



Hochschule für Angewandte Wissenschaften Hamburg

Hamburg University of Applied Sciences

Ausarbeitung zur Veranstaltung „Projekt 1“ im Masterstudiengang Informatik SoSe 2010

Lorenz Barnkow

Eine Multitouch-fähige Küchentheke:
Vorbereitende Arbeiten für den Tagesplaner

Inhaltsverzeichnis

Abbildungsverzeichnis	1
Listings	1
1 Einführung	2
2 Kalenderinfrastruktur	2
2.1 Standards für den Datenaustausch	3
2.1.1 Kalenderdaten	3
2.1.2 Kontaktdaten	3
2.1.3 E-Mail	4
2.2 Evaluation von Implementierungen	4
2.2.1 DAViCal	4
2.2.2 Zimbra	5
3 Kalender- und Wetter-Agent	5
3.1 Nachrichtenformat	6
3.2 Kalender-Agent	7
3.3 Wetter-Agent	8
4 Multitouch-Software-Entwicklung	9
4.1 Breeze for WPF 3.5	11
4.2 Multi-Touch Vista	11
4.3 .NET 4	12
5 Zusammenfassung und Ausblick	13
A Beispielkommunikation: Kalender-Agent	14
A.1 Anfrage getEvents	14
A.2 Antwort getEvents	14
B Beispielkommunikation: Wetter-Agent	16
B.1 Anfrage getWeatherForCity	16
B.2 Antwort getWeatherForCity	16
C Beispielausgabe Google Weather	18
Literatur	18

Abbildungsverzeichnis

1	Screenshot der Zimbra-Webanwendung	6
2	Beispielanwendung, die den Wetter-Agenten nutzt	10
3	Gemeinsame Multitouch-API für Surface und WPF 4-Anwendungen aus Feld- kamp (2009)	13

Listings

1	Struktur Anfrage-Nachricht	6
2	Struktur Antwort-Nachricht	6
3	Regionale Vorhersage des Deutschen Wetterdienstes	9
4	getEvents-Anfrage an den Kalender-Agenten	14
5	Antwort des Kalender-Agenten auf getEvents-Anfrage	14
6	getWeatherForCity-Anfrage an den Wetter-Agenten	16
7	Antwort des Wetter-Agenten auf getWeatherForCity-Anfrage	16
8	Vorhersage von Google Weather für Cuxhaven	18

1 Einführung

Die vorliegende Ausarbeitung gibt einen Überblick über die ersten praktischen Arbeiten, um die Projektskizze der Veranstaltung „Anwendungen 1“ (AW1) des vergangenen Semesters (siehe [Barnkow \(2010a\)](#)) umzusetzen. Thematisch ging es in AW1 um einen Multitouch-fähigen Tabletop für den Living Place Hamburg und eine exemplarische Tagesplaner-Anwendung.

An der HAW Hamburg entsteht im Rahmen des Living Place Hamburg ein Labor, in dem u. a. Themen wie Ambient Assisted Living, Smart Homes und Computer-Supported Cooperative Work insbesondere im Bezug auf die Gestaltung zukünftiger Wohn- und Arbeitsräume erarbeitet werden können. Die interdisziplinäre Ausrichtung dieses Labors und die Tatsache, dass mit dem Living Place Hamburg eine reale Wohnumgebung geschaffen wird, erlauben vielfältige Arbeiten und Experimente, wie eine intelligente und kontextabhängige Anpassung des Wohnraums an die Bedürfnisse und Wünsche der Bewohner stattfinden kann. Weitere Beschreibungen des Projektes finden sich unter anderem in [HAW Hamburg \(2010\)](#), [Rahimi und Vogt \(2010\)](#) oder [Gregor u. a. \(2009\)](#).

Der sog. Tagesplaner, der in AW1 skizziert wurde, sollte es dem fiktiven Bewohner des Living Place Hamburg erlauben, sich morgens auf den bevorstehenden Tagesablauf vorzubereiten. Hierfür sollten die Termine des Tages und verknüpfte Informationen visualisiert werden. Auf intuitive Art und Weise sollte der Tagesplaner Korrespondenzen, Fahrt-Routen, Kontakte oder das heutige Wetter aufbereiten und darstellen.

Im Hinblick auf diese Ideenskizze wurden in diesem Semester verschiedene Teilaspekte untersucht und Lösungen erarbeitet. Zunächst wurde geeignete Software für die Implementierung einer Terminverwaltung evaluiert (siehe Kapitel 2). Aufbauend auf einer bestehenden Nachrichtenvermittlungsschicht wurden zwei Software-Agenten entwickelt, die Termin- und Wetterdaten zur Verfügung stellen (siehe Kapitel 3). Im Laufe des Semesters wurden außerdem verschiedene Applikationsframeworks für die Entwicklung von Multitouch-Anwendungen evaluiert (siehe Kapitel 4).

2 Kalenderinfrastruktur

Im Rahmen des Living Place Hamburg gibt es, neben dem o. g. Tagesplaner, auch andere Projekte, die mit Termin- bzw. Kalenderdaten umgehen müssen. Hierzu zählen u. a. der sog. „Wecker 2.0“ von [Ellenberg \(2009\)](#) und ggf. die Bemühungen zu „Home Office 2.0“ von [Panier \(2009\)](#) und anderen.

Weiterhin soll der fiktive Bewohner des Living Place Hamburg auch die Möglichkeit haben, seinen Computer oder sein Smartphone für den Zugriff auf solche Daten zu verwenden. Da gerade innerhalb des Living Place Hamburg eine ganzheitliche Lösung entstehen soll und

die einzelnen Software-Systeme zum emergenten Verhalten der intelligenten Wohnumgebung beitragen werden, ist eine gemeinsame Terminverwaltung erforderlich. Neben einer zentralen Terminverwaltung wäre ggf. auch eine gemeinsame Lösung für Kontaktdaten und E-Mails wünschenswert, um eine bessere Integration der Tagesplaner-Anwendung mit dem Home Office 2.0-Arbeitsbereich zu erreichen.

Hierzu wurden zunächst geeignete Standards ermittelt, die von verfügbarer Standard-Software, wie z. B. Apples Mail, iCal und Address Book unter Mac OS X¹, unterstützt werden. Anschließend wurden verschiedene Implementierungen evaluiert, um eine entsprechende Infrastruktur für den Living Place Hamburg bereit zu stellen.

2.1 Standards für den Datenaustausch

2.1.1 Kalenderdaten

Ein verbreiteter Standard zum Austausch von Terminen und Kalenderdaten ist das iCalendar-Format, das in RFC 5545 (Desruisseaux (2009)) beschrieben wird. Es wird üblicherweise in Dateien mit der Endung `.ics` gespeichert und ausgetauscht. Hierbei handelt es sich jedoch nur um ein Format zur Persistierung der Daten und ist für sich genommen nicht für konkurrierenden Zugriff aus mehreren Anwendungen geeignet.

RFC 4791 beschreibt mit CalDAV ein Protokoll zum Austausch von Kalenderdaten (Daboo u. a. (2007)), die wiederum dem iCalendar-Format entsprechen müssen. Bei CalDAV handelt es sich um eine Erweiterung des WebDAV-Protokolls aus RFC 4918 (Dusseault (2007)), einem Internetprotokoll für konkurrierenden Zugriff auf versionierbare Dokumente. Mit Hilfe eines CalDAV-fähigen Servers ist es möglich mit mehreren Clients gleichzeitig (auch schreibend) auf Kalenderdaten zuzugreifen und diese auszutauschen.

Beide Standards werden u. a. von Apple iCal, Google Calendar² und Mozilla Lightning³ unterstützt.

2.1.2 Kontaktdaten

Für den Austausch von Kontaktdaten bzw. als digitale Visitenkarte hat sich das, in RFC 2426 beschriebene, vCard-Format (Dawson und Howes (1998)) etabliert. Analog zum iCalendar-Format sind die Dateien mit der Endung `.vcf` bzw. `.vcard` lediglich ein Container für Kontaktdaten, wie Namen, Adressen und Telefonnummern, und stellen somit keinerlei Unterstützung für konkurrierende Zugriffe zur Verfügung.

¹<http://www.apple.com/macosx/what-is-macosx/mail-ical-address-book.html>

²<http://www.google.com/calendar/>

³<http://www.mozilla.org/projects/calendar/>

Auch hier wurde ein auf WebDAV-basierendes Protokoll entworfen, damit mehrere Clients gleichzeitig die Kontaktdaten bearbeiten können. Das relativ junge CardDAV-Protokoll ist in einem IETF⁴-Draft beschrieben (Daboo (2009)).

Während das vCard-Format ähnlich weit verbreitet ist, wie iCalendar, ist die Client-seitige Unterstützung des CardDAV-Protokolls zurzeit eher eine Ausnahme. Sowohl vCard, als auch CardDAV werden beispielsweise von Apples Address Book unterstützt.

Eine deutlich verbreitetere Alternative zur Speicherung und zum konkurrierenden Zugriff auf Kontaktdaten stellt prinzipiell das LDAP-Protokoll dar (RFC 4511, Sermersheim (2006)). Bei LDAP handelt es sich um ein Zugriffsprotokoll für Verzeichnisdienste, in denen jedoch – abgesehen von Personen – beliebige Objekte, wie Drucker, Gruppen oder Dokumente angeordnet sein können. Da Einträge in einem LDAP-Verzeichnis eine Menge von beliebigen Attributen sind, ist die Kompatibilität beim Austausch von Kontaktdaten schwer zu gewährleisten.

2.1.3 E-Mail

Der Versand und Empfang von E-Mails über SMTP (RFC 5321, Klensin (2008)), POP3 (RFC 1939, Myers und Rose (1996)) und IMAP (RFC 3501, Crispin (2003)) wird von nahezu allen gängigen E-Mail-Clients und -Servern unterstützt.

2.2 Evaluation von Implementierungen

Auf der Suche nach geeigneten Server-Implementierungen, um eine Infrastruktur für den Living Place Hamburg bereit zu stellen, wurden verschiedene Programme in Betracht gezogen. Die Unterstützung von E-Mail-Diensten war dabei nicht von Anfang an geplant und wurde erst später zu den Anforderungen hinzugefügt.

Praktisch getestet wurden die beiden Lösungen DAViCal und Zimbra, wobei die endgültige Wahl, nach Rücksprache mit anderen Projektteilnehmern, auf Zimbra gefallen ist.

2.2.1 DAViCal

Bei DAViCal⁵ handelt es sich um eine freie CalDAV-Implementierung, die weder Kontakte noch E-Mails verwalten kann. Sie ist in PHP⁶ realisiert und speichert die Kalenderdaten in eine PostgreSQL⁷ Datenbank.

⁴Internet Engineering Task Force – <http://www.ietf.org/>

⁵<http://www.davical.org/>

⁶<http://www.php.net/>

⁷<http://www.postgresql.org/>

DAViCal konnte, im Rahmen der vorliegenden Arbeit, mit Apple iCal und Mozilla Lightning erfolgreich getestet werden. Darüber hinaus wird DAViCal von einer kleinen Gruppe von Projektteilnehmern des Living Place Hamburg verwendet, um sich untereinander zu koordinieren.

Die Einfachheit von DAViCal ist jedoch auch ein Nachteil. Es kann zwar ein externes LDAP-Verzeichnis zur Benutzerverwaltung und Authentifizierung angebunden werden, dabei werden die Benutzer aber innerhalb der DAViCal-Datenbank redundant gespeichert. Eine Anbindung an E-Mail-Dienste wird nicht unterstützt und eine Erweiterung für CardDAV ist zurzeit nicht im stabilen Release enthalten.

2.2.2 Zimbra

Die Zimbra Collaboration Suite (ZCS)⁸ ist ein kompletter Software-Stack für E-Mail-, Instant-Messaging-, CalDAV- und CardDAV-Dienste sowie für einfache Wikis. Die Basis der ZCS bildet bestehende, freie Software, u. a. HTTP- und LDAP-Server sowie Virens Scanner. Hinzu kommt, eine von Zimbra angefertigte, freie Java⁹-Implementierung der CalDAV- und CardDAV-Dienste und ein Client in Form einer Webanwendung (siehe Abb. 1).

Die CalDAV- und CardDAV-Implementierungen konnten erfolgreich mit Apple iCal und Address Book getestet werden. ZCS und DAViCal ließen sich beide relativ einfach installieren und einrichten. Die Wahl fiel letztendlich jedoch auf Zimbra, da neben CalDAV auch CardDAV und E-Mail unterstützt wurden. Die Benutzerverwaltung erfolgt innerhalb eines mitgelieferten LDAP-Verzeichnisses. Zusätzlich ist ein Instant-Messaging-Dienst, basierend auf XMPP (RFC 3920, [Saint-Andre \(2004\)](#)), verfügbar.

Außerdem können ein großer Teil der Funktionen der genannten Standard-Protokolle und viele administrative Zimbra-Aufgaben auch über Webservices angesprochen werden. Eine (undokumentierte) Client-Bibliothek, die u. a. in der Zimbra-Webanwendung zum Einsatz kommt, wird mit Zimbra ausgeliefert.

3 Kalender- und Wetter-Agent

Auch vor dem Living Place Hamburg wurden an der HAW, im Rahmen der Ambient Intelligence Labore, Konzepte für intelligente Wohnungen untersucht und entwickelt. Ein Kernaspekt des sog. iFlat-Projektes war, dass die einzelnen Prozesse innerhalb der Wohnung über eine Kommunikationsinfrastruktur lediglich eine lose gekoppelte Verbindung hatten. Analog zu der im iFlat verwendeten Blackboard-Architektur (siehe auch [Stegelmeier u. a. \(2009\)](#)) soll im Living

⁸<http://www.zimbra.com/>

⁹<http://www.java.com/>

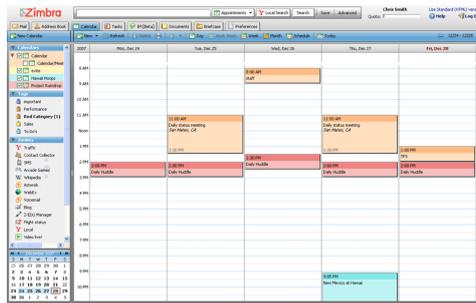


Abbildung 1: Screenshot der Zimbra-Webanwendung

Place Hamburg die Software ActiveMQ¹⁰ als Lösung zur Nachrichtenvermittlung eingesetzt werden. In diesem Modell melden sich die verschiedenen Prozesse beim ActiveMQ-Dienst an, um Nachrichten zu bestimmten Themen zu veröffentlichen oder zu abonnieren.

3.1 Nachrichtenformat

In der konkreten Umsetzung im Living Place Hamburg werden zurzeit alle Nachrichten mit JSON¹¹ kodiert. Weiterhin sind lediglich Anfrage- und Antwort-Nachrichten definiert. Der generelle Aufbau dieser Nachrichten ist in Listing 1 bzw. 2 zu sehen, wobei hier nur die Pflichtfelder zu sehen sind.

Listing 1: Struktur Anfrage-Nachricht

```
{
  "version" : "1.0",
  "id"      : "1234567890",
  [ . . . ]
}
```

Listing 2: Struktur Antwort-Nachricht

```
{
  "version" : "1.0",
  "id"      : "1234567890",
  "status"  : "OK",
  [ . . . ]
}
```

Über das Versions-Feld kann der Sender der Anfrage festlegen, welche Version des jeweiligen Anwendungsprotokolls verwendet werden soll. Idealerweise sollten Agenten, die Anfragen erhalten, bei Änderungen im Protokoll auch weiterhin ältere Versionen unterstützen. So wird, bei einer Erweiterung des Protokolls, keine Anpassung an den beteiligten Agenten erforderlich.

Über das Id-Feld kann der Empfänger einer Antwort ermitteln, ob sich diese auf eine seiner eigenen Anfragen bezieht. Zusammengehörige Anfrage-Antwort-Paare müssen die gleiche Id aufweisen.

Im Status-Feld kann der Sender der Antwort mitteilen, ob es Probleme bei der Verarbeitung der Anfrage gab. Für Fehlerfälle sind bislang zwei allgemeine Fehler definiert. Sie treten dann

¹⁰<http://activemq.apache.org/>

¹¹JavaScript Object Notation – <http://json.org/>

ein, wenn die Anfrage eine Version anfordert, die vom Antwortenden nicht (oder nicht mehr) unterstützt wird, oder wenn die Anfrage unvollständig ist, da z. B. Pflichtfelder fehlen.

Beide Nachrichten können mit beliebig vielen Feldern erweitert werden. Die jeweiligen Erweiterungen sind Anwendungsspezifisch und werden von den Autoren der jeweiligen Dienste festgelegt.

3.2 Kalender-Agent

Der Kalender-Agent soll die Komplexität des CalDAV-Protokolls vor potentiellen Nutzern der Kalenderdaten verbergen. Hierfür wurden zunächst bestehende Java-Bibliotheken für die CalDAV-Anbindung evaluiert.

Die Bibliothek `iCal4j`¹² stellt zunächst nur eine Programmierschnittstelle für iCalendar-Daten dar. Mit Hilfe des sog. Connector-Subprojektes¹³ können solche Daten von einem CalDAV-Server bezogen werden. Während `iCal4j` relativ ausgereift ist und u. a. auch in Zimbra zum Einsatz kommt, ist der Connector zurzeit eher als proof-of-concept anzusehen. Im Testbetrieb wurden alle Termine erfolgreich aus dem Zimbra-Server ausgelesen, ein selektiver Abruf über einen bestimmten Zeitraum wurde nicht unterstützt. Da der Abruf aller Termine ein ungünstiges Skalierungsverhalten hat, wurde von dieser Lösung jedoch abgesehen.

Als Alternative wurde die Bibliothek `caldav4j`¹⁴ getestet. Diese Bibliothek konnte sich jedoch nicht mit dem Zimbra-Server verbinden (eine Testverbindung zu Google Calendar war erfolgreich). Laut Webseite des Projektes, wird die Bibliothek zurzeit nur gegen eine bestimmte CalDAV-Implementierung¹⁵ getestet. Andere Server werden vorerst nicht offiziell unterstützt.

Beide Bibliotheken verfügten, abgesehen von Kommentaren im Quellcode, über keinerlei Dokumentation. Fragen nach Programmbeispielen in den Mailinglisten der Projekte wurden häufig mit kurzen Verweisen auf den *selbsterklärenden* Quellcode beantwortet.

Da eine eigene Implementierung der WebDAV- und CalDAV-Spezifikationen (mehrere hundert Seiten) aus Zeitgründen nicht in Frage kam, wurde die Webservice-Schnittstelle von Zimbra (siehe Abschnitt 2.2.2) getestet. Auch hier existiert nur sehr wenig Dokumentation, aber es gab einige Code-Beispiele aus dem Support-Forum von Zimbra. Aufbauend auf diesen Beispielen und der Zimbra-Bibliothek konnte der Zugriff auf die Kalenderdaten realisiert werden.

In der aktuellen Fassung unterstützt der Kalender-Agent im Wesentlichen eine Anfrage, bei der alle Termine für einen bestimmten Tag abgerufen werden können. Aufbauend darauf bietet er aber auch zwei Hilfs-Anfragen, mit denen sich die Termine für den heutigen oder morgigen

¹²http://wiki.modularity.net.au/ical4j/index.php?title=Main_Page

¹³<http://wiki.modularity.net.au/ical4j/index.php?title=Connector>

¹⁴<http://code.google.com/p/caldav4j/>

¹⁵Chandler, The Note-to-Self Organizer – <http://chandlerproject.org/>

Tag abrufen lassen, ohne ein explizites Datum angeben zu müssen. Beispiele für eine JSON-kodierte Anfrage und Antwort befinden sich in Anhang A.

Neben dem in Java entwickelten Kalender-Agenten, wurde außerdem eine Java-basierte Client-Bibliothek implementiert. Somit müssen potenzielle Benutzer der Kalender-Agenten weder die Kodierung und Dekodierung von JSON, noch die Verbindung zum ActiveMQ-Dienst selber durchführen. Es existiert ein vollständiges Objekt-Modell für die ausgetauschten Daten, das durch die Bibliothek mit Daten befüllt wird.

3.3 Wetter-Agent

Mit Hilfe des Wetter-Agenten sollten Informationen über das aktuelle Wetter im Living Place Hamburg zur Verfügung gestellt werden. Durch die Verwendung eines Agenten, als Schnittstelle zwischen einem Anbieter von Wetterdaten und den eigentlichen Nutzern, wird eine losere Kopplung erreicht. So können mit der Zeit wechselnde Anbieter verwendet werden, ohne dass sich die Nutzer der Daten anpassen müssen.

Ein Szenario für die Verwendung des Wetter-Agenten besteht beim Verlassen der Wohnung. Wird erkannt, dass der Bewohner die Wohnung verlassen möchte, dann kann anhand des Wetter-Agenten das aktuelle Wetter geprüft werden. Regnet es jetzt gerade oder ist die Regenwahrscheinlichkeit sehr hoch und der Bewohner hat keinen Regenschirm bei sich, dann könnte die Wohnung ihn daran erinnern.

In diesem Abschnitt soll lediglich die zur Verfügung gestellte Infrastruktur in Form des Wetter-Agenten beschrieben werden. Eine Regelmaschine, wie sie hier angedeutet wurde, war nicht Gegenstand der beschriebenen Projektarbeit.

Über das Internet können Wetterdaten von einer Vielzahl von Anbietern (teils kostenfrei) bezogen werden. Viele dieser Dienste, wie z.B. [wetter.de](http://www.wetter.de/)¹⁶, bieten Schnittstellen, um die Wetterdaten in eigene Projekte einzubinden. In vielen Fällen werden jedoch nicht die Daten zur Verfügung gestellt, sondern nur animierte Bilddateien, die in die eigene Homepage integriert werden können. Die Anpassungsmöglichkeiten beschränken sich dabei oft auf die Wahl der Farben. Die Nutzungsbedingungen der Webseiten erlauben auch nicht das Auslesen der textuellen Daten aus dem Quelltext der Webseite. Solche Dienste sind für die Umsetzung des Wetter-Agenten deshalb grundsätzlich ungeeignet.

Auch der Deutsche Wetterdienst (DWD)¹⁷ bietet diverse Wetterdaten kostenlos an. Von besonderem Interesse sind bei diesem Angebot, dass die Vorhersagedaten für registrierte Benutzer in textueller und maschinenlesbarer Form angeboten werden und teilweise alle 15 Minuten aktualisiert werden. Eine exemplarische Vorhersage kann Listing 3 entnommen werden.

¹⁶<http://www.wetter.de/>

¹⁷<http://www.dwd.de/>

Listing 3: Regionale Vorhersage des Deutschen Wetterdienstes

Vorhersage von Tiefsttemperatur und Wetter für Mittwoch früh, 12.05.2010			
List/Sylt	5	leichter Regen	
Helgoland	7	leichter Regen	Windböen
Schleswig	5	leichter Regen	
Kiel	5	Regen Windböen	
Fehmarn	6	Regen Windböen	
Norderney	6	leichter Regen	Windböen
Cuxhaven	6	leichter Regen	Windböen
Hamburg	6	Regen	
Schwerin	6	Regen	
Emden	5	leichter Regen	
Bremen	6	leichter Regen	
Münster	6	leichter Regen	
Hannover	6	kräftiger Regen	
Bad Lippspringe	7	Regen	
Brocken	3	Regen	
Temperatur in Grad Celsius			

Der DWD bietet solche Vorhersagen für die Tageszeitabschnitte *früh*, *mittag*, *spät* und *nacht* an. Diese Granularität wird jedoch nur für deutsche Wetterstationen angeboten, während bei den europäischen Stationen nur *früh* und *abends* zur Verfügung stehen. Internationale Stationen enthalten nur Vorhersagen für den gesamten Tag.

Ergänzend zu den relativ feingranularen, tagesaktuellen Daten des DWD für die deutschen Wetterstationen, wurde eine Anbindung an eine undokumentierte Wetter-Schnittstelle von Google¹⁸ implementiert. Diese wurde insbesondere verwendet, um eine Vorhersage für die kommenden drei Tage anzubieten. Außerdem können Windstärke und -richtung sowie Luftfeuchtigkeit ermittelt werden. Eine Beispielausgabe dieser Schnittstelle ist in Anhang C in Listing 8 abgebildet.

Die aktuelle Implementation des Wetter-Agenten erlaubt lediglich eine Anfrage, bei der das Wetter für eine bestimmte Stadt abgefragt werden kann. Beispiele für die JSON-kodierte Anfrage und Antwort befinden sich in Anhang B.

Analog zum Kalender-Agenten wurde auch hier eine Client-Bibliothek implementiert, die sowohl die JSON-Kodierung bzw. -Dekodierung, als auch die ActiveMQ-Kommunikation übernimmt. Ein einfaches Objekt-Modell steht für den Datenzugriff zur Verfügung.

Auf Basis dieser Client-Bibliothek wurde eine Beispielanwendung implementiert, die die Verwendung des Wetter-Agenten demonstriert. Abbildung 2 zeigt einen Screenshot.

4 Multitouch-Software-Entwicklung

An der HAW Hamburg wurden bereits verschiedene Arbeiten und Projekte im Bezug auf Multitouch durchgeführt (siehe z. B. Barnkow (2010a)). Obwohl die Entwicklung von Plattform-unabhängigen Anwendungen mit Hilfe von Java, Flash¹⁹ o. ä. wünschenswert wäre, wurde hiervon aufgrund der Empfehlungen erfahrenerer Studierenden abgesehen. Der Großteil der

¹⁸Beispiel-Abfrage für Cuxhaven: <http://www.google.com/ig/api?weather=Cuxhaven>

¹⁹<http://www.adobe.com/flashplatform/>

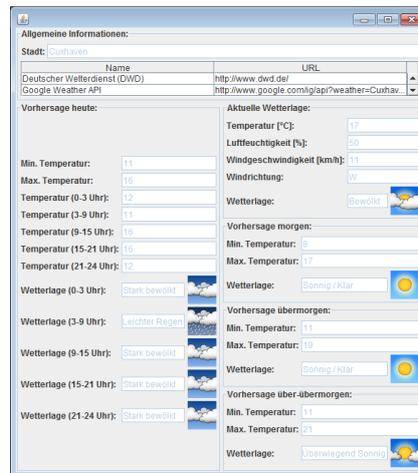


Abbildung 2: Beispielanwendung, die den Wetter-Agenten nutzt

Multitouch-Entwicklungen an der HAW Hamburg wurden mittels C#²⁰ und der .NET-Plattform²¹ durchgeführt. Um sowohl von diesen Vorkenntnissen zu profitieren, als auch bessere Möglichkeiten des Code-Austausches mit aktuell laufenden Multitouch-Projekten an der HAW Hamburg zu schaffen, wurde für dieses Projekt auch mit den genannten Technologien gearbeitet. Für die Gestaltung der grafischen Oberflächen wurde auf Windows Presentation Foundation (WPF)²² zurückgegriffen.

Als Testhardware kommt ein dreaMTouch²³-Schirm zum Einsatz, der bis zu 32 gleichzeitige Berührungen unterscheiden bzw. erkennen und verfolgen kann. Die mitgelieferte Treibersoftware übersetzt die Berührungen in das TUIO-Protokoll²⁴ (siehe z. B. Barnkow (2010a)). Außerdem können die Berührungen auch direkt an das Windows 7-Betriebssystem²⁵ übergeben werden, welches nativ Multitouch-Eingaben (im Folgenden als Windows Touch bezeichnet) unterstützt. Diese Funktionalität ist im dreaMTouch-Treiber zurzeit noch in der Entwicklung, weshalb für die Tests auf die Software Multi-Touch Vista²⁶ zurückgegriffen wurde. Diese enthält auch einen Windows 7-Treiber, der TUIO-Nachrichten in Windows Touch-Nachrichten für Windows 7 übersetzt.

Rund um die .NET-Plattform sind in den letzten Jahren eine größere Anzahl verschiedener Multitouch-Frameworks entstanden. Im Rahmen des Projektes wurden sowohl zwei Frameworks von Drittanbietern, als auch die Multitouch-Unterstützung der aktuellen .NET-Plattform (Ver-

²⁰<http://msdn.microsoft.com/en-us/vcsharp/aa336809.aspx>

²¹<http://msdn.microsoft.com/netframework/>

²²<http://msdn.microsoft.com/en-us/library/ms754130.aspx>

²³<http://www.dreamtouchblog.com/>

²⁴<http://www.tuio.org/>

²⁵<https://www.microsoft.com/windows/windows-7/>

²⁶<http://multitouchvista.codeplex.com/>

sion 4) evaluiert. Die entsprechenden Erfahrungen werden in den nächsten Abschnitten kurz beschrieben. Das wichtigste Auswahlkriterium für die nähere Betrachtung eines Frameworks war, dass die Standard-Steuererelemente von WPF unterstützt werden. Die Eigenentwicklung solcher Steuererelemente, wie z. B. Buttons oder Listen, sollte nach Möglichkeit vermieden werden.

4.1 Breeze for WPF 3.5

Bei Breeze for WPF 3.5²⁷ handelt es sich um ein freies Framework, das aus einer kommerziellen Entwicklung heraus freigegeben wurde und mit dem TUIO-Protokoll arbeitet. Grundlegende Manipulationen, wie Rotation, Skalierung oder Translation, werden direkt und mit relativ wenig Aufwand unterstützt. Dabei dienen die sog. Multitouch-Container als Wrapper für die dargestellten WPF-Anzeigeelemente. Über die Eigenschaften des Containers können für jedes Element oder eine ganze Hierarchie von Elementen die unterstützten Manipulationsgesten an- bzw. ausgeschaltet werden.

Vorteilhaft war auch, dass für einige WPF-Standard-Steuererelemente bereits sog. Handler zur Verfügung standen. Diese setzen die erkannten Gesten dann so um, dass beispielsweise Button angetippt oder durch Listen gescrollt werden kann. Bei den ersten Tests hat sich jedoch gezeigt, dass eben diese Unterstützung nur teilweise gegeben war.

Für die sog. Comboboxen, die beim Anklicken eine Liste von Optionen nach unten öffnen, existierte überhaupt kein Handler. Listboxen konnten zwar per Touch-Eingabe durchlaufen werden, die Auswahl einer Option in der Liste war aber nur dann erfolgreich möglich, wenn diese Option auf der ersten „Bildschirmseite“ der Liste war.

Da der Quellcode des Frameworks offen liegt, wäre die Implementierung eigener und neuer Handler denkbar, wurde aber im Rahmen des Projektes nicht weiter verfolgt.

4.2 Multi-Touch Vista

Multi-Touch Vista wurde für WPF 3.5 und TUIO entwickelt und sollte die bestehenden WPF-Steuererelemente Touch-fähig machen. Erste Tests haben aber gezeigt, dass die Verarbeitungsgeschwindigkeit des Frameworks zu gering ist. Objekte konnten nur ruckelnd bewegt, rotiert oder skaliert werden. Bei der Internetrecherche wurden zwar einige Mail-Listen- oder Foren-Beiträge gefunden, die dasselbe Problem, jedoch keine Lösungen beschrieben. Aufgrund dieser Probleme wurde das Verhalten mit WPF-Steuererelementen gar nicht erst näher betrachtet.

Interessant an dem Projekt ist jedoch, dass der o. g. Windows 7-Treiber mitgeliefert wird, der TUIO-Nachrichten auf Windows Touch-Eingaben abbilden kann. Auch aufgrund dieses Treibers

²⁷<http://code.google.com/p/breezemultitouch/>

ist die weitere Entwicklung des Frameworks im Hinblick auf WPF 3.5 offiziell eingestellt, da WPF 4 (.NET-Framework 4) direkt Windows Touch-Eingaben verarbeitet.

4.3 .NET 4

Mit der Veröffentlichung von .NET 4 im April 2010 wurde WPF um die Unterstützung von Windows Touch-Eingaben erweitert. WPF stellt dabei sowohl die „rohen“ Berührungen, als auch daraus abgeleitete Gesten zur Verfügung.

Die mitgelieferten Steuerelemente können selbst Berührungen verarbeiten, aber auch eine eigene Logik könnte hinzugefügt werden. Hierbei ist zu beachten, dass diese Steuerelemente fast nur einfache Gesten wie das Anklicken einer Buttons oder das Scrollen in einer Liste unterstützen. Das Vergrößern eines Fensters mit zwei Fingern muss beispielsweise selber implementiert werden. Weiterhin scheint in die Unterstützung der bestehenden Steuerelemente nicht besonders viel Zeit investiert worden zu sein. Listboxen scrollen z. B. nicht Pixel-genau mit der Verschiebung des Fingers, sondern zeilenorientiert. Außerdem gibt es vom System kein gesondertes optisches Feedback bei Berührungen. Wird etwa ein Finger auf einem Button abgesetzt, so wird dies zwar korrekt registriert, jedoch dem Benutzer fast nicht sichtbar gemacht.

Zeitgleich mit .NET 4 wurde auch das Surface Toolkit for Windows Touch²⁸ in einer Beta-Version freigegeben. Hierbei handelt es sich um eine Portierung einiger Steuerelemente des Surface-Tabletops²⁹ auf WPF 4. Diese Steuerelemente verwenden verschiedene grafische Animationen, um dem Benutzer ein optisches Feedback über die erkannten Berührungen zu liefern. Insgesamt wirken die Steuerelemente deutlich ausgereifter und haben bei ersten Testanwendungen einen guten Eindruck hinterlassen.

Hinzu kommt, dass unter Verwendung dieses Toolkits ein stärkerer Erfahrungsaustausch mit den Studierenden stattfinden kann, die an der HAW Hamburg Software-Entwicklung an einem Surface-Tabletop betreiben. Ein aktueller Themenschwerpunkt aus diesem Bereich ist beispielsweise die Erprobung neuer Interaktionsmöglichkeiten für Menüs bzw. Kontextmenüs für Multitouch-Eingaben. Für die nächste Version des Surface SDKs³⁰ ist darüber hinaus geplant, die Verarbeitung von Touch-Eingaben auch in Surface-Anwendungen auf WPF 4 basieren zu lassen (siehe Abbildung 3, Feldkamp (2009)).

²⁸<http://www.microsoft.com/downloads/details.aspx?FamilyID=801907A7-B2DD-4E63-9FF3-8A2E63932A74&displaylang=en&displaylang=en>

²⁹<http://www.microsoft.com/surface/>

³⁰Software Development Kit

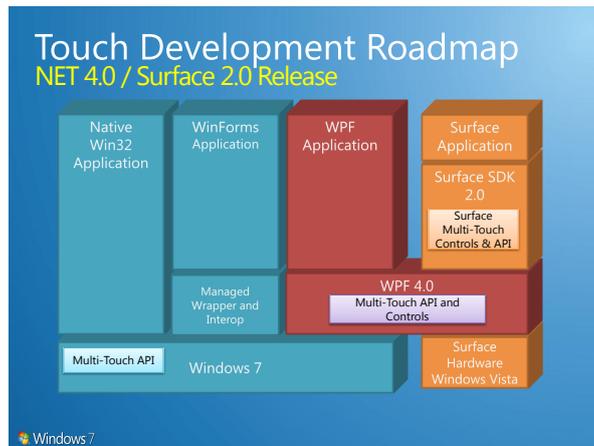


Abbildung 3: Gemeinsame Multitouch-API für Surface und WPF 4-Anwendungen aus [Feldkamp \(2009\)](#)

5 Zusammenfassung und Ausblick

Im Rahmen der Veranstaltung „Projekt 1“ konnten einige vorbereitende Arbeiten umgesetzt und Erkenntnisse gewonnen werden. Es wurde mittels der Zimbra Collaboration Suite (ZCS) eine Infrastruktur für Kalender-, Kontaktverwaltungs- und E-Mail-Dienste geschaffen werden. Diese soll u. a. dem einleitend erwähnten Tagesplaner und dem Wecker 2.0 von [Ellenberg \(2009\)](#) als Datenbasis im Living Place Hamburg dienen.

Aufbauend auf dem ActiveMQ-Nachrichtendienst wurden zwei Software-Agenten implementiert. Der Kalender-Agent stellt die ZCS-Terminaten zur Verfügung, während der Wetter-Agent nach dem aktuellen und kommenden Wetter befragt werden kann. Beide Agenten dienen als Adapter und sollen die dahinter liegenden Dienste verbergen. So bleiben die eigentlich genutzten Dienste austauschbar, ohne dass alle Nutzer im Living Place Hamburg davon betroffen wären.

Abschließend wurden verschiedene Frameworks zur Erstellung von Multitouch-Anwendungen praktisch erprobt. Hierbei ist letztlich die Wahl auf WPF 4 und Windows Touch gefallen, um den Austausch von Code unter den verschiedenen Projekten an der HAW Hamburg zu fördern.

Somit ist ein großer Teil der Grundlagen für die weitere Entwicklung von Anwendungen für einen Multitouch-Tabletop im Living Place Hamburg erarbeitet. Neben der eigentlichen Umsetzung des Tagesplaners, muss in Zukunft auch daran gearbeitet werden, wie Multitouch-Anwendungen die Gruppenarbeiten an großen Tabletops sinnvoll unterstützen können (siehe auch [Barnkow \(2010b\)](#)). Es müssen konkrete Anwendungsszenarien erstellt werden, anhand derer Lösungen erarbeitet und evaluiert werden können.

A Beispielkommunikation: Kalender-Agent

A.1 Anfrage getEvents

Listing 4: getEvents-Anfrage an den Kalender-Agenten

```
{
  "version"      : "0.1",
  "id"           : "barnko_1-PC-50135-1275642564381-1:08787e4dd57",
  "requesttype"  : "getEvents",
  "date"         : "2010-12-24",
  "timezone"     : "Europe/Berlin"
}
```

A.2 Antwort getEvents

Listing 5: Antwort des Kalender-Agenten auf getEvents-Anfrage

```
{
  "version"      : "0.1",
  "id"           : "barnko_1-PC-50135-1275642564381-1:08787e4dd57",
  "status"       : "OK",
  "responsetype" : "getEvents",
  "date"         : "2010-12-24",
  "timezone"     : "Europe/Berlin",
  "events"       : [
    {
      "uid"       : "123456890@livingplace.org",
      "summary"   : "Weihnachtsessen 2010",
      "description" : "Fülltext... Fülltext... Fülltext.",
      "location"  : "Zuhause",
      "organizer" :
        {
          "name"   : "Max Mustermann",
          "address" : "max.mustermann@livingplace.org"
        },
      "attendees" : [
        {
          "name"   : "Meike Mustermann",
          "address" : "meike.mustermann@livingplace.org"
        },
        {
          "name"   : "Hans Dampf",
          "address" : "hans.dampf@livingplace.org"
        },
        {
          "name"   : "Hans Hummel",
          "address" : "hans.hummel@livingplace.org"
        }
      ]
    }
  ],
}
```

```
    "startDate" : {  
      "date" : "2010-12-24 15:00:00",  
      "timezone" : "Europe/Berlin"  
    },  
    "endDate" : {  
      "date" : "2010-12-24 22:00:00",  
      "timezone" : "Europe/Berlin"  
    }  
  ]  
}
```

B Beispielkommunikation: Wetter-Agent

B.1 Anfrage getWeatherForCity

Listing 6: getWeatherForCity-Anfrage an den Wetter-Agenten

```
{
  "version"      : "0.1",
  "id"           : "barnko_1-PC-50135-1275642564381-1:08787e4dd57",
  "requesttype"  : "getWeatherForCity",
  "city"         : "Hamburg"
}
```

B.2 Antwort getWeatherForCity

Listing 7: Antwort des Wetter-Agenten auf getWeatherForCity-Anfrage

```
{
  "version"      : "0.1",
  "id"           : "barnko_1-PC-50135-1275642564381-1:08787e4dd57",
  "status"       : "OK",
  "responsetype" : "getWeatherForCity",
  "sources"      : [
    {
      "name" : "Google Weather API",
      "url"  : "http://www.google.com/ig/api?weather=Hamburg"
    },
    {
      "name" : "Deutscher Wetterdienst (DWD)",
      "url"  : "http://www.dwd.de/"
    }
  ],
  "city"         : "Hamburg",
  "current_conditions" : {
    "temperature" : 22, // in °C
    "humidity"    : 39, // in %
    "wind_speed"  : 8,  // in km/h
    "wind_direction" : "NW",
    "condition"   : "sunny"
  },
  "todays_forecast" : {
    "temperature_min" : 13,
    "temperature_max" : 25,
    "temperature_night_a" : 13, // von 0 bis 3 Uhr
    "temperature_morning" : 19, // von 3 bis 9 Uhr
    "temperature_noon" : 15, // von 9 bis 15 Uhr
    "temperature_evening" : 21, // von 15 bis 21 Uhr
    "temperature_night_b" : 14, // von 21 bis 24 Uhr
    "condition_night_a" : "chance_of_rain", // von 0 bis 3 Uhr
  }
}
```

```
    "condition_morning" : "chance_of_rain", // von 3 bis 9 Uhr
    "condition_noon"    : "chance_of_rain", // von 9 bis 15 Uhr
    "condition_evening" : "chance_of_rain", // von 15 bis 21 Uhr
    "condition_night_b" : "chance_of_rain" // von 21 bis 24 Uhr
  },
  "forecast_1" : {
    "temperature_min" : 13,
    "temperature_max" : 25,
    "condition"       : "chance_of_rain"
  },
  "forecast_2" : {
    "temperature_min" : 13,
    "temperature_max" : 25,
    "condition"       : "mostly_cloudy"
  },
  "forecast_3" : {
    "temperature_min" : 13,
    "temperature_max" : 25,
    "condition"       : "sunny"
  }
}
```

C Beispielausgabe Google Weather

Listing 8: Vorhersage von Google Weather für Cuxhaven

```
<?xml version="1.0"?>
<xml_api_reply version="1">
  <weather module_id="0" tab_id="0" mobile_row="0" mobile_zipped="1" row="0" section="0" >
    <forecast_information>
      <city data="Cuxhaven, Lower Saxony"/>
      <postal_code data="Cuxhaven"/>
      <latitude_e6 data=""/>
      <longitude_e6 data=""/>
      <forecast_date data="2010-08-30"/>
      <current_date_time data="2010-08-30 14:20:00 +0000"/>
      <unit_system data="US"/>
    </forecast_information>
    <current_conditions>
      <condition data="Partly Cloudy"/>
      <temp_f data="64"/>
      <temp_c data="18"/>
      <humidity data="Humidity: 52%"/>
      <icon data="/ig/images/weather/partly_cloudy.gif"/>
      <wind_condition data="Wind: N at 18 mph"/>
    </current_conditions>
    <forecast_conditions>
      <day_of_week data="Mon"/>
      <low data="51"/>
      <high data="64"/>
      <icon data="/ig/images/weather/chance_of_rain.gif"/>
      <condition data="Chance of Rain"/>
    </forecast_conditions>
    <forecast_conditions>
      <day_of_week data="Tue"/>
      <low data="51"/>
      <high data="60"/>
      <icon data="/ig/images/weather/chance_of_rain.gif"/>
      <condition data="Chance of Rain"/>
    </forecast_conditions>
    <forecast_conditions>
      <day_of_week data="Wed"/>
      <low data="50"/>
      <high data="62"/>
      <icon data="/ig/images/weather/chance_of_rain.gif"/>
      <condition data="Chance of Rain"/>
    </forecast_conditions>
    <forecast_conditions>
      <day_of_week data="Thu"/>
      <low data="48"/>
      <high data="64"/>
      <icon data="/ig/images/weather/chance_of_rain.gif"/>
      <condition data="Chance of Rain"/>
    </forecast_conditions>
  </weather>
</xml_api_reply>
```

Literatur

[Barnkow 2010a] BARNKOW, Lorenz: *Eine Multitouch-fähige Küchentheke: Im Kontext des Living Place Hamburg.* Referat/Hausarbeit. 2010. – URL

- <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw1/vortraege.html>. – abgerufen am: 27.07.2010
- [Barnkow 2010b] BARNKOW, Lorenz: *Eine Multitouch-fähige Küchentheke: Related Work*. Referat/Hausarbeit. 2010. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw2/vortraege.html>. – abgerufen am: 31.08.2010
- [Crispin 2003] CRISPIN, M.: *INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1*. RFC 3501 (Proposed Standard). März 2003 (Request for Comments). – URL <http://www.ietf.org/rfc/rfc3501.txt>. – Updated by RFCs 4466, 4469, 4551, 5032, 5182, 5738
- [Daboo 2009] DABOO, C.: *vCard Extensions to WebDAV (CardDAV)*. IETF Internet-Draft. 2009. – URL <http://ietfreport.isoc.org/idref/draft-ietf-vcarddav-carddav/>. – abgerufen am: 23.08.2010
- [Daboo u. a. 2007] DABOO, C. ; DESRUISSEAUX, B. ; DUSSEAULT, L.: *Calendaring Extensions to WebDAV (CalDAV)*. RFC 4791 (Proposed Standard). März 2007 (Request for Comments). – URL <http://www.ietf.org/rfc/rfc4791.txt>. – Updated by RFC 5689
- [Dawson und Howes 1998] DAWSON, F. ; HOWES, T.: *vCard MIME Directory Profile*. RFC 2426 (Proposed Standard). September 1998. – URL <http://www.ietf.org/rfc/rfc2426.txt>
- [Desruisseaux 2009] DESRUISSEAUX, B.: *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*. RFC 5545 (Proposed Standard). September 2009 (Request for Comments). – URL <http://www.ietf.org/rfc/rfc5545.txt>. – Updated by RFC 5546
- [Dusseault 2007] DUSSEAULT, L.: *HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)*. RFC 4918 (Proposed Standard). Juni 2007 (Request for Comments). – URL <http://www.ietf.org/rfc/rfc4918.txt>. – Updated by RFC 5689
- [Ellenberg 2009] ELLENBERG, Jens: *Wecker 2.0*. Referat/Hausarbeit. 2009. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw1/vortraege.html>. – abgerufen am: 21.02.2010
- [Feldkamp 2009] FELDKAMP, Olaf: *Windows 7 for Developers*. Presentation. 2009. – URL <http://download.microsoft.com/download/D/8/0/D800DB5F-C9B0-4D92-90A2-911F561FD9A5/MSDN%20TechTalk%20Windows%20for%20developers%20-%20Part%20I.pptx>. – abgerufen am: 30.08.2010
- [Gregor u. a. 2009] GREGOR, S. ; RAHIMI, M. ; VOGT, M. ; SCHULZ, T. ; VON LUCK, K.: *Tangible Computing revisited: Anfassbare Computer in Intelligenten Umgebungen*. In: 4. *Kongress Multimediatechnik* IFM Institut für Multimediatechnik gGmbH, Wismar (Veranst.),

- URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/papers/MMWisamar2009.pdf>, 2009. – abgerufen am: 22.02.2010
- [HAW Hamburg 2010] HAW HAMBURG: *living place hamburg: A place for concepts of IT based modern living*. Projektkurzbeschreibung. 2010. – URL http://livingplace.informatik.haw-hamburg.de/content/LivingPlaceHamburg_en.pdf.. – abgerufen am: 26.07.2010
- [Klensin 2008] KLENSIN, J.: *Simple Mail Transfer Protocol*. RFC 5321 (Draft Standard). Oktober 2008. – URL <http://www.ietf.org/rfc/rfc5321.txt>
- [Myers und Rose 1996] MYERS, J. ; ROSE, M.: *Post Office Protocol - Version 3*. RFC 1939 (Standard). Mai 1996 (Request for Comments). – URL <http://www.ietf.org/rfc/rfc1939.txt>. – Updated by RFCs 1957, 2449
- [Panier 2009] PANIER, Karsten: *Home Office 2.0 - Collaborate Working*. Referat/Hausarbeit. 2009. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-awl/vortraege.html>. – abgerufen am: 21.02.2010
- [Rahimi und Vogt 2010] RAHIMI, Mohammadali ; VOGT, Matthias: *Aufbau des Living Place Hamburg*. Projektbericht. 2010. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-proj/rahimi-vogt.pdf>. – abgerufen am: 27.07.2010
- [Saint-Andre 2004] SAINT-ANDRE, P.: *Extensible Messaging and Presence Protocol (XMPP): Core*. RFC 3920 (Proposed Standard). Oktober 2004. – URL <http://www.ietf.org/rfc/rfc3920.txt>
- [Sermersheim 2006] SERMERSHEIM, J.: *Lightweight Directory Access Protocol (LDAP): The Protocol*. RFC 4511 (Proposed Standard). Juni 2006. – URL <http://www.ietf.org/rfc/rfc4511.txt>
- [Stegelmeier u. a. 2009] STEGELMEIER, Sven ; WENDT, Piotr ; VON LUCK, Kai: *iFlat - Eine dienstorientierte Architektur für intelligente Räume*. In: BAEHREN, Thomas (Hrsg.) ; CZOGALA, Dieter (Hrsg.) ; GEHRING, Carmen (Hrsg.) ; KLAUSING, Helmut (Hrsg.) ; WAHLSTER, Wolfgang (Hrsg.): *Tagungsband des 2. Deutschen AAL-Kongresses. Ambient Assisted Living (AAL-09), in Conjunction with 2. Deutscher Kongress mit Ausstellung / Technologien - Anwendungen - Management, January 27-28, Berlin, Germany*. Berlin : VDE-Verlag, Berlin, Offenbach, 2009. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/papers/aal2009.pdf>. – ISBN 978-3-8007-3138-1