



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Seminarausarbeitung

Andreas Herglotz

Community Driven Development
and Social Networking
Becomes Pervasive

Andreas Herglotz
Community Driven Development
and Social Networking
Becomes Pervasive

Ausarbeitung im Rahmen des Seminars
im Studiengang Informatik (M. Sc.)
am Studiendepartment Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Gutachter : Prof. Dr. Kai von Luck und Prof. Dr. Bernd Schwarz

Abgegeben am 28. Februar 2008

Andreas Herglotz

Thema der Seminararbeit

Community Driven Development and Social Networking Becomes Pervasive

Stichworte

Community Driven Development, Social Networking, Pervasive Gaming, mobile Computing

Kurzzusammenfassung

Diese Ausarbeitung entspricht der Vision der folgenden Masterarbeit. Der generelle Kontext dabei ist Pervasive Gaming. Das Ziel ist es Social Networking und die Möglichkeit, Inhalt von einer Community schaffen zu lassen, in den Pervasive Gaming Bereich zu portieren. Dazu soll ein Messenger entwickelt werden, der dies ermöglicht. Dies wird am Beispiel von Android umgesetzt.

Andreas Herglotz

Title of the paper

Community Driven Development and Social Networking Becomes Pervasive

Keywords

Community Driven Development, Social Networking, Pervasive Gaming, mobile Computing

Abstract

This paper is the vision of my master thesis. In general it is about Pervasive Gaming. Social Networking and content created by a community should be transported to Pervasive Gaming. Therefore will be a messenger developed. It will be realised with the new Android platform.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motivation	1
1.2. Zielsetzung	1
2. Vision	2
2.1. Szenarien	2
2.2. Pervasive Community Messenger (PCM)	3
3. Aspekte des Messengers	5
3.1. Ubiquitous Computing	5
3.2. Social Networking	5
3.3. Community Driven Development	6
3.4. Fazit	6
4. Vorgehen	7
4.1. Analyse	7
4.1.1. Anforderungen	7
4.1.2. Android Framework	8
4.2. Entwurf	9
4.2.1. Architektur	9
4.2.2. Prototyping	10
4.2.3. Benutzerführung	10
4.3. Risiken	11
5. Zusammenfassung	12
5.1. Fazit	12
5.2. Ausblick	12
Literaturverzeichnis	13
A. Android	15

1. Einleitung

Diese Ausarbeitung entspricht der Vision für die Masterarbeit. Dieses Kapitel beschreibt zuerst die zugrunde liegende Motivation und dann die Zielsetzung.

1.1. Motivation

Spiele sind ein wichtiger Teil dieser Gesellschaft, sie machen Spaß, ermöglichen neue Fertigkeiten zu erlernen, treiben Kommunikation und interkulturellen Austausch voran, etc. Dies und die Tatsache, dass der derzeitige Stand der mobilen Technik und Technologie weit genug fortgeschritten und verbreitet ist, um ernsthaft über Pervasive Gaming nachzudenken, gaben den Anlass sich mit dieser relativ neuen Art des Spielens näher auseinander zusetzen. In diesem Zusammenhang lohnt sich ein Blick auf das Gesetz von [Moore \(Jan 1998\)](#), das besagt, dass sich alle 18 Monate die Leistungsfähigkeit von Prozessoren und Speichern (bzw. seine Verkleinerung und Verbilligung bei gleicher Leistungsfähigkeit) verdoppelt. Dies zeigt wie schnell sich neuere leistungsfähigere Geräte auf dem Markt zu akzeptablen Preisen durchsetzen und damit die Verbreitung von Pervasive Gaming mit unterschiedlichen Einsatzziele vereinfachen können.

1.2. Zielsetzung

Auf dieser Basis ist es das Ziel Social Networking und Community Driven Development in den Alltag sinnvoll zu integrieren. Die Hypothese ist, dass mit heutiger Technik überall und zu jeder Zeit Social Networking und Community Driven Development stattfinden kann. Das heißt, dass man überall zu jeder Zeit Orte mit Zusatzinformationen versehen kann. Beispielsweise wurde ein neues Cafe besucht, die Begeisterung ist groß und soll anderen mitgeteilt werden - man nimmt einfach das bereits vorhandene Handy und tut dies. Gleiches gilt auch für Social Networking, das bisher auf rein virtuelle Treffen vor dem PC beschränkt war, sich aber mit aktueller Technik überall umsetzen und sogar bis hin zu realen Treffen erweitern lässt. Im Folgenden wird die Vision näher ausgeführt und beschrieben wie diese umgesetzt werden kann und was dabei zu berücksichtigen ist.

2. Vision

Dies Kapitel stellt kurz drei verschiedene Szenarien vor, um dann zu zeigen, dass diese verschiedenen Szenarien alle mit dem generischen Pervasive Community Messenger umsetzbar sind.

2.1. Szenarien

Messe

Bei diesem Szenario befindet man sich indoor auf einer großen Messe.

Problemstellung Zumeist sind große Messen unübersichtlich, man weiß nicht wo sich welcher Stand befindet und vor allem weiß man nicht, welcher Stand besonders interessant ist oder wo ein spannendes Event stattfindet.

Lösung Dem Besucher und auch dem Aussteller ist es möglich die Stände zu beschreiben, wobei für den Besucher zusätzlich die Möglichkeit besteht die Stände zu bewerten. So kann man anhand aktueller Tipps von Gleichgesinnten und der Möglichkeit spannende Events über einen Newsticker mitgeteilt zu bekommen besser herausfinden was einen wirklich interessiert und wann bzw. wo es stattfindet. Als weitere Option kann man sich und gegebenenfalls andere auf einer Karte finden.

Flirt Maschine

Hierbei handelt es sich um eine Möglichkeit bei Flirtevents das Flirten aktiv und zielorientiert zu unterstützen, bzw. die erste Hürde zu verringern. Dieses Szenario dient als Outdooranwendung beispielsweise in einem Beachclub.

Problemstellung Das Problem wildfremde Menschen anzusprechen ist weit verbreitet. Vor allem wenn dahinter eine Intention wie das Kennenlernen steht, da in diesem Fall die Chance zurückgewiesen zu werden auf Grund einfacher Tatsachen wie falscher Typ, andere Interessen oder ähnliches relativ groß ist.

Lösung Die Flirt Maschine übernimmt dabei folgende Aufgaben: Sie überprüft im Hintergrund die auf den Handys hinterlegten Profile der Personen, die sich in einer definierbaren Reichweite befinden, auf Gemeinsamkeiten. Sind die Gemeinsamkeiten groß genug erscheint sofort eine Nachricht auf dem Handy. Damit wurde die erste Hemmschwelle überwunden. Nun kann man weitere Aktivitäten über das Handy ausführen oder direkten Kontakt herstellen, wenn beide Personen damit einverstanden sind.

Reeperbahn

Dies Szenario findet sowohl indoor als auch outdoor statt und bezieht sich auf die berühmte Partymeile in Hamburg.

Problemstellung Es handelt sich hierbei um einen relativ großen Bereich mit einer Vielzahl an Clubs bei denen man schnell den Überblick verliert und ohne sich vorher vernünftig zu informieren auch nicht weiß, wo gleich die spannenden Events stattfinden.

Lösung Auf dem Weg zur Partymeile updated man diese Applikation und bekommt dadurch sämtliche interessanten Informationen, die sich selbstverständlich filtern lassen. Diese Informationen bestehen aus einer Karte, um zu sehen wo man sich auf der Reeperbahn aufhält und wo sich Freunde aus der Freundesliste aufhalten. Clubs können ihre Events dort einstellen und vor allem können die Benutzer ihre Tipps dazu einstellen und die Clubs bzw. Veranstaltungen bewerten. Dadurch weiß man genau wann man wo sein sollte.

2.2. Pervasive Community Messenger (PCM)

Die Quintessenz aus den drei vorgestellten Szenarien, die direkt nichts miteinander zu tun haben ist die, dass sich trotz ihrer Unterschiede durch eine entsprechende Abstraktion ein Konsens finden lässt. Diese Szenarien lassen sich auf eine Anwendung abbilden. Folgende Tabelle 2.1 verdeutlicht dies.

	Exhibition	Flirt-Machine	Reeperbahn	Community Messenger
Indoor / Outdoor	Indoor	Outdoor	Hybrid	Hybrid
Location-Services	Bluetooth / WLAN / Barcodes	GPS	GPS / WLAN / Bluetooth / Barcodes	GPS / WLAN / Bluetooth / Barcodes
Communication Service	Nice to Have	Essential (Direct)	Nice to Have	Essential
Description + Rating Service for Locations	Essential	Nice to Have	Essential	Essential
News Service	Essential	Essential	Essential	Essential
Server Component	Essential	Essential	Essential	Essential

Abbildung 2.1.: Abbildung verschiedener Szenarien auf den generischen Messenger

Wie man der Grafik entnehmen kann sind also die Mindestanforderungen des PCM eine Indoor- sowie Outdoorfunktionalität. Man muss die Position bestimmen können. Relativ oder absolut ist dabei erstmal zweitrangig. Wichtig ist, dass das Anhand der zur Verfügung stehenden Technologie funktionieren soll. Selbstverständlich muss man dabei mit anderen kommunizieren können und auch das Erstellen von Inhalt und Bewerten ist essentiell, damit das Projekt eine Eigendynamik entwickeln kann. Wenn etwas Neues oder Interessantes passiert, muss dem Nutzer dies auch sofort mitgeteilt werden, da dieser nicht selbst in regelmäßigen Abständen alles nach Neuigkeiten durchsuchen möchte. Eine Serverkomponente, auf der zum Beispiel der geschaffene Inhalt zur Verfügung gestellt wird, ist ebenfalls unentbehrlich.

3. Aspekte des Messengers

An dieser Stelle werden kurz die wichtigen Aspekte des PCM definiert und der Bezug dazu hergestellt.

3.1. Ubiquitous Computing

Definition Ubiquitous Computing ist ein von Mark Weiser geprägter Begriff. Dieser bezieht sich darauf, dass heutzutage mehrere Computer einen Menschen unterstützen und bezeichnet das Gegenteil von der virtuellen Realität. Das Entscheidende daran ist, dass die Computer mit den Menschen leben sollen ohne diese zu behindern oder irgendwie einzuschränken. Dies ist zweifellos eine große Herausforderung, die nicht nur an die Informatik sondern auch andere Fachbereiche wie die Sozialwissenschaften gestellt wird. Folgendes Zitat von Mark Weiser bringt das noch einmal auf den Punkt:

The most profound technologies are those who disappear. They weave themselves into fabric of everyday life until they are indistinguishable from it.

Projektbezug Genau das was Ubiquitous Computing beschreibt bildet die Grundlage dafür, dass diese Vision funktionieren kann. Denn nur wenn jeder ein solches Gerät (Handy, Smartphone oder PDA) bereits besitzt und die Benutzung intuitiv ist, weil sich das Gerät in das Leben bereits als fester Bestandteil integriert hat, lässt sich dies Projekt sinnvoll realisieren.

3.2. Social Networking

Definition Social Networking als Teilaspekt von Web 2.0 bezeichnet erstmal alle Tools, die es Menschen ermöglichen miteinander in Kontakt zu treten. Es existieren verschiedene Formen von Social Software wie beispielsweise Instant Messenger, Online Rollenspiele, kollaborative Editiertools, etc.

Technisch gesehen besteht ein solches Netzwerk typischer Weise aus Knoten und

Verbindungen zwischen den Knoten. Dabei sind die Knoten die Teilnehmer bzw. die Menschen und die Verbindungen die Beziehung bzw. Freundschaft zwischen diesen Menschen.

Projektbezug Der Bezug zu dieser Vision ist einfach herzustellen, denn dieser Messenger soll benutzt werden, um mit anderen Menschen in Kontakt zu treten. Das Besondere hieran ist, dass dies nicht mehr nur „zu Hause vor dem PC“ stattfinden kann, sondern wo immer man sich gerade befindet. Dies ermöglicht auch einen Schritt weiter zu gehen als man es bisher konnte. Mit dieser Lösung können sich Leute oder Freunde real treffen, die sich bisher nur virtuell treffen konnten, weil man nun mobil ist. Zum Beispiel lässt sich nun feststellen, dass sich ein Freund zufällig ganz in der Nähe aufhält, den man sowieso gerne mal wiedertreffen wollte.

3.3. Community Driven Development

Definition Beim Community Driven Development geht es generell darum, dass es eine Gemeinschaft gibt, die freiwillig, bzw. aus gegenseitigem Nutzen oder um sich zu profilieren etwas erstellt. Dabei gibt es verschiedene Unterscheidungsmöglichkeiten. Eine ist nach dem Expertengrad zu unterscheiden. Man muss zum Beispiel, um bei Linux mit zu entwickeln, ein Experte auf dem Gebiet sein. Andererseits kann zum Beispiel bei Second Life jeder Laie mitmachen und Inhalt erstellen. Zusätzlich kann man Communities noch nach offenen oder geschlossenen Communities unterteilen. Das bedeutet, dass im einen Fall jeder mitmachen kann und im anderen Fall nur ausgewählte Personen mitmachen dürfen.

Projektbezug In diesem Projekt geht es darum wie bei Wikipedia, dass jeder Inhalt erschaffen kann, oder wie bei Flickr Bilder bewerten kann. Das besondere ist, dass dies nun zu jeder Zeit und vor allem von überall aus geht. Hierzu passt das Beispiel mit dem Cafe aus dem Kapitel 1.2 Zielsetzung.

3.4. Fazit

Betrachtet man diese drei Aspekte und fasst sie in einem Projekt zusammen sind eine Menge Kriterien für pervasive Anwendungen erfüllt. Eine detaillierte Vorstellung zu diesem Thema vermittelt unter anderem die Ausarbeitung aus AW1 [Herglotz \(2007\)](#). Dadurch entsteht das Potential die Kommunikation der Menschen nachhaltig zu verändern.

4. Vorgehen

Im Projekt des Masterstudiengangs wurde sich bereits mit der Entwicklung eines Frameworks und zweier Anwendungen für Pervasive Gaming auseinandergesetzt. Eine der beiden Anwendungen ist der PS Community Messenger [Herglotz \(2008b\)](#), der eine ähnliche Funktionalität wie der PCM aufweist. Ein großer Unterschied hierbei ist die Plattform. Während im Projekt auf J2ME gesetzt wurde, wird der neue PCM auf Android aufsetzen.

4.1. Analyse

Dieser Abschnitt setzt sich zuerst mit den Anforderungen an den PCM auseinander und im Weiteren mit einer kurzen Analyse von Android.

4.1.1. Anforderungen

Bei der Erstellung der Anforderungen wird zwischen fachlichen und technischen Anforderungen unterschieden.

Fachliche Anforderungen

- Benutzerverwaltung (z.B. Registrierung, Anmeldung, Freunde suchen/einladen usw.)
- Objektverwaltung auf dem mobilen Endgerät (z.B. Orte/Ziele eingeben, bearbeiten, löschen)
- Kommunikation mit anderen Spielteilnehmern (z.B. Chat, Nachrichten schicken, Status und Aufenthaltsort anzeigen).
- Erkennen von bestimmten Objekten in Abhängigkeit vom Ort.
- Der Messenger soll darauf ausgelegt sein die ganze Zeit im Hintergrund zu laufen.
- Der aktuelle Standort soll auf einer Karte visualisiert werden.

Technische Anforderungen Aus den fachlichen Anforderungen sind die Grundfunktionalitäten erkennbar. Daraus lassen sich jetzt technische Anforderungen entwickeln.

- Zentrale Objektverwaltung/Spieleverwaltung auf einem Server über Serverkommunikation.
- Daten müssen auf dem mobilen Gerät und auf dem Server persistent gehalten werden können.
- Kommunikation (Client/Server und ggf. P2P).
- Möglichkeiten zur Lokalisierung.

4.1.2. Android Framework

Wie zuvor erwähnt, wurde sich mit der Umsetzung von Pervasive Gaming bereits im Projekt beschäftigt. Dort wurde ein Framework (siehe Ausarbeitung von [Tutzschke \(2008\)](#)) entwickelt. Dieses zu portieren lohnt sich allerdings nicht, da Android viel mehr bietet (siehe Abbildung [A.1](#)). Dem Architekturbild ist zu entnehmen, dass Android mehrschichtig aufgebaut ist. Die unterste Schicht enthält den Linux Kernel mit direktem Bezug zur Hardware. Darüber liegen die Libraries, die aus Performanzgründen in C++ entwickelt wurden. Auf dieser Ebene befindet sich ebenfalls noch die Android Runtime bestehend aus der Virtual Machine und den Core Libraries. Darüber liegen, in Java entwickelt, das Application Framework und darüber die Applikationen selbst. Da Android Open Source ist, besteht die Möglichkeit diese Komponenten einfach nur zu nutzen, gegebenenfalls auch zu erweitern oder gar neue zu entwickeln. Es gibt viele Neuheiten, die es bisher für mobile Geräte noch nicht gab. Daher sollen insbesondere aus der Schicht des Frameworks einige Bereiche kurz erklärt werden:

- **Location Manager** Anhand einer erstellten Beschreibung, welche Informationen die Location bieten muss, wird der passendste Location Service ausgewählt.
- **Activity Manager** Dieser ermöglicht, dass mehrere Anwendungen parallel laufen können und sorgt dafür, dass immer genügend Ressourcen zur Verfügung stehen. Dabei werden gegebenenfalls die am längsten nicht benutzten Anwendungen geschlossen. Greift der Benutzer wieder auf diese Anwendung zu, merkt er davon nichts, da der interne Zustand vorher gespeichert wurde.
- **Notification Manager** Tritt im Hintergrund ein entsprechendes Ereignis auf, wird der Benutzer über diesen Manager darauf aufmerksam gemacht und kann gleich zu der Anwendung wechseln.
- **SQLite** Eine kleine Datenbank ist ebenfalls in diese Geräte integriert.

4.2. Entwurf

Aufbauend auf der Analyse gibt es eine gewünschte Zielarchitektur. Des Weiteren ist ein Vorgehensmodell in diesem Fall besonders geeignet. Darüber hinaus ist auch eine konkrete Idee der Benutzerführung entstanden. Diese werden nachfolgend näher beschrieben.

4.2.1. Architektur

Ein Peer-To-Peer System ist grundsätzlich interessant. Da es sich aber in der Praxis sehr schwierig gestaltet ein reines Peer-To-Peer System umzusetzen und sich das „Always On“-Paradigma immer mehr durchsetzt, wird sich für das einfacher zu realisierende Client-Server-Modell entschieden. Da können sich zentral alle Clients anmelden, Nachrichten austauschen, Updates erhalten, etc. Trotz der Client-Server-Architektur und dem „Always On“ Paradigma ist eine gewisse Offlinefunktionalität gewünscht. Ein Beispiel hierfür ist das Finden von interessanten Locations, da nur GPS dazu benötigt wird. Dazu ist es notwendig, dass der Client abhängig vom aktuellen Aufenthaltsort innerhalb einer gewissen Umgebung die Locations lokal zwischenspeichert. Dazu benötigt der Client eine gewisse Intelligenz. Android bietet mit der SQLite Datenbank eine gute Möglichkeit dies zu lösen. Die Abbildung 4.1 veranschaulicht dies.

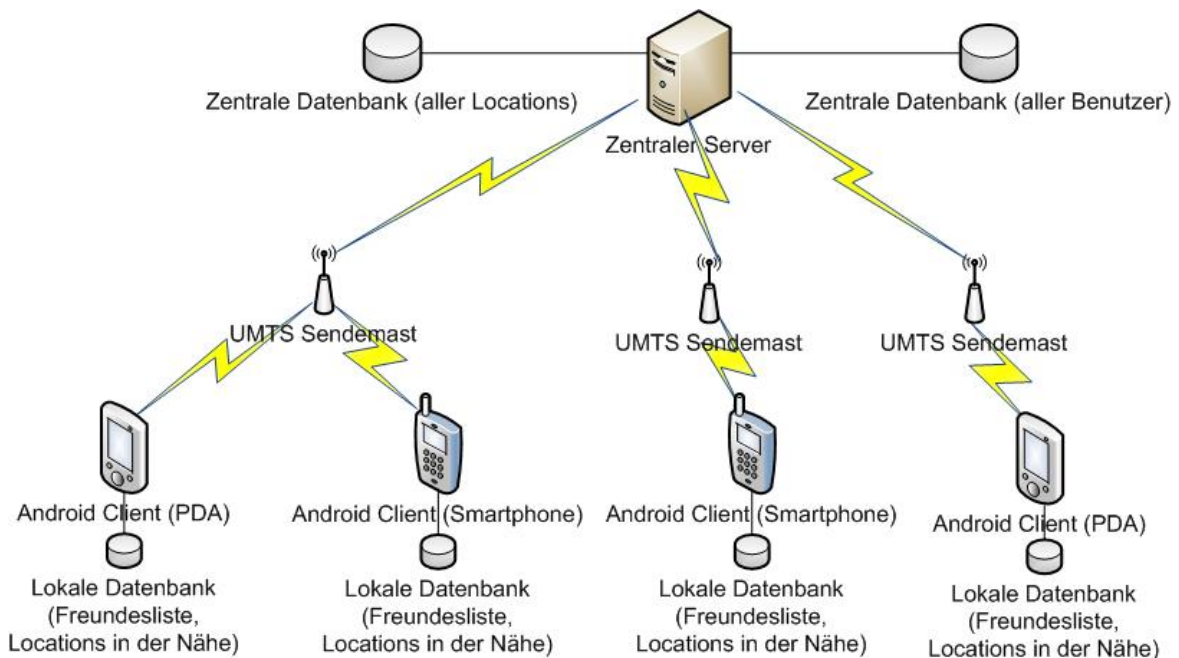


Abbildung 4.1.: Client-Server-Architektur

Client Auf den entsprechenden Endgeräten wird Android als Betriebssystem und die entsprechende Java Laufzeitumgebung integriert sein. Daher wird sich die Clientanwendung stark an dem in Abbildung [A.1](#) vorgegebenen Schichtenmodell orientieren.

Server Die Serverkomponente wird benötigt, damit jedem Client die neuesten Informationen zugänglich gemacht und Daten unter den Clients ausgetauscht werden können. Der Aufbau des Servers soll dabei erst einmal so einfach wie möglich gehalten werden, da der Schwerpunkt auf dem Client liegt.

4.2.2. Prototyping

Als Methode der Softwareentwicklung eignet sich in diesem Fall das Prototyping sehr gut. Dabei werden zwei Arten des Prototyping verwendet. Zu Beginn eignet sich das explorative Prototyping, da bei dieser Methode nur eine GUI erstellt wird, die noch keine Funktionalität besitzt. Somit hat man schnell erste Ergebnisse und dadurch ein erstes Feedback, ob das angestrebte Ziel umsetzbar ist. Besonders im Hinblick auf den Developer Challenge von Google ist dies interessant. Wurde diese Phase erfolgreich absolviert soll zum evolutionären Prototyping gewechselt werden. Zu diesem Zeitpunkt werden die Funktionalitäten Schritt für Schritt entwickelt und getestet. Mittels dieser Methode lässt sich das für die Masterarbeit notwendige „Proof of Concept“ gut realisieren.

4.2.3. Benutzerführung

Die Benutzerführung soll wie folgt realisiert werden: Nach dem Starten der Anwendung befindet man sich im Login. Von hier kann man sich entweder einloggen oder die Anwendung wieder beenden. Bei erfolgreichem Login gelangt man zur Hauptansicht, dem zentralen Punkt der Anwendung. Von dort aus kann man überall hin navigieren - am einfachsten kann man sich gleich wieder ausloggen. Des Weiteren kann man sein eigenes Profil anschauen und nach Wunsch anpassen, neue Nachrichten von Freunden lesen, Informationen über getaggte Orte in der Nähe anschauen und verändern, sich die Profile der Freunde anschauen und diesen eine Nachricht über den Server zukommen lassen, Freundschaftseinladungen annehmen oder neue Freunde suchen, die Suchergebnisse angezeigt bekommen und gegebenenfalls ein Suchergebnis hinzufügen. Dies wird durch die Abbildung [4.2](#) veranschaulicht.

Eine ähnliche Benutzerführung gab es beim PS Community Messengers aus dem Projekt auch. Ein deutlicher Unterscheidungspunkt wird der sein, dass die relativ starre sequentielle Benutzerführung mindestens durch den Notification Manager an Dynamik gewinnt. Treten neue News bzw. Events auf, über die der Benutzer informiert werden soll, werden diese in der

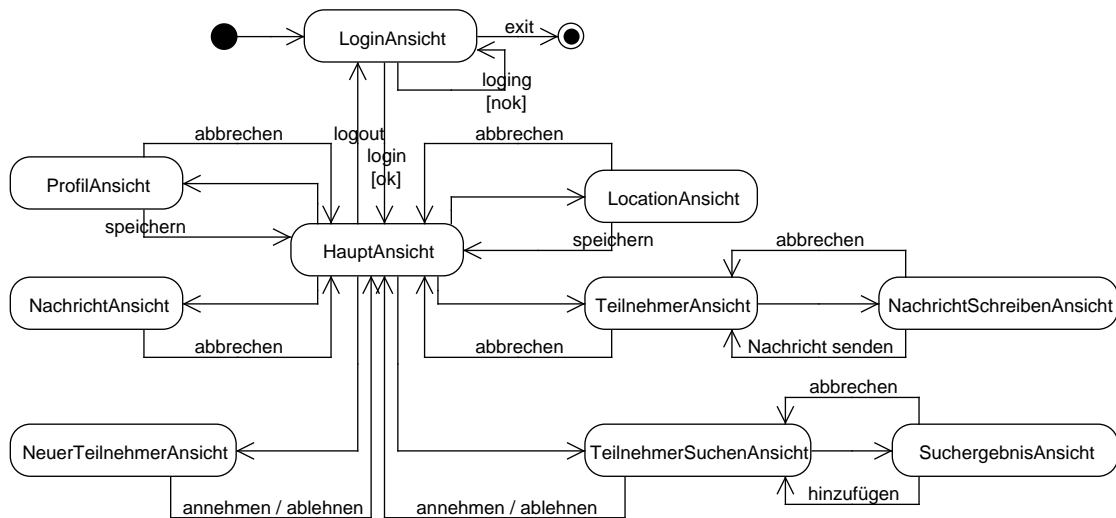


Abbildung 4.2.: Benutzerführung des Pervasive Community Messengers

Titelleiste angezeigt. Der Unterschied zum Newsticker bei J2ME ist die Weiterentwicklung von einer rein textuellen Anzeige zu einer Anzeige, bei der man direkt mit der aufrufenden Stelle verbunden ist und somit zum aktuellen Event direkt wechseln kann.

4.3. Risiken

Diese Vision bringt eine Reihe von Risiken mit sich. Zum einen gibt es Risiken bezogen auf Android. Es ist die frühe Version des SDK und damit eine möglicherweise unausgereifte Version mit vielen Fehlern anzusprechen. Zusätzlich wird es vermutlich keine Hardware bis Mitte des Jahres geben. Dadurch ist man ausschließlich auf den Emulator angewiesen. Zum anderen ist die Community ein wichtiger Risikofaktor, mit dem der Erfolg und die Sinnhaftigkeit des Projektes einhergeht. In dem Zusammenhang ist auch der mitunter „virtuelle Striptease“ ein Schlüsselfaktor. Bisher ist dies kaum ein Problem gewesen, wenn man „Friend Finder“ wie StudiVZ betrachtet. Sollte jedoch das Bewusstsein bezüglich der Privatsphäre geschärft werden, könnte dies zum Ausführen führen. Diese Frage ist wichtiger denn je, da z.B. manipulative Werbung auf Grund der Positionsdaten noch besser eingesetzt werden kann. Über den Missbrauch dieser Daten muss nachgedacht werden.

5. Zusammenfassung

Abschließend wird ein kurzes Fazit zur Vision und dem Vorgehen gegeben und ein Ausblick auf die Masterarbeit.

5.1. Fazit

Auf Grund der groben Machbarkeitsstudie scheint diese Vision trotz der Risiken umsetzbar zu sein. Ob es im Detail noch große vielleicht unüberwindbare Probleme geben wird, kann allerdings nicht ausgeschlossen werden. Dennoch bietet diese Vision eine Chance in dem Forschungsgebiet des Pervasive Gaming aktiv mitzuwirken. Da die Android Plattform noch jünger als vier Monate ist, kann zumindest bezogen auf die Plattform etwas Neues erschaffen werden, das es so noch nicht gibt. Auch das Betrachten und Integrieren von Community Driven Development und Social Networking in den Pervasive Gaming Kontext und die neuen Möglichkeiten, die damit geschaffen werden, bleiben ein spannendes Feld mit viel Potential.

5.2. Ausblick

Ziel ist es diese Vision in der Masterarbeit umzusetzen. Dabei soll nach Möglichkeit als Teilziel an dem von Google ausgeschriebenen Developer Challenge teilgenommen werden. Das Datum bis zu dem der Vorschlag in Form einer Anwendung eingereicht werden muss liegt allerdings ziemlich zeitnah am 14. April 2008.

Literaturverzeichnis

- [FIT - IPerG 2007] FIT, Fraunhofer: *IPerG*. 2007. – URL <http://www.fit.fraunhofer.de/projects/mixed-reality/iperg.html>. – Abrufdatum 28.02.2008
- [Herglotz 2007] HERGLOTZ, Andreas: *Pervasive Gaming im Kontext des Frankfurter Flughafens*, HAW Hamburg, Seminararbeit, 2007
- [Herglotz 2008a] HERGLOTZ, Andreas: *Community Driven Development*, HAW Hamburg, Seminararbeit, 2008
- [Herglotz 2008b] HERGLOTZ, Andreas: *Pervasive Spine - Ein Framework für mobile Anwendungen*, HAW Hamburg, Projektbericht, 2008
- [Jegers und Wiberg 2006] JEGERS, K. ; WIBERG, M.: Pervasive Gaming in the Everyday World. In: *Pervasive Computing, IEEE* 5 (2006), Nr. 1, S. 78–85. – URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1593575
- [Moore Jan 1998] MOORE, G.E.: Cramming More Components Onto Integrated Circuits. In: *Proceedings of the IEEE* 86 (Jan 1998), Nr. 1, S. 82–85. – ISSN 0018-9219
- [O'Reilly 2005] O'REILLY, Tim: *What Is Web 2.0? Design Patterns and Business Models for the Next Generation of Software*. www.oreilly.com. September 2005
- [Schmidt u. a. 1999] SCHMIDT, Albrecht ; BEIGL, Michael ; GELLERSEN, Hans-W: There is more to context than location. In: *Computers & Graphics* 23 (1999), December, Nr. 6, S. 893–901. – URL [http://dx.doi.org/10.1016/S0097-8493\(99\)00120-X](http://dx.doi.org/10.1016/S0097-8493(99)00120-X)
- [Stearns u. a. Jan. 2006] STEARNS, H. ; GARGUS, J. ; SCHUETZE, M. ; LOMBARDI, J.: Simplified Distributed Authoring Via Component-based Object Construction and Deconstruction in Collaborative Croquet Spaces. In: *Creating, Connecting and Collaborating through Computing, 2006. C5 '06. The Fourth International Conference on* (Jan. 2006), S. 79–87
- [Tepper 2003] TEPPER, Michele: The rise of social software. In: *netWorker* 7 (2003), Nr. 3, S. 18–23. – ISSN 1091-3556

- [Tomiyasu u. a. 05–08 April 2005] TOMIYASU, H. ; MAEKAWA, T. ; HARA, T. ; NISHIO, S.: Social Network Applications using Cellular Phones with E-mail Function. In: *Data Engineering Workshops, 2005. 21st International Conference on* (05-08 April 2005), S. 1253–1253
- [Tutzschke 2008] TUTZSCHKE, Jan P.: *Pervasive Spine - Ein Framework für mobile Anwendungen*, HAW Hamburg, Projektbericht, 2008
- [Weiser 1995] WEISER, Mark: The computer for the 21st century. (1995), S. 933–940. ISBN 1-55860-246-1
- [Weiser 1996] WEISER, Mark: *Ubiquitous Computing*. 1996. – URL <http://www.ubiq.com/hypertext/weiser/UbiHome.html>. – Abrufdatum 28.02.2008

A. Android

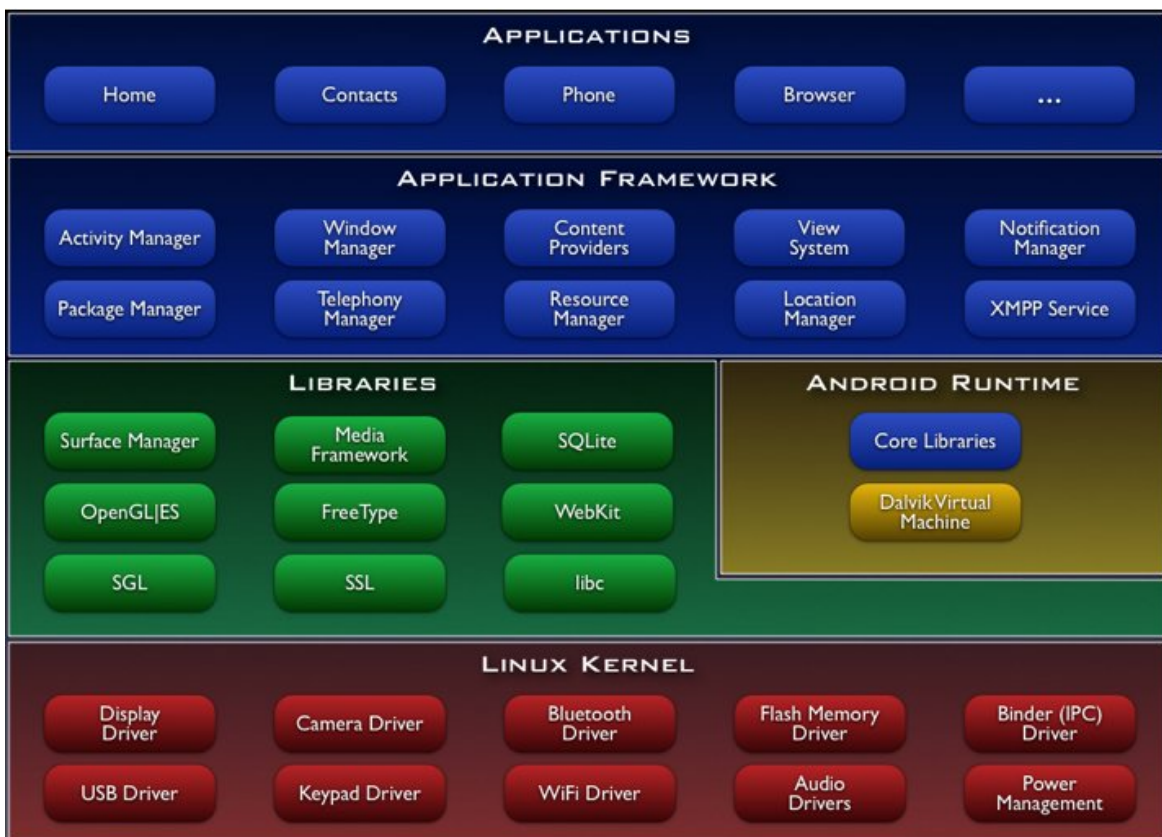


Abbildung A.1.: Android Architektur