Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Projektbericht Sommersemester 2009

Mohammadali Rahimi, Matthias Vogt

Vorbereitende Arbeiten zum Living Place Hamburg

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Inhaltsverzeichnis

1 Einleitung **3**
1.1 Vision . . . 3
1.2 Anforderungen . . . 4
1.3 Living Place Planung . . . 5
Kollaboratives Umfeld . . . 8
1.4 Fazit . . . 9

2 Computer Interaktion **9**
2.1 Einleitung . . . 9
2.2 TUIO . . . 10
2.3 Multitouch Klassenbibliothek . . . 11
2.4 Prototyp . . . 13
Erste Iteration . . . 14
Zweite Iteration . . . 14
Dritte Iteration . . . 15
Vierte Iteration . . . 16
2.5 Fazit . . . 17

3 Schluss **17**

Literatur **18**

1 Einleitung

1.1 Vision

Intelligente Umgebungen stellen seit geraumer Zeit eine große Herausforderung in der Forschung dar. Auch an der HAW beschäftigt sich ein Forschungsschwerpunkt mit Ambient Intelligence. Grundidee dabei ist, dass es nicht wenige Großrechner gibt, welche mit viel künstlicher Intelligenz überfrachtet werden, sondern viele kleine intelligente Computer, die sich in der direkten Umgebung des Menschen befinden. Ausgehend von Mark Weisers Vision (siehe Weiser (1991)) des Ubiquitous Computing wird seit 2002 an diesem Forschungsschwerpunkt gearbeitet. In erster Version wurde vorwiegend ein Labor für kollaboratives Arbeiten (Labor 1080) entwickelt, dass von Terry Winograds Vorarbeiten im Kontext des Forschungsbereiches IWork[1] an der Stanford University inspiriert wurde (siehe Abb. 1).

Abbildung 1: Standford iRoom (von Winograd (2003))

Typischerweise sind intelligente Umgebungen mit vielen Computern in unterschiedlichen Formen ausgestattet. Wichtiger Teilaspekt dabei ist, dass sich diese Computer nicht als graue Blechkisten manifestieren, sondern in der Umgebung aufgehen und somit nicht mehr zu sehen sind. Das so entstehende Konzept des calm oder dissapearing Computers wurde schon von Weiser vorausgesagt und ist auch in diesem Projekt ein verfolgtes Ziel.

[1] http://hci.stanford.com

Die Computer sind nicht nur Rechneneinheiten, sondern können über Sensorik ihre lokale Umgebung wahrnehmen. Die auf diese Weise erreichte Context Awareness, soll dazu genutzt werden situationsgerechte Handlungen durchzuführen. Ein essentieller Bestandteil dabei sind auch Feedback Mechanismen, die dem Benutzer auf subtile und intuitive Art und Weise den Systemzustand präsentieren[2].

Bisherige Ansätze intelligente Umgebungen zu Konzipieren beruhten auf Agentenmodellen, bei denen die Einrichtung autonom viele Eigenschaften bestimmt. Bei unserem Ansatz sollen viele Entscheidungen auch vom Benutzer getroffen werden, ohne diesen zu überfordern. Als Folge daraus wurden Interaktionsmetaphern, wie Natural und Tangible Interaction ein wichtiger Bestandteil des Projektes.
Erstes Ziel war, multimodale Interaktion zu entwickeln und auf Tauglichkeit zu überprüfen. Als Basis für weitere Arbeiten sollte ein Framework entstehen, das die Wiederverwendbarkeit der erstellten Softwarelösungen ermöglicht. Die eingesetzte Hardware soll dabei nicht störend auffallen und unnötig viele Aktionen eigenmächtig ausführen. Die Ubiquitous Computer oder auch Smart Devices sollen Teil der Umgebung sein und sich auf eine natürliche und einfache Art und Weise Bedienen lassen. Für eine genauere Beschreibung dieser Konzepte siehe AW2 Ausarbeitungen von Vogt und Rahimi (siehe Anwendungen 2 Ausarbeitungen (2009)).
Zweites Ziel war es eine Wissensbasis über die Vernetzung von Geräten in intelligenten Umgebungen aufzubauen. Resultierend daraus und aus den Erfahrungen mit dem Labor 1080, als Konferenzraum der Zukunft, soll eine Wohnumgebung der Zukunft geschaffen werden. Das als Living Place bezeichnete Wohnkonzept ist die erste Umsetzunng einer intelligenten Wohnumgebung an der HAW, welches für Realexperimente genutzt wird.

Das Breite Spektrum an Informationen und die Vielfalt der auftretenden Technologien und Teilaspekte innerhalb des Projektes, erfordert eine Aufteilung in verschiedene Teilbetrachtungen. Evaluation von Technologien, deren Verbindung und Integration, Anpassungen früherer Arbeiten und das Schaffen eines Grundverständnisses für den gesamten Themenkomplex sind Aufgaben die unter anderem im Rahmen des Projektes in Angriff genommen wurden. Viele der getätigten Aufgaben werden nachfolgenden geschildert und der daraus resultierende Zusammenhang zwischen diesen wird dann im Abschnitt 3 hergestellt.

1.2 Anforderungen

Im Rahmen des Konferenzraumes der Zukunft wurden schon einige Konzepte entwickelt und umgesetzt, die hier wiederverwendet werden sollen. So lassen sich bestehende Systeme nutzen, um schneller zu Ergebnissen zu gelangen. Im Gegenzug dazu werden diese Systeme in iterativen Prozessen verbessern. Wichtig dabei sind die vorhandenen Geräte, wie

[2] Denkbar sind hier adaptive Tapeten, welche das Raumambiente situationsgerecht anpassen.

der Multitouch Tisch, die Wand und die Powerwall. An diesen wurden schon viele Interaktionskonzepte umgesetzt und erprobt. Auch die schon geschriebene Software sollte möglichst genutzt werden, um den Workload gering zu halten.

Durch erste Diskussionen hat sich schnell ergeben, dass eine gemeinsame Entwicklungsumgebung unumgänglich ist. Um das zu forcieren wurde eine Einigung auf eine Programmiersprache und die dazu passende Entwicklungsumgebung gefunden. Hier haben sich C# als Sprache und Visual Studio als Umgebung unter Windows durchgesetzt.

Die Entscheidung fiel auf C#, da es eine objektorientierte Sprache ist, welche Java sehr ähnlich ist, aber unter Windows deutliche Performancevorteile bietet. Als Framework für Oberflächenprogrammierung eignet sich in dieser Umgebung die Windows Presentation Foundation (WPF), da diese native Fähigkeiten der Hardware optimal nutzt[3]. Als Vergleichsumgebung wurde Java mit JavaFX miteinbezogen, welches zu dem Zeitpunkt allerdings nur in einer sehr frühen Version zur Verfügung stand und aus performance Gesichtspunkten nicht mithalten konnte.

Da sich Java und C# in vielen Punkten ähnlich sind, stellt eine Umstellung kein größeres Problem dar, allerdings ist es für Einzelheiten doch wichtig genauere Kenntnisse über die Umgebung zu haben. Aus diesem Grund konnte mit Hilfe eines Microsoft Senior Student Partners[4] ein C# Workshop organisiert werden, in dem die Grundlagen von Visual Studio, C# und WPF vermittelt wurden. Die gute Atmosphäre des Workshops hat es zugelassen auch speziellere Themen der Oberflächenprogrammierung anzugehen, die sich einem Anfänger nicht erschließen.

Die Kommunikation und der Kontext innerhalb des Projektteams waren damit klar. Der Übergeordnete Kontext des Living Place wurde daraufhin aktiv mitgestaltet und geplant.

1.3 Living Place Planung

Ein Teilziel war es das Living Place mitzugestalten. Voraussetzung dafür ist allerdings das Wissen über verfügbare und kommende Technologien im Bereich der Haustechnik und Haushaltsgeräte. Hierzu wurde viel Zeit in Recherche und Evaluation von interessanten Technologien und Konzepten investiert. Desweiteren wurden einige Meetings mit Herstellern aus den Bereichen aktiv mitverfolgt und geplant.

[3] Als Test wurden hier mehrere Full HD Videos (1920x1080 Pixel) in einer Applikation gleichzeitig abgespielt. Die gewählte Kombination zeigte hierbei die beste Performance

[4] Kai Rosseburg

Da das Living Place u.a. auch als Usability Labor ausgeprägt sein wird, ist hierfür eine spezielle Ausstattung nötig. Erfahrungen aus dem bestehenden Usability Labor wurden zusammen mit der COMO Computer & Motion GmbH besprochen und ein entsprechendes Angebot konnte daraufhin entwickelt werden. Diese Ausstattung umfasst Kameras, Mikrofone und Aufnahmerechner. Es wurde eine Kamera von Como vorgeschlagen, welche Full HD aufnahmen ermöglicht und diese über Ethernet Schnittstelle bereitstellt. Diese Kamera wurde vom uns evaluiert und für den Einsatz im Living Place für geeignet befunden. Hierbei stellte sich heraus, dass die Kamera einen Mechanismus zur Bewegungserkennung besitzt, sich somit also auch zu Interaktionszwecken im Living Place einsetzten lässt[5].

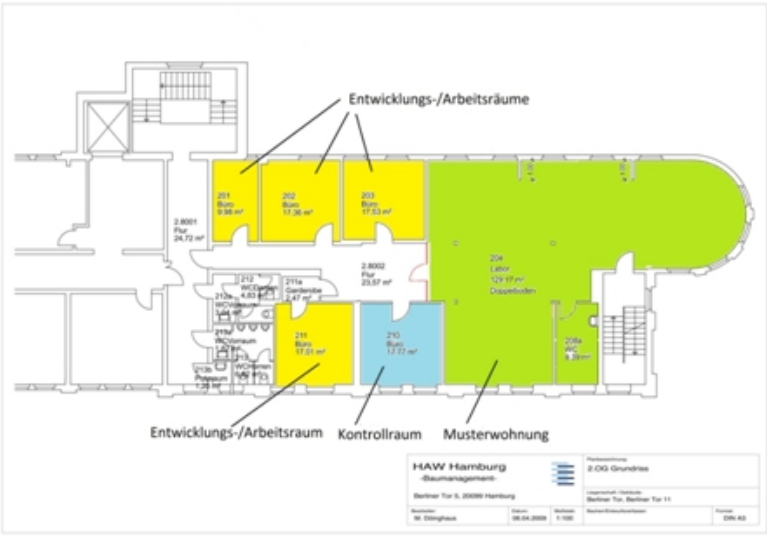

Abbildung 2: Aufbau des Living Place Hamburg

Im Rahmen der Planung wurde ein Meeting organisiert, bei dem alle beteiligten Studenten eingeladen wurden. Als erstes Meeting, diente es der Konkretisierung der Planung, so dass alle auf einem Stand sind. Zudem wurden so Teilnehmer, die sich mit ähnlichen Themengebieten beschäftigen, verbunden. Es wurde Wert darauf gelegt, dass jedem Studenten in dem Kontext eine klare Verstellung vermittelt wird, wie sich sein Thema in das Gesamtkonzept (siehe hierzu auch 2) integriert. Es wurden einzelne Projektverantwortliche ausgemacht, um die Kommunikation zu vereinfachen. Wichtiger Bestandteil dieses Meetings ist die Definition, dass nicht Technologie basiert, sondern Szenario basiert gearbeitet werden sollte. Die Entwicklung eines Szenarios wird damit in den Vordergrund gestellt, so dass Technologie Debatten weniger wichtig werden.

[5] Genauere Einzelheiten zur kamerabasierten Bewegungserkennung sind in Arne Bernins Projektbericht von SS2009 zu lesen (siehe ∗http://besteseitederwelt.de/projects)

Trotzdem sollen im Living Place viele unterschiedliche Technologien platz finden. Um eine reibungslose Kommunikationsplattform bereitstellen zu können, wurden unterschiedliche Konzepte zur Kommunikation von verschiedener Hardware aus der professionellen Haustechnik untersucht. Um ein besseres Bild von dem dort eingesetzten Bussystemumfeld zu bekommen wurde die Itenos Gmbh hinzugezogen. Bei einem Meeting vor Ort wurden unterschiedliche Bussysteme vorgeschlagen, die häufiger Einsatz finden. Hierzu gehören u.a. auch BACnet und KNX, welche in großem Maße verbaut werden. Auch Ethernet und drahtlose Kommunikation über ZigBee und WLAN wurden von in Betrachtetgezogen.

Im Bereich der Home Appliances (Haushaltsgeräte) gibt es seit 2003 die DLNA (Digital Living Network Alliance) Standards[6], welche sich mit der Vernetzung von Haushaltsgeräten beschäftigen. In dem Gremium beteiligen sich viele am Markt etablierte Firmen, wie Microsoft, Philips, Samsung und Toshiba. DLNA zertifizierte Geräte haben eine kabelgebundene Netzwerkschnittstelle (WLAN wird auch im Standard aufgenommen), über die sie per UPnP[7] (Universal Plug and Play) Kontakt zu anderen Geräten im Netzwerk aufnehmen können.
UPnP ist ein auf Strings basierendes Protokoll[8], welches heutzutage in vielen Routern eingesetzt wird um automatisches Port Forwarding zu ermöglichen.

Diese Recherchen wurden im Konferenzraum der Zukunft, also Labor 1080, gemacht. Das Labor war durch Schränke virtuell in zwei Abschnitte aufgeteilt. Zum einen in den „Showroom“, in dem laufende Shows Besuchern und Interessierten gezeigt wurden. Und zum anderen in den Entwicklerbereich, in dem Arbeitsplatzrechner aufgestellt sind. Die Arbeit wurde aber in beiden Bereichen ausgeführt und die Trennung hatte nur noch Beengung zur Folge. Aus diesem Grund und der Konzeption des Living Place als offener Raum wurden die Schränke zur Seite gestellt und so eine Art Großraumbüro geschaffen, welches sehr viel offener wirkt.
In Zusammenhang dazu wurde für die Powerwall ein Ambient Light entwickelt. Diese indirekte Beleuchtung dient einerseits dazu das Raumambiente durch passendes Licht zu verbessern, stellt aber auch ein Testbett für eine adaptive Tapete im Living Place dar. Hierzu wurden LED-Stripes hinter den Monitoren der Powerwall angebracht. Die Ansteuerung der LEDs wird über einen Arduino vorgenommen, der an einem Rechner angeschlossen ist. Dieser Rechner nimmt als Server die Anfragen aus dem lokalen Netzwerk entgegen und leitet diese aufbereitet an den Arduino weiter.
Die Verbindung all der Geräte im Labor und im Living Place erfordert ein Umfeld, auf dessen Basis diese, jetzt noch einzelnen Entitäten, kommunizieren können, um so ein homogenes Netz zu bilden.

[6] http://www.dlna.org
[7] http://www.upnp.org/
[8] In seiner Grundsubstanz ist UPnP, also ähnlich zu TUIO (s.a. 2.2)

Kollaboratives Umfeld

Eine Integration aller Geräte ist unumgänglich, um die das angestrebte Ziel, welches die Ambient Intelligence darstellt, zu erreichen. Als Basis dafür musste eine Middleware gefunden werden, die eine Kommunikation ermöglicht. Der Einsatz von Middleware als Kollaborationssoftware für zusammenarbeitende Menschen ist zwar grundsätzlich unterschiedlich zu deren Einsatz für kommunizierende Geräte, allerdings stellt sie eine gute Basis dar, die erweitert werden kann. Es wurden unterschiedliche Middleware und Kommunikationsplattformen evaluiert. Darunter gehörten unter Anderem auch der Eventheap, die Windows Communication Foundation, Google Wave und eine Eigenentwicklung auf Basis von Muliticast.

Im Rahmen der Arbeiten rund um den Konferenzraum der Zukunft und dem iFlat Labor, wurde der Eventheap aus Stanford als Middleware verwendet. Die Integration im Laufe des Projektes profitierte stark von den Vorarbeiten von Denis Holatz und den im Projekt stattfindenden Arbeiten von Andre Goldflamm (s. 2.4). Der Eventheap ermöglicht es mittels einer Blackboardarchitektur Events diverser Geräte entgegen zunehmen, zu verwalten und Geräten eine Art Abonnementendienst auf Events zur Verfügung zu stellen. Der Eventheap benutzt als Runtime die Java Virtual Mashine. Die Anbindung von Clients, welche nicht auf Java basieren stellte sich als schwierig heraus, da die Entwickler eine eigene Serialisierung von Objekten entwickelt haben, die eine Client Implementation in Java voraussetzt. Diese Lücke konnte mit Hilfe eines C# Wrapper geschlossen werden.

In der Vorlesung Technik und Technologie 1 wurde das Konzept der Live Distributed Objects[9] vorgestellt. Bei Live Distributed Objects wird „normal“ objektorientiert Programmiert, allerdings werden die Objekte im Netz mittels Multicast Techniken im Netz synchronisiert. So können Veränderung einzelner Objekte in Echtzeit wahrgenommen werden. Es stellte sich allerdings als sehr aufwändig heraus auf dieser Basis ein komplettes Kollaborationsumfeld zu entwickeln. Gerade die Einrichtung innerhalb der Entwicklungsumgebung und Einbindung der Funktionalitäten in ein Projekt stellten sich als zu aufwendig heraus. Die Konzepte und ersten Umsetzungen sind durchaus viel versprechend und sollten noch genauerer Prüfung und weiterer Beobachtung unterzogen werden.

Als weitere alternative wurde Google Wave betrachtet[10], eine online Kommunikations- und Kollaborationsplattform welche Google unter einer Open Source Lizenz anbietet. Eine Idee seitens Google ist die Bearbeitung von Dokumente mit mehreren Teilnehmern in Echtzeit. Als weiteres soll es, zumindest in einer Instanz von Google, ebenfalls als Kommunikationsplattform dienen. Es ist durch das Konzept von Robots und Widgets, Server- sowie Clientseitig, einfach erweiterbar. Er wird als Modul für Openfire[11] installiert, einem Echtzeit Kollaborati-

[9] http://liveobjects.cs.cornell.edu/
[10] Google Wave Federation Server http://wave.google.com
[11] http://www.igniterealtime.org/projects/openfire/

onsserver auf Basis des Jabber Protokolls[12].
Die Installation des Federation Servers stellte sich als unkompliziert heraus und es konnte mit rudimentären Clients ein Funktionstest durchgeführt werden. Die föderale Struktur der Server und die gute Erweiterbarkeit des Protokolls machen eine weitere Beobachtung der Entwicklung durchaus lohnenswert. Nach aktueller Einschätzung ist allerdings noch ein erheblicher Entwicklungsaufwand seitens Google von Nöten, um eine stabile Plattform bereitzustellen.

Der Federationserver wurde auf dem BladeCenter des Labors installiert, welcher als Persistenzebene für alle anderen Arbeiten dient. Für das Projekt wurden auf dem BladeCenter unterschiedliche Server installiert, um eine Serverinfrastruktur bereitszustellen. Hierzu gehören neben dem Google Wave Server auch LDAP, SVN und Web Server.

1.4 Fazit

Das formulierte Ziel eine intelligente Umgebung zu gestalten, erforderte sehr viel Recherche und Evaluationsarbeit. Die vielen Informationen in diesem Bereich machen die Bewertung der einzelnen Technologien und Konzepte noch schwieriger. Viele der angebotenen Ansätze führen in Sackgassen, weil die Konzepte nicht mehr weiterentwickelt werden oder die Zielsetzungen nach einiger Zeit zu stark abweichen. Sehr viele ähnliche Labore in Deutschland beschäftigen sich mehr mit dem Bereich des Ambient Assisted Living (AAL), wie z.B. das Bremen AAL Labor. Durch einen Besuch in Bremen konnte festgestellt werden, dass AAL nicht der richtige Ansatz für das Living Place darstellt. Es wurden auch viele Infrastrukturelle arbeiten im Laufe des Projektes durchgeführt, welche als Basis für die nachfolgend geschilderten Arbeiten dienen.

2 Computer Interaktion

2.1 Einleitung

Die Evaluation der verschiedenen Technologien erforderten im laufe des Projektes viel Zeit. Dazu gehörten zum einen Gespräche z.B. über den Eventheap und andere Technologien, als auch die Projektkoordinierung an sich. Im zuge der Kooperation mit der Firma Elektrosil GmbH Elektrosil (2009) bestand die Möglichkeit den weltweit ersten Prototypen eines entstehendes Produktes, dem DreaMTouch[13], der Firma Ciron GmbH Citron (2009) zu evaluieren.

[12] XML basiertes Kommunikationsprotokoll für Instant Messenger Dienste. Siehe http://de.wikipedia.org/wiki/Jabber.
[13] http://www.citron.de/index.php?id=127&L=0

Dieses Infrarot basierte Touchoverlay ermöglicht die Erkennung von 32 simultanen Berührungspunkten. Der große Vorteil liegt hierbei in der schlanken Bauweise und der Möglichkeit es auf beliebige Oberflächen anzubringen. Ebenfalls wird die Sichtbarkeit des Display nicht beeinträchtigt. Im Vorfeld wurden schon Absprachen über in Frage kommende Protokolle getroffen und wenige Spezifikationen bekannt gegeben. Im Labor konnten dann auf Basis des TUIO Protokolls, welches noch genauer beschrieben wird, erste Evaluation durchgeführt werden. Neben Hardwareproblemen, die durch einen vor Ort Besuch der Firma Citron gelöst wurden, konnten ein paar kleine Fehler in der Umsetzung des Protokoll gefunden und behoben werden. Durch den stetigen Austausch mit den Entwicklern von Citron gelang es eine Einschätzung der Hardware zu erhalten und erste Testapplikationen zu entwickeln. Da die Ergebnisse des Prototypen in einer so frühen Phase der Entwicklung überzeugen waren und das eingesetzte Protokoll ein offener Standard ist, stellt er aller Voraussicht nach in seiner endgültigen Fassung eine ernst zu nehmende Alternative zu aktuell verfügbarer Multitouch Hardware dar. Folge dieser Evauluation war die Einarbeitung in das TUIO Protokoll und die Anpassung der aktuellen Klassenbibliothek für Multitouch an TUIO Daten.

2.2 TUIO

Mit der Zielsetzung die aktuellen Multitouch Elemente des entstandenen Frameworks möglichst Hardware unabhängig zu gestalten bot sich die Verwendung von TUIO an. Das TUIO Protokoll ist eine generelle Beschreibung und Api von Tangible und Multitouch eingaben. Entstanden im Rahmen des reactvision[14] Projektes (Abb. 3 und 4) entwickelt sich TUIO zu einem sehr weit verbreiteten Standard im Bereich dieser Eingabegeräten. So nutzt Community Core Vision[15], eine der meist genutzten Software für Rückprojektionsbasierte Multitouchgeräte, TUIO als Protokoll. Aber auch Neuentwicklungen wie der oben beschriebende Multitouch Prototyp der Firma Citron, setzen auf den offenen Standard.
TUIO nutzt als Kommunikationsschnittstelle das normale Netzwerkinterface und bedient sich dabei OSC[16] als Austauschformat. TUIO beschreibt in seiner Spezifikation drei verschiedene Profile bietet aber auch die Möglichkeit eigene Profile zu definieren. Die zwei, für das Projekt, wichtigsten sind hierbei das *2D*- und *3D Interactive Surfaces* Profil. Das 2D-Profil beinhaltet für die erkannten Punkte, x und y Position, den Winkel der Ausrichtung, Bewegungsvektorn in x und Y Richtung und die Beschleunigungen für die verschiedenen Werte. Im 3D Profil werden zusätzlich noch Höhen Informationen mit verarbeitet, was es ermöglicht Aktionen oberhalb der Oberfläche zu beschreiben. Alle erkannten Punkte sowie Objekte haben eine eindeutige ID, welche nach Möglichkeit[17] bis zum loslassen gleich bleibt. Ebenfalls wird

[14] http://reactivision.sourceforge.net/
[15] http://ccv.nuigroup.com/
[16] Open Sound Control
[17] Es ist vom genutzten Punkte-Tracking abhängig.

zwischen sogenannten *cursor* (siehe Abb. 4) und *objects* (siehe Abb. 3) unterschieden. *Objects* repräsentieren dabei einen Tangible Anteil, *cursor* stellen eine Tocheingabe dar.
Neben der Vielfältigkeit, die das Protokoll ermöglicht, ist eine sehr aktive Entwicklergemeinschaft in das Projekt involviert. Referenzimplementationen in verschiedenen Programmiersprachen ermöglichen einen einfachen Einsatz in Projekten. Ebenso stehen diverse Simulatoren und Brücken zu nicht TUIO basierender Hardware[18] bereit.
Durch die positiv zu bewertenden Eigenschaften dieses Protokolls wurden die auf WPF

Abbildung 3: Interaktion mit dem Reactable (Bild von http://www.reactable.com/)

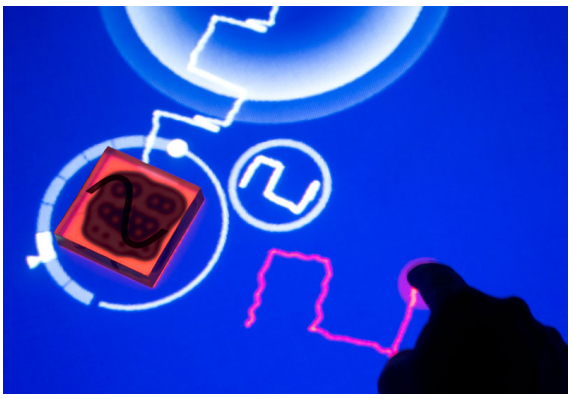

Abbildung 4: Fingereingabe beim Reactable (Bild von http://www.reactable.com/)

basierenden Controls um eine Interpretation von TUIO Daten erweitert. Basis ist die TUIO Client Referenzimplementation in C#. Bei der Implementation und und Einbindung fiel ein Fehler bei der Winkelberechnung der einzelnen Punkte in der Implementation auf, welche den Entwicklern mitgeteilt und somit behoben werden konnten. Im Zuge des Masterprojektes konnten die bestehenden Multitouch Klassenbibliotheken, welche im Rahmen von der Bachelorarbeit Rahimi und Vogt (2008) entwickelt, und später auf C# portiert wurden, um die Nutzung von TUIO als Input erweitert werden.

2.3 Multitouch Klassenbibliothek

Die aktuelle Version der Klassenbibliothek basiert auf WPF und wurde in C# implementiert. Sie unterstützt Daten Input von IR-Touch Infrarot Rahmen und TUIO Daten. Aufgrund der schon oftmals beschrieben Schwierigkeiten (siehe z.B. Gehn (2007)) war es bis jetzt nicht möglich eine TUIO Brücke für den IR Touch Rahmen zu erstellen. So müssen zur Zeit

[18] Es existieren unter andrem Brücken für den Microsoft Surface oder das Apple Touchpad

unterschiedliche Interfaces bei den unterschiedlichen Eingabedaten verwendet werden. Die Lösungen für den IR Touchrahmen wurden schon ausführlich in anderen Arbeiten beschrieben, so dass ergänzend hier nur die aktuellen Änderungen vorgestellt werden.

Die verwendete TUIO Implementierung verwaltet intern, dass Punkten aktuelle Daten zugeordnet werden und beinhaltet eine Historie über die Bewegung. Es wurde ein Adapter erstellt, der nach einer kompletten Aktualisierung der Touchdaten innerhalb der TUIO Client Implementation, die Daten den berührten Elementen zuordnet. Nachdem alle Daten zugeordnet sind berechnet jedes Element selbständig die aktuellen Veränderungen und reagiert entsprechend der Interpretierten Manipulation. Alle entstanden Elemente erben von einer Basisklassse, welche ein Standard verhalten implementiert. So Erbt ein Elemente Grund verhalten wie Bewegen, Skalieren und vergleichbares. Soll ein Element ein abweichendes Verhalten bei bestimmten Gesten oder erkannten Manipulationen haben, können die entsprechenden Methoden Überschrieben werden und ein gesondertes verhalten implementiert werden.

Im Laufe des Projektes sind somit verschiedenste wiederverwendbare Multitouch Elemente entstanden. Grundfunktionalitäten sind Bewegen, Skalieren, Rotieren und Drehen (um die Z-Achse), welche sich über das setzen einer Eigenschaft für jedes Element aktivieren und deaktivieren lassen .

Image Zeigt ein per Pfad angegebenes Bild an.

Video Zeigt ein per Pfad angegebenes Video und dessen Fortschritt an. Hiermit lassen sich aber auch Tondateien abspielen.

Browser Ermöglicht es eine Webseite anzuzeigen. Leider ist das Elemente aufgrund von WPF internen Implementation nur begrenzt einsetzbar. Durch die Interne Implementierung seitens Microsofts lässt sich das Element nicht in den Hintergrund verschieben. Somit bleibt es immer im Vordergrund und überdeckt andere Elemente. Eine Lösung wurde bis heute nicht gefunden und scheint sich auch nicht ohne ein Nachbesserung von Microsoft realisieren lassen. Für ein Proof of Concept reicht die aktuelle Fassung allerdings aus.

Button Da keine Betriebssystem Events emuliert werden sollten musste ein eigener Button erstellt werden, welcher auf Multitouch Eingabe funktioniert.

TimedButton Um ungewollte Eingaben zu ermöglichen sollte ein Button erstellt werden, der erst nach einer gewissen Zeit permanenten drückens aktiviert wird.

Flicker Das Flicker Element ermöglicht es mittels einer Wischgeste, durch verschiedene Bilder zu blättern.

InfoPanel Das Infopanel im Vergleich kann andere Elemente beinhalten und ermöglicht es durch die Elemente blättern zu können. Mittels einer Wischgeste kann das nächste oder vorherige Element angezeigt werden. Die Transition zwischen den Elementen wird mittels einer Animation deutlich gemacht. Hier gab es Schwierigkeiten in Bezug auf die Animation, einige Elemente ließen sich nicht der bisher genutzten Methode animieren, so dass auf eine Bibliothek mit Shaderbasierte Animation zurückgegriffen wurde.

Message Das Message Control, ist eine erste Kombination aus Multitouch und Speech. Es lassen sich die Gundfunktionalitäten von der Multitouch bedienungen durchführen. Beim erstellen erlaubt es aber ein Diktat welches dann als in dem Element text angezeigt wird.

Scroll Das Scroll Element stell eine einfach Textbox da, bei der mit 2 Fingern durch den Text navigiert werden kann.Die Geste ist dabei an die von Apple MacBook eingeführte Scroll Geste angelehnt.

Durch den gegebenen Rahmen seitens WPF und der entwickelten Struktur der Klassenbiblithek, lassen sich somit schnell neue Controls entwickeln. So können sich schnell diverse Standard Controls von Windows adaptieren oder komplette Eigenentwicklungen in einem akzeptablen Zeitrahmen umsetzen.

2.4 Prototyp

Um die gewonnenen Erfahrungen der Recherche überprüfen zu können und eine Einschätzung der Technologien zu bekommen wurde ein Proof of Concept erstellt. Hierbei sollte nicht nur das Einbinden der Infrastruktur betrachtet werden sondern auch im speziellen die Interaktion untersucht werden. Es wurde in der Gruppe entschieden ein interaktives Interface zu schaffen, welches die Möglichkeit bietet Informationen zwischen Computern auszutauschen und diese zu verarbeiten. Als Basis der User- und Service-Discovery wurde der Event heap gewählt. Da aber kein Adapter für C# zur Verfügung stand wurde übergangsweise die selbst entwickelte Lösung auf Basis von Multicast und Objektserialisierung verwendet. Zum Ende des Semesters wurde auch die Anbindung an den Event Heap von André Goldflam bereit gestellt so das ein erster Test kurz vorm Ende des ersten Teil vom Projekt mit dem Event Heap durchgeführt werden konnte. Die Implementation der aktuellen Funktionalitäten steht allerdings noch aus.
In erster Iteration wurde noch davon ausgegangen, eine Information über Position der Nutzer und der Geräte im Raum zu haben. Da dies ein zu großer zusätzlicher Aufwand gewesen wäre, welcher sich zwar realisieren ließe aber nicht Hauptbestandteil in erster Aufbaustufe war, wurde in erster Linie das Zusammenspiel zwischen Multitouch und Speech getestet.

Ein mit einbeziehen von Positionen wurde nicht verworfen aber aus Zeitgründen verschoben.

Erste Iteration

In einer Diskussion der beteiligten wurde beschlossen, nach dem Vorbild von James Bond (s. Abb. 5) Bilder nahtlos zwischen Rechner auszutauschen. Aktuell verfügbare Rechner wurden als farbige Rechtecke an den Seiten repräsentiert. Da zu diesem Zeitpunkt noch keine Art der dynamischen Bentuzerfindung im Netzwerk verfügbar war, wurden die Rechner fest einprogrammiert gewesen. Bilder konnten dann mittels eines Speech befehlt geöffnet und dann weiter mittels Toucheingabe bearbeitet werden. Beim Schieben des Bildes über die Rechtecke wurde das Bild dem der Farbe zugeordneten Rechner gesandt. Zu Erkennung, ob ein Bild über einem entsprechenden Rechteck liegt wurde eine temporäre Lösung verwendet, da WPF zwar eine Vielzahl an Funktionalitäten mitbringt aber es mehr Aufwand war diese Erkennung zu realisieren als zuvor angenommen. Das Aussehen der ersten Iteration ist Abbildung 6 zu sehen.

Abbildung 5: Multitouch Interaktion aus James Bond, ein Quantum Trost (Screenshot aus http://www.vimeo.com/4765993)

Zweite Iteration

Nach erster Betrachtung der in der ersten Iteration gesammelten Ergbnisse, wurde schnell deutlich, dass eine dynamische User discovery nötig ist. Die Anbindung an den Eventheap

Abbildung 6: Erste Iteration

konnte zu der Zeit aber noch nicht vorgenommen werden, da es sich nicht als trivial gestaltete eine Anbindung zu schaffen. Im Zuge dessen wurde eine einfache Kommunikationsmöglichkeit, auf Basis von Multicast, zwischen diesen Systemen realisiert, was dann für den Datenaustausch aber auch zur Nutzerfindung verwendet wurde. Zur Erkennung und bekanntmachung von Nutzern wurde einfacher Handshake zwischen den im Netzwerk befindlichen Systemen entwickelt. Meldet sich ein Benutzer an, versendet er eine Nachricht mit Kennzeichnung „Hello“ und seinem Rechnernamen ins Netzwerk. Ebenfalls wird ein „getusers“ geschickt, was alle anderen auffordert ihre Kennung, unabhängig davon ob sie sich schon einmal gemeldet haben, zu versenden. Dieses Vorgehen ermöglicht es einerseits, dass der neue Nutzer über seine „Nachbarn“ informiert wird und anderseits schon bekannte Rechner ihre Listen auf Aktualität prüfen konnten. Zudem wurde gezielt drauf geachtet, dass keinerlei Interna dieser Netzwerk Ebene Einfluss auf die eigentliche Anwendung haben sondern nur mittels austauschbarer Schnittstellen zugegriffen wird, um ein späteres Einbinden des Eventheaps zu ermöglichen.
Ebenfalls wurde ein Control eingebunden, welches es ermöglicht MP3 per Sprachbefehl zu laden und diese zu kontrollieren. Mittels einfacher Sprachkonstrukte wie „open album xyz“ wurde das entsprechende Album geladen der Playlist hinzugefügt. Mittels weiteren Sprach befehlen, wie „next“, „play“ oder „stop“ konnte Abspielen dann kontrolliert werden.

Dritte Iteration

In der dritten Iteration wurde das „Message“ Element erstellt. Der Nutzer kann das Element erstellen einen Text diktieren und es an andere Nutzer schicken. Beim empfangen dieser Message wird diese dann vorgelesen. Im zuge dessen wurde ein erstes Review der aktuellen Oberfläche und Mischung von Touch und Spracheingabe vorgenommen. Wurden bis

zu dem Zeitpunkt fast alle Elemente automatisch (beim erscheinen eines neuen Benutzers) oder per Sprachbefehl gesteuert hat sich mit der Zeit gezeigt, dass die Spracheingabe für einfach Befehle zu lange dauert und zu ungenau, gerade bei untrainierten Sprachprofilen, ist. So wurden an der Seite Buttons erstellt welche einen schnelleren Zugriff auf Elemente erlauben. Das öffnen der „Message“ wurde somit über den Button gelöst. Die Eingabe des Textes wurde weiterhin mit der Spracheingabe realisiert, da sich Bildschirmtastaturen schon in der Vergangenheit, gerade bei langen Texten, als ungeeignet herrausgestellt haben.
Ein weiter Punkt welcher geändert wurde ist die Repräsentation der anderen Teilnehmer. Anstatt sie an den Rändern als Rechtecke anzuordnen[19], wurden Sie nun als verschiebbare Elemente mit der Rechnerbezeichnung angezeigt. Für die Experimente und Versuche erwies sich dies als bessere Methode, da die statische Lösung oftmals im Weg war somit die Interaktion störte und ein ungewolltes Verhalten provozierte.

Vierte Iteration

In der letzten Iteration der Projektphase, konnten erstmals der Eventheap effektiv in das Projekt eingebunden werden. So war es möglich Events zu generieren, zu abonnieren und zu empfangen. Die Umsetzung, der aktuellen Funktionalitäten die über den Event Heap gelöst werden sollen, steht noch aus. Ein paar Detail Verbesserungen wurden noch im Bereich der Überschneidungserkennung zweier Objekte gemacht. So konnte ein wesentlich eleganterer Weg gefunden werden zu erkennen ob ein Objekt z.B. zu einem anderen Rechner geschickt werden soll. Eine prozentuale Angabe der Überschneidung wird berechnet und ein mittels eines Schwellwert wird ermittelt ob eine ausreichende Überdeckung vorliegt. Probleme wurden hierbei aber beim Vergrößern von Bildern über den ganzen Bildschirm festgestellt. Ein Ansatz das Problem zu lösen ist die Betrachtung des letzten Bewegungsvektors der Fingereingabe. Eine Bewegung auf die symbolischen Repräsentation eines Rechner zu, könnte die Intention zeigen das gerade angefasste Element zu verschicken. Alternativ könnte man über den Einsatz eines Eye-Trackers nachdenken, da davon auszugehen ist, dass ein Nutzer ein Element anschaut wenn er es benutzen will.
Eine der letzten Änderungen war das einführen eines Mülleimer Elements. Bei etwas länger andauernden Tests fiel auf, dass die Möglichkeit fehle Elemente zu löschen. Dies wurde erstmal mittels eines Elementes, welches wieder mit Überschneidung zweier Elemente funktioniert, realisiert. Ob ein Anfassen des Elements und ein Sprachkommando „delete“ nicht eine bessere oder intuitivere Interaktion darstellt bleibt zu untersuchen. Die aktuelle Oberflächengestaltung ist in Abbildung 7 dargestellt.

[19] Es war ursprünglich ein Ziel die Repräsentation der Rechner entsprechend der Positionen und deren Ausrichtung zueinander anzupassen.

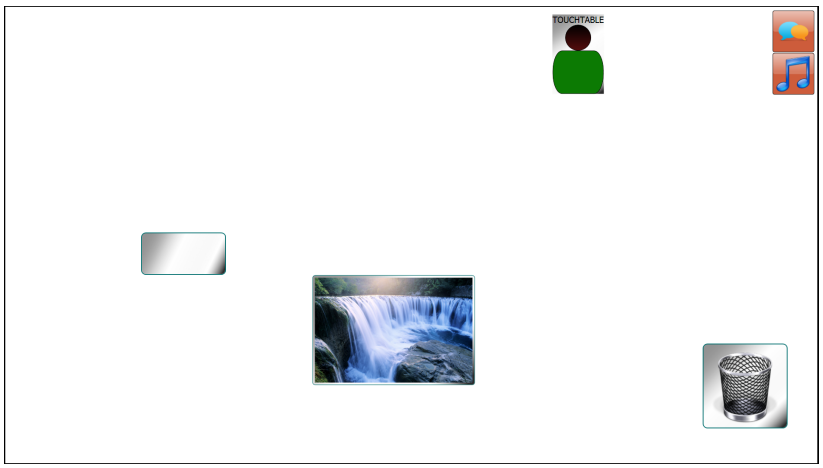

Abbildung 7: Vierte Iteration

2.5 Fazit

Die gleichzeitige Erstellung des Prototyps neben der Recherche hat sehr geholfen ein gutes Verständnis für die Technologien zu erlangen welche eingesetzt werden sollen. Problemstellungen wurden deutlich und ein gemeinsame Wissensbasis und Erfahrungen konnten bei den beteiligten gesammelt werden. Ein erstes Gefühl für die Kombination von Speech in Verbindung mit Multitouch konnte erlangt werden. Ebenfalls wurde deutlicher welcher Aufwand nötig ist gewisse Technologien zu verwenden, oder welche Mittel noch nötig sind um die Vision zu erreichen.

3 Schluss

Die Evaluation und Vorstellung der einzelnen Teilaspekte macht deutlich welches Ausmaß die Planung und Konzipierung einer intelligenten Umgebung hat.
So wurden in diesem Masterprojekt zwei Ziele erreicht. Zum einen konnte eine große Wissensbasis über Smart Homes und deren einzelne Bestandteile erlangt werden. Diese Wissensbasis stellt nun ein Fundament für die zukünftige technologische Ausstattung des Living Place dar. Gerade das Wissen über die Ansteuerung von heute aus dem Handel erhältlicher Hardware mittels z.B. UPnP oder Bussystemen aus dem Haustechnikbereich ermöglicht die Beurteilung zukünftiger Anschaffungen.

Zum anderen wurde mit ersten Experimenten untersucht wie multimodale Interaktion funktioniert. Es konnte festgestellt werden wie Technologien ineinander greifen können. Als Resultat daraus ist ein Framework entstanden auf dessen Basis weitere Arbeiten ausgeführt werden

können. Die Wahl eines offenen Standards als Schnittstelle machen es auch für andere Interaktionkonzepte einsetzbar. So konnten neben der Projektapplikation, auch weitere Proof of Concept Appliaktion im Bereich der Indoornavigation und der Anlagensteuerung entwickelt werden. Dies zeigt die Flexibilität und Erweiterbarkeit des Frameworks.

Auch mit sehr vielen Schwierigkeiten hatte das Projektteam zu kämpfen, aus diesen wurden viele wertvolle Erfahrungen gewonnen, welche im nächsten Projekt angewendet werden können.
Gerade bei der Zusammenführung der Technologien im Kontext von Smart Homes bzw. Smart Enviroments wurden oftmals kleine Fehler und Details der einzelnen Bestandteile zu großen Ärgernissen. Trotz aller Widrigkeiten wurde eine Basis geschaffen, anhand derer Interaktionen mit Smart Devices, wie sie im Living Place auftauchen werden, untersucht werden können.

Der Aufwand die dafür nötigen unterschiedlichen Technologien in diesem Kontext zu evaluieren ist dabei nicht zu unterschätzen und hat auch in diesem Projekt viel Zeit in Anspruch genommen. Die Vielfalt an Informationen ist einerseits ein guter Einstiegspunkt, andererseits ist es aber auch schwierig eine solche Vielfalt zu Bewerten und die richtigen Schlüsse zu ziehen.

Da Teil der Zielsetzung die Wiederverwendung und iterative Verbesserung vorhandener Hard- und Software war, haben sich diverse Anforderungen ergeben. Die vorhandenen Technologien aus dem iFlat und dem Office Bereich aufzuarbeiten, so dass eine homogene Plattform entsteht waren wesentlicher Bestandteil dieses Projektes. Die Integration vom Home Office, als Arbeitsumgebung innerhalb der Wohnumgebung, erfordern u. a. weitere Untersuchungen von Kollaborationsplattformen, wie Google Wave.

Durch das gesammelte Wissen ist es uns nun möglich das Living Place aktiv mitzugestalten. Zukünftige Ziele im Master Projekt 2 sind nun die Erstaustattung des Living Place Hamburg mit aufzubauen und somit eine Infrastruktur auf Basis der in diesem Bericht vorgestellten Technologien für zukünftige Projekte im Living Place zu schaffen. Das erstellte Multitouch Framework sollte um weitere Funktionalitäten ergänzt werden. Denkbar ist hier eine Erweiterung der Elemente und die Integration in die Eventheap Struktur. Ebenso wurden im Laufe des Projektes Überlegung zu Tangible Computing vertieft, so dass eine Einbeziehung in die Betrachtung von multimodaler Interaktion sinnvoll erscheint.

Literatur

[Anwendungen 2 Ausarbeitungen 2009] ANWENDUNGEN 2 AUSARBEITUNGEN: *Anwendungen 2 Ausarbeitungen SoSe 2009*. Webseite. 2009. – URL http://users.informatik.haw-hamburg.de/~ubicomp/projects.html

[Citron 2009] CITRON: *Citron GmbH*. Webseite. 2009. – URL http://www.citron.de/. – Letzter Aufruf am 31. August 2009

[Elektrosil 2009] ELEKTROSIL: *Elektrosil GmbH*. Webseite. 2009. – URL http://www.elektrosil.com/. – Letzter Aufruf am 31. August 2008

[Gehn 2007] GEHN, Stefan: *Evaluation einer infrarotbasierten Multitouchhardware*. Projektbericht an der Hochschule für Angewandte Wissenschaften Hamburg. 2007. – URL http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08-proj/gehn/report.pdf

[Rahimi und Vogt 2008] RAHIMI, Mohammadali ; VOGT, Matthias: Gestenbasierte Computerinteraktion auf Basis von Multitouch-Technologie / Hochschule für Angewandte Wissenschaften Hamburg. 2008. – Bachelorarbeit

[Weiser 1991] WEISER, Mark: The Computer for the Twenty-First Century. In: *Scientific American* 265 (1991), S. 94–104

[Winograd 2003] WINOGRAD, Terry: *Stanford Interactive Workspaces Project Overview*. Webseite. 2003. – URL http://iwork.stanford.edu/. – Letzter Aufruf am 14. August 2008