



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Seminararbeit

Dennis Hollatz

Managing Information –
Personal Information Environments
auf Basis von iROS

Dennis Hollatz
Managing Information –
Personal Information Environments
auf Basis von iROS

Seminararbeit
eingereicht im Rahmen der Veranstaltung Seminar/Ringvorlesung
im Studiengang Informatik (Master of Science)
am Studiendepartment Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Kai von Luck

Abgegeben am 3. März 2008

Inhaltsverzeichnis

Abbildungsverzeichnis	4
1 Einführung	5
1.1 Zielsetzung und Motivation	5
1.2 Gliederung	5
2 Gerätekommunikation im persönlichen Umfeld	6
2.1 Personal Information Environments	6
2.2 Szenario	7
2.3 Schaffung der Infrastruktur	8
2.3.1 Allgemeine Annahmen zur Umgebung	8
2.3.2 Event Manager	9
2.3.3 Agentenbasierte Adapter	10
2.4 Absicherung des Nachrichtenverkehrs	11
2.4.1 Auswirkungen auf die Architektur	11
2.4.2 Verschlüsselung	12
2.4.3 Authentifizierung	13
2.5 Alternative Möglichkeiten	13
3 Vorgehen in der Master-Arbeit	14
3.1 Design des Event Managers	14
3.2 Design der Service-Adapter	15
3.3 Design des Simulationsservice	15
4 Resümee	16
4.1 Zusammenfassung	16
4.2 Chancen und Risiken	16
4.3 Ausblick	17
Literaturverzeichnis	18

Abbildungsverzeichnis

2.1	Gerätevielfalt	7
2.2	Konzeptionelle Architektur	10

1 Einführung

1.1 Zielsetzung und Motivation

Die in [Dreyer \(2008\)](#) formulierte Version einer intelligenten Wohnumgebung erfordert eine verlässliche und stabile Architektur, die die Interaktion der partizipierenden Geräte unterstützt und absichert. Der in [Hollatz \(2007\)](#) vorgestellte und in [Hollatz \(2008b\)](#) praktisch evaluierte *EventHeap* stellt hierfür eine gute Grundlage dar. Für den produktive Einsatz in einer *Personal Information Environment* ([Pierce und Nichols, 2007](#)) erfordert es jedoch noch einiger Erweiterungen auf der Server-Seite. Dieses stellt den ersten Teil dieser Arbeit dar.

Der zweite Teil soll sich mit der Schaffung eines Frameworks für die Anbindung von Clients in diese Umgebung beschäftigen. Der *Event Manager* stellt zwar die Grundlage der Kommunikation, ist allerdings auch sehr einfach gehalten und bietet Clients vielfältige Möglichkeiten die Kommunikation zu implementieren. Ziel dieses Teils der Arbeit soll sein diese Vielfältigkeit in eine einfach zu implementierende Lösung zu überführen, die den Anforderungen der Umgebung angepasst ist, die Vielfältigkeit der möglichen Geräte allerdings nicht einschränkt.

1.2 Gliederung

Die Arbeit gliedert sich fünf Kapitel. Das vorliegende [Kapitel 1](#) gibt einen kurzen Überblick über die Motivation, Zielsetzung und den Aufbau der Arbeit.

[Kapitel 2](#) beschreibt die Idee einer integrierten Kommunikationsinfrastruktur für *Ambient Intelligence* und ihre mögliche Umsetzung in einer Masterthesis. Im anschließenden [Kapitel 3](#) werden die konkreten Ideen zur Umsetzung dargelegt und im letzten [Kapitel 4](#) folgt eine Bewertung des vorgestellten Vorhabens.

2 Gerätekommunikation im persönlichen Umfeld

Mark Weiser beschreibt in seiner Vision allgegenwärtiger Computersysteme ([Weiser, 1991](#)) einen Zustand, den wir heute nahezu erreicht haben. Die meisten von uns haben besitzen mehr als einen Computer, beispielsweise in Form eines PCs, Notebooks, MP3-Players oder Mobiltelefons. In [Winograd und Flores \(1986\)](#) wird diese Entwicklung bereits angedeutet. Sie beziehen hier jedoch mehr auf die wachsende Leistungsfähigkeit der Computer, die immer mehr Aufgaben im persönlichen Umfeld übernehmen. Es werden hier die Punkte *readiness to hand*, *social embedding*, sowie der Einsatz des Computers zur Unterstützung Entscheidungsfindung herausgestellt (s.a. [Hollatz \(2007\)](#)).

2.1 Personal Information Environments

Die vermehrte Nutzung und Vernetzung von Computern im persönlichen Umfeld birgt allerdings auch eine Reihe von Hürden, die es zu bewältigen gilt. So ist in einer solch heterogenen Ansammlung von Geräten nicht jedes Gerät für jede Form von Daten geeignet. Beispielsweise ist ein Mobiltelefon mit kleinem Display nicht dazu geeignet, einen umfangreiche Zeitungsartikel zu lesen. Für eine solche Aufgabe ist die Ausgabe auf einem großen Bildschirm besser geeignet. Das Handy ist allerdings sehr gut geeignet eine Anfrage, einen solchen Artikel zu empfangen und an ein geeignetes Gerät weiterzuleiten. Sollen dem Nutzer wiederum Bilder oder Videos gesendet werden, so kann das Telefon sehr gut für den Empfang geeignet sein. Die Empfangenen Daten können so unterwegs auf dem Telefon betrachtet und bei Ankunft zuhause, auch auf einem großen Bildschirm angezeigt werden. Diese Ideen stützen sich auf ein Konzept, dass in [Pierce und Nichols \(2007\)](#) als *Personal Information Environment* beschrieben wird.

[Abbildung 2.1](#) zeigt ein paar typische Arbeitsplätze. jeder von ihnen verfügt über mehrere Bildschirme und Rechner. Viele Menschen besitzen heute mehr als ein intelligentes elektronisches Gerät. So ist es nicht unüblich, dass eine Person sowohl einen PC, ein Notebook, ein Mobiltelefon und häufig noch einen MP3-Player besitzt. Ziel der Schaffung von *Personal Information Environments* ist die nahtlose Vernetzung aller Geräte, die ein Nutzer im Zugriff



Abbildung 2.1: Gerätevielfalt
(Quelle: [Pierce und Nichols \(2007\)](#))

hat. Ein Gerät soll Kenntnis darüber besitzen, ob und welche Geräte sich außer ihm noch in seiner Reichweite befinden und auf welche von ihnen es Zugriff hat. *Reichweite* bezieht sich dabei auf die Möglichkeit des Zugriffs über unterschiedlichste Wege, wie z. B. Bluetooth, WLAN oder das Internet.

Praktisch gesehen handelt es sich bei diesem System um einen *Instant-Messaging-Dienst* für Geräte. Die einzelnen Geräte haben über diesen Dienst die Möglichkeit untereinander zu kommunizieren. Über diesen Dienst wird es dem Benutzer ermöglicht Dokumente über mehrere Geräte hinweg zu verteilen und zu synchronisieren, entfernte Bildschirmausgaben zu steuern oder Nachrichten zu verschicken.

2.2 Szenario

Das Szenario *Your Home in Your Hand* ([Dreyer, 2008](#)) beschreibt einen Wohnraum, für den solch eine *Personal Information Environment* eine Grundlage bilden kann. Es illustriert eine interaktive Wohnung, in der sämtliche Gerätedienste miteinander kommunizieren können. So bemerkt die Wohnung beispielsweise die Ankunft eines angemeldeten Nutzers und kann ihm, sofern es der Kontext erlaubt, seine E-Mails auf dem großen Fernsehschirm präsentieren. Ein Telefonat wird in der Küche auf den dort installierten Lautsprechern wiedergegeben, so dass die Zubereitung einer Mahlzeit nicht unterbrochen werden muss. Ein unerwarteter

Besuch wird zunächst per Video auf dem Fernseher angezeigt, die aktuell laufende Sendung wird währenddessen automatisch via Timeshift aufgezeichnet und kann so zu einem späteren Zeitpunkt an der Stelle der Unterbrechung fortgesetzt werden. Es handelt sich hierbei um Berührungspunkte zum Gebiet der *Context Awareness*. Für eine einleitende Erläuterung des Themas siehe [Dey \(2001\)](#).

Als Steuergerät wird hier beispielsweise das Mobiltelefon genannt, das über eine Anbindung über WLAN oder Bluetooth selbst ein Teil der Umgebung werden kann. Die Kommunikation der Geräte muss nicht nur innerhalb der Wohnung stattfinden. Es ist auch denkbar, dass über einen öffentlichen Zugangspunkt – wie z. B. über das Internet – ein Abrufen von von Informationen aus oder die Steuerung von Geräten in der Wohnung denkbar ist.

2.3 Schaffung der Infrastruktur

Dieser Abschnitt beschreibt die Anforderungen an die Infrastruktur in einer *Ambient Intelligence Environment*. Es wird ausgehend von einigen allgemeinen Annahmen erläutert, wie sich die Kommunikationsinfrastruktur und die Clients in dieser Umgebung gestalten. Zusätzlich bezieht sich ein kurzer Abschnitt auf das Thema Sicherheit, welches beim Design der Umgebung nicht außer acht gelassen werden sollte.

2.3.1 Allgemeine Annahmen zur Umgebung

Dieser Abschnitt beschreibt einige allgemeine Annahmen, die bezüglich der Umgebung getroffen werden können. Das beschriebene Szenario soll so auf eine allgemeine technologische Basis gestellt werden können, von der ausgehend Architekturentscheidungen für das weitere vorgehen festgelegt werden können.

Die einsetzbaren Geräte lassen sich in verschiedene Geräteklassen unterteilen. Kategorien für eine Einteilung sind z. B.

- Bildschirmgröße,
- Öffentlichkeit,
- verfügbarer Speicher,
- Rechenleistung,
- Mobilität oder
- die allgemeine Verfügbarkeit des Geräts.

Die *Bildschirmgröße* ist entscheidend, wenn es um den Umfang der darzustellenden Informationen geht. Ein großes Display kann diese in viel höherem Umfang noch übersichtlich darstellen, während ein kleines Display, z.B. das eines Mobiltelefons, hier von einer Reduzierung der Informationen profitieren würde. Ebenso entscheidend für diese Faktoren ist die *Öffentlichkeit* eines Geräts. Diese hängt nicht nur vom Standort des Geräts ab, sondern ist auch abhängig von den anwesenden Personen. Ein privates Telefonat soll höchstwahrscheinlich nicht automatisch auf den Lautsprechern im Wohnzimmer wiedergegeben werden, wenn ich Besuch habe.

Der *verfügbare Speicher* und die *Rechenleistung* sind entscheidend, wenn es um den Empfang und das Ablegen von Dateien geht. So ist es sicherlich nicht immer sinnvoll, alle Fotos eines Nutzers auf seinem Mobiltelefon abzulegen. Wenn er allerdings seine Fotos vom letzten Urlaub seinen Freunden zeigen möchte, so ist das Mobiltelefon sicherlich ein geeignetes Transportmedium, da es die *Mobilität* besitzt, die feste Umgebung meiner Wohnung zu verlassen. Dieser Punkt stellt einen weiteren Aspekt einer *Personal Information Environment*: Die *Verfügbarkeit*. Geräte sind nicht immer verfügbar und es ist nicht möglich zu jeder Zeit auf jedes Gerät zuzugreifen. Es ist zwar davon auszugehen, dass verfügbarer Speicher und Rechenleistung auf mobilen Geräten immer weiter zunehmen werden, allerdings werden sie mittelfristig nicht die Leistungsfähigkeit aktueller PCs erreichen.

Die Architektur soll auf der Basis einer zentralen Kommunikationskomponente ruhen. Diese soll im folgenden als *Event Manager* bezeichnet werden. Die Struktur des Systems entspricht der bereits im Masterprojekt entworfenen Architektur zur Kommunikation der Komponenten des *amrt:shelf* (Hollatz, 2008b; Meißner, 2008; Urich, 2008). Die grundlegende Gliederung in einzelne Komponenten ist in [Abbildung 2.2](#) dargestellt. Die gezeigte konzeptionelle Architektur entspricht der dort vorgestellten konzeptionellen Architektur des *smart:shelf*. Sie ist auch für das geplante System anwendbar, was sich aus der sehr allgemeinen Auslegung der Kommunikationsarchitektur des *smart:shelf* ergibt.

2.3.2 Event Manager

Bereits in [Hollatz \(2007\)](#) wurden die *iROS-Middleware* und insbesondere der *EventHeap* als zentraler Bestandteil dieser Middleware vorgestellt¹. Der *EventHeap* eignet sich grundlegend als Kommunikationsplattform für eine solche Umgebung. Die Arbeit im Masterprojekt ([Hollatz, 2008b](#)) hat gezeigt, dass es sich hierbei um eine stabile und funktionsfähige Plattform handelt.

Der *EventHeap* stellt die serverseitige Komponente und die Grundlage der Kommunikation. Wie die Clients diese Plattform nutzen ist ihnen vollständig selbst überlassen. Jeder Client

¹s. [Johanson \(2002\)](#) und [Johanson und Fox \(2004\)](#) für eine tiefer gehende Erläuterung

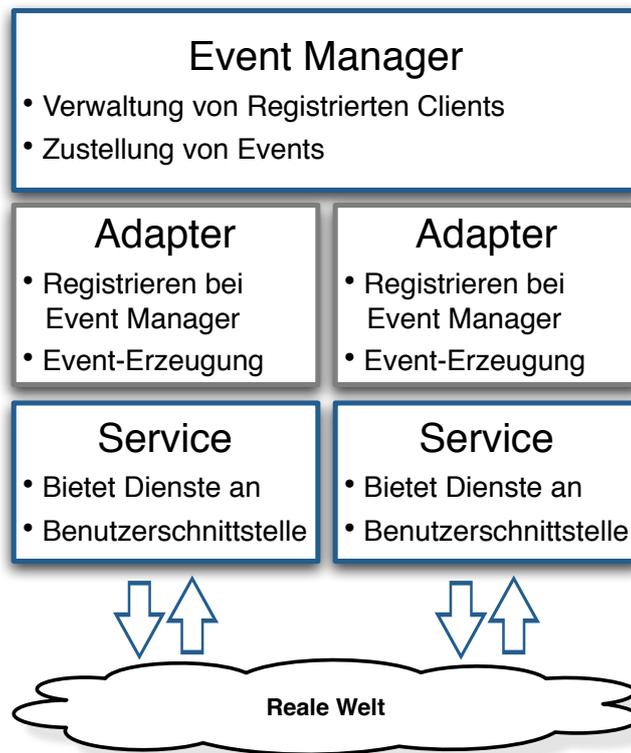


Abbildung 2.2: Konzeptionelle Architektur

steht selbst in der Verantwortung sich für Nachrichten zu registrieren (bzw. diese abzurufen) und aus ihrer semantischen Bedeutung Aktionen abzuleiten oder zu blockieren. Es wurde im Masterprojekt bereits ein Ansatz für eine standardisierte Schnittstelle für Clients geschaffen. Aufbauend auf den hier gesammelten Erfahrungen soll das Konzept der Adapter weiter verfeinert werden.

Die Aufgaben eines Event Managers bestehen in der Zwischenspeicherung und Weiterleitung von Nachrichten. Diese Aufgabe soll möglichst plattform- und implementierungsunabhängig geschehen. Mit steigendem Nachrichtenaufkommen rücken auch vermehrt Performance-Anforderungen in den Vordergrund, die bei einer späteren der Bewertung des Prototypen eine Rolle spielen werden.

2.3.3 Agentenbasierte Adapter

Aufbauend auf dieser Basis soll ein Rahmenwerk geschaffen werden, dass es Entwicklern erleichtert neue Geräte einfach in die bestehende Umgebung zu integrieren. In [Hel-](#)

lenschmidt (2005a) und Hellenschmidt (2005b) werden die Gerätedienste auch als Agenten bezeichnet. Unter der Annahme, dass es sich bei einer Ambient-Intelligence-Umgebung auch um ein Multiagentensystem handeln kann, lassen sich eine Reihe von Erkenntnissen aus diesen überführen. Es soll, basierend auf der Idee eines Multiagentensystems, eine Umgebung geschaffen werden, in der intelligente Geräte miteinander kommunizieren und interagieren können. Es sollen dabei die Erkenntnisse von Pierce und Nichols umgesetzt und erweitert werden. Dienste können sich über ihre Adapter für Events registrieren oder beim Manager nach konkreten Events fragen.

Während die in Hollatz (2008b) beschriebenen Adapter sehr wenig Entscheidungsfreiheiten haben und hauptsächlich den Zweck der Weiterleitung von Informationen zwischen dem Gerätedienst und dem *Event Manager* erfüllen, sind auch erheblich komplexere Adapter vorstellbar. Sie könnten basierend auf dem Konzept der BDI-Agenten² entsprechend der Umgebung, Events unterdrücken oder erzeugen. Ihren Zustand anderen Agenten mitteilen und so als kleiner Teil einer gesamten *Ambient Intelligence* dienen. Das Verhalten des Gesamtsystems wäre auf dieser Grundlage nicht explizit vorgegeben, sondern wäre aus dem Verhalten der einzelnen Agenten emergent. Diese Agenten können basierend auf JADEX implementiert und ausgeführt werden (s.a. Pokahr u. a. (2003) und Georgeff u. a. (1999)).

Auf Jadex basierende Adapter können so implementiert werden, dass sie untereinander kommunizieren können. Als Basis für diese Kommunikation kann auch der *Event Manager* dienen.

2.4 Absicherung des Nachrichtenverkehrs

In einer vollständig vernetzten Umgebung ist es essentiell wichtig, dass der Nachrichtenverkehr gegen Fremdeingriffe abgesichert ist. Es muss zu jeder Zeit sichergestellt sein, dass ausschließlich autorisierte Nutzer auf Bestandteile des Systems zugreifen können. In diesem Abschnitt soll erläutert werden, wie eine einfache Absicherung das System schützen kann.

2.4.1 Auswirkungen auf die Architektur

In der Grundvariante des Systems gibt es kein Sicherheitskonzept. Jeder Client kann sich für jede Eventart registrieren und künftig entsprechende Nachrichte empfangen. Es wird hierbei davon ausgegangen, dass die Clients untereinander wohlgesonnen sind und kein Missbrauch der Funktionalitäten stattfindet.

²BDI: Belief, Desire, Intention

Der geplante Austausch sensibler Daten in dieser Umgebung erfordert allerdings ein flexibles Sicherheitskonzept, das es erlaubt, auch anderen Nutzern den Zugang zu Teilen des Systems zu ermöglichen. So soll es z. B. möglich sein, dass Gäste auf einem Mobiltelefon mitgebrachte Fotos auch auf dem Fernseher wiedergeben können, jedoch keinen Zugang zu privaten E-Mails erhalten.

Es ist von Vorteil zusätzlich zu der in [Abschnitt 2.3.1](#) erfolgten Einteilung in Geräteklassen auch eine Einteilung der Geräte in *Vertrauensklassen* vorzunehmen. Diese Vertrauensklassen werden von dem jeweiligen Eingabegerät für entsprechende Ausgabegeräte definiert. Hierbei gibt es grundsätzlich zwei Klassifizierungskriterien:

1. Ist die Umgebung sicher?
2. Ist das Aus-/Eingabegerät sicher?

Beide Fragen hängen stark von der Einschätzung des Benutzers ab. Für gewöhnlich können Geräte, die dem Benutzer gehören, als sicher angesehen werden. Fremde oder öffentliche Geräte sind nur dann sicher, wenn der Benutzer dies ausdrücklich wünscht. Entsprechend verhält es sich mit der Umgebung. Die Heimumgebung des Benutzers wird bekannten Geräten generell vertraut, während fremde Umgebungen hier zunächst eine gesonderte Authentifizierung erfordern.

Wiederum ausgehend von der in [Pierce und Nichols \(2007\)](#) beschriebenen Umgebung sollen einzelne Geräte, wie z. B. das Mobiltelefon, auch als Eingabegeräte für nicht vertrauenswürdige andere Geräte dienen können.

2.4.2 Verschlüsselung

Es soll im Betrieb des Systems keine generelle Einschränkung im Nachrichtenverkehr geben. Benutzer sollen Nachrichten – auch teilweise – lesen können. Als Beispiel soll hier die öffentliche Darstellung von Terminen gelten. Bei der Koordination von Terminen zwischen mehreren Personen sollen nicht alle privaten Termine öffentlich dargestellt werden. Es ist hier lediglich interessant, wann und über welche Dauer bereits Termine liegen.

In [Imamura u. a. \(2002\)](#) wird ein Standard vorgestellt, das es erlaubt, XML-Dokumente partiell zu verschlüsseln. Eine solche partielle Verschlüsselung ist auch für die vorgeschlagene Infrastruktur geplant.

2.4.3 Authentifizierung

Die Verschlüsselung der Nachrichten und die bei – ausreichender Berechtigung des Geräts – folgende Entschlüsselung der Nachrichten erfordert eine zentrale Authentifizierungsinstanz gegenüber der sich die Nutzer und Geräte identifizieren müssen.

Eine Möglichkeit, diese Authentifizierung zu realisieren ist die Bereitstellung eines Kerberos-Dienstes ([MIT Kerberos consortium, 2007](#)). Dieser implementiert sowohl die Authentifizierung von Benutzern, als auch von Geräten.

2.5 Alternative Möglichkeiten

Eine weitere Möglichkeit wäre die Authentifizierung der Geräte und Benutzer direkt am EventHeap-Server. Dieser würde daraufhin bei der Verteilung der Nachrichten an die Clients, diesen nur die Feldinhalte übermitteln, die diese auch Empfangen dürfen.

Diese Lösung hätte den Vorteil, dass der Nachrichtenverkehr signifikant gesenkt werden könnte. Sie benötigt allerdings einen Server, der sich des Nachrichteninhalts bewusst ist, und der in der Lage ist, Inhalte zu filtern. Diese Variante der Lösung schränkt zudem die Verwendungsmöglichkeiten von Multicast stark ein, da im Extremfall jeder Client eine andere Nachricht empfangt.

Eine weitere Alternative wäre in der Kommunikationsarchitektur zu sehen: Die beschriebene Kommunikationslösung ist abhängig von einem zentralen Server, der den *Event Manager* betreibt. Es wäre sicherlich auch denkbar, das System auf Peer-to-Peer-Kommunikation auszulagern, wie es z. B. in [Hellenschmidt \(2005b\)](#) der Fall ist. Die Implementation eines solchen Systems ist allerdings deutlich aufwendiger, so dass in einer ersten Implementation darauf verzichtet werden sollte. Durch die Trennung in Dienste und Kommunikationsadapter ist ein späterer Wechsel der Kommunikationsarchitektur allerdings denkbar und realisierbar.

3 Vorgehen in der Master-Arbeit

Diese Erkenntnisse sollen im Rahmen meiner Master-Arbeit praktisch umgesetzt werden. Ausgehend von den in [Abschnitt 2.3](#) beschriebenen Adaptern zwischen *Event Manager* und Service soll ein Rahmenwerk geschaffen werden, das ähnlich einer Service Oriented Architecture eine transparente Eingliederung in ein verteiltes System ermöglicht.

Der Schwerpunkt der Arbeit soll dabei auf der Implementation eines Rahmenwerks liegen. Die zugehörigen Clients sollen u. a. aus den Ergebnissen des Master-Projekts entnommen ([Hollatz, 2008b](#); [Meißner, 2008](#); [Urich, 2008](#)) und angepasst werden. Die Verwendung der Projektergebnisse soll einen Vergleich zwischen den *Event Managern* ermöglichen und die Leistungsfähigkeit des neuen *Event Managers* vergleichbar machen. Zusätzlich soll eine Simulationsumgebung geschaffen werden, die auch ohne das konkrete Vorhandensein von realen Event-Quellen und -Empfängern das Testen des Systems ermöglicht. Hierbei wird ebenfalls auf die Erkenntnisse aus [Meißner \(2008\)](#) zum Aufbau einer Testumgebung zurückgegriffen werden können.

3.1 Design des Event Managers

Basierend auf den im Projekt erlangten Erkenntnissen ([Hollatz, 2008b](#)) und den Vergleichen mit ähnlichen Systemen in [Hollatz \(2008a\)](#), soll ein verbesserter Event Manager realisiert werden.

Dieser neue *Event Manager* hat folgende Eigenschaften, die sich von denen des *Event Heap* unterscheiden:

Nachrichten im XML-Format. Der *EventHeap* soll in seiner ursprünglichen Fassung als Basis für unterschiedlichste Clients sein (vgl. [Johanson und Fox \(2002\)](#)). Die aktuelle Version kommuniziert jedoch über serialisierte Java-Objekte, was auf der Client-Seite spezielle Wrapper erfordern würde. Eine auf XML basierendes Nachrichtenformat soll hier als Schnittstelle zu Clients ermöglichen, die in anderen Sprachen implementiert wurden. Dieser Umstieg auf ein standardisiertes Nachrichtenformat ermöglicht somit auch Clients, die z. B. in Ruby oder .NET entwickelt wurden, nativ über den *Event Manager* zu kommunizieren.

Unterstützung von Bereichsangaben für Event-Registrierungen. Bei der Registrierung von Events am *EventHeap* hat man die Möglichkeit sich für Events zu registrieren, bei denen Felder einen festgelegten Wert haben müssen. Es wäre somit möglich events mit der Angabe von geeigneten Geräte- oder Bildschirmklassen. Ein Mobiltelefon könnte sich so für Events registrieren, die eine Bildschirmauflösung von 320 x 240 Pixeln oder weniger bestimmt sind.

Verschlüsselung. Die Einführung von Verschlüsselung einzelner Nachrichten(-bestandteile) soll für den Client absolut transparent erfolgen. Da der *Event Manager* keine Verarbeitung der Nachrichten durchführt ist dieser auf den ersten Blick nicht von dieser Änderung betroffen. Da dieser jedoch das Routing der Events auch auf Basis der Feldinhalte durchführen kann, ist für ihn ebenso eine Funktionalität zum Umgang mit verschlüsselten Feldern erforderlich. Je nach Vertrauensstand zwischen *Event Manager* und Nachrichtenquelle kann der *Event Manager* in der Lage sein, die verschlüsselten Felder zu lesen oder diese zu ignorieren.

3.2 Design der Service-Adapter

Die Service-Adapter haben den Zweck eine transparente Form der Verteilung zu ermöglichen (Coulouris u. a., 2005). Sie verbergen die Logik der Kommunikation, Authentifikation und Verschlüsselung vor der Anwendungslogik. Der Adapter bietet einen einfachen Adapter für die Kommunikation mit anderen Geräten der Umgebung (s.a. Adapter-Pattern in Gamma u. a. (1995)).

Wie bereits in [Abschnitt 2.3.3](#) angesprochen, sollen die Adapter als BDI-Agenten implementiert werden, was nach Möglichkeit auf Basis des Jadex-Frameworks geschehen soll.

3.3 Design des Simulationsservice

Da insbesondere in der Anfangsphase der Entwicklung implementierte Gerätedienste in sehr begrenzten Umfang zur Verfügung stehen werden und auch zu späterer Zeit eine Entwicklung und Test der Kommunikation in produktiver Umgebung nicht unbedingt von Vorteil ist, soll ein Simulationsservice entwickelt werden. Der Simulationsservice ermöglicht es, den Nachrichtenverkehr zu überwachen und das senden und Empfangen von Events zu Testen.

Der Simulationsservice kann auch eingesetzt werden, um Lasttests auf dem System auszuführen. Ein wichtiger Test, der das Verhalten der Anwendung bei starkem Nachrichtenaufkommen überprüfbar macht.

4 Resümee

4.1 Zusammenfassung

In dieser Arbeit wurde eine eine Architektur vorgestellt, die basierend auf einem modifizierten *EventHeap* über das versenden öffentlicher Events kommuniziert. Als Grundlage der Kommunikation wurde ein XML-basiertes Format vorgeschlagen, dass den nahtlosen Austausch von Informationen zwischen mehreren Plattformen ermöglicht. Im weiteren Verlauf der Arbeit wurde gezeigt, wie ein Rahmenwerk für Clients auf der Grundlage von BDI-Agenten implementiert werden kann und inwiefern sich Sicherheitsaspekte in die Kommunikation einfügen lassen.

4.2 Chancen und Risiken

Die erfolgreiche Umsetzung des hier beschriebenen Systems wird zu einer flexiblen, sicheren und erweiterbaren Infrastruktur für das an der Hochschule für Angewandte Wissenschaften Hamburg geplante *Ambient Intelligence Labor* führen.

Die hauptsächlichen Risiken, die gegen eine Umsetzung sprechen, sind: fehlende Integrationsmöglichkeit der BDI-Agenten, zu hoher Rechenaufwand der Kommunikation durch zusätzliches Parsen von XML-Daten oder die fehlende Akzeptanz einer Client-/Server-Architektur.

Die Integration der Client-Adapter als BDI-Agenten ist ein zentraler Bestandteil der Idee. Der Einsatz von Jadex als Plattform für die Adapter sollte nach dem jetzigen Kenntnisstand möglich sein. Jedoch wird die Verwendung von BDI-Agenten auf mobilen Endgeräten schwierig, da es dem Autor für hier noch an entsprechender praktischer Erfahrung mit konkreten Implementierungen fehlt. Durch die offene Planung der Kommunikationsstrukturen kann auf bestimmten Clients bewusst und ohne große Nachteile auf den Einsatz von Softwareagenten verzichtet werden. Es kann so bei der Implementierung allerdings nicht auf eine standardisierte Architektur zurückgegriffen werden.

Es ist zu erwarten, dass die Kommunikation der Clients über XML eine erhebliche höhere Rechenleistung erfordert, als es beim *EventHeap* der Fall ist. Sollte sich hieraus eine zu große Latenzzeit in der Kommunikation zwischen *Event Manager* und Adapter ergeben, so kann auf diese Art der Kommunikation verzichtet werden. Dieser Aspekt ist nicht essentiell für die Umsetzung des Systems, da zu erwarten ist, dass alle Clients in der ersten Generation auf Java basieren können.

4.3 Ausblick

Es ist geplant die Beschriebene Architektur als Bestandteil des an der Hochschule für Angewandte Wissenschaften Hamburg geplanten Labors für *Ambient Intelligence* zu installieren. Die Umsetzung der vorgestellten Ideen soll in einer Master-Arbeit realisiert werden und anschließend im Labor zum Einsatz kommen und als Basis der Kommunikation zwischen den jeweiligen Geräten dienen.

Literaturverzeichnis

- [Coulouris u. a. 2005] COULOURIS ; DOLLIMORE, Jean ; KINDBERG, Tim: *Distributed Systems: Concepts and Design (4th Edition) (International Computer Science)*. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 2005. – URL <http://www.cdk4.net/>. – Zugriffsdatum: 28.02.2008. – ISBN 0321263545
- [Dey 2001] DEY, Anind K.: Understanding and Using Context. In: *Personal Ubiquitous Comput.* 5 (2001), Nr. 1, S. 4–7. – URL <http://dx.doi.org/10.1007/s007790170019>. – Zugriffsdatum: 28.02.2008. – ISSN 1617-4909
- [Dreyer 2008] DREYER, Markus: *Your Home in Your Hand*, Hochschule für Angewandte Wissenschaften Hamburg, Seminararbeit, 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08-aw/vortraege.html>. – Zugriffsdatum: 28.02.2008
- [Gamma u. a. 1995] GAMMA, E. ; HELM, R. ; JOHNSON, R. ; VLISSIDES, J.: *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1995
- [Georgeff u. a. 1999] GEORGEFF, Michael P. ; PELL, Barney ; POLLACK, Martha E. ; TAMBE, Milind ; WOOLDRIDGE, Michael: The Belief-Desire-Intention Model of Agency. In: *ATAL '98: Proceedings of the 5th International Workshop on Intelligent Agents V, Agent Theories, Architectures, and Languages*. London, UK : Springer-Verlag, 1999, S. 1–10. – URL <http://www.csc.liv.ac.uk/~mjw/pubs/atal98b.pdf>. – Zugriffsdatum: 28.02.2008. – ISBN 3-540-65713-4
- [Hellenschmidt 2005a] HELLENSCHMIDT, Michael: Distributed implementation of a self-organizing appliance middleware. In: *sOc-EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence*. New York, NY, USA : ACM, 2005, S. 201–206. – URL <http://doi.acm.org/10.1145/1107548.1107600>. – Zugriffsdatum: 28.02.2008. – ISBN 1-59593-304-2
- [Hellenschmidt 2005b] HELLENSCHMIDT, Michael: DynAMITE: Self-organizing middleware technologies for Ambient Intelligence. In: *Computer Graphik Topics* (2005), Nr. Vol. 17. – URL http://www.inigraphics.net/press/topics/2005/issue1/1_05a02.pdf. – Zugriffsdatum: 28.02.2008

- [Hollatz 2007] HOLLATZ, Dennis: *Konzepte für interaktive Räume*, Hochschule für Angewandte Wissenschaften Hamburg, Seminararbeit, 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2007/vortraege.html>. – Zugriffsdatum: 28.02.2008
- [Hollatz 2008a] HOLLATZ, Dennis: *Managing Information - Infrastructures for Ambient Intelligence*, Hochschule für Angewandte Wissenschaften Hamburg, Seminararbeit, 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08-aw/vortraege.html>. – Zugriffsdatum: 28.02.2008. – in Vorbereitung
- [Hollatz 2008b] HOLLATZ, Dennis: *SmartShelf – Projektbericht*, Hochschule für Angewandte Wissenschaften Hamburg, Projektbericht, 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projects.html>. – Zugriffsdatum: 28.02.2008. – in Vorbereitung
- [Imamura u. a. 2002] IMAMURA, Takeshi ; DILLAWAY, Blair ; SIMONAL, Ed: *XML Encryption Syntax and Processing*. 2002. – URL <http://www.w3.org/TR/xmlenc-core/>. – Zugriffsdatum: 28.02.2008
- [Johanson und Fox 2002] JOHANSON, B. ; FOX, A.: The Event Heap: a coordination infrastructure for interactive workspaces. In: *Mobile Computing Systems and Applications, 2002. Proceedings Fourth IEEE Workshop on (2002)*, S. 83–93. – URL <http://ieeexplore.ieee.org/search/wrapper.jsp?arnumber=1017488>. – Zugriffsdatum: 28.02.2008
- [Johanson und Fox 2004] JOHANSON, Brad ; FOX, Armando: Extending tuplespaces for coordination in interactive workspaces. In: *J. Syst. Softw.* 69 (2004), Nr. 3, S. 243–266. – URL [http://dx.doi.org/10.1016/S0164-1212\(03\)00054-2](http://dx.doi.org/10.1016/S0164-1212(03)00054-2). – Zugriffsdatum: 28.02.2008. – ISSN 0164-1212
- [Johanson 2002] JOHANSON, Bradley E.: *Application Coordination Infrastructure for Ubiquitous Computing Rooms*, Department Of Electrical Engineering Of Stanford University, Dissertation, 2002. – URL <http://graphics.stanford.edu/~bjohanso/dissertation/>. – Zugriffsdatum: 28.02.2008
- [Meißner 2008] MEISSNER, Stefan: *Projektbericht – smart:shelf*, Hochschule für Angewandte Wissenschaften Hamburg, Projektbericht, 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicomp>. – Zugriffsdatum: 28.02.2008. – in Vorbereitung
- [MIT Kerberos consortium 2007] MIT KERBEROS CONSORTIUM: *Kerberos: The Network Authentication Protocol*. 10 2007. – URL <http://web.mit.edu/Kerberos/>. – Zugriffsdatum: 28.02.2008
- [Pierce und Nichols 2007] PIERCE, Jeff ; NICHOLS, Jeff: *Personal Information Environ-*

- ments. 2007. – URL <http://www.almaden.ibm.com/cs/projects/pie/>. – Zugriffsdatum: 28.02.2008
- [Pokahr u. a. 2003] POKAHR, Alexander ; BRAUBACH, Lars ; LAMERSDORF, Winfried: Jadex: Implementing a BDI-Infrastructure for JADE Agents. In: *EXP - In Search of Innovation (Special Issue on JADE)* 3 (2003), September, S. 76–85. – URL <http://vsis-www.informatik.uni-hamburg.de/projects/jadex/pubs.php>. – Zugriffsdatum: 28.02.2008
- [Streitz u. a. 1994] STREITZ, Norbert A. ; GEISLER, Jorg ; HAAKE, Jorg M. ; HOL, Jeroen: DOLPHIN: Integrated Meeting Support Across Local and Remote Desktop Environments and LiveBoards. In: *Computer Supported Cooperative Work*, URL <http://citeseer.ist.psu.edu/streitz94dolphin.html>. – Zugriffsdatum: 28.02.2008, 1994, S. 345–358
- [Urich 2008] URICH, Jaroslaw: *Projektbericht – smart:shelf*, Hochschule für Angewandte Wissenschaften Hamburg, Projektbericht, 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicomp>. – Zugriffsdatum: 28.02.2008. – in Vorbereitung
- [Weiser 1991] WEISER, Mark: The computer for the twenty-first century. In: *Scientific American* September (1991), S. 94–100
- [Winograd und Flores 1986] WINOGRAD, Terry ; FLORES, Fernando: *Understanding Computers and Cognition: A New Foundation for Design*. Ablex Publishing Corp., 1986. – ISBN 0-201-11297-3