



UNIVERSITÄT ZU LÜBECK

Schlussbericht
BMBF Forschungsvorhaben

SmartAssist
**Plattform zur Unterstützung von sozialen und
gesundheitlichen Aspekten bei der
Gestaltung eines altersgerechten autonomen Lebens**

Teilvorhaben
Technische Plattformentwicklung und Signalauswertung

Autoren: Stefan Fischer, Darren Carlson, Peter Rothenpieler, Andreas Schrader
Institut für Telematik, Universität zu Lübeck
Alfred Mertins, Dierk Matern
Institut für Signalverarbeitung, Universität zu Lübeck
Jens-Martin Träder
Institut für Allgemeinmedizin, Universität zu Lübeck

Laufzeit 08/2009 – 07/2012

des Vorhabens: kostenneutral verlängert bis 12/2012

Förderkennzeichen: 16KT0942

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

Inhaltsverzeichnis

I.	Kurze Darstellung	3
1.	Aufgabenstellung	3
2.	Voraussetzungen	3
3.	Planung und Ablauf des Vorhabens	8
4.	Stand von Wissenschaft und Technik	11
4.1.	Zielgruppen von SmartAssist.....	11
4.2.	Standards.....	14
4.3.	Sensorauswertung	21
4.4.	Kontext.....	21
5.	Zusammenarbeit mit anderen Stellen.....	23
II.	Eingehende Darstellung	24
1.	Resultate des Forschungsvorhabens.....	24
1.1.	Serviceportal	24
1.2.	Sensordatenanalyse	36
1.3.	Mobile Plattform.....	42
1.4.	Testbetrieb und Evaluation	65
1.5.	Medizinische Begleitforschung	65
2.	Die wichtigsten Positionen des zahlenmäßigen Nachweises	68
3.	Notwendigkeit und Angemessenheit der geleisteten Arbeit	69
4.	Voraussichtlicher Nutzen und Verwertbarkeit der Ergebnisse	73
4.1.	Wirtschaftliche Erfolgsaussichten nach Projektende.....	73
4.2.	Wissenschaftlich-technische Erfolgsaussichten nach Projektende.....	74
4.3.	Wissenschaftliche und wirtschaftliche Anschlussfähigkeit.....	74
5.	Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen	75
III.	Veröffentlichungen.....	81
1.	Workshops und Konferenzen	81
2.	Vorträge.....	82
3.	Internationale Preise	83
4.	Studentische Arbeiten	83
5.	Events	84
6.	Presseberichte.....	85
IV.	Quellen	85

I. Kurze Darstellung

1. Aufgabenstellung

Im Rahmen der Förderprogramms „Altersgerechte Assistenzsysteme für ein gesundes und unabhängiges Leben“ wurden ganzheitliche Lösungen, die auf neuen Technologien und Dienstleistungen basieren, vom Bundesministerium für Bildung und Forschung gefördert. SmartAssist ist eines von insgesamt 18 geförderten Projekten. Ziel des Projekts SmartAssist ist die Schaffung einer Plattform zur Unterstützung des altersgerechten autonomen Lebens. Anstelle einer frühzeitigen Einweisung in ein Heim, in dem eine dauerhafte Überwachung geleistet wird, soll eine bedarfsorientierte Betreuung alter Menschen durch Paten erfolgen. Solche Patenschaften können Freunde, Verwandte, Bekannte und Nachbarn, ebenso wie kommerzielle Dienstleister übernehmen.

Der im Projekt verfolgte Ansatz sieht den Einsatz von Sensoren im häuslichen Umfeld vor, die als Basis für die Erkennung dieser gesundheitsrelevanten Zustände auf einem Internet-basierten Serviceportal dienen. Ausgehend von dem Portal erfolgt dann die automatische Benachrichtigung der Paten. So wird dieses Portal zu einem Marktplatz für eine Vielzahl von unterschiedlichen Diensten und Anwendungen rund um das Thema „Assisted Living“, der auch Drittanbietern offen steht. Diese Dienste sollen eine alters- und situationsgerechte Unterstützung von Menschen im eigenen Wohnumfeld ermöglichen, wobei das Serviceportal lediglich als Plattform für diese Dienste genutzt werden soll. Es sollen Schnittstellen für Drittanbieter zur Verfügung gestellt werden, um diese Dienste auf einfache Weise in das Portal zu integrieren. Das Portal bildet somit die technische Infrastruktur, um Dienste rund um das Thema „Assisted Living“ anzubieten. So können sich Kunden hier anmelden, Dienste suchen und sich für deren Nutzung registrieren. Sie können aber auch Funktionen zur sozialen Interaktion mit anderen Plattform-Teilnehmern nutzen. Als weitere Unterstützungskomponente wird eine Systemarchitektur geschaffen, welche die Entwicklung von Assistenzanwendungen auf mobilen Geräten (Mobiltelefone und Tablets) unter Anbindung an die Serviceplattform ermöglicht. Hierzu wird eine neuartige Softwareplattform geschaffen, die sich an dynamische und heterogene Umgebungen ohne fest bekannte Instrumentalisierung anpasst und so auch mobile Dienste in wechselnden Kontexten und Domänen ermöglicht (z.B. in der Arztpraxis, im Krankenhaus oder auf Reisen).

Das System soll in einem Feldversuch in ausgewählten Wohnungen in Lübeck und Umgebung getestet werden. Es soll ebenfalls eine medizinische Begleitforschung bezüglich der positiven Auswirkungen auf die Lebensqualität durchgeführt werden. Dazu wird vom Institut für Allgemeinmedizin ein geriatrisches Basis-Assessment bei allen Probanden und Kontrollpersonen durchgeführt.

2. Voraussetzungen

Die zunehmende Überalterung unserer Gesellschaft und der verstärkte Wunsch nach selbständigem Wohnen im Alter stellen hohe Anforderungen an unser Pflege- und Dienstleistungssystem. Entscheidend ist ein gut funktionierendes soziales System, das einem im Bedarfsfall die notwendige Unterstützung gewährleistet. Ein wesentlicher Hinderungsfaktor für ein selbstbestimmtes und autonomes Leben ist die zunehmende Individualisierung unserer Gesellschaft, die im Alter schnell zur Vereinsamung führt. Wesentliche Gründe sind zum einen die zunehmende Zahl von Ein-Personen-Haushalten und zum anderen die ständig steigende geographische Verteilung von Familien oft bis ins Ausland. Der hohe Kostendruck im Bereich der sozialen Dienstleistungen erhöht zusätzlich den Bedarf an unterstützenden technischen Systemen.

Das Projekt SMART ASSIST realisiert ein sozio-technisches System zur Unterstützung von älteren Menschen in der eigenen Wohnung bis ins hohe Alter. Wesentliches Ziel ist die engere Vernetzung der bestehenden sozialen Systeme durch den Einsatz technischer Hilfsmittel, mit deren Hilfe schleichende Verschlechterungen des Gesundheitszustands erkannt, vorbeugende Maßnahmen gegen Krankheiten ergriffen und damit ein vorzeitiger Umzug ins Pflegeheim vermieden werden kann. Durch die Einbindung aller relevanten Akteure (Haushalt, Nachbarschaft, Freundeskreis, Familie, Ärzte, Betreuungseinrichtungen, Dienstleistungsunternehmen, etc.) werden das soziale Netzwerk gestärkt, sowie Versorgung und Dienstleistungen optimiert.

Bisherige Systeme erreichen diese Ziele nur sehr bedingt. Einerseits wird in den meisten Fällen die Datenerfassung mit einem hohen Grad an Instrumentalisierung der Gebäude erkaufte. Andererseits wird durch den Einsatz von sehr direkten Sensoren (Kamera, Mikrofon, etc.) die Akzeptanz erschwert.

Ziel im vorliegenden Verbundprojekt ist daher die Entwicklung eines Systems, das die Technik durch einfache und kostengünstige Komponenten mit drahtloser Vernetzung in den Hintergrund treten lässt und durch intensiven Einsatz von Algorithmen zur Merkmalsextraktion auf den Einsatz von direkten Beobachtungssensoren verzichten kann. Durch die Verwendung von Standardkomponenten und die Öffnung der Schnittstellen wird eine hohe Kosteneffizienz erzielt und nicht zuletzt durch die Unterstützung sozialer Plattformen und Kommunikationskanäle der menschliche Aspekt der Betreuung in den Vordergrund gerückt.

Das Ziel war, 50 Probanden und 50 Kontrollpersonen (>70 Jahre, möglichst allein lebend, nicht in Pflegeheimen, sondern maximal in betreutem Wohnen untergebracht) zu finden, die bereit waren, ihr Einverständnis zur Studienteilnahme über einen Zeitraum von mindestens zwei Jahren zu geben.

Das Projekt SmartAssist wurde in der Zeit von 08/2009 bis 12/2012 von mehreren Lübecker Partnern gemeinsam bearbeitet.

- **Lübecker Wachunternehmen:** Koordinator, Notrufzentrale und Betrieb der Anwendungen



Das Lübecker Wachunternehmen (LW) wurde 1904 gegründet und ist damit eines der ältesten Wach- und Sicherheitsunternehmen Deutschlands. Neben den klassischen Sicherheitsdienstleistungen wird seit etwa 30 Jahren auch eine Notruf- und Service-Leitstelle betrieben. Auf die-

se wurden schon sehr früh Hausnotrufsysteme aufgeschaltet. Seit 1992 führt das LW die Hintergrundtätigkeit des Hausnotrufes im Rahmen der Pflegeversicherung für die Vorwerker Diakonie durch. Hier gibt es daher Erfahrung sowohl mit der Aufschaltung von Hausnotrufgeräten, den betroffenen Personen und deren Verhaltensänderung bei sich erhöhendem Pflegebedarf als auch mit der Pflegeeinrichtung und der Feuerwehr. Darüber hinaus hat die Schwesterfirma CProjekt GmbH mit sicherheitsrelevanten Systemen im Bereich der GPS-Ortung Erfahrung sowohl mit der entsprechenden Hardware als auch mit der Übertragungs- und Servertechnik vornehmlich unter GPRS.

- **Vorwerker Diakonie:** Integration der Dienste in eigene Einrichtungen



Die Vorwerker Diakonie ist ein Zusammenschluss unterschiedlicher Einrichtungen der Alten, Behinderten-, Jugend-, Sucht- und Gefährdetenhilfe sowie von Einrichtungen für psychisch Kranke und der Fachklinik für Kinder- und Ju-

gendpsychiatrie und Psychotherapie. Die Einrichtung entstand zu Beginn des 20. Jhd. und ist mit ca. 1600 hauptamtlichen Mitarbeitern der drittgrößte Arbeitgeber der Hansestadt Lübeck. Die Angebote umfassen Beratung und Diagnostik, Wohnen und Assistenz, Bildung und Arbeit, Therapie und Betreuung. Der Betrieb erfolgt im Einklang zwischen betriebswirtschaftlichem Handeln und diakonischer Hilfeleistung aus einer christlichen Grundhaltung. Die Diakonie bietet Unterstützung unabhängig von religiösem Bekenntnis, kultureller Herkunft, Weltanschauung und sozialer Stellung eines Menschen. Das gemeinsame Ziel aller Einrichtungen ist die Begleitung, Förderung und Hilfe von Menschen in ihren spezifischen Lebenssituationen.

- **coalesenses GmbH:** Technische Entwicklung (Hardware)

coalesenses

Die coalesenses GmbH ist ein junges Unternehmen, das Lösungen im Bereich massiv verteilter Systeme anbietet. Ziel ist es, Technologien aus der aktuellen Forschung zur Marktreife zu führen. Auf diese Weise werden überdurchschnittlich innovative Produkte entwickelt und frühzeitig am Markt platziert. Das Unternehmen wurde von Forschern gegründet, die auf diesem Gebiet mehrere Jahre im universitären Umfeld gearbeitet haben. Diese Herkunft wird explizit als Stärke angesehen, und die Zusammenarbeit mit öffentlichen wie privaten Einrichtungen der Spitzenforschung gepflegt. Die coalesenses GmbH hat in den vergangenen Jahren eine modulare Hard- und Softwareplattform für funkvernetzte Sensorkomponenten entwickelt und bietet darauf basierend kundenspezifische Messsysteme an.

- **Institut für Telematik der Universität zu Lübeck:** Systemarchitektur, Portal, mobile Komponenten



UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR TELEMATIK

Das Institut für Telematik (ITM) ist an der Technisch-Naturwissenschaftlichen Fakultät der Universität zu Lübeck angesiedelt und wird von Prof. Dr. Stefan Fischer geleitet. Mit seinen zurzeit etwa 15 Wissenschaftlern arbeitet es vor allem an Fragen der Vernetzung von Computern aller Art in modernen Informationssystemen, insbesondere im Umfeld des Ubiquitous/Pervasive Computing. Zurzeit stehen vor allem Projekte im Umfeld von Sensornetzen (Algorithmik, Programmierung, Middleware, Lokalisierung, praktische Realisierung), Serviceorientierte Architekturen (SOA) sowie die Entwicklung von adaptiven Infrastrukturen für kontext-sensitive Dienste im Vordergrund. Das Institut hat sich mittlerweile eine große Kompetenz in Aufbau und Betrieb größerer Sensornetze erworben. Beispiele sind die Projekte MarathonNet (mobiles Sensornetz bei einem Marathon-Lauf), FleGSens (Überwachung einer langen grünen Grenze) und WISEBED (europaweiter Verbund von Sensornetzen). Das Teilvorhaben des Instituts für Telematik befasste sich mit der Konzeption und Entwicklung des internetbasierten Service-Portals sowie der Kommunikation des Portals mit den Sensorkomponenten und Mobilgeräten. Des Weiteren wurde vom Institut für Telematik eine Dienstplattform implementiert, welche Basisdienste zur Verfügung stellt und die Einbindung externer Dienste von Drittanbietern ermöglicht. Hierfür wurden sichere Mechanismen zur Authentifizierung und Autorisierung in die Dienstplattform integriert und diese um Konzepte der „Service Oriented Architecture“ erweitert. Zu Beginn der Projektlaufzeit hatte das Institut für Telematik bereits mehrere aktuelle Projekte im Umfeld von Sensornetzen (Algorithmik, Programmierung, Middleware, Lokalisierung, praktische Realisierung) sowie orthogonal dazu zu Service-orientierten Architekturen (SOA) abgeschlossen.

Die Kompetenz des Instituts auf den bearbeiteten Gebieten wird auch durch eine Vielzahl von Publikationen sowie eine Reihe von Lehrbüchern belegt¹.

- **Institut für Signalverarbeitung der Universität zu Lübeck:** Mustererkennung und Merkmalsextraktion



Das Institut für Signalverarbeitung (ISIP) ist ebenfalls an der Technisch-Naturwissenschaftlichen Fakultät der Universität zu Lübeck angesiedelt und wird von Prof. Dr.-Ing. Alfred Mertins geleitet. Zu den Forschungsgebieten des Instituts zählen die statistische Mustererkennung, Signal- und Bildverarbeitung sowie die Biosignalverarbeitung und das Neuro-Engineering. Im Zentrum der Untersuchungen stehen dabei meist Fragestellungen nach der optimalen Merkmalsextraktion und statistischen Analyse von Sensordaten, Sprache und Bildern. In einem von der DFG geförderten Grundlagenprojekt werden zum Beispiel Invarianzprinzipien für die Merkmalsextraktion in der sprecherunabhängigen automatischen Spracherkennung untersucht. In einem von der Innovationsstiftung Schleswig-Holstein geförderten Vorhaben werden Methoden für die optimale Klassifikation von sich gegenseitig beeinflussenden und an mehreren Sensoren beobachtbaren Objekten untersucht. Weitere Beispiele sind die Klassifizierung von Krankheitszuständen aus Ultraschallaufnahmen und die industrielle Fertigungsüberwachung mit Methoden der Bildverarbeitung. Die wissenschaftlichen Publikationen befassen sich entsprechend im Wesentlichen mit mathematischen Methoden der Signalverarbeitung, der Merkmalsextraktion sowie der Klassifikation².

Das Institut für Signalverarbeitung (ISIP) ist ebenfalls an der Technisch-Naturwissenschaftlichen Fakultät der Universität zu Lübeck angesiedelt und wird von Prof. Dr.-Ing. Alfred Mertins geleitet. Zu den Forschungsgebieten des Instituts zählen die statistische Mustererkennung, Signal- und Bildverarbeitung sowie die Biosignalverarbeitung und das Neuro-Engineering. Im Zentrum der Untersuchungen stehen dabei meist Fragestellungen nach der optimalen Merkmalsextraktion und statistischen Analyse von Sensordaten, Sprache und Bildern. In einem von der DFG geförderten Grundlagenprojekt werden zum Beispiel Invarianzprinzipien für die Merkmalsextraktion in der sprecherunabhängigen automatischen Spracherkennung untersucht. In einem von der Innovationsstiftung Schleswig-Holstein geförderten Vorhaben werden Methoden für die optimale Klassifikation von sich gegenseitig beeinflussenden und an mehreren Sensoren beobachtbaren Objekten untersucht. Weitere Beispiele sind die Klassifizierung von Krankheitszuständen aus Ultraschallaufnahmen und die industrielle Fertigungsüberwachung mit Methoden der Bildverarbeitung. Die wissenschaftlichen Publikationen befassen sich entsprechend im Wesentlichen mit mathematischen Methoden der Signalverarbeitung, der Merkmalsextraktion sowie der Klassifikation².

- **Zentrum für Bevölkerungsmedizin und Versorgungsforschung:** Interpretation medizinischer Daten, Merkmalsextraktion



UNIVERSITÄT ZU LÜBECK
ZENTRUM FÜR BEVÖLKERUNGSMEDIZIN
UND VERSORGUNGSFORSCHUNG



Das Institut für Allgemeinmedizin im Rahmen der Aktivitäten des Zentrum für Bevölkerungsmedizin und Versorgungsforschung (ZBV) beinhaltet den Pflichtkurs Allgemeinmedizin, das Blockpraktikum in ausgewählten Lehrpraxen, die Durchführung eines Wahlteriales des Praktischen Jahres in ausgewählten akkreditierten Lehrpraxen und den Kurs „Medizin des Alterns und des alten Menschen“ (Querschnittsfach). Prof. Träger hat ferner ein Curriculum zum Erwerb der Zusatzqualifikation „Geriatric“ an der Akademie für medizinische Fort- und Weiterbildung der Ärztekammer Schleswig-Holstein entwickelt und diese Kurse mehrfach durchgeführt. Aus dieser Fortbildungsreihe resultiert eine gute Zusammenarbeit mit anderen medizinischen Disziplinen, vor allem mit Entwicklern und Vertreibern von medizinischen Hilfsmitteln. Eine Gründung und Leitung eines Qualitätszirkels für Palliativmedizin, die Entwicklung und Einführung eines Kursmoduls für Palliativmedizin schließen dieses Betätigungsfeld inhaltlich ab. Hinsichtlich der Forschung liegt der Schwerpunkt von Prof. Träger auf der Versorgungsforschung – mehrere Forschungsprojekte (Betreuung von Dissertationen, firmenunabhängige Forschung, Grundlagenforschung) in diesem Gebiet sind momentan in entscheidenden Phasen der Ausführung.

¹ siehe: <https://www.itm.uni-luebeck.de/research/publications/>

² Siehe: <http://www.isip.uni-luebeck.de>

Durch die Einbindung der Dienstleistungspartner ist eine direkte Umsetzung der Forschungsergebnisse in Produkte und Dienstleistungen vorgesehen. Die jahrelange Erfahrung der beteiligten Partner im Bereich Diakonie, Pflege und Wachdienst ermöglicht die Gestaltung eines belastbaren Geschäftsmodells. Dabei wird durch den Verbund eine regionale Umsetzung einerseits in der Testphase in Privathaushalten im Raum Lübeck garantiert, andererseits entsteht durch die bundesweite Öffnung der Plattform die Möglichkeit einer schnellen Verbreitung der Ergebnisse in regionalen und überregionalen Initiativen, sowohl durch den Verbund als auch durch Drittanbieter.

Die Zusammenarbeit erfolgte innerhalb des Projektes mit allen an diesem Projekt beteiligten Teilnehmern in guter bis vorbildlicher Weise. Alle Fortschritte wurden über ein internes web-basiertes Informationssystem und bei monatlichen Arbeitstreffen besprochen und abgeglichen.

3. Planung und Ablauf des Vorhabens

Das SMART ASSIST Konzept vereint eine Server-basierte Infrastruktur mit lokalen Komponenten in den Wohnungen der Kunden zur Datenerfassung und zur Unterstützung der Kommunikation. Wesentliches Element ist die einfache Installation und Konfiguration aller Systemkomponenten. Durch offene Schnittstellen können Drittanbieter ihre Dienstleistungen in das System integrieren. Die Kommunikation mit dem System erfolgt über stationäre und/oder mobile Geräte.

Mit Hilfe von drahtlosen Sensoren werden Zustände und Aktionen im Haushalt erkannt und algorithmisch nach sozialen und medizinischen Gesichtspunkten gefiltert und ausgewertet. Die einzelnen Sensoren verknüpfen sich im Verbund des Haushalts automatisch zu einem Sensornetzwerk und erlauben so die unaufdringliche Erkennung bestimmter sozialer Parameter (z.B. Duschen, Fernsehen, etc.) von schleichenden Veränderungen (z.B. Häufigkeit der Flüssigkeitsaufnahme) oder auch Notfällen (z.B. Sturz). Das Netzwerk aus Sensoren unterstützt einerseits eine direkte Verarbeitung der Sensordaten vor Ort, kann sich aber mit Hilfe einer Relay-Station über das Internet mit dem Serviceportal verbinden und wichtige Aktionen initiieren (z.B. SMS an Nachbarn oder Anruf des Arztes).

Das Serviceportal dient als technisches Zentrum des Systems, erlaubt die Registrierung und Authentifizierung von Nutzern und Dienstleistern, kommuniziert mit den Sensornetzwerken und mobilen Geräten, realisiert Datenverarbeitung und –Speicherung unter Berücksichtigung von Datenschutz und Privatsphäre und realisiert ein Webportal. Das Portal soll als ein Marktplatz für eine Vielzahl von unterschiedlichen Diensten und Anwendungen rund um das Thema „Assisted Living“ dienen, der auch Drittanbietern offen steht. Dabei liegt ein besonderer Schwerpunkt des Projekts auf dem geeigneten Schutz der Privatsphäre und der Einhaltung der relevanten Datenschutzbestimmungen.

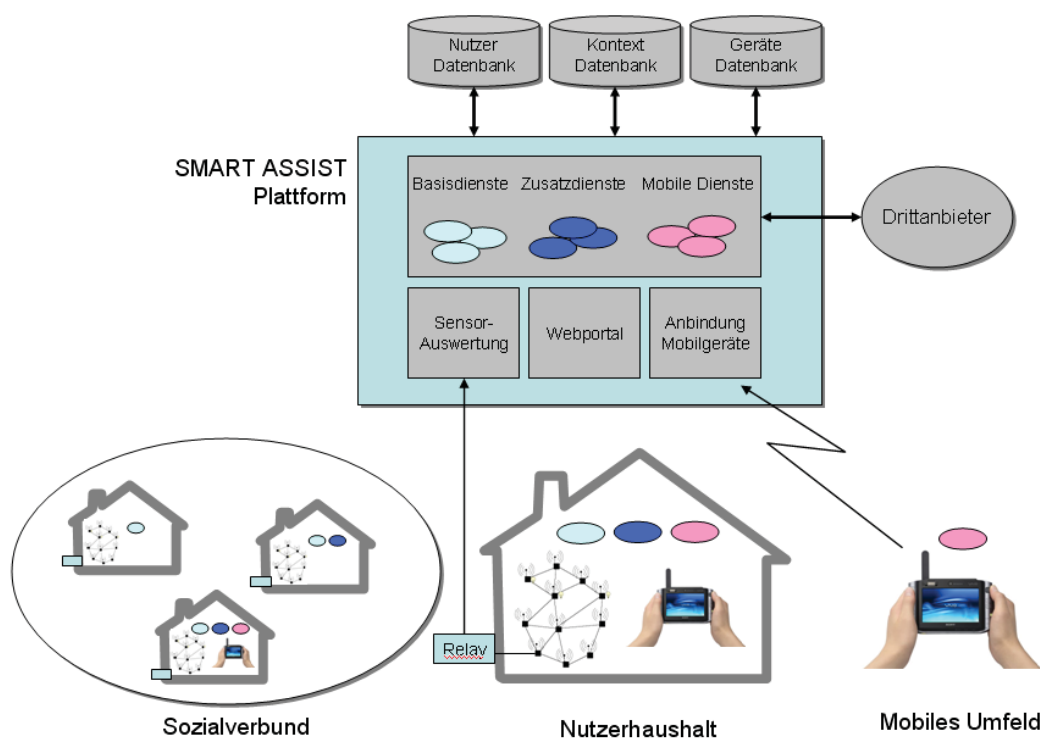
Die Interaktion mit dem System erfolgt personalisiert in unterschiedlichen Service-Klassen. In der Basis-Variante ist nur eine indirekte Interaktion vorgesehen und das System arbeitet autonom (ambient) im Hintergrund. Der Kunde hat keine direkte Bedienoberfläche, bekommt aber die Möglichkeit, die Parameter der gewünschten Dienstleistungen vorzugeben. In einer erweiterten Variante bietet das System Dienste über ein Web-Portal. Der Zugriff kann über Internet-PC oder spezielle Eingabegeräte (z.B. Touchscreen) erfolgen. Hier sind Community-Dienste im Vordergrund, mit denen eine aktive Teilnahme des Kunden in seinem sozialen Verbund unterstützt werden sollen (z.B. Videokonferenz oder auch Telemedizinische Anwendungen).

Als zusätzliche Komponente werden auch mobile Endgeräte unterstützt. Durch eine innovative Software-Architektur für SmartPhones oder PDAs integriert sich das System nahtlos mit den tragbaren Geräten des Benutzers und erlaubt zudem erweiterte Dienstleistungen in unterschiedlichen Kontexten auch außerhalb der eigenen Wohnung. Diese Variante verbindet Lifestyle-Aspekte (z.B. iPhone) mit intelligenten Dienstleistungen in wechselnden Kontexten und Domänen (z.B. automatische Taxi-Bestellung am Ende einer Dialyse-Behandlung oder Zugriff auf Vitalwert-Meßgeräte in der Apotheke).

Der Arbeitsplan zur Realisierung des Gesamtvorhabens gliederte sich in insgesamt fünf Aufgabengebiete:

- Serviceportal: Dies umfasst die Anforderungsanalyse und Spezifikation der einzelnen Komponenten sowie deren Implementierung. Dies umfasst unter anderem die Kommunikation mit den Sensoren in den Wohnungen der Benutzer, die Kommunikation mit der Architektur für Mobilgeräte sowie die Implementierung der Dienst-Plattform.

- **Sensorik:** In diesem Aufgabengebiet wurden sowohl die Hard- als auch die Softwarekomponenten entwickelt und erprobt, welche zum Betrieb der Sensornetze in den Wohnungen der Benutzer sowie die Weiterleitung und Speicherung der gesammelten Daten notwendig sind.
- **Sensordatenanalyse:** Nach der Erfassung und Analyse der Trainingsdatensätze wurden in diesem Gebiet die entsprechenden Algorithmen zur Merkmalsreduktion und Klassifikation der Ereignisse entwickelt und implementiert.
- **Mobile Plattform:** In SmartAssist wurde das Dynamix Framework erstellt, welches es Anwendungsentwicklern auf einfache Art und Weise ermöglicht, mobile und Kontext-sensitive Dienstleistungen in SmartAssist zu integrieren.
- **Anwendungen:** In diesem Abschnitt wurden einige Beispieldienste sowohl für das Serviceportal als auch die Mobile Plattform entwickelt, welche die Funktionalität beider Plattformen zeigt.
- **Testbetrieb und Evaluation:** Der Testbetrieb umfasst die Einrichtung eines Probebetriebs in Lübeck mit ausgewählten Haushalten und die Evaluation aus technischer, sozialer und medizinischer Sicht.



Ziel des Projekts SMART ASSIST ist es, ein technisches Unterstützungssystem zu entwickeln, das es älteren Menschen ermöglicht, länger eigenständig in der eigenen Häuslichkeit zu leben. Ein wichtiges Ziel besteht darin, dass die technischen Komponenten in der Wohnung keiner komplizierten Installation bedürfen und dass ihr Betrieb praktisch unmerklich im Hintergrund stattfindet. Anstelle einer frühzeitigen Einweisung in ein Heim, in dem eine dauerhafte Begleitung geleistet wird, soll eine bedarfsorientierte Betreuung durch Paten erfolgen. Solche Patenschaften können Freunde, Verwandte, Bekannte und Nachbarn ebenso wie kommerzielle Dienstleister übernehmen. Wird eine gesundheitsrelevante Veränderung der Lebensgewohnheiten erkannt, wird der Pate informiert und kann sich persönlich um die Bedürfnisse des älteren Menschen kümmern.

Um dieses Ziel zu erreichen, bedarf es der Erfassung der Lebensgewohnheiten im häuslichen Umfeld, der Erkennung von gesundheitsrelevanten Zuständen, der Unterstützung der Kommunikation mit den Paten sowie der Assistenz außerhalb des Hauses. Durch die Einbindung der Vorwerker Diakonie und der medizinischen Einrichtungen der Universität zu Lübeck wird die soziale und medizinische Kompetenz garantiert und die Sinnhaftigkeit der angebotenen Dienstleistungen im täglichen Alltag evaluiert und verifiziert.

Das SMART ASSIST System verspricht

- Unaufdringlichkeit – durch eine möglichst einfache technische Installation und Bedienbarkeit verbunden mit einer wartungsarmen und sicheren Funktionalität, die die menschlichen Aktivitäten und Bedarfe aus dem Hintergrund unterstützen; im besten Fall für den betreuten Menschen völlig transparent.
- Nutzerakzeptanz – durch eine ausgewogene Balance zwischen Überwachung und Assistenz und die wesentliche Verwendung indirekter Sensoren (z.B. Licht, Strom, Gas, Wasser, etc.) werden genügend relevante Daten für sinnvolle Dienste generiert und gleichzeitig die Privatsphäre der Benutzer beachtet. Das System sollte idealerweise nicht als Einstieg in den Pflegezustand, sondern als Ergänzung und Bereicherung des täglichen Lebens, als Lifestyle-Komponente begriffen werden, lange bevor ein Pflegefall eintritt.
- Kosteneffizienz – die preisgünstige technische Infrastruktur unter Verwendung möglichst vieler bereits im Haushalt vorhandener Geräte (PC, Mobilfunkgerät, Internet-Anschluss, etc.) ermöglicht realistische Preismodelle.
- Integrierbarkeit – das SMART ASSIST System zeichnet sich durch offene Schnittstellen und der Möglichkeit aus, Drittanbieter und Partner zu integrieren. Dadurch wird eine Auswahl aus einer Vielzahl von Leistungen möglich, die von einem isolierten Einzelsystem nicht erbracht werden können (z.B. Lebensmittel-Lieferungen, Online-Apotheken, Unterhaltung, Bildung, etc.)

SMART ASSIST bietet eine Plattform statt geschlossener Einzeldienste. Die offene Architektur motiviert Dienstleister aus dem AAL-Umfeld und darüber hinaus zur Entwicklung spezifischer Angebote. Durch die sozio-technische Fokussierung steht bei SMART ASSIST im Wesentlichen die Unterstützung sozialer Kompetenz und Aktivität im Vordergrund. Der minimale Installationsaufwand (drahtlose Sensormodule) bietet bessere Möglichkeiten einer optischen Anpassung an das jeweilige Wohnumfeld. Zudem ist das System skalierbar und erlaubt unterschiedliche Ausbaustufen und Kostenfaktoren. Durch die Integration der Mobilkomponenten ist der Benutzer nicht auf seine Wohnung beschränkt und kann seine Dienstleistungen auch unterwegs und in wechselnden Umgebungen nutzen.

Im Rahmen dieses Teilvorhabens wurden das Serviceportal und die mobile Plattform entwickelt und implementiert, die Sensordaten analysiert und die medizinische Begleitforschung durchgeführt.

4. Stand von Wissenschaft und Technik

4.1. Zielgruppen von SmartAssist

Nach Angaben des Statistischen Bundesamtes wird sich der Anteil an pflegebedürftigen Menschen in Deutschland in den kommenden 20 Jahren um 58% erhöhen [61][61]. Dieser durch die Überalterung der Gesellschaft hervorgerufene Effekt stellt eine organisatorische und vor allem finanzielle Herausforderung für das Gesundheitssystem und die Altenpflege dar. Die Zunahme der pflegebedürftigen Menschen verursacht nicht nur steigende Kosten, sondern geht auch einher mit einer sinkenden Zahl erwerbstätiger und pflegender Menschen. Von dieser Entwicklung sind neben Deutschland auch viele andere Länder betroffen – vor allem in Asien und Europa. SmartAssist soll durch AAL-Technologien hierbei eine bedarfsorientierte Betreuung in der eigenen Wohnung durch Privatpersonen im sozialen Netzwerk sowie kommerzielle Dienstleister als Alternative zur Einweisung in ein Pflegeheim ermöglicht werden. Zusätzlich zur Kostenersparnis für das Gesundheitssystem und den Benutzer durch die bedarfsorientierte Pflege bieten sich Einnahmequellen durch die Integration von Dienstleistungen externer Anbieter in das AAL-System. Ein wirtschaftlicher Vorteil darf dabei indes nicht zu Lasten des Datenschutzes oder der Transparenz für die Betroffenen führen, so dass diese Aspekte bereits bei der Konzeption von AAL-Technologien hinreichende Berücksichtigung finden müssen. Die Benutzergruppe von SmartAssist teilt sich in insgesamt zwei Untergruppen auf:

Die Zielgruppe des Serviceportals von SmartAssist besteht aus Personen, denen das altersgerechte autonome Leben in ihrer eigenen Wohnung als Alternative zu einer Einweisung in ein Heim ermöglicht werden soll. Basierend auf der von Birgid Eberhardt in [60] aufgestellten Liste der Zielgruppen für AAL-Technologien entspricht dies den folgenden Gruppen:

- Personen mit Unterstützungsbedarf (Zielgruppe 1.2)
- An Prävention interessierte Menschen (Zielgruppe 1.1.3), hauptsächlich zur Sekundär- und Primärprävention

Hierbei handelt es sich folglich um Personen, die zu einem autonomen Leben (eingeschränkt) fähig sind, jedoch durch Alterserscheinungen, eine Krankheit oder Behinderung jetzt oder in absehbarer Zukunft auf Unterstützung angewiesen sind. Weiterführende Informationen zur Zielgruppe und mit der einhergehenden Zielstellung von AAL Technologien finden sich in der Arbeit von Peter Georgief unter [61].

Für die Rolle der Paten kommt eine Vielzahl von Personen- und Altersgruppen in Frage, welche zum Monitoring und zur Notfallhilfe eingesetzt werden können. Die folgende Liste gibt eine Übersicht, welche sich analog zu oben an der in [60] aufgestellten Liste orientiert:

- Menschen, die sich um Menschen mit Unterstützungs- oder Pflegebedarf kümmern (Zielgruppe 2.1.1), z.B. Verwandte und Bekannte (Kinder, Enkel, Freunde, Nachbarn): Wird bei Veränderungen oder im Notfall informiert
- Dienstleistungsanbieter, z.B. Pflegedienste oder Wachunternehmen (Zielgruppen 2.4 und 3.8): Wird bei Veränderungen oder im Notfall informiert, Zugriff auf einige Sensorwerte
- Medizinisches Personal, z.B. Hausarzt, Krankenwagen (Zielgruppe 2.2): Zugriff auf alle Sensorwerte und Ferndiagnose, wird bei Veränderungen oder im Notfall informiert

Die Hauptaufgaben der SmartAssist Portalseite können in mehrere Funktionsblöcke eingeordnet werden. Zunächst werden diese Funktionsblöcke im Einzelnen beschrieben und deren Nutzen für AAL Anwendungen allgemein und SmartAssist im speziellen vorgestellt. Die

Funktionsblöcke werden im weiteren Verlauf des Abschnittes bei der Recherche herangezogen, wobei zunächst existierende Standards und anschließend etablierte Softwarelösungen vorgestellt werden. Den Abschluss bildet eine Auswahl von Lösungen mit welchen das Smart Assist-Portal realisiert werden kann. Die Funktionsblöcke lauten:

- **Monitoring:** Darstellung und Überwachung von Sensormesswerten
- **Kommunikation:** Kommunikation zwischen und Vernetzung von Personen
- **Information:** Austausch von (selbst erzeugten) Inhalten (z.B. Erfahrungsberichte)
- **Dienstplattform:** Schnittstellen für (externe) Dienste, die das Portal erweitern

In der Arbeit von Becks, Dehm und Eberhardt [62] wird als ein Trend für diese Funktionsblöcke erwartet, dass in Zukunft diese „Technologien (...) aus den Bereichen TeleMonitoring, Web-Portaldienste, [...] und moderne Kommunikationstechniken in einer Single-Point-of-Contact-Lösung [...] zusammengeführt (werden).“ Durch die Kombination der Module Monitoring, Information und Kommunikation lassen sich die Patienteninformationen „in idealer Weise, nämlich personalisiert, medizinisch begründet und objektiviert zusammenstellen“. Als positive Effekte werden eine Verbesserung der Compliance, ein verbessertes Selbstmanagement der Patienten und eine breitere Wissensbasis der behandelnden Ärzte in Aussicht gestellt. Die einzelnen Funktionsblöcke werden im Folgenden genauer beschrieben.

Monitoring

Mit dem Begriff Monitoring wird die (medizinische) Überwachung von Vital- oder Umgebungsparametern eines Patienten bezeichnet. Neben der Überwachung sind die Untersuchung und Auswertung ebenfalls möglich. Die Parameter bzw. Messwerte werden hierbei von Sensoren erfasst, welche sich entweder am Patienten oder in dessen (häuslichem) Umfeld befinden und z.B. dessen Blutdruck messen oder einen Sturz detektieren können. Eine Basisstation erfasst diese Daten vor Ort und kann die Messergebnisse entweder direkt verarbeiten und/oder über ein Übertragungssystem an ein Telemedizinisches Zentrum und damit an das überwachende (medizinische) Personal übermitteln.

Kommunikation (Soziales Netzwerk)

Ein soziales Netzwerk beherbergt eine Gemeinschaft von Personen, welche innerhalb des Netzwerkes durch ein Profil repräsentiert werden und zwischen denen sich Verbindungen wie z.B. Freundschaften oder Gruppen bilden können. Das Profil einer Person kann neben Personendaten wie Name, Geburtsdatum und Anschrift weitere Daten ähnlich eines Lebenslaufes oder auch die Interessen, Hobbies, Lieblingsbücher, Stimmungen, Meinungen und aktuelle Handlungen („Gehe jetzt ins Kino“) beinhalten. Die Sichtbarkeit dieser Profildaten kann vom Benutzer selbst festgelegt werden und entweder für alle oder nur einige ausgewählte Personen sichtbar sein, die z.B. mit dem Benutzer befreundet sind.

Ein soziales Netzwerk dient dazu, mit anderen Benutzern Informationen über sich selbst oder bestimmte Themen auszutauschen, um z.B. Freunde auf dem Laufenden zu halten, mit Bekannten in Kontakt zu bleiben oder sich mit Gleichgesinnten über gemeinsame Hobbies auszutauschen. Hierfür stellen Soziale Netzwerke unter Anderem Funktionen zum Versand und Empfang von Nachrichten zur Verfügung, zur Einrichtung von Gruppen, zum Suchen von Personen und zur Veröffentlichung von kurzen Texten, ähnlich einem (We)Blog („online Tagebuch“). Einen Überblick über den Begriff und die Bestandteile eines Sozialen Netzwerkes, und deren Geschichte bietet [63].

Die Grenzen zwischen einem Sozialen Netzwerk und den im Folgenden vorgestellten Sozialen Medien sind fließend. Viele Soziale Netzwerke sind ebenfalls Soziale Medien und umgekehrt. So gehört das Anbieten von Hörproben und Downloads von selbstproduzierten Musik-

stücken bei einem der bekanntesten Sozialen Netzwerken, MySpace³ seit der Gründung zu dessen Kernfunktionen. Musiker und Bands können direkt mit ihren Fans und Gleichgesinnten Informationen austauschen und über neue Lieder oder Tourneen informieren. Hierbei kann jeder Nutzer seine eigene Musik anbieten, so dass MySpace sowohl von großen kommerziellen Bands genutzt wird, jedoch auch kleine semi-professionelle oder Amateurbands das Portal nutzen. Andere bekannte Beispiele für Soziale Netzwerke sind Facebook⁴ oder die VZ-Netzwerke⁵ in Deutschland.

Die im Sozialen Netzwerk vom Benutzer mit dem Zweck der Selbstbeschreibung hinterlegten Daten (z.B. Aktivitäten) können dem medizinischen Personal (oder Paten) als breite Wissensbasis bei der Auswertung der Monitoring Daten dienen. Darüber hinaus soll die soziale Integration in und die Kommunikation der Benutzer mit ihrem soziales Umfeld unterstützt werden.

Information (Soziales Medium)

Soziale Medien zeichnen sich dadurch aus, dass ihr Inhalt nicht wie in klassischen Medien von einer kleinen Gruppe von (beruflichen) Redakteuren erstellt und erwaltes wird. Stattdessen kann jeder Konsument dem Medium auf einfache Art und Weise ebenfalls Inhalte beisteuern und so selbst zum Produzenten werden. Im Zuge des sogenannten „Web 2.0“ spricht man in diesem Zusammenhang auch von „User generated content“, also von Inhalten, welche von den Benutzern eines Systems selbständig erstellt werden. Weitere Informationen finden sich in der Arbeit von Kaplan und Haenlein [64].

Ein Beispiel für Soziale Medien sind die sogenannten Wikis wie z.B. die Wikipedia, eine online Enzyklopädie, bei der jeder Benutzer nicht nur Informationen abrufen, sondern auch selbst Informationen bereitstellen und bestehende Informationen verändern kann. Ein anderes Beispiel sind Video Portale wie YouTube, auf denen Benutzer selbstproduzierte Videos online stellen, welche dann von anderen Benutzern bewertet, kommentiert oder auch mit eigenen Videos beantwortet werden können.

Auf diese Art und Weise können die Benutzer des Serviceportals beispielsweise ihre eigenen Erfahrungsberichte und Ratgeber für andere Benutzer beisteuern, auf Neuigkeiten oder Ereignisse im regionalen Umfeld hinweisen (z.B. Veranstaltungen von Vereinen) und sich über ihre Hobbies austauschen. Dies kann zum Beispiel im Rahmen eines Forums, wie es in der SmartAssist Portalseite implementiert ist, geschehen. (Externe) Dienstleister können ebenfalls Informationen für die Benutzer bereitstellen, die sich für entsprechende Themen interessieren oder bestimmte Dienste abonniert haben. Dies kann z.B. der Speiseplan eines Essenbringdienstes sein, bei dem die Benutzer die gelieferten Mahlzeiten im Anschluss bewerten und weiterempfehlen können, was eine wertvolle Informationsquelle für den Anbieter und die andere Kunden darstellen kann.

Dienstplattform

Eine Dienstplattform ermöglicht es, eine bestehende Plattform durch die Integration von weiteren internen oder externen Diensten zu erweitern. Die Plattform bietet hierfür (technische) Schnittstellen für die Dienste an, um auf die Daten der Plattform zugreifen zu können und/oder um der Plattform Daten zur Verfügung zu stellen. Zusätzlich stellt die Plattform Schnittstellen zur Darstellung der Dienste und Informationen zur Verfügung, um dem Benutzer Zugriff auf die Dienste bieten zu können. Benutzer können sich auf diese Art und Weise

³ <http://www.myspace.com/>

⁴ <http://www.facebook.com/>

⁵ <http://www.meinVZ.de/> sowie <http://www.studiVZ.de/> und <http://www.schuelerVZ.de/>

eine individuelle Gruppe von Diensten zusammenstellen, welche für Sie nützliche oder attraktive Funktionen bereitstellt.

In der Arbeit von Schmidt, Wolf, Klein und Balfanz [65] werden Eigenschaften einer solchen Dienstplattform im Rahmen von AAL vorgestellt. Laut [65] „sind die Anforderungen, Wünsche, Voraussetzungen und finanziellen Möglichkeiten“ der Zielgruppe von AAL Lösungen zu verschieden für eine einheitliche Komplettlösung. Die Flexibilität und Erweiterbarkeit der Lösungen ist notwendig, um unterschiedliche „Zielgruppen, aber auch unterschiedliche Lebensphasen“ anzusprechen. Des Weiteren wird darauf hingewiesen, dass „dies [...] nur möglich (ist), wenn man statt einer Vielzahl von speziellen Einzellösungen auf eine gemeinsame Plattform setzt, die sich leicht durch unterschiedliche kostengünstige Komponenten erweitern lässt. Wie viele Beispiele (...) zeigen, darf sich das Komponentenangebot dabei nicht nur auf einen einzelnen Hersteller (den Plattformanbieter) beschränken, sondern muss für andere Marktteilnehmer offen stehen, um ein breites Angebot (auch an Nischenkomponenten) machen zu können und gleichzeitig die Skalenvorteile der Plattform zu nutzen“.

4.2. Standards

Die in diesem Abschnitt vorgestellten Standards bieten Funktionen zur Identifikation und Autorisierung von Benutzern, zum Datenaustausch zwischen einem Sozialen Netzwerk und Drittanbietern und für die Implementierung von Anwendungen für diese Netzwerke. Sie dienen der Kompatibilität und Erweiterbarkeit des SmartAssist Serviceportals, so dass (externe) Dienste über Schnittstellen auf sichere Art und Weise auf die Benutzerdaten zugreifen können oder ihre Dienste direkt in der Portal integrieren können.

Autentifizierung, Authorisation und Access Control

Das OpenID System ermöglicht es einem Benutzer, nachdem er sich bei einem OpenID Provider mittels Benutzername und Passwort registriert hat, sich mit diesem Konto bei anderen OpenID unterstützenden Webseiten anzumelden. Auf diese Art und Weise kann ein Benutzer eine OpenID Identität für mehrere Webseiten verwenden und muss sein Passwort nur dem OpenID Provider gegenüber zur Verfügung stellen, welcher die Identität des Benutzers für die besuchten Internetseiten bestätigt. Dies wird auch als Single-Sign-On bezeichnet.

Während OpenID zur Identifizierung eines Benutzers verwendet wird, bietet OAuth eine Schnittstelle zur Zugriffskontrolle an. OAuth kann in Kombination mit OpenID oder einer anderen Identifizierungslösung verwendet werden und regelt die Art und Weise des Zugriffs von Drittanbietern auf die (persönlichen) Daten des Benutzers. Mittels OAuth ist es möglich, einem Drittanbieter für einen begrenzten Zeitraum lesenden Zugriff auf bestimmte Daten (wie z.B. Photos, Dokumente oder Adressen) zu erlauben.

Sowohl OpenID als auch OAuth sind offene Protokolle, deren Spezifikationen frei verfügbar sind und für welche Open Source Bibliotheken verfügbar sind.

OpenSocial

Der von Google im November 2007 veröffentlichte OpenSocial Standard besteht aus einer Sammlung von Programmierschnittstellen, welche in beliebigen Sozialen Netzwerken Verwendung finden können. OpenSocial definiert unter anderem Schnittstellen für den Zugriff auf Personendaten, Freundschaftsbeziehungen und Aktivitäten über eine REST-Schnittstelle und stellt eine API für sogenannte OpenSocial Gadgets zur Verfügung. Gadgets sind Anwendungen, welche z.B. von Drittanbietern in die Internetseite eines OpenSocial Netzwerkes integriert werden können und diese um zusätzliche Funktionen erweitert. Der Zugriff auf die persönlichen Daten der Benutzer wird hierbei mittels OAuth gesteuert.

OpenSocial wird unter anderem von den drei großen VZ Netzwerke in Deutschland (StudiVZ, MeinVZ und SchülerVZ), MySpace, Orkut und LinkedIn verwendet. Die Open Source Implementierung der Apache Software Foundation mit dem Namen Shindig versteht sich als Referenzimplementierung des OpenSocial Standards und ermöglicht die Integration von OpenSocial in bestehende Netzwerke.

Facebook Connect

Facebook Connect stellt ähnlich wie OpenID eine Single-Sign-On Lösung für Benutzer zur Verfügung und bietet darüber hinaus wie OpenSocial Schnittstellen zum Zugriff auf die persönlichen Daten eines Benutzers. Auf diese Weise kann der Benutzer nach dem Anmelden mittels Facebook Connect seine Profildaten, Fotos und Kontaktlisten zu anderen Portalseiten mitnehmen. Auch die Gegenrichtung wird unterstützt, so dass die Aktivitäten eines Benutzers auf diesen Seiten in Facebook dargestellt werden können.

Facebook Connect (proprietär) und OpenSocial (Open Source) sind konkurrierende Plattformen, welche sich in Funktionsumfang und Zielsetzung stark ähneln.

WebServices

WebServices und die damit verbundenen Spezifikationen bieten eine Implementierungstechnik für eine SOA (Service Oriented Architecture) und dienen damit der Maschine-Maschine-Kommunikation, vor allem zur Verbindung von Diensten über Internetprotokolle und -formate. Als Grundlage oder Basis von WebServices hat sich laut [66] die Verwendung von SOAP durchgesetzt, es finden sich jedoch auch Realisierungen von WebServices mittels REST.

Der W3C Standard SOAP stellt eine „Spezifikation zur Kommunikation mit Diensten in einer auf Basis von WebServices implementierten Service-Orientierten Architektur“ dar [66]. Im Gegensatz zu SOAP ist REST (“Representational State Transfer”) kein Protokoll, sondern ein Architekturstil, welcher existierende Web-Protokolle verwendet, um Web Services zu realisieren. So werden die Aufrufe eines Services ausschließlich mittels des Hypertext Transfer Protokolls (HTTP) auf eine Ressource (URI/URL) mittels der in HTTP definierten Operationen (z.B. GET) realisiert. REST ist daher auf Anwendungen im World Wide Web fokussiert und profitiert laut [66] vor allem in den Punkten Performance, Skalierbarkeit, Verfügbarkeit, Erweiterbarkeit und Sicherheit von der Verwendung existierender Web-Protokolle wie HTTP und SSL/TLS (HTTPS).

Barrierefreiheit

Der barrierefreie Zugang zum Serviceportal ergibt sich unter anderem durch die Zielgruppe des Serviceportals. Das deutsche Gesetz zur Gleichstellung behinderter Menschen definiert den Begriff der Barrierefreiheit in §4 folgendermaßen: „Barrierefrei sind [...] Systeme der Informationsverarbeitung [...], wenn sie für behinderte Menschen in der allgemein üblichen Weise, ohne besondere Erschwernis und grundsätzlich ohne fremde Hilfe zugänglich und nutzbar sind.“. Der Einsatz von Techniken zur Barrierefreiheit dient jedoch nicht nur Men-

schen mit Behinderungen, sondern auch Benutzern mit technischen (Textbrowser, PDA, Smartphone) und altersbedingten Einschränkungen und darüber hinaus auch der besseren Indizierung durch Suchmaschinen.

Die WAI (Web Accessibility Initiative) Arbeitsgruppe des W3C veröffentlichte im Jahr 1999 die Richtlinie WCAG (Web Content Accessibility Guidelines), welche Empfehlungen zur barrierefreien Gestaltung der Inhalte von Internetangeboten enthält. Die WCAG ist die Basis der BITV (Barrierefreie Informationstechnik-Verordnung), einer Erweiterung des Behindertengleichstellungsgesetzes, und ist für öffentlich zugängliche Internetauftritte der Behörden der Bundesverwaltung verbindlich, wird jedoch auch auf freiwilliger Basis von privaten und kommerziellen Internetangeboten verwendet.

Es existieren mehrere automatisierte Testverfahren, für die Barrierefreiheit einer Internetseite:

- BIK BITV Test (<http://www.bitvtest.de>), ein vom Bundesministerium für Arbeit und Soziales gefördertes Gemeinschaftsprojekt der DIAS GmbH und der Blinden- und Sehbehindertenverbände DBSV und DVBS
- AChecker (<http://www.atutor.ca/achecker/>), vom ATRC Center der University of Toronto, Kanada. Testet WCAG und BITV
- WAVE web accessibility evaluation tool (<http://wave.webaim.org/>), kostenloser WCAG Test der Firma WebAIM
- Functional Accessibility Evaluator, FAE (<http://fae.cita.uiuc.edu/>) von der University of Illinois at Urbana-Champaign, überprüft die WCAG Richtlinien

Beispiele Sozialer Gesundheits-Netzwerke

Der Einsatz von sozialen Netzwerken für Gesundheit und bezogene Themen ist vor allem im Amerikanischen Raum schon seit einigen Jahren verbreitet und wird von vielen Menschen regelmäßig genutzt. Die folgende Liste führt einige Beispiele hierfür auf.

- **Eons.com** ist ein soziales Netzwerk für die sogenannte „Baby Boom“ Generation im Alter zwischen etwa 45 und 65+ Jahren. Eons.com existiert seit Juli 2006, hat etwa 25-30.000 Mitglieder⁶ und beinhaltet neben sozialen Funktionen wie Profilen, Gruppen und Photoalben vor allem Interviews, Artikel und Ratgeber zu den Themen Gesundheit, Partnerschaft, Fitness, Ruhestand und Versicherungen.



Abbildung 1 Eons.com

⁶ <http://www.alexa.com/siteinfo/Eons.com>

- **PatientsLikeMe.com** ist ein im Jahr 2005 gestartetes soziales Netzwerk, in welchem sich die Mitglieder Krankheitsspezifisch über Symptome, Behandlungen und Erfahrungen austauschen können. Das Netzwerk beinhaltet unter anderem „Communities“ zu den folgenden Krankheiten: HIV/AIDS, Multiple Sklerose oder Parkinson.



Abbildung 2 PatientsLikeMe.com

- Das soziale Netzwerk **Dailystrength.org** bietet seit August 2006 ähnliche Funktionen wie PatientsLikeMe.com, bietet also eine Austauschplattform für Krankheitsverläufe, und deren Behandlungen, erweitert dies jedoch auch auf allgemeine Lebenssituationen wie Scheidungen, Einsamkeit, Kindererziehung und Schwangerschaft.



Abbildung 3 Dailystrength.org

- Die online Community **MedHelp.org** bietet Diskussionsforen zum Austausch über „Healthcare“ Themen ähnlich zu den oben genannten Internetseiten, arbeitet jedoch direkt mit Ärzten aus Krankenhäusern und Forschungseinrichtungen zusammen und Benutzer können auf diese Weise professionellen Rat einholen. MedHelp ist nach eigenen Angaben die „größte Health Community der Welt“ und wird monatlich von 10 Millionen Besuchern besucht.



Abbildung 4 MedHelp.org

Übersicht existierender Lösungen

Bei der Realisierung des Serviceportals ist es sinnvoll, existierende und vor allem freie Software hinsichtlich ihrer Eignung für das SmartAssist Serviceportal zu überprüfen. Die Verwendung freier Software, welche bereits eine große Verbreitung gefunden hat, kann hierbei den Vorteil besitzen, dass bereits eine Vielzahl von Funktionen implementiert, getestet, von Fehlern beseitigt und auf Sicherheitslücken überprüft wurde. Darüber hinaus wird bei der Recherche darauf geachtet, dass die Softwarelösungen um die von SmartAssist benötigten Funktionen erweitert werden können. Es folgt eine Auswahl der ihm Rahmen der Recherche ermittelten Softwarelösungen und eine Untersuchung hinsichtlich deren Eignung für das SmartAssist Serviceportal. Den Abschluss bildet eine Prognose hinsichtlich der am besten geeigneten Softwarelösung.

Die folgende Liste orientiert sich an der von der Internetseite „Backend Battles“ im Januar 2010 veröffentlichten Statistik⁷, welche die am häufigsten eingesetzten Content-Management-Systeme unter den „Alexa’s 10,000 most popular websites“⁸ beinhaltet. In dieser Statistik wird die Häufigkeit verglichen, mit welcher ein bestimmtes CMS in den 10.000 am häufigsten genutzten Internetseiten verwendet wird. Die Top 3 sind:

Platz 1: Drupal (65 Internetseiten)

Platz 2: Joomla (29 Internetseiten)

Platz 3: MediaWiki (22 Internetseiten)

Neben den oben gelisteten Content-Management-Systemen werden im Folgenden auch die auf Social Networking Funktionen spezialisierten Projekte Dolphin, XOOPS und Elgg beschrieben und hinsichtlich ihres Funktionsumfangs verglichen.

⁷ http://www.backendbattles.com/Content_Management_Systems

⁸ <http://www.alexa.com/topsites>

Drupal

Bei Drupal handelt es sich um ein Content-Management-System, welches aktuell in der Version 6.16 vorliegt und eine große Verbreitung besitzt. Drupal wird unter anderem vom Weißen Haus⁹, Amnesty International¹⁰ und dem Welternährungsprogramm¹¹ der Vereinten Nationen zur Realisierung deren Internetseiten eingesetzt. Drupal 7 befindet sich gegenwärtig im Alpha-Stadium, noch ist jedoch kein Zeitpunkt für das Release von Drupal 7 festgelegt worden. Drupal ist in PHP geschrieben, verwendet MySQL als Datenbank und steht unter der GNU General Public License. Drupal lässt sich durch Module in seiner Funktionalität und seinem Erscheinungsbild umfassend erweitern und anpassen. Es stehen unter anderem Module zur Realisierung eines Sozialen Netzwerkes, zur Integration von Apache Shindig und damit dem OpenSocial Standard zur Verfügung. Darüber hinaus stehen Module für OpenID, OAuth, WebServices, FOAF und Facebook Connect zur Verfügung. Drupal bietet außerdem Themes zur Anpassung des Erscheinungsbildes, welche Validierungstests gegen XHTML 1.0 Strict, CSS 2,1 & 3.0, WAVE und FAE erfolgreich bestehen¹².

Joomla

Joomla ist ein PHP Open Source Content-Management-System, welches 2005 aus dem CMS Mambo entstanden ist, es liegt gegenwärtig in Version 1.5.15 vor und verwendet MySQL. Joomla lässt sich ähnlich wie Drupal durch Module und Komponenten in seiner Funktionalität erweitern und unterstützt auf diese Weise einen vergleichbaren Funktionsumfang.

MediaWiki

Auch MediaWiki fällt in die Kategorie der unter GPL veröffentlichten, in PHP geschriebenen und MySQL verwendenden CMS Systeme. MediaWiki liegt in Version 1.15 vor und kann durch sog. Extensions in seiner Funktionalität erweitert werden. Im Extension Repository von MediaWiki finden sich Erweiterung für die Unterstützung von OpenID, OAuth, Facebook Connect und zur Realisierung von Social Network Funktionen, eine Schnittstelle zu OpenSocial fehlt jedoch. Auch Erweiterungen für WebServices via REST oder SOAP fehlen.

Dolphin

Im Gegensatz zu den oben genannten CMS ist Dolphin eine spezialisierte Plattform zur Umsetzung von Sozialen Netzwerken. Dolphin bietet bereits OpenSocial, Facebook Connect und OpenID, ohne auf Module angewiesen zu sein. Darüber hinaus bietet Dolphin Funktionen wie Videochat und iPhone Integration. Obwohl Dolphin unter der Creative Commons Lizenz steht, kostenlos heruntergeladen, verwendet und verändert werden kann, benötigt man eine Prime Mitgliedschaft (\$998 für 5 Jahre), um das Internetportal und die mobile Anwendung vom Branding und Werbung für andere Produkte des Herstellers Boonex zu befreien.

XOOPS

Das unter GPL Lizenz veröffentlichte „eXtensible Object Oriented Portal System“ XOOPS liegt gegenwärtig in Version 2.4.4 vor, ist in PHP implementiert und verwendet MySQL als Datenbank. XOOPS gewann unter anderem den SourceForge.net 2006 Community Choice Award (Kategorie: Development) und mehrere andere Preise auf dem Open Source Sektor. Die XOOPS Erweiterung Yogurt bietet Social Networking Funktionen, eine Anbindung an OpenSocial, Facebook Connect oder OAuth ist jedoch nicht vorhanden.

⁹ <http://www.whitehouse.gov/>

¹⁰ <http://www.amnesty.org/>

¹¹ <http://www.wfp.org/>

¹² http://openconcept.ca/blog/mgifford/function_assessment_of_valid_drupal_6_themes

Elgg

Elgg ist eine Open Source Plattform zur Erstellung eines Sozialen Netzwerkes und verwendet wie alle anderen oben genannten Systeme ebenfalls die Kombination aus PHP & MySQL. Elgg gewann im Jahr 2008 den Preis für die beste Open Source Software in der Kategorie „Collaboration“, herausgegeben von InfoWorld. Es unterstützt bereits in der Grundinstallation in Version 1.7 die Standards OpenSocial sowie OpenID und verfügt über eine erweiterbare RESTful API für JSON und XML. Elgg lässt sich durch Module für z.B. OAuth, SOAP WebServices und Facebook Connect erweitern.

Auswahl

Eine Zusammenfassung der im vorangegangenen Abschnitt vorgestellten Eigenschaften der untersuchten Softwarelösungen findet sich in Tabelle 1. Aufgrund der Vielzahl an Erweiterungsmöglichkeiten durch Module und Themes, die große Verbreitung und Skalierbarkeit wird Drupal zum gegenwärtigen Zeitpunkt als aussichtsreichster Kandidat für die Umsetzung des SmartAssist Serviceportal eingestuft. Drupal bietet bereits Module, welche sämtliche identifizierten Standards erfüllen und die ebenfalls freie Drupal Distribution Pressflow¹³ bringt erweiterbare Funktionen zur Skalierbarkeit und Verfügbarkeit. Die Elgg Plattform erscheint ebenfalls als sehr gut geeignet und wird im Rahmen der nachfolgenden Arbeitspakete ebenfalls untersucht werden. Die im folgenden Kapitel beschriebene Produktumgebung basiert aus diesem Grund auf den Angaben zu Drupal 6.16 sowie Elgg 1.7.

	Drupal	Joomla	MediaWiki	Dolphin	XOOPS	Elgg
Version	6.16	1.5.15	1.15	6.16	2.4.4	1.7
Lizenz	GPL	GPLv2	GPLv2	CC-BY	GPL	GPLv2
Sprache	PHP	PHP	PHP	PHP	PHP	PHP
Datenbank	MySQL	MySQL	MySQL	MySQL	MySQL	MySQL
Soziales Netzwerk	Modul	Modul	Modul	Ja	Modul	Ja
OpenSocial	Modul	Modul	Nein	Ja	Nein	Ja
Facebook Connect	Modul	Modul	Modul	Ja	Nein	Modul
OpenID	Modul	Modul	Modul	Ja	Modul	Ja
OAuth	Modul	Modul	Modul	Nein	Nein	Modul
REST	Modul	Modul	Nein	Nein	Nein	Ja
SOAP	Modul	Modul	Nein	Nein	Modul	Modul

Tabelle 1: Übersicht Open Source Softwarelösungen

¹³ <https://launchpad.net/pressflow>

4.3. Sensorauswertung

Die verwendeten Verfahren zur Sensorauswertung basieren auf mathematischen Modellen, die keinen Schutzrechten unterliegen. Hierbei sei insbesondere auf die verwendeten Conditional Random Fields und Maximum-Entropy-Markov-Modelle verwiesen [98] [99], die die Basis der verwendeten Methoden bilden und selbst auf Hidden-Markov-Modellen [103] aufbauen. Diese Modelle wurden im Rahmen der Arbeit am Projekt SmartAssist modifiziert und erweitert. Durch die Publikationen sind auch diese Erweiterungen frei zugänglich. Hidden-Markov-Modelle sind allgemein anerkannte Modelle, die ebenfalls zur Ereignisdetektion genutzt werden [105].

Markov-Modelle stellen einen Zusammenhang zwischen Daten und abstrakten Zuständen dar. Zu jedem Zeitpunkt wird ein eigener 'Zustand' angenommen. Anschließend folgt der Übergang in einen anderen Zustand. Diese zeitliche Reihenfolge der Zustände kann dafür genutzt werden, die Abläufe von Aktionen zu modellieren, daher werden Markov-Modelle bevorzugt für die Überwachung eingesetzt.

Im Rahmen der Sensorauswertung wurde die Ereignisdetektion auf Conditional Random Fields übertragen, die modernere und effektivere Markov-Modelle darstellen als Hidden-Markov-Modelle. Insbesondere werden dadurch bestimmte Annahmen vermieden, die Hidden-Markov-Modelle an die Daten stellen und dadurch höchstens bedingt für die Verwendung in SmartAssist geeignet wären. Insbesondere werden für Hidden-Markov-Modelle die statistischen Verteilungen der Sensorwerte modelliert, was im Falle der inhomogenen Sensor-Netzwerke sehr komplex und ungenau ist. Darum sind herkömmliche Hidden-Markov-Modelle nur bedingt geeignet für das Projekt SmartAssist, und es wurde eine Entscheidung zugunsten der Conditional Random Fields getroffen, die die Vorteile der Markov-Modelle mit einer flexiblen Datenverarbeitung vereinen.

Zur Implementierung wurde neben dem genannten Mathworks Matlab auch GNU Octave¹⁴ verwendet. In beiden Fällen erfolgte die Implementierung anhand des standardmäßigen Funktionsumfangs. Zur Implementierung in C++ wurden insbesondere die Bibliotheken libcurl¹⁵ und Armadillo¹⁶ verwendet, als Compiler der Open-Source-Compiler GCC.

4.4. Kontext

In der Literatur finden sich unterschiedliche Berichte über erste Erfahrungen mit dem Einsatz von Sensoren zur Erkennung ungewöhnlichen Verhaltens. So finden sich einerseits Bemühungen, Vitaldaten wie Puls, Blutsauerstoffgehalt oder EKG-Daten mit Hilfe von Sensoren am Körper der Nutzer zu messen [102], aber auch Bewegungen mit Hilfe von Beschleunigungssensoren. Andererseits wird auch von einer Instrumentierung des Umfeldes der Nutzer berichtet. Dieses geschieht beispielsweise mit Sensoren für Helligkeit, Temperatur, Bewegung oder Bettbenutzung [100], aber beispielsweise auch durch Messung des Stromverbrauches [101]. Ebenfalls diskutiert wird der Einsatz von Kameras und Bilderkennungsverfahren, allerdings ist bei diesem Vorgehen mit zusätzlichen Akzeptanzproblemen zu rechnen.

Kontextsensitivität

Die rasanten Entwicklungen in den Bereichen Mikrosysteme, Kommunikationsnetzwerke und Rechentechnologie haben eine Erweiterung des Computers auf Bereiche jenseits des Desktop-Paradigmas ermöglicht. Während neue verteilte Rechenkonzepte basierend auf mobilen Gerä-

¹⁴ <http://www.gnu.org/software/octave/>

¹⁵ <http://curl.haxx.se/libcurl/>

¹⁶ <http://arma.sourceforge.net/>

ten entstehen, besteht die Herausforderung in der Entwicklung von Anwendungen in der Gestaltung von Software, die sich kontinuierlich an die Umgebung des mobilen Nutzers anpasst [112]. In den letzten zwei Jahrzehnten wurde dieser Herausforderung durch die Kombination von Aspekten des mobilen und verteilten Rechnens mit zusätzlichen Methoden der dynamischen Adaptation in kontextsensitiven Systemen begegnet. Kontextsensitive Systeme passen ihr Verhalten während der Laufzeit an den jeweiligen Kontext des Anwenders an. Dey und Abowd definieren Kontext als "... any information that can be used to characterize the situation of entities (i.e., whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves" [118]. Kontextinformationen sind meist hochdynamisch und anfällig gegenüber systemischen Störungen, wie z.B. Netzstörungen, de-kalibrierten Sensoren, Sensordrift, Verschleiß, Feuchtigkeit, etc. [121] Daher umfasst der Prozess der Kontexterfassung mehrere Phasen: Sensormessung, Datenvorverarbeitung, Merkmalsextraktion, Quantisierung und semantische Annotation [121]. Zudem müssen die Kontextdaten in einem maschinenlesbaren Format, dem Kontextmodell, repräsentiert werden, bevor eine Weiterverarbeitung in adaptiven Systemen möglich wird. Strang and Linnhoff-Popien identifizieren eine Reihe von üblichen Modellierungsansätzen, inklusive key-value, markup-scheme, graphisch, objektorientiert, und basierend auf Logiken oder Ontologien [126]. Die Daten aus dem Kontextmodell werden für ein Domain-spezifisches Entscheidungsmodell zur Adaptation des Systems verwendet. Übliche Entscheidungsmodelle umfassen Condition Matching [117], Situationsanalyse [128] und probabilistische Ansätze [110]. Beispiele für kontextabhängige Adaption sind die Veränderung von Systemparametern und Einstellungen (z.B. Stummschaltung eines Mobiltelefons während des Arztbesuches), Interaktion mit identifizierten Umgebungskomponenten (z.B. Ausdruck einer Überweisung auf dem Praxisdrucker) oder die Darstellung von relevanten Informationen aus dem Kontext (z.B. Anzeige der Körperfettwerte als Ergebnis der Messung auf der Badezimmerwaage auf dem Mobilfunkgerät des Benutzers).

Kontextsensitive Infrastrukturen

Die inhärente Komplexität in der Kontexterfassung und -Modellierung haben zur Entwicklung einer Reihe von kontextsensitiven Infrastrukturen und Systemen geführt, deren Ziel die vereinfachte Entwicklung von adaptiven Anwendungen ist. Die meisten aktuellen Ansätze basieren auf Middleware-Konzepten zur Unterstützung von verteilter Kommunikation und Mobilität [112]. Chen hat die folgende Klassifizierung vorgeschlagen [111]: „Direct Sensor Access“-Verfahren sind direkt mit der zugrundeliegenden Sensorhardware, den Gerätetreibern und der Anwendung gekoppelt. Frühe Beispiele dieser Kategorie sind z.B. ParcTAB [129] und das Active Badge System [123]. Obwohl diese Ansätze innerhalb ihrer Zielanwendung effektiv sind, ist eine Erweiterung oder Anpassung an andere Umgebungen nur schwer zu erreichen. „Middleware Infrastructure“ führt eine Abstraktionsschicht zwischen den Anwendungen und dem System zur Kontexterfassung ein. Dies führt zu einer erhöhten Erweiterungsfähigkeit. Existierende Middleware Ansätze aus dieser Kategorie sind meist auf eine spezifische Instrumentalisierung der Umgebung ausgerichtet. Context Toolkit [113] und JCAF [109] sind typische Beispiele. In der dritten Kategorie „Context Server“ wird die Kontexterfassung auf verteilten Einheiten realisiert, die Kontextinformationen von mehreren Geräten aggregieren. Dieses Modell unterstützt die gleichzeitige Verarbeitung vielfältiger Sensordaten auf mehreren Geräten und benötigt meist eine umfangreiche Infrastruktur mit aufwändigen Installationen, die eine Anwendung außerhalb der Kernumgebung verhindert. Typische Beispiele sind Aura [114], Hydrogen [127] und hybride Middleware Konzepte wie Gaia [122]. Zusätzlich wurden eine Reihe von Ad-hoc-Ansätzen, wie UPnP [118], Zeroconf [125], Jini [120] und Jadabs [108] entwickelt. Allerdings versagen die meisten Ad-hoc-Ansätze für domänenübergreifende Szenarien in Situationen, in denen die Dichte der kompatiblen Geräte zu gering ist.

Trotz mehr als zwanzig Jahren Forschung steht die Entwicklung von kontextsensitiven Systemen immer noch vor großen Herausforderungen und die meisten Ansätze sind auf die Entwicklung von Prototypen limitiert, die nur in spezifischen Installationsumgebungen funktionieren [115][116][124]. Eine Studie von Endres et al. hat ergeben, dass die meisten Ansätze reinen Forschungszwecken dienen und keiner kommerziellen Implementierung überführt wurden [116]. Zahlreiche technische und soziale Herausforderungen wurden in der Studie identifiziert. Eine der gravierendsten Einschränkungen der meisten Ansätze liegt in den Begrenzungen für die effektive Modellierung von Kontextinformationen in sehr großen und domänenübergreifenden Szenarien.

Im Projekt SmartAssist wurde für die Entwicklung der mobilen Architektur auf Vorarbeiten der Universität zu Lübeck zurückgegriffen und die ALADIN-Architektur erweitert. Zur Evaluierung des Konzepts wurde bereits ein erster Prototyp (C#, .NET) implementiert und mit verschiedenen Testanwendungen getestet [79]. Zusätzlich wurden einige übliche Sensortypen (Infrarot, Bluetooth, RFID, WLAN) durch Module für Kontexterfassung und -Modellierung unterstützt. Auf Basis des Prototypen wurden einige Anwendungen aus dem Kulturbereich erfolgreich umgesetzt: ein mobiles, interaktives Filmkonzept [78], ein Museumsführer [79] und ein pervasives Computerspiel [80].

Das Projekt SmartAssist wird das prototypische Aladin-Konzept in einer kommerziell nutzbaren Variante für Java-basierte (Mobil-)Geräte implementieren und durch AAL-relevante Komponenten erweitern. Drittanbietern wird durch die Freilegung der APIs eine offene Schnittstelle für die Integration eigener Anwendungen ermöglicht. Ein Toolkit unterstützt diesen Integrationsprozess. Die Möglichkeiten existierender Laufzeitumgebungen, wie Java MIDP oder Google Android können auf diese Weise mit Hilfe der Kontextmodellierung bereichert werden.

5. Zusammenarbeit mit anderen Stellen

Im Rahmen der Spezifikation der Teilsysteme ergaben sich verschiedene Fragestellungen hinsichtlich des Datenschutzes und der Privatsphäre der Benutzer. Da das SmartAssist-Konsortium auf diesem Gebiet über kein ausreichendes Fachwissen verfügte, entschieden wir uns für eine Kooperation mit dem Unabhängigen Landeszentrum für Datenschutz Schleswig-Holstein (ULD). Im Rahmen dieser Kooperation fand unter anderem ein gemeinsamer Datenschutzworkshop am 16. Juni 2010 in Kiel beim ULD statt. Die Tagungspunkte umfassten die Vorstellung des Projektes SmartAssist, eine Einführung in Datenschutzrechtliche Anforderungen und die Vorstellung der Schutzziele und Maßnahmen im Themengebiet Datenschutz und Datensicherheit von Seiten des ULD. Im Rahmen dieser Kooperation entstand außerdem eine gemeinsame Publikation, welche im Rahmen des 4. AAL Kongress 2011 als Poster vorgestellt wurde [16].

Die statistische Bearbeitung der Ergebnisse der medizinischen Begleitforschung wurde von Mathematikern und Epidemiologen aus dem Zentrum für Bevölkerungsmedizin und Versorgungsforschung, dem das Institut für Allgemeinmedizin angegliedert ist, supervidiert.

II. Eingehende Darstellung

1. Resultate des Forschungsvorhabens

1.1. Serviceportal

Das SmartAssist Serviceportal basiert auf dem Content-Management-System Drupal 6 mit zusätzlichen Modulen, welche unter anderem die Kommunikation sowie Freund- und Partnerschaftsbeziehungen zwischen den Benutzern des Portals ermöglichen. Über das Portal haben Familienangehörige, Pflegedienste oder der Hausarzt Zugriff auf die vom Benutzer freigegebenen Sensordaten und deren Auswertungen sowie die persönlichen Informationen wie Name, Alter und Anschrift.

Darüber hinaus dient das Portal als Marktplatz für eine Vielzahl von unterschiedlichen Diensten und Anwendungen rund um das Thema Assisted Living, der auch Drittanbietern offen steht und zur Attraktivität des Systems beiträgt. Damit externe Anbieter solche Dienstleistungen für den Benutzer erbringen können, müssen die Anbieter einerseits über eine Schnittstelle auf die im Portal über den Benutzer gespeicherten Informationen und Sensorwerte zugreifen können. Andererseits soll auch die direkte Integration und Darstellung dieser Dienstleistungen innerhalb des Serviceportals ermöglicht werden, und es müssen Vorkehrungen getroffen werden, welche den Benutzern die Kontrolle über den Zugriff auf ihre Daten gewährleisten und den unberechtigten Zugriff verhindern.

In SmartAssist wird hierfür die OpenSocial API verwendet, welche eine REST-basierende Webservice Schnittstelle für den externen Zugriff auf die Personendaten innerhalb des sozialen Netzwerkes definiert und durch sogenannte OpenSocial Gadgets die Möglichkeit zur Integration von externen Dienstleistungen in das Portal bietet. OpenSocial wird unter anderem von den drei großen deutschen VZ Netzwerken (MeinVZ, StudiVZ, SchülerVZ) sowie von iGoogle, XING und MySpace verwendet, so dass bereits eine Vielzahl von Anwendungen für diese Plattform existiert und viele Entwickler Erfahrungen im Umgang mit dieser API sammeln konnten. Zur Integration der API in SmartAssist wird die OpenSocial Referenzimplementierung Apache Shindig 2.0 mit Erweiterungen zum Zugriff auf Sensorwerte verwendet.

Die einzelnen Bestandteile werden in den folgenden Abschnitten genauer erläutert und anhand von Beispielen veranschaulicht.

1.1.1. Benutzungsschnittstelle

Das Serviceportal wurde mit Hilfe des Content-Management-Systems Drupal 6 implementiert und kann von jedem Benutzer mittels eines beliebigen Webbrowsers benutzt werden. Der Benutzer identifiziert sich über einen Benutzernamen und das zugehörige Passwort. Die gesamte Kommunikation wird hierbei mittels SSL/TLS verschlüsselt. Nach dem erfolgreichen Einloggen in das Portal können die Benutzer eine Reihe von Nachrichtenartikeln rund um das Projekt SmartAssist und die Benutzung des Portals lesen oder aber mittels der Hauptnavigation zu den einzelnen Bereichen des Portals navigieren.

Die wesentlichen Punkte der Hauptnavigation werden im Folgenden beschrieben.

Unter dem Menüpunkt „Mein Profil“ hat der Benutzer die Möglichkeit, seine Eigenen Profilinformationen zu verwalten. Dies ist am Beispiel der Profilseite des Benutzers „Peter“ in Abbildung 5 dargestellt: Unter der Hauptnavigation befindet sich die Unternavigation, welche neben der Ansicht auch die Bearbeitung des Profils ermöglicht sowie zur Ansicht der Senso-

ren des Benutzers führen kann. Darunter befinden sich zwei Menüpunkte zur Interaktion mit den Paten dieses Benutzers: Falls jemand das Profil des Benutzers „Peter“ besucht, kann er durch einen Klick auf diese beiden Menüpunkte mit den Paten dieses Benutzers Kontakt aufnehmen, indem er eine Benachrichtigung für diese erstellt („Bitte Hausbesuch durchführen“ / „Bitte anrufen und nachfragen, ob alles in Ordnung“) oder den SmartAssist Fragebogen ausfüllt, nachdem er diesen Benutzer besucht hat. Der Fragebogen enthält insgesamt 10 Multiple Choice Fragen zu unterschiedlichen Bereichen (Bewusstsein, Ernährungszustand, Umfeld, ...) und kann der Einschätzung des Gesundheitszustandes des Benutzers durch einen Arzt dienen.

Der darunterliegende „Status“ Bereich kann vom Benutzer dazu verwendet werden, seinen subjektiven Gesundheits- und Gemütszustand zu erfassen, besondere Vorkommnisse oder Ereignisse (Ausflüge, Spaziergänge, Geburtstage) festzuhalten und seinen Freunden und Paten in SmartAssist mitzuteilen. Zusätzlich können andere Benutzer auf diese Weise Nachrichten für den Benutzer hinterlassen und auf diese Weise mit ihm kommunizieren. Darunter befinden sich der Bereich mit den Profilinformatoren des Benutzers und die Sicherheitseinstellungen. Hier kann der Benutzer sich einerseits seine Profilinformatoren anzeigen lassen, diese bearbeiten und den Zugriff verwalten. So hat er die Möglichkeit, seine Benutzerdaten entweder für alle Benutzer von SmartAssist (öffentliches Profil), nur für seine Freunde und/oder Paten sowie für niemanden freizugeben. Dieselben Optionen stehen ebenfalls für seine Sensordaten zur Verfügung.

Über den Menüpunkt „Beziehungen“ können die Benutzer untereinander Freund- und Patenschaften schließen, wobei zwei Benutzer sowohl Freunde als auch Paten sein können. In Abhängigkeit von der gewählten Beziehung, werden die im Profil hinterlegten Sicherheitseinstellungen wirksam. Freundschaften sollen in erster Linie dem gegenseitigen Austausch dienen, so dass befreundete Benutzer miteinander über den SmartAssist Chat direkt kommunizieren können, wie in Abbildung 6 dargestellt. Patenschaften hingegen dienen dazu, auf andere Benutzer bewusst aufzupassen und sollten daher vor allem von den engsten Verwandten oder dem Hausarzt verwendet werden. Beide Beziehungsarten können jedoch von beliebigen Benutzern eingegangen werden, benötigen jedoch ein beidseitiges Einverständnis.

Der Menüpunkt „Sensoren“ erlaubt es dem Benutzer, die Sensoren in seiner eigenen Wohnung zu verwalten. Unabhängig von der Anzahl oder dem Einsatzort der Sensoren kann sich der Benutzer hier dazu entscheiden, nur eine Untermenge der Sensoren in seinem Haushalt im Portal zu aktivieren oder einen bestimmten Sensor zu einem späteren Zeitpunkt wieder zu entfernen (z.B. weil der Sensor defekt ist oder aber weil der Benutzer seine Privatsphäre verletzt fühlt). Er kann darüber hinaus für jeden einzelnen Sensor einen Namen vergeben, mit welchem er den aktuellen Aufenthaltsort beschreiben kann (z.B. „Fensterbrett in der Küche“). Weitere Informationen zur Anzeige der Sensoren und der Visualisierung der Messwerte im SmartAssist Portal befinden sich im Abschnitt „Kommunikation mit der mobilen Plattform“.

Das SmartAssist Portal kann durch Dienste von Drittanbietern erweitert werden. Über den Menüpunkt „Anwendungen“ gelangt der Benutzer auf eine Übersichtsseite, auf welcher er die aktuell von ihm abonnierten Dienste benutzen kann. Darüber hinaus kann er neue Dienste aus dem Dienstkatalog abonnieren und das Abo für bereits abonnierte Dienste beenden und somit den Datenzugriff für diese Dienste sperren. Weitere Informationen finden sich im Abschnitt „Dienstimplementierung und Basisdienste“.

Der Zugriff auf die Profil- und Sensordaten des Benutzers durch die mobile Plattform kann über den Menüpunkt „Mobile Geräte“ erreicht werden. Weitere Informationen finden sich im Abschnitt „Kommunikation mit der mobilen Plattform“.



Start Benachrichtigungen Neuigkeiten Mein Profil Beziehungen Sensoren Anwendungen Mobile Geräte Forum

Ansicht Bearbeiten Sensoren dieses Benutzers Statusmeldungen

Peter Unternavigation des Profils

Interaktionsmöglichkeit mit den Paten

- Eine neue Benachrichtigung für die Paten von Peter erstellen
- Ich habe Peter besucht und möchte den SmartAssist Fragebogen ausfüllen.

Status

Wie geht es dir? Was gibt es Neues?

(Bis zu 140 Zeichen)

Teilen

Eingabefeld für eigene Statusmeldungen

CarstenBuschmann » Hi Peter, könntest du Jan und Jan Zugriff auf unsere Sensoren geben, bzw. uns sagen, wie wir das machen können?

Bearbeiten Löschen Antworten

JensMartinTraeder » Ich wäre gerne bei allen Teilnehmern der Projektgruppe in der Freundesliste, geht das?

Bearbeiten Löschen Antworten

LutzKleinfeld » Hallo! Würde mich mich über die Freundschaft freuen!!

Bearbeiten Löschen Antworten

Über mich

Vorname

Peter

Nachname

Rothenpieler

Geburtsdatum

26.02.1982

Geschlecht

männlich

Stadt

Lübeck

Interessen

Research, Teaching

Profilinformationen

Nachrichten anderer Benutzer

Sicherheit

Zugriff auf Benutzerdaten erlauben für

Alle Benutzer (Öffentliches Profil)

Zugriff auf Sensordaten erlauben für

Meine Paten und Freunde

Privatsphäreinstellungen

Verlauf

Mitglied seit

2 Jahre 44 Wochen

Abbildung 5 Profilsicht eines Benutzers

Der SmartAssist Chat (siehe Abbildung 6 links) zeigt in einer Liste die aktuell angemeldeten Freunde und Paten des jeweiligen Benutzers an und erlaubt es, sich gegenseitig Textnachrichten zuzuschicken. Dies dient dem direkten Austausch zwischen den angemeldeten Benutzern und der privaten Kommunikation, da sämtliche geschriebenen Texte nur zwischen diesen Benutzern ausgetauscht werden. Das Forum hingegen (siehe Abbildung 6 rechts) dient dem öffentlichen Informationsaustausch zwischen allen Benutzern von SmartAssist. Im Forum können Fragen zur Benutzung von SmartAssist gestellt werden, Ankündigungen gemacht werden und Hilfestellungen gegeben werden. Hierbei sind alle Benutzer von SmartAssist dazu eingeladen, sich über Themengebiete wie Gesundheit, Familie, Freunde und Bekannte oder die kleinen und großen Problemen des Alltags auszutauschen.

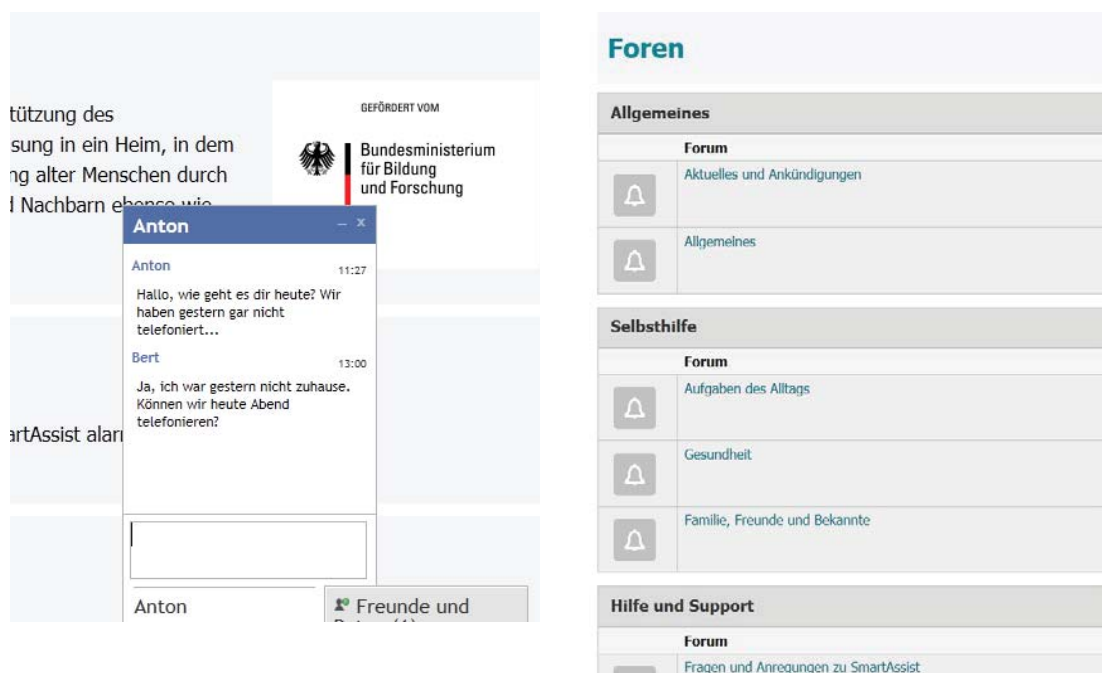


Abbildung 6: Forum und Chat von SmartAssist

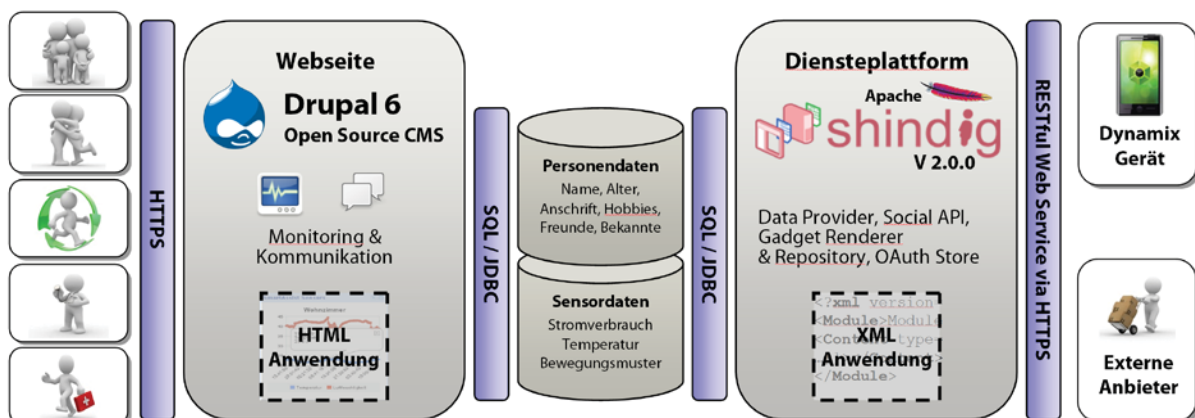


Abbildung 7: Überblick über die Dienstplattform von SmartAssist

1.1.2. Dienstplattform

Neben der für den Benutzer sichtbaren Benutzungsschnittstelle im Browser besteht das SmartAssist Serviceportal noch aus einer Dienstplattform, welche den Zugriff auf die im Portal gespeicherten Daten durch Computerprogramme und -Dienste ermöglicht. Benutzungsschnitt-

stelle und Dienstplattform teilen sich einen gemeinsamen Datenbestand wie in Abbildung 6 dargestellt ist: Der Benutzer kann mittels seines Browsers über die Webseite seine Personendaten (Name, Alter, Anschrift, ...) eingeben und die Rechte für den Zugriff definieren wie in den vorangegangenen Abschnitten erläutert wurde. Die Sensordaten werden automatisch durch die im häuslichen Umfeld verteilten Sensoren gesammelt. Sowohl die Personen- als auch die Sensordaten werden in einer Datenbank gespeichert.

Die Dienstplattform wurde auf Basis von Apache Shindig¹⁷ in der Version 2.0 implementiert. Bei Shindig handelt es sich um eine open source-Implementierung des OpenSocial Standards in der Programmiersprache Java, welche in SmartAssist als Servlet auf einem Apache Tomcat Server installiert wurde. Apache Shindig wurde in vielen Teilbereichen an die Bedürfnisse von SmartAssist angepasst: So wurden zum einen sog. Provider für die Personendaten geschrieben, welche die Abbildung der Profilinformatoren in der SmartAssist Datenbank auf die OpenSocial Profilinformatoren erlauben. Dies geschah analog für die Beziehungen zwischen den einzelnen Benutzern, wobei neben den Freundschaftsbeziehungen auch Patenschaften unterstützt werden. Darüber hinaus wurden zusätzliche Schnittstellen für den Zugriff auf die Liste der Sensoren eines Benutzers sowie für die Messwerte der einzelnen Sensoren geschaffen und in Shindig integriert.

Der Zugriff auf die Daten über das OpenSocial Protokoll kann auf zwei Arten geschehen:

- Zugriff über die REST Schnittstelle
- Zugriff über die Javascript Schnittstelle (JSON RPC)

Hierbei kann die REST Schnittstelle von einem beliebigen Computerprogramm verwendet werden, während die Javascript Schnittstelle für die Dienste der Drittanbieter in Form von sogenannten OpenSocial Gadgets gedacht ist. Die REST Schnittstelle wird unter anderem von dem SmartAssist Context Plugin für die mobile Plattform verwendet, um auf die Personen- und Sensordaten eines Benutzers zugreifen zu können.

Die Authentifizierung erfolgt im Fall der REST Schnittstelle über OAuth18, wobei in diesem Fall das 2-Legged OAuth verwendet wird, d.h. der Benutzer erstellt zunächst einen Zugang in Form eines Consumer key und Consumer secret erstellt (ähnlich einer Benutzernamen/Passwort Kombination). Dieser Zugang wird vom Benutzer für einen bestimmten Consumer erstellt, wobei ein Consumer einem mobilen Gerät (z.B. ein Smartphone) des Benutzers oder einem bestimmten Drittanbieter entsprechen kann. Auf diese Weise kann jeder einzelne Zugang vom Benutzer zu einem späteren Zeitpunkt wieder gelöscht werden, ohne dass der SmartAssist Zugang des Benutzers oder die anderen Consumer davon beeinflusst werden.

Für die Javascript Schnittstelle wird ein sogenanntes Token zur Authentifizierung verwendet, welches beim Aufruf eines Dienstes durch den Benutzer zur Laufzeit vom SmartAssist Server erstellt wird und mehrere verschlüsselte Informationen enthält, welche die Art des Zugriffs durch das Token beschränken. Zum einen werden sowohl der Name des Dienstes, für welchen das Token ausgestellt wurde als auch der Name des Benutzers darin gespeichert, auf dessen Daten mit diesem Token zugegriffen werden dürfen.

Sämtliche Zugriffe, welche mittels der OAuth und der Token Authentifizierung durchgeführt werden, beinhalten zusätzlich einen (vom SmartAssist Server ausgestellten) Zeitstempel, welcher die Gültigkeitsdauer dieses Zugriffs einschränkt. Auf diese Weise wird sichergestellt, dass ein ausgestelltes Token nicht beliebig oft oder zu einem beliebigen späteren Zeitpunkt

¹⁷ Siehe: <https://shindig.apache.org/>

¹⁸ Siehe: <http://oauth.net/>

wiederverwendet werden kann (Replay Angriff). Auf welche Daten des Benutzers zugegriffen werden kann, wird im Abschnitt „Dienstimplementierung und Basisdienste“ beschrieben.

Eine Übersicht der zur Verfügung stehenden Schnittstellen befindet sich in Tabelle 2, welche diese am Beispiel der REST Endpunkte vorstellt.

	Schnittstelle & Parameter
Personen	/social/rest/people/{ID}/@self Profil des Benutzers {ID}, z.B. Peter (GET, PUT, POST, DELETE) /social/rest/people/{ID}/@friends Alle Freunde des Benutzers {ID} (GET, PUT, POST, DELETE)
Anwendungsdaten	/social/rest/appdata/{ID}/@self/{appid} Daten von Benutzer {ID} zur Anwendung {AppID} (GET, PUT, POST, DELETE)
Liste der Sensoren eines Benutzers	/sensors/get_sensors_of_user?{ID} Liste aller Sensoren dieses Benutzers (GET)
Sensordaten	/sensors/get_data_by_node?&device_id={sensor} Daten eines Sensors abrufen (Start und Endzeitpunkt müssen angegeben werden) (GET)

Tabelle 2: Schnittstellen der Dienstplattform

1.1.3. Kommunikation mit den Sensorkomponenten und Darstellung im Portal

Um auf die in den Haushalten gesammelten Daten im Portal zugreifen zu können, wurde die OpenSocial Implementierung erweitert. Die Kommunikation mit der vom Lübecker Wachunternehmen bereitgestellten Infrastruktur folgt hierbei der technischen Spezifikation des „FleetGuide Geographical Processing Information (GPI) Interface“ der Firma CProjekt. Diese erlaubt es, die Daten der einzelnen Sensoren, welche jeweils einem Haushalt zugeordnet sind, über eine REST Schnittstelle abzufragen. Hierbei können sowohl die jeweils aktuellsten Werte als auch der zeitlichen Verlauf abgerufen werden.

Die auf diese Art und Weise abgerufenen Daten werden von der SmartAssist Plattform über das „Chart Tools Datasource“¹⁹ Protokoll zur Verfügung gestellt. Dieses Protokoll wurde von Google entwickelt und unterstützt eine SQL-ähnliche Anfragesprache, wodurch die abgerufenen Daten gefiltert werden können (z.B. „Temperaturwerte von Sensor A in einem bestimmten Zeitraum“). Die Daten können von dieser Datenquelle entweder im JSON, CSV oder HTML Format ausgegeben werden. Durch die Verwendung dieses Protokolls konnte für die Darstellung der Daten im ServicePortal auf eine Vielzahl von Visualisierungen zurückge-

¹⁹ Siehe: https://developers.google.com/chart/interactive/docs/dev/implementing_data_source

griffen werden, welche die Google Visualization API²⁰ verwenden. Ein Beispiel für eine solche Visualisierung ist in Abbildung 8 zu finden, welche die Daten von zwei unterschiedlichen Sensoren im Haushalt einer Testperson zeigen. Auf der linken Seite befindet sich die Darstellung des Temperatur- und Luftfeuchtigkeitsverlaufs im Wohnzimmer, während auf der rechten Seite die Anzahl von Bewegungs-Ereignisse im Flur der Wohnung dargestellt werden, welche durch einen Bewegungssensor (PIR Sensor = Passive Infrared Sensor) aufgezeichnet wurden. Beide Diagramme sind interaktiv, d.h. der Betrachter kann mit der Maus über die dargestellten Kurven fahren und erhält für den jeweils markierten Datenpunkt die Messwerte sowie den Zeitpunkt der Messung im oberen Bereich des Diagramms dargestellt – z.B. links eine Temperatur von 30.2°C und eine relative Luftfeuchtigkeit von 40.5% am Freitag, den 27.07.2012 um 11:28 und 32 Sekunden. Des Weiteren kann der Benutzer mit dem Mausrad im Diagramm zoomen und mittels eines Klicks den angezeigten Ausschnitt verändern.

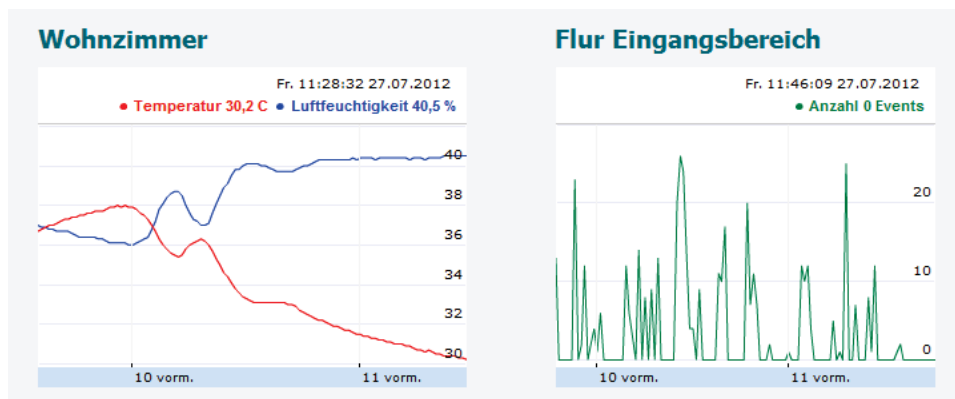


Abbildung 8: Visualisierung der Sensorwerte im Portal

1.1.4. Kommunikation mit der mobilen Plattform

Der Datenaustausch mit der mobilen Plattform (siehe Kap. 1.3) verwendet die OpenSocial Implementierung, welche bereits im Abschnitt über die Dienstplattform beschrieben wurde. Der Zugriff auf die Profildaten (Name, Alter, ...) geschieht dabei über die OpenSocial eigenen Schnittstellen, während die Sensorwerte über die im vorangegangenen Abschnitt beschriebenen Erweiterung mittels des „Chart Tools Datasource“ Protokoll übertragen werden, welcher in die OpenSocial Implementierung integriert wurde. Um den Zugriff auf diese Schnittstellen für den Anwendungsprogrammierer zu vereinfachen, wurde ein Plugin für das Dynamix Framework erstellt, welches die Daten in Form von Context Events bereitstellt. Hierfür muss ein Programmierer innerhalb seiner Anwendung zwei Context Subscriptions erstellen, wie in Abbildung 9 dargestellt. Die erste Subscription dient dazu, um die Profildaten des Benutzers zu erhalten, während die zweite für den Zugriff auf die Sensordaten dient.

```

1 dynamix.addContextSubscription(dynamixCallback, "de.smartassist.contextplugin.profileinfo");
1 dynamix.addContextSubscription(dynamixCallback, "de.smartassist.contextplugin.simplesensorvalues");

```

Abbildung 9: Erstellung von zwei Context Subscriptions für die Profil- und Sensordaten

²⁰ Siehe: <https://developers.google.com/chart/>

Die Context Events von SmartAssist werden durch einen ContextRequest abgerufen. In Abbildung 10 wird dies am Beispiel der Sensordaten gezeigt: Hierfür werden die einzelnen Parameter der Abfrage innerhalb eines sogenannten Bundle abgelegt. Die SensorId identifiziert den Sensor, dessen Daten abgerufen werden sollen, die Parameter beginTime und endTime geben den Zeitraum an, in welchem die Daten abgerufen werden sollen und der optionale Parameter Limit gibt an, wie viele Sensorwerte maximal abgerufen werden sollen. Über den Parameter Limit können z.B. nur die neuesten 25 Werte in einem Zeitraum abgerufen werden. Eine Liste der SensorIds eines bestimmten Benutzers ist über die Profildaten verfügbar, wobei auch nur genau diese SensorIds für diesen Benutzer abgefragt werden können. Ein Zugriff auf die Sensorwerte eines anderen Benutzers ist auf diese Art und Weise nicht Möglich (siehe unten). Dies entspricht den Schnittstellen, wie sie in Tabelle 2 beschrieben sind.

```

1 // Obtain the user's sensor data
2 Bundle config = new Bundle();
3 config.putString("sensorId", "123456789"); // Sensor id from user's profile
4 config.putString("beginTime", "2012-09-25%2014:00:00"); // Query start time
5 config.putString("endTime", "2012-09-25%2014:10:00"); // Query end time
6 config.putString("limit", "25"); // Optional response limit
7 String requestId = dynamix.configuredContextRequest(dynamixCallback,
8     "de.smartassist.contextplugin",
9     "de.smartassist.contextplugin.simplesensorvalues",
10    config);

```

Abbildung 10: Abfragen der Sensordaten durch einen Context Request

```

1 @Override
2 public void onContextEvent(ContextEvent event) throws RemoteException {
3     /*
4     * Log some information about the incoming event
5     */
6     Log.i(TAG, "onContextEvent received from plugin: " + event.getEventSource());
7     Log.i(TAG, "Event context type: " + event.getContextType());
8     Log.i(TAG, "Event timestamp " + event.getTimeStamp().toLocaleString());
9     if (event.expires())
10        Log.i(TAG, "Event expires at " + event.getExpireTime().toLocaleString());
11    else
12        Log.i(TAG, "Event does not expire");
13    // Check for native IContextInfo
14    if (event.hasIContextInfo()) {
15        Log.i(TAG, "Event contains native IContextInfo: " + event.getIContextInfo());
16        IContextInfo nativeInfo = event.getIContextInfo();
17        if (nativeInfo instanceof IProfileInfo) {
18            IProfileInfo data = (IProfileInfo) nativeInfo;
19            Log.i(TAG, "Received profile for: " + data.getId() + " with name " + data.getGivenName());
20        }
21        if (nativeInfo instanceof ISimpleSensorValues) {
22            ISimpleSensorValues data = (ISimpleSensorValues) nativeInfo;
23            for (ISimpleSensorValue i : data) {
24                Log.i(TAG, "Received data from: " + i.getSensor() + " with value " + e.getSensorValue());
25            }
26        }
27    }
28 }
29 }

```

Abbildung 11: Zugriff auf die Profil- und Sensordaten bei Erhalt eines Context Events

Nachdem ein ContextRequest abgeschickt wurde, wird dieser vom SmartAssist Server beantwortet und mit einem ContextEvent beantwortet. Auf die angefragten Daten kann dann wie in Abbildung 11 dargestellt zugegriffen werden. In Abhängigkeit von der Art des ContextEvent, können dann entweder die Profildaten oder die Sensordaten ausgelesen werden. Im Beispiel oben wird über „data.getGivenName()“ der Vorname des Benutzers ausgegeben, während „e.getSensorValue()“ einen der zurückgelieferten Sensorwerte ausgibt, welcher sich in einer Liste von „ISimpleSensorValue“ befindet.

Um mit dem mobilen Gerät auf die Sensor- und Profildaten eines Benutzers zugreifen zu können, muss dieser Benutzer einen der Zugriff für dieses Gerät erstellen. Ein Benutzer kann hierfür beliebig viele Geräte seinem Profil hinzufügen und für jedes einzelne Gerät einen getrennten Namen und ein Passwort erstellen, so dass er den Zugriff für ein ausgewähltes Gerät auch jederzeit wieder entfernen kann (z.B. wenn das Gerät gestohlen wird oder er der Person, welcher das Gerät gehört den Zugriff entziehen möchte). Die Authentifizierung des Geräts erfolgt über OAuth und der Datenaustausch wie auch bei der Dienstplattform verschlüsselt über HTTPS.

```
<?xml version="1.0" encoding="UTF-8"?>
<Module>
  <ModulePrefs
    title="Profile Data Demo"
    description="Dies ist eine Demo, ..."
    author="Peter Rothenpieler" author_email="rothenpieler@itm.uni-luebeck.de"
    thumbnail="https://rubus.itm.uni-luebeck.de/smart-gadgets/gadget_thumbnail.png"
    screenshot="https://rubus.itm.uni-luebeck.de/smart-gadgets/gadget_screenshot.png"
    height="600">

    <Require feature="opensocial-0.8" />

  </ModulePrefs>
  <Content type="html">
    <script type="text/javascript" src="https://www.google.com/jsapi"></script>

    <script type="text/javascript">
      gadgets.util.registerOnLoadHandler(loadViewer);

      function loadViewer() {
        var req = opensocial.newDataRequest();
        var params = {};
        params[opensocial.DataRequest.PeopleRequestFields.PROFILE_DETAILS] = [
          opensocial.Person.Field.ID,
          opensocial.Person.Field.NAME,
          opensocial.Name.Field.GIVEN_NAME,
          opensocial.Name.Field.FAMILY_NAME,
          opensocial.Person.Field.DATE_OF_BIRTH,
          opensocial.Person.Field.AGE,
          opensocial.Person.Field.THUMBNAIL_URL,
          opensocial.Person.Field.BODY_TYPE,
          opensocial.BodyType.Field.HEIGHT,
          opensocial.BodyType.Field.WEIGHT,
          opensocial.Person.Field.INTERESTS,
          opensocial.Person.Field.TAGS,
          opensocial.Person.Field.URLS,
          opensocial.Person.Field.GENDER,
          opensocial.Person.Field.EMAILS,
          opensocial.Person.Field.ADDRESSES,
          opensocial.Address.Field.STREET_ADDRESS,
          opensocial.Address.Field.POSTAL_CODE,
          opensocial.Address.Field.LOCALITY,
          opensocial.Address.Field.COUNTRY,
          opensocial.Person.Field.PHONE_NUMBERS
        ];
        req.add(req.newFetchPersonRequest(opensocial.IdSpec.PersonId.VIEWER, params), 'viewer');
        req.send(onLoadViewer);
      }
    </script>
  </Content>
</Module>
```

Abbildung 12 Beispiel einer Gadget Implementierung (Meta Daten und Initialisierung)

1.1.5. Dienstimplementierung und Basisdienste

Zusätzlich zu den Grundfunktionalitäten, welche das Portal seinen Benutzern bietet, kann das Portal in seiner Funktionalität durch die Anwendungen von Drittanbietern erweitert werden. Diese Dienste werden von den Drittanbietern in Form einer XML Datei entwickelt, welche sowohl HTML Elemente zur Gestaltung der Benutzungsoberfläche als auch Javascript Programmcode zur Steuerung der Funktionalität beinhalten können. Die XML Datei folgt hierbei

der durch den OpenSocial Standard vorgegebenen Gadget Spezifikation in der Version 2.0²¹. Ein Beispiel dafür, wie ein solches Gadget aufgebaut ist, findet sich in Abbildung 12, welche den Meta-Daten Bereich und die Initialisierung des Gadgets zeigt. Die Meta Daten enthalten Informationen zum Namen, der Beschreibung und dem Autor des Gadgets. In der darunterstehenden Funktion „loadViewer()“ wird ein OpenSocial DataRequest erstellt, welcher alle Profilwerte des aktuellen Benutzers erfragen soll (z.B. Name, Alter, Größe, Geschlecht). Dieser DataRequest wird mittels des Aufrufes „req.send()“ abgeschickt und gleichzeitig die Funktion „onLoadViewer“ als Callback Funktion registriert, welche sich beim Empfang der Daten um die Verarbeitung kümmern soll.

Die Funktion „onLoadViewer“ ist in Abbildung 13 dargestellt: Hier werden die einzelnen Elemente der Anfrage nach dem Empfang durch die Anwendung verarbeitet – so kann z.B. der Benutzername des aktuellen Benutzers durch „viewer.getDisplayName()“ abgerufen werden, während die geschachtelten Attribute wie der Vor- und Nachname als Unterelemente des gemeinsamen Oberelements „opensocial.Person.Field.NAME“ abgerufen werden können.

```
function onLoadViewer(data) {
    var viewer = data.get("viewer").getData();

    appendSocialData("Benutzername", viewer.getDisplayName());
    appendSocialData("Benutzer ID", getProperty(viewer, opensocial.Person.Field.ID));

    var name = viewer.getField(opensocial.Person.Field.NAME);
    appendSocialData("Vorname", getProperty(name, opensocial.Name.Field.GIVEN_NAME));
    appendSocialData("Nachname", getProperty(name, opensocial.Name.Field.FAMILY_NAME));

    var gender = getProperty(viewer, opensocial.Person.Field.GENDER);
    if (gender.displayValue == 'male') {
        appendSocialData("Geschlecht", "Männlich");
    } else {
        appendSocialData("Geschlecht", "Weiblich");
    }

    var bday = new Date(viewer.getField(opensocial.Person.Field.DATE_OF_BIRTH));
    var bday_format = bday.getDate() + "." + (bday.getMonth()+1) + "." + bday.getFullYear();
    appendSocialData("Geburtstag", bday_format);

    var phone = viewer.getField(opensocial.Person.Field.PHONE_NUMBERS);

    appendSocialData("Telefon: ", phone[0].getField('number'));

    var url = viewer.getField(opensocial.Person.Field.URLS);
    appendSocialDataURL("Homepage", url[0].getField('address'));

    var address = viewer.getField(opensocial.Person.Field.ADDRESSES);

    appendSocialData("Straße", address[0].getField(opensocial.Address.Field.STREET_ADDRESS));
    appendSocialData("Postleitzahl", address[0].getField(opensocial.Address.Field.POSTAL_CODE));
    appendSocialData("Stadt", address[0].getField(opensocial.Address.Field.LOCALITY));

    var bodytype = viewer.getField(opensocial.Person.Field.BODY_TYPE);
    appendSocialData("Größe", getProperty(bodytype, opensocial.BodyType.Field.HEIGHT)+' cm');
    appendSocialData("Gewicht", getProperty(bodytype, opensocial.BodyType.Field.WEIGHT)+' kg');

    appendSocialData("Hobbies", getProperty(viewer, opensocial.Person.Field.TAGS));
    appendSocialData("Interessen", getProperty(viewer, opensocial.Person.Field.INTERESTS));
}
}
```

Abbildung 13: Beispiel einer Gadget Implementierung (Verarbeitung der persönlichen Daten)

Auf diese Weise ist es der Anwendung möglich, die für die Erfüllung des angebotenen Dienstes notwendigen Daten des Benutzers abzufragen. Hierbei ist jedoch zu beachten, dass eine Anwendung nur genau dann auf die Daten eines Benutzers zugreifen kann, wenn dieser die

²¹ Siehe: <https://opensocial-resources.googlecode.com/svn/spec/2.0/Core-Gadget.xml>

Anwendung installiert hat und seine entsprechenden Daten in SmartAssist selbständig eingepflegt hat. Vor der Installation einer Anwendung werden dem Benutzer die von der Anwendung erfragten Daten präsentiert. Die Liste der erfragten Daten muss hierbei zuvor vom Dienstbetreiber für die entsprechende Anwendung eingestellt werden und kann im Nachhinein nicht geändert werden, im Gegensatz zur Implementierung, welche der Dienstbetreiber zur Laufzeit aktualisieren kann, um neue Funktionen hinzuzufügen oder Fehler zu beseitigen. Auf diese Weise ist sichergestellt, dass der Benutzer nur genau die Daten an die Anwendung weitergibt, welche er auch weitergeben möchte. Darüber hinaus kann ein Benutzer eine Anwendung jederzeit wieder deinstallieren.

Zur Veranschaulichung, welche Dienste auf diese Art und Weise implementiert werden können, wurden mehrere Basisdienste für das SmartAssist Portal entwickelt, wovon eine Auswahl in Abbildung 14 dargestellt ist. Einige dieser Dienste (Lieferdienst, Apotheken Finder) verwenden die öffentliche API eines Drittanbieters, um die Funktionsfähigkeit der SmartAssist-Plattform zu veranschaulichen.

The screenshot displays four service widgets in a grid layout:

- BMI Rechner:** Shows user data for Peter (182.5 cm, 79.4 kg, 30 years) and a BMI of 23.84 kg/m² (Normalgewicht). A table lists BMI categories from 'starkes Untergewicht' to 'Adipositas Grad III'.
- Ihr Horoskop:** Displays the zodiac sign for Peter (Fische) and a horoscope text. It also shows horoscopes for friends.
- Apotheken Finder:** Shows search criteria for pharmacies in Lübeck (Ratzeburger Allee 160, 23562) and a 'Notdienst finden' button.
- Geburtstags-Erinnerer:** Lists birthdays of friends with their dates and ages.

Abbildung 14: Basisdienste im Portal

- **BMI Rechner:** Diese Anwendung ruft den Namen, das Geburtsdatum, die Größe und das Gewicht des Benutzers ab. Hieraus wird dann das Alter und der BMI des Benutzers errechnet und eine Einordnung in eine BMI-Kategorie vorgenommen.
- **Ihr Horoskop:** Diese Anwendung ruft den Namen und das Geburtsdatum des Benutzers und seiner Freunde ab und errechnet das Sternzeichen der einzelnen Benutzer. Auf der Basis des Sternzeichens wird dann ein Horoskop angezeigt.
- **Lieferdienst:** Diese Anwendung ruft den Namen und die Postleitzahl des Benutzers ab und kann diese an die Suchmaschine „Lieferheld.de“ übergeben, wodurch eine Übersicht der Lieferdienste angezeigt wird, welche an die angegebene Postleitzahl liefern. Die Seite „Lieferheld.de“ bietet hierfür eine Schnittstelle an, welche von SmartAssist angesteuert werden kann. Diese Anwendung zeigt, wie ein externes Angebot aus einer SmartAssist Anwendung heraus angesprochen werden kann.
- **Apotheken Finder:** Diese Anwendung ruft den Namen und die Adresse (Straße, Hausnummer, Postleitzahl und Ort) des Benutzers ab und kann diese an die Suchmaschine

„Apotheke.com“ übergeben, wodurch eine Übersicht der Apotheken angezeigt wird, welche sich in der Nähe der angegebenen Adresse befinden und welche Apotheke für den Apotheken-Notdienst zuständig ist. Die Seite „Apotheke.com“ bietet hierfür eine Schnittstelle an, welche von SmartAssist angesteuert werden kann. Diese Anwendung zeigt, wie ein externes Angebot aus einer SmartAssist Anwendung heraus angesprochen werden kann.

- Geburtstag-Erinnerung: Diese Anwendung ruft den Namen und das Geburtsdatum des Benutzers und seiner Freunde ab und errechnet, in wie vielen Tagen die entsprechende Person Geburtstag hat. Die sortierte Liste (nächste Geburtstage oben) wird dann angezeigt.

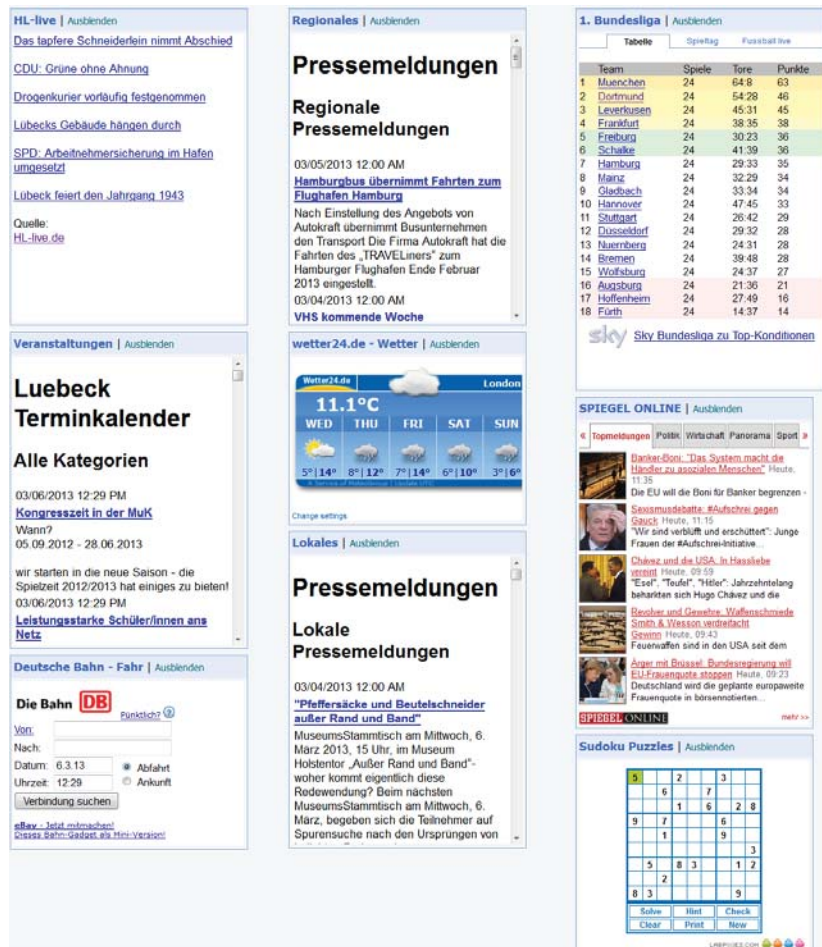


Abbildung 15: Externe Dienste (von iGoogle übernommen)

Zusätzlich wurden noch weitere externe Dienste integriert, welche ursprünglich von Dritten für andere OpenSocial Plattformen entwickelt wurden und welche durch die Nutzung des OpenSocial Standards ohne weiteres in SmartAssist lauffähig sind. Hierzu gehören mehrere Dienste, welche für die Plattform iGoogle entwickelt wurden und welche in Abbildung 15 dargestellt sind, z.B. Wettervorhersage von wetter24.de, die Tabelle der Fußballbundesliga, Nachrichten von Spiegel Online, die Fahrplanauskunft der Deutschen Bahn und ein Sudoku Puzzle von Labpixies. Außerdem wurden die RSS Nachrichtenfeeds mehrerer Lübecker Nachrichtenseiten integriert (HL-live.de, Veranstaltungen und Regionale sowie Lokale Pressemeldungen von Luebeck.de).

1.2. Sensordatenanalyse

Die Sensoren liefern die gemessenen Daten an einen zentralen Server. Aus diesem Server werden die Daten abgerufen und ausgewertet. Aus praktischen Gründen erfolgt dieses nicht im selben System. Für die Auswertung werden die Daten in zwei Gruppen eingeteilt: in der Anfangszeit dienen die Daten als Trainingsdaten, das heißt, die Algorithmen können anhand dieser Daten die Gewohnheiten der Bewohner in den überwachten Wohnungen erlernen. Anschließend erfolgt eine statistische Auswertung, ob die Gewohnheiten sich in nicht normalen Maßen ändern oder ob große Diskrepanzen zu den Messungen in der Trainingsphase existieren.

Die Auswertung erfolgt nach dem Prinzip der Ereignisdetektion: die Messwerte, die als Referenz dienen (Trainingsdaten), beinhalten nur Messwerte des Normalfalls, das heißt, zu diesen Zeitpunkten ist der Kunde gesund im Sinne der Selbstständigkeit und ein Eingriff von außen ist nicht notwendig. Von ihm werden die Ereignisse (Zeitpunkte, an denen eine Intervention notwendig ist) getrennt.

Dabei sind mehrere Besonderheiten des Netzwerkes und der Aufgabenstellung zu beachten:

- Die Sensoren sind batteriebetrieben und drahtlos. Systembedingt kommt es hierbei zu Verzögerungen sowie einzelnen Ausfällen bei der Übertragung. Ferner werden die Daten zu unterschiedlichen Zeitpunkten übermittelt. Die Algorithmen müssen demnach in der Lage sein, Werte an festen Zeitpunkten aus den gemessenen Daten zu extrahieren, um eine Kontinuität der Auswertung zu garantieren.
- Die Haushalte können mit unterschiedlichen Sensoren ausgestattet werden. Ferner können sie an unterschiedlichen Stellen im Haushalt platziert werden. Folglich ist es wichtig, den Algorithmus ohne Informationen der Natur und der Position des Sensors anzuwenden.
- Sollte ein Ereignis detektiert werden, muss eine Meldung in das System eingepflegt werden. Somit muss die Ereignisdetektion mit dem Portal kommunizieren.
- Die Algorithmen dienen der Analyse der Verhaltensmuster. Dennoch muss die Privatsphäre geschützt werden. Die Algorithmen müssen also sowohl aus den Daten ein Verhaltensmuster extrahieren als auch Rückschlüsse auf das tatsächliche Verhalten von außen her nicht ermöglichen.

Die Anforderungen ergeben sich aus der Problemstellung, dass mit Hilfe typischerweise nicht-idealer Sensoren ein Verhaltensmuster rekonstruiert werden muss. Wie viele Sensoren in zukünftigen Ausfertigungen des Systems Verwendung finden, ist ebenfalls nicht festgesetzt. Das Projekt ist durchgehend auf eine große Adaptivität ausgelegt, dieses muss ebenfalls im Algorithmus zur Datenanalyse bedacht werden.

Im Nachfolgenden werden die Lösungen zu diesen Anforderungen erläutert und der Algorithmus kurz vorgestellt. Zu den Details des Algorithmus sei hier auf die entsprechenden Veröffentlichungen verwiesen.

1.2.1. Merkmalsextraktion

Die Daten werden in einem zentralen Server zwischengespeichert. Neben den Rohdaten werden außerdem der Typ des Sensors, eine Identifikationsnummer, der Batteriestatus, der Zeitpunkt der Messung sowie im Einzelfall auch spezielle Parameter übermittelt. Von diesen Informationen werden insbesondere neben den Rohdaten die Zeitpunkte benutzt.

Die Zeitpunkte der Messungen unterscheiden sich wesentlich. Daher werden die Daten zunächst in einem Intervall von 60 Minuten gemittelt. Dadurch ergibt sich eine durchgehende

Serie von Datenpunkten zu vordefinierten Zeitpunkten. Befinden sich zu wenige Daten eines Sensors in diesem Intervall (zum Beispiel, weil der Sensor wegen mangelnder Energie ausgeschaltet ist), werden sie nicht zur Klassifikation berücksichtigt.

Abbildung 16 zeigt das Flussdiagramm des Klassifikators zur Ereignisdetektion. Die gemittelten Daten werden zunächst quantisiert. Dieses ist insbesondere notwendig für die zur Ereignisdetektion verwendeten Verfahren. Die Quantisierungsstufen werden aus den gemessenen Daten ermittelt, wodurch die Verfahren unabhängig von der Skalierung der gegebenen Messwerte werden. In der Auswertungsphase werden dann die einkommenden Daten klassifiziert. Ist die Abweichung vom Normalfall zu groß, wird eine Benachrichtigung an das Portal gesendet.

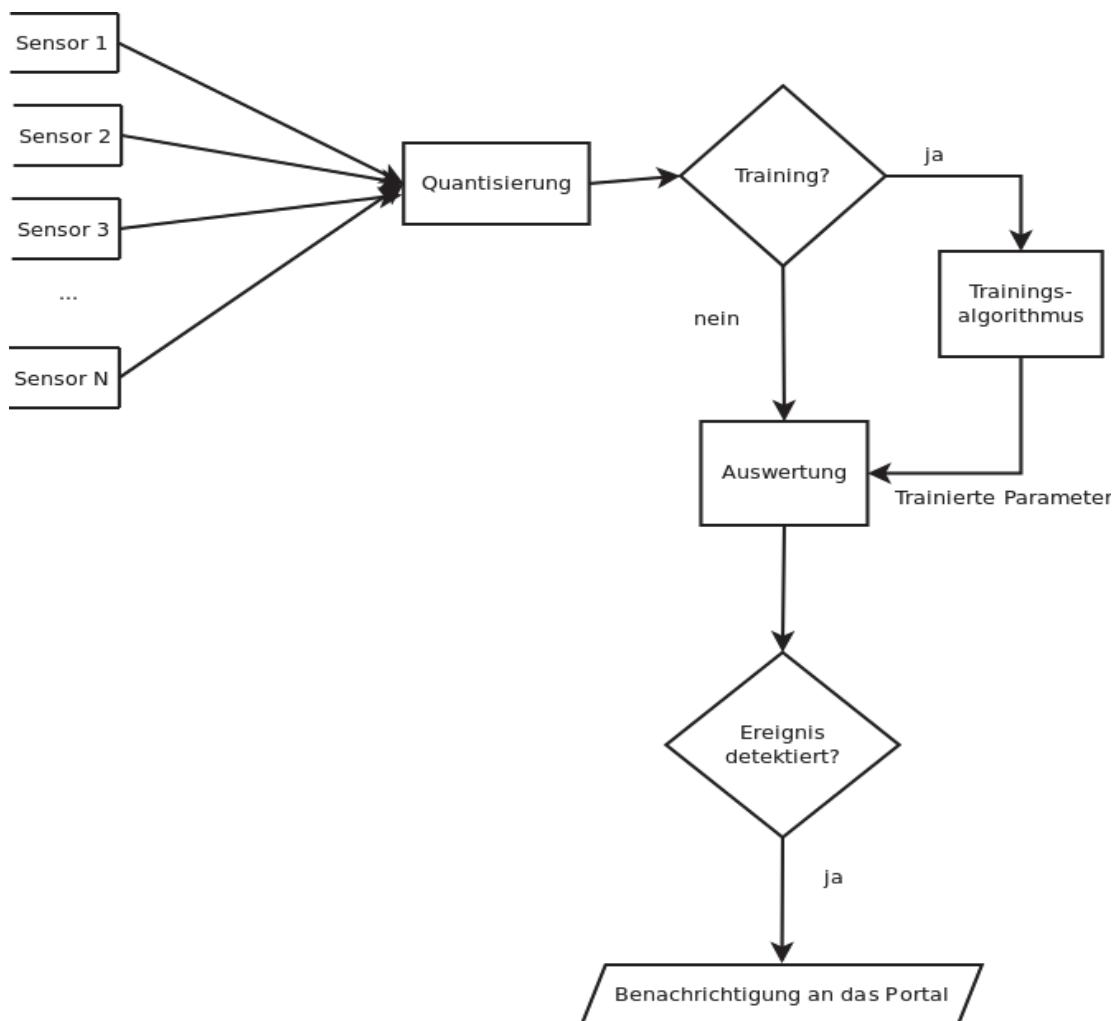


Abbildung 16: Flussdiagramm des Klassifikators zur Ereignisdetektion.

1.2.2. Merkmalsreduktion

Zur eventuellen Beschleunigung des Systems wurde untersucht, inwiefern sich die Merkmale reduzieren lassen. Insbesondere im Hinblick auf mögliche Ausfälle von Sensoren und deren Berücksichtigung im Detektionssystem hat sich jedoch die direkte Verwendung der Merkmale (ohne Linearkombination im Rahmen einer linearen Diskriminanzanalyse oder anderer Reduktionsverfahren) als vorteilhaft erwiesen.

1.2.3. Klassifikator

Der verwendete Klassifikator basiert auf der Theorie von bedingten Markov-Zufallsfeldern (conditional random fields). Diese wurden für die Ereignisdetektion angepasst.

Markov-Zufallsfelder sind oft verwendete Methoden zur Modellierung von zeitlichen Abläufen. Da das Verhalten der Bewohner ebenfalls ein zeitlicher Ablauf ist, können wir annehmen, dass die gemessenen Sensorwerte ähnliche zeitliche Charakteristika haben. Somit sind Markov-Zufallsfelder geeignete Methoden für die Modellierung der Verhaltensmuster.

Conditional Random Fields werden oft auf vektorielle, binäre Daten angewendet. Hierfür dient auch die Quantisierung der Daten. Jedes Quantum stellt eine Komponente des Vektors dar. Der gemittelte Datenwert innerhalb eines speziellen Quantums ist an einer entsprechenden Position im Vektor entweder eins oder null. Diese Informationen werden von allen Sensoren zu den gegebenen Zeitpunkten in einem gemeinsamen Vektor zusammengefasst.

Auf diese Vektoren wird der Klassifikator trainiert. Die Trainingsdaten entsprechen Messungen in einem Anfangszeitraum von ca. drei Monaten, in denen die Bewohner begleitend überwacht werden, um sicherzugehen, dass die Voraussetzungen zum Training des Systems erfüllt sind. Zum Beispiel soll der Bewohner in dieser Trainingsphase bei voller Gesundheit sein. Nach dem Training werden neu eintreffende Vektoren mittels des Klassifikators analysiert.

1.2.4. Analyse der Daten

Die Daten werden mittels des auf Conditional Random Fields basierten Klassifikators analysiert. Dabei wird untersucht, wie ähnlich sie zu den üblichen Messungen zu entsprechenden Tageszeiten sind. Sind sie sehr ähnlich, wird angenommen, dass keine Intervention notwendig ist. Weichen diese Messungen von den üblichen Messungen ab, so wird angenommen, dass ein Ereignis eingetreten ist, welches eine Intervention erfordert. Insbesondere wenn eine solche Situation über einen längeren Zeitraum vorliegt, wird diese als Ereignis klassifiziert.

Da sich Gewohnheiten täglich zu bestimmten Zeiten wiederholen, wurde das Modell so gewählt, dass die Messwerte entsprechend dem Zeitraum zugeordnet werden (zum Beispiel liegen die Schlafenszeiten in einem üblichen individuellen Zeitraum; gleiches gilt für Koch- oder Abwesenheitszeiten). Variationen in diesen Gewohnheiten werden durch den Trainingszeitraum im Modell berücksichtigt, das heißt dieser wird so groß gewählt, dass die üblichen Variationen und dadurch Abweichungen vom Tagesrhythmus erfasst werden.

1.2.5. Unterschiedliche Konfiguration der Haushalte

Der Klassifikator basiert auf Conditional Random Fields und somit letztendlich auf dem Prinzip der maximalen Entropie. Das heißt, dass aus den gegebenen Daten ein sehr hohes Maß an Information genommen wird. Die größte Information bedeutet in diesem Fall, dass eine größere Anzahl von Sensoren und damit mehr Messwerte eine genauere Klassifikation ermöglichen, da alle Sensoren zu jeder Zeit des Tages Informationen sammeln.

Andererseits muss ebenfalls damit umgegangen werden, dass einzelne Sensoren über längere Zeiträume ausfallen könnten. Dieses kann aufgrund von Defekten, notwendigen Batterie-wechseln oder durch Störungen der drahtlosen Übertragung auftreten. Hierbei handelt es sich um übliche Probleme vergleichbarer Systeme, die in der Auswertung unbedingt mit zu berücksichtigen sind. Durch den Ausfall eines Sensors wird zwar die Genauigkeit der Klassifikation herabgesetzt, es ist allerdings weiterhin möglich, die Klassifikation durchzuführen.

Die Adaptivität ist ein großer Vorteil der genutzten Algorithmen. Ferner sind sie in der Lage, die Klassifikation mit sehr unterschiedlichen Sensoren durchzuführen. Das Sensornetz ist zu-

dem auch erweiter- und reduzierbar. Sensoren lassen sich ab- und anschalten sowie neue Sensoren hinzufügen.

1.2.6. Übermittlung der Ereignisse an das Portal

Zur Auswertung wurde eine Übermittlung an das Portal implementiert; das System zur Ereignisdetektion kann unterschiedliche Stufen der Dringlichkeit senden. Sollte eine Messung einem Ereignis entsprechen, das heißt, sollte nach der Auslegung der Daten eine Intervention stattfinden, so wird zunächst eine Nachricht mit niedriger Dringlichkeitsstufe gesendet. Sollte dieser Zustand über einen längeren Zeitraum anhalten, so wird eine Nachricht höherer Dringlichkeit gesendet. Nähere Informationen zum Portal sind im Abschnitt 1.1 zu finden.

1.2.7. Einhalten der Privatsphäre

Es werden ausschließlich die detektierten Ereignisse an das Service-Portal gesendet. Genaue Informationen über den Inhalt der Ereignisse werden nicht weitergegeben. Dadurch obliegt es autorisierten Personen (Patent, insbesondere einem vertrauten Arzt) zu entscheiden, welche Informationen über den gesundheitlichen Zustand der Bewohner weitergegeben werden und in welcher Form dies geschehen soll.

Insbesondere ermöglichen es die verschiedenen Abstraktionsebenen, dass die Analyse in rückwärts gerichteter Form ausgeschlossen werden kann. Es ist also nicht möglich, den Tagesablauf der Personen zurückzuverfolgen, weder mit Hilfe der an das Portal gesendeten Informationen noch durch Zugriff auf den Server, der für die Auswertung genutzt wird. Dadurch ist die Privatsphäre trotz der Analyse der Gewohnheiten mit Hilfe der Sensoren geschützt.

1.2.8. Experimente

Neben mehreren Experimenten, die insbesondere die Fähigkeiten der Algorithmen zur Ausführung der Klassifikation mit sehr unterschiedlichen Sensoren nachweisen, sind ferner auch Experimente durchgeführt worden, die die Klassifikation an sich genauer betrachten. Neben der eigentlichen Klassifikation wurden auch Experimente durchgeführt, bei denen Sensorausfälle simuliert wurden.

In den folgenden Abbildungen wird ein Maß dafür gezeigt, wie ähnlich eine Messung zu einem bestimmten Zeitpunkt zu den üblichen Werten ist. Dieses Maß wird durch das Formelzeichen Q bezeichnet und ist als $Q = -\log(p(s(t)))$ definiert. Dabei gilt, je höher ein Wert Q ist, desto ähnlicher ist der Sensormesswert den üblichen Werten. Ist der Q -Wert unter null, so wird ein Ereignis detektiert. Das Intervall, in dem hier gemessen wurde, ist ca. ein Monat. Die Schwankungen zeigen, dass die Zustände je nach Sensortyp unterschiedlich deutlich erkannt werden; das heißt, nicht alle Aktionen des normalen typischen Tagesverlaufs können mit den im gegebenen Beispiel verwendeten Sensoren erfasst werden.

In Abbildung 17 ist das Ergebnis eines der Experimente zu sehen, in dem einer der Sensoren über einen Zeitraum von mehreren Tagen ausgeschaltet wird. Die Ergebnisse sind vergleichbar zu den Zeiten, in denen alle Sensoren zur Klassifikation genutzt werden. Das bedeutet, dass die Klassifikation trotz Ausfall eines Sensors weiterhin funktioniert.

In Abbildung 18 ist das Ergebnis eines simulierten Ereignisses zu sehen. Dabei wurde eine Abweichung einzelner Sensorwerte in bisher nicht beobachtete Bereiche simuliert (zum Beispiel lieferte der Temperatursensor geringere oder höhere Werte als üblich, Bewegungen wurden stark reduziert oder erhöht). Der Algorithmus ist in der Lage, diese und ähnliche Ereignisse zu detektieren.

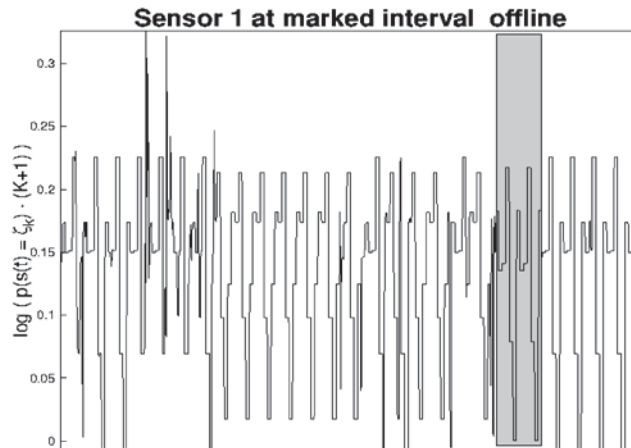


Abbildung 17: Datenanalyse mit einem deaktivierten Sensor über den markierten Zeitraum

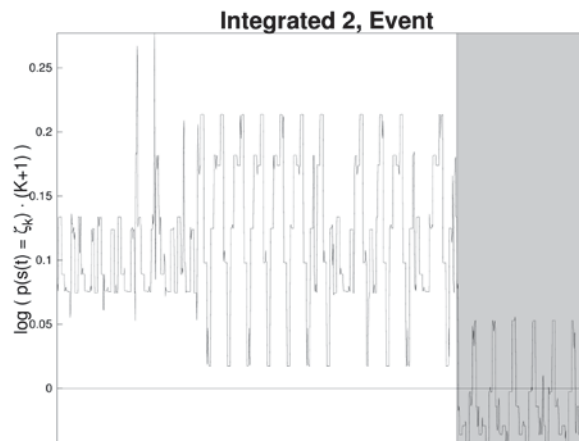


Abbildung 18: Simulation eines Ereignisses im markierten Zeitraum

In Abbildung 17 ist das Ergebnis eines der Experimente zu sehen, in dem einer der Sensoren über einen Zeitraum von mehreren Tagen ausgeschaltet wird. Die Ergebnisse sind vergleichbar zu den Zeiten, in denen alle Sensoren zur Klassifikation genutzt werden. Das bedeutet, dass die Klassifikation trotz Ausfall eines Sensors weiterhin funktioniert.

In Abbildung 18 ist das Ergebnis eines simulierten Ereignisses zu sehen. Dabei wurde eine Abweichung einzelner Sensorwerte in bisher nicht beobachtete Bereiche simuliert (zum Beispiel lieferte der Temperatursensor geringere oder höhere Werte als üblich, Bewegungen wurden stark reduziert oder erhöht). Der Algorithmus ist in der Lage, diese und ähnliche Ereignisse zu detektieren.

Ebenfalls verringert sich das Maß für die Normalität, wenn die Messungen nicht nur einzeln, sondern insgesamt abweichen. Dadurch werden auch stärkere Änderungen der Gewohnheiten detektiert, die auf eine Veränderung der Gewohnheiten hindeuten können, die baldige Interventionen benötigen. Dadurch erhöht sich die Wahrscheinlichkeit für ein Ereignis deutlich, auch im System. Schleichende Veränderungen werden somit ebenfalls detektiert.

1.2.9. Weitere Analysen

Die verwendeten Sensoren sind sehr unterschiedlicher Natur, so dass auch die erfassten Signale unterschiedliche Charakteristika aufweisen. Hier wurde unter anderem der Einfluss der Abtastrate untersucht. Generell zeigt sich, dass prinzipiell mehrere Messwerte pro Stunde nötig sind. Einzelne Sensoren sollten jedoch weitaus höher abgetastet werden. Insbesondere die Bewegungssensoren benötigen eine höhere Abtastung für eine genaue Analyse.

Für die räumliche Auflösung, gegeben durch die Anzahl und Position der Sensoren im Haushalt, gilt, dass sich der Zustand des Bewohners mit wachsender Auflösung auch genauer analysieren lässt. Eine Empfehlung für zukünftige Netzwerke lautet daher, dass möglichst viele Sensoren pro Haushalt anzustreben sind, um eine genauere Überwachung des gesundheitlichen Zustandes zu garantieren. Ein Sättigungseffekt konnte in den Experimenten noch nicht festgestellt werden, so dass noch keine Aussage darüber getroffen werden kann, wie viele Sensoren je Haushalt maximal sinnvoll sind. Mit dem aktuellen System lassen sich nur spezielle Ereignisse detektieren, es bestehen im üblichen Tagesverlauf mehrere Perioden, die nicht oder nicht genügend erfasst werden. Dieses Problem ist im Projekt SmartAssist bewusst in Kauf genommen worden, um eine höhere Akzeptanz der Bewohner gegenüber den momentan noch sehr auffälligen Sensoren zu erreichen.

1.2.10. Implementierung

Die Implementierung ist derzeit auf Mathworks Matlab, GNU Octave und in C++ lauffähig. Für den praktischen Einsatz ist die C++-Version vorgesehen. Die Implementierungen in Matlab und Octave sind insbesondere hilfreich für weitergehende Studien über Langzeitveränderungen, die aus technischen Gründen noch nicht durchgeführt werden konnten. Für derartige Studien werden Datensätze benötigt, die über deutlich längere Zeiträume erfasst wurden, als dies im durchgeführten Projekt möglich war. Ferner sind die Matlab- und Octave-Implementierungen anpassungsfähiger als die C++-Version. Hiermit sind veränderte Parameter und Alternativen in Zukunft leichter zu testen und zu verwenden. Die Implementierung in unterschiedlichen Programmiersprachen soll späteren Projekten, die eine stetige Modernisierung des gesamten Systems ermöglichen, den Einstieg erleichtern.

1.2.11. Zusammenfassung

Zur Datenanalyse der Sensordaten im Projekt SmartAssist wurden Algorithmen entwickelt, die insbesondere der allgemeinen Problemstellung der Ereignisdetektion zuzuordnen sind. Diese Algorithmen analysieren das Verhaltensmuster der Bewohner, sie respektieren aber dennoch deren Privatsphäre und lassen keine Rückschlüsse auf das Verhaltensmuster zu. Dies wird dadurch erreicht, dass nur die Abweichung, nicht aber der Regelfall oder die Art der Abweichung an das Portal gemeldet wird.

Die Algorithmen wurden in unterschiedlichen Szenarien und in einer Simulation von Ereignissen getestet, und ihre Effektivität wurde gezeigt. Dabei wurde auch nachgewiesen, dass das System durchaus auch mit einer Untermenge der üblichen Sensoren zur Ereignisdetektion genutzt werden kann. Sensorausfälle während des Betriebs führen somit nicht zum Ausfall des Systems. Weiterhin lassen sich neue Sensoren problemlos im laufenden Betrieb hinzufügen. Für diese neuen Sensoren wird dann eine eigene Trainingsphase durchlaufen, während die Detektion noch mit den bisher vorhandenen Sensoren erfolgt. Nach Ablauf des Trainings werden die neuen Sensoren dann mit in die Entscheidungsfindung aufgenommen.

1.3. Mobile Plattform

1.3.1. Einführung

Aktuelle mobile Geräte bilden eine hervorragende Basis für die Unterstützung von Smart-Assist-Nutzern sowohl im häuslichen Umfeld als auch unterwegs. Allerdings bildet die extreme Heterogenität unterschiedlicher Umgebungen eine Hürde für den Einsatz mobiler Anwendungen für das Erfassen der wechselnden Kontexte und das intelligente Interagieren mit den identifizierten smarten Objekten der jeweiligen Umgebung. Zur Lösung dieser Problematik wurde im Projekt die offene Plattform Ambient Dynamix („Dynamix“) entwickelt. Dynamix ist ein Community-zentrierter Plug-and-Play-Ansatz für die Entwicklung und Unterstützung kontext-sensitiver Anwendungen. Module für die sensorische Erfassung und aktorische Modifikation von Kontext werden zur Laufzeit als Plug-ins auf mobile Endgeräte geladen und den entsprechenden mobilen Anwendungen mit wenigen Zeilen Quellcode verfügbar gemacht.

Dynamix läuft als ein leichtgewichtiger Hintergrunddienst auf dem mobilen Gerät des Nutzers und erweitert das System um Sensorik, Verarbeitung und Kommunikation. Das einfache Application Programming Interface (API) von Dynamix unterstützt sowohl native als auch web-basierte Anwendungen (Apps) mit Kontext-Informationen. Dynamix erkennt automatisch die notwendigen plug-ins für einen gegebenen Kontext, lädt diese von einem Server und installiert sie auf dem Gerät. Wenn ein Nutzer seine Umgebung wechselt, werden im Bedarfsfall automatisch neue Plug-ins installiert oder vorhandene Plug-ins aktualisiert. Dies geschieht im Hintergrund, ohne die anfordernde Anwendung unterbrechen zu müssen. Dynamix bringt bereits eine wachsenden Satz von frei zu verwendenden Plug-ins. Durch offene Software Development Kits (SDKs) und ein skalierbares Repository werden Drittanbieter in der Entwicklung und Verbreitung von Plug-ins in der Community unterstützt

1.3.2. Technische Randbedingungen

Für die Unterstützung der Dynamix-Architektur werden die folgenden Anforderungen an Geräte und Betriebssysteme gestellt:

1. **Netzwerkunterstützung:** Zugang zum Netz ist für den Betrieb von Dynamix nicht zwingend erforderlich, allerdings muss das Gerät mit dem Internet verbunden sein, um automatischen Download und Installation von Plug-ins während der Laufzeit zu unterstützen. Für offline-Anwendungen kann eine statisches Set von Plug-ins auch vorab installiert werden. Für die meisten Anwendungen ist allerdings Kommunikation essenziell.
2. **Zugang zur Geräte-Hardware:** Die Erfassung von Kontext und die Techniken zur Modellierung basieren in Dynamix-Anwendungen zum großen Teil auf eingebaute Sensoren auf dem mobilen Gerät. Daher sind Geräte mit einer Vielzahl von Sensoren günstig. Zudem muss eine Möglichkeiten des dedizierten Zugriffs auf Sensor-Werte durch Dynamix gegeben sein. Idealerweise ist ein abgestuftes Sicherheitsmodell für die Kontrolle der Hardware gegeben. Die Dynamix-Plattform kann somit programmatischen Zugang unter Berücksichtigung von Privacy-Aspekten zur Verfügung stellen, z.B. für Netzwerk-Erkennung, Auswahl von Netzwerk-Interfaces, Betrieb von Bluetooth, Kamera-Zugriff, GPS-Positionsabfrage, etc.
3. **Hintergrundprozesse:** Das Design von Dynamix sieht einen Hintergrundprozess für die gleichzeitige Unterstützung von mehreren Anwendungen vor. Daher müssen Dynamix-Geräte eine entsprechende Möglichkeit der Einrichtung eines solchen Prozesses bieten (d.h. das Betriebssystem muss parallele Prozesse unterstützen). Insbesondere

empfehlenswert ist hierfür ein dediziertes Life-Cycle-Modell mit entsprechender Thread-Prioritätskontrolle.

4. Inter-Prozess-Kommunikation: Aufgrund des Architektur-Designs werden die Dynamix-Anwendungen in einem von der Plattform separierten Adressraum ausgeführt. Die Kommunikation zwischen den Anwendungen und der lokalen Dynamix-Komponenten erfordert daher Mechanismen der Inter-Process Communication (IPC). Dies umfasst z.B. Techniken wie Message Passing, Pipes, Sockets, etc.
5. Component Late binding: Dynamix setzt die Möglichkeit des Betriebssystems voraus, Software-Komponenten während der Laufzeit nachzuladen. Daher muss eine Dynamix-Plattform Techniken des Code-on-Demand-Access und Late Binding während der Laufzeit unterstützen (z.B. mit Hilfe eines Class-Loaders). Wichtig ist dabei die Unterstützung von sogenannten „Lazy-Loading“-Mechanismen mit einem dedizierten Sicherheits-Modell, optional auch Dependency-Management und Versionskontrolle von Komponenten.

1.3.3. Auswahl Betriebssystem

Unter Berücksichtigung der oben genannten technischen Randbedingungen wurden verschiedene mobile Plattformen für die potentielle Nutzung in SmartAssist untersucht. Dabei wurde nicht nur die Kompatibilität zu Dynamix, sondern auch die Marktdurchdringung in der EU berücksichtigt, insbesondere eine Marktvorhersage für die Laufzeit von drei Jahren. Aufgrund der beschränkten personellen Ressourcen im Projekt konnte nur eine Plattform ausgewählt werden und eine Unterstützung von mehreren Plattformen wurde ausgeschlossen. Wesentlicher ist daher die Unterstützung einer möglichst breiten Vielfalt von Endgeräten. Zu Beginn des Projekts stellte sich die Marktsituation aus Abbildung 19 dar.

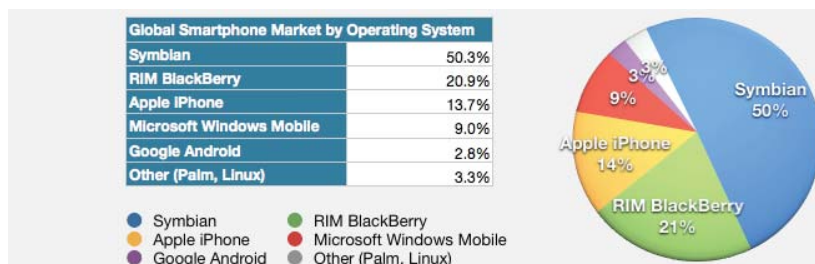


Abbildung 19: Weltweite Verteilung von Smartphone-Betriebssystemen im 2. Quartal 2009 (Canalys)

Hinter dieser Momentaufnahme sind mehrere wichtige Trends sichtbar. Der Marktführer (Symbian OS) hat signifikant Anteile verloren (von 72% in 2006 bis 50.3% in 2009). Einen großen Teil der Anteile wurde von Apple's iPhone übernommen, das seit 2007 eine immer wichtigere Rolle spielt. Während das Android-Betriebssystem von Google nur einen relativ kleinen Anteil von 2.8% am Markt hält, sind die Wachstumsraten enorm. Die aktuellen (2009) Android-basierten Initiativen von Firmen wie HTC, Dell, Samsung, LG, Kyocera, Sony Ericsson und Motorola verdeutlichen die zunehmende Bedeutung des freien Google-OS. Insbesondere unterstützt die Referenz-Implementierung von Android eine zunehmende Zahl unterschiedlicher Hardware-Plattformen, mit alleine 12 neuen Geräten bis zum Ende des Jahres 2009 und zahlreichen weiteren angekündigten Modellen in 2010 (inkl. Netbooks). Gartner Research sieht voraus, dass Android bis 2012 den zweiten Platz unter den Marktführern einnehmen wird. Unter diesen Voraussetzungen wurden fünf potentielle mobile Plattformen näher untersucht. Die Ergebnisse zeigt Abbildung 20.

	iPhone	MIDP 2	MIDP 3	Windows Mobile	Android
Network support	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Hardware access	<input type="checkbox"/> ²²	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Background proc.	<input type="checkbox"/>	<input type="checkbox"/> ²³	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Inter-process comm.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> ²⁴	<input checked="" type="checkbox"/>
Comp. late binding	<input checked="" type="checkbox"/>	<input type="checkbox"/> ²⁵	<input type="checkbox"/> ²⁶	<input type="checkbox"/> ²⁷	<input checked="" type="checkbox"/>

Abbildung 20: Erfüllung der Anforderungen von Dynamix durch aktuelle Betriebssysteme
(Legend: ☒ = not supported / □ = limited support / ☑ = full support)

Aufgrund der stark wachsenden Marktanteile und den vollständigen Unterstützung der Dynamix-Anforderungen wurde Google Android als die aussichtsreichste Plattform für die Entwicklung ausgewählt. Daraufhin wurden weitere detailliertere Untersuchungen in Android durchgeführt:

1. Android SDK, Dokumentation und Werkzeuge: Das Android SDK wurde installiert und eine Reihe von Beispiel-Anwendungen erfolgreich entwickelt. Insgesamt wurde die Entwickler-Dokumentation als hilfreich und aktuell beurteilt. Mehrere Bücher und Web-Ressourcen lagen vor und deuten auf eine aktive Entwickler-Community hin. Google bietet frei verfügbare Werkzeuge für Entwickler, die leicht in die Standard-Umgebung (IDE) von Eclipse integriert werden können. Dies umfasst einen Dalvik Compiler, ein Eclipse Build-System und einen vollständigen Geräte-Emulator. Für diesen Emulator wurden ebenfalls einige Test-Anwendungen entwickelt. Die Kommunikation zwischen Eclipse und dem Emulator war teilweise problematisch, aber insgesamt wurden die Werkzeuge als stabil eingestuft und wiesen ausreichende Möglichkeiten zur Fehlersuche.
2. OSGi-Kompatibilität: Da Dynamix mobilen Code erfordert, wurde OSGi als mögliche Grundlage evaluiert. Dafür wurden einige verfügbare OpenSource OSGi-Implementierungen getestet, darunter Equinox, Apache Felix, Knopflerfish und die mBedded Server Equinox Edition. Aufgrund von zahlreichen Test-Anwendungen wurden eine gute Android-Kompatibilität und ein relativ geringer Ressourcen-Verbrauch festgestellt. Um Felix in Android-Szenarien zu validieren, wurden einige Projekte als OSGi-Container realisiert und im Emulator getestet. Die Ergebnisse waren vielversprechend und Felix wurde auch aufgrund des kompatiblen Lizenzmodells (Apache 2.0) als OSGi-Plattform ausgewählt.

²² Limited hardware access

²³ Phone dependent

²⁴ Limited data can be sent

²⁵ Lacking dependency management, lacking dependency versioning

²⁶ Lacking dependency management, lacking dependency versioning

²⁷ Lacking dependency management, lacking dependency versioning, limited security model

Zusätzlich wurden in der Verlängerungsphase des Projekts auch iOS-basierte Geräte (z.B. iPhone und iPad) aufgrund der zunehmenden Dominanz von Apple im Bereich mobiler Geräte untersucht. Die Auswertungen ergaben zahlreiche technische Limitierung im iOS-Betriebssystemen von Apple, die eine Portierung von Dynamix zu iOS zum aktuellen Zeitpunkt (2012) unmöglich machen. Insbesondere die mangelnde Unterstützung von Hintergrund-Prozessen in iOS ist problematisch, da diese auf einen Satz von sehr eingeschränkten Anwendungen limitiert ist:

- Apps, die Audio im Hintergrund abspielen können (z.B. Music Player Apps)
- Apps, die den Nutzer über seinen aktuellen Aufenthaltsort informieren wollen (z.B. Navigations-Apps)
- Apps, die Voice over Internet Protocol (VoIP) unterstützen (z.B. Conference-Apps)
- Apps, die im Hintergrund neue Informationen laden (z.B. Nachrichten-Apps)
- Apps, die regelmäßig Updates von externen Geräten erhalten.

Zudem unterstützt iOS nur sehr rudimentäre IPC-Funktionen und bietet nicht die effizienten und sicheren IPC-Techniken von Android. In iOS können selbstdefinierte URL-Präfixe und Type-Handler verwendet werden, um Daten durch das Clipboard, über Anwendungsspezifische String-basierte Protokolle oder durch das private Dateisystem der App auszutauschen. Die Clipboard-Methode ist auf String-basierte Daten limitiert und bietet keinen Locking-Mechanismus. Daher sind Kollisionen und Datenverlust möglich. Spezifische Protokolle sind inhärent String-basiert und erfordern komplexe Parser sowie Fehler-Behandlungsmethoden von der empfangenden App. Private Dateisysteme erfordern eine aktive Beteiligung des Endnutzers für die Weitergabe der Nachrichten (Auswahl der Ziel-App). Dies macht Dynamix-Szenarien unmöglich. Zudem unterstützt iOS nicht die sichere Identifikation des aufrufenden Prozesses und macht daher die Entwicklung von Sicherheits-Mechanismen und die Einhaltung von Privacy unmöglich. Auch das Late-Binding von Komponenten wird von iOS nicht aufgrund der Apple Terms of Service nicht unterstützt. Apps, die Code-on-Demand anfordern und Late-Bindung durchführen wollen, werden im App Store nicht zugelassen.

1.3.4. Architektur

Aufgrund der oben genannten Aspekte wurde Google Android als die Basis für die Entwicklung des Dynamix-Frameworks ausgewählt. Dynamix ist als [Android Service](#) konzipiert, der im Hintergrund läuft und (potentiell mehreren Android-Anwendungen) Kontext-sensitive Dienste anbietet. Das Dynamix-Framework unterstützt das normale Android Service Lifecycle-Modell und läuft, solange genügend Ressourcen auf dem Gerät vorhanden sind (oder wird wenn möglich automatisch neu gestartet). Obwohl Dynamix im Wesentlichen unsichtbar im Hintergrund läuft, können Anwender mit Hilfe eines graphischen Nutzer-Interfaces verschiedene Konfigurationen festlegen: Context Firewall, Zugriffsrechte, Updates, Plug-in Settings, etc. Dynamix informiert über wichtige Zustandsänderungen mit Hilfe des Android Notification Systems.

Abbildung 21 verdeutlicht die Platzierung des Dynamix Service zwischen der lokalen Geräte-Hardware und den (potentiell vielen) Dynamix-Anwendungen, die in ihrer eigenen Laufzeitumgebung operieren. Dynamix unterstützt zwei prinzipielle Arten von Anwendungen: Native Apps und Web Apps. Native Apps sind übliche Android-Anwendungen die mit dem lokalen Dynamix Service über die [AIDL](#)-basierte API aus dem Dynamix App SDK kommunizieren. Diese Interfaces beinhalten eine AIDL Facade-API, das Anwendungen ermöglicht, Kontext-Support anzufordern und zu kontrollieren, sowie eine AIDL Event-API, mit deren Hilfe Anwendungen vom Framework über Nachrichten und Kontext-Events informiert werden. Web Apps laufen in nativen Webbrowsern, wie Google Chrome oder Firefox. Um auch mit Web

Apps kommunizieren zu können, stellt Dynamix zwei lokale REST-basierte APIs zur Verfügung und integriert einen spezifischen Web Server innerhalb des Dynamix Service. Web Apps kommunizieren mit Dynamix via Ajax unter Nutzung von vorgehaltenen JavaScript Modulen. Dadurch wird die Service-Anbindung vereinfacht, API Interaktionen ermöglicht, (De-)Serialisierung unterstützt, sowie Fehlerbehandlung und Event-Verarbeitung realisiert. Die konkrete Unterstützung von Web Apps wird weiter unten diskutiert (siehe 1.4.6).

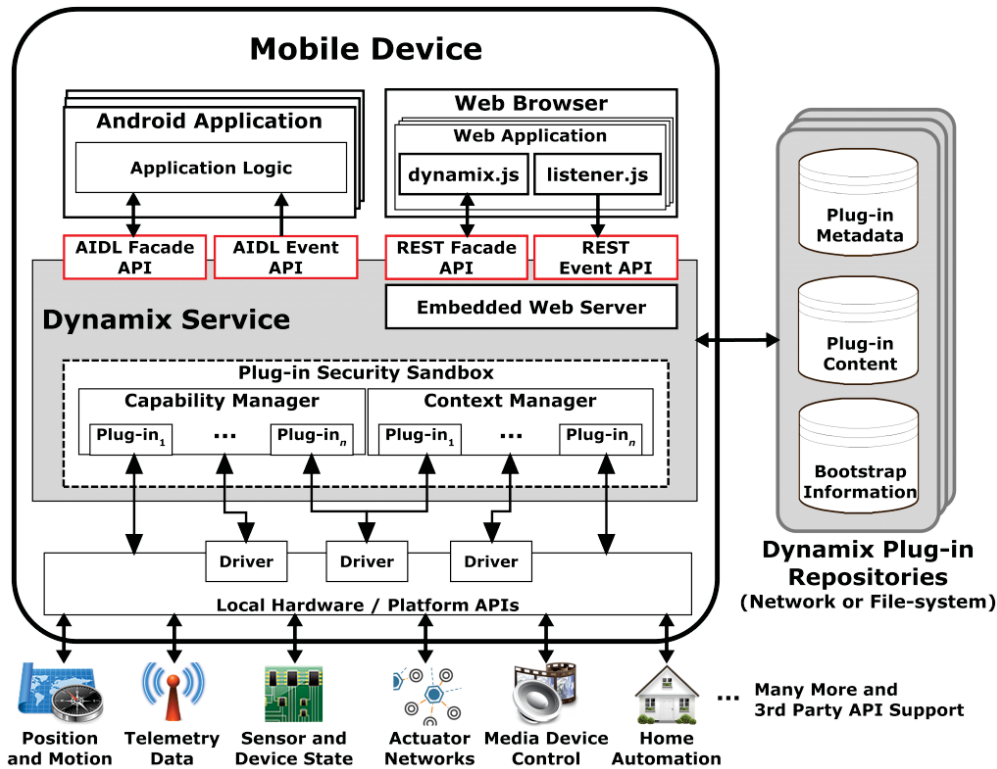


Abbildung 21: Überblick über das Dynamix Framework

Die Verarbeitung von Kontext (über Sensoren oder Aktoren) wird vom Dynamix Service im Auftrag der Apps mit einem entsprechend ausgewählten Satz von Kontext-Plug-ins realisiert. Diese werden als Standard OSGi-Bundles in Form von Java Archive (JAR)-Dateien mit zusätzlichen OSGi-Metadaten erzeugt, verpackt und versendet. Plug-ins werden auf einem konfigurierbarem Satz von öffentlichen oder privaten Repositories (im Netzwerk oder auf dem lokalen Dateisystem) vorgehalten und dort entsprechend gesucht, verpackt und zum Client geladen und im Dynamix OSGi-Container installiert. Plug-ins stellen den Anwendungen einfache, abstrakte APIs zur Verfügung, und entlasten die App-Entwickler von den typischen Domain-spezifischen Aufgaben, wie maschinelles Lernen, Sensor Fusion, Time-of-Flight-Berechnungen, Bildverarbeitung, etc. Zur internen Verwaltung der Plugin-Ins wird in Dynamix ein industriell standardisierter OSGi-Container verwendet, der dynamische Integration von mobilem Sourcecode und sicheren Zugriff auf lokale Ressourcen ermöglicht, ein vollständiges Plug-in Lifecycle-Management unterstützt (inkl. Thread-Prioritäten), Abhängigkeiten organisiert und Versionshaltung erzwingt. Dadurch wird der Download, die Installation (oder Upload) und die Integration von Kontext Plug-in Bundles während der Laufzeit möglich, ohne das Framework oder die Anwendung neu starten zu müssen. Die Plug-ins werden in das Dynamix-Laufzeitsystem (unter Berücksichtigung zusätzlicher Sicherheitsbeschränkungen) geladen und innerhalb eines streng kontrollierten Lebenszyklus ausgeführt. Private Repositories können von den Nutzern konfiguriert werden. Öffentliche Dynamix-Repositories verwenden eine Server-Architektur mit hoher Verfügbarkeit und Last-Management. Zur Re-

duzierung der Server-seitigen Komplexität hält der Dynamix Service Client-seitig Logik für die Suche von Plug-ins (Discovery Service), Kapazitätsanalyse und Filterung (aufgrund von lokalen Kapazitäten und Plug-in-Anforderungen) vor.

Der Dynamix-Service verwaltet automatisch den Energieverbrauch aller Kontext-Plug-ins durch die Weiterreichung der Nutzerpräferenzen während der Initialisierung, während der Laufzeit (z.B. Auswahl bestimmter Lastprofile) und während bestimmter Ereignisse des Android-Systems (z.B. Bildschirm an/aus). Zusätzlich werden Kontext-Events (optional) auch im eigenen Kontextspeicher (Context Cache) der Architektur zwischengespeichert. Dadurch können wertvolle Ressourcen auf dem Gerät eingespart werden, da bei wiederholter Anforderung von bestimmten Kontext-Events vermieden wird, den Kontext erneut mit Hilfe von (ressourcen-intensiven) Plug-ins abzufragen (z.B. Barcode-Scan durch Kamera).

1.3.5. Kontext-Events

Wenn Plug-ins mit der physikalischen Umgebung interagieren, senden sie eventuell Kontextinformationen als Events an die registrierten Apps. Die Weiterreichung der Events an die Apps erfolgt durch Dynamix aufgrund von Sicherheitseinstellungen in der Dynamix Context Firewall (siehe unten). Plug-ins kodieren die Events durch [Plain Old Java Objects \(POJOs\)](#), und unterstützen Apps in der Arbeit mit Objekt-basierter Kontextinformation. Eine potentiell aufwändige Bearbeitung mit sonst oft üblicher Repräsentation in Zeichenketten entfällt somit. Plug-ins können manuell installiert werden oder automatisch im Hintergrund aufgrund von Anforderungen durch die App in den Dynamix-Service geladen werden. Abbildung 22 zeigt den Reiter für den Dynamix Service (links) und den Reiter für Plug-ins (rechts) während der Installation.

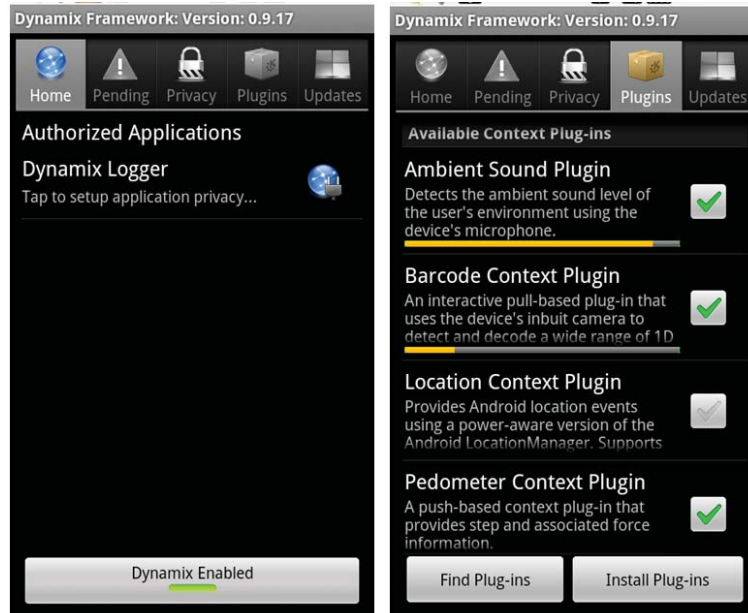


Abbildung 22: The Dynamix Service's Home and Plug-ins Tabs

Da Kontextinformation in bestimmten Kontext-Domänen oft sehr spezifisch definiert ist, stellt Dynamix das Interface *IContextInfo* zur Verfügung, mit dem Entwickler in die Lage versetzt werden, beliebig komplexe Kontextinformationen mittel POJOs zu erzeugen. Das Interface erweitert *android.os.Parcelable* und daher muß entsprechende Logik für die (De-)Serialisierung zur Verfügung gestellt werden, um die Kontext-Repräsentation über die Pro-

zess-Grenzen von Android hinweg zu kommunizieren. Die Implementierung von IContextInfo ist selbstbeschreibend und stellt sowohl den Kontext-Typen als auch vollständige Klassennamen zur Verfügung. Plug-in-Entwickler können ihre eigene Implementierung von IContextInfo verwenden oder die existierenden IContextInfo-Typen verwenden, die von der Dynamix-Entwicklergemeinschaft zur Verfügung gestellt werden.

Da Kontext-Events potentiell sensible Informationen beinhalten, stellt Dynamix eine integrierte Context Firewall zur Verfügung. Nutzer können Sicherheitsregeln festlegen, mit denen das Weiterreichen von Events von Plug-ins zur Anwendung festgelegt werden. Im Gegensatz zu der in Android üblichen Vorgehensweise, die Zugriffsrechte bei der Installation einer App einmalig zu genehmigen, können die Regeln der Context Firewall jederzeit angepasst werden. Dadurch sind Nutzer von Dynamix-Apps in der Lage, dynamisch zur Laufzeit die Sicherheitseinstellungen der App zu aktualisieren, bestimmte Apps vollständig zu blockieren oder sogar die Weiterleitung von Kontext-Events vollständig im System zu unterbinden. Die Regeln der Context Firewall werden für jede Anwendung in vorkonfigurierten Privacy Policies und/oder Custom Privacy Settings festgelegt. Voreingestellt ist eine vollständige Blockierung beim Zugriff auf Plug-ins und Kontextinformationen („Blocked“). Während der ersten Benutzung einer App oder einer Website mit Dynamix Web-App informiert Dynamix den Anwender durch entsprechende Android-Nachrichten (optional auch durch Geräusche oder Vibrationen), woraufhin dann spezifische Einstellungen für die jeweilige App durchgeführt werden können. Ein Klick auf das Android Notification Icon öffnet das Kontroll-Interface der Context Firewall und erlaubt die Konfiguration einer Regel. Abbildung 23 zeigt beispielhaft die Festlegung von Sicherheitsregeln für die Beispielanwendung „Dynamix Logger“.

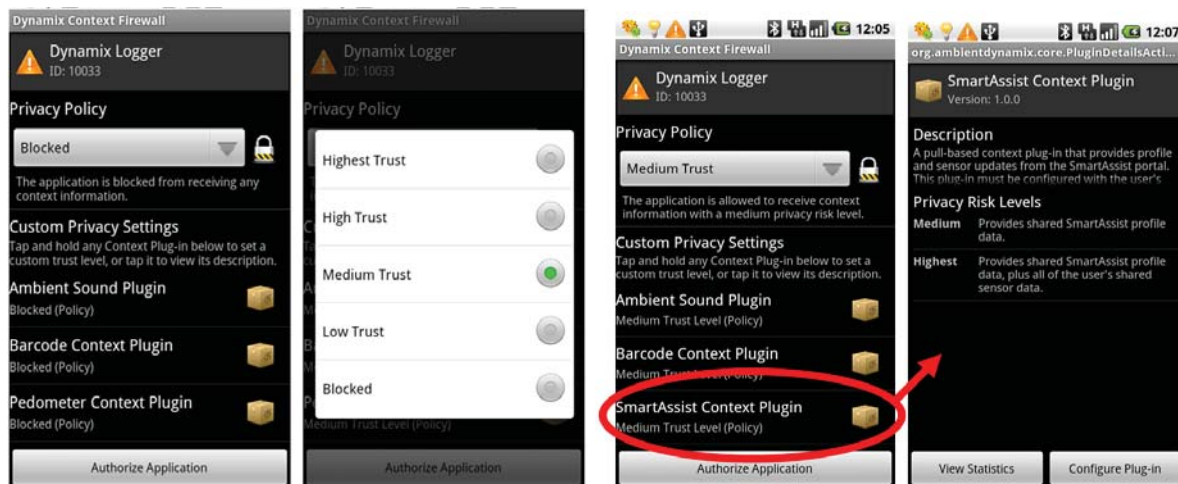


Abbildung 23: Festlegung einer Sicherheitsregel in der Dynamix Context Firewall

Eine Sicherheitsregel definiert das generelle Vertrauen an eine bestimmte Anwendung (Trust Level) von „Blocked“ bis „Highest Trust“ in aktuell fünf Stufen. Damit eine App Kontext-Informationen eines spezifischen Sicherheitslevels erhalten kann, muss ein entsprechender Trust Level gegeben sein (Privacy Policy) oder Anwender-spezifische Einstellungen vorgenommen werden (Custom Privacy Settings). Eine Privacy Police überträgt automatisch die entsprechenden Trust Level auf jedes installierte Plug-in, mit gleichem Level (z.B. erlaubt „Low Trust“ lediglich die Weiterleitung von Events mit niedriger Risikostufe von Plug-in zur App). Für eine detailliertere Steuerung können für bestimmte Plug-ins auch eigene Regeln (Custom Privacy Settings) festgelegt werden, mit denen die Einstellungen der Privacy Policy überschrieben werden. Eine Anwendung kann so bspw. den Trust Level „Medium Trust“ erhalten aber trotzdem vom Empfang der Events eines bestimmten Plug-ins ausgeschlossen

werden. Um den Nutzer in der Definition geeigneter Einstellungen zu unterstützen, stellen die Plug-in-Entwickler detaillierte textuelle Beschreibungen der entsprechenden Sicherheitsrisiken des Plug-ins auf den verschiedenen Ebenen zur Verfügung (siehe Abbildung 23 rechts).

Für den abgesicherten Zugriff der Plug-ins auf Systemressourcen, Nutzerdaten und Android-Services dient die Dynamix Plug-in Security Sandbox. Die Sandbox wurde mit Hilfe einer Kombination aus OSGi Sicherheitsfeatures, einem eigens konstruierten Java Security Manager und einer abgesicherten Version des Android-basierten Zugriffs auf das Betriebssystem und Sensor-Manager umgesetzt. Zusätzlich wird ein Modell für die Zugriffsrechte entwickelt, mit dem Anwender die jeweiligen Zugriffsrechte im System für jedes Plug-in festlegen oder modifizieren können. Alle Interaktionen zwischen dem Dynamix-Service und den Plug-ins sind Multi-Threaded (um eine Blockierung zu vermeiden) und unterliegen einem Monitor-Prozess zur kontrollierten Abfolge.

1.3.6. Web App Support

Dynamix Web Apps werden von native Webbrowsern (z.B. Google Chrome oder Firefox) gehostet. Zur Kommunikation mit Web Apps integriert Dynamix einen Webserver und stellt zwei lokale REST-basierte APIs zur Verfügung. Web Apps kommunizieren mit Dynamix via Ajax mit Hilfe zweier JavaScript-Dateien zur Vereinfachung der Dienstanbindung, API-Interaktion, (De-)Serialisierung, Fehlerbehandlung und Eventverarbeitung. Der eingebettete Web Server wartet auf Anfragen von Seiten der Web Apps und übersetzt diese in standardmäßige Methodenaufrufe des Dynamix-Service (und umgekehrt). Abbildung 24 zeigt einen Überblick über die Erweiterungen von Dynamix in Richtung Web.

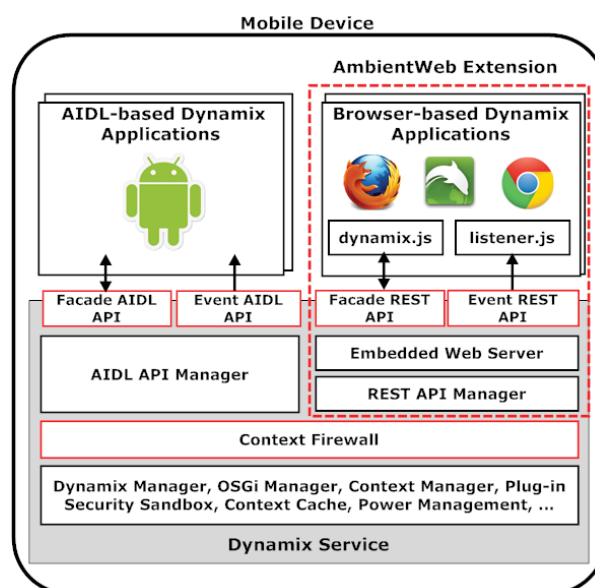


Abbildung 24: AmbientWeb - Dynamix Erweiterung für Web Apps

Der Dynamix Webserver erlaubt Anfragen der Web Apps über den localhost und verwendet einen angegliederten REST API Manager (REST Manager) für die Übersetzung der Anfragen in Dynamix Methodenaufrufe. Sowohl der Webserver als auch der REST Manager sind Multi-Threaded und mehrere Anfrager und Browser gleichzeitig bedienen zu können. Da AmbientWeb auf dem Dynamix Framework beruht, haben Web Apps vollständigen Zugriff auf den umfassenden Satz von Kontexttypen und Dynamix-Dienstleistungen, inkl. der dynamischen Erkennung und Installation von Plug-ins, adaptiver Kontext-Erfassung, Event-Caching, Power Management, etc.

AmbientWeb Apps interagieren mit dem lokalen Dynamix Service mit Hilfe von zwei JavaScript Dateien: *dynamix.js* und *listener.js*. Diese ermöglichen die Bindung an Dynamix, Interaktion über die REST API, Zustandsverwaltung, (De-)Serialisierung, Fehlerbehandlung und Kontext-Vorverarbeitung. Die Datei *dynamix.js* stellt eine vollständige Implementierung des Dynamix Façade API dar, inklusive der Möglichkeit, neue Listener einzurichten, sich für bestimmte Kontext-Typen zu registrieren, Kontextinformation anzufordern (z.B. Pulsrate oder Lokation) und ad-hoc Interaktionen mit lokalen oder entfernten Akteuren durchzuführen (z.B. Heimautomatisierung zu kontrollieren). Die Datei *listener.js* stellt eine vollständige Implementierung der Dynamix Event API zur Verfügung, inkl. der Methoden für Callbacks, Updates für Kontext-Subskriptionen, Zustandsinformationen des Frameworks, Statusinfos für die Installation von Plug-ins, etc. Um die Entwicklung von AmbientWeb Apps zu vereinfachen, unterstützt die Methoden in *listener.js* die Anreicherung von eingehenden Kontext-Events mit JavaScript-Eigenschaften. Dadurch können Web Apps auf die Kontextinformationen zugreifen, ohne String-Parsing betreiben zu müssen (obwohl auch diese unterstützt werden).

Um eine für den Nutzer transparente Interaktion zwischen AmbientWeb Apps und dem lokalen Dynamix-Service zu realisieren, wird eine Kombination aus synchronem und asynchronem Ajax verwendet. Im Prinzip stellt Ajax eine Sammlung von Client-seitigen Techniken dar, zusätzliche Teile einer Webseite im Hintergrund anzufordern, ohne die Seite neu laden zu müssen oder mit den Nutzerinteraktionen und Prozessen auf der Seite zu kollidieren. Synchrones Ajax wird für die unmittelbare Rückmeldung der Status-Nachrichten in Façade API-Methodenaufrufen verwendet (z.B. Success oder Fail), während asynchrones Ajax für die Rückgabe von Daten Operationen mit langer Laufzeit verwendet wird (z.B. Kontext-Erfassung via Event API).

AmbientWeb Apps können mittels standardisierter Webtechnologien (HTTP, CSS, JavaScript) entwickelt werden. Dazu lädt ein Web-Entwickler die obengenannten JavaScript Dateien von der Dynamix-Webseite und integriert dieses mit Hilfe des HTML `<script>` tags. Im Repository können geeignete Plug-ins identifiziert und deren ID-Code notiert werden. Falls Kontext von der Anwendung verwendet werden soll, werden Funktionen aus *dynamix.js* verwendet, um die Web App mit dem Dynamix Service zu verbinden und einen entsprechenden Listener für den Empfang asynchroner Callbacks einzurichten. Anschließend werden Funktionen aus *dynamix.js* für die Subskription von Kontext, die Kontrolle der Kontext-Erfassung und/oder die Interaktion mit lokalen oder entfernten Geräten verwendet. Façade API-Anforderungen werden automatisch von der Web App über das lokale Netzwerk zum Dynamix-Service weitergeleitet und dort entsprechend veranlasst. Die Ergebnisse in Form von Kontext-Events und Status Update-Informationen werden vom Dynamix Service über die Callback-Methoden in die *listener.js* Datei übertragen und dort auf einfache Weise mit der internen Logik der Web App verbunden (z.B. um die Farbgebung einer Webseite an die Beleuchtung der Umgebung anzupassen).

Da AmbientWeb Apps in nicht modifizierten Browsern betrieben werden, unterliegen sie den üblichen Regeln der [same-origin policy \(SOP\)](#) als typischerweise realisierte Sicherheitsrestriktion. SOP verhindert normalerweise den Zugriff von Web-Anwendungen auf Daten oder Methoden in Skripten außerhalb der eigenen Domäne (scheme, host, port). Daher ist eine Web App normalerweise nur in der Lage, Ajax-Anforderungen an die Webdomäne zu richten, aus der die Anwendung in den Browser geladen wurde. Um dies zu umgehen und Anfragen an den lokalen Dynamix-Service zu ermöglichen, wird der Dynamix REST Manager derart konfiguriert, dass er [Cross-Origin Resource Sharing \(CORS\)](#) unterstützt. CORS ist eine Spezifikation für den sicheren Cross-Site HTTP-Zugriff einer Web App bestimmten Ursprungs (z.B. <http://ambientdyanmix.org>) und einem Webserver in einer anderen Domäne (e.g.,

http://localhost). In Dynamix wird der eingebettete Webserver mit einer CORS-Policy konfiguriert, die die Interaktion zwischen AmbientWeb Apps und dem REST-Manager über den localhost erlaubt.

Der REST-Manager kann Web Apps auf sichere Weise durch die “origin” und “referrer“ URL im HTTP-Header identifizieren, die vom Browser generiert wird. Zur Vermeidung von http Header Spoofing durch schadhafte Apps, die einen Webbrowser nur simulieren, validiert Dynamix sämtliche Apps. Die Identifikation und Verifikation wird durch die Abfrage der Linux user ID realisiert, die Anwendungen unter Android eindeutig unterscheidbar macht. Dazu wird der Socket-Port der App mit der Liste der offenen Netzwerkverbindungen in den virtuellen Dateisystemen im Kernel von Android (/proc/net/tcp und /proc/net/tcp6 virtuelle Dateisysteme) verglichen. Sobald eine anrufende App identifiziert wurde, werden die zugehörigen X509-Zertifikate der App aus Android extrahiert und mit einer Liste der zugelassenen X509-Zertifikate verglichen, die in Dynamix vorgehalten wird. Im Falle einer Abweichung wird die Anfrage abgelehnt und ein unberechtigter Zugriff durch kritische Anwendungen verhindert. Dynamix enthält eine erweiterbare Liste von geprüften Zertifikaten der Browser, für die bekannt ist, dass sie HTTP Header-Spoofing nicht unterstützen.

Web Apps unterliegen in der Kommunikation mit Dynamix über AmbientWeb den gleichen Restriktionen wie konventionelle Dynamix Apps (durch die Einstellungen der Context Firewall). Standardmäßig werden auch AmbientWeb Apps zunächst vollständig geblockt. Während der ersten Interaktion mit einer neu verbundenen AmbientWeb App informiert Dynamix den Anwender mit Hilfe der Android-Notification (optional mit Geräuschen oder Vibrationen). Anschließend können analog der Vorgehensweise bei nativen Dynamix Apps die Einstellungen für Sicherheitsregeln vorgenommen werden. Diese können permanent für eine spezifische Web App oder nur für eine Browser-Sitzung vereinbart werden. Eine Änderung durch den Nutzer ist jederzeit ohne erneutes Laden der Webseite möglich.

1.3.7. Community

Wir verstehen das Dynamix-Framework als die technische Grundlage einer breiten Nutzer-Community für die einfache Nutzung der zahlreichen Expertisen in der Erfassung und Verarbeitung von Kontext, die den Entwicklern von mobilen Anwendungen im Normalfall nicht zur Verfügung steht. Abbildung 25 deutet die geplante Zusammenarbeit der verschiedenen Gruppen in Dynamix an. Infrastruktur-Entwickler sind verantwortlich für die Pflege und Weiterentwicklung des eigentlichen Frameworks und der zugehörigen Plug-in-Repositories. Plug-in-Entwickler umfassen die Domänen-Experten, die ihre Algorithmen und Daten in Form von Plug-ins und Kontext-Typen zur Verfügung stellen wollen. Anwendungs-Entwickler erzeugen Software für Android-Systeme auf Basis von Dynamix und den zur Verfügung gestellten Plug-ins.

Zur Unterstützung der Plug-in-Entwickler stellen wir das Dynamix Plug-in SDK zur Verfügung. Hier sind alle notwendigen Klassen, Dokumentationen und Beispiel-Codes zusammengefasst, um Drittanbietern einen leichten Einstieg in die Entwicklung unterschiedlichster Plug-ins für Dynamix zu ermöglichen. Da Kontextinformationen oftmals sehr spezifisch für eine bestimmte Domäne definiert wird, hält das SDK auch das Interface *IContextInfo* vor, mit dem beliebige Kontexttypen mit Hilfe von Plain Old Java Objects (POJOs) definiert werden können. *IContextInfo*-Implementierungen sind selbst-beschreibend und stellen neben der Definition eines Kontexttyps auch die Methoden zur Kodierung und vollständige Klassennamen für die Implementierung zur Verfügung. Plug-in-Entwickler können ihre eigenen Implementierungen von *IContextInfo* realisieren oder existierende *IContextInfo*-Typen der Dynamix-Community modifizieren und erweitern.

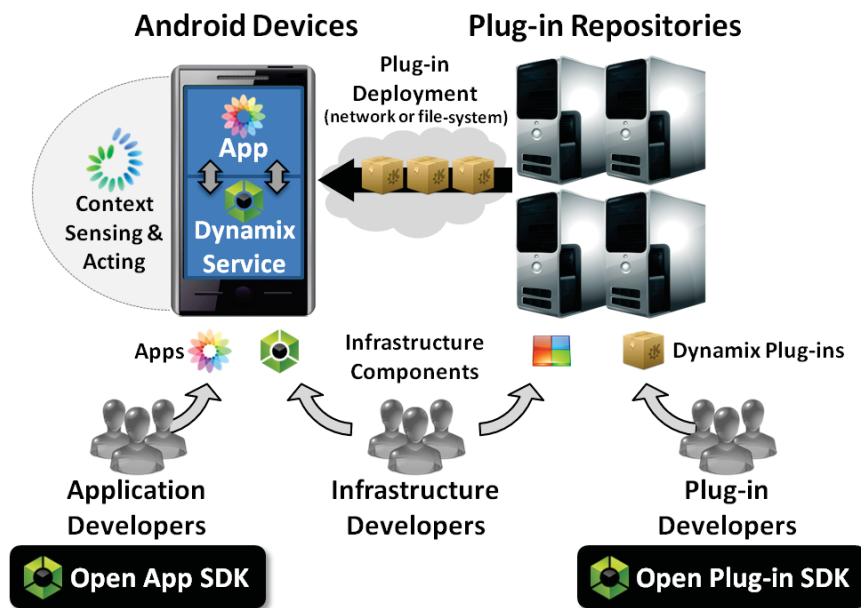


Abbildung 25: Dynamix Entwickler-Communities

Für die Anwendungs-Entwickler wird das offene App SDK zur Verfügung gestellt. Dieses enthält alle Basis-Klassen, Dokumentation und Beispielcode für die Entwicklung von Dynamix-Apps. Für die Entwicklung kann auf die ständig wachsende Zahl von Plug-ins auf der Dynamix-Webseite zurückgegriffen werden. Hier werden die entsprechenden Identifier, die Funktionalität, die inhärente Semantik, die unterstützen Kontext-Typen, Hardware-Voraussetzung, etc. beschrieben. Nach der Integration der im SDK enthaltenen JAR-Dateien in die Anwendung, können Entwickler die Dynamix Service Facade API verwenden, um Informationen über den Zustand des Frameworks zu erhalten, Listener zu installieren oder zu entfernen, nach geeigneten Plug-ins zu suchen, Events aus dem Cache anzufordern und Kontext-Events zu abonnieren. Eine *context subscription* ist eine Registrierung der Dynamix App, um Events eines bestimmten Kontext-Typs zu abonnieren oder den Kontext des entsprechenden Typs mit Aktoren zu modifizieren.

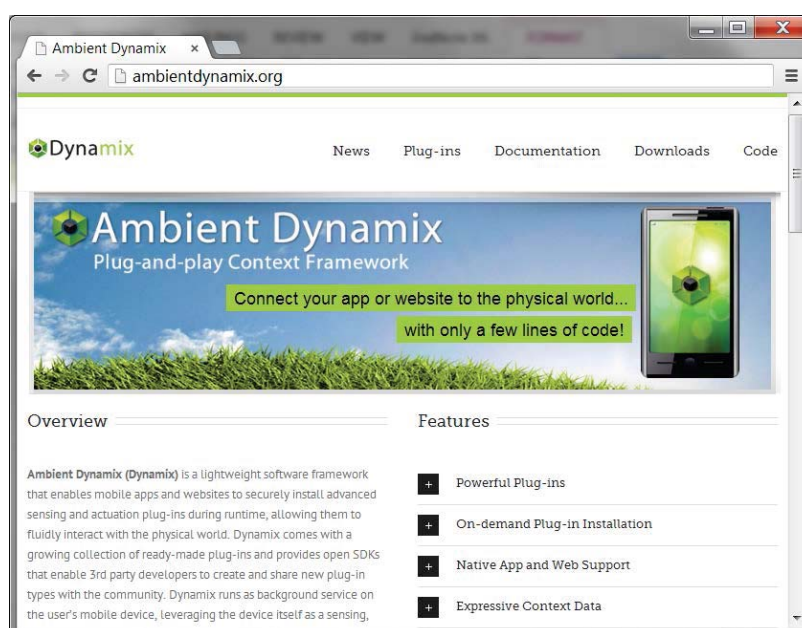


Abbildung 26: Dynamix Webseite

Während der Subskription wird der angeforderte Kontext-Typ mit den lokal installierten Plug-ins abgeglichen. Wenn der Dynamix-Service in der Lage ist, die Anfrage zu erfüllen (d.h. ist ein kompatibles Plug-in installiert), wird die Anmeldung registriert, die entsprechend Kontexterfassung oder Modifikation angestoßen und die App mit Hilfe der Event API informiert. In Situationen, wenn der Dynamix Service die Anfrage nicht lokal bearbeiten kann, wird in den konfigurierten Repositories nach geeigneten Plug-ins gesucht und kompatible Instanzen geladen. Die Installation erfolgt dann zur Laufzeit und die Subskription wird automatisch etabliert. Sobald die App Kontextinformationen erhält, ist die Reaktion darauf nur noch Bestandteil der entsprechenden Logik innerhalb der App.

Sowohl das App SDK als auch das Plug-in SDK sind auf der Dynamix Entwickler-Webseite (<http://ambientdynamix.org>) frei verfügbar. Hier sind zusätzliche Informationen zum Projekt, sowie alle Dokumentationen, Beispiel-Codefragmente und Plug-in-Beschreibungen vorgehalten. Ebenso erhalten Entwickler Informationen über das Einpflegen eigener Komponenten (z.B. mit Versionsverwaltung, Online-Foren, etc.)

1.3.8. Evaluierung

Die Dynamix-Architektur wurde vollständig implementiert. Das Framework ist kompatibel mit Android 2.1 (und höher) und wurde auf verschiedenen Geräten getestet, u.a. Motorola Milestone, Samsung Galaxy Tab, sowie Smartphone von Google (Nexus), LG und HTC (Hero). Während der Laufzeitverlängerung des SmartAssist-Projekts wurde Dynamix erweitert, um auch Android 4.x zu unterstützen. Ebenso wurden die App und Plug-in SDKs implementiert und die Webseite erstellt.

Für die Evaluierung des Frameworks wurden zahlreiche Beispiel-Plug-ins erstellt, darunter Vitalwert-Messung (Bluetooth Pulsrate), Erkennung von Radiosendern (Radio Beacon), Geoposition, Datenerfassung aus Sensornetzwerken, Orientierung, Umgebungslautstärke, Schrittzähler mit Schrittstärke, Near-Field-Communication (NFC), Luftqualität, Gewicht, BMI, Barcodes, u.a. Zum Test der Plug-ins wurden Test-Anwendungen implementiert.

Zur Evaluierung der Leistungsanforderung von Dynamix bzgl. CPU und Speicher während üblicher Operationen (wie Plug-in download oder Kontexterfassung) wurde Dynamix version 0.9.37 unter Android 4.0.2 auf einem Samsung Galaxy Nexus installiert (weitere Evaluierungen werden zurzeit durchgeführt). Um die Last während der Laufzeit zu messen, haben wir ein eigenes Monitoring-Framework entwickelt, mit dem in automatisierten Tests kontinuierlich Aufzeichnungen für Speicher und CPU für alle Dynamix-Threads erfolgen können. Das Monitoring-Framework läuft in einem eigenen Prozess, der den Dynamix-Prozess während der Messung isoliert. Zusätzlich wurde ein Lastgenerator entwickelt, der eine konfigurierbare Zahl von Kontext-Events mit spezifischen Payload-Größen erzeugen kann.

Das Monitoring Framework wurde derart konfiguriert, daß während Testphasen von 30 Sekunden Länge eine konstante Folge von Kontext-Anfragen mit fester Größe erzeugt wird. CPU-Auslastung und Speicherverbrauch wurden mit einer Rate von 2 Hz erfasst. Zu Beginn der Tests wurde eine Verzögerung von 10 Sekunden eingehalten, um Dynamix Gelegenheit zu geben, Garbage Collection durchzuführen und einen vergleichbaren Ausgangszustand zu erreichen. In den Testläufen wurde dann bei gleicher Payload-Größe die Frequenz der Anfragen solange erhöht (von 1 Hz bis max. 600 Hz), bis instabile Verhältnisse im Speicher auftraten (d.h. ständig wachsende Heap-Größen). Die Tests wurden dann mit steigender Last wiederholt (von 32 Byte bis 512 KByte). Mit diesen Größenordnungen sollte ein relativ großer Prozentsatz möglicher realistischer Situationen von Anwendungen und Kontext-Typen abgedeckt sein.

Für eine erste Analyse wurde das Monitoring für eine typische Dynamix Laufzeit-Phase durchgeführt. Abbildung 27 zeigt die Ergebnisse.

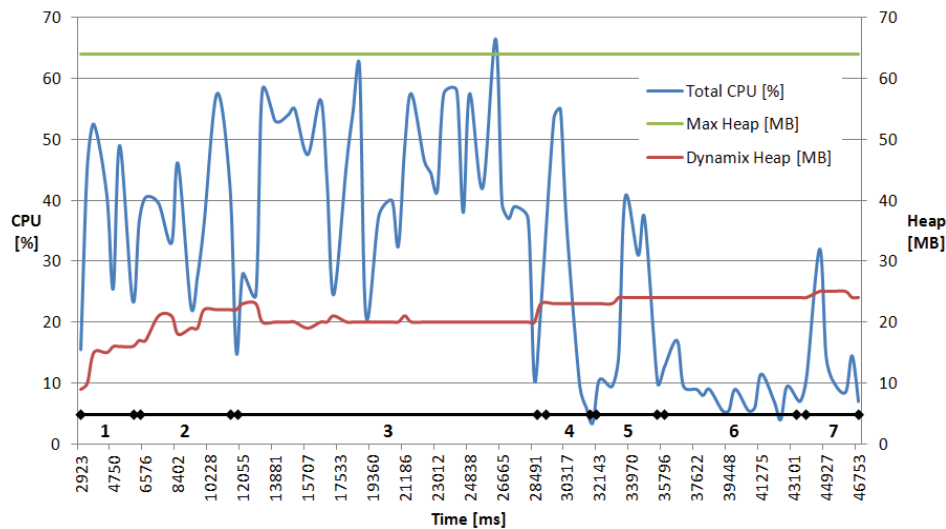


Abbildung 27: Auswertung typischer Dynamix Laufzeit-Phasen

Wie im Diagramm ersichtlich, benötigt Dynamix in der Boot-Phase (1) ca. 3 Sekunden und erzeugt dabei eine durchschnittliche CPU-Last von ca. 35% (wobei Speicher von 9MB bis 17MB ansteigt). Während der Auswahl eines Plug-in (2), sind einige kurze CPU-Lastspitzen sichtbar und der Speicherverbrauch steigt an (Laden der entsprechenden Android Activity). Während der Installation des Plug-in (3), erzeugt Dynamix eine typische CPU-Last von ca. 39% und der Speicherverbrauch stabilisiert sich (aufgrund von Garbage Collection). Die Installation von 9 Plug-ins benötigt ca. 15 Sekunden über eine WIFI-Verbindung (mit insgesamt 1,74 Mbyte Download). Während der Auswahl des Homescreens in der Dynamix-GUI (4) entsteht eine kurze Lastspitze und der Speicherbedarf wird moderat höher. Das Erfassen von Kontext startet in (5) während ca. 3,5 Sekunden und erzeugt ebenfalls eine kurze Lastspitze (ca. 25%) und eine leichten Steigerung des Speicherbedarfs auf nun 24 Mbyte (für die Instanziierung der Plug-ins mittels OSGi). Während des laufenden Tests mit Kontext-Erfassung in den 9 geladenen Plug-ins (6) bleibt die CPU-Auslastung stabil bei ca. 8% mit einem Speicherbedarf von ca. 24 Mbyte. Beim Stopp der Kontext-Erfassung (7) entsteht erneut eine kurze Lastspitze (ca. 20%). Danach reduziert sich der Heap kontinuierlich, da die Plug-ins wieder entfernt werden.

In einem Folgetest mit dem oben beschriebenen Testaufbau wurden 438 Kontext-Anfragen mit dem Lastgenerator erzeugt. Jeder Test dauerte 30 Sekunden, in denen die durchschnittliche CPU-Auslastung und der durchschnittliche Speicherverbrauch bestimmt wurden. Die Sequenz aller Payload-Größen und Anfrageraten dauerte insgesamt ca. 5 Stunden. Wie in Abbildung 28 ersichtlich, hat die von Dynamix erzeugte CPU-Auslastung einen relativ linearen Charakter für typische Eventgrößen (32 Bytes bis 16 Kbyte). Innerhalb dieses Bereichs erzeugten Abfrageraten von 1 Hz bis 60 Hz eine CPU-Auslastung unter 20%.

Da die grundlegenden Android-Dienste in der Regel zwischen 1% und 5% der CPU beanspruchen, benötigt Dynamix folglich ca. 15-19 %. Abfrageraten unter 20 Hz reduzierten die Gesamtbelastung der CPU auf unter 10% (mit einem Dynamix-Anteil von 5-9 %). Die Abfragerate konnte bis ca. 500-600 Hz gesteigert werden (für typische Payload-Größen). Allerdings erzeugen solche hohen Raten auf eine sehr hohe CPU-Last (> 80%) und sollten nur in sehr kurzen Abschnitten verwendet werden. Es ist allerdings zu beachten, dass in diesem Text nur die kontinuierliche Abfrage evaluiert wurde. Szenarien mit wechselnden Abfrageraten wurden nicht durchgeführt, da hier die CPU-Auslastung deutlich niedriger ist. Die Tests zeigen somit schon ein Worst-Case-Szenario.

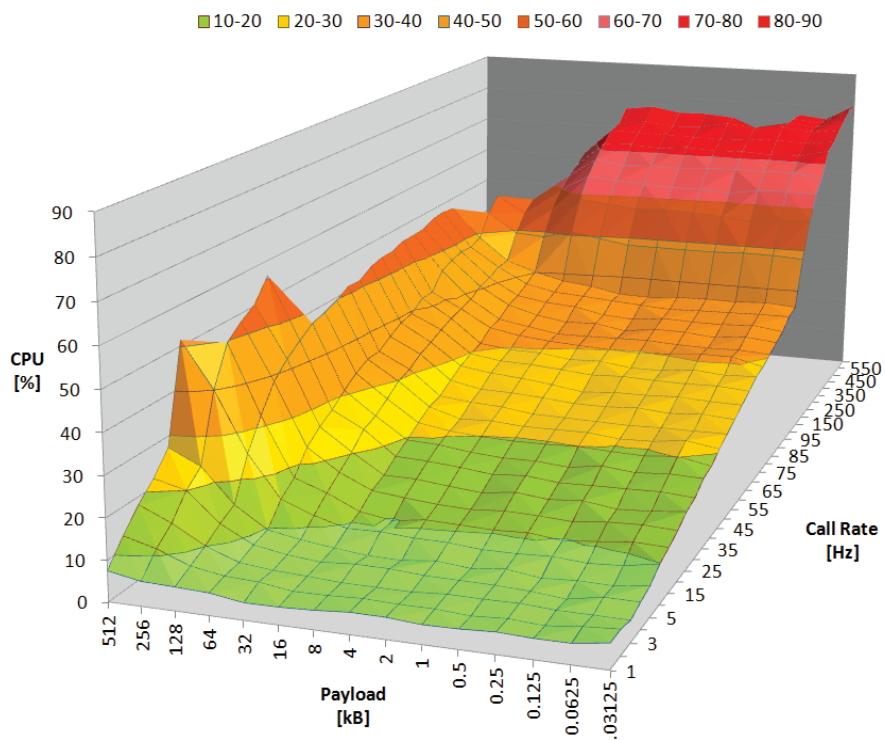


Abbildung 28: CPU-Auslastung für verschiedene Payloads und Abfrageraten

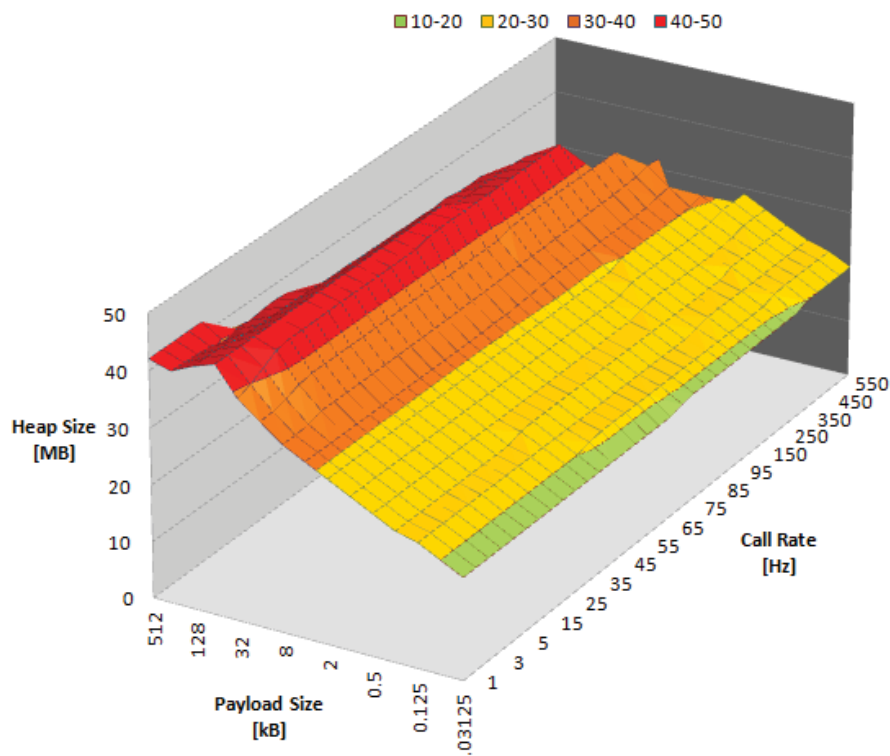


Abbildung 29: Speicherverbrauch für verschiedene Payload-Größen und Anfrageraten

1.3.9. Plug-ins

Abbildung 30 werden die im Projekt realisierten Dynamix Plug-ins vorgestellt.

<i>Plug-in</i>	<i>Description</i>
SmartAssist	Dieses Plug-in verwendet das OpenSocial-Protokoll und Open Authorization (OAuth), um Profildaten und Sensordaten aus dem sozialen Netzwerk von SmartAssist abzufragen. Die Abfrage und die Fehlerbehandlung werden vom Plug-in automatisch durchgeführt und ermöglichen die einfache Integration von SmartAssist-Daten in Apps.
Heart Rate	Dieses Plug-in stellt Pulsrate, Schrittzähler und Geschwindigkeit des Nutzers aus einem Zephyr Hx Fitneß-Gerät zur Verfügung, das über Bluetooth mit dem Android-Gerät verbunden ist. Das Plug-in verbindet sich automatisch mit dem Zephyr-Gerät und übernimmt alle Aspekte der biotelemetrischen Kommunikation für die Anwendung.
Sound Pressure Level	Dieses Plug-in erfasst die Schalldruckleistung der Umgebung mit Hilfe des im Gerät verbauten Mikrophons.
Battery Level	Dieses Plug-in stellt Apps den Ladezustand der Batterie zur Verfügung. Das Plug-in erzeugt automatisch Events für bestimmte Ladezustände und kann jederzeit abgefragt werden.
ArtNet Light Control	Dieses Plug-in unterstützt das Detektieren und Kontrollieren von Lampen, die das Artnet Protokoll (DMX over IP) unterstützen.
Light Sensor	Dieses Plug-in detektiert die Leistung der Umgebungsbeleuchtung. Das Plug-in erzeugt automatische Events bei Variation der normalisierten Lux-Leistung um mehr als 5%. Apps können aber auch jederzeit aktiv abfragen.
NFC	Dieses Plug-in erlaubt die Abfrage und Decodierung einer breiten Palette von Near Field Communications (NFC) tags. Das Plug-in setzt entsprechende Hardware-Unterstützung auf dem Gerät voraus.
Sleep State	Dieses Plug-in erlaubt die Abfrage von Information über den Schlafverlauf von einem via Bluetooth verbundenen Zeo Mobile Sleep Management System . Das Plug-in liefert die Schlafdauer, REM-Phasen und qualitative Angaben zum Schlaf, sowie Hardware-Informationen vom Zeo-Kopfband.
Acoustic Fingerprint	Dieses Plug-in identifiziert automatisch Musik in der Umgebung durch das eingebaute Mikrophon und Acoustic Fingerprint-Algorithmen. Der Kontext-Event enthält Titel, Interpret, ein Konfidenz-Intervall und weitere Daten.
Speech-to-Text	Dieses Plug-in wandelt mit Hilfe des eingebauten Mikrophons gesprochene Sprache in eine Auswahl von Texten mit Konfidenzwerten.
Barcode Scanner	Dieses Plug-in unterstützt die Erfassung und Decodierung einer breiten Palette von 1D/2D Barcodes mit Hilfe der eingebauten Kamera. Das Plug-in organisiert Kamera-Management, Bildverarbeitung und Barcode-Extraktion. Events enthalten das Barcode Format und den Barcode Wert. Zusätzliche Software ist nicht notwendig.
Health and Fitness	Dieses Plug-in erfasst das Gewicht, Fettanteil, Puls und Größe einer Nutzers mit Hilfe des Withings API der Withings – Wi-Fi Körperwaage und/oder des Withings Blutdruck-Meßgerätes (link).
Air Quality	Dieses Plug-in erfasst die lokale Luftqualität von einem Server des Umweltbundesamts. Zurzeit umfassen die Werte Ozon, Feinstaub,

Pedometer	CO, SO2 und NO2, wenn die nächste Station näher als 60km ist. Dieses Plug-in verwendet die integrierten Beschleunigungssensoren für die Detektion von Schritten und Schrittkraft.
Wireless Beacon	Dieses Plug-in scannt die Umgebung nach drahtlosen Meldebarken, wie z.B. WLAN-Hotspots ab und liefert detaillierte Informationen inkl. ID und Signalstärke. Die Scanrate kann mit Hilfe des Dynamix Power Profiles adjustiert werden.
Enhanced Location	Dieses Plug-in erweitert den Android Location Manager zur präzisen Erfassung des aktuellen Standorts. Dazu wird automatisch die jeweils genaueste Hardware auf dem Geräte (z.B. GPS oder WLAN-Triangulation) ausgewählt und die Abtastrate an das Dynamix Power Profile angepasst.
Load Generator	Dieses Plug-in kann als Testgenerator für Kontext-Anfragen verwendet werden, um Lastanalysen von Dynamix auf bestimmten Geräten durchzuführen. Das Plug-in wurde auf Zusammenarbeit mit dem Dynamix Performance Monitoring Framework ausgelegt, mit dem Dynamix Entwickler ihre Implementierung evaluieren können.
Ambient Media	Dieses Plug-in scannt die Umgebung nach Mediengeräten, die das AirPlay oder UpnP-Protokoll unterstützen (z.B. Smart TVs, Drahtlose Projektoren oder Multimedia-Laptops). Nach der Verbindung mit einem Gerät können Apps die volle Kontrolle übernehmen.

Abbildung 30: Kontext Plug-ins

1.3.10. Beispiel-Anwendungen und Dienstleistungen

Im Rahmen des Projekts wurden am Institut für Telematik unter Leitung von Prof. Dr.-Ing. Andreas Schrader eine Reihe von studentischen Arbeiten durchgeführt, die spezielle Aspekte des Themas vertiefen oder Beispiel-Dienste unter Dynamix oder dem Serviceportal realisieren.

- Augmented Reality in Ambient Assisted Living [38]

Frau Ahrens analysiert in Ihrer Arbeit das Potential von Augmented Reality (AR) beim Einsatz in Ambient Assisted Living (AAL)-Szenarien mit speziellem Fokus auf Demenz. Dazu werden anhand einer Anforderungsanalyse mögliche AR-Anwendungen aufgeführt und die technischen Randbedingungen strukturell erörtert. Eine mögliche Anwendung (Bedienungsanleitung für Haushaltsgeräte) wird prototypisch am Beispiel der Bedienung einer Kaffeemaschine implementiert und beschrieben. Zusätzlich wird die Möglichkeit der Verwendung von speziellen Markov-Prozessen (POMDP) für künftige Erweiterungen untersucht. Das System war ursprünglich für die Dynamix-Architektur vorgesehen, wurde aber aufgrund der technischen Komplexität zunächst auf einem Desktop-System realisiert.



Abbildung 31: AR-Unterstützung für Kaffeemaschine und zugehöriges Zustandsdiagramm

- Medication Support System [37]

Herr Kaluza beschreibt in seiner Ausarbeitung die Konzeption, Implementierung und Evaluation eines Systems zur Optimierung von Medikamenteneinnahmen. Dazu wird eine Architektur aus mobiler Nutzerkomponente und Server-basiertem Backend vorgestellt. Dazu wird eine umfassende Analyse des Problemfeld und deren potentiellen Benutzer, sowie verwendbarer Technologien durchgeführt. Die Arbeit führt ausführlich in die Problematik der Arzneimittel-Verwendung, insbesondere im Hinblick auf die Zielgruppe älterer Menschen ein (Wechselwirkungen, Compliance/Adhärenz, etc.) Es wird dann ein prototypisches System für Mobilgeräte (Java/Android) implementiert. Problematisch erwies sich im Laufe des Projektes das Fehlen einer kostenlosen Arzneimitteldatenbank mit offenen Schnittstellen. Dies wurde durch eine eigene Datenbank gelöst. Die Anwendung verwendet das Dynamix Barcode-Plugin, um Arzneimittelpackungen zu scannen und den Beipackzettel anzuzeigen, bildet einen Arzneimittelschrank ab, informiert über Nebenwirkungen, generiert Erinnerungen zur Einnahme, etc.

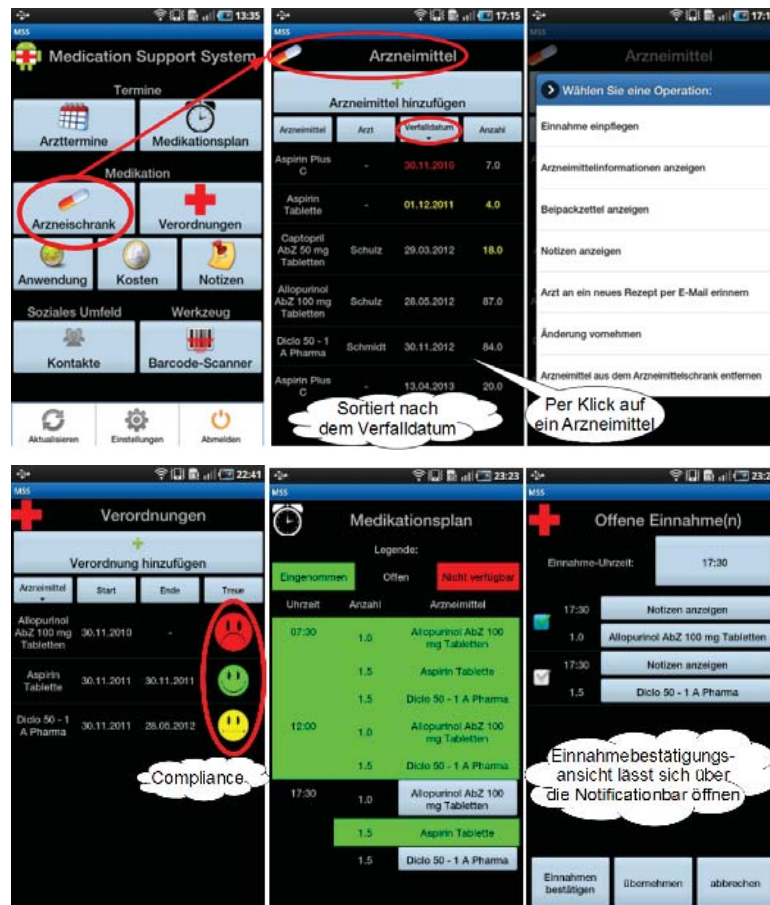


Abbildung 32: Einige Dialogbeispiele der mobilen Arzneimittel-Anwendung.

- Mobile Device Shake Compensation [36]

Zittern von menschlichen Gliedmaßen ist häufig ein Symptom vieler verschiedener Krankheiten. In Rehabilitationsprozessen oder im fortgeschrittenen Alter lassen sich bei der Kontrolle von Händen seitens der betroffenen Personen Schwierigkeiten beobachten. Dennoch bestehen das Bedürfnis und die Notwendigkeit mobile Geräte, wie Smartphones, Blutdruckmessgeräte oder ähnliche Geräte, zu handhaben, um z.B. Körperfunktion zu überwachen oder mit der Gesellschaft zu kommunizieren. Herr Sieb untersucht in seiner Arbeit die Möglichkeit, den optischen Eindruck von unfreiwillig bewegten Bildschirminhalten auf Mobilgeräten durch eine entsprechende Gegenbewegung der Pixel zu kompensieren.

sieren. Auf diese Weise soll die Lesbarkeit von Anzeigen für Nutzer mit eingeschränkter Feinmotorik (z.B. Parkinson) optimiert werden. Zu diesem Zweck analysiert Herr Sieb die theoretischen und technischen Randbedingungen und implementiert einen Algorithmus unter Verwendung der Bewegungssensoren eines Smartphones. Die Arbeit analysiert die wesentlichen Aspekte von Bewegungskompensation durch eine Diskussion der optischen Randbedingungen, verursachenden Krankheiten und möglichen Anwendungsgebieten. Ebenso wird eine Einordnung und Bewertung existierender mechanischer, optischer und elektronischer Verfahren aus dem Bereich der Bildstabilisierung in Kamerasystemen durchgeführt. Im zweiten Teil der Arbeit wird eine prototypische Realisierung mit Hilfe von Bewegungssensoren und entsprechender Kalibrierungssoftware durchgeführt und die Ergebnisse mit Hilfe einer High-Speed-Kamera evaluiert.

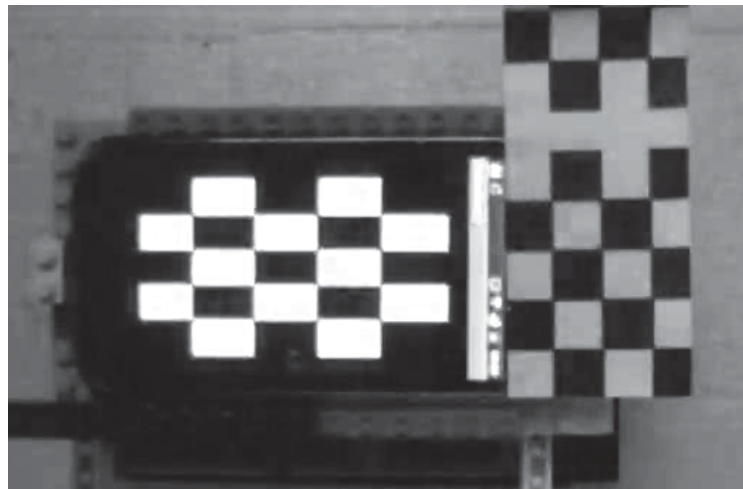


Abbildung 33: Evaluation der Bewegungskompensation durch Schachbrettmuster

- Long term tremor assessment using mobile devices [35]

Herr Awada beschreibt in seiner Masterarbeit die Entwicklung einer Software zur Erkennung eines Tremors mit Hilfe von handelsüblichen Smartphones. Wesentliches Ziel der Arbeit ist die Vermeidung von zusätzlich notwendigen Hardware-Komponenten und die Möglichkeit, Langzeit-Messungen im Hintergrund einer normalen Telefon-Nutzung zu realisieren. Die Erkennung erfolgt ausschließlich mit Hilfe der eingebauten Sensoren für Beschleunigung (Accelerometer) und Richtung (Gyroscope). Hierfür wird ein selbst-kalibrierender Algorithmus auf Basis von Quaternionen entwickelt, mit dessen Hilfe die Ausrichtung und Orientierung und damit die Amplitude und Frequenz von Eckpunkten des Geräts bestimmt werden kann. Dies kann zur direkten Bestimmung der Intensität von Tremor in der Hand eines Patienten verwendet werden. Der Algorithmus wird mit Hilfe eines Testaufbaus mit einer Highspeed-Kamera einer quantitativen Evaluation unterzogen und in einer Begutachtung durch Neurologen in einer qualitativen Diskussion auf mögliche Anwendungsfelder analysiert. Grundsätzlich sind die Ergebnisse vielversprechend und können durch die Möglichkeit, das Gerät preisgünstig und unauffällig im alltäglichen Umfeld (z.B. während eines Telefonats) einzusetzen, als Frühindikator verwendet werden, um bspw. die medikamentöse Einstellung bei Depression zu überwachen und Nebenwirkungen zu minimieren.

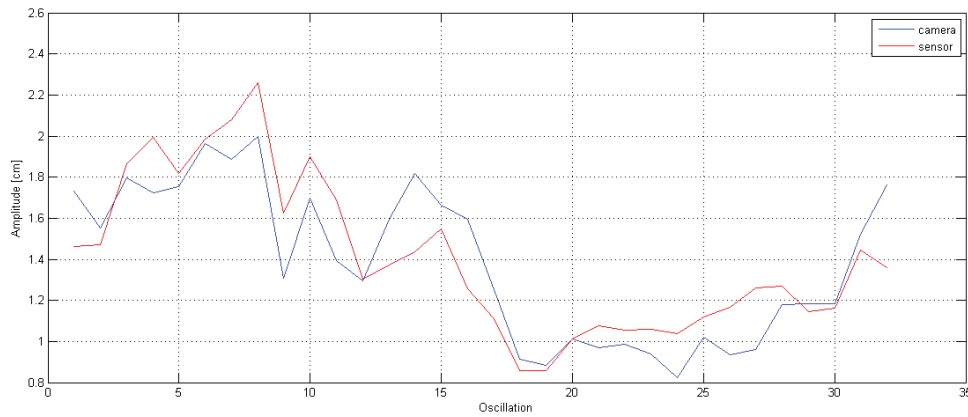


Abbildung 34: Vergleich von realer (Kamera – blau) und sensorisch (rot) erfasster Tremor-Amplitude

- Context-aware Device and Access Rights Management for Smart Homes in Ambient Assisted Living [34]

Eine häufige und folgenreiche Erkrankung des Alters ist die Demenz. In vielen Fällen ist eine autonome Lebensführung in den eigenen vier Wänden nur noch bedingt möglich. Insbesondere elektrische Geräte stellen eine Gefahrenquelle dar. Herr Stück untersucht in seiner Masterarbeit technische Möglichkeiten zur Realisierung einer flexiblen Zugangskontrolle für Haushaltsgeräte. Das System basiert auf dem Ansatz, Smart-Home-Technologie (EZControl) für die Steuerung der Geräte zu verwenden und die Funktionalität durch einen Betreuer vor Ort per Smartphone und RFID-Zugangskontrolle freischalten zu können. Das System integriert die persönlichen Profildaten aus dem SmartAssist-Server und verwendet für die sensorische Erkennung von Geräten und Personen das RFIF-Plugin des Dynamix-Frameworks. Beim Scannen der Wohn-ID, kann der Betreuer festlegen, welche Geräte anschließend auf dem Gerät des Nutzers freigeschaltet werden.

Ein Video zum Projekt steht auf der Webseite www.ambient.uni-luebeck.de zur Verfügung.



Abbildung 35: GUI des Betreuers, Scannen des RFID-Tags in der Wohnung, GUI des Patienten (von links nach rechts).

- Ambient Health Feedback System für Android [33]

Die modernen Arbeits- und Lebenswelten begünstigen vielfach eine ungesunde Lebensweise mit falscher Ernährung, zu wenig Bewegung, etc. Verbesserungen sind durch Umstellung von Ernährung, körperlicher Aktivität, Schlafverhalten, etc. relativ leicht zu erreichen, sofern die Motivation mitspielt. Leider geht die Konsequenz im Durchhalten entsprechender Vorhaben im Alltagsstress schnell unter.

Hier setzt Herr Schrick an und beschreibt in seiner Arbeit die Konzeption und prototypische Implementierung eines Ambient Health Feedback-Systems für Android-basierte Smartphones. Ein solches Gerät bietet sich an, da es meist ständig am Körper getragen wird und der Bildschirm häufig betrachtet wird. Herr Schrick untersucht die Möglichkeiten, unauffällige Grafiken auf dem Hauptschirm zu präsentieren, die gleichzeitig die Privatsphäre schützen. Er verwendet das Paradigma der Ambient-Awareness (Anzahl Blätter am Baum zur Repräsentation des Gesundheitszustands) und realisiert diese in Form von Widgets, LiveWallpapers und als App. Der Gesundheitszustand wird durch zahlreiche Dynamix-Plugins erfasst (REM, Gewicht, Schrittzähler).

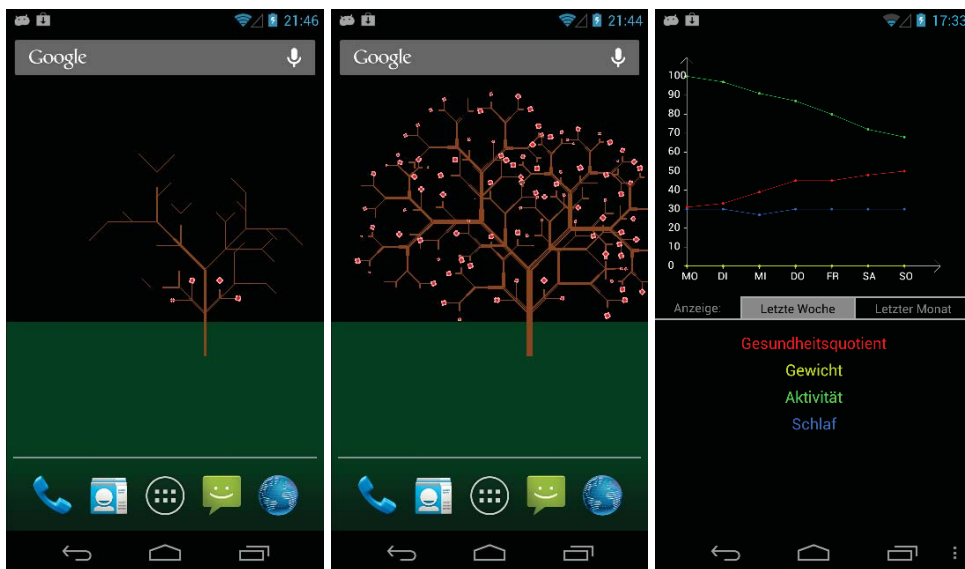


Abbildung 36: Live-Wallpaper bei unterschiedlichen Gesundheitszuständen und detaillierter Angabe der Historie im Wochenverlauf

- Aktivitätserkennung für Ambient Health Monitoring [32]

Die Integration von Sensoren in aktuelle Smartphone-Generationen bietet Entwicklern ein breites Spektrum von Anwendungsmöglichkeiten. Die Nutzbarkeit für Ambient Health Monitoring wird beispielsweise durch die Erfassung von Vitalparametern mithilfe dieser Sensoren sowie Einbindung externer Sensordaten aus dem SmartAssist-Projekt ermöglicht. Hierfür wird aufbauend auf einer Analyse von Benutzern, organisatorischer Strukturen und möglicher Nutzungskontexte eine Android-basierte Applikation für einen Personal Health Record unter Einbeziehung des Gamification-Konzeptes prototypisch innerhalb eines nutzerzentrierten Feature-Driven-Development-Prozesses umgesetzt. Parallel zur Entwicklung der Applikation wird anhand von Trainingsdaten, einer Merkmalsextraktion sowie Offline-Training und Analyse geeigneter Klassifikatoren eine Aktivitätserkennung in Form eines Ambient Dynamix-Plug-Ins realisiert und ergänzend in die Applikation integriert. Eine summative Evaluation der Aktivitätserkennung sowie der Anwendung soll Vor- und Nachteile der umgesetzten Konzepte überprüfen und Anhalt für eine Weiterentwicklung des Systems geben.

Das Projekt wurde auf dem 1st International Workshop on Lifelogging for Pervasive Health at the 7th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) in Venedig vorgestellt [4].



Abbildung 37: Screenshots der des Personal Health Record: Parameter, Ziele, Aktivität, Erfolge (von links nach rechts)

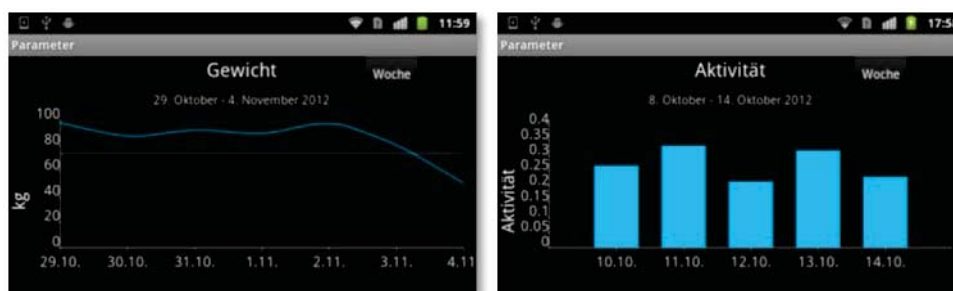


Abbildung 38: PHR: Visualisierung des historischen Verlaufs von kontinuierlichen (links) und diskreten (rechts) Werten.

- BikeWars: Entwicklung einer AAL-Anwendung für die Smart-Assist Plattform [31]

In BikeWars! treten die Mitspieler in einem Fahrradrennen über das Internet miteinander an, indem Sie ihr Fahrrad oder ihren Heimtrainer mittels eines Android-Tablets verbinden. Durch die Kombination aus Spiel und Wettkampf mit Freunden und Familie (sogenannte Social Casual Games) sollen die Benutzer zu sportlicher Betätigung animiert werden. BikeWars wurde von Martin Bomba im Rahmen seiner Bachelor Arbeit weiterentwickelt und in die BMBF ZukunftsWerkStadt Lübeck integriert. Öffentliche Vorführungen sind in Vorbereitung.

Nach Angaben des Statistischen Bundesamtes wird sich der Anteil an pflegebedürftigen Menschen in Deutschland in den kommenden 20 Jahren um 58% erhöhen. Dieser durch die Überalterung der Gesellschaft hervorgerufene Effekt stellt eine organisatorische und vor allem finanzielle Herausforderung für das Gesundheitssystem und die Altenpflege dar. Sport gilt hierbei nach wie vor als eine der wesentlichen vorbeugenden Maßnahmen, um auch im Alter noch gesund und aktiv bleiben zu können. Um mehr Menschen die Möglichkeit und die Motivation zu geben, auch im hohen Alter oder trotz gesundheitlicher Einschränkungen sportlich aktiv zu bleiben, wird am Institut für Telematik der Universität zu Lübeck das Spiel BikeWars! entwickelt. BikeWars! bietet die Möglichkeit, mittels eines Heimtrainers in der eigenen Wohnung oder auch draußen in der freien Natur mit dem Fahrrad an einem gemeinsamen Fahrrad-Rennen teilzunehmen. Hierbei wird der Schwierigkeitsgrad des Rennens an die persönlichen Parameter und Umweltdaten aller Spieler angepasst. Diese Technik ermöglicht es, die individuellen Eigenschaften aller Mitspieler zu berücksichtigen, um hierdurch eine Überanstrengung zu verhindern, ein

optimales Trainingsergebnis zu erzielen und das Rennen trotzdem zu einer echten Herausforderung zu machen.

So kann der individuelle Schwierigkeitsgrad an das Alter jedes einzelnen Spielers angepasst werden, ohne dabei die körperliche Leistungsfähigkeit zu vernachlässigen, welche während des Rennens über den Puls gemessen wird. Gleichzeitig kann die virtuelle Geschwindigkeit des Spielers erhöht oder gesenkt werden, um eine motivierende Wettkampfsituation zu schaffen, bei welcher die Spieler auf den hinteren Plätzen einfacher auf den Erstplatzierten aufschließen können.

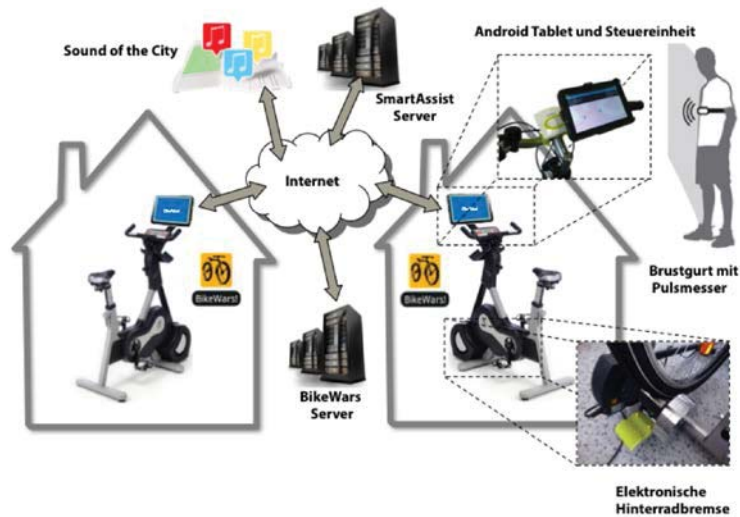


Abbildung 39: Übersicht über die BikeWars Architektur

Durch die Verwendung der Daten aus [SmartAssist](#) ist es möglich, den aktuellen Schwierigkeitsgrad des Rennens und die relative Geschwindigkeit der Teilnehmenden Spieler dynamisch an den Rennverlauf und die Mitspieler anzupassen. Hierfür dienen Soziale Daten wie das Geschlecht, das Alter und der Körperbau (Gewicht, BMI) sowie die Messwerte der in SmartAssist verwendeten Umgebungssensoren wie z.B. Raumtemperatur und Luftfeuchtigkeit in der Wohnung.

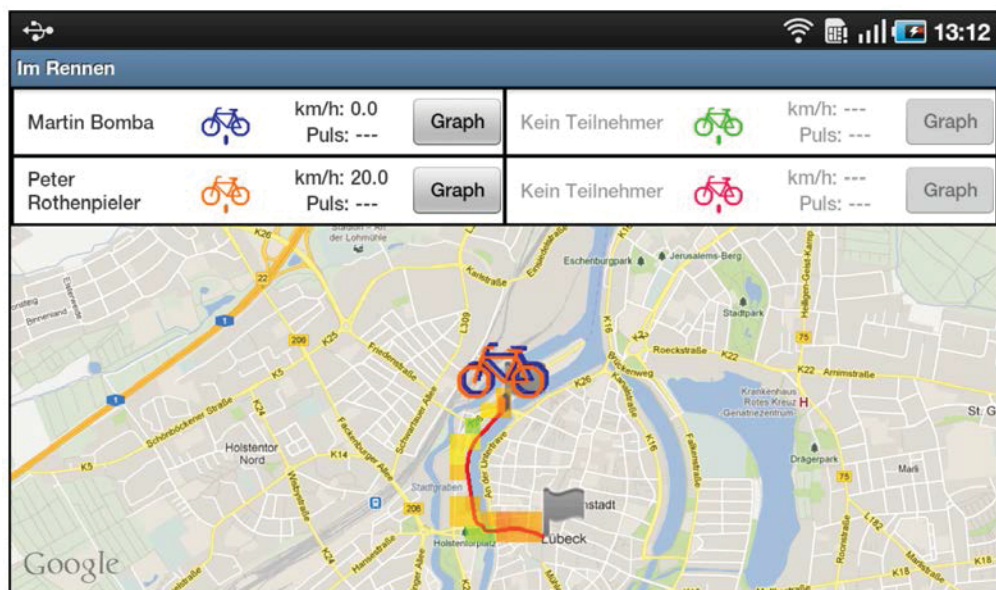


Abbildung 40: Screenshot BikeWars – Rennansicht mit zwei Spielern

1.3.11. Veröffentlichungen und Präsentationen

Das Dynamix-Framework wurde zahlreich publiziert (iiWAS‘2011[13], Mobiquitous‘2011 [14], IoT‘2012 [8][9], PerCom‘2013 [4]) und hat internationale Preise gewonnen.

Das Institut für Telematik hat die beiden ersten Plätze der „IoT Challenge“ auf der 3. Internationalen Konferenz für das Internet der Dinge (IoT‘2012) belegt, die vom 24. - 26. Oktober 2012 in Wuxi, China, stattfand. Die Internationale Konferenz für das Internet der Dinge ist die führende wissenschaftliche Konferenz auf diesem Forschungsgebiet. Die erstmals im Rahmen dieser Konferenz ausgerichtete „IoT Challenge“ prämiert innovative Forschungsbeiträge, die einen wesentlichen Beitrag zur Realisierung des Internet der Dinge leisten. Die Auswahl der Gewinner erfolgte aufgrund einer ausführlichen schriftlichen Beschreibung des Forschungsbeitrages, eines Vortrages sowie einer Demonstration eines voll funktionsfähigen Prototyps vor dem Fachpublikum.

Mit Hilfe der in der Arbeitsgruppe entwickelten Software Ambient Dynamix können reale und virtuelle Sensoren auf einem Android-Smartphone per JavaScript in normalen HTML-Seiten ausgelesen werden. Ebenso wird es möglich, Geräte in der Umgebung entsprechend zu steuern, ohne weitere Software installieren zu müssen. Ambient Dynamix realisiert diese Unterstützung auch während der Laufzeit und lädt notwendige Komponenten dynamisch aus dem Internet. Das Team um Dr. Darren Carlson und Bashar Altakrouri präsentierte ein Beispiel-Szenario zur Erkennung von Pulsraten und Umgebungslicht zur Steuerung der Beleuchtung einer fiktiven Wellness-Anlage aus einem üblichen Webbrowser heraus. Das Preisgeld in Höhe von 1.500 Euro wird für die Anmietung von Servern verwendet und ermöglicht damit die Bereitstellung des Open-Source-Systems für die Wissenschaft in den nächsten Jahren [29]. Siehe auch Presseerklärung der Universität zu Lübeck vom 27.10.2012 (<http://www.uni-luebeck.de/aktuelles/pressemitteilung/artikel/iot-challenge-fuer-zwei-luebecker-informatik-institute.html>).



Abbildung 41: Demonstration und Preisverleihung der IoT Challenge (IoT‘2012)

Auf der 11. IEEE Pervasive Computing and Communication Conference (PerCom 2013) in San Diego, USA, ging der “Best Demonstration Award” ebenfalls an die Arbeitsgruppe Ambient Computing am Institut für Telematik der Universität zu Lübeck unter Leitung von Prof. Dr.-Ing. Andreas Schrader. Das Team um Dr. Darren Carlson (inzwischen an der National University of Singapore) und Bashar Altakrouri präsentierte eine Beispiel-Anwendung für Facebook, mit der sich Bilder aus Photoalben direkt aus der Webseite heraus auf beliebigen Anzeigegeräten in der lokalen Umgebung anzeigen lassen [30]. Siehe auch Presseerklärung der Universität zu Lübeck vom 25.03.2014 (<http://www.uni-luebeck.de/forschung/aktuelles->

zur-forschung/aktuelles-zur-forschung/artikel/webseiten-interagieren-mit-der-realen-welt.html).



Abbildung 42: Vorführung und Best Demonstration Award der IEEE PerCom'2013

1.4. Testbetrieb und Evaluation

Für den Testbetrieb in ausgewählten Haushalten zeichnet der Projektpartner Lübecker Wachunternehmen verantwortlich. Hier wurden zunächst einige Testwohnungen eingerichtet und dann in Zusammenarbeit mit der Vorwerker Diakonie weitere Teilnehmer beworben und deren Wohnungen eingerichtet. Die volle Zahl von geplanten 50 Haushalten wurde innerhalb der Projektlaufzeit nicht erreicht. Die geplante Testung der Dienstleistungen auf Webportal und mobiler Plattform konnte daher nicht erfolgen. Ein funktionaler Test aller Komponenten der Infrastruktur in den ausgewählten Testhaushalten wurde allerdings erfolgreich durchgeführt. Die Daten der Sensoren sind sowohl im Portal, als auch in Dynamix-Testanwendungen unter Android zugreifbar.

Die Testphase konnte im Rahmen dieses Projektes allerdings nicht erfolgreich abgeschlossen werden.

1.5. Medizinische Begleitforschung

Die Aufgabe des Institutes für Allgemeinmedizin bestand in der Durchführung des geriatrischen Basis-Assessments bei allen Probanden und der Kontrollpersonen. Dazu wurden die Untersuchungen (Barthel-Index, Geriatrische Depressionsskala, Mini-Mental-Test, Uhrenzeitentest und Symptomfragebogen 36) bei allen Probanden und Kontrollpersonen halbjährlich erhoben. Die Ergebnisse wurden in einer Datenbank (Excel®) und einem Statistikprogramm (SPSS®) zusammengeführt und ausgewertet.

Das Ziel war, 50 Probanden und 50 Kontrollpersonen (>70 Jahre, möglichst allein lebend, nicht in Pflegeheimen, sondern maximal in betreutem Wohnen untergebracht) zu finden, die bereit waren, ihr Einverständnis zur Studienteilnahme über einen Zeitraum von mindestens zwei Jahren zu geben. Die Zahl 50 pro Gruppe war durch die technischen Möglichkeiten der Installation von Sensoren in den Wohnungen und die finanziellen Ressourcen (Kosten für Sensoren und die Datenzusammenführung) begrenzt und ist nicht auf eine statistisch fundierte Fallzahlberechnung zurückzuführen. Da es sich um eine komplette Neuentwicklung handelt, sind Pilotstudien und Wirkstärkeabschätzungen vorher nicht möglich gewesen.

Die Rekrutierung der Kontrollprobanden erfolgte weitgehend in der Praxis von Dr. Träger (Hausarztpraxis am südlichen Stadtrand von Lübeck, in unmittelbarer Nähe zur Universität). Wir konnten 40 Kontrollpersonen für diese Studie gewinnen. Die Zahl der Kontrollpersonen wurde nicht weiter gesteigert, da die Anzahl der Probanden der Interventionsgruppe nicht in gleicher Weise leicht zu steigern war. Durch den etwas schleppenden Beginn der Rekrutierung hat sich aber eine gleichmäßige Arbeitsbelastung für die Untersuchung ergeben, wodurch der Rahmen der eingeplanten Personalmittel gut eingehalten werden konnte.

Die Probanden und die Kontrollpersonen wurden von jeweils einer Person in halbjährlichem Abstand untersucht und zu soziodemografischen Angaben befragt.

Wir haben zum Jahreswechsel 2012/2013 14 Probanden und 36 Kontrollpersonen eingeschlossen und dokumentiert. Die ersten Daten nach einem Jahr Beobachtung sind in den folgenden Tabellen zusammengefasst.

Item	Probanden	Kontrollen	Unterschied
Anzahl	14	36	
Alter (Jahre)	84,2	84,7	nicht signifikant
Geschlecht (w:m%)	84,6	83,3	nicht signifikant
Wohnsituation Betreutes Wohnen	23,1%	24,9%	nicht signifikant

Tabelle 1: soziodemografische Daten

Item	Probanden	Kontrollen	Unterschied
Allgemeine Gesundheit	4,3	3,2	Probanden gesünder
Vergleich Status zum Vorjahr	3,2	3,2	identisch
Körperl. Fitness	3,1	3,1	identisch
Psychische Stabilität	3,4	3,3	nicht signifikant

Tabelle 2: Ergebnisse Fragebogen SF-36

Item	Probanden	Kontrollen	Unterschied
Barthel-Index	94	94	identisch
MMST (normal: 27-30)	28,3	27,6	Probanden besser
Depressionsskala	2,6	2,5	identisch
Uhrentest	1,6	2,2	Probanden besser

Tabelle 3: Ergebnisse Barthel-Index, Mini-Mental-Test, geriatrische Depressionsskala und Uhrenzeichentest

Zusätzlich wurde die Häufigkeit von notfallmäßigen Krankenhaus- oder Heimunterbringungen, Notfallhausbesuchen des Hausarztes oder des Bereitschaftsarztes und Kontrollen durch Angehörige und Pflegedienste erfragt. Zum Schluss jeder Untersuchung wurde bei den Probanden die Akzeptanz des Sensorsystems abgefragt.

Die oben angeführten Tests (SF 36, Barthel-Index, MMST, GDS, Uhrenzeichentest) sind akzeptierte und weit verbreitete Testsysteme. Mit Ausnahme des SF 36 sind diese Testungen kostenlos. Sie sind weltweit verbreitet im Einsatz. Es gibt ausreichend Vergleichsdaten zu Probandinnen und Probanden der verschiedenen Altersgruppen. Wir können unsere Ergebnisse an diesen Datenbanken messen und vergleichen.

Der Test SF 36 muss beim Hogreve-Verlag Göttingen käuflich erworben werden. Dieser Test ist seit 1998 weltweit im Einsatz. Auch für diese Untersuchung gibt es bundes-, europa- und weltweite Vergleichsdaten, geschlechts- und alterskorrigiert.

Die Datenbankstrukturen (Excel® und SPSS®) sind geläufige Softwareprodukte, die im Institut für Allgemeinmedizin vorhanden sind. Die Auswertung der Daten erfolgte ausschließlich mit diesen Produkten.

Das Ergebnis dieses Teilbereiches wird innerhalb einer Dissertation im Fach Allgemeinmedizin bearbeitet und im Rahmen dieses Verfahrens publiziert werden.

2. Die wichtigsten Positionen des zahlenmäßigen Nachweises

Das Teilvorhaben hatte ein Gesamtvolumen von 628.021,44 Euro. Insgesamt sind die folgenden Mittel verausgabt worden:

Kostenart	812	831	843	846	850	
Zweck	Personal	Gegenstände bis 400€	Allgemeine Sachausgaben	Reisen	Investitionen bis 5000 €	Gesamt
Bewilligt	592.394,94 €	3.379,88 €	4.433,49 €	10.713,75 €	17.099,38 €	628.021,44 €

Tabelle 4: Projektkosten

Die Sachausgaben umfassen im Wesentlichen Laptops mit Dual-Betriebssystem (Windows/Mac) zum Testen der Kompatibilität der Software-Entwicklung unter verschiedenen Betriebssystemen und im mobilen Einsatz. Zudem wurden verschiedene Testgeräte (Smartphones - iPhone/Droid, Tablets - iPad/Galaxy, Laptops - Eee PC/Latitude) für die Entwicklung von mobilen Beispiel-Diensten (siehe Liste der Arbeiten unter Kap. 1.3.10) und zum Testen des Portalzugangs verwendet.



Abbildung 43: SmartAssist Portal auf unterschiedlichen Hardwareplattformen, v.l.n.r: Samsung Galaxy Tab (Android 2.3), Apple iPhone 4S (iOS 6.1), Motorola Milestone (Android 2.2), Galaxy Nexus (Android 4.2, unten), Asus EEEPC (Windows 7)

Für die Speicherung der Analyse-Daten wurde ein externes RAID-System erweitert.

Zudem wurden drei Fahrräder und zwei Heimtrainer-Unterbauten angeschafft. Diese wurden für das Demo-Projekt BikeWars verwendet (siehe Kap. 1.3.10).

Die verbleibenden Sachmittel wurden für Kleinteile, Prepaid-Karten, Batterien, Spezialliteratur, Bildrechte für Marketing, Verbrauchsmaterial, etc. verwendet.

Die Reisemittel umfassen die Kosten für die Teilnahme an Veranstaltungen zur Analyse des State-of-the-Art (IMC'2009 Rostock, Prime Life Summer School 2010 Helsingborg [18], AAL Kongress 2010 Berlin [27], ICASSP 2010 Dallas), zur Vorstellung des Projekts und zur Diskussion der geeigneten Ausgestaltung von Dienstschnittstellen (AAL Forum 2010 Odense [26], MedComm'2011 Bad Bramstedt [23], AAL Kongress 2011 Berlin [16], Medientage Nord 2010 Kiel [24][25]), für den gemeinsam mit dem ULD durchgeführten Workshop für Datenschutz und Sicherheit (Kiel 2010) und zur Präsentation wissenschaftlicher Ergebnisse und Akquise von Projektpartnern für die gemeinsame Arbeit am Teilprojekt Dynamix (Pervasive'2010 Helsinki, BMI'2010 Karlsruhe [17], iiWAS'2011 Ho Chi Mihn City Vietnam [13], Sitzung des Beirats für Menschen mit Behinderungen 2011 in Kiel [22], Mobiquitous'2011 Kopenhagen [14]).

Insgesamt wurden die Reisemittel gegenüber den ursprünglich beantragten Mittel reduziert, da Kosten für die Besuche der Testwohnungen aufgrund von Zeitverschiebungen im Projekt entfallen sind.

Es wird darauf hingewiesen, das Mittel aus der Projektpauschale für die Präsentation von Ergebnissen des Projekts Dynamix auf den Konferenzen IoT'2012 in Wuxi, China und Pervasive'2013 in San Diego, USA verwendet wurden. Auf beiden Events konnten wir den weltweit ersten Platz der Präsentation und Demo erreichen [4][9][8][29][30].

Die Mittel für die medizinische Begleitforschung wurden für eine Entlohnung der Mitarbeiterinnen eingesetzt, welche die Untersuchungen der Probanden und der Kontrollpersonen durchgeführt haben. Es war in einem Zeitraum von vier Jahren im Schnitt eine Viertelstelle (mit zeitlichen Zwischenräumen) nach TVÖD 13 besetzt. Die noch ausstehenden Untersuchungen und die weiteren Dokumentationen sowie die Auswertungen erfolgen kostenneutral durch eine Doktorandin. Die kommenden zwei Jahre werden noch weitergehende Untersuchungen und Befragungen durchgeführt, die notwendig sind, um das Projekt zu einem sinnvollen Ende zu führen und auf dem Sektor der medizinischen Untersuchungen messbare Ergebnisse zu präsentieren.

3. Notwendigkeit und Angemessenheit der geleisteten Arbeit

Die Arbeiten zur Signalanalyse und Ereignisdetektion wurden wie im Antrag beschrieben ausgeführt, und die gesetzten Ziele wurden allesamt erreicht. Ohne diese Arbeiten wäre die Erzeugung eines robusten Ereigniserkennungssystems nicht möglich gewesen.

Arbeitspakete und Meilensteine	Stichtag	Ergebnisse
ISIP-01: <i>Anforderungsanalyse und Erfassung von ersten Trainingsdatensätzen</i>	31.01.2010	<ul style="list-style-type: none"> • Testdaten wurden aufgezeichnet, um die prinzipiellen Eigenschaften der zu erwartenden Daten zu erfassen.
M-ISIP-01: <i>Erste Klassifikationsergebnisse liegen vor</i>	31.01.2010	<ul style="list-style-type: none"> • Erste Klassifikationsergebnisse liegen in Form von ROC-Kurven vor. Ergebnis: Vertauschungsmatrizen sind ein Gütemaß für Multi-Klassen-Probleme und demnach im aktuellen Problem nicht anwendbar.
ISIP-02: <i>Analyse der Trainingssätze</i>	31.07.2010	<ul style="list-style-type: none"> • Erste Erkennungsmethoden sowie Methoden zur Verifikation der Klassifikationsgenauigkeit wurden implementiert.
M-ISIP-02: <i>Klassifikationsergebnisse mit Merkmalsreduktion liegen vor</i>	31.07.2010	<ul style="list-style-type: none"> • Erkenntnisse über Merkmalsauswahl und -reduktion liegen vor.
ISIP-03: <i>Merkmalsreduktion</i>	31.10.2010	<ul style="list-style-type: none"> • Verfahren zur Merkmalsreduktion durch Merkmalskombination und Merkmalsselektion wurden implementiert und hinsichtlich der Eigenschaften getestet.
M-ISIP-03: <i>Untersuchungen zur</i>	31.10.2010	<ul style="list-style-type: none"> • Detaillierte Erkenntnisse über Merkmalsauswahl und -reduktion liegen vor.

<i>Merkmalsreduktion und Klassifikation auf Trainingsdaten sind abgeschlossen</i>		<ul style="list-style-type: none"> • ROC-Kurven belegen die Auswahl.
ISIP-04: <i>Datenerfassung</i>	28.02.2011	<ul style="list-style-type: none"> • Datenerfassung und –aufbereitung mit den zuvor festgelegten Methoden.
ISIP-05: <i>Datenanalyse</i>	31.05.2011	<ul style="list-style-type: none"> • Erstellung von statistischen Modellen für die Ereigniserkennung • Entwicklung einer neuen, robusten Analyseverfahren basierend auf der Theorie bedingter Zufallsfelder
M-ISIP-04: <i>Trainingsdaten aus Ziel-Umgebung sind vorhanden und Klassifikationsalgorithmen sind verfügbar</i>	31.05.2011	<ul style="list-style-type: none"> • Erste Datensätze aus Zielumgebung liegen vor. • Klassifikatoren sind trainiert.
ISIP-06: <i>Implementierung</i>	31.10.2011	<ul style="list-style-type: none"> • Erzeugung einer lauffähigen Software auf der Ziel-Hardware.
M-ISIP-05: <i>Algorithmen sind auf Zielhardware implementiert</i>	31.10.2011	<ul style="list-style-type: none"> • Die Zielhardware ist im Projekt auf Standard-PCs festgelegt worden. • Die Algorithmen sind darauf lauffähig.
ISIP-07: <i>Tests und Analysen</i>	31.07.2012	<ul style="list-style-type: none"> • Verbesserung der Methoden zur Hinzunahme neuer Daten und neuer Erkenntnisse über die Sensorzuverlässigkeit (z. B. Batterieausfall).
ISIP-08: <i>Trennung von Vorgängen</i>	31.07.2012	<ul style="list-style-type: none"> • Bereitstellung von Algorithmen für die Detektion von akuten Ereignissen wie auch der Beschreibung von Langzeittrends.
M-ISIP-06: <i>Ergebnisse</i>	31.07.2012	<ul style="list-style-type: none"> • Ergebnisse zur Leistungsfähigkeit liegen vor, und Methoden zur Detektion von schleichenden und plötzlichen Veränderungen sind verfügbar • Ergebnisse über die Leistungsfähigkeit des Systems liegen vor.

Der Verlauf der Arbeit am Serviceportal folgte der im Projektantrag formulierten Planung. Alle im Arbeitsplan formulierten Aufgaben wurden erfolgreich bearbeitet, es waren mit Ausnahme der kostenneutralen Verlängerung der Projektlaufzeit keine zusätzlichen Ressourcen für das Projekt nötig.

Arbeitspakete und Meilensteine	Stichtag	Ergebnisse
ITM-01: <i>Anforderungsanalyse</i>	31.10.2009	<ul style="list-style-type: none"> • Internes Pflichtenheft (5 Seiten) • Identifikation der Leistungsmerkmale des Por-

		<ul style="list-style-type: none"> tals • Schnittstellen zu Projektpartnern
ITM-02: <i>Grobkonzept und Feinspezifikation</i>	31.03.2010	<ul style="list-style-type: none"> • Internes Lastenheft (25 Seiten) • Spezifikation des Gesamtsystems
M-ITM-01: <i>Grobkonzept und Feinspezifikation</i>	31.03.2010	<ul style="list-style-type: none"> • Erweiterung des internen Lasten- und Pflichtenheftes • Definition Zielgruppen/Rollen Serviceportal • Definition Funktionsblöcke des Portals
ITM-03: <i>Kommunikation mit den Sensorkomponenten</i>	30.06.2010	<ul style="list-style-type: none"> • Schnittstelle zu Demo-Sensoren von coalesenses (Campus) • Schnittstelle zu Haushalt-Sensoren des Lübecker Wachunternehmens
ITM-04: <i>Kommunikation mit der Architektur für Mobilgeräte</i>	30.09.2010	<ul style="list-style-type: none"> • Schnittstelle zu Dynamix → SmartAssist Plug-in über OpenSocial API, Authentifizierung über OAuth
ITM-05: <i>Authentication, Authorization, Accounting</i>	31.01.2011	<ul style="list-style-type: none"> • Website Authentifizierung (HTTPs) • Dienst Authentifizierung (OAuth 1.0)
M-ITM-02	31.01.2011	<ul style="list-style-type: none"> • Spezifikation des Gesamtsystems und Verfügbarkeit eines ersten Prototypen
M-ITM-03	28.02.2011	<ul style="list-style-type: none"> • Verfügbarkeit von Basisdiensten im Portal (Apothekenfinder, BMI Rechner, Horoskop, Lieferservice, Lokale Nachrichten und Anzeigen)
ITM-06: <i>Web Seite</i>	31.01.2012	<ul style="list-style-type: none"> • CMS Drupal, Erweiterungen für SmartAssist • Anschluss an Dienstplattform
ITM-07: <i>Dienstplattform</i>	31.01.2012	<ul style="list-style-type: none"> • Anpassung der OpenSocial Implementierung Shindig für SmartAssist
ITM-08: <i>SOA</i>	31.07.2012	<ul style="list-style-type: none"> • Restful Webservices zum Abrufen von Personen und Sensordaten im Portal
ITM-09: <i>Entwicklung von Basisdiensten</i>	31.07.2012	<ul style="list-style-type: none"> • Konzeption eigenständiger Dienste • Implementierung der Dienste (Apothekenfinder, BMI Rechner, Horoskop, Lieferservice, Lokale Nachrichten und Anzeigen)
ITM-10: <i>Integration der Anwendungen</i>	28.02.2011	<ul style="list-style-type: none"> • Integration existierender Dienste (iGoogle Gadgets der Bundesliga, Deutsche Bahn, Sudoku, Spiegel Online, Wetter24.de, ...) • Integration der eigenständigen Dienste
ITM-11: <i>Entwicklung von mobilen Zusatzdiensten</i>	30.11.2011	<ul style="list-style-type: none"> • Implementierung des SmartAssist Cockpit • Anpassung der Webseite und Dienstplattform an mobile Browser

M-ITM-04	30.11.2011	<ul style="list-style-type: none"> • Verfügbarkeit einer kompletten Dienstimplementierung
ITM-12: <i>Aufbau Testbetrieb Haushalte</i>	31.08.2011	<ul style="list-style-type: none"> • Testbetrieb unter Laborbedingungen beim Lübecker Wachunternehmen und Coalesenses – Anbindung ans Portal und Integration der Sensoren
ITM-13: <i>Aufbau Testbetrieb Betreutes Wohnen</i>	31.01.2012	<ul style="list-style-type: none"> • Testbetrieb in einem Haushalt
M-ITM-05	31.07.2012	<ul style="list-style-type: none"> • Verfügbarkeit der Gesamtplattform

Auch die Entwicklung der mobilen Komponente Ambient Dynamix konnte mit den veranschlagten Ressourcen erfolgreich realisiert werden. Durch den Einsatz von zahlreichen Bachelor- und Masterarbeiten wurden darüber hinaus weitere mögliche Dienstleistungen prototypisch entwickelt und getestet.

Arbeitspakete und Meilensteine	Stichtag	Ergebnisse
M-ISNM-01: <i>Aufbau Entwicklungsumgebung</i>	31.10.2009	<ul style="list-style-type: none"> • Dokument: Spezifikation mit Requirement Analyse, Marktübersicht Geräte, Plattformen und Tools (10 S.) • Machbarkeitsstudie unter Berücksichtigung von Android, Felix OSGi and Tools. • Mobile Entwicklungsumgebung
M-ISNM-02: <i>Systemkomponenten für Java-basierte mobile Endgeräte</i>	31.07.2010	<ul style="list-style-type: none"> • Erster Prototyp des Aladdin-Frameworks gemäß Spezifikation in M-ISNM-01 • Ersters Design der Context Firewall • Aladdin Framework Version 0.5.0. • Context Plug-in API (v0.5.0) • Application API (v0.5.0)
M-ISNM-03: <i>Module für Context Acquisition and Modelling</i>	31.01.2011	<ul style="list-style-type: none"> • Erweiterung von Aladdin um dynamisch installierbare Plug-ins (inkl. Dokumentation und API Updates) • Aladdin Framework Update 0.6.0. • Plug-in Pack 1: Enhanced Location; Wireless Beacon; Barcode Scanner; SmartAssist (inkl. SmartAssist-Portal integration). • Plug-in Deployment Tests
ISNM-04: <i>Repository</i>	31.07.2011	<ul style="list-style-type: none"> • Web-basierte Repository-Architektur Version 1.0 • Dateisystem für Repository-Architektur Version 1.0 • Maven Repository-Architektur Version 1.0
ISNM-10 / M-ISNM-04: <i>Entwicklung von mobilen</i>	30.11.2011	<ul style="list-style-type: none"> • Beispiel-Anwendung für Entwickler (logger) • Erste Version eines SmartAssist Cockpit-

Zusatzdiensten		Services
ISNM-05 / M-ISNM-05: <i>Erste Prototypen für Context Interpreter</i>	31.01.2012	<ul style="list-style-type: none"> • Plug-in Pack 2: Sound Pressure Level; Pedometer; Heart Rate monitor; Near Field Communications (NFC).
ISNM-06 / M-ISNM-06 <i>Lauffähiges Gesamtsystem</i>	31.01.2012	<ul style="list-style-type: none"> • Änderung des Namens von Aladdin auf Ambient Dynamix • Framework für Performanz-Tests und Optimierungen (Veröffentlicht auf IoT 2012). • Dynamix Framework Update 0.9.0. • Context Plug-in API Version 0.9.0. • Application API Version 0.9.0.
ISNM-07/08 / M-ISNM-07 <i>Toolkit für Drittanbieter / Einbindung von Partnerdiensten</i>	31.07.2012	<ul style="list-style-type: none"> • Dynamix: Webseite für Entwickler Version 1.0 (http://ambientdynamix.org) • Developer SDK Version 0.9.44 (inkl. APIs, Dokumentation und Beispiele) • Plug-in Pack 3 (Drittanbieter-Code): Zeo Sleep State; Echoprint Acoustic Fingerprint; Google Speech-to-Text; Withings Health and Fitness; Umweltbundesamt Air Quality Monitor.
Verlängerungsphase	31.12.2012	<ul style="list-style-type: none"> • Unterstützung von Android 4.x. • Dynamix Web Connector zur Unterstützung von Browser-basierten Anwendungen • Plug-in Pack 4: Battery Level; Light Sensor; ArtNet Light Control (Hausautomatisierung); Ambient Media (Zugriff auf Geräte mit Air-Play and UPnP)

Die Aufwendungen für die Untersuchungstätigkeit in der medizinischen Begleitforschung sind – im Verhältnis zum Arbeitseinsatz und dem Zeitumfang – sehr gering. Ein geriatrisches Basis-Assessment dauert normalerweise ca. 30 Minuten, die Bearbeitung des Fragebogens „SF 36“ dauert nochmals ca. 20 Minuten. Die restlichen Fragen können normalerweise in ca. 10 Minuten erhoben werden. Pro Proband (und pro Kontrollperson) ist also halbjährlich jeweils eine Stunde zu investieren, dazu kommen An- und Abfahrt zum Wohnort des Probanden. Im Schnitt haben wir mit 40 Probanden und 40 Kontrollpersonen pro Halbjahr mal ca. zwei Stunden pro Person zu rechnen (160 Arbeitsstunden). Dateneingabe und Organisation der Besuchstermine erfordern zusätzlich nochmals etwa 40 Stunden pro Halbjahr. Durch geschickte Planung kann man die Fahrtrouten zumindest teilweise synchronisieren, wodurch weitergehende Fahrtzeiten eingespart werden können. Die Eingabe der Daten erfolgte meist gebündelt, häufig auch am häuslichen Rechner, wodurch ebenfalls Fahrt- und Arbeitszeit eingespart werden konnte.

4. Voraussichtlicher Nutzen und Verwertbarkeit der Ergebnisse

4.1. Wirtschaftliche Erfolgsaussichten nach Projektende

Die Universität verfolgt keine wirtschaftlichen Interessen an diesem Projekt. Für eine detaillierte Abschätzung der wirtschaftlichen Aussichten sei auf die Berichte der Projektpartner verwiesen.

4.2. Wissenschaftlich-technische Erfolgsaussichten nach Projektende

Der Einsatz des SmartAssist Serviceportals ermöglicht den Zugriff auf die im Portal gesammelten Informationen und Sensorwerte sowie der hierauf aufbauenden Dienstleistungen zu jedem Zeitpunkt von jedem Ort der Welt. Auf diese Weise haben die Paten eines Benutzers jederzeit die Möglichkeit, sich über den Gesundheitszustand des Benutzers zu informieren, mit diesem Kontakt aufzunehmen oder durch das System auf Veränderungen aufmerksam gemacht zu werden. Durch den einfachen Datenzugriff über Standardisierte Web-Technologien ist der Zugriff von einer Vielzahl von Geräten (Tablets, Smartphones, PCs, Laptops) möglich.

Insbesondere ist durch die Erweiterung auf mobilen Plattformen mit Hilfe des Dynamix-Frameworks die Möglichkeit gegeben, auch in nativen Anwendungen (Android-Apps) auf einfache Weise auf die SmartAssist-Profildaten zuzugreifen. Darüber hinaus kann das mobile Gerät durch die verbauten Sensoren weitere Kontext-Informationen, teilweise auch Vitalwerte (Schrittzähler) liefern und somit auch zur Entwicklung eines Mobile Health Record führen (siehe auch [32]). Durch die drahtlose Anbindung von weiteren tragbaren, mobilen oder auch stationären Meßgeräten außerhalb von SmartAssist ermöglicht Dynamix die Einbindung prinzipiell beliebiger Datenquellen und öffnet die Architektur für Weiterentwicklungen und neue Produkte in diesem Bereich.

Für die Mediziner waren das Befinden und die Einschätzung der Probanden im Vergleich zur Kontrollpopulation wichtig. Hierbei sind natürlich auch die Einsparungen von wichtigen Ressourcen (z.B. Kosten bei Krankenhaus- oder Pflegeheimunterbringung, Kosten für Notfallhausbesuche des Haus- oder des Bereitschaftsarztes) interessant, wenngleich die kleine Population und der relativ kurze Zeitraum volkswirtschaftlich verlässliche Angaben nicht zulassen. Eine Trendbestimmung sollte aber sehr wohl möglich sein.

Die Pflegeheimunterbringung ist für ältere Menschen eine letzte, meist nicht erfreuliche Möglichkeit der Lebensführung und beendet weitgehend die eigenständige Entscheidungsfreiheit. Der Gedanke von SmartAssist leitet sich vom Plan her, die Zeit bis zur Heimunterbringung bei allein lebenden älteren Menschen durch technische Systeme (Sensortechnik, Funkübertragung, interaktive Netze) möglichst lange hinauszuzögern. Die Anzahl der zur Verfügung stehenden Heimplätze reicht zurzeit bei der demografischen Entwicklung nicht aus, alle Menschen frühzeitig in Heimen unterzubringen, außerdem ist es meist der Wille der Menschen, möglichst lange eigenbestimmt in ihrer Wohnung zu verbleiben. Volkswirtschaftlich ist die Heimunterbringung meist erheblich kostenintensiver, es kann also durch die Sensorsysteme eine erhebliche Menge Geldes eingespart werden.

4.3. Wissenschaftliche und wirtschaftliche Anschlussfähigkeit

Durch die einfache und weit verbreitete Schnittstelle (OpenSocial) ist die flexible Erweiterbarkeit des Serviceportals durch Dienste von Drittanbietern möglich, wodurch der Mehrwert des Serviceportals für alle Benutzergruppen gesteigert werden kann. Auf diese Weise können neue Dienstleistungen sehr einfach in das Portal integriert und getestet werden. Darüber hinaus ergeben sich von wissenschaftlicher Seite Fragestellungen zum erweiterten Schutz der Privatsphäre der Benutzer, etwa durch die Möglichkeit, dass ein Benutzer mehrere Identitäten oder Pseudonyme zum Schutz der eigenen Daten verwenden kann.

Während die Daten auf dem SmartAssist-Server noch relativ leicht gegen Mißbrauch gesichert werden können, entstehen auf mobilen Geräten unter Verwendung der Ambient Dynamix-Architecture auch völlig neue Fragestellungen. Insbesondere muß für die kontrollierte Absicherung von Informationsflüssen zwischen Dynamix Plug-Ins und Dynamix Apps eine intuitive Möglichkeit gefunden werden, diese zu personalisieren. Erste Ansätze wurden mit der Context Firewall bereits unternommen. Allerdings sind hier weitere Arbeiten notwendig.

Zurzeit läuft eine Kooperation mit der Universität Wien und der National University of Singapur zur Entwicklung eines vollständigen Sicherheitskonzepts in Dynamix.

Das Konzept von Dynamix ist nicht auf die Anwendung im Projekt SmartAssist beschränkt und kann grundsätzlich für eine Vielzahl von unterschiedlichen mobilen Anwendungen zum Einsatz kommen. Insbesondere die Erweiterung auf Web-basierte Anwendungen im Browser eröffnet völlig neue Konzepte im Bereich der Kontext-Sensitivität im Internet.

5. Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen

Serviceportal

Auf dem Gebiet des Serviceportals haben sich in der Zwischenzeit mehrere technische Neuerungen ergeben, da viele der verwendeten Technologien mittlerweile in aktualisierten Versionen vorliegen. So liegt das Open Source Content Management System Drupal mittlerweile in Version 7.22 (April 2013) vor und auch an Drupal 8 wird bereits gearbeitet. Die neueren Versionen von Drupal bieten viele Neuerungen, da neue Module, Erweiterungen und Sicherheitsupdates nicht länger für ältere Versionen zur Verfügung stehen werden. Auch OpenSocial bzw. Apache Shindig haben sich in der Zwischenzeit weiterentwickelt – so steht Shindig nun in Version 3.0 als Beta Version zur Verfügung, welches OpenSocial in Version 2.5 implementiert. Zusätzlich hat sich Apache Rave entwickelt, welches neben Shindig noch weitere Bestandteile integriert und diese als ein gemeinsames Projekt weiterführt.

Alle existierenden Dienste, welche den Google Gadget Standard verwenden und welche in SmartAssist zur Demonstration der Erweiterbarkeit integriert wurden, entstammen dem iGoogle Verzeichnis. Dieser ursprünglich als „personalisierte Startseite“ von Google konzipierte Dienst wird im November 2013 eingestellt, wodurch vermutlich auch die Dienste als solche eingestellt werden. Die in SmartAssist entwickelten Dienste sind hiervon nicht betroffen, da diese direkt im Serviceportal bereitgestellt werden – dasselbe gilt für einige Dienste von Drittanbietern, welche nicht direkt über iGoogle angeboten werden. Der Google Gadget Standard, welcher die Basis für die in OpenSocial verwendeten Gadgets (Dienste) bildet wird jedoch noch im Rahmen von OpenSocial weiter entwickelt und findet auch in abgewandelter Form in Googles aktuellen Sozialen Netzwerk Google+ Verwendung. Es bleibt abzuwarten, wie die Zukunft des OpenSocial Standards verlaufen wird, da diese zwar als offener Standard von der OpenSocial Foundation entwickelt wird, jedoch ursprünglich von Myspace und Google initiiert wurde. Die Weiterentwicklung des Standards wird gegenwärtig in erster Linie von IBM, SugarCRM und Jive Software vorangetrieben, welche OpenSocial für ihre Enterprise Lösungen verwenden.

Verhaltensanalyse

In jüngerer Vergangenheit wurden System zur automatischen Überwachung und Verhaltensanalyse weiter entwickelt [104][105]. Allerdings befassen sich sehr viele der bekannten Systeme mit Kameras, Mikrofonen und tragbaren Sensoren, die im Rahmen des Projekts Smartassistent ausgeschlossen wurden. Nach unserem Kenntnisstand existiert kein System mit einer vergleichbar inhomogenen Sensor-Ausstattung, wobei zugleich auf tragbare Sensoren verzichtet wird.

Ein System, das dem von SmartAssist nahe kommt, wurde in [106] vorgestellt. Allerdings befasst sich dieses ausschließlich mit Bewegungssensoren. Von der analytischen Seite werden in diesem System Subsequenzen verglichen. Dies stellt eine sinnvolle Ergänzung zu den Methoden dar, die im Projekt SmartAssist verwendet wurden, sie kann aber nicht als Alternative betrachtet werden, da ein anderes, homogenes Sensor-Netzwerk verwendet wird und die Sensoren vor allem untereinander verglichen werden, was im Projekt SmartAssist nicht mit jedem

beliebigen Sensorpaar möglich ist. Dadurch ist die Integration dieser neuen Erkenntnisse nicht trivial und wegen der Aktualität der Veröffentlichung noch nicht erfolgt.

Betrachtet man insgesamt die Systeme zur Überwachung [107], so werden weiterhin Markov-Modelle bevorzugt verwendet, um das Verhalten der Bewohner zu analysieren. Ähnlichkeiten der Methoden ergeben sich vor allem daraus, dass die Informationen zur Entwicklung frei verfügbar sind und allgemein anerkannte Standards darstellen. Conditional Random Fields werden jedoch selten zur direkten Analyse der Daten eingesetzt; uns ist kein direkt vergleichbares System bekannt. Da diese Modelle in Hinblick auf das Gesamtsystem von SmartAssist bedeutende Vorteile gegenüber klassischen Hidden-Markov-Modellen besitzen, die insbesondere bei dem inhomogenen Sensor-Netzwerk wichtig sind, ist die Verwendung dieser Modelle als Erweiterung der aktuellen Kenntnislage zu sehen und dadurch ein positiver Beitrag des Forschungsgebiets der Verhaltensanalyse.

Mobilplattform

In den letzten Jahren, hat sich die Open Services Gateway Initiative (OSGi) [81] kontinuierlich als Industriestandard für die Verteilung von Komponenten in Java etabliert. OSGi verwendet einen modularen Ansatz, in dem Softwarefunktionalität in logische Einheiten (Bundles) gekapselt werden. Bundles enthalten sowohl ausführbaren Code, als auch Metadaten, die Sichtbarkeiten und Abhängigkeiten des jeweiligen Moduls beschreiben. Eine OSGi-Anwendung besteht aus einem oder mehreren Bundles, die dynamisch während der Laufzeit mit Hilfe eines OSGi-Frameworks integriert werden können. Bundle-Integration wird dabei von mehreren Schichten des Frameworks (Layer) unterstützt. Dies umfasst eine *execution environment layer* (z.B. Java virtuelle Maschine); eine *module layer*, die ein erweitertes Java-Package Management zur Verfügung stellt; eine *life cycle layer* für das Laufzeitmodell der Bundles; eine *service layer*, die Bundle-Interfaces bindet; und eine *crosscutting security layer*, die das Java 2 Sicherheitsmodell erweitert.

Aktuelle Erweiterungen der Apache Felix OSGi-Implementierung erlauben die Nutzung auf nicht modifizierten Android-basierten Geräten durch Cross-Kompilierung von normalen Java .class-Dateien in das Android-native Dalvik Bytecode-Format. Diese Erweiterungen von OSGi in Kombination mit den Möglichkeiten von Android im Bereich Multitasking (z.B. Hintergrund-Prozesse) und Inter Process Communication (IPC) haben die Entwicklung von hochadaptiven Kontext-Architekturen für das stark wachsende Android-Ökosystem ermöglicht.

Daher haben einige Projekte begonnen, weitreichende Kontext-Modellierung auf Android-Geräte auszuloten. Frühe Projekte, wie Local [82] nutzen die im Gerät integrierten Sensoren, um bei Kontext-Wechseln (z.B. Position) automatisierte Anpassungen der Telefoneinstellungen durchzuführen (z.B. Wechsel der Alarmfunktion). Ähnliche Architekturen, wie Context-DROID [83], bieten generalisierte Unterstützung für Kontext-Interpretation (d.h. Kontext-Repräsentation höherer Ordnung) und Energie-Überwachung. Das Context-Aware Machine learning Framework for Android (CAMF) untersucht ebenfalls Techniken der Sensor-Abstraktion und Kontext-Interpretation höherer Ordnung. Insbesondere schlägt CAMF die Nutzung von *processing widgets* als diskrete Abstraktion zur Abstraktion komplexer Verfahren des maschinellen Lernens vor (basierend auf einer ähnlichen Notation im Context Toolkit [84]). Die Einführung von *processing widgets* ist einer der ersten Ansätze für Plugin-basierte Adaptation unter Android. Das Funf Open Sensing Projekt des MIT [85] verwendet ebenfalls eine erweiterbare Plug-In-Architektur zur Integration von Plug-Ins zur Kontextmodellierung von Drittanbietern vor.

Im Gegensatz zu CAMF und Funf, welche die Integration der Komponenten nur statisch unterstützen, ist das Mobile USers In Ubiquitous Computing Environments (MUSIC) System [86] eines der ersten, das OSGi Container auch dynamisch zur Laufzeit in Android-Anwendungen integriert. 2010 [87] wurde eine Architektur zur Verkehrsüberwachung und

Generierung von Verkehrsdienstleistungen vorgestellt. 2011 wurde eine Architektur zur Datensammlung basierend auf OSGi, Android und dem eXtensible Messaging and Presence Protocol (XMPP) vorgestellt [88].

Obwohl diese Projekte Ähnlichkeiten zu Dynamix aufweisen, unterscheiden sie sich in mehreren wichtigen Aspekten. (1) Dynamix unterstützt das automatische Erfassen und Integrieren von Komponenten (Kontext-Plug-Ins) während der Laufzeit und nicht nur während der Entwicklung. (2) Dynamix bietet umfangreiche Unterstützung für die Entwicklung von Plug-Ins und die individuelle Repräsentation von Kontext basierend auf *Plain Old Java Objects* (POJOs) und ermöglicht damit Anwendungen den Umgang mit klassischen Java-Objekten, statt eine Kodierung mit Zeichenketten zu erzwingen (obwohl auch diese unterstützt werden). (3) Die *Context Firewall* des erweiterten Sicherheitskonzepts von Dynamix ermöglicht eine sichere Isolation dynamisch installierter Plug-Ins (*plug-in sandboxing*) und ermöglicht den Anwendern den Informationsfluß zu den Anwendungen präzise zu konfigurieren. (4) Dynamix integriert sich nahtlos in das Anwendungsmodell von Android, daher können Android-Programmierer ihre Kenntnisse und Werkzeuge wie gewohnt verwenden. (5) Dynamix unterhält eine Community-Initiative um eine kontinuierliche Weiterentwicklung der Architektur zu beschleunigen und die spezifischen Bedarfe der Entwickler zu berücksichtigen.

Nur sehr wenige Architekturen für mobilen Kontext unterstützen klassische Web-Anwendungen in nicht modifizierten Webbrowsern. Die standardisierte Webumgebung bietet nur wenig Unterstützung für Kontext-Sensitivität. Die Entwicklung von Kontext-sensitiven Web-Anwendungen hat daher mit der Modifizierung von Webbrowsern begonnen, um Kontext-Daten in Webseiten integrieren zu können [89][90].

Normalerweise werden Kontext-Informationen von einer dedizierten Kontext-Datenbank auf einem externen Server abgefragt, wobei Typen und Datenformate vordefiniert sind. COOL-TOWN [91] bildet einen der ersten Versuche, Kontext-sensitive Komponenten in die Web-Architektur zu integrieren. Allerdings sind die Mechanismen noch proprietär und nicht sehr verbreitet. Das Web der Dinge (Web of Things – WoT) ist der Versuch, Architekturen zu entwickeln, bei denen ambiante Geräte retrospektiv mit eingebetteten Webservern und/oder zusätzlichen Gateways anzureichern, um lokal erfasste Kontextdaten zu einem Web-basierten Speicher weiterzureichen. WoT befindet sich in der Entwicklung und bisher bietet die überwiegende Mehrheit ambianter Umgebungen noch keinen standardisierten Web-basierten Zugang.

Projekte wie PhoneGap²⁸ ermöglichen die Entwicklung von hybriden Anwendungen, in denen Web-basierte Nutzerschnittstellen (User Interface – UI) mit den Funktionen installierbarer Anwendungen (Apps) verbunden werden. PhoneGap verwendet einen festen Satz von UI-Komponenten in der Oberfläche eines in die App integrierten Webbrowsers. PhoneGap ermöglicht Zugang zu den lokalen Ressourcen des Gerätes (z.B. Kameras oder Sensoren), allerdings sind Apps in PhoneGap (und vergleichbaren Ansätzen²⁹) nicht mit nativen Browsern kompatibel und müssen daher im Gerät installiert werden.

Jones und Brown [92] haben für die Suche im virtuellen Web auf Basis von Situationen in der physikalischen Umgebung den Begriff Context-Aware Retrieval (CAR) vorgeschlagen. Die Herausforderung besteht darin, die Suche nach Webressourcen durch Mechanismen zur Interpretation und Filterung von Kontext zu verbessern. Es gab bereits eine Reihe von Ansätzen CAR durch die Verknüpfung von Ubiquitous Computing und Web-Technologien zu realisie-

²⁸ <http://phonegap.com>

²⁹ <http://www.mosync.com/>

ren. Erwähnenswert sind die Ansätze von stick-e notes [93], Socialight³⁰ und Tagzania³¹. Die Sentient Graffiti-Architektur von Lopez u.a. [94] verwendet Monitore zur Erfassung von Kontext und Kontext-Wechsel auf dem lokalen Gerät und unterstützt sowohl *Pull* als auch *Push*-Funktionalitäten für lokale Browser. Coppola et al. [95] erweitern das CAR-Konzept durch einen konfigurierbaren *Context-Aware Browser* (CAB) und unterstützen heterogene, unsichere und komplexe Kontext-Typen.

Andere Ansätze unterstützen auch nicht modifizierte Web Browser. Die PinPoint-Plattform [96] ergänzt den (üblicherweise mobilen) Client um einen Proxy-Prozess. Der Proxy reichert die HTTP-Anforderungen durch einen lokalen Kontext-Manager an, der die lokalen Kontextinformationen von der Geräte-Hardware abgreift. Optional werden noch zusätzliche semantischer Informationen von einem externen semantischen Server ergänzt. Der Proxy leitet die angereicherten HTTP-Anforderungen im Namen des lokalen Browsers an den Webserver weiter. Kontextinformationen werden in Webseiten und URLs durch neue Tags integriert, die vorab zwischen jedem Client und Server vereinbart werden müssen. Noltes beschreibt ein Browser-Plug-In für den Microsoft Internet Explorer zur Unterstützung von lokalen Informationen durch *extended PIDV* (IETF RFC4119) im http-Header.

Zur Unterstützung von nativen Browsern wird von der W3C Geolocation Working Group eine Spezifikation für eine Geolocation-API³² (für den Zugriff auf lokale geographische Informationen) und DeviceOrientation-Events³³ (für den Zugriff auf die Ausrichtung und Bewegung des lokalen Gerätes) erarbeitet. Zusätzlich definiert die Device API Working Group des W3C Client-seitige APIs zur Interaktionen von Web-Anwendungen mit lokalen Sensoren und Multimedia-Hardware.

Soweit bekannt ist, unterstützt keiner der existierenden Ansätze konventionelle Browser-basierte Web-Anwendungen mit einem plug-and-play-Zugriff auf höherwertige Kontextinformationen oder adaptive ad-hoc-Interoperation mit Geräten und Diensten in der lokalen Umgebung. Den bisherigen Ansätzen fehlt die Unterstützung für dynamische Erweiterung während der Laufzeit, beliebige Kontexttypen, eine Erweiterbarkeit durch einen Community-Mechanismus und robuste Sicherheitsfunktionen. Während einige Ansätze Web-Anwendungen einen limitierten Zugang zu Sensordaten ermöglichen, beschränkt sich dies meist auf den Zugriff auf einen festen Satz von Werten, die die Hardware direkt zur Verfügung stellt.

Daher sind Browser-basierte Web-Anwendungen nach wie vor nicht in der Lage, mit der reichen Vielfalt von Kontextquellen und Aktoren in heterogenen mobilen Umgebungen dynamisch zu interagieren. Der im Projekt entwickelte Ansatz von Dynamix ist hier weltweit ein Novum und ermöglicht völlig neue Anwendungen, auch über die spezifischen AAL-Anforderungen in SmartAssist hinaus.

³⁰ <http://socialight.com/>

³¹ <http://www.tagzania.com>

³² <http://www.w3.org/TR/2010/CR-geolocation-API-20100907/>

³³ <http://dev.w3.org/geo/api/spec-source-orientation.html>

Abbildungsverzeichnis

Abbildung 1 Eons.com.....	16
Abbildung 2 PatiensLikeMe.com.....	17
Abbildung 3 Dailystrength.org.....	17
Abbildung 4 MedHelp.org	18
Abbildung 5 Profilansicht eines Benutzers	26
Abbildung 6: Forum und Chat von SmartAssist	27
Abbildung 7: Überblick über die Dienstplattform von SmartAssist	27
Abbildung 8: Visualisierung der Sensorwerte im Portal.....	30
Abbildung 9: Erstellung von zwei Context Subscriptions für die Profil- und Sensordaten.....	30
Abbildung 10: Abfragen der Sensordaten durch einen Context Request.....	31
Abbildung 11: Zugriff auf die Profil- und Sensordaten bei Erhalt eines Context Events.....	31
Abbildung 12 Beispiel einer Gadget Implementierung (Meta Daten und Initialisierung).....	32
Abbildung 13: Beispiel einer Gadget Implementierung (Verarbeitung der persönlichen Daten)	33
Abbildung 14: Basisdienste im Portal.....	34
Abbildung 15: Externe Dienste (von iGoogle übernommen)	35
Abbildung 16: Flussdiagramm des Klassifikators zur Ereignisdetektion.	37
Abbildung 17: Datenanalyse mit einem deaktivierten Sensor über den markierten Zeitraum	40
Abbildung 18: Simulation eines Ereignisses im markierten Zeitraum	40
Abbildung 19: Weltweite Verteilung von Smartphone-Betriebssystemen im 2. Quartal 2009 (Canalys)	43
Abbildung 20: Erfüllung der Anforderungen von Dynamix durch aktuelle Betriebssysteme.	44
Abbildung 21: Überblick über das Dynamix Framework	46
Abbildung 22: The Dynamix Service's Home and Plug-ins Tabs	47
Abbildung 23: Festlegung einer Sicherheitsregel in der Dynamix Context Firewall	48
Abbildung 24: AmbientWeb - Dynamix Erweiterung für Web Apps	49
Abbildung 25: Dynamix Entwickler-Communities	52
Abbildung 26: Dynamix Webseite.....	52
Abbildung 27: Auswertung typischer Dynamix Laufzeit-Phasen.....	54
Abbildung 28: CPU-Auslastung für verschiedene Payloads und Abfrageraten	55
Abbildung 29: Speicherverbrauch für verschiedene Payload-Größen und Anfrageraten.....	55
Abbildung 30: Kontext Plug-ins	57
Abbildung 31: AR-Unterstützung für Kaffeemaschine und zugehöriges Zustandsdiagramm	57
Abbildung 32: Einige Dialogbeispiele der mobilen Arzneimittel-Anwendung.....	58
Abbildung 33: Evaluation der Bewegungskompensation durch Schachbrettmuster	59

Abbildung 34: Vergleich von realer (Kamera – blau) und sensorisch (rot) erfasster Tremor-Amplitude.....	60
Abbildung 35: GUI des Betreuers, Scannen des RFID-Tags in der Wohnung, GUI des Patienten (von links nach rechts).	60
Abbildung 36: Live-Wallpaper bei unterschiedlichen Gesundheitszuständen und detaillierter Angabe der Historie im Wochenverlauf.....	61
Abbildung 37: Screenshots der des Personal Health Record: Parameter, Ziele, Aktivität, Erfolge (von links nach rechts)	62
Abbildung 38: PHR: Visualisierung des historischen Verlaufs von kontinuierlichen (links) und diskreten (rechts) Werten.	62
Abbildung 39: Übersicht über die BikeWars Architektur.....	63
Abbildung 40: Screenshot BikeWars – Rennansicht mit zwei Spielern	63
Abbildung 41: Demonstration und Preisverleihung der IoT Challenge (IoT‘2012).....	64
Abbildung 42: Vorführung und Best Demonstration Award der IEEE PerCom‘2013.....	65
Abbildung 43: SmartAssist Portal auf unterschiedlichen Hardwareplattformen, v.l.n.r.: Samsung Galaxy Tab (Android 2.3), Apple iPhone 4S (iOS 6.1), Motorola Milestone (Android 2.2), Galaxy Nexus (Android 4.2, unten), Asus EEEPC (Windows 7).....	68

III. Veröffentlichungen

1. Workshops und Konferenzen

- [1] Daniel Burmeister & Andreas Schrader. *Aktivitätserkennung für Ambient Health Monitoring*. In: GMDS'2013, In: Proceedings of 58. Jahrestagung der Deutschen Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie (GMDS), Student Workshop, Lübeck, Germany, Sep. 1-5, 2013.
- [2] Darren Carlson, Peter Rothenpieler, Andreas Schrader. *An Open Infrastructure and Platform for AAL Services*, International Conference on Distributed, Ambient and Pervasive Interactions (DAPI'13), special session on Smart Healthcare Applications and Ambient Assisted Living, at 15th International Conference on Human-Computer Interaction (HCI International 2013), Las Vegas, USA, Juli 21-26, 2013.
- [3] Dierck Matern, Alexandru Condurache, Alfred Mertins. *Adaptive and Automated Ambiance Surveillance and Event Detection for Ambient Assisted Living*, Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'2013), Juli 3-7, 2013.
- [4] Dierck Matern, Alexandru Condurache, Alfred Mertins. *Automated Intrusion Detection for Video Surveillance Using Conditional Random Fields*, In Proceedings of International Conference on Machine Vision Applications (IAPR'2013), Kyoto, Japan, Mai 20-23, 2013.
- [5] Daniel Burmeister & Andreas Schrader & Darren Carlson. *A Modular Framework for Ambient Health Monitoring*. In: LLPH 2013, Proceedings of the 1st International Workshop on Lifelogging for Pervasive Health at the 7th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), Venice, Mai 5-8, 2013.
- [6] Darren Carlson, Bashar Altakrouri & Andreas Schrader. *Reinventing the Share Button for Physical Spaces*. In: Percom'2013, Demo Session at IEEE International Conference on Pervasive Computing and Communication (PerCom 2013), San Diego, California, USA, März 18-22, 2013.
- [7] Dierck Matern, Alexandru Condurache, Alfred Mertins. *Automated Sequence Clustering of Audio Signals using Conditional Random Fields*, Proceedings of AIA-DAGA Conference on Acoustics, Merano, Italy, März 18-21, 2013.
- [8] Darren Carlson, Andreas Schrader: *Dynamix: An Open Plug-and-Play Context Framework for Android*, In Proceedings of the 3rd International Conference on the Internet of Things (IoT2012), Wuxi, China, Oktober 2012.
- [9] Darren Carlson, Bashar Altakrouri, Andreas Schrader: *AmbientWeb: Bridging the Web's Cyber-physical Gap*, In Proceedings of the 3rd International Conference on the Internet of Things (IoT2012), Wuxi, China, Oktober 2012.
- [10] Dierck Matern, Alexandru Condurache, Alfred Mertins. *Event Detection using Log-Linear Models for Coronary Contrast Agent Injections*, In Proceedings of the International Conference on Pattern Recognition Applications and Methods (ICPRAM'2012), Vilamoura, Portugal, Februar 6-8, 2012.

- [11] Dierck Matern, Alexandru Condurache, Alfred Mertins. *Linear Prediction based Mixture Models for Event Detection in Video Sequences*, In Proceedings of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA), pp. 25-32, Las Palmas de Gran Canaria. Spain. Juni 8-10 2011.
- [12] Andreas Schrader, Peter Rothenpieler, Stefan Fischer, Jens-Martin Träder. *Ambient Socio-Technical Support for Assisted Autonomous Living*, In ERCIM News, Special Issue on Ambient Assisted Living, 2011.
- [13] Darren Carlson, Andreas Schrader. *A Wide-area Context-awareness Approach for Android*, In Proceedings of the 13th International Conference on Information Integration and Web-based Applications & Services (iiWAS'2011), Ho Chi Minh City, Vietnam, Dezember 2011.
- [14] Darren Carlson, Andreas Schrader. *Adaptive OSGi-based Context Modeling for Android*, In Proceedings of the 8th International ICST Conference on Mobile and Ubiquitous Systems (MobiQuitous 2011), Dezember 2011.
- [15] Andreas Schrader. *Technische Innovationen zur ambulanten Betreuung pflegebedürftiger Menschen – SmartAssist*, In Berufung Gesundheit Gesundheitsland Schleswig-Holstein - Jahrbuch 2010/2011, Ministerium für Arbeit, Gesundheit und Soziales Schleswig-Holstein, 2011.
- [16] Peter Rothenpieler, Harald Zwingelberg, Darren Carlson, Andreas Schrader, Stefan Fischer. *Datenschutz im AAL Service System SmartAssist*, In 4 Deutscher AAL-Kongress 'Innovative Assistenzsysteme im Dienste des Menschen - Von der Forschung für den Markt' (AAL 2011), Januar 2011.
- [17] Andreas Schrader, Darren Carlson, Peter Rothenpieler. *SmartAssist - Wireless Sensor Networks for Unobtrusive Health Monitoring*. In Proceedings of the 5th BMI Workshop on Behaviour Monitoring and Interpretation, volume 678, KI'2010, Karlsruhe, Mai 2010.
- [18] Peter Rothenpieler, Claudia Becker, Stefan Fischer. *Privacy concerns in a remote monitoring and social networking platform for assisted living*, In Sixth International PrimeLife/IFIP Summer School on Privacy and Identity Management for Emerging Internet Applications throughout a Person's Lifetime, Springer-Verlag, 2010.
- [19] Andreas Hein, Andreas Schrader. *Ambient Assisted Living (AAL 2009) - Gestaltung altersgerechter Lebenswelten mit IuK-Technologien*, In Proceedings of INFORMATIK 2009, Im Focus das Leben (S.; Maehle E.; Reischuk R. Fischer, ed.), Lecture Notes in Informatics (LNI), September 2009.

2. Vorträge

- [20] Andreas Schrader. *SmartAssist - Platform for the support of social and health aspects in an age appropriate autonomous life*. AAL Conference Hamburg, Innovation Norway, Nov. 27-28, 2012.
- [21] Andreas Schrader. *Welche Unterstützung kann moderne Technik bieten? Weiter wie gewohnt? Wohnen im Alter*. Regionalkonferenz Schleswig-Holstein, LVGSHS, September 26th, 2012.

- [22] Andreas Schrader. Hilfreiche Technik für Menschen mit Behinderungen. 27. Sitzung des VIII Beirates für Menschen mit Behinderungen (Kiel), November 4th, 2011.
- [23] Andreas Schrader. Eigenständigkeit im Alter – die Technik hilft. Med-Com'2011, Bad Bramstedt, Germany, May 18th, 2011.
- [24] Andreas Schrader. SmartAssist - Eine offene Systemplattform für Ambient Assisted Living. Zukunftsmarkt Assistenzsysteme im Dienste des Menschen, Medientage Nord 2010, Kiel, Germany, November 18th, 2010.
- [25] Peter Rothenpieler & Darren Carlson. SmartAssist - Möglichkeiten und Techniken für Drittanbieter. Zukunftsmarkt Assistenzsysteme im Dienste des Menschen, Medientage Nord 2010, Kiel, Germany, November 18th, 2010.
- [26] Andreas Schrader. SmartAssist - Smart Assistance with Ambient Computing. AAL Forum 2010, Odense, Denmark, September 16, 2010.
- [27] Andreas Schrader. The SmartAssist Research Project. 3rd AAL Kongress 2010, Congress Center, Berlin, Germany, January 27, 2010.
- [28] Andreas Schrader. Ambient Assisted Living - Länger selbständig leben im Alter. Abenteuer Informatik. Exhibition at the GI Informatik 2009, Lübeck, September 29, 2009.

3. Internationale Preise

- [29] Best Demo Award! Darren Carlson, Bashar Altakrouri & Andreas Schrader. Reinventing the Share Button for Physical Spaces. In: Percom'2013, Demo Session at IEEE International Conference on Pervasive Computing and Communication (PerCom 2013), San Diego, California, USA, March 18-22, 2013.
- [30] Winner of IoT Challenge Competition! Darren Carlson, Bashar Altakrouri & Andreas Schrader. AmbientWeb: Bridging the Web's Cyber-physical Gap. 3rd International Conference for Industry and Academia (IoT'2012), Wuxi, China, Oct. 24-26, 2012.

4. Studentische Arbeiten

- [31] Martin Bomba, BikeWars: Entwicklung einer AAL-Anwendung für die Smart-Assist Plattform, B.Sc., Institut für Telematik, Universität zu Lübeck, Juni 2013
- [32] Daniel Burmeister, Aktivitätserkennung für Ambient Health Monitoring, M.Sc., Institut für Telematik, Universität zu Lübeck, Januar 2013
- [33] Manuel Schrick, Ambient Health Feedback System für Android, B.Sc., Institut für Telematik, Universität zu Lübeck, Dezember 2012
- [34] Richard Stück, Context-aware Device and Access Rights Management for Smart Homes in Ambient Assisted Living, M.Sc., November 2012
- [35] Ibrahim Awada, Long term tremor assessment using mobile devices, M.Sc., Institut für Telematik, Universität zu Lübeck, November 2012
- [36] Christoph Sieb, Mobile Device Shake Compensation, M.Sc., Institut für Telematik, Universität zu Lübeck, Februar 2012

- [37] Michael Kaluza, Medication Support System, M.Sc., Institut für Telematik, Universität zu Lübeck, Januar 2012
- [38] Reneé Ahrens, Augmented Reality in Ambient Assisted Living, M.Sc., Institut für Telematik, Universität zu Lübeck, Dezember 2011

5. Events

- [39] Am 29. September 2009 wurde im Zusammenhang der GI-Jahrestagung Informatik 200 eine Ausstellung *Abenteuer Informatik* in Lübeck organisiert. Prof. Schrader präsentierte das Projekt mit dem Vortrag „Ambient Assisted Living – Länger selbständig leben im Alter“.
- [40] Am 27. Januar 2010 hat Prof. Schrader das Projekt auf dem 3. Deutschen AAL-Kongress in Berlin mit einem Kurzvortrag vorgestellt.
- [41] Am 9. Juni 2010 hat Peter Rothenpieler das Projekt auf dem Doktorandentag der Universität zu Lübeck in der St. Petri-Kirche in Lübeck mit dem Poster „SmartAssist – Dienstplattform für mehr Autonomie und Sicherheit im Alter“ vorgestellt.
- [42] Am 16. Juni 2010 wurde in Zusammenarbeit mit dem Unabhängigen Landeszentrum für Datenschutz Schleswig-Holstein ein Workshop mit dem Titel „Datenschutz in AAL-Projekten“ abgehalten. Aus dem Workshop ist eine gemeinsame Publikation für den 4. Deutschen AAL-Kongress entstanden [16]
- [43] Am 16. September 2010 hat Prof. Schrader auf dem AAL Forum 2010 in Odense einen Vortrag „SmartAssist – Smart Assistance with Ambient Computing“ gehalten (Workshop „Northern European AAL projects and business cases“), siehe <http://www.aalforum.eu/page/track-e-aal-the-viking-way>.
- [44] Am 30. Oktober 2010 hat Prof. Schrader auf dem Tag der Wissenschaft „Hanse trifft Humboldt“ in der Universitäts-Kirche St. Petri in Lübeck den Vortrag „SmartLübeck – Visionen für eine intelligent vernetzte Stadt“ gehalten.
- [45] Am 18. November 2010 wurden im Rahmen des Workshops „Zukunftsmarkt Assistenzsysteme im Dienste des Menschen“ der MediaTage Nord zwei Vorträge über SmartAssist gehalten („SmartAssist - Eine offene Systemplattform für Ambient Assisted Living“ und „SmartAssist - Möglichkeiten und Techniken für Drittanbieter“), siehe <http://www.mediatage-nord.de/index.php?id=244>.
- [46] Am 18. Mai 2011 hat Prof. Schrader das Projekt SmartAssist auf dem MedCom‘2011 Kongress in Bad Bramstedt mit dem Vortrag „Eigenständigkeit im Alter – die Technik hilft“ vorgestellt.
- [47] Am 4. November 2011 hat Prof. Schrader das Projekt auf der 27. Sitzung des VIII Beirates für Menschen mit Behinderungen in Kiel mit dem Vortrag „Hilfreiche Technik für Menschen mit Behinderungen“ vorgestellt.
- [48] Am 26. September 2012 hat Prof. Schrader das SmartAssist auf der Regional-konferenz „Weiter wie gewohnt? Wohnen im Alter“ LVGSH in Lübeck mit dem Vortrag „Welche Unterstützung kann moderne Technik bieten?“ vorgestellt.

- [49] Am 27. November 2012 hat Prof. Schrader das Projekt auf dem AAL Conference Hamburg der Innovation Norway mit dem Vortrag „SmartAssist – Plattform fort he support of social and health aspects in an age appropriate autonomous life“ vorgestellt.

6. Presseberichte

- [50] Michael Hollinde. Uni-Forscher in San Diego geehrt. Lübecker Nachrichten, 03.04.2013.
- [51] HL-Live. Lübecker Informatik-Projekte international ganz vorn, <http://www.hl-live.de>, 08.11.2012.
- [52] Lübecker Nachrichten. Uni-Informatik-Projekte international ganz vorn. Lübecker Nachrichten, 07.11.2012.
- [53] Michael Hollinde. Die technisch unterstützte Nachbarschaftshilfe im Alter. Lübecker Nachrichten, 21.12.2011.
- [54] Günter Knackfuß. Warum Wohnungen "Mitaltern" sollen. ReiseTravel, Reise, Moden, Lifestyle, 2010, Nov. 24.
- [55] DiWiSH. Altersgerechte Assistenzsysteme sind ein Zukunftsmarkt. DiWiSH, Digitale Wirtschaft Schleswig-Holstein, Clustermanagement, News, 2010, Nov. 22.
- [56] Heike Stüben. Roboter in der Pflege: Chance oder Gefahr? Kieler Nachrichten, 2010, Nov. 13.
- [57] Prof. Jens Martin Träger. Smart Assist: Älter werden, aber zuhause leben. Nordlicht, Nr. 4/2010, pp. 34-35.
- [58] Olaf Berger. Telemedizin ermöglicht langes Verbleiben im eigenen Heim. Hamburger Abendblatt, 2009, April 11.
- [59] Philipp Grätzel von Grätz. Intelligente Haustechnik steht vor dem Durchbruch. Ärzte Zeitung, 2009, Feb. 04.

IV. Quellen

- [60] Birgid Eberhardt, „Zielgruppen für AAL-Technologien und –Dienstleistungen“, AAL Kongress- und Fachbeiträge der AG Kommunikation der BMBF/VDE Innovationspartnerschaft AAL, 2009, http://www.vde.com/de/Technik/AAL/Publikationen/Kongress-undFachbeitraege/documents/zielgruppen%20f%C3%BCr%20aal%20_tabelle_.pdf
- [61] Peter Georgief, „Ambient Assisted Living - Marktpotenziale IT-unterstützter Pflege für ein selbstbestimmtes Altern“. FAZIT-Forschung, MFG Stiftung Baden-Württemberg, ISSN 1861-5066, 2008

- [62] Dr. Thomas Becks, Johannes Dehm, Birgid Eberhardt, „Neue „intelligente“ Assistenzsysteme für Prävention, Homecare und Pflege“, VDE Initiative Mikro-Medizin, Frankfurt am Main, 2007
- [63] D.M. Boyd / N.B. Ellison, “Social Network Sites: Definition, History, and Scholarship”. *Journal of Computer-Mediated Communication* Vol.13 Issue 1, UC Berkeley / Michigan State University, 2007
- [64] Kaplan Andreas M., Haenlein Michael, ”Users of the world, unite! The challenges and opportunities of social media”, *Business Horizons*, Vol. 53, Issue 1, p.59-68, 2010
- [65] Schmidt, Wolf, Klein, Balfanz, “SOPRANO Ambient Middleware: Eine offene, flexible und marktorientierte semantische Dienstplattform für Ambient Assisted Living”, *Ambient Assisted Living 2009*, 2. Deutscher AAL-Kongress mit Ausstellung / Technologien - Anwendungen / 27.- 28. Januar 2009 in Berlin
- [66] Ingo Melzer et al., “Service-Orientierte Architekturen mit Web Services”, 3. Auflage, Spektrum-Verlag, ISBN: 978-3-8274-1993-4, 2008
- [67] M. Rom, et al., “Gaia: a middleware platform for active spaces,” *Mobile Computing and Communications Review*, SIGMOBILE, vol. 6, no. 4, 2002, pp. 65-67.
- [68] N.B. Priyantha, A. Chakraborty and H. Balakrishnan, “The cricket location-support system,” *Proc. Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, Mobicon2000, ACM Press, New York, NY, USA, 2000, pp. 32-43.
- [69] C. Endres, A. Butz and A. MacWilliams, “A Survey of Software Infrastructures and Frameworks for Ubiquitous Computing,” *Mobile Information Systems*, vol. 1, no. 1, 2005, pp. 41-80.
- [70] N. Davies and H.-W. Gellersen, “Beyond Prototypes: Challenges in Deploying Ubiquitous Systems,” *Pervasive Computing*, vol. 1, no. 1, 2002, pp. 26 - 35.
- [71] M. Blackstock, R. Lea and C. Krasic, “Toward Wide Area Interaction with Ubiquitous Computing Environments,” *Smart Sensing and Context*, Springer Berlin/Heidelberg, 2006, pp. 113-127.
- [72] E. Niemelä and J. Latvakoski, “Survey of Requirements and Solutions for Ubiquitous Software,” *Proc. Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia* ACM Press, New York, NY, USA, 2004, pp. 71-78.
- [73] B.N. Schilit, N. Adams and R. Want, “Context-Aware Computing Applications,” *Proc. Proceedings of the Workshop on Mobile Computing Systems and Applications*, IEEE Computer Society, 1994, pp. 85-90.
- [74] J.E. Bardram, “The Java Context Awareness Framework (JCAF) – A Service Infrastructure and Programming Framework for Context-Aware Applications,” *Proc. 2005 Conference on Pervasive Computing*, 2005.
- [75] K. Henriksen, J. Indulska, T. McFadden and S. Balasubramaniam, “Middleware for distributed context-aware systems,” *Proc. International Symposium on Distributed Objects and Applications*, DOA 2005.
- [76] W.G. Griswold, R. Boyer, S.W. Brown and T.M. Truong, “A component architecture for an extensible, highly integrated context-aware computing infrastruc-

- ture,” Proc. 25th International Conference on Software Engineering, IEEE Computer Society, 2003, pp. 363-372.
- [77] P.K. McKinley, S.M. Sadjadi, E.P. Kasten and B.H.C. Cheng, “Composing Adaptive Software,” *Computer*, vol. 37, no. 7, 2004, pp. 56-64.
- [78] D.V. Carlson, A. Schrader and D. Busch, “Modular framework support for context-aware mobile cinema,” *Personal and Ubiquitous Computing*, vol. 12, no. 4, 2008, pp. 299-306.
- [79] Darren V. Carlson, *Aladin - An Extensible Ubiquitous Computing Infrastructure*, Masterarbeit, Universität zu Lübeck, 2005.
- [80] Guido Heumer, Darren Carlson, Bernhard Jung and Andreas Schrader. *Paranoia Syndrome - A Pervasive Multiplayer Game*. In: Röcker, C. and Magerkurth, C. (Hrsg.): *Pervasive Games – Applications*, Shaker Verlag, 2007.
- [81] OSGi Alliance, “OSGi Service Platform Release 4,” 2011; <http://www.osgi.org/Specifications/HomePage>.
- [82] C. Jernigan, C. Bayley, J. Lin and C. Wright, “The Locale Project,” 2008; <http://www.twofortyfouram.com/>.
- [83] B.v. Wissen, N. Palmer, R. Kemp, T. Kielmann and H. Bal, “ContextDroid: an expression-based context framework for Android.,” Proc. International Workshop on Sensing for App Phones (PhoneSense), 2010.
- [84] D. Salber, A.K. Dey and G.D. Abowd, “The Context Toolkit: Aiding the development of context-enabled applications,” Proc. Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit, ACM Press, New York, NY, USA, 1999, pp. 434-441.
- [85] A. Nadav, P. Wei, I. Cory, K. Inas and P. Alex, “The social fMRI: measuring, understanding, and designing social mechanisms in the real world,” Proc. Proceedings of the 13th international conference on Ubiquitous computing % @ 978-1-4503-0630-0, ACM, 2011, pp. 445-454.
- [86] IST-MUSIC Consortium, “IST-MUSIC: Context-aware self-adaptive platform for mobile applications,” 2010; <http://ist-music.berlios.de>.
- [87] C. Ming-Chiao, C. Jiann-Liang and C. Teng-Wen, “Android/OSGi-based vehicular network management system,” *Comput. Commun.*, vol. 34, no. 2, 2010, pp. 169-183.
- [88] M. Kuna, H. Kolaric, I. Bojic, M. Kusek and G. Jezic, “Android/OSGi-based Machine-to-Machine context-aware system,” Proc. Proceedings of the 2011 11th International Conference on Telecommunications (ConTEL), IEEE, 2011, pp. 95 - 102.
- [89] G.D. Abowd, et al., “Cyberguide: a mobile context-aware tour guide,” *Wireless Networks*, vol. 3, no. 5, 1997, pp. 421-433.
- [90] C. Keith, D. Nigel, M. Keith, F. Adrian and E. Christos, “Developing a context-aware electronic tourist guide: some issues and experiences,” Proc. SIGCHI conference on Human factors in computing systems, ACM, 2000, pp. 17-24.
- [91] T. Kindberg, et al., “People, places, things: web presence for the real world,” *Mobile Networks and Applications*, vol. 7, no. 5, 2002, pp. 365-376.

- [92] G.J.F. Jones and P.J. Brown, "Context-Aware Retrieval for Ubiquitous Computing Environments," *Mobile and ubiquitous information access*, Springer Lecture Notes in Computer Science, 2004, pp. 227--243.
- [93] P.J. Brown, J.D. Bovey and C. Xian, "Context-Aware Applications: From the Laboratory to the Marketplace," *IEEE Personal Comm*, vol. 4, no. 5, 1997, pp. 58–64.
- [94] D.L.d. Ipiña, J.I. Vazquez and J. Abaitua, "A Context-Aware Mobile Mash-up Platform for Ubiquitous Web," *Proc. 3rd IET Int'l Conf. Intelligent Environments, Inst. Eng. and Tech.*, 2007, pp. 116–123.
- [95] P. Coppola, et al., "The Context-Aware Browser," *IEEE Intelligent Systems*, vol. 25, no. 1, 2010, pp. 38 - 47.
- [96] J. Roth, "Context-Aware Web Application Using The Pinpoint Infrastructure," *Proc. IADIS International Conference WWW/Internet*, 2002.
- [97] J. Noltes, "An architecture for context-aware mobile web browsing," *Proc. 9th TSCONIT*, 2008.
- [98] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01, San Francisco, CA, USA, 2001, pp. 282–289. [Online]. Available: <http://dl.acm.org/citation.cfm?id=645530.655813>
- [99] A. McCallum, D. Freitag, and F. C. N. Pereira, "Maximum entropy Markov models for information extraction and segmentation," in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML '00, San Francisco, CA, USA, 2000, pp. 591–598. [Online]. Available: <http://dl.acm.org/citation.cfm?id=645529.658277>
- [100] G G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, and J. Stankovic. An advanced wireless sensor network for health monitoring. In *Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2)*, Apr. 2006.
- [101] Gupta GS, Mukhopadhyay SC, Sutherland M, Demidenko S. Wireless Sensor Network for Selective Activity Monitoring in a home for the Elderly, *Proceedings of 2007 IEEE IMTC conference*. Poland, Warsaw 2007; 1(3): 1-6.
- [102] Harvard University. CodeBlue project: Wireless Sensor Networks for Medical Care. <http://www.eecs.harvard.edu/~mdw/proj/codeblue/>
- [103] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Feb 1989, vol. 77, no. 2, pp. 257-286.
- [104] P. Chung and C. Liu, "A daily behavior enabled hidden Markov model for Human behavior understanding", *Pattern Recognition* vol. 41, no. 5, pp1589-1597, 2008.
- [105] S. W. Lee, Y. S. Kim and Z. Bien, "A nonsupervised learning framework of human behavior patterns based on sequential actions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no4, pp. 479-492, 2010.
- [106] K. Park, Y. Lin, V. Metsis, Z. Le and F. Makedon, "Abnormal human behavioral pattern detection in assisted living environments", *Proceedings of the International Conference on Pervasive Technol. Relat. Assist. Environ.*, 2010.

- [107] A. A. Sodemann, M. P. Ross and B. J. Borghetti, "A Review of Anomaly Detection in Automated Surveillance", *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, 2012.
- [108] Andreas, F. and Gustavo, A. A Dynamic Lightweight Platform for Ad-Hoc Infrastructures Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications, IEEE Computer Society, 2005.
- [109] Bardram, J.E. The Java Context Awareness Framework (JCAF) – A Service Infrastructure and Programming Framework for Context-Aware Applications, Pervasive 2005, Munich, Germany, 2005, 98-115.
- [110] Biegel, G. and Cahill, V. A Framework for Developing Mobile, Context-aware Applications Pervasive Computing and Communications, Orlando, Florida, USA, 2004.
- [111] Chen, H. An Intelligent Broker Architecture for Context-Aware Systems, University of Maryland, Baltimore, USA, 2003.
- [112] Cristiano Andre da, C., Adenauer Correa, Y. and Claudio Fernando Resin, G. 2008. Toward a General Software Infrastructure for Ubiquitous Computing. *IEEE Pervasive Computing*, 7 (1). 64-73.
- [113] Daniel, S., Anind, K.D. and Gregory, D.A. The Context Toolkit: Aiding the Development of Context-enabled Applications. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: the CHI is the limit, ACM, Pittsburgh, Pennsylvania, United States, 1999.
- [114] David, G., Dan, S., Asim, S. and Peter, S. 2002. Project Aura: Toward Distraction-Free Pervasive Computing. *IEEE Pervasive Computing*, 1 (2). 22-31.
- [115] Edwards, W.K. and Rebecca, E.G. At Home with Ubiquitous Computing: Seven Challenges Proceedings of the 3rd International Conference on Ubiquitous Computing, Springer-Verlag, Atlanta, Georgia, USA, 2001.
- [116] Endres, C., Butz, A. and MacWilliams, A. 2005. A Survey of Software Infrastructures and Frameworks for Ubiquitous Computing. *Mobile Information Systems*, 1 (1). 41 - 80.
- [117] Fox, A., Johanson, B., Hanrahan, P. and Winograd, T. 2000. Integrating Information Appliances into an interactive Workspace. *IEEE Computer Graphics and Applications*, 20 (3). 54-65.
- [118] Gregory, D.A., Anind, K.D., Peter, J.B., Nigel, D., Mark, S. and Pete, S. Towards a Better Understanding of Context and Context-Awareness Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing, Springer-Verlag, Karlsruhe, Germany, 1999.
- [119] Jeronimo, M. and Weast, J. UPnP Design by Example: A Software Developer's Guide to Universal Plug and Play. Intel Press, 2003.
- [120] Jim, W. 1999. The Jini Architecture for Network-centric Computing. *Commun. ACM*, 42 (7). 76-82.
- [121] Korpipaa, P., Mantyjarvi, J., Kela, J., Keranen, H. and Malm, E.-J. 2003. Managing Context Information in Mobile Devices. *IEEE Pervasive Computing*, 02 (03). 42-51.

- [122] Manuel, R., Christopher, H., Renato, C., Anand, R., Roy, H.C. and Klara, N. 2002. Gaia: A Middleware Platform for Active Spaces. SIGMOBILE Mob. Comput. Commun. Rev., 6 (4). 65-67.
- [123] Roy, W., Andy, H., Veronica, F. and Jonathan, G. 1992. The Active Badge Location System. ACM Trans. Inf. Syst., 10 (1). 91-102.
- [124] Roy, W. and Trevor, P. System Challenges for Ubiquitous & Pervasive Computing Proceedings of the 27th International Conference on Software Engineering, ACM, St. Louis, MO, USA, 2005.
- [125] Steinberg, D.H. and Cheshire, S. Zero Configuration Networking: The Definitive Guide. O'Reilly Media, Inc., 2005.
- [126] Strang, T. and Linnhoff-Popien, C. A Context Modeling Survey Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham, England, 2004.
- [127] Thomas, H., Wieland, S., Mario, P., Gerhard, L., Josef, A. and Werner, R. Context-Awareness on Mobile Devices - the Hydrogen Approach Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 9 - Volume 9, IEEE Computer Society, 2003.
- [128] Thomson, G., Nixon, P. and Terzis, S., Towards Adhoc Situation Determination. in The Sixth International Conference on Ubiquitous Computing (UbiComp2004). Proceedings of the First International Workshop on Advanced Context Modelling, Reasoning and Management, (2004).
- [129] Want, R., Schilit, B.N., Adams, N.I., Gold, R., Petersen, K., Goldberg, D., Ellis, J.R. and Weiser, M. 1995. The ParcTab Ubiquitous Computing Experiment. IEEE Personal Communications, 2 (6). 28-43.
- [130] Medical WSN System of the Computer Science Department (UVA), Available: <http://www.cs.virginia.edu/wsn/medical>
- [131] A. Krause, A. Smailagic, D. P. Siewiorek, "Context-Aware Mobile Computing: Learning Context-Dependent Personal Preferences from a Wearable Sensor Array", IEEE Transactions on Mobile Computing, vol. 5, no. 2, pp. 113-127, Feb. 2006.