

2011

Kontextsensitive Service-Infrastruktur für die mobile Nutzung von Home-IT

Marc Herbrechter

Universität Siegen, marc.herbrechter@uni-siegen.de

Benedikt Ley

Universität Siegen, benedikt.ley@uni-siegen.de

Martin Stein

Universität Siegen, martin.stein@uni-siegen.de

Follow this and additional works at: <http://aisel.aisnet.org/wi2011>

Recommended Citation

Herbrechter, Marc; Ley, Benedikt; and Stein, Martin, "Kontextsensitive Service-Infrastruktur für die mobile Nutzung von Home-IT" (2011). *Wirtschaftsinformatik Proceedings 2011*. 14.

<http://aisel.aisnet.org/wi2011/14>

This material is brought to you by the Wirtschaftsinformatik at AIS Electronic Library (AISeL). It has been accepted for inclusion in Wirtschaftsinformatik Proceedings 2011 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Kontextsensitive Service-Infrastruktur für die mobile Nutzung von Home-IT

Marc Herbrechter

Wirtschaftsinformatik und
neue Medien
Universität Siegen

marc.herbrechter@uni-siegen.de

Benedikt Ley

Wirtschaftsinformatik und
neue Medien
Universität Siegen

benedikt.ley@uni-siegen.de

Martin Stein

Wirtschaftsinformatik und
neue Medien
Universität Siegen

martin.stein@uni-siegen.de

ZUSAMMENFASSUNG

Die Digitalisierung der Medienlandschaft und die damit einhergehenden Veränderungen für die Home-IT ermöglichen eine Vielzahl neuer Dienste. Beispielsweise sind Fernsehgeräte nicht mehr nur reine Informations- und Unterhaltungsmedien sondern bieten durch Netzwerkfähigkeit bereits heute eine Reihe von interaktiven Zusatzfunktionalitäten. Um solche Dienste den Nutzern sinnvoll zugänglich zu machen wurde im Rahmen dieser Arbeit ein Framework entwickelt, das eine dynamische Service-Infrastruktur für Home-IT Umgebungen bereitstellt und die es Benutzern ermöglicht lokal verfügbare Dienste zu finden und diese über ihr mobiles Endgerät zu nutzen.

SCHLÜSSELWÖRTER

Home-IT, Kontextsensitive Systeme, Serviceorientierte Architekturen, Komponentensbasierte Architekturen, OSGi

1. EINLEITUNG

Die Digitalisierung der Medienlandschaft bringt eine Vielzahl an Veränderungen für die häusliche Anwendungs- und Informationstechnik (Home-IT) mit sich. Das Fernsehen ist weiterhin immer noch eines der am stärksten genutzten Informations- und Unterhaltungsmedien in der deutschen Gesellschaft. Gleichzeitig aber gewinnt das Internet als Unterhaltungs- und vor allem als Kommunikationsmedium immer mehr an Bedeutung [17]. Bereits heute integrieren viele Fernsehgeräte, häufig in Kombination mit Media-Center PCs oder Set-Top Boxen, eine Netzwerkschnittstelle und ermöglichen somit zahlreiche internetbasierte Zusatzfunktionen wie Webradio, On-Demand Videodienste oder den Zugang zu sozialen Netzwerken. Die damit verbundene Veränderung des Nutzungsverhaltens stellt allerdings gleichzeitig neue Anforderungen an die Gestaltung von Benutzerschnittstellen um solche interaktiven Mehrwertangebote sinnvoll nutzen zu können. Wir haben ein Framework konzipiert, das eine dynamische Service-Infrastruktur für Home-IT Umgebungen bereitstellt und die Nutzung von Diensten erleichtert. Über ein

Smartphone oder ein anderes mobiles Endgerät können die Benutzer erschließen, welche Dienste verfügbar sind und die für sie Interessanten auf dem Endgerät installieren, um z.B. eine Funktion auf einem lokalen Media Center PC nutzen zu können. Damit die Benutzer nicht mit einer zu großen Auswahl an Diensten überfordert werden, verfolgen wir darüber hinaus einen kontextsensiblen Ansatz, so dass nur solche Dienste angezeigt werden, die in der unmittelbaren Umgebung des Anwenders verfügbar und nutzbar sind. Das Framework basiert auf einer komponentenbasierten Architektur, die eine plattform-unabhängige Nutzung sowie das dynamische Hinzufügen und Entfernen von Diensten ermöglicht. Gleichzeitig werden aber auch die für eine serviceorientierte Architektur (SOA) typischen Kommunikationsmuster zwischen Service-Consumer, Service-Broker (Service-Registry) und Service-Provider implementiert, um den Benutzern dynamisch neue oder aktualisierte Dienste anzubieten. Eine Softwarekomponente entspricht hierbei einem Service, der jedoch für die Nutzung lokal auf dem Endgerät installiert wird, so dass auch native Funktionalitäten der Endgeräte genutzt werden können.

2. STAND DER FORSCHUNG

Existierende Arbeiten haben gezeigt, dass Fernsehen keine isolierende Aktivität ist, sondern Gesprächsthemen bereitstellt und damit soziale Interaktion fördert, die dank Informationstechnologie auch trotz großer Distanzen unterstützt werden kann [10]. Es wurde außerdem deutlich, dass auch eine parallele Nutzung von Internetdiensten und Fernsehen von Interesse ist, beispielsweise bei Chat-Anwendungen [14].

Die wachsende Verfügbarkeit von Diensten auf immer mehr Endgeräten erfordert in Verbindung mit der Erwartung der Benutzer nach immer stärker personalisierten Lösungen, die deren Wünsche und Bedürfnisse widerspiegeln [24], hoch dynamische Strukturen. Es wurden verschiedene Paradigmen entwickelt, die Anpassung und Erweiterung bestehender Anwendungssysteme erleichtern. Geeignete Ansätze bieten beispielsweise serviceorientierte Architekturen oder Komponentenmodelle, die durch Abstraktion von Services bzw. Komponenten gut in heterogenen und sich wandelnden Infrastrukturen eingesetzt werden können. Komponentenmodelle basieren auf Softwarekomponenten, welche gemäß eines Kompositions-Standards mit anderen Komponenten verknüpft und ausgeführt werden können, ohne dass Änderungen an den Komponenten vorgenommen werden müssen [11]. Demnach besitzen Softwarekomponenten definierte Schnittstellen über die mit anderen Komponenten

kommuniziert werden können. In welcher Form sich eine solche Komponente darstellt, hängt vom darunterliegenden Komponentenmodell ab. Sollten Komponenten nicht vollständig unabhängig ausgeführt werden können, so werden Abhängigkeiten mit benötigten zusätzlichen Komponenten explizit durch die Komponenten angegeben. Im Vergleich dazu sind SOA ein Paradigma für die Verteilung von Anwendungen, wobei einzelne Funktionalitäten (Services) bei unterschiedlichen Anbietern liegen können. SOA spezifizieren einheitlich, wie solche Funktionalitäten bzw. Dienste angeboten und gefunden werden können, wie mit diesen interagiert werden kann und wie diese genutzt werden können [3].

Eine Integration von serviceorientierten Ansätzen und komponentenbasierten Modellen wird bereits im „Gravity“ Projekt verfolgt [4]. Die Autoren beschreiben ein Komponentenmodell bei dem die Kommunikation zwischen den Komponenten über Services abläuft. Abhängig von der Verfügbarkeit bestimmter Services kann so eine dynamische Anpassung der Anwendung ermöglicht werden. Eine weitere Arbeit von Sánchez-Nielsen et al. beschreibt ein Framework auf der Basis von SOA, welches speziell für mobile Endgeräte entwickelt wurde und die dynamische Integration von Services sowie das dynamische Auffinden neuer Services unterstützt [22].

Ansätze in denen komponentenbasierte Frameworks mit Multimediaanwendungen verknüpft wurden, finden sich beispielsweise in der Arbeit von Redondo et al. Sie beschreiben eine kooperative Umgebung zwischen OSGi (Open Service Gateway Initiative) als komponentenbasierte Plattform und MHP (Multimedia Home Platform). Die entwickelte Plattform ermöglicht auf Basis von OSGi die Nutzung von Diensten verschiedener Geräte im Haushalt innerhalb von MHP Anwendungen [21]. Auch Gu et al. verwenden eine OSGi-Plattform. In ihrer Arbeit stellen sie eine Infrastruktur für kontextsensible Anwendungen vor. Das dynamische Verhalten dieser Anwendung basiert dabei auf einem formalen ontologiebasierten Kontextmodell [9].

Passen sich Systeme an den jeweiligen Kontext an so spricht man von kontextsensiblen Systemen. Nach Dey handelt es sich um ein kontextsensibles System, „wenn es den Kontext benutzt, um dem Nutzer relevante Informationen und/oder Dienste zur Verfügung zu stellen, wobei die Relevanz aufgabenabhängig ist“ [7]. Um den Kontext bewerten zu können, müssen Informationen des Umfeldes abstrahiert und formalisiert werden. Dabei werden nach Schilit et al. und Pascoe folgende Aspekte betrachtet: Wo ist der Nutzer, welche Personen sind in der Nähe und welche Ressourcen sind verfügbar [19,23]? In unserer Arbeit wird der Kontext auf der Basis von ortsbezogenen und infrastrukturellen Informationen gemessen.

Die Erfassung ortsbezogener Kontextinformationen kann im Freien heute problemlos mit GPS realisiert werden. Die Lokalisierung innerhalb geschlossener Gebäude hingegen stellt immer noch ein Problem dar. Cheng stellt in seiner Arbeit einen möglichen Lösungsansatz vor, der eine Ortung auf der Basis des Funknetzstandards ZigBee realisiert [5]. Auch das System „Locawe“ setzt bei der Ortung innerhalb von Gebäuden auf ZigBee-Netzwerkinformationen auf, jedoch zusätzlich kombiniert mit WLAN-Netzwerkinformationen [15]. Eine andere Möglichkeit wurde von Betz et. al entwickelt. Sie

beziehen die Kontextinformationen über RFID-Tags, die an Gegenständen befestigt werden und zu diesem Informationen beinhalten. Die ortsbezogene Kontextinformation die durch das Auslesen eines Tags erfasst wird stellt hierbei keine absolute Ortsangabe dar, da die ortsbezogene Information an den getaggen Gegenstand gebunden ist. Wird dieser bewegt, ändert sich auch die Position des Tags [2].

Neben der Identifizierung des räumlichen Kontexts existiert das Problem der Darstellung der im Kontext verfügbaren Dienste. Es bestehen verschiedene Möglichkeiten solche Dienste zu visualisieren, jedoch muss die Darstellung zum einem generisch genug sein um für Geräte bzw. Dienste unterschiedlichster Art spezifiziert werden zu können, aber zum anderem nicht zu abstrakt sein, um für den Nutzer gebrauchstauglich zu bleiben. Ein Ansatz zur Realisierung einer solchen generischen Visualisierung wird in [16] dargestellt. Die Autoren stellen eine geräteunabhängige Sprache zur Spezifizierung von Oberflächen vor, die zudem die Möglichkeit bietet, Oberflächen gemäß den Anforderungen des Nutzers zu individualisieren. Die Sprache eignet sich zur Beschreibung der Funktionalität von über 30 verschiedenen Gerätearten, wie Fernsehgeräten, Videorecordern und Kopierern, aber auch zur Steuerung von Microsoft Powerpoint und des Windows Media Players. Als Konzept für den Einsatz der Beschreibungssprache stellen die Autoren folgenden Ablauf vor:

1. Das zu steuernde Gerät überträgt eine Beschreibung seiner Funktionen in der vorgestellten Beschreibungssprache.
2. Das mobile Endgerät generiert aus dieser Beschreibung eine Oberfläche, die vom Benutzer individuell angepasst werden können.
3. Durch Betätigen der generierten Steuerelemente werden, die durch die Beschreibung assoziierten Steuersignale an das zu bedienende Gerät übertragen.

Die Arbeit stellt einen Ansatz zur Steuerung von stark heterogenen Umgebungen dar. Im Vergleich dazu nutzen wir für unseren Ansatz ein ähnliches Konzept zur Beschreibung von Benutzerschnittstellen, die jedoch nicht den Kernfokus dieser Arbeit bilden und daher nur rudimentär erarbeitet wurden.

Problematisch bei der Entwicklung mobiler Anwendungen sind die begrenzten Ressourcen mobiler Endgeräte (Rechenleistung, Speicherkapazität, Akkuleistung etc.). Es ist daher sinnvoll, dass Anwendungen nur dann laufen, wenn dies auch wirklich erforderlich ist. Um dies zu erleichtern werden u.a. „Push“-Dienste eingesetzt. Sie ermöglichen es, dass ein mobiles Endgerät bzw. eine Anwendung erst dann aktiv wird, wenn bestimmte Umstände eintreten, wie bspw. die Aktualisierung von Daten oder der Erhalt einer Nachricht. Hierfür stellen verschiedene Plattformanbieter mobiler Endgeräte Push-Server bereit, über die Anwendungsentwickler Benachrichtigungen an die jeweiligen Endgeräte schicken können, ohne dass die entsprechende Anwendung auf dem Gerät gestartet sein muss [8][1].

3. HOME-IT FRAMEWORK

Im folgenden Abschnitt wird auf das grundlegende Konzept für die Plattform und unsere prototypische Umsetzung eingegangen. Als grundlegendes Gerüst für die Implementierung wurde das

OSGi Framework ausgewählt, welches später kurz erläutert werden soll. Auf dieser Grundlage wurden verschiedene Komponenten entwickelt, die zum einen grundlegende Dienste zur Steuerung, Kommunikation und Bedienung bereitstellen und zum anderen die Entwicklung darauf aufbauender Pakete strukturieren und erleichtern.

3.1 KONZEPT

In dieser Arbeit stellen wir einen integrierten Ansatz zur Nutzung und Steuerung von Home-IT Technologie auf mobilen Endgeräten vor. Hierbei sollen zum einen den individuellen Anforderungen heterogener Home-IT Umgebungen Rechnung getragen werden und zum anderen durch ein plattformunabhängiges Design beliebige mobile Endgeräte unterstützt werden. Die Anpassung an verschiedene Umgebungen zur Laufzeit erfordert eine hohe Dynamik, da ein Umgebungswechsel des Benutzers häufig mit einer vollständigen Veränderung der vorhandenen technischen Infrastruktur und den damit verbundenen Diensten und Funktionalitäten einhergeht. Die hohe Komplexität heterogener Umgebungen verhindert, dass ein Service oder eine Funktion (z.B. zur Steuerung eines bestimmten Fernsehgerätes) in beliebigen Home-IT Umgebungen genutzt werden kann. In unserer Arbeit haben wir ein Konzept für ein Framework entwickelt, das umgebungsspezifische Dienste und Anwendungen den Benutzern auf deren mobilen Endgeräte zur Nutzung bereitstellt. Durch die Beschränkung auf lokal verfügbare Dienste wird vor allem die Komplexität der Visualisierung verfügbarer Services vermindert, so dass der Benutzer kontextrelevante Dienste schneller auffinden und nutzen kann.

Für die Konzeption eines solchen Frameworks bedarf es einer hoch flexiblen Architektur die es ermöglicht, dass Dienste und Anwendungen zur Laufzeit plattformunabhängig den Anwendern zur Nutzung bereitgestellt werden können. Bei unserem Ansatz setzen wir auf ein Konzept, das das Rollenmodell serviceorientierter Architekturen mit den Vorteilen komponentenbasierter Ansätze integriert. Dienste werden demnach nicht wie z.B. Webservices entfernt aufgerufen, sondern lokal auf dem mobilen Gerät des Benutzers installiert. Dies hat den Vorteil, dass auch native Sensoren und Funktionen des Endgeräts (z.B. Bewegungssensoren, Vibration, Kamera, Mikrofon) von der Anwendung genutzt werden können.

Analog zum Rollenkonzept serviceorientierter Architekturen wird zwischen drei verschiedenen Rollen differenziert [3]. Services sind dabei als einzelne Komponenten realisiert. Zunächst gibt es die Rolle des Service-Nutzers, welcher Services auswählt und nutzt. Eine Auswahl von Services wird vom Service-Broker bereitgestellt. Der Service-Broker hält dazu ein Verzeichnis von Komponenten bereit, welches verschiedene Metadaten wie Name, Version und Ort, zu den jeweiligen Softwarekomponenten beinhaltet. Der Service-Provider ist, der im Brokerverzeichnis spezifizierte Ort bzw. Server der Komponente. Der konzeptuelle Aufbau einer beispielhaften Home-IT Umgebung ist in **Abbildung 1** dargestellt. Betritt demnach ein Benutzer mit seinem mobilen Endgerät eine solche Umgebung, werden Informationen über verfügbare Dienste von lokalen Service-Brokern (z.B. TV-Gerät, Hi-Fi-Anlage) auf das

Endgerät gepusht (1). Der Nutzer hat nun die Möglichkeit aus der dargestellten Liste die für ihn relevanten Dienste (z.B. Fernbedienung für TV-Gerät) auszuwählen und zu installieren. Dabei kann die zu installierende Komponente entweder direkt beim Service-Broker liegen oder aber auf einem beliebigen externen Server (2). Wurde eine Komponente erfolgreich installiert, so kann der Benutzer diese z.B. zur Steuerung des lokalen TV-Geräts verwenden. Beim Verlassen der Home-IT Umgebung werden die installierten Anwendungen wieder entfernt bzw. ausgeblendet, so dass dem Benutzer immer nur kontextrelevante Dienste angezeigt werden. Das Installieren und Entfernen von Diensten erfolgt dabei zu Laufzeit, ohne dass ein Beenden oder Neustarten des Geräts erforderlich wird.

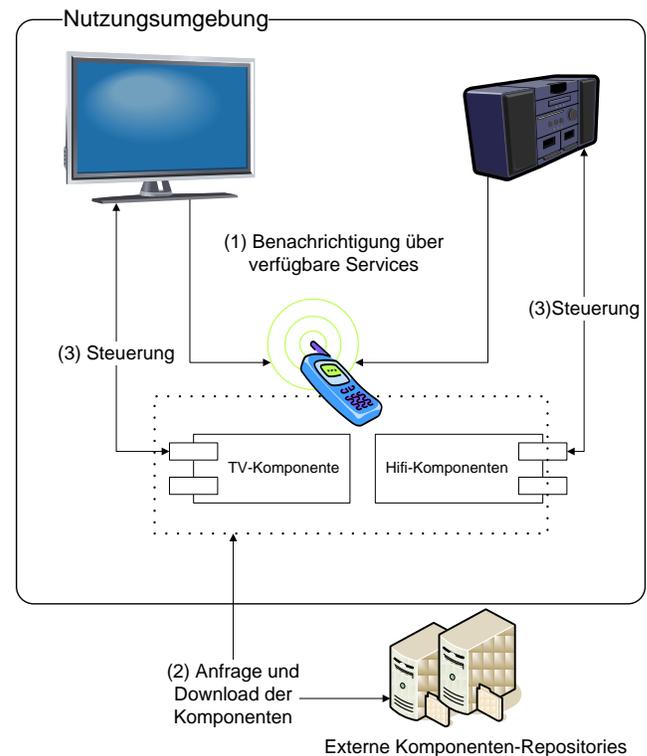


Abbildung 1: konzeptueller Aufbau der Serviceinfrastruktur für Home-IT Umgebungen

3.1.1 Architektur

Für die Realisierung eines solchen hochdynamischen, kontextsensitiven Frameworks bedarf es einer architektonischen Basis, die die Konzepte von SOA und komponentenbasierten Architekturen integriert und die erforderliche Flexibilität bereitstellt. Daher haben wir uns bei der Konzeption und der späteren prototypischen Realisierung des Home-IT Frameworks für den Einsatz von OSGi entschieden

OSGi ist ein auf Java basiertes Framework für dynamische, komponentenbasierte Softwareentwicklung. Es erlaubt das dynamische Laden so genannter Bundles zur Laufzeit, also ohne Neustart der Umgebung oder des Geräts. Weiter erlaubt es eine entfernte Wartung und Konfiguration sowie das entfernte

Installieren, Starten, Stoppen und Updaten von Bundles. Die Pakete werden von der Laufzeitumgebung implizit einem Lebenszyklus unterstellt und können nebenläufig oder interaktiv ausgeführt werden. OSGi als Komponentenmodell findet Anwendung in vielen renommierten Open-Source Projekten wie Eclipse. Zahlreiche namhafte Unternehmen wie IBM und SAP haben sich in der OSGi Alliance zusammengeschlossen, um die Standardisierung voran zu treiben.

Das OSGi Framework basiert auf einer mehrschichtigen Architektur, welche in Abbildung 2 dargestellt ist. Zur Bereitstellung von Sicherheitsaspekten setzt OSGi auf die Java 2 Security Architecture in Verbindung mit spezifischen OSGi Sicherheitsmechanismen. Hierzu können Bundles auf Basis einer Public Key Infrastructure mittels X.509 Zertifikaten gesichert werden um Aspekte wie Authentizität und Integrität zu gewährleisten. Die Sicherheitsschicht zieht sich Vertikal durch das gesamte Framework. Als Ausführungsumgebung kommen insbesondere Java Umgebungen wie die Java Standard Edition in Frage, wobei ein bestimmter Umfang der Java API vorausgesetzt wird. In der Lebenszyklusschicht wird eine API für Funktionen wie das Installieren, Starten, Stoppen, Updaten und Deinstallieren von Bundles bereitgestellt welche auch über entfernte Aufrufe erfolgen können. In der Serviceschicht werden Methoden für das Publizieren, Finden und Nutzen von Services bereitgestellt. Als serviceorientierte Architektur betrachtet ist diese Schicht mit einem Service Broker vergleichbar, wobei zusätzlich Persistenzdienste zur Verfügung stehen [18].

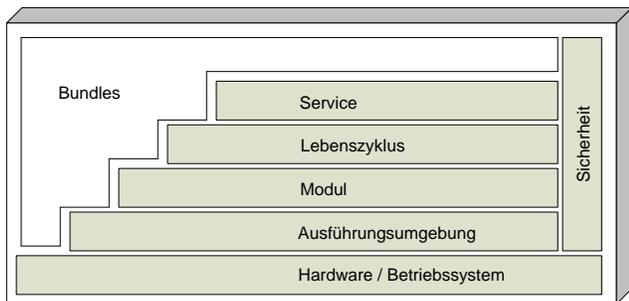


Abbildung 2: Schichten des OSGi-Frameworks

OSGi Bundles können unabhängig voneinander genutzt und ausgeliefert werden können. Dies ermöglicht den Entwicklern einen hohen Grad an Modularisierung, das insbesondere bei häufig vorkommenden partiellen Änderungen an Anwendungssystemen hilfreich sein kann. Durch einen hohen Grad an Wiederverwendbarkeit kann zudem der Implementierungsaufwand für zukünftige Module verringert und durch die Einbindung von standardisierten Kommunikationsformen die Interoperabilität – z.B. mit Webservices – gefördert werden.

Für das auf der OSGi Plattform aufbauende Home-IT Framework wurde eine nachrichtenorientierte Kommunikation gewählt, die auf dem Internetprotokollstapel aufbaut (s. Abbildung 3). Hierdurch wird der ereignisorientierte Nachrichtenaustausch abstrahiert, um auf verschiedenen Wegen stattfinden zu können. Für die Kommunikation zwischen mobilem Endgerät und Basisstation wurde eine einfache

Implementierung auf Socket Basis gewählt während für die Kommunikation zwischen Media Center und Basis-Station Webservices zum Einsatz kommen.

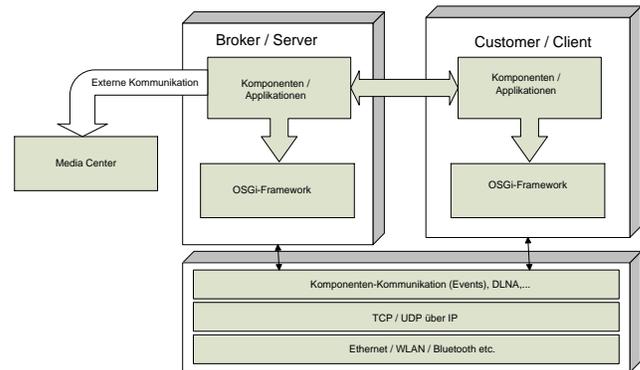


Abbildung 3: Aufbau von Server und Client und deren Kommunikation untereinander und mit externen Systemen

Durch den hohen Grad der Modularisierung des Gesamtsystems ist es wichtig, einen einheitlichen Weg der Ereignisverarbeitung zu schaffen, um komponentenübergreifend und transparent auf Ereignisse zu reagieren. Um dies sicherzustellen wurde das Konzept der komponentenbasierten Softwareentwicklung mit Konzepten der nachrichtenorientierten Kommunikation kombiniert. Ereignisse die innerhalb einer Komponente des Gesamtsystems auftreten, werden von dieser an eine zentrale Komponente weitergegeben, die aufgrund der Informationen innerhalb der Nachricht entscheidet, über welchen Kommunikationsweg und an welche Komponente das Ereignis weitergegeben werden soll.

3.1.2 Generische Oberflächenbeschreibung

Da Komponenten plattformunabhängig auf unterschiedlichen Geräten ausgeführt werden sollen, ist es erforderlich, dass die Beschreibung der grafischen Oberflächen der Anwendungen als Bestandteil der Komponente im Bundle mit abgelegt werden, so dass diese auf den jeweiligen Geräten interpretiert und nativ dargestellt werden können. Hierfür verwenden wir eine XML-basierte Beschreibungssprache (ähnlich XUL).

3.2 PROTOTYPISCHE UMSETZUNG

Die Kommunikation zwischen den Hauptkomponenten Home-IT Geräte, Basis Station und mobilem Endgerät wurde nach dem Client- und Server Paradigma realisiert wobei unterschiedliche Techniken zum Einsatz kommen. Um die Kommunikation zwischen Basis und mobilem Endgerät möglichst leichtgewichtig zu halten wird diese über Socketverbindungen auf Basis von TCP und UDP umgesetzt. Die Kommunikation zwischen Basis und Media Center wird aufgrund der einfacheren Handhabbarkeit durch Webservices realisiert (Basis und Home-IT Gerät können hierbei auch in einem Gerät vereint sein). Über diese Kommunikationskanäle werden ereignisorientiert Nachrichten versandt, welche im Framework selber generiert und ausgewertet werden können. Der Einfachheit halber werden diese Nachrichtenobjekte als JSON Repräsentationen auf den jeweiligen Kanälen übertragen [6].

Für die Umsetzung auf den mobilen Endgeräten wurde Google Android als exemplarische Plattform gewählt sowie die OSGi

Implementierung der Firma Prosys GmbH [20]. Diese Kombination stellt zum Zeitpunkt der Veröffentlichung den größten Funktionsumfang zur Verfügung und erlaubt die schnelle und einfache Implementierung verschiedener Pakete für die exemplarischen Nutzungsfälle.

Da Google Android eine besondere Form der Java Umgebung bereitstellt, sind einige Besonderheiten zu beachten und diverse Änderungen an den Bundles vorzunehmen. Durch die bei der Android Plattform verwendete Dalvik Java VM muss der Bytecode vor dem Deployment umgewandelt werden um auf dem Zielsystem lauffähig zu sein. Weiter fehlen der Java Umgebung auf der Android Plattform diverse Pakete im Vergleich zur Java Standard Edition, wie etwa Java RMI. Dies erschwert die Benutzung von externen Bibliotheken oder macht diese gänzlich unmöglich. Aufgrund dessen wurde die Implementierung verschiedener Module, bei denen die Re-Implementierung bestehender Bibliotheken durch zu hohen Aufwand nicht sinnvoll wäre, nach dem Paradigma des Thin Client vorgenommen, um diese möglichst einfach und performant zu halten. Andere Pakete implementieren zudem gewisse Teile der Anwendungslogik im mobilen Endgerät, insbesondere um eine Unabhängigkeit von der Basis-Station zu erreichen.

Für die Umsetzung der Basis-Station wurde die Java Standard Edition in Verbindung mit dem Equinox OSGi Framework verwandt. Diese Kombination bietet einen großen Funktionsumfang und eine große Community. Durch die Verwendbarkeit von externen Bibliotheken zur Anbindung bestehender Dienste wie Twitter und Facebook auf der Basis Station, nicht jedoch auf den mobilen Endgeräten, bot sich eine Implementierung der Dienste dort an, um die Schnittstellen möglichst komfortabel nutzen zu können. Dies führt zu einer deutlich gesunkenen Einstiegshürde bei der Implementierung neuer Module und beschleunigt das Anfertigen von Prototypen. Inhalte dieser Module werden durch das Nachrichtensystem des Frameworks in komprimierter Form an das mobile Endgerät weitergereicht, das diese ohne weitere Verarbeitung sofort darstellen kann.

Auf der Basis dieses Frameworks wurde anschließend das grundlegende Bundle entwickelt das es ermöglicht, dass andere Komponenten (OSGi-Bundles) dynamisch installiert und genutzt werden können. Dabei ist die Komponente selbst durch zwei OSGi-Bundles realisiert. Eine Komponente dient nur zur Erzeugung und Darstellung der Oberfläche und die andere stellt die Fachlogik der Service-Komponente dar. Dadurch wird die Plattformunabhängigkeit gewährleistet.

Bei der Realisierung der Kontextsensitivität ist die Granularität der Umgebung durch das verwendete Übertragungsmedium (WLAN, Bluetooth, etc.) begrenzt. Die Kommunikation erfolgt über TCP/IP- oder UDP/IP-Protokolle. Im Verzeichnis des Brokers kann spezifiziert werden, welche Komponenten speziell in der aktuellen Umgebung nutzbar sind und ob Komponenten installiert werden müssen (Realisierung der Push-Semantik). Befindet sich der Nutzer bereits in einer Umgebung, während der Push einer Komponenten erfolgt, so wird dieser über spezielle Framework-Ereignisse darüber informiert und veranlasst die Installation. Zusätzlich können ortsunabhängige bzw. umgebungsunabhängige Komponenten angeboten werden,

die ebenfalls installiert und genutzt werden können (Realisierung Pull-Semantik initiiert durch den Client).

Beim Verlassen der Umgebung werden ortsabhängigen Dienste wieder entfernt. Bei der Konzipierung des Prototyps wurde vorgesehen, dass Komponenten lediglich gestoppt und nicht wieder vollständig entfernt werden, jedoch hat sich während der Implementierung herausgestellt, dass ein Entfernen die momentan praktikabelste Lösung darstellt. Dies hängt u.a. damit zusammen, dass die frühe Version des OSGi-Frameworks, das verwendet wurde, noch keine ausgereiften Mechanismen für eine Versionskontrolle der Komponenten angeboten hat. Um hieraus resultierende Inkonsistenzen zu vermeiden, werden Komponenten vollständig deinstalliert und grundsätzlich neu installiert. In späteren Versionen wäre die zunächst geplante Realisierung vorteilhaft um unnötige Übertragungen zu vermeiden.

Broker in einer Umgebung geben sich durch Broadcast-Nachrichten bekannt. Betritt ein Client die Umgebung so registriert sich der Client beim Broker und speichert den Broker zur Kennzeichnung der Umgebung ab. Der Broker sendet daraufhin dem Client sein Verzeichnis von Komponenten (in XML-Repräsentation). Dieser führt anschließend die vorgeschriebenen Installationen und ggf. optionale Installationen durch.

Beim Verlassen der Umgebung (oder bei Unerreichbarkeit des Brokers) wird durch einen Time-Out die Deinstallation der ortsabhängigen Komponenten veranlasst. Der Service-Broker kann gleichzeitig Service-Provider sein, d.h. er hält die zu installierenden Komponenten bereit. Unterscheiden sich Broker und Provider, so ist sicherzustellen, dass der Provider zum Zeitpunkt der Installation verfügbar ist. Neben der Möglichkeit die Installation von Services vom Broker initiieren zu lassen (Service Push), können weitere Broker manuell hinzugefügt werden.

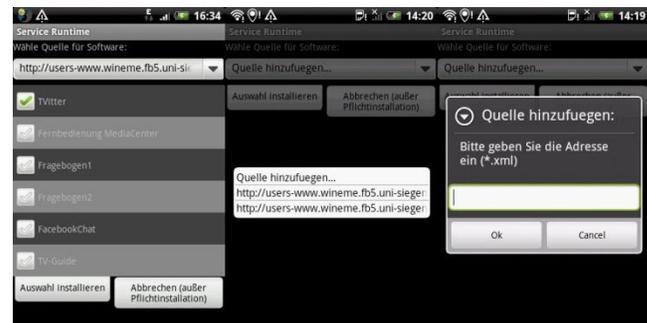


Abbildung 4: Verzeichnis, Quellenauswahl und Dialoges zum Hinzufügen von Quellen

4. ANWENDUNGSFALL: TTwitter

Um den entwickelten Prototypen evaluieren zu können, wurde ein Beispieldienst implementiert der in Verbindung mit einem TV-Gerät in Verbindung mit einem MediaCenter PC die einfache Einbindung externer Social-Network Dienste zeigen soll. Dies wurde exemplarisch am Beispiel des Microblogging Dienstes Twitter umgesetzt.

Die Applikation besteht aus drei Bundles, die zum einen die Anbindung an die reguläre Twitter API und zum anderen die

Anbindung an eine Streaming API bereitstellen, die Informationen der Twitter Timeline in einem simulierten Push-Verfahren in Echtzeit zur Verfügung stellt. Das dritte Bundle dient der Steuerung auf dem mobilen Endgerät auf dem Beiträge verfasst und versandt werden können. Dieses Bundle orientiert sich explizit am Thin-Client Paradigma, um die Plattformunabhängigkeit gewährleisten zu können. Die Interaktion mit dem Webservice Twitter geschieht allein auf der Basis-Station, die die entsprechenden Twitter-Bibliotheken nutzt und die notwendige Funktionalität bereitstellt. Auf dem Media Center PC wird zudem eine Applikation benötigt, die die Informationen der Basisstation annehmen und entsprechend darstellen kann. Dazu wird von der Basis-Station ein vorher spezifizierter Webservice aufgerufen, wobei die Implementierung für die Basis-Station keine Rolle spielt. Die Basis-Station kann auf dem gleichen Rechner laufen auf dem auch die Media Center Software läuft, so dass in diesem Fall beide Komponenten in einem Gerät vereint sind.

Ein beispielhafter Ablauf für eine Nutzung der Komponente könnte sich wie folgt darstellen: Der Anwender betritt den Raum des Media Centers und nutzt die TV Funktion um die Tagesschau anzusehen. In dieser wird über das Thema Steuererhöhungen berichtet, wodurch beim Anwender ein Interesse nach der Meinung Anderer geweckt wird, die die Sendung ebenfalls verfolgen. Dazu startet er auf seinem mobilen Endgerät die Anwendung TVitter, die sich bei der Streaming API mit dem aktuellen Sender als Suchbegriff registriert und die aktuellen Beiträge auf dem Bildschirm darstellt. Dies kann entweder in Form eines Overlay oder als separates Fenster erfolgen. Nun kann der Anwender sowohl die Meinung anderer Nutzer verfolgen als auch eigene Beiträge beisteuern, die den anderen Teilnehmern sowie ihm sofort angezeigt werden.

Auf technischer Ebene erzeugt das Starten der TVitter Komponente ein Ereignis auf dem mobilen Endgerät. Dieses Ereignis wird durch eine zentrale Routingkomponente über eine Socket-Verbindung an die Basis Station übergeben, die daraufhin über eine Webservice Schnittstelle die aktuellen Senderdaten der TV Applikation abrufen. Diese werden ebenfalls als Nachricht auf dem umgekehrten Weg an die TVitter Applikation zurückgegeben, die daraufhin dem Benutzer kontextuelle Vorschläge zur Nutzung der Twittersuche unterbreitet. Weiterhin kann der Benutzer automatisch die Senderinformationen zur Annotation seiner Beiträge nutzen, was bei Twitter üblicherweise semi-strukturiert über so genannte Hash-Tags geschieht (bspw. #Tagesschau). Wechselt der Benutzer den Sender, wird dies von der Basisstation an das mobile Endgerät übermittelt und entsprechend neue Vorschläge unterbreitet.

5. EVALUATION

Der entwickelte Prototyp wurde abschließend unter kontrollierten Bedingungen im Labor evaluiert. Aufgrund der noch fehlenden Dienste und Geräte war es in dieser Phase noch nicht möglich die Evaluation unter realweltlich-alltäglichen Bedingungen durchzuführen. Außerdem war es nicht möglich ein reales Setting nachzustellen, da auch hierfür eine minimale Anzahl an Services und unterschiedliche Home-IT Umgebungen notwendig gewesen wären. Für die Evaluation wurden daher bewusst Lead-User nach [13] ausgewählt, also Benutzer die

bereits ein grundlegendes technisches Wissen und Erfahrungen im Kontext neuer Unterhaltungs- und Kommunikationsmedien vorweisen können. Im Fokus der Evaluation stand demnach auch nicht die Verwendung des Systems im Anwendungskontext sondern mehr das Verständnis und die Sinnhaftigkeit des umgesetzten Konzepts sowie die Untersuchung des TVitter Dienstes.Durchführung

Die Evaluation wurde mit sechs Teilnehmern durchgeführt. Die Partizipanten wurden dazu in einem mit Fernseher und daran angeschlossenen Notebook ausgestatteten Raum interviewt und bei der Durchführung verschiedener Nutzungstests beobachtet. Die Befragung erfolgte auf der Basis semi-strukturierter Interviews mittels einem vorgegebenem Leitfaden, von dem auch abgewichen werden konnte. Um die Evaluation möglichst kontextnah zu gestalten, wurde diese in einem Labor durchgeführt, das einem Wohnzimmer nachempfunden ist (s. Abbildung und Bestandteil eines Living Lab-Konzepts an der Universität Siegen ist [12]). Alle Tests wurden für die spätere Analyse mit Audiogeräten aufgezeichnet sowie von einem passiven Beobachter protokolliert



Abbildung 5: Umgebung für Interviews und Evaluation

Der Test gliederte sich in drei Teile:

1. Pre-Test Interview
2. Nutzungstest des Prototyp
3. Post-Test Interview

Vor dem Test des Prototyps wurden demographische Informationen der Teilnehmer erfasst und diese zu ihren medialen Nutzungsgewohnheiten befragt. Hierbei lag der Fokus darauf, einen guten Überblick über die derzeitige Nutzungspraxis der Teilnehmer im Kontext von Home-IT und Medien zu gewinnen.

Nachfolgend wurde die prototypische Umsetzung präsentiert. Hierbei handelte es sich insbesondere um die Umsetzung der Direct Service Push Applikation und des TVitter Use-Case. Die Teilnehmer wurden mit dem jeweils in den Ausgangszustand versetzten Mobiltelefon gebeten den Raum und damit die Reichweite des WLANs zu verlassen, um kurz darauf wieder hereinzutreten. In diesem Moment wurden die zur Verfügung stehenden Services vorgeschlagen und installiert, im konkreten Nutzungstest ausschließlich die TVitter Anwendung. Diese wurde im weiteren Verlauf ausführlich erläutert und von den Teilnehmern genutzt. Dazu wurde beispielhaft eine Fernsehsendung eingeschaltet, die von anderen Twitter

Benutzern kommentiert wurde und zu der die Benutzer einen eigenen Beitrag verfassen sollten. Passend zu Sendung wurde mit der Wizard-of-Oz Methode geeignete Suchbegriffe an das Mobiltelefon des Nutzers gesandt. Die Teilnehmer konnten nun entscheiden diese zu übernehmen, zu erweitern oder ganz zu verwerfen um einen eigenen Suchbegriff zu verwenden. Durch das Absenden der Suchanfrage wurden entsprechende Twitter-Beiträge unter dem Fernsehbild in einer transparenten Applikation dargestellt. Während der Nutzung wurde stets mit den Teilnehmern kommuniziert, um zu erfahren wie diese im Moment interagieren, wie das Konzept und dessen Umsetzung empfunden wird und ob Hilfestellung benötigt wird.

Im Anschluss an den Nutzungstest wurden die Teilnehmer ein zweites Mal befragt. Hierbei lag der Fokus auf dem Grundkonzept der Infrastruktur und der konkreten Implementierung der medialen Integration am Beispiel von Fernsehen und Twitter. Die Teilnehmer wurden gebeten ihre Haltung bzgl. der gezeigten Anwendung, Usability Aspekte und alternative Anwendungsszenarien zu erörtern. Hierbei wurden konkrete Vorstellungen und Änderungswünsche der Teilnehmer herausgearbeitet und detailliert besprochen.

5.1 Teilnehmer

Insgesamt wurden sechs Teilnehmer während der Evaluation befragt (s. Tabelle 1). Bei der Auswahl der Teilnehmer wurden keine besonderen Kriterien berücksichtigt, jedoch wurden bewusst Personen ausgewählt die eine hohe Technik-Affinität sowie Erfahrungen im Kontext neuer Unterhaltungs- und Kommunikationsmedien vorweisen konnten, weshalb die hier betrachtete Gruppe von Nutzern als Lead-User bezeichnet werden kann [13].

Tabelle 1: Anonymisierte Übersicht der Teilnehmer

Name	Beruf	Alter
Achim	Student	23
Nikolas	Student	25
Tobias	Doktorand	30
Andre	Doktorand	30
Kristofer	Student	24
Andrea	Fachleiterin	34

Die Teilnehmer waren im Durchschnitt 27,7 Jahre alt, besaßen alle einen hohen Bildungsgrad und waren an neuen Technologien und an Medien interessiert. Die Probanden arbeiten zum Teil auch in der Forschung oder Entwicklung, waren aber nicht an der vorliegenden Arbeit beteiligt. Alle sehen regelmäßig fern und nutzen unabhängig davon verschiedene soziale Netzwerke wie Facebook, Twitter oder Wer-Kennt-Wen.

5.2 Ergebnisse

Die Ergebnisse lassen sich in die folgenden drei Kategorien einteilen:

1. Anwendungsbezogen
2. Interfacebezogen
3. Soziologisch

Bei den anwendungsbezogenen Ergebnissen standen Usabilityaspekte im Vordergrund. Diese wurden durchweg als unzureichend und unbefriedigend empfunden. Grundlegende Kriterien der Dialoggestaltung wurden bemängelt und die Implementierung als sehr technisch empfunden. Vor dem Hintergrund, hier einen eher technisch orientierten Prototyp zum Test gestellt zu haben, waren diese Ergebnisse zu erwarten. In kommenden Versionen werden diese berücksichtigt und erneut überprüft bevor der Prototyp unter realen Bedingungen evaluiert wird.

Die zweite Ergebniskategorie bezieht sich auf Bedienung und Darstellung. Hierbei wird zwischen Ein- und Ausgabeinterface unterschieden, insbesondere weil am Beispiel der TVitter-Anwendung eine dichotome Trennung vorliegt. Auf der Eingabeseite wurden insbesondere die folgenden drei Aspekte angesprochen:

1. Textuelle vs. sprachliche Eingabe
2. Mobiltelefon als Mittler
3. Formfaktor für textuelle Eingabe

Ein Teilnehmer stellte gleich zu Beginn der Evaluation das Konzept der textuellen Eingabe in Frage, da diese im Kontext des Fernsehkonsums für sein Empfinden vermutlich nicht die optimale Form der Interaktion darstelle. Statt einer textuellen Eingabe wurde eine sprachgesteuerte Interaktion vorgeschlagen, abhängig vom Stand der Technik im Bereich Spracherkennung. Diese würde dem Teilnehmer als geringere Nutzungshürde erscheinen.

„...deshalb weiß ich nicht ob tippen hier die richtige Eingabeform ist, weil es zu lange dauert und das Fernsehen zu schnell ist...“ - Tobias

Das Mobiltelefon wurde weitgehend als Eingabemedium akzeptiert. Durch die ständige Verfügbarkeit erschien dieses Gerät als ideale Verknüpfung zu lokalen Technologien. Auch die Eingabeform mithilfe von Tastatur oder Handydisplay wurde als akzeptabel, wenn auch nicht ideal bewertet. Ein Trade-Off zwischen ständiger Verfügbarkeit und komfortabler Eingabe wurde von den Teilnehmern beschrieben und als solcher akzeptiert.

„...gerade wenn man das Gerät immer bei sich haben will, weil es eben location-basiert ist, denke ich ist ein Handy die beste Wahl...“ - Andre

Alternativ wurden von den Teilnehmern Eingabegeräte wie eine erweiterte Fernbedienung, klassische Tastaturen und Tablet Computer genannt. In allen Fällen wurden diese Eingabeformen als optional gesehen und stellten für die Teilnehmer eine Optimierung dar.

„Da habe ich nicht den Overkill einer normalen Tastatur.“ – Achim

Bei der Ausgabeform wurde wiederholt auf die Intrusivität der Einblendung von Informationen hingewiesen. Die Benutzer empfanden die Vorstellung der Überlagerung des Fernsehbildes durch die Nachrichten als störend und würden selber Steuern wollen in welcher Form diese Nachrichten angezeigt werden. Dabei sollten Aspekte wie Art, Dauer und Intensität der Darstellung durch den Benutzer selbst einstellbar sein.

„...wenn ich etwas sehe was mich interessiert, dann will ich glaube ich keine Einblendungen darin haben...es hängt dann glaube ich viel davon ab, wie intrusiv das Ganze ist...“ – Tobias

„... die Informationen kurz ein- und dann wieder ausblenden...wenn es die ganze Zeit bleibt stört es glaube ich.“ – Kristofer



Abbildung 5: Fernseher mit laufender Sendung, Twitter Stream und Debugging-Ausgabe auf Notebook

Bei der dritten Kategorie handelt es sich um soziologische Aspekte insbesondere im Kontext der sozialen Netzwerke. Aspekte wie Datenschutz, Privatsphäre und bisherige Nutzungspraxis stellen wichtige Kriterien für die Nutzung einer Plattform wie der hier vorgestellten dar. Für einen der Teilnehmer waren insbesondere die Verwendung der von ihm bereitgestellten Daten ein wichtiger Aspekt. Hier stand die Sorge um eine kommerzielle Nutzung der Daten im Vordergrund. Weitere Aspekte wie die Zugänglichkeit der Beiträge wurden ebenfalls als wichtig beschrieben, jedoch nicht als K.O. Kriterium. Bei den befragten Teilnehmern zeichnete sich ein deutliches Bewusstsein für die Öffentlichkeit von sozialen Netzwerken wie Facebook oder Twitter ab, wodurch auch bei fehlenden Einstellungsmöglichkeiten zur Privatsphäre die Selbstreflexion als Selektionsmechanismus genutzt würde.

„Fernsehen ist für mich eine ganz private Sache. ... Ich müsste wissen, dass das jetzt nicht genutzt wird um Benutzerprofile zu erstellen.“ - Andre

„Ich wäre nicht bereit dafür viel Geld auszugeben, deshalb hätte ich die Vermutung wenn es umsonst angeboten würde, dass die Unternehmen mit Werbung Geld verdienen möchten. Das wäre es mir vielleicht dann nicht wert.“ - Andre

Als ein ebenfalls wichtiger Aspekt stellte sich die Privatsphäre heraus, die von allen Teilnehmern angesprochen wurde. Hierbei standen die Sichtbarkeit der Beiträge und die Definition der Gruppe im Vordergrund sowie die bisherige Nutzungspraxis bzw. Zugehörigkeit zu sozialen Netzwerken. Die Steuerung der Sichtbarkeit der Beiträge wurde regelmäßig als ein zu bedenkender Faktor genannt, jedoch nicht als Ausschlusskriterium empfunden. Einige Teilnehmer sahen einen öffentlichen Raum als ideal an, andere sahen einen Nutzen primär im bestehenden sozialen Netzwerk.

„...grundsätzlich würde ich immer versuchen so anonym wie möglich zu sein wenn es denn öffentlich ist, aber für mich liegt der Mehrwert tatsächlich im persönlichen Bereich...“ - Andrea

Die Teilnehmer wurden auch gebeten, losgelöst vom konkreten Anwendungsfall, also der Integration von Fernsehen und Twitter im Wohnbereich, weitere Szenarien darzustellen in denen sie das zu Grunde liegende Konzept der dynamischen Home-IT als nützlich empfinden würden. Neben diversen Hinweisen die stark im Zusammenhang mit klassischen Home-Automation Ansätzen lagen, wurden die zwei folgenden Szenarien aufgezeigt. Zum einen wurde eine Verwendung im semi-öffentlichen Bereich vorgeschlagen, also beispielsweise einer Messe. Hier könnte einem geschlossenen Benutzerkreis in einem vordefinierten Raum durch dynamische Erkennung der Location ein Mehrwert geboten werden. Ein anderes Szenario zielte auf die Integration weiterer Medien wie Musik, die dem Benutzer im Haus folgen könne.

„...die Musik folgt mir sozusagen. Das fände ich sehr praktisch.“ - Tobias

5.3 Methodische Reflexion

Für die Evaluation wurden insgesamt sechs Teilnehmer persönlich befragt und während der Nutzung des Prototyps beobachtet. Die Auswahl der Teilnehmer war nicht an spezifische Eigenschaften oder Kenntnisse gebunden. Um die Evaluation möglichst realitätsnah zu gestalten wurden diese in einem wohnzimmerähnlichen Raum durchgeführt. Durch die technische Orientierung des Prototyps wurde im zweiten Teil der Evaluation, der Benutzer stark geführt, um die Funktionalität zu verdeutlichen und Missverständnissen vorzubeugen.

Die Benutzer können aufgrund ihres Wissens und ihrer Erfahrung als Lead-User bezeichnet werden und so sind die Ergebnisse der Studie entsprechend zu interpretieren. Aus einer Genderperspektive betrachtet, wurden 5 Männer und lediglich eine Frau befragt. Insbesondere die unterschiedliche Nutzung von sozialen Netzwerken sollte hier berücksichtigt werden. Durch die Evaluation unter Laborbedingungen sind evtl. andere Arten der Nutzung beobachtet worden, als sie im tatsächlichen häuslichen Umfeld der Teilnehmer beobachtet worden wäre. Weiter wurde im konkreten Fall keine Vorstudie der tatsächlichen häuslichen Nutzungspraxis vorgenommen, sondern nur eine Befragung hierzu. Bei zukünftig geplanten Studien innerhalb des Siegener Living Labs können diese Aspekte berücksichtigt werden. Beim konkreten Test der TVitter Implementierung wurden die Teilnehmer stark geführt und so möglicherweise in ihrer Nutzungspraxis zu stark beeinflusst, weshalb viele Beobachtungen aus dieser Phase der Evaluation nicht Einzug in diese Studie halten.

5.4 Fazit

In der hier vorliegenden Studie der prototypisch implementierten dynamischen Home-IT Infrastruktur konnten wichtige Erkenntnisse zur Nutzung und Empfindung erfasst werden. Es wurden grundlegende Informationen zu technischen und soziologischen Aspekten des Konzeptes und der konkreten Implementierung gesammelt, die für weitere Forschung in diesem Feld wichtig und hilfreich sind. Grundlegend wurde das Konzept der dynamischen Home-IT positiv bewertet, ebenso wie die konkrete Implementierung des Anwendungsfalls TVitter.

„Ich finde das sehr gut, denn es kommt meinem Bedürfnis grundsätzlich entgegen, in dem Sinne das ich ja das Notebook normalerweise an habe.“ – Andrea

Bei der Auswertung der Interviews zeigten sich insbesondere die folgenden Punkte als wichtige Kriterien für die Weiterentwicklung:

1. Usability
2. Ein- und Ausgabe
3. Privatsphäre und Datenschutz

Diese Aspekte sollen bei einer weitergehenden Implementierung des Frameworks berücksichtigt und in weiteren Tests erprobt werden.

6. AUSBLICK

Das von uns entwickelte Framework bietet die Möglichkeit Dienste in Home-IT Umgebungen bereitzustellen und verschiedene Medien zu integrieren. Bei der Implementierung haben wir auf einem komponentenbasierten Modell aufgesetzt, orientieren uns aber bei der Interaktion an serviceorientierten Architekturen. Durch die Dynamik beider Paradigmen ist das Framework in der Lage komplexere Funktionalität in Abhängigkeit von Ort bzw. vorhandener Infrastruktur anzubieten und medienübergreifend nutzbar zu machen. Gleichzeitig wird auf diese Weise eine Plattformunabhängigkeit gewährleistet, wodurch der Einsatz in heterogenen Landschaften erleichtert wird. Exemplarisch wurde die Komponente TVitter entwickelt, um das Framework im konkreten Anwendungsfall vorstellen und testen zu können. Im Hinblick auf die Evaluation des Frameworks sollen in anschließenden Arbeiten neben der TVitter Anwendung weitere Komponenten entwickelt werden, um eine sinnvolle Nutzung des Frameworks zu ermöglichen. Zusätzlich zu der hier durchgeführten Evaluation soll anschließend in Testhaushalten eine Langzeitevaluation durchgeführt werden. Hierzu wurden 16 Haushalte, die ebenfalls Teil des Siegener Living Labs sind, ausgewählt und mit Media Center PCs und Smartphones ausgestattet. Über eine Projektlaufzeit von etwa zwei Jahren sollen kontinuierlich weitere Komponenten entwickelt und über das Framework den Testhaushalten bereitgestellt werden.

7. DANKSAGUNG

Wir danken Stefan Hering für die Mitarbeit bei der Entwicklung des Prototyps sowie Joachim Ritter von Prosys für seine Unterstützung bei Fragen bezüglich des OSGi Frameworks.

Diese Arbeit wurde gefördert vom Ministerium für Innovation, Wissenschaft, Forschung und Technologie des Landes NRW mit Mitteln der Europäischen Union sowie der DFG.

8. LITERATUR

- [1] Apple Inc. About Apple Push Notification Service. *About Apple Push Notification Service*, 2010. <http://support.apple.com/kb/HT3576>.
- [2] Betz, M., Ley, B., Pipek, V., and Wulf, V. Folksonomies for real things: Tagging objects with RFID as a source for context-awareness. *Informatik 2008 - Beherrschbare Systeme dank Informatik*, Springer Verlag (2008), 787-794.
- [3] Brown, P.F., Metz, R., and Hamilton, B.A. Reference model for service oriented architecture 1.0. (2006).
- [4] Cervantes, H. and Hall, R.S. Autonomous Adaptation to Dynamic Availability Using a Service-Oriented

Component Model. *Proceedings of the 26th International Conference on Software Engineering*, IEEE Computer Society (2004), 614-623.

- [5] Cheng, Y.-M. Using ZigBee and Room-Based Location Technology to Constructing an Indoor Location-Based Service Platform. *Intelligent Information Hiding and Multimedia Signal Processing, International Conference on*, IEEE Computer Society (2009), 803-806.
- [6] Crockford, D. JSON. *RFC-4627 JSON*, 2006. <http://www.ietf.org/rfc/rfc4627.txt?number=4627>.
- [7] Dey, A.K. Understanding and Using Context. *Personal Ubiquitous Comput.* 5, 1 (2001), 4-7.
- [8] Google. Android Cloud to Device Messaging Framework - Google Projects for Android. *Google Projects for Android: C2DM*, 2010. <http://code.google.com/intl/de-DE/android/c2dm/index.html>.
- [9] Gu, T., Pung, H.K., and Zhang, D.Q. Toward an OSGi-Based Infrastructure for Context-Aware Applications. *IEEE Pervasive Computing* 3, 2004, 66-74.
- [10] Harboe, G., Massey, N., Metcalf, C., Wheatley, D., and Romano, G. The uses of social television. *Comput. Entertain.* 6, (2008), 8:1-8:15.
- [11] Heineman, G.T. and Councill, W.T. *Component-Based Software Engineering: Putting the Pieces Together*. Addison-Wesley Longman, Amsterdam, 2001.
- [12] Hess, J. and Ogonowski, C. Steps toward a living lab for socialmedia concept evaluation and continuous user-involvement. *Proceedings of the 8th international interactive conference on Interactive TV&Video*, ACM (2010), 171-174.
- [13] Hippel, E. von. Lead Users: A Source of Novel Product Concepts. *MANAGEMENT SCIENCE* 32, 7 (1986), 791-805.
- [14] Kiesler, S., Zhang, H., Ren, Y., Kraut, R.E., Konstan, J.A., and Weisz, J.D. Watching together: integrating text chat with video. 877.
- [15] Luimula, M. and Kuutti, K. Locawe: a novel platform for location-aware multimedia services. *Proceedings of the 7th International Conference on Mobile and Ubiquitous Multimedia*, ACM (2008), 122-129.
- [16] Nichols, J. and Myers, B.A. Creating a lightweight user interface description language: An overview and analysis of the personal universal controller project. *ACM Trans. Comput.-Hum. Interact.* 16, 4 (2009), 1-37.
- [17] Oehmichen, E. and Schröter, C. Zur Differenzierung des Medienhandelns der jungen Generation. *Media Perspektiven* 2009, 2009.
- [18] OSGi. OSGi Alliance. *OSGi Specification*, 2010. <http://www.osgi.org/Specifications/HomePage>.
- [19] Pascoe, J. Adding Generic Contextual Capabilities to Wearable Computers. (1998), 92--99.
- [20] Prosys. *Prosys OSGi Framework*. Prosys, 2008.
- [21] Redondo, R.P.D., Vilas, A.F., Cabrer, M.R., and Pazos, J.J. Exploiting OSGi capabilities from MHP applications. *Journal of Virtual Reality and Broadcasting* 4, (2007), 16.
- [22] Sanchez-Nielsen, E., Martin-Ruiz, S., and Rodriguez-Pedrianes, J. An open and dynamical service oriented architecture for supporting mobile services. *Proceedings of the 6th international conference on Web engineering*, ACM (2006), 121-128.

- [23] Schilit, B., Adams, N., and Want, R. Context-Aware Computing Applications. *IN PROCEEDINGS OF THE WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS*, (1994), 85--90.
- [24] Schumacher, J., Feurstein, K., and Pitkänen, O. *European Living Labs: A new approach for human centric regional innovation*. wvb Wissenschaftlicher Verlag Berlin, 2008.