Infrastrukturelle Interoperabilität für pervasive Lehr- und Lernarrangements

Raphael Zender

Institut für Informatik
Universität Potsdam
August-Bebel-Str. 89
14482 Potsdam
zender@uni-potsdam.de

Abstract: Mit der wachsenden Bedeutung von Mobilität, allgegenwärtigem IT-Zugang und dynamischer Allokation anpassbarer Inhalte ist eine zunehmende Integration von Technologien des Pervasive Computing in Lehr- und Lernprozesse erkennbar. Dieser Artikel analysiert die infrastrukturellen Herausforderungen für pervasive Umgebungen im Allgemeinen und pervasive Hochschulen im Speziellen. Er bewertet basierend auf dieser Analyse die Eignung heutiger Kommunikationsmodelle als infrastrukturelle Basis und motiviert als Ergebnis dieses Vergleichs den Einsatz Service-orientierter Architekturen (SOA). Zur erforderlichen Überbrückung der SOA-Heterogenität werden existierende Interoperabilitätsansätze vorgestellt und hinsichtlich ihrer Eignung zur Interoperabilität in pervasiven Umgebungen bewertet. Auf der Grundlage dieser Bewertung wurde ein systematischer Interoperabilitätsansatz konzipiert und beispielhaft umgesetzt. Die erzielten Ergebnisse wurden im Rahmen universitärer Fallstudien evaluiert von denen zwei in diesem Artikel vorgestellt werden. Durch die resultierende Systematik, Transparenz und Flexibilität werden traditionelle Lehr- und Lernarrangements bereichert und neue Innovationen im universitären Umfeld entstehen.

1 Motivation

Den Menschen des 21. Jahrhunderts umgibt eine IT-Wolke. Diese besteht aus einer Vielzahl an Geräten und Anwendungen, die Informationen konsumieren, verarbeiten, liefern oder umwandeln. Das nach wie vor gültige *Mooresche Gesetz* führt zu immer kleineren Gerätedimensionen bei steigender Leistung. So reicht das breite Gerätespektrum heute von Personal Computern und Laptops über das allgegenwärtige Mobiltelefon oder sogar Smartphone bis hin zu mehr oder weniger unsichtbarer IT in unseren Wohnungen und Fahrzeugen. In diesem Kontext hat die *Pervasive Computing*-Forschung das Ziel, aus Nutzersicht die Konzentration auf Geschäftsprozesse und persönliche Aufgaben zu ermöglichen und strebt sogar eine proaktive Unterstützung beim Erreichen dieser Ziele an [EK05]. Eine konkrete Ausprägung der Pervasive Computing-Forschung sind *pervasive Umgebungen*. Diese weitgehend transparent von IT durchdrungenen Räume oder sogar raumübergreifenden Bereiche sind in der Lage Nutzeraktivitäten zu erfassen, auszuwerten und darauf derart zu reagieren, dass sie den Nutzer im Erreichen seiner Ziele unterstützen.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

Bei vielen pervasiven Umgebungen [CFBS97] [BMK+00] muss bereits der Entwickler beim Design der Umgebung wissen, welche Zustände diese haben kann und wie die dazu passende Systemreaktion aussieht. Das Resultat dieser Arbeiten ist, dass pervasive Umgebungen heute statische und teure Einzellösungen sind, die fast ausschließlich im Bereich der akademischen oder industriellen Forschung eine Rolle spielen. Damit diese auch für Privatanwender erschwinglich werden ist ein Umdenken in Richtung dynamischer Geräteansammlungen (Ensembles [HK06]) aus bereits heute weit verbreiteten Geräten (z. B. Laptops, Smartphones) anstelle statischer vorkonfigurierter Systeme erforderlich. Die neue Dynamik erfordert eine Infrastruktur, die ohne Vorwissen in der Lage ist unterschiedliche Geräte und Anwendungen transparent zu integrieren. Die Entwicklung und Einführung derartiger Infrastrukturen ist eine der großen Herausforderungen moderner pervasiver Umgebungen [Zen11].

Moderne Bildungseinrichtungen sind mehr und mehr auf dem Weg pervasive Institutionen zu werden, eine Ausprägung pervasiver Umgebungen mit auf Hochschulen zugeschnittenen Komponenten und Interaktionsmustern. Untermauert wurde dies schon 2009 durch den Horizon Report, der sechs Technologien vorstellte, die sich innerhalb der nächsten fünf Jahre in Lehre, Lernen oder kreativer Forschung wahrscheinlich durchsetzen werden [JLS09]:

- Mobile Endgeräte
- Cloud Computing
- Geo-Referenzierung
- Personenbezogenes Web
- Semantische Anwendungen
- Intelligente Objekte

Diese Themen sind auch in der Pervasive Computing-Forschung hochaktuell und erfordern ebenfalls eine entsprechend dynamische Infrastruktur [ZT09]. Die mit dem Begriff *Pervasive University* [TL09] bezeichnete Hochschule der Zukunft muss sich demnach ebenfalls den infrastrukturellen Herausforderungen moderner pervasiver Umgebungen stellen.

Dieser Artikel motiviert auf der Darstellungsschicht des OSI-Referenzmodell den Einsatz Service-orientierter Architekturen (SOA) als Grundlage derartiger Umgebungen und behandelt deren Einsatz für Szenarien im Universitätskontext. Ein zentrales Problem ist dabei SOA-Heterogenität, die durch die verschiedenartigen SOA-Implementierungen (z.B. Web Services, Jini, UPnP) der verfügbaren Gerätetypen entsteht. Zur Behandlung dieses Problems wurden vorhandene SOA-Interoperabilitätsansätze für die fokussierten Pervasiven Umgebungen verglichen und im Hinblick auf die Anforderungen dieser Umgebungen bewertet. Der Einsatz eines konkreten Ansatzes wird in diesem Beitrag vorgeschlagen und im Kontext der Pervasive University evaluiert.

Im Folgenden werden zunächst verschiedene Kommunikationsmodelle für pervasive Umgebungen verglichen und bewertet. Es wird die Eignung Service-orientierter Architektu-

ren motiviert. Im Anschluss wird ein klassifizierender Überblick über existierende SOA-Implementierungen gegeben und ein systematischer Interoperabilitätsansatz vorgestellt. In Abschnitt 4 wird die prototypische Implementierung dieses Ansatzes umrissen, bevor die vorgestellten Ergebnisse im Rahmen der Service-orientierten Universität als Grundlage der Pervasive University evaluiert werden.

2 Infrastrukturen für dynamische, pervasive Umgebungen

Pervasive Umgebungen bestehen aus einer Reihe unterschiedlicher Geräte, die in der Lage sein müssen, miteinander zu kommunizieren. So muss eine fest installierte Infrastruktur beispielsweise Informationen mit den persönlichen mobilen Geräten der im Raum befindlichen Nutzer austauschen, um diesen ggf. zu identifizieren und seine Intention in der Umgebung zu ermitteln. Derartige Grundlagen intelligenten Verhaltens erfordern eine Kommunikationsinfrastruktur, die die besonderen Bedingungen in pervasiven Umgebungen bereits auf Ebene des zugrundeliegenden Kommunikationsmodells berücksichtigt. An diese Architektur werden folgende Anforderungen gestellt (nach [HIR01]):

- Um größtmögliche Interoperabilität und Transparenz zu gewährleisten, muss **von Unterschieden in der Hard- und Software abstrahiert** werden können.
- Für eine hohe Robustheit und Mobilität muss das Wegfallen von Komponenten schnell bemerkt und (falls verfügbar) eine Ersatzkomponente **zur Laufzeit eingebunden** oder andere Komponenten **spontan angepasst** werden können.
- Für die Unterstützung mobiler Nutzer und Geräte muss die Infrastruktur Mechanismen zur **Verteilung und Migration** von Komponenten anbieten.
- Für maximale Flexibilität müssen neue Komponenten **ohne umfassendes Vorwissen gefunden** eingebunden werden können.
- Anwendungen müssen zur Laufzeit in die Infrastruktur eingebunden werden können, wodurch diese entsprechende **Laufzeitumgebungen** zur Verfügung stellen muss.
- Intelligente Systemreaktionen erfordern eine Infrastruktur, die die **Einbindung von Kontextquellen** sowie die **Übertragung der gemessenen Kontextinformationen** unterstützt.
- Zur Verbergung der Systemkomplexität vor dem Nutzer, ist eine **gut skalierbare** aber auch **interaktive** Infrastruktur erforderlich.
- Die Geräteheterogenität und neuartige Mensch/Maschine-Interaktionsmethoden erfordern **generische Nutzerschnittstellen mit einer ausgereiften Bedienbarkeit**.

Ergänzend wurden durch uns zwei weitere Anforderungen identifiziert, die bereits auf Ebene der Infrastruktur Beachtung finden sollten [Zen11]:

Universität Rostock Traditio et Innovatio

Deutsche Forschungsgemeinschaft DFG

	Client / Server	Publ. / Subs.	P2P	SOA
HW/SW-Abstraktion	nein	nein	nein	ja
spontane Komponenteneinbindung	nein	nein	ja	ja
spontane Komponentenanpassung	nein	nein	nein	einige
Komponentenverteilung und –migration	nein	nein	ja	ja
kein Vorwissen erforderlich	nein	nein	nein	ja
bereitgestellte Laufzeitumgebungen	nein	nein	nein	einige
Kontextunterstützung	nein	nein	nein	nein
gute Skalierbarkeit	nein	nein	ja	ja
Integration existierender Komponenten	nein	nein	nein	ja
variable Integrationsintensität	ja	ja	nein	ja

Abbildung 1: Übersicht über die Eignung der grundlegenden Kommunikationsmodelle für die Infrastrukturen pervasiver Umgebungen

- Um pervasive Umgebungen möglichst kurzfristig zu realisieren müssen bereits auf dem Markt **verfügbare Komponenten ohne zusätzliche Modifikationen** integrierbar sein.
- Um Sicherheitsaspekte einzelner Komponenten zu berücksichtigen müssen die integrierten Komponenten autonom bleiben dürfen und ihre **Integrationsintensität selbst definieren** können.

Der in Abbildung 1 dargestellte Vergleich bewertet die Eignung der vier grundlegenden Kommunikationsmodelle für pervasive Umgebungen. Beispielsweise ist die Variierung der Integrationsintensität im Client/Server- und Publisher/Subscriber-Modell durch eine initiale Aushandlung zwischen den Kommunikationspartnern möglich, während in P2P-Netzen angebotene Leistungen für gewöhnlich nicht an den konkreten Kommunikationspartner Peer) angepasst werden können. Service-orientierte Architekturen setzen auf die Verwendung einer Vermittlerinstanz (z.B. Broker) und wodurch insbesondere in Bezug auf spontane, dynamische und systematische Komponentenintegration gravierende Vorteile gegenüber den anderen Ansätzen existieren.

Durch den SOA-Hype hat sich allerdings in den vergangenen Jahren für verschiedene Anwendungsfelder ein breites Spektrum an SOA-Implementierungen etabliert, die nicht kompatibel zueinander sind. Diese SOA-Heterogenität führt zu einer Vielzahl von Unterschieden aber auch Gemeinsamkeiten zwischen den einzelnen Implementierungen, die bereits ausführlich in der Literatur diskutiert wurden [ZMN05][Edw06].

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

Universität Rostock Traditio et Innovatio

Deutsche Forschungsgemeinschaft DFG

	Web Services	UPnP	Jini	DNS-SD	Bluetooth SDP
Broker	ja	nein	ja	optional	nein
Service Bus	Request	Request und Advertisement	Request	Request und Advertisement	Request
Service-Nutzung	SOAP	SOAP	SOA-spezifisch	nicht spezifiziert	SOA-spezifisch
Technologiebindung	theoretisch nicht, faktisch IP	IP	theoretisch nicht, faktisch IP & Java	IP	Bluetooth

Abbildung 2: Übersicht über zur Interoperablilität relevanten Merkmale ausgewählter SOAs.

3 SOA-Interoperabilität

Pervasive Umgebungen setzen sich heute aus einer Vielzahl von Komponenten aus unterschiedlichen Anwendungsbereichen zusammen. Der resultierenden SOA-Heterogenität muss durch Interoperabilitätsansätze begegnet werden. Eine Infrastruktur für diese Umgebungen wäre unvollständig ohne die Berücksichtigung ihrer Probleme.

05.05.2011 UNIVERSITÄT ROSTOCK | Raphael Zender 2

3.1 Klassifikation existierender SOA-Implementierungen

Für die Einführung von Interoperabilität zwischen diesen Implementierungen wurden von uns zunächst folgende technische Klassifizierungsmerkmale identifiziert [Zen11]:

- **Einsatz von Brokern:** Falls überhaupt, welche Art von Broker wird eingesetzt?
- **Aktivität des Service Busses:** Werden Services durch Advertisements bekannt gemacht oder muss der Consumer aktiv anfragen?
- **Spezifikation der Service Nutzung:** Ist spezifiziert, wie Services genutzt werden?
- **Technologiebindung:** Von welchen Technologien ist die SOA-Implementierung abhängig (z.B. IP, Java)?

Basierend auf diesen Kriterien wurden bestehende SOA-Implementierungen als Grundlage für die Interoperabilität klassifiziert. Abbildung 2 zeigt beispielhaft die Anwendung der Klassifikationskriterien für ausgewählte konkrete SOA-Implementierungen.

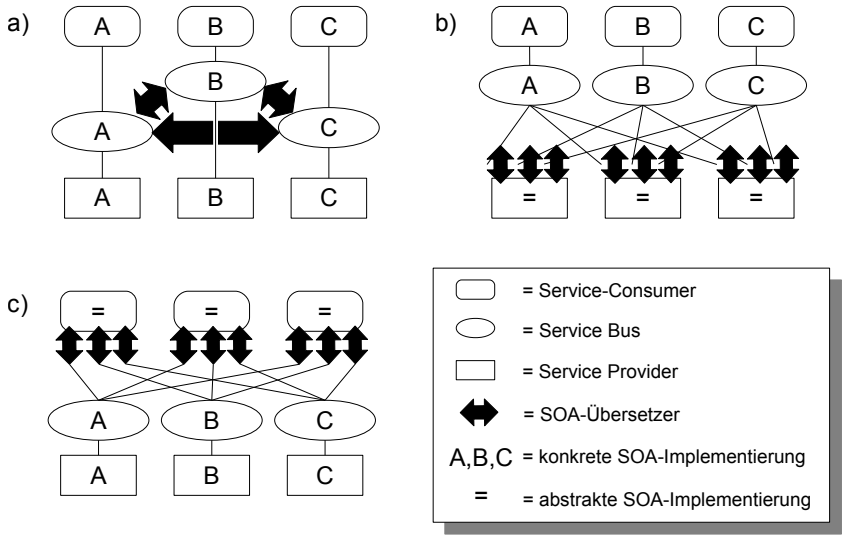

Abbildung 3: Technologiebrücken (a), abstrakte Provider (b) und abstrakte Consumer (c) sind Ansätze zur SOA-Interoperabilität)

3.2 Ansätze zur SOA-Interoperabilität

Die folgenden Interoperabilitätsansätze werden vor allem im Hinblick auf ihre Eignung bewertet, volle Interoperabilität zwischen verfügbaren SOA-Implementierungen zu realisieren [Zen11]:

Definition: Volle Interoperabilität *zwischen n SOA-Implementierungen $t_1, t_2, \ldots t_n$ ist dann erreicht, wenn für jeden Service s einer SOA-Implementierung $t_i \in \{t_1, t_2, \ldots t_n\}$ n Service-Beschreibungen $b_j(t_j, s)$ existieren oder zur Laufzeit erzeugt werden können, für die gilt: $t_j \in \{t_1, t_2, \ldots t_n\}$ und $b_j(t_j, s) \equiv b_i(t_i, s)$. D. h. dass für jeden Service mehrere Service-Beschreibungen existieren, mindestens eine pro SOA-Implementierung.*

Die erste Klasse von Interoperabilitätsansätzen verwendet *Technologiebrücken*. Geboren aus der Notwendigkeit zwei SOA-Implementierungen und miteinander zu verknüpfen, existieren derzeit viele duale Interoperabilitätslösungen. Wie in Abbildung 3a illustriert, werden Services einer SOA-Implementierung A auf Services einer SOA-Implementierung B abgebildet. Somit können von A-Providern angebotene Services, von B-Consumern genutzt werden und umgekehrt. Die Technologiebrücke arbeitet jeweils als Provider und Consumer in beiden SOAs.

Eine Vereinheitlichung von SOA-Implementierungen über **abstrakte Provider** bietet keine Services in einer konkreten SOA-Implementierung an, sondern verwendet abstrakte Service-Beschreibungen (Abbildung 3b). Implementierungsspezifische SOA-Übersetzer generieren aus diesen Beschreibungen Service-Beschreibungen für alle unterstützten SOA-Implementierungen. Analog dazu bieten Provider ihre Services beim Ansatz **abstrakte**

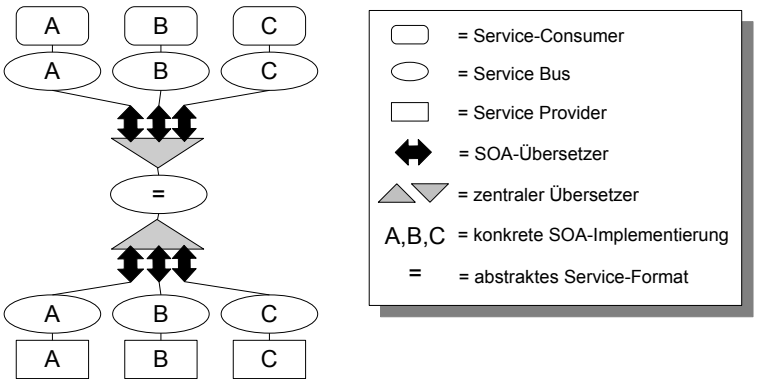

Abbildung 4: Zentrale Vereinheitlichung nutzt eine zentrale Übersetzerinstanz und einen zentralisierten abstrakten Service Bus.)

Consumer (Abbildung 3c) in ihrer ursprünglichen SOA-Implementierung an, während die Consumer SOA-Übersetzer verwenden, um Services aller unterstützen SOA-Implementierungen einheitlich nutzen zu können.

Bei diesen beiden Abstraktionsansätzen besteht ein zentraler Nachteil darin, dass Anpassungen bei den Providern bzw. Consumern vorgenommen werden müssen. Da es allerdings auf dem Markt bereits ein großes Spektrum SOA-spezifischer Provider und Consumer gibt und diese modifiziert werden müssten, ist dieser Ansatz nicht praktikabel.

Durch den Ansatz der **zentralen Vereinheitlichung** werden, wie in Abbildung 4 dargestellt, mit zentralen Übersetzern neue Komponenten eingeführt. Jeder zentrale Übersetzer bündelt mehrere einzelne SOA-Übersetzer. Alle Service-Beschreibungen, die SOA-spezifische Provider anbieten, empfängt ein Übersetzer. Er erzeugt abstrakte Beschreibungen, die der zentrale Übersetzer in einem abstrakten Service-Bus verwaltet. Diese können wiederum in SOA-spezifische Beschreibungen umgewandelt und in allen unterstützten Implementierungen angeboten werden. Der abstrakte Service-Bus kann sowohl lokal auf einem einzigen zentralen Übersetzer existieren, als auch verteilt auf mehreren, um Service-Beschreibungen auch über größere Distanzen hinweg auszutauschen.

Verglichen mit den anderen Ansätzen ist die zentrale Vereinheitlichung somit funktional als auch im Hinblick auf die Effizienz für pervasive Umgebungen zu bevorzugen [ZLT10]. Zwar kann der zentrale Übersetzer zu einem *Single Point of Failure* werden, durch eine entsprechend robuste Komponente und/oder Reserve-Übersetzer kann dieser Fall jedoch im praktischen Einsatz verhindert oder behandelt werden. Aufgrund dieser Betrachtungen ist die zentrale Vereinheitlichung die Grundlage für den durch uns entwickelten *General Purpose Access Point* (GPAP) als zentralen Übersetzer.

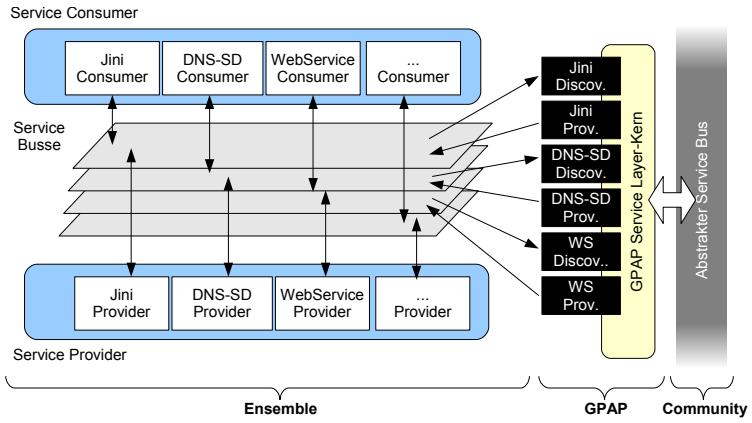

Abbildung 5: Auf seinem Service-Layer überwindet der GPAP SOA-Heterogenität mittels zentraler Vereinheitlichung.

4 Service Layer des General Purpose Access Point

Die Basis der angestrebten dynamischen pervasiven Umgebungen ist ein interoperables und effizientes Kommunikationssystem. Heutige pervasive Umgebungen (z. B. Smart Homes) sind oftmals komplexe und statische Einzellösungen. Die Heterogenität sowohl auf dem Netzwerk- als auch auf dem Service-Layer verhindert die transparente Einbindung neuer Geräte in derartige Systeme. Der entwickelte GPAP überwindet diese Heterogenität systematisch auf diesen beiden kritischen Layern. Details zum Netzwerk-Layer des GPAP sind nicht Bestandteil dieses Artikels und können der Literatur entnommen werden [Dre10].

Um SOA-Interoperabilität zu erreichen, wurde auf dem Service Layer des GPAPs der Ansatz der zentralen Vereinheitlichung umgesetzt. Wie Abbildung 5 illustriert, verbindet der GPAP zudem eine lokale Umgebung (Ensemble) mit einer Community. Communitys sind Ensemble-Netzwerke in denen abstrahierte Service-Beschreibungen zwischen GPAPs ausgetauscht werden können. Der GPAP ist somit sowohl für die SOA-übergreifende Kommunikation innerhalb seines Ensembles als auch die Verbindung seines Ensembles mit einer Community aus abstrakten Diensten zuständig.

Innerhalb der Ensembles mit GPAP bedienen Provider und Consumer weiterhin ihre implementierungsspezifischen Service Busse. Der abstrakte Service Bus wurde über die XML-basierte Meta-Sprache *Service Technology-independent Language* (STiL) realisiert [Zen11]. Die Übersetzung implementierungsspezifischer Service-Beschreibungen und der Suchanfragen auf diesen Beschreibungen wird auf dem GPAP über Plugins für jede SOA-Implementierung und ein übergeordnetes Plugin-Konzept erreicht. Für jede SOA-Implementierung, die vom GPAP unterstützt wird, sind Plugins der folgenden Klassen zu erstellen:

- **Discovery-Plugins** sind Service-Consumer jeweils einer SOA-Implementierung. Diese Plugin-Klasse ist dafür zuständig, Service-Beschreibungen der jeweiligen Imple-

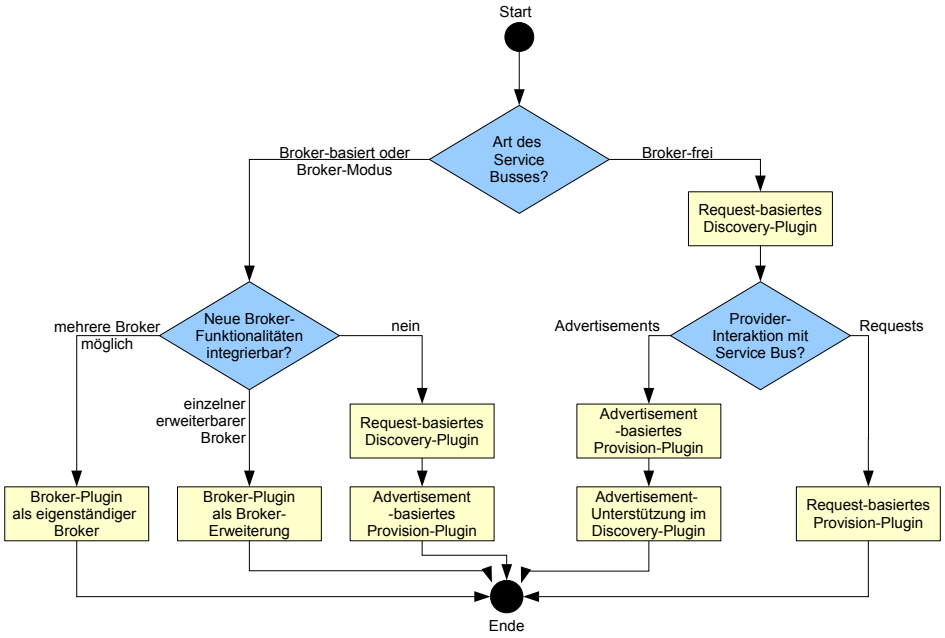

Abbildung 6: Entscheidungsbaum: Welche Plugin-Art und -Ausprägung ist für welche SOA sinnvoll?

mentierung im Ensemble zu finden und äquivalente abstrakte STiL-Beschreibungen zu erstellen.

- **Provision-Plugins** sind Service-Provider jeweils einer SOA-Implementierung. Sie entnehmen dem abstrakten Service Bus STiL-Beschreibungen, erstellen äquivalente Service-Beschreibungen ihrer jeweiligen Implementierung und bieten diese stellvertretend im lokalen Ensemble an.
- **Broker-Plugins** sind Service-Broker jeweils einer SOA-Implementierung bzw. Erweiterungen vorhandener Broker dieser Implementierung. Sie sind Kombinationen aus Discovery- und Provision-Plugins der dieser Implementierung.

Diese grundlegenden Plugin-Klassen können je nach SOA-Implementierung unterschiedliche Strategien verfolgen, um ihre Aufgabe zu erfüllen. Daher wird im Folgenden eine weitere Untergliederung in Plugin-Unterklassen vorgenommen. Abbildung 6 fasst die Entscheidungsfindung für die Auswahl der benötigten Plugin-Arten in Abhängigkeit von ausgewählten Klassifikationskriterien zusammen.

Für jede Plugin-Art und -Ausprägung wurden von uns konkrete Strategien entwickelt, die ein Vorgehensmodell zur Integration neuer SOA-Implementierungen in die GPAP-Architektur bilden. Auf diese Art wurden Plugins zur Integration der SOA Implementierungen Web Services, Jini, DNS-SD und Bonjour SDP in den GPAP realisiert. Im Rahmen

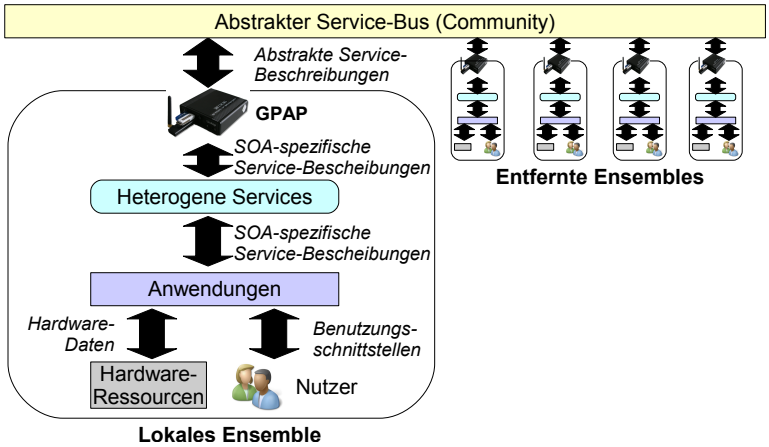

Abbildung 7: Abstraktes Modell einer Service-orientierten (Bildungs)Einrichtung

einer Service-orientierten Universität als Grundlage der pervasiven Umgebung *Pervasive University* [ZT09] wurde die hier vorgestellte Interoperabilitätslösung evaluiert.

5 Evaluierung

5.1 Pervasive University

Wie andere pervasive Umgebungen kann auch die Pervasive University, eine Bildungseinrichtung mit nahtloser und intelligenter IT-Unterstützung, in Ensembles und Communitys unterteilt werden. Ein Ensemble könnte beispielsweise die Infrastruktur einer Abteilung der Bildungseinrichtung (in einer Hochschule z. B. Institute, Lehrstühle, Labore oder Mensen) umfassen. In ihm gibt es Hardware-Ressourcen und dazu gehörige Anwendungen, die von Nutzern über Benutzungsschnittstellen genutzt werden. Die Anwendungen können in zwei Arten unterteilt werden: hardwarefokussierte Anwendungen, die direkt der Kontrolle der Hardware dienen (z. B. Fernbedienungen für industrielle Maschinen) und Software-fokussierte Anwendungen, die lediglich auf Hardware gehostet werden (z. B. HTTP-Server und andere Web-Anwendungen). Entsprechend dieser Einteilung gibt es auch zwei Service-Arten: Services, die Hardware kontrollieren und Services, die Anwendungen nutzen [ZT09]. Die Service-Beschreibungen der Services werden wie in Abbildung 7 dargestellt stets von Anwendungen angeboten.

Um die SOA-Heterogenität der angebotenen Services zu überwinden wird die bereits beschriebene GPAP-Komponente eingesetzt. Die abstrakten Service-Beschreibungen des Ensembles werden in den abstrakten Service Bus eingespeist, der Ensemble-übergreifend die Community bildet und an Universitäten beispielsweise die Fakultät, die gesamte Universität oder gar hochschulübergreifende Organisationen umfasst.

Die im Folgenden aufgeführten Fallstudien verdeutlichen wie Basisdienste einer Universität als Services im Rahmen einer SOA umgesetzt werden können. Nach der beispielhaften Erläuterung der Service-Implementierung eines Basisdienstes (Druckdienst) wird mit der Fallstudie *Immersive Learning* auch ein komplexeres Service-basiertes Lehr- und Lernarrangement betrachtet.

5.2 Fallstudie 1: Druck-Service als Beispiel für einen Basis-Service

Die Anbindung vorhandener universitärer Dienste an eine Service-orientierte Architektur erfolgt nach folgendem generellen Vorgehen auf Provider-Seite:

1. Auswahl einer SOA-Implementierung für die geplante Anbindung
2. Zerlegung des Dienstes in Teildienste (falls praktikabel)
3. Auswahl einer Provider-Instanz für jeden Teildienst
4. Implementierung von Services für jeden Teildienst
5. Deployment der implementierten Services

Auf Consumer-Seite lässt sich kein generelles Vorgehen erkennen. Die bestehenden Anwendungen müssen um eine Consumer-Schnittstelle erweitert werden. Der Aufwand und das Vorgehen bei dieser Erweiterung ist stark von der gewählten SOA-Implementierung sowie der konkreten Anwendung abhängig und kann daher nicht generalisiert werden.

Universitäten sind Institutionen mit einer Vielzahl von Basis-Services. Dazu gehören unter anderem einfache Services zur Nutzung von Infrastruktur, Lernplattformen, Lernmaterialien, und diversen Administrationswerkzeuge. Diese Basis-Services machen den Großteil der universitären Services aus und ihre Umsetzung wird daher anhand eines infrastrukturellen Beispiel-Services für den Zugriff auf einen handelsüblichen Drucker betrachtet. Angestrebt wurde ein Druck-Service, der es Consumern ermöglicht Dokumente auf einen herkömmlichen Drucker auszudrucken ohne über Vorwissen (z. B. Netzwerkadresse des Druckers) und druckerspezifische Software (z. B. Druckertreiber) zu verfügen. Das Vorgehen zur SOA-Anbindung auf Provider-Seite konkretisiert das generelle Vorgehen:

1. Als SOA-Implementierung für die Anbindung wurde Jini gewählt, da für die Benutzungsoberfläche auf dem Druck-Client (Consumer) bereits eine Vorarbeit in Java vorlag.
2. Eine Zerlegung des bereits sehr einfachen Druckdienstes in Teildienste war nicht erforderlich.
3. Der Drucker selbst konnte nicht als Provider des Services fungieren. Daher wurde der Drucker an einem PC (Mediator) angeschlossen. Der netzwerkfähige Mediator konnte den Drucker über seinen Treiber ansprechen und war aus SOA-Sicht der Provider des angestrebten Drucker-Services.

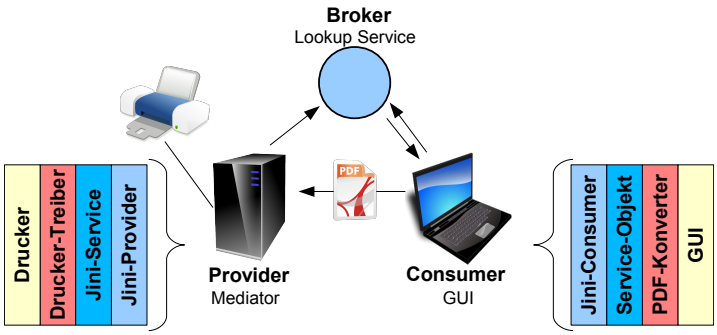

Abbildung 8: SOA-Anbindung am Beispiel eines Druckers als Jini-Service

4. Der Jini-Service konnte über die Java-VM auf den Drucker-Treiber zugreifen und somit den Drucker ansprechen. Der eigentliche Service erwartete ein Dokument im PDF-Format und gibt dieses auf dem Drucker aus.

5. Der Jini-Service wurde über die lokale Jini-Software publiziert und konnte somit von möglichen Consumern gefunden werden.

Wie Abbildung 8 illustriert, benötigt der Druck-Client als Consumer ebenfalls ein lokales Jini-System. Über eine extra zu diesem Zweck entwickelte Benutzungsoberfläche kann der Client einen der verfügbaren Drucker und sein zu druckendes Dokument auswählen. Dieses wird ggf. in ein PDF-Dokument umgewandelt und an den Service bzw. den Drucker gesendet. Für diesen Prototyp ist auf Consumer-Seite neben der entwickelten Software kein weiterer Treiber notwendig und verfügbare Drucker-Services können ohne weiteres Vorwissen gefunden und genutzt werden. Dies sind klare Vorteile gegenüber herkömmlichen Print-Mechanismen wie etwa denen der Windows-Betriebssysteme. Während der Aufwand bei einer geringen Druckeranzahl noch hoch erscheinen mag, skaliert er wiederum gut bei einer großen Druckeranzahl, wie sie an einer Universität vorliegt, zumal ein Mediator für mehrere Drucker zuständig sein kann.

Der Prototyp verdeutlicht beispielhaft, wie einfache Basisdienste einer Universität als Services in eine SOA integriert werden. Neben diesem Beispiel sind weitere grundlegende Services entstanden, wie etwa ein Messaging-Service, über den dem Provider (z. B. Mobiltelefon eines Studenten) eine einfache Textnachricht gesendet werden kann. Derart einfache Services verdeutlichen die grundlegende Vorgehensweise bei der “SOAfizierung” einer Infrastruktur. Das wahre SOA-Potential offenbart sich jedoch erst bei der Einbettung dieser Services in komplexere Anwendungen. So ist beispielsweise der Messaging-Service Teil einer größeren *Immersive Learning*-Lösung zur Vernetzung heterogener Lehr-/Lernumgebungen.

ISBN 978-3-88579-286-4

5.3 Fallstudie 2: Immersive Learning

Neben der klassischen rechnergestützten Präsenzlehre, bei der Lehrende und Lernende zum Zeitpunkt der E-Learning-Veranstaltung am selben Ort sind, hat sich mit der Online-Lehre auch die Verwendung des Internets zur Überbrückung räumlicher Entfernungen durchgesetzt. *Virtual Learning* hat als eine besondere Form der Online-Lehre im vergangenen Jahrzehnt durch die öffentliche Verfügbarkeit virtueller Welten (allen voran Second Life) an Bedeutung gewonnen. Sowohl rechnergestützte Präsenzlehre als auch Virtual Learning haben jeweils Vor- und Nachteile. Vor allem die Ortsgebundenheit der Präsenzlehre und die oftmals komplexe Einrichtung virtueller Lehrveranstaltungen und nicht-triviale Teilnahme an diesen sind dabei hervorzuheben. Um die jeweiligen Nachteile der Paradigmen zu kompensieren wurde über eine SOA-basierte, systematische Kombination zur gegenseitigen Bereicherung realisiert: *Immersive Learning* [ZDLT09].

Ein Lehr-/Lernarrangement des Immersive Learnings beinhaltet Umgebungen der Präsenzlehre und virtuelle Lehr-/Lernumgebungen. Die Umgebungen der Präsenzlehre werden um die Integration entfernter Teilnehmer und die Integration virtueller Lehrmaterialen (z.B. virtuelle 3D-Modelle von Lehrobjekten) bereichert. Im Gegenzug können virtuelle Lehrveranstaltungen um Teilnehmer einer Präsenzlehrumgebung erweitert werden (die ohne technische Hürden wie gewohnt eine Veranstaltung der rechnergestützten Präsenzlehre besuchen) und virtuelle Teilnehmer an realen Lehrveranstaltungen teilnehmen.

Der entwickelte Immersive Learning-Prototyp verwendet einen Seminarraum mit vielfältiger Geräteunterstützung mitsamt seiner exportierten Services als Präsenzumgebung. Dazu gehören:

- Videostreams (vom Vortragenden bzw. Publikum sowie dem Vortragenden-Desktop)
- Audiostreams (von der Stimme des Vortragenden bzw. den Teilnehmern)
- Messaging (individuelle Messaging-Kanäle für alle Teilnehmer, z.B. Mobiltelefone)
- Raumsteuerung (z.B. zur Zuweisung von Videoquellen zu Streamingservern)

Die Services verwenden unterschiedliche SOA-Implementierungen (z.B. Web Services, DNS-SD, Bluetooth SDP). Diesen Services standen die Services der virtuellen Welt *Second Life* (SL) gegenüber, die als Web Services vorlagen. Die Services aus beiden Umgebungen konnten über den GPAP abstrahiert und auf der jeweiligen Gegenseite wieder konkretisiert werden. Technische Details zu diesem Prozess können der referenzierten Literatur entnommen werden [Zen11]. Zu den Services gehören vor allem Video-Input (virtuelle Leinwand) und Messaging (individuelle Messaging-Kanäle für alle Avatare über Chat). Zusätzlich war es über den SL-Client möglich Video- und Audio-Signale aus der virtuellen Welt zu erhalten. Diese wurden jedoch bisher nicht als Services exportiert.

Anhand des in Abbildung 9 illustrierten Anwendungsszenarios wird das Immersive Learning-Potential verdeutlicht. In dem Szenario findet eine Vorlesung in dem Seminarraum statt und einige Teilnehmer nehmen über Second Life an der Veranstaltung teil. Durch den Service-basierten Austausch von Video- und Audio-Signalen wurde dem virtuellen Publikum ermöglicht, die Präsenzvorlesung zu konsumieren und sich sogar interaktiv an der

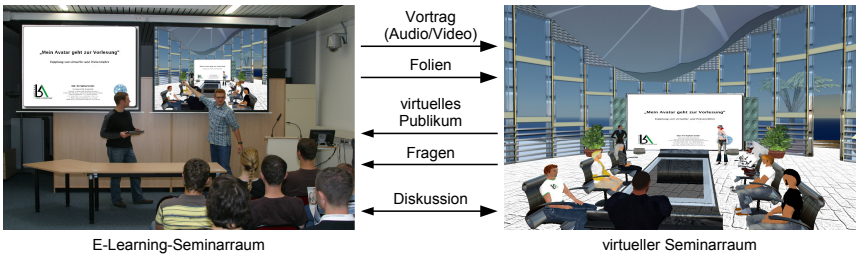

Abbildung 9: Eine reale Vorlesung aus dem E-Learning-Seminarraum wird “virtualisiert”.

Diskussion zu beteiligen. Der entstandene Prototyp wurde in darüber hinaus in diversen universitären Szenarien erfolgreich eingesetzt. Dazu gehörten unter anderem:

- Vorlesungen mit virtuell und lokal anwesenden Studierenden (evaluiert in [LZ11])
- Diplomarbeitsverteidigung mit virtuell anwesenden Gutachtern
- Workshop mit der Übertragung von virtuell gehaltenen Vorträgen in die reale Welt

6 Zusammenfassung

Dieser Artikel thematisierte die infrastrukturellen Herausforderungen für pervasive Umgebungen im Allgemeinen und begründet deren Gültigkeit für pervasive Hochschulen im Speziellen. Basierend auf dieser Analyse wurde die Eignung heutiger Kommunikationsmodelle als infrastrukturelle Basis zur Erfüllung der speziellen Anforderungen derartiger Umgebungen bewertet und die Service-orientierte Architektur als infrastrukturelle Grundlage vorgeschlagen. Es wurde jedoch mit SOA-Heterogenität auch eine zentrale Schwäche Service-orientierter Architekturen dargelegt. Pervasive Umgebungen setzen sich heute aus einer Vielzahl unterschiedlicher Komponenten aus verschiedenen Anwendungsbereichen zusammen. Da sich in den jeweiligen Bereichen unterschiedliche SOA-Implementierungen etabliert haben, müssen pervasive Umgebungen Mechanismen zur Behandlung der SOA-Heterogenität bereithalten. Nach einer Betrachtung der relevanten SOA-Implementierungen wurde in dieser Arbeit ein systematischer Interoperabilitätsansatz auf der Grundlage zentralisierter Abstraktion erläutert und dessen Implementierung umrissen.

Anhand konkreter Szenarien an einer Universität der Zukunft, wurde das Potential einer Berücksichtigung verschiedener SOA-Implementierungen dargestellt. Die Konzentration auf eine einzige SOA-Implementierung würde entweder auf Kosten der Transparenz gehen, da zusätzliche Modifikationen vorhandener Komponenten erforderlich wären, oder sogar Komponenten von vornerein ausschließen.

Der Weg traditioneller Hochschulen zur Pervasiven Universität führt auf infrastruktureller Ebene über die SOA-Einführung. Diese gestaltet sich aufgrund der oftmals proprietären IT-Einzellösungen im universitären Umfeld nicht immer einfach. Hier ist in den nächsten

Jahren ein Aufbrechen der monolithischen Softwarekomponenten erforderlich. Im Einzelfall muss dabei geklärt werden, welche Teilkomponenten wie als Services angeboten werden sollten und ob einzelne Funktionalitäten und Inhalte auch von externen Dienstleistern (z.B. anderen Bildungseinrichtungen) integriert werden können. Neben neuen Prozessmodellen für die Interaktion zwischen internen und externen Services sowie SOA-freien Komponenten ist auch ein Umdenken und Kooperationsbereitschaft von Seiten der Verantwortlichen für universitäre Lösungen erforderlich. Dieser Weg zur Pervasiven Universität ist zwingend erforderlich und klar erkennbar. Er wird weiterhin durch diverse Projekte an den einzelnen Hochschulen und durch die neue Mobilität und Flexibilität der Studenten und Mitarbeiter gefördert und gefordert.

Literatur

[BMK+00] B. Brumitt, B. Meyers, J. Krumm, A. Kern und S. Shafer. EasyLiving: Technologies for Intelligent Environments. S. 12–29. Springer., 2000.

[CFBS97] J. R. Cooperstock, S. S. Fels, W. Buxton und K. C. Smith. Reactive Environments: Throwing Away Your Keyboard and Mouse. *Communications of the ACM*, 40:65–73, 1997.

[Dre10] E. Dressler. *Zuverlässige, kontextbezogene Geräte-Sensoren-Netzwerke in horizontalen und vertikalen drahtlosen Kommunikationsarchitekturen (Dissertationsschrift)*. Universität Rostock, Rostock, Germany, 2010.

[Edw06] W. K. Edwards. Discovery systems in ubiquitous computing. *IEEE Pervasive Computing*, 5/2:70–77, April 2006.

[EK05] J. L. Encarnaço und T. Kirste. Ambient Intelligence: Towards Smart Appliance Ensembles. In M. Hemmje et al, Hrsg., *From Integrated Publication and Informations Systems to Virtual Information and Knowledge Environments*, S. 261–270. Springer LNCS 3379, 2005.

[HIR01] K. Henricksen, J. Indulska und A. Rakotonirainy. Infrastructure for Pervasive Computing: Challenges. In *Workshop on Pervasive Computing at INFORMATIK 01*, S. 214–222, Vienna, Austria, September 2001.

[HK06] T. Heider und T. Kirste. Building smart environments: The Ensemble Challenge. *Information Management and Consulting*, 4/06:6–12, 2006.

[JLS09] L. Johnson, A. Levine und R. Smith. The 2009 Horizon Report. Bericht, The New Media Consortium, Austin, TX, USA, 2009.

[LZ11] U. Lucke und R. Zender. 3D Interactions between Virtual Worlds and Real Life in an E-Learning Community. *Advances in Human-Computer Interaction*, "2011, , Article ID 684202", 2011.

[TL09] D. Tavangarian und U. Lucke. Pervasive University - A Technical Perspective (Die Pervasive University aus technischer Perspektive). *it - Information Technology*, 51(1):6–13, Januar 2009.

[ZDLT09] R. Zender, E. Dressler, U. Lucke und D. Tavangarian. Pervasive Media and Messaging Services for Immersive Learning Experiences. In *Proceedings of PerEL 2009, Workshop at 7th IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Galveston, TX, USA, Marz 2009. IEEE Computer Society.

[Zen11] R. Zender. *Service-basierte Infrastruktur für pervasive Lehr- und Lernarrangements (Dissertationsschrift)*. Universität Rostock, Rostock, Germany, 2011.

[ZLT10] R. Zender, U. Lucke und D. Tavangarian. SOA Interoperability for Large-scaled Pervasive Environments. In *5th IEEE Int. Workshop on Service Oriented Architectures in Converging Networked Environments (SOCNE) im Rahmen der IEEE 24th Int. Conference on Advanced Information Networking and Applications (AINA)*, Perth, Australia, April 2010. IEEE Computer Society.

[ZMN05] F. Zhu, M. W. Mutka und L. M. Ni. Service discovery in pervasive computing environments. *IEEE Pervasive Computing*, 4/4:81–90, Oktober 2005.

[ZT09] R. Zender und D. Tavangarian. Service-oriented University: Infrastructure for the University of Tomorrow. In *International Conference on Intelligent Interactive Assistance and Mobile Multimedia Computing 2009 (IMC 2009)*, S. 73–84, Rostock, Germany, November 2009. Springer.