte dabei auf ein separates Tracking&Tracing-System zurückgreifen. Das Erkennen und Bewerten solcher Nutzenpotenziale ist ein wesentlicher Beitrag, den die Wissenschaft in diesem Forschungsgebiet leisten muss.

SODA Designprobleme

Das Konzept der serviceorientierten Geräteintegration wurde zunächst auf sehr abstraktem Level beschrieben, insbesondere um die Idee zu transportieren. Geräte sind jedoch als physikalische Objekte nicht nur Teil der IT-Unterstützung von Prozessen,

sondern vielmehr Teil der Prozesse selbst. Aufgrund dieser Besonderheit von Geräten im Vergleich zu Software konnten sieben SODA-spezifische Designprobleme identifiziert werden [27]. Abbildung 3 ordnet diese in das Geflecht von Geräten, Geräteservices, Servicekonsumenten und Service Registry ein. Die grundsätzliche Herausforderung bei der Anwendung des SODA-Konzepts ist die Entkopplung der Servicekonsumenten von den Geräten. Designproblem 1 adressiert hierbei die Kommunikation zwischen Geräten und den zugehörigen Geräteservices, während Designproblem 2 den Aspekt der Entkopplung auf Ebene des Servicevertrags behandelt. In Designproblem 3 wird das Problem der wechselnden proprietären Schnittstellen im Falle eines Gerätewechsels aufgegriffen. Designproblem 4 adressiert die logische Trennung von Geräteservices und Softwareservices in der Service Registry. Die Möglichkeit eines exklusiven Zugriffs auf Geräte ist Gegenstand von Designproblem 5. In Designproblem 6 wird aufgegriffen, wie Geräteservices dynamisch in der Service Registry publiziert bzw. entfernt werden können. Der Umgang mit

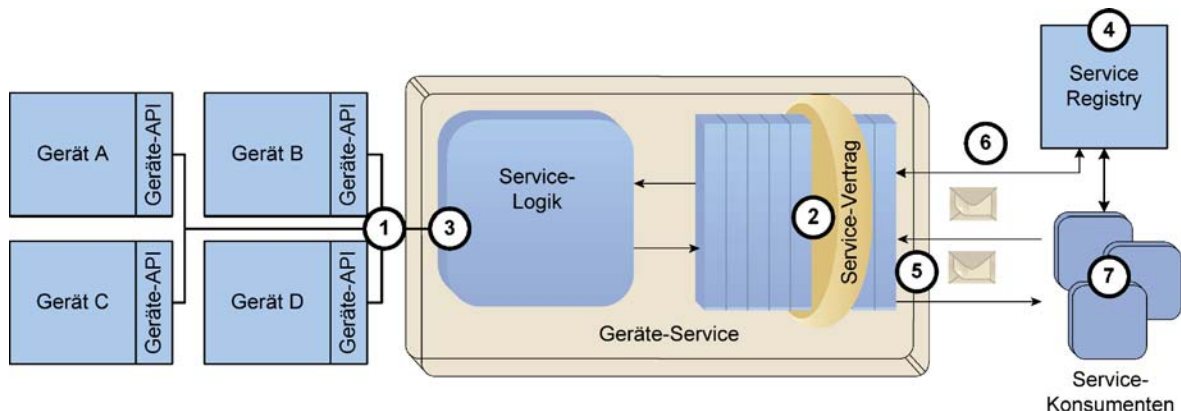


Abb. 3 Lokalisierung der Designprobleme (in Anlehnung an Mauro et al. [27])



Patterns für SODA Designprobleme (neu entworfene Patterns sind kursiv gekennzeichnet)

Designproblem	Design Patterns
1	<i>Integrated Adapter, External Adapter, Server Adapter</i>
2	Legacy Wrapper (Erl, Roy)
3	<i>Dynamical Adapter</i>
4	Service Layers (Erl), Utility Abstraction (Erl)
5	Trusted Subsystem (Hogg et al.), Contract Centralization (Erl), Service Callback (Karmakar), Asynchronous Queuing (Little, Rischbeck, Simon), Event-Driven Messaging (Little, Rischbeck, Simon)
6	<i>Auto-Publishing und/oder Decentralized Service Discovery</i>
7	<i>Device Concentrator</i>

Geräteservices durch die Servicekonsumenten wird in Designproblem 7 adressiert.

Im Rahmen dieses Beitrags können nicht alle sieben Designprobleme im Detail erläutert werden. Stattdessen werden beispielhaft zwei dieser Designprobleme aufgegriffen, die grundlegende Problematik erläutert sowie im nächsten Abschnitt Lösungsansätze in der Form von Design Patterns vorgestellt. Jedoch wurden auch für die anderen, hier nicht näher betrachteten, Designprobleme zugehörige Design Patterns entwickelt (Tabelle 1).

Designproblem 2: Serviceverträge enthalten neben Service Level Agreements insbesondere die Definition der technischen Schnittstellen [10]. Beinhaltet diese Definition herstellerspezifische Aspekte, z. B. Teile der proprietären Geräteschnittstelle, läge indirekt eine enge Kopplung zwischen Servicekonsumenten und Geräten vor. Würde z. B. der Hersteller gewechselt werden, müsste der Servicevertrag entsprechend angepasst werden, weil die

herstellerspezifischen Aspekte nicht mehr zutreffend sind. Als Konsequenz müssten auch die Servicekonsumenten ihre Implementierung anpassen. Es stellt sich daher die Frage, wie der Servicevertrag zu gestalten ist, um eine solche enge Kopplung zu vermeiden.

Designproblem 7: Eine Besonderheit von Geräteservices ist, dass diese jederzeit spontan instanziiert oder aufgelöst werden können, z. B. wenn das zugehörige Gerät ein- bzw. ausgeschaltet wird. Dies soll am Beispiel von Spritzenpumpen verdeutlicht werden, welche zur automatisierten kontinuierlichen Verabreichung von Medikamenten und flüssiger Nahrung verwendet werden. Abbildung 4a zeigt vier Spritzenpumpen eines Patienten, von denen zwei eingeschaltet sind. Für diese aktiven Pumpen wurde je ein Geräteservice instanziiert und an die Service Registry publiziert. Mehrere Servicekonsumenten haben diese Services über die Service Registry gefunden und nutzen diese. Aus statischer

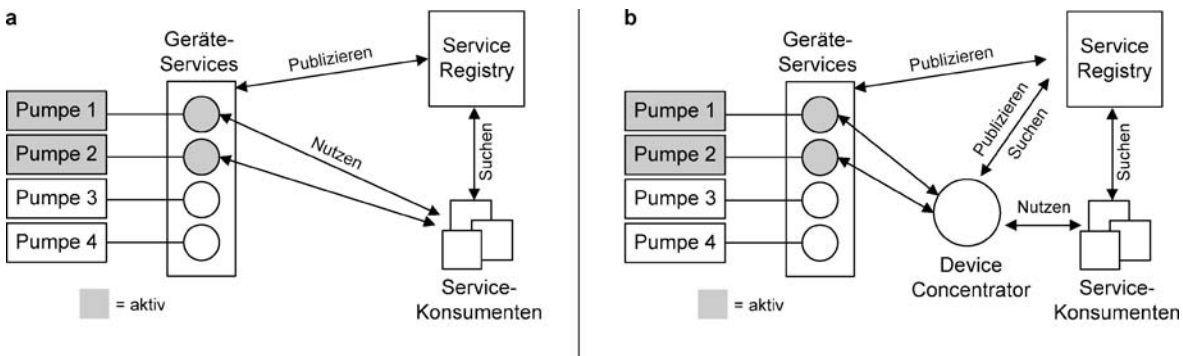


Abb. 4 Geräteservices für Spritzenpumpen

Sicht ist dieses Beispiel unproblematisch, aus dynamischer Sicht jedoch können Pumpen jederzeit ein- oder ausgeschaltet werden. Wird z. B. Pumpe 3 aktiviert und ein zugehöriger Geräteservice instanziiert und publiziert, ist den betroffenen Servicekonsumenten der neue Geräteservice zunächst nicht bekannt. Alle Servicekonsumenten müssten somit eigenständig, z. B. durch regelmäßige Anfragen an die Service Registry, verwalten, welche Pumpen aktuell für einen bestimmten Patienten verfügbar sind. Dies ist aufwendig, erzeugt redundante Implementierungen aufseiten der Servicekonsumenten und ist entsprechend fehleranfällig. Es stellt sich daher die Frage, wie diese Problematik zu lösen ist.

Design Patterns zur Lösung der identifizierten Designprobleme

Die eben beschriebenen Designprobleme adressieren Fragen, die in jedem Projekt relevant sein können, die sich dem Konzept serviceorientierter Geräteintegration bedienen. Daher ist es zielführend, nach allgemeingültigen Lösungskonzepten für diese Designprobleme zu suchen. Lösungen für wiederkehrende Probleme werden auch als Patterns bezeichnet. Die Idee von Patterns geht im Bereich der Architektur auf Alexander [1] zurück sowie im Bereich der Softwareentwicklung auf Gamma [13]. Erl [11] definiert eine Sammlung von Patterns, die SOA-spezifische Designprobleme adressieren.

Eine Literaturanalyse ergab, dass sich drei der Designprobleme mit bereits existierenden SOA Design Patterns lösen lassen, während für die verbleibenden vier Designprobleme keine geeigneten Patterns in der Literatur identifiziert werden konnten (für die vollständige Analyse siehe [27]). Für die offenen Designprobleme wurden daher neue Pattern-Kandidaten entwickelt. In Tabelle 1 sind existierende und neue Patterns den Designproblemen zugeordnet. Für die Erläuterung der einzelnen Patterns sei auf [11, 27, 28] verwiesen. Wie Tabelle 1 zu entnehmen ist, sind den im letzten Abschnitt beispielhaft erläuterten Designproblemen 2 und 7 das Legacy Wrapper Pattern bzw. das Device Concentrator Pattern zugeordnet. Im Folgenden werden die grundlegenden Lösungsideen dieser Patterns im Kontext der vorgestellten Designprobleme erläutert.

Durch Anwendung des Legacy Wrapper Patterns sollen Servicekonsumenten von den Geräten entkoppelt werden (vgl. Designproblem 2). Im

Wesentlichen empfiehlt das Pattern das Erstellen eines standardisierten Servicevertrags, der von proprietären Aspekten des Legacy Systems (hier: Geräte) bereinigt ist [11]. Statt der Verwendung herstellerspezifischer Datenmodelle empfiehlt sich die Berücksichtigung existierender Standards, z. B. ISO/IEEE 11073-10101 [31]. Typische herstellerspezifische Daten sind z. B. auch Fehlercodes oder Nachrichten-IDs [11]. Diese sollten in vereinheitlichte Formate transformiert werden. Auf diese Weise wird den Servicekonsumenten eine stabile Schnittstelle angeboten, die sich auch bei einem Gerätewechsel nicht ändert. Zu erwähnen ist, dass durch die Standardisierung des Servicevertrags herstellerspezifische Funktionalitäten nicht ansprechbar sind, da nur der größte gemeinsame Nenner in den Servicevertrag einfließt. Dies kann jedoch als unkritisch angesehen werden. Ist die Nutzung einer solchen herstellerspezifischen Funktionalität zwingend notwendig, ist ohnehin nur die Verwendung von Geräten möglich, die diese Funktionalität anbieten. Somit ist diese Funktionalität auch Teil des größten gemeinsamen Nenners.

Im letzten Abschnitt wurde erläutert, welche Probleme sich für Servicekonsumenten durch den dynamischen Charakter von Geräteservices ergeben (Designproblem 7). Für dieses Problem konnte in der Literatur kein passendes Design Pattern zur Lösung gefunden werden. Zu diesem Zwecke wurde ein neuer Pattern-Kandidat – das Device Concentrator Pattern – entwickelt. Das Pattern empfiehlt die Implementierung eines Services, der für eine bestimmte Auswahl an Geräteservices zuständig ist und deren Verfügbarkeit überwacht. Im Falle des Beispiels aus dem letzten Abschnitt könnte dies die Menge aller Geräteservices sein, welche mit den Spritzenpumpen eines bestimmten Patienten kommunizieren (Abb. 4b). Servicekonsumenten müssen auf diese Weise nicht jede Pumpe einzeln ansprechen. Beispielsweise könnte ein solcher Service eine Funktionalität anbieten, welche die aktuelle Medikamentierung (Medikamente, Dosen etc.) eines Patienten angibt. Eine andere Möglichkeit, das Device Concentrator Pattern umzusetzen, ist die Implementierung eines Services, der z. B. im intensivmedizinischen Bereich alle Geräte zu einem bestimmten Patienten anspricht und überwacht. Alle Geräte am Patientenbett wären somit über eine zentrale Serviceinstanz ansprechbar. Für das Identifizieren solcher Servicekandidaten

kann auf bewährte SOA-Methoden zurückgegriffen werden. Geeignet erscheint das Meet-in-the-Middle-Verfahren, welches das Top-Down-Verfahren (ausgehend von den Prozessen) mit dem Bottom-Up-Verfahren (ausgehend von den existierenden Softwaresystemen) kombiniert [24]. Diese Kombination ist für SODA-Projekte nötig, da neben den medizinischen und administrativen Prozessen insbesondere auch die von den Geräten zur Verfügung gestellten Funktionalitäten berücksichtigt werden müssen.

Verwandte Arbeiten

Im Forschungsgebiet serviceorientierter Geräteintegration gibt es eine Reihe an Veröffentlichungen, die unterschiedlichste Domänen adressieren. Für eine Analyse dieser Literatur sei auf [26] verwiesen. An dieser Stelle sollen nicht einzelne Publikationen sondern die wichtigsten Forschungsprojekte genannt werden. Das EU-Projekt SIRENA (Service Infrastructure for Real time Embedded Networked Applications) setzte sich als eines der ersten Vorhaben mit der Portierung des SOA-Konzeptes auf Geräte auseinander [3]. Als Domäne wurde insbesondere die industrielle Automatisierung adressiert [17]. Follow-Up-Projekte sind SODA² und SOCRADES³ (Service-Oriented Cross-layer Infrastructure for Distributed Smart). Während SODA den Aufbau eines serviceorientierten Ecosystems zum Ziel hatte, das auf den Ergebnissen des SIRENA-Projekts aufbaut, adressiert SOCRADES neue Methoden, Technologien und Werkzeuge zu Modellierung, Design, Implementierung und Betrieb von vernetzten eingebetteten Systemen. Darüber hinaus wurde das SOA4D-Projekt⁴ initiiert, das eine Open-Source-Implementierung des DPWS Stacks (Devices Profile for Web Services) zum Ziel hat. Auch die Projekte OSAMI⁵ (Open Source Ambient Intelligence Commons for an Open and Sustainable Internet) und LOMS⁶ (Local Mobile Services) basieren auf diesem Stack.

Im medizinischen Kontext seien die vom Bundesministerium für Bildung und Forschung (BMBF) geförderten Projekte FUSION⁷ (Future Environ-

ment for Gentle Liver Surgery Using Image-Guided Planning and Intra-Operative Navigation) und orthoMIT⁸ (Minimal-Invasive Orthopädische Therapie) erwähnt, die sich in die BMBF-Leitvision „Schonendes Operieren mit innovativer Technik (SOMIT)“⁹ eingliedern. Auf technischer Ebene adressieren diese Projekte auch die serviceorientierte Integration medizinischer Geräte im Operationssaal.

Ein weiteres relevantes Projekt ist das Eclipse Open Health Framework (OHF)¹⁰. Ein Teilprojekt des OHF (SODA) adressiert die serviceorientierte Geräteintegration auf Basis von OSGI. Kernelemente des Teilprojekts sind das Device Kit, das die Kommunikation mit den Geräten auf Basis von Java ermöglicht sowie das Service Activator Toolkit, das die Entwicklung von OSGI Bundles adressiert.

Fazit und offene Fragestellungen

Die serviceorientierte Geräteintegration ist ein vielversprechendes Konzept, um eine durchgängige IT-Unterstützung von Geschäftsprozessen im Krankenhaus umzusetzen. Nach der Integration von Softwaresystemen mittels serviceorientierter Konzepte ist die Integration von Geräten mit ähnlichen Mitteln der nächste logische Schritt.

Wie im Beitrag gezeigt, ergeben sich durch die Besonderheiten von Geräten spezifische Designprobleme. Diese konnten teilweise mit existierenden SOA-Design-Patterns gelöst werden. Für noch offene Designprobleme wurden neue Design Patterns vorgeschlagen. Gegenstand weiterer Forschung muss es nun sein, die entworfenen Design Patterns in der Praxis zu evaluieren und gegebenenfalls zu verbessern.

Neben der Analyse von allgemeinen Designproblemen sind in dem jungen Forschungsgebiet aber noch eine Reihe weiterer Aspekte zu betrachten sowie kritisch zu hinterfragen, z. B.:

- Technologieauswahl: Welche Technologien sind zur Umsetzung des Konzepts in welchen Szenarien geeignet?
- Performanz und Stabilität: Sind (in der Regel auf XML basierende) SOA-Technologien schnell und stabil genug, um sich im medizinischen Umfeld etablieren zu können?

² <http://www.soda-itea.org>

³ <http://www.socrates.eu>

⁴ <https://forge.soa4d.org>

⁵ <http://www.osami-commons.org>

⁶ <http://www.loms-itea.org>

⁷ <http://www.somit-fusion.de>

⁸ <http://www.orthomit.de>

⁹ <http://www.gesundheitsforschung-bmbf.de/de/984.php>

¹⁰ <http://www.eclipse.org/ohf/components/soda>

- Wie lässt sich das Konzept einsetzen, um eine bestmögliche IT-Unterstützung medizinischer, informationstechnischer und abrechnungstechnischer Prozesse zu realisieren?
- Lässt sich das Konzept mit den umfassenden gesetzlichen Vorgaben (ausgehend insbesondere von der Medical Device Directive, vgl. [25]) in Einklang bringen? Welche Auswirkungen ergeben sich hieraus für Hersteller und Betreiber?

Auch lässt sich beobachten, dass der Nutzen des Konzepts in existierenden Arbeiten hauptsächlich auf abstrakter argumentativer Basis dargelegt wird. Analysen, die z. B. eine höhere Behandlungssicherheit bescheinigen oder effektivere Prozesse nachweislich belegen, sollten daher das Ziel weiterer Forschung sein, um eine wissenschaftlich fundierte Evaluierung des Konzepts zu ermöglichen.

Es bleibt abzuwarten, in welche Richtung sich der Medizintechnikmarkt bewegt. Pessimistisch geschätzt wird es auch zukünftig keinen führenden Standard für medizinische Geräte geben. In jedem Fall kann das serviceorientierte Konzept helfen, proprietäre Schnittstellen zu kapseln und somit die damit verbundenen Probleme nicht in die IT-Infrastruktur zu transportieren.

Literatur

1. Alexander C (1979) *The Timeless Way of Building*. Oxford University Press, New York
2. Becker K, Eckardt J (2006) Grundlagen Integrierter Behandlungspfade. In: Eckardt J, Sens B (Hrsg) *Praxishandbuch Integrierte Behandlungspfade: Intersektorale und sektorale Prozesse professionell gestalten*. Economica, Heidelberg, S 39–64
3. Bohn H, Bobek A, Golasowski F (2006) SIRENA – Service Infrastructure for Real-time Embedded Networked Devices: A service oriented framework for different domains. *International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies*, 23.–29.4.2006, Mauritius (ICN/CONS/MCL 2006)
4. Bohn H (2009) *Web Service Composition for Embedded Systems: WS-BPEL Extension for DPWS*. Sierke, Göttingen
5. Bridges MW (2005) SOA in healthcare. *Health Manag Technol* 28:6–10
6. Claßen R (2009) Basisinformationen zum Gesundheitswesen. In: Johnner C, Haas P (Hrsg) *IT im Gesundheitswesen: Erfolgreich einführen, entwickeln, anwenden und betreiben*. Carl Hanser, München, S 109–130
7. de Deugd S, Carroll R, Kelly KE, Millett B, Ricker J (2006) SODA: service oriented device architecture. *Pervasive Comput* 5:94–96
8. Düllings J (2003) Einführung des DRG-Systems und seine Auswirkungen auf die Krankenhausplanung. In: Thiele G (Hrsg) *Einführung der DRGs in Deutschland*. UTB, Stuttgart
9. Eckardt J (2006) Was sind integrierte Behandlungspfade (IBP)? In: Eckardt J, Sens B (Hrsg) *Praxishandbuch Integrierte Behandlungspfade: Intersektorale und sektorale Prozesse professionell gestalten*. Economica, Heidelberg, S 9–37
10. Erl T (2007) *SOA Principles of Service Design*. Prentice Hall International, Boston
11. Erl T (2009) *SOA Design Patterns*. Prentice Hall International, Boston
12. Fähling J, Köbler F, Leimeister JM, Krcmar H (2009) *Wahrgenommener Wert von IT in Krankenhäusern – eine empirische Studie*. *Wirtschaftsinformatik*, Wien
13. Gamma E (1995) *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Massachusetts
14. gematik (2009) Einführung der Gesundheitskarte: Gesamtarchitektur. http://gematik.de/upload/Release_2.3.4_Architektur_und_uebergreifende_Dokumente_080919_4153.zip, letzter Zugriff 17.3.2010
15. Haas P (2005) *Medizinische Informationssysteme und Elektronische Krankenakten*. Springer, Berlin
16. Haas P (2006) *Gesundheitstelematik: Grundlagen, Anwendungen, Potenziale*. Springer, Berlin
17. Jammes F, Smit H (2005) *Service-Oriented Paradigms in Industrial Automation*. *IEEE Transactions Industrial Informatics*, pp 62–70
18. Kheterpal S (2008) *Architecture*. In: Ruskin K, Stonemetz J (eds) *Anesthesia Informatics*. Springer, London, pp 147–165
19. Krafzik D, Banke K, Slama D (2006) *Enterprise SOA – Service-Oriented Architecture Best Practices*. Pearson Education, Indiana
20. Krcmar H (2009) *Informationsmanagement*, 5. Aufl. Springer, Berlin
21. Laprell S, Begemann S (2002) *Entwicklung, Integration und Visualisierung von Klinischen Pfaden mit DV-Unterstützung*. In: Hellmann W (Hrsg) *Klinische Pfade: Konzepte, Umsetzung, Erfahrungen*. Ecomed, Landsberg/Lech, S 194–205
22. Lesh K, Weininger S, Goldman JM, Wilson B, Himes G (2007) *Medical Device Interoperability – Assessing the Environment*. *Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability*, 25.–27.6.2007, Boston, MA
23. Liebhart D (2007) *SOA goes real: Service-orientierte Architekturen erfolgreich planen und einführen*. Hanser, München
24. Lublinsky B (2004) *SOA Design: Meet in the Middle*. Java PRO
25. Mauro C, Sunyaev A, Dünnebeil S, Leimeister JM, Krcmar H (2009) *Mobile Anwendungen im Kontext des Medizinproduktegesetzes*. In: Fischer S, Maehle E, Reichschuk R (Hrsg) *Informatik 2009 – Im Fokus das Leben*. Köllen, Lübeck
26. Mauro C, Sunyaev A, Leimeister JM, Krcmar H (2009) *Service-orientierte Integration medizinischer Geräte – eine State of the Art Analyse*. In: *Proceedings of Wirtschaftsinformatik 2009 – Business Services: Konzepte, Technologien und Anwendungen*. Wien, S 119–128
27. Mauro C, Leimeister JM, Krcmar H (2010) *Service Oriented Device Integration – An Analysis of SOA Design Patterns*. *43st Annual Hawaii International Conference on System Sciences (HICSS-43) Kauai, Hawaii*, 2010
28. Mauro C, Sunyaev A, Leimeister JM, Krcmar H (2010) *Standardized Device Services – A Design Pattern for Service Oriented Integration of Medical Devices*. *43st Annual Hawaii International Conference on System Sciences (HICSS-43) Kauai, Hawaii*, 2010
29. Melrose JP (2007) e-health is the way via SOA. *Healthc Financial Manag* 61:120–122
30. Naranjo JC, Fernandez C, Pomes S, Valdivieso B (2006) *Care-Paths: Searching the way to implement pathways*. In: *Proceedings of Computers in Cardiology*, Valencia, pp 285–288
31. Pöhlsen S, Schlichting S, Strähle M, Franz F, Werner C (2009) *A Concept for a Medical Device Plug-and-Play Architecture based on Web Services*. In: *Proceedings of 2nd Joint Workshop on High-Confidence Medical Devices, Software and Systems (HCMDSS) and Medical Device Plug-and-Play Interoperability (MD PnP)*, San Francisco, 2009, pp 52–65
32. Schmitz A (2008) *SOA schafft Gerechtigkeit: Marktbeherrschende Systeme geraten ins Abseits*. <http://www.cio.de/healthcareit/aktuelles/857755/>, letzter Zugriff 17.3.2010
33. Schweiger A, Sunyaev A, Leimeister JM, Krcmar H (2007) *Toward seamless healthcare with software agents*. *Commun Assoc Inf Sys* 19:692–709
34. Shaikh A, Memon M, Memon N, Misbahuddin M (2009) *The Role of Service Oriented Architecture in Telemedicine Healthcare System*. *Proceedings of International Conference on Complex, Intelligent and Software Intensive Systems, CISIS '09*, 16.–19.3.2009, Fukuoka, Japan, S 208–214
35. Sunyaev A, Leimeister JM, Schweiger A, Krcmar H (2006) *Integrationsarchitekturen für das Krankenhaus – Status quo und Zukunftsperspektiven*. *Inf Manag Consult* 21:28–35
36. Sunyaev A, Leimeister JM, Schweiger A, Krcmar H (2008) *IT-Standards and Standardization Approaches in Healthcare*. In: *Wickramasinghe N, Geisler E (Hrsg) Encyclopedia of Healthcare Information Systems*. Idea Group, pp 813–820
37. Thun S (2009) *Integrierte Behandlungspfade*. In: Johnner C, Haas P (Hrsg) *IT im Gesundheitswesen: Erfolgreich einführen, entwickeln, anwenden und betreiben*. Carl Hanser, München, S 179–188