

Cyber-physische Systeme im OP-Saal - Ein Machbarkeitsnachweis -

Martin Kasparick, Frank Golatowski, Dirk Timmermann

Institut für Angewandte Mikroelektronik und Datentechnik
Fakultät für Informatik und Elektrotechnik
Universität Rostock
Richard-Wagner-Straße 31
18119 Rostock
martin.kasparick@uni-rostock.de
frank.golatowski@uni-rostock.de
dirk.timmermann@uni-rostock.de

Abstract: In einem OP-Saal befinden sich eine große Anzahl von Medizingeräten und Informationssystemen. Eine herstellerübergreifende Kommunikation bzw. Interoperabilität ist derzeit kaum bzw. gar nicht möglich. Dieses Problem adressiert die vorliegende Arbeit. Es wird von einer serviceorientierten Kommunikationsarchitektur ausgegangen und gezeigt, wie heutige Medizingeräte ohne entsprechende Hard- und/oder Software-Schnittstellen integriert werden können. Anhand einer prototypischen Umsetzung eines Teils eines endoskopischen Arbeitsplatzes wird die Machbarkeit des vorgestellten Ansatzes gezeigt. Ausgehend von derart vernetzten cyber-physikalischen Systemen (CPS) im OP-Saal ergeben sich erstmals Möglichkeiten der Big-Data-Analysen der anfallenden Daten, da diese nicht mehr lokal auf die Geräte beschränkt sind, sondern global verknüpft werden können.

1 Einleitung

In einem heutigen OP-Saal befinden sich eine Vielzahl medizinischer Geräte zur Überwachung und Beeinflussung der Vitalfunktionen, zur Durchführung des operativen Eingriffs sowie Systeme zur Dokumentation und Einsichtnahme in Patientendaten etc. Nach heutigem Stand agieren diese Geräte nicht vernetzt und somit ohne Interaktion und Datenaustausch untereinander. Lediglich Geräte desselben Herstellers nutzen teilweise In-sellösungen; eine herstellerübergreifende Kommunikation ist aufgrund unterschiedlicher Hardwareschnittstellen und proprietärer Protokolle nicht möglich. In der vorliegenden Arbeit wird gezeigt, dass das beschriebene Problem mit einer SOA-basierten Architektur gelöst werden kann.

Cyber-Physical Systems (CPS) werden nach Lee und Seshia [LS11] als die Integration der Berechnungen (computation) in die physikalischen Prozesse definiert. Dabei überwachen und steuern eingebettete Computer und Netzwerke die physikalischen Prozesse. Dies geschieht typischerweise mittels Regelkreisen, wobei die physikalischen Prozesse die Be-

rechnungen beeinflussen und umgekehrt. Für ein solches System ist es nicht ausreichend die physikalischen und rechnenden Komponenten einzeln zu betrachten, sondern deren Zusammenspiel. Ein moderner Operationssaal stellt demnach ein solches CPS dar. Lee et al. [LSC⁺12] betrachten Medical Cyber-Physical Systems (MCPSs) sogar als eigene Klasse von CPSs. Die Begründung liegt in der Kombination von eingebetteter Software zur Steuerung der Geräte, der (zukünftigen) Netzwerkfähigkeit der Geräte, der komplexen physikalischen Welt und den Auswirkungen auf die Patientensicherheit. Die Komplexität bezieht sich hierbei vor allem auf den menschlichen Körper als physikalische Welt.

Auf den einzelnen Geräten im OP-Saal fällt eine große Menge von Daten an. Bezogen auf das Big Data 3-V-Modell (Data Volume, Data Velocity und Data Variety) [Lan01] [Bey11] stellt das OP-Umfeld vor allem eine Herausforderung in Bezug auf die Geschwindigkeit (Velocity) und Mannigfaltigkeit (Variety) der Daten dar. Für viele Aspekte, wie z. B. die Gerätesteuerung, müssen Echtzeitanforderungen eingehalten werden. Außerdem unterscheiden sich die anfallenden Daten von Geräte- über Vitalparameter bis hin zu Patienten-, Befund- und Bilddaten stark. Aus Sicht der Big Data Analysen besteht ein enormes Potential zur Steuerung der medizinischen Geräte an sich sowie etwa im Hinblick auf die Qualitätssicherung oder die Weiterentwicklung der medizinischen Methoden und Vorgehensweisen. Dies kann aber nur erschlossen werden, wenn eine Interoperabilität der Medizinprodukte untereinander, bis hin zu den Krankenhausinformationssystemen hergestellt werden kann. Die vorliegende Arbeit thematisiert die Herausforderung, einen solchen interoperablen Datenaustausch zu ermöglichen. Zur Realisierung werden neue Methoden der Gerätekommunikation zum Einsatz kommen. Bedingt durch die langen Lebenszyklen von Medizinprodukten ist es zwingend notwendig, die heutigen Systeme in die neue Infrastruktur einzubinden. Daher liegt ein Hauptfokus dieser Arbeit auf dem Thema der Integration von nicht SOA-kompatiblen Systemen. Die Integration wird anhand eines Teils eines medizinischen Endoskopie-Arbeitsplatzes gezeigt.

Die vorliegende Arbeit ist folgendermaßen gegliedert: Zunächst wird auf einige Grundlagen und den Stand der Technik eingegangen. Es folgt eine Beschreibung der SOA-basierten Gesamtarchitektur. In Kapitel 4 wird das Einbinden von nicht DPWS-kompatiblen Systemen beschrieben. Danach wird die prototypische Umsetzung dargestellt. Zum Abschluss werden eine Zusammenfassung und ein Ausblick gegeben.

2 Grundlagen und Stand der Technik

2.1 DPWS

Um Web Services auf eingebetteten Systemen mit beschränkten Ressourcen nutzen zu können wurde das Geräteprofil DPWS (Devices Profile for Web Services) [OAS09] entwickelt. DPWS setzt auf SOAP (Simple Object Access Protocol) auf und stellt eine Untermenge der verfügbaren Web Service Standards dar.

Auf dem Gerät, welches gesteuert werden soll bzw. welches Daten zur Verfügung stellen soll, wird die DPWS-Server-Seite implementiert. Auf der Steuereinheit bzw. Datensinke

wird die DPWS-Client-Seite implementiert. Bei DPWS ist vor allem die serverseitige Implementierung ressourcenschonend. Soll die Kommunikation zwischen den eingebetteten Geräten erfolgen, müssen beide Seiten implementiert werden.

DPWS nutzt den Standard WS-Discovery [OAS09] zum Auffinden von Geräten (Discovery). WS-MetadataExchange und WS-Transfer werden zum XML-basierten Austausch von Geräte- und Servicebeschreibungen genutzt. Zum Suchen von Diensten senden Clients sogenannte Probe-Nachrichten per UDP-Multicast. Die Geräte, die einen entsprechenden Dienst anbieten, antworten mit einem Probe Match via Unicast. Zur Auflösung der Geräteadresse, die als Teil der Probe Match-Nachricht übermittelt wird, werden Resolve-Nachrichten versandt. Die entsprechende Resolve Match-Nachricht enthält eine gültige Transportadresse. Zusätzlich zum beschriebenen Verfahren ist im DPWS-Standard vorgesehen, dass sich neue Geräte mit einer sogenannten Hello-Nachricht bekannt machen und mit einer Bye-Nachricht abmelden. Beides erfolgt per UDP-Multicast.

Für die Funktionalität des Abonnierens und Versendens von Ereignissen (Events) wird der Standard WS-Eventing genutzt. Dieser ermöglicht es, dass ein Client vom Server Benachrichtigungen u. a. über Werteänderungen erhält, ohne ein aktives Polling zu betreiben.

2.2 DPWS im OP-Saal

Eine Analyse des Standes der Technik präsentieren Mauro et al. in [MSLK09]. Forschungslücken im Bereich der SOA-Integration medizinischer Geräte werden ausgehend von anderen Anwendungsgebieten, z. B. der industriellen Automatisierung, identifiziert.

Verschiedene Projekte beschäftigen sich mit der Nutzung von gerätenaher SOA im OP. Im Zuge der Projekte smartOR [iMuB] und orthoMIT [BORN09] wurde ein SOA-basiertes Framework für den OP entwickelt. Dieses wird u. a. von Ibach et al. [IBR10] [IBS⁺12] und Benzko et al. [BINR10] beschrieben. Es kommen standardisierte Protokolle und Interfacebeschreibungen zum Einsatz. Als Basistechnologie wird DPWS eingesetzt.

Ein medizinisches Geräteprofil auf Basis von DPWS wird von Pöhlsen et al. [PSS⁺09] vorgeschlagen. Wesentlich sind hierbei das Geräteprofil MDPWS (Medical Devices Profile for Web Services) und das klinische Integrationsprofil BICEPS (Basic Integrated Clinical Environment Protocol Specification) [Sch13]. Die technischen Hintergründe werden im Whitepaper von Schlichting und Pöhlsen [SP14b] ausführlich dargestellt. Es erfolgte eine Einreichung als Standardisierungsvorschlag [SP14a] bei den 11073/HL7 Standard Weeks.

Im Zuge des DOOP-Projekts [doo] (Dienst-orientierte OP-Integration) wurden Lösungsstrategien für eine Plug-and-Play-artige Medizingerätevernetzung vorgestellt. Im Fokus steht das SOA-Paradigma zur Integration von Medizingeräten im OP-Saal und auf der Intensivstation. [GBM12]

Die beschriebenen (Vor-)Projekte konnten den Stand der Technik nicht soweit vorantreiben, dass eine herstellerunabhängige Gerätevernetzung in heutigen OP-Sälen realisiert werden kann. Daher wurde das BMBF geförderte Projekt „OR.NET - Sichere dynamische

sche Vernetzung in Operationssaal und Klinik“¹ [orn12] [KBB13] initiiert, das Partner aus den Bereichen der Wissenschaft, Industrie und Standardisierung vereint. Das Ziel ist eine standardisierte, herstellerübergreifende und sichere Interoperabilität der Medizinprodukte und IT-Systeme. Außerdem werden Themen, wie die sichere Bedienung durch die Optimierung der MMI (Mensch-Maschine-Interaktionen) sowie Zulassungsfragen bearbeitet. Die Gerätevernetzung erfolgt nach dem Prinzip einer SOA mittels DPWS. [orn14]

2.3 SOA-Integration von nicht SOA-kompatiblen Systemen

Mit dem Pattern des „Legacy Wrappers“ zeigt Erl [ER09] konzeptionell die Integration von Legacy-Systemen in eine SOA. Das Gerät wird als Service gekapselt, sodass es für den Servicekonsumenten transparent ist und Daten ausschließlich über die entsprechenden Services ausgetauscht werden können. Abbildung 1 verdeutlicht die Funktionsweise des Patterns. Durch die enge Kopplung der Servicelogik sowohl zur Legacy-API, also auch zum Servicevertrag kann eine Kopplung des Servicenutzers und der Implementierung vermieden werden. So kann das SOA-Grundprinzip der losen Kopplung umgesetzt werden.

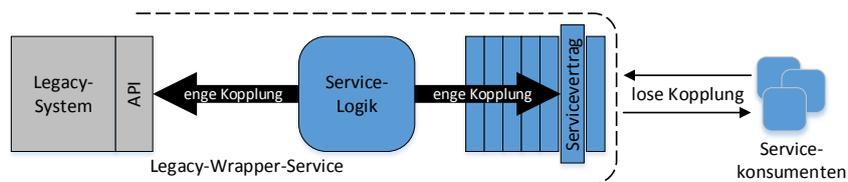


Abbildung 1: Legacy-Wrapper-Service nach Erl [ER09] (Seite 442).

Mit der Service-Oriented Device Architecture (SODA) stellen de Deugd et al. [dDCK⁺06] ein System zur serviceorientierten Kapselung von Geräten vor. Ein Gerät wird mit zwei verschiedenen Adaptern ausgestattet. Der Device-Adapter realisiert die Kommunikation zu den (proprietären) Geräte-Schnittstellen und Protokollen. Auf der anderen Seite enthält der Device-Adapter eine Geräte und Szenario spezifische Logik, um ein abstraktes Servicemodell des Gerätes zur Verfügung zu stellen. Der Bus-Adapter stellt die Verbindung zur SOA her, in dem das abstrakte Servicemodell in spezifische SOA-Binding-Mechanismen überführt wird.

3 Gesamtarchitektur

Die Architektur des Gesamtsystems muss die Kommunikation sowohl medizinischer Geräte untereinander als auch mit den Informationssystemen des Krankenhauses² ermög-

¹Die vorliegende Arbeit entstand im Zuge des OR.NET-Projekts.

²wie z. B. KIS (Krankenhausinformationssystem) oder PACS (Picture Archiving and Communication System)

lichen. Die Notwendigkeit zur Integration der heute im Betrieb befindlichen Geräte ergibt sich aus den langen Lebenszyklen und hohen Kosten von Medizinprodukten.³ Daher ist die möglichst einfache und kostengünstige Integration eine Grundvoraussetzung für eine Marktakzeptanz der neuartigen Kommunikationsprinzipien. Die Abbildung 2 zeigt einen (vereinfachten) Architekturentwurf, wie er im OR.NET-Projekt verfolgt wird. Die Kommunikation erfolgt nach den Prinzipien einer Serviceorientierten Architektur (SOA) und nutzt hierfür die konkrete Umsetzung des Devices Profile for Webservices (DPWS). Die Webservicekommunikation wird sowohl zwischen den einzelnen medizinischen Geräten als auch zum sogenannten Konnektor IS⁴ genutzt.

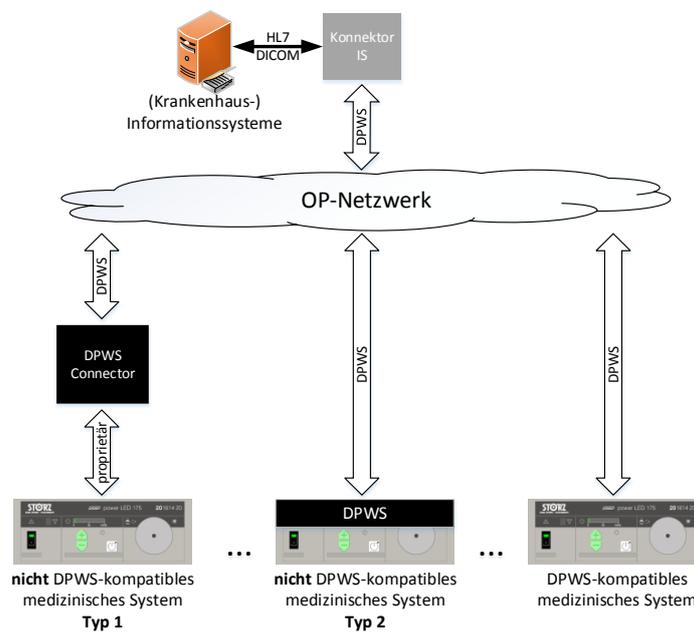


Abbildung 2: Gesamtarchitektur der DPWS-basierten Vernetzung von OP-Geräten. (Vereinfachte Darstellung des Architekturentwurfs im OR.NET-Projekt [orn12].)

Die im unteren Bereich von Abbildung 2 dargestellten medizinischen Systeme repräsentieren die Geräte, die sich im OP-Saal befinden. Die Unterscheidung zwischen DPWS-kompatiblen und nicht DPWS-kompatiblen Systemen ergibt sich aus der Art der Einbindung der Geräte in den DPWS-basierten Kommunikationsverbund. Ein nicht DPWS-kompatibles System vom Typ 1 verfügt weder über die entsprechende Hardwareschnittstelle noch über einen DPWS-Stack. Daher wird zur Einbindung zusätzliche Hardware benötigt. Der Typ 2 von nicht DPWS-kompatiblen Systemen verfügt bereits über eine

³Die im Zusammenhang mit der Integration von Systemen in eine neue Umgebung gebräuchliche Bezeichnung des „Legacy Systems“ ist ggf. irreführend. Der Begriff „Legacy“ bezieht sich nicht auf die medizinische Hauptfunktionalität der Geräte, sondern lediglich auf ihre Kommunikationsfähigkeit. Daher werden im Verlauf der Arbeit meist Bezeichnungen wie „nicht SOA- / DPWS-kompatibles System“ genutzt.

⁴IS wird als Abkürzung für Informationssystem genutzt.

IP-basierte Netzwerkschnittstelle, sodass lediglich die Firmware erweitert werden muss. In Abschnitt 4 dieser Arbeit wird detailliert auf die Einbindung von derartigen Systemen eingegangen. Als DPWS-kompatible Systeme werden solche bezeichnet, die zum Zeitpunkt der Erstinbetriebnahme bereits hardware- und softwareseitig in der Lage sind an der DPWS-Kommunikation teilzunehmen.

Der Konnektor IS bildet die Schnittstelle zu den Informationssystemen des Krankenhauses. Zwischen dem Konnektor IS und den Informationssystemen werden typischerweise standardisierte Protokolle wie HL7 oder DICOM genutzt. Durch die direkte Anbindung der medizinischen Geräte an die Informationssysteme können große Mengen von Daten persistent gespeichert werden. Außerdem steht ausreichend Rechenleistung zur Verfügung, um Methoden der Big Data für Analysen dieser Daten zu nutzen.

Aspekte von Security und Safety sollen nicht Gegenstand der vorliegenden Arbeit sein. Mechanismen zur Zuordnung von Geräten zu Patienten bzw. medizinischen Eingriffen, Authentifizierung und Autorisierung etc., die sicherstellen, dass der Informationsaustausch von Patienten- und Steuerdaten nur zwischen den „richtigen“ Geräten möglich ist, sind in der Architektur enthalten.

4 Einbindung von nicht DPWS-kompatiblen Systemen

In der vorliegenden Arbeit wird ein Blackbox-Ansatz genutzt, um die Lücke zwischen den existierenden OP-Geräten mit ihren unterschiedlichen (Hardware-) Schnittstellen und proprietären Protokollen und der zukünftig standardisierten webservicebasierten Kommunikation zu schließen. Für nicht DPWS-kompatible Systeme vom Typ 1 (siehe Abschnitt 3) stellt die Blackbox ein System aus Hard- und Softwarekomponenten dar. Im Fall eines Typ 2 Systems reduziert sich die Komplexität entsprechend auf Softwarekomponenten. Im Folgenden wird vor allem auf die Umsetzung für Typ 1 Systeme eingegangen. Die softwareseitige Umsetzung des vorgestellten Systems kann aber analog auf Systeme des zweiten Typs übertragen werden.

Hardwareseitig muss sowohl eine Schnittstelle zum bestehenden herstellerspezifischen System zur Verfügung gestellt werden, als auch eine IP-basierte Schnittstelle um an der Webservice-Kommunikation teilnehmen zu können.

Die Blackbox-Software besteht aus zwei wesentlichen Teilkomponenten: Legacy-Adapter und DPWS-Adapter. Der Legacy-Adapter implementiert einerseits die Schnittstellen und Protokolle des einzubindenden Gerätes, andererseits die geräte- und anwendungsspezifische Logik (vergleiche auch de Deugd et al. [dDCK⁺06]). Des Weiteren kann eine optionale Abstraktionsschicht als Subkomponente integriert werden, die eine Kapselung von als schützenswert eingestuften firmenspezifischen Lösungen vornimmt. Diese Abstraktionsschicht kann sowohl oberhalb, als auch unterhalb der Subkomponente erfolgen, die die spezifische Anwendungslogik enthält. Der DPWS-Adapter (nach de Deugd et al. auch Bus-Adapter) realisiert die Einbindung des Systems in die mittels DPWS realisierte Serviceorientierte Architektur. Abbildung 3 illustriert schematisch den Aufbau der Blackbox.

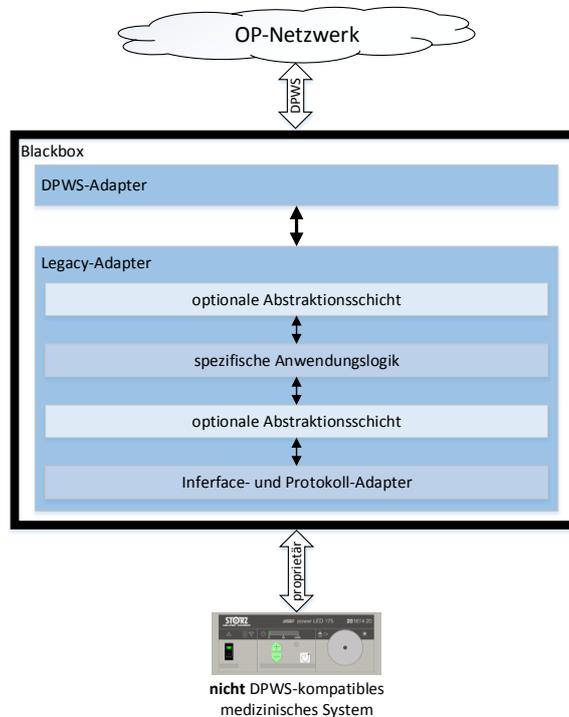


Abbildung 3: Modell einer Blackbox zur Integration von nicht DPWS-kompatiblen Systemen.

5 Prototypische Umsetzung

Die beschriebene Blackbox-Architektur zum Einbinden von nicht DPWS-kompatiblen Systemen wurde in einem prototypischen Demonstrator evaluiert. Abbildung 4 illustriert den Gesamtaufbau, der im Folgenden beschrieben wird. Eine schematische Darstellung zeigt Abbildung 5.

Im rechten Teil der Abbildung 4, gekennzeichnet mit a_1 - a_3), befinden sich zwei Geräte eines endoskopischen Arbeitsplatzes, die mit einem DPWS-Konnektor ausgestattet sind. Dies ist zum einen eine endoskopische Kamera (Karl Storz Image 1), zum anderen eine dazugehörige Kaltlichtquelle (Karl Storz PowerLED 175). Für die Kamera ist ein Service implementiert, der es ermöglicht ein Standbild abzurufen. Die Kaltlichtquelle stellt Services zum Auslesen der aktuellen Lichtintensität in Prozent sowie zum Erhöhen und Verringern der Intensität bereit. Weiterhin kann ein Eventing-Service abonniert werden, der bei einer Änderung des Intensitätswertes ausgelöst wird. Die Blackbox zur Integration dieser beiden Systeme in die DPWS-basierte Kommunikation wird inklusive der beschriebenen Abstraktionsschicht realisiert, die firmenspezifisches Know-how effektiv kapselt. Daher wird der geräteseitige Teil des DPWS-Konnektors (Hardware und die Softwarekom-

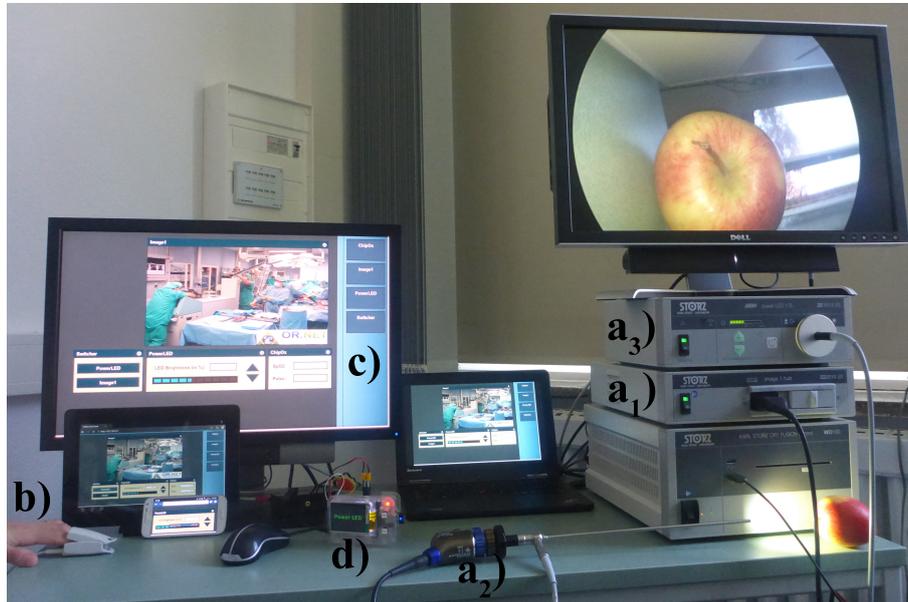


Abbildung 4: Demonstration einer prototypischen Umsetzung der DPWS-Integration von medizinischen Geräten. a₁) Endoskopkamera mit a₂) Kamerakopf und a₃) Kaltlichtquelle; b) Pulsoximeter-Fingerclip; c) Dashboard; d) Eingabeschalter.

ponente Legacy-Adapter) nicht beschrieben. Die DPWS-Adapter Softwarekomponente ist auf Basis des JMEDS⁵ [Mat13] DPWS-Stacks realisiert.

Weiterer Bestandteil des Demonstrators ist ein Pulsoximeter (corscience ChipOx), dessen Fingerclip in der linken unteren Ecke von Abbildung 4, mit der Markierung b), zu sehen ist. Ein Pulsoximeter ermittelt die Vitalparameter Puls und Sauerstoffsättigung. Für dieses Gerät fungiert ein Raspberry Pi Board als DPWS-Konnektor Blackbox. Die geräteseitige Schnittstelle wird über den UART des Boards realisiert. Die Softwarekomponenten Legacy- und DPWS-Adapter sind in Java implementiert. Es wird ebenfalls der JMEDS DPWS-Stack genutzt. Es sind Services zum manuellen Lesen der aktuellen Werte von Puls und Sauerstoffsättigung realisiert, sowie Eventing-Services, die über die Änderungen von Puls bzw. Sauerstoffsättigung informieren.

Alle beschriebenen Geräte stellen außerdem einen für Medizingeräte üblichen Heartbeat zur Verfügung. Dieser informiert periodisch über die Präsenz der Geräte, sodass Ausfälle nach einer definierten Zeitschranke erkannt werden können. Die Realisierung erfolgt mithilfe von Eventing-Services.

Zur Anzeige und Steuerung der Geräte- und Vitaldaten dient ein Dashboard, das in Abbildung 4 mit c) hervorgehoben ist. Dieses Mensch-Maschine-Interface ist webbasiert umgesetzt, sodass es von beliebigen Geräten mit einem Webbrowser angezeigt und genutzt wer-

⁵Java Multi Edition DPWS Stack

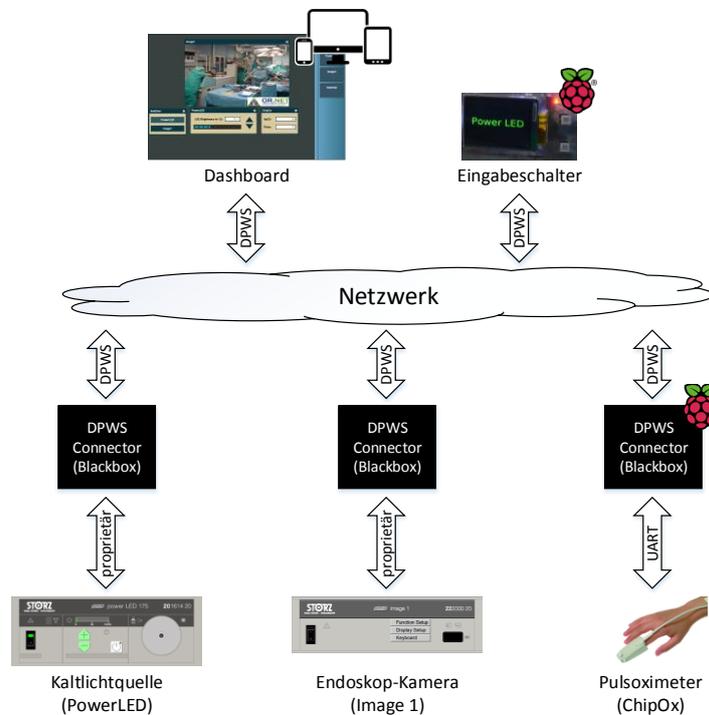


Abbildung 5: Schematische Darstellung des prototypischen Demonstrators.

den kann. Wie im mittleren und linken Teil von Abbildung 4 zu erkennen ist, werden exemplarisch ein Standard-PC, Laptop, Android-Tablet und -Smartphone genutzt. Der Webserver, der das Dashboard bereitstellt, implementiert die DPWS-Client-Seite. Es können so die beschriebenen Services genutzt werden. Als weiteres Steuergerät fungiert ein Raspberry Pi basierter Schalter mit zwei Tasten, der ebenfalls an der DPWS-Kommunikation teilnimmt. Dieser ist in Abbildung 4 mit d) gekennzeichnet. Über das Dashboard kann entschieden werden, ob der Schalter die Intensität der Kaltlichtquelle erhöht bzw. verringert oder ob durch einen Tastendruck ein Standbild von der Endoskopkamera abgeholt und auf dem Dashboard angezeigt werden soll.

Der vorgestellte Prototyp demonstriert wichtige Aspekte des zukünftigen OP-Saales: Einerseits wird eine herstellerunabhängige Vernetzung gezeigt. So erfolgt z. B. die Steuerung der Kaltlichtquelle wahlweise über einen auf Raspberry Pi Basis realisierten Schalter oder über das webbasierte Dashboard, sodass beliebige Geräte genutzt werden können. Aus diesem Einsatz von Standardhardware kann abgeleitet werden, dass eine Kommunikation zwischen spezialisierten Herstellern in der DPWS-basierten Umgebung keine Herausforderung mehr darstellen wird. Ein weiterer Aspekt ist die dynamische Vernetzung. Durch die Möglichkeit mit einer Mensch-Maschine-Schnittstelle mehrere Geräte zu steuern, kann die Anzahl der Eingabegeräte reduziert werden. Dies führt zu einer geringeren

Kabelanzahl im OP, sodass beispielsweise das Risiko des Stolperns oder hygienische Probleme reduziert werden können.

Des Weiteren ermöglicht das vorgestellte Konzept des Dashboards neue Steuerungskonzepte und Möglichkeiten im Bereich der Telemedizin. In einem realen OP-Saal können deutlich mehr Daten und Parameter angezeigt und verändert werden, als im Prototyp gezeigt. Jeder beteiligte Akteur an einer Operation kann genau jene Informationen angezeigt bekommen, die benötigt werden. Diese können sich stark unterscheiden, beispielsweise zwischen Chirurgen und Anästhesisten. Durch die Zusammenführung aller Daten, die in vollen Umfang an Spezialisten an anderen Orten weitergegeben werden können, wird der Aspekt der intraoperativen Telemedizin neue Impulse erfahren.

6 Zusammenfassung und Ausblick

Die vorliegende Machbarkeitsstudie stellt einen Blackbox-Ansatz zur Integration von nicht DPWS-kompatiblen (medizinischen) Systemen in eine serviceorientierte Kommunikationsumgebung vor. Dies ermöglicht eine herstellerunabhängige und dynamische Vernetzung. Die Blackbox beinhaltet einen Legacy-Adapter, der die proprietäre Schnittstelle zum Gerät realisiert und einen DPWS-Adapter, der die Web-Service-Kommunikation implementiert. In einem prototypischen Demonstrator konnte die Funktionsfähigkeit unter Beweis gestellt werden.

Zukünftige Arbeiten werden sich mit der Verbesserung der Echtzeit-Eigenschaften der DPWS-Kommunikation beschäftigen. Dies ist sowohl für die Steuerung der medizinischen Geräte notwendig, als auch von Relevanz für den Big Data Aspekt der Velocity (Geschwindigkeit). In Bezug auf die semantische Interoperabilität wird mithilfe der Normenfamilie ISO/IEEE 11073 gearbeitet. Diese ist vor allem für den Bereich der Personal-Health-Geräte definiert. OP-Geräte sind derzeit nicht berücksichtigt, sodass eine Erweiterung angestrebt wird. Ebenso werden die Bestrebungen der Standardisierung des Kommunikationsprotokolles vorangetrieben und Aspekte der Zertifizierbarkeit von Medizinprodukten mit SOA-basierter Kommunikation betrachtet.

In einem OP-Saal fallen eine große Menge Daten an, die nur mit Hilfe einer (herstellerunabhängigen) Vernetzung genutzt werden kann. Mit Big Data Methoden können diese gezielt analysiert werden. So könnten beispielsweise relevante Daten für die Dokumentation der Operation gefiltert werden, um den Ärzten und dem OP-Personal die Arbeit zu erleichtern. Auch Aspekte der Qualitätssicherung und Optimierung der Abläufe sind relevante Fragestellungen, bis hin zur Weiterentwicklung der medizinischen Forschung an sich.

Acknowledgement

Diese Arbeit ist im Projekt OR.NET entstanden. OR.NET wird unter dem Förderkennzeichen 16KT1238 (im Förderprogramm IKT 2020) durch das Bundesministerium für Bildung und Forschung (BMBF) gefördert. Der prototypische Demonstrator wurde mit Unterstützung der Firma KARL STORZ GmbH & Co. KG (Tuttlingen, Deutschland) entwickelt. Weiterer Dank gilt Priv.-Doz. Dr. med. Klaus F. Wagner (Chefarzt der Klinik für Anästhesiologie und Intensivmedizin des Klinikums Südstadt in Rostock) für die Beratung in medizinisch technischen Fragen.

Literatur

- [Bey11] M. Beyer. Gartner Says Solving „Big Data“ Challenge Involves More Than Just Managing Volumes of Data, 2011. URL: <http://www.gartner.com/newsroom/id/1731916>. Datum: 18.05.2014.
- [BINR10] J. Benzko, B. Ibach, M. Niggemeyer, K. Radermacher. A novel SOA based approach towards the integration of a robotic system into a modular surgical work system and IT network. In *Computer assisted radiology and surgery : proceedings of the 24rd international congress and exhibition*, Seiten 193–194, Geneva, Switzerland, 2010. Heinz U. Lemke. Heidelberg : Springer, 2010.
- [BORN09] C. Buschmann, J. A. K. Ohnsorge, K. Radermacher, F. U. Niethard. Das Projekt orthoMIT. *Bundesgesundheitsblatt - Gesundheitsforschung - Gesundheitsschutz*, 52(3):287–296, 2009. doi: 10.1007/s00103-009-0790-z. URL: <http://dx.doi.org/10.1007/s00103-009-0790-z>.
- [dDCK⁺06] S. de Deugd, R. Carroll, K. E. Kelly, B. Millett, J. Ricker. SODA: Service Oriented Device Architecture. *Pervasive Computing, IEEE*, 5(3):94–96, 2006. doi: 10.1109/MPRV.2006.59.
- [doo] DOOP-Projekt (Dienst-orientierte OP-Integration). URL: <http://www.doop-projekt.de/>. Datum: 22.05.2014.
- [ER09] T. Erl, S. Roy. *SOA Design Patterns*. Prentice Hall, 2009. doi: 10.1016/j.artmed.2009.05.004.
- [GBM12] D. Gregorczyk, T. Bußhaus, R. Mildner. Strategien zur Konstruktion von Medizingeräteprofilen. *White Paper*, 2012. URL: http://www.doop-projekt.de/tl_files/ebooks/medizingeraeteprofile/files/assets/downloads/publication.pdf.
- [IBR10] B. Ibach, J. Benzko, K. Radermacher. OR-Integration based on SOA - Automatic detection of new Service Providers using DPWS. In *Proceedings of the 24rd international Congress and Exhibition: Computer assisted Radiology and Surgery (CARS)*, Seiten 195–196, Geneva, Switzerland, 2010. Heinz U. Lemke. Heidelberg : Springer, 2010.
- [IBS⁺12] B. Ibach, J. Benzko, S. Schlichting, A. Zimolong, K. Radermacher. Integrating medical devices in the operating room using service-oriented architectures. *Biomedizinische Technik. Biomedical engineering*, 57(4):221–228, August 2012. doi: 10.1515/bmt-2011-0101. URL: <http://www.degruyter.com/view/j/bmte.2012.57.issue-4/bmt-2011-0101/bmt-2011-0101.xml>.

- [iMuB] DGBMT Innovationen in Medizintechnik und BioEngineering. DGBMT - Innovationen in Medizintechnik und BioEngineering - smartOR. URL: <http://www.vde.com/de/fg/DGBMT/Arbeitsgebiete/Projekte/Seiten/Medizintechnik-Biomedizintechnik-Wissenschaft-Forschung-Studium-Innovationen-smartOR.aspx>.
- [KBB13] C. Kücherer, J. Benzko, T. Bußhaus. Forschungsprojekt OR.NET: Abschied von Insellösungen. *Deutsches Ärzteblatt* 2013; 110(46), 4/2013, 2013. URL: <http://www.aerzteblatt.de/pdf.asp?id=149231>.
- [Lan01] D. Laney. 3D data management: Controlling data volume, velocity and variety. *Application Delivery Strategies - META Group Research Note*, 6, 2001.
- [LS11] E. A. Lee, S. A. Seshia. *Introduction to Embedded Systems - A Cyber-Physical Systems Approach*. LeeSeshia.org, first edition (version 1.08). Auflage, 2011. URL: <http://leeseshia.org>.
- [LSC⁺12] I. Lee, O. Sokolsky, S. Chen, J. Hatcliff, E. Jee, B. Kim, A. King, M. Mullen-Fortino, S. Park, A. Roederer, K. K. Venkatasubramanian. Challenges and Research Directions in Medical Cyber-Physical Systems. *Proceedings of the IEEE*, 100(1):75–90, Januar 2012. doi: 10.1109/JPROC.2011.2165270.
- [Mat13] Materna GmbH. JMEDS (Java Multi Edition DPWS Stack) — Free Development software downloads at SourceForge.net, 2013. URL: <http://sourceforge.net/projects/ws4d-javame/>.
- [MSLK09] C. Mauro, A. Sunyaev, J. M. Leimeister, H. Krcmar. Service-orientierte Integration medizinischer Geräte - eine State of the Art Analyse. *Proceedings of Wirtschaftsinformatik 2009 (WI 2009), Business Services: Konzepte, Technologien, Anwendungen*, Band 1:119–128, 2009.
- [OAS09] OASIS. Devices Profile for Web Services Version 1.1 - OASIS Standard, 2009. URL: <http://docs.oasis-open.org/ws-dd/dpws/1.1/os/wsdd-dpws-1.1-spec-os.html>. Datum: 18.05.2014.
- [orn12] Homepage: OR.NET - Sichere dynamische Vernetzung in Operationssaal und Klinik, 2012. URL: www.ornet.org. Datum: 20.05.2014.
- [orn14] Broschüre: OR.NET - Sichere dynamische Vernetzung in Operationssaal und Klinik, 2014. URL: http://www.ornet.org/wp-content/uploads/2014/04/Broschüre_final_2.0.pdf. Datum: 22.05.2014.
- [PSS⁺09] S. Pöhlens, S. Schlichting, M. Strähle, F. Franz, C. Werner. A Concept for a Medical Device Plug-and-play Architecture Based on Web Services. *SIG-BED Rev.*, 6(2):6:1—6:7, 2009. doi: 10.1145/1859823.1859829. URL: <http://doi.acm.org/10.1145/1859823.1859829>.
- [Sch13] S. Schlichting. BICEPS - Ein Protokoll zur Integration von Medizinprodukten am klinischen Arbeitsplatz. In *GMDS 2013. 58. Jahrestagung der Deutschen Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie e.V. (GMDS)*, Lübeck, Germany, 2013.
- [SP14a] S. Schlichting, S. Pöhlens. An architecture for distributed systems of medical devices in high acuity environments - A Proposal for Standards Adoption. In *HL7, 2014. 11073/HL7 Standards Week*, San Antonio, Texas, USA, 2014.
- [SP14b] S. Schlichting, S. Pöhlens. An architecture for distributed systems of medical devices in high acuity environments - A technical whitepaper. *White Paper*, 2014. URL: http://sourceforge.net/projects/opensdc/files/framework/1.0-BETA_02/Documentation/TechnicalWhitepaper.pdf/download.