



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Projektausarbeitung

Andreas Herglotz

Pervasive Spine

Ein Framework für mobile Anwendungen

Andreas Herglotz
Pervasive Spine
Ein Framework für mobile Anwendungen

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

Betreuende Professoren: Prof. Dr. Olaf Zukunft, Prof. Dr. Gunter Klemke

Abgegeben am 29. Februar 2008

Andreas Herglotz

Thema der Projektausarbeitung

Pervasive Spine

Stichworte

Pervasive Gaming, J2ME, Framework

Kurzzusammenfassung

Diese Ausarbeitung handelt von einem Projekt über Pervasive Gaming, welches acht Masterstudenten im Rahmen eines Semesters durchgeführt haben. Dabei wurde auf Basis von J2ME ein Framework inklusive zweier Beispielanwendungen als Proof of Concept entwickelt, die PS Orientierungseinheit und der PS Community Messenger. Diese Ausarbeitungen beschäftigen sich im speziellen mit dem PS Community Messenger.

Andreas Herglotz

Title of the paper

Pervasive Spine

Keywords

Pervasive Gaming, J2ME, Framework

Abstract

This paper deals with pervasive gaming. A group of eight master students developed a pervasive gaming framework and two applications as a proof of concept during one term. These applications are PS Orientierungseinheit and PS Community Messenger. The focus of this paper is the PS Community Messenger.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motivation	1
1.2. Zielsetzung	1
2. Projekt	2
2.1. Ablauf	2
2.2. Aufgaben	2
2.3. Bluetooth	3
3. PS Community Messenger	4
3.1. Analyse	4
3.1.1. Anwendung	4
3.1.2. Anforderungen	5
3.1.3. Fazit	6
3.2. Design	6
3.2.1. Benutzerführung	6
3.2.2. Pattern	7
3.3. Realisierung	7
3.3.1. Prototyping	7
3.3.2. Struktur der Anwendung	8
3.3.3. Location-based Services	10
3.4. Test	10
4. Zusammenfassung	11
4.1. Fazit	11
4.2. Ausblick	11
Literaturverzeichnis	13
A. Framework	14
B. PS Orientierungseinheit	15
C. PS Community Messenger	16

1. Einleitung

Das Masterprojekt erstreckte sich über 16 Wochen zu je acht Stunden. Es haben sich acht Studenten zum Thema „Pervasive Gaming“ zu einem Team zusammengeschlossen. Der Inhalt dieses Projektes wird anhand der Ausarbeitungen der Projektmitglieder dargestellt.

1.1. Motivation

Spiele sind ein wichtiger Teil der Gesellschaft, sie machen Spaß, ermöglichen neue Fertigkeiten zu erlernen, treiben Kommunikation und interkulturellen Austausch voran. Mit der heutigen mobilen Technik lässt sich ein neuer Bereich des Spielens erschließen - das Pervasive Gaming. Dabei handelt es sich um Spiele, die in unser tägliches Leben integriert sein sollen. Es kommen mobile Geräte, orts- und kontextbezogene Technologien, etc. zum Einsatz, die virtuelle und physische Spielelemente verknüpfen.

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 die Möglichkeiten, die im Pervasive Gaming stecken. Bereits heute gibt es Smartphones, die mit einem entsprechenden Funktionsumfang ausgestattet sind.

1.2. Zielsetzung

Unser Ziel war es nicht einfach eine Anwendung oder ein Spiel zu programmieren, sondern einen Mehrwert in diesem Forschungsgebiet zu erschaffen. Dabei sollen nicht nur neueste Technologien zum Einsatz gebracht werden, sie sollen auch für ähnliche Szenarien leicht wieder verwendet werden können. Deswegen wurde sich dafür entschieden ein Framework für Pervasive Gaming und als „Proof of Concept“ zwei verschiedene Beispielapplikationen zu entwickeln. Damit wird eine Möglichkeit geschaffen, ähnliche Spielkonzepte schnell und einfach umsetzen zu können. Auf Grund der begrenzten Zeit gehört es nicht zu den vorrangigen Zielen, dass konkrete Produkte dabei entstehen, sondern Erfahrung in dem Bereich der Entwicklung auf mobilen Endgeräten zu sammeln.

2. Projekt

In diesem Kapitel werden kurz der allgemeine Projektablauf und die Einteilung in die verschiedenen Gruppen vorgenommen. Im Weiteren wird auf Grund meiner direkten Beteiligung konkret auf den Bereich Bluetooth eingegangen. Das folgende Kapitel [3](#) stellt die Entwicklung der Anwendung PS Community Messenger (PSCM) dar.

2.1. Ablauf

Das gesamte Projekt lässt sich in verschiedene Phasen einteilen. Zu Beginn gab es eine Phase der Orientierung, in der zum Beispiel die Frage der Plattform geklärt wurde. Dabei haben wir uns für J2ME entschieden. Des Weiteren wurden grundlegende Dinge wie ein CVS, etc. vorbereitet. In der Anforderungsphase wurden die Ziele genauer definiert. Um möglichst effizient zu arbeiten, fand zu diesem Zeitpunkt eine Einteilung in verschiedene Gruppen (Framework, PS Orientierungseinheit und PS Community Messenger) statt. Danach folgte eine Phase in der entwickelt wurde, gefolgt von einer Testphase. Daraufhin wurden die Entwicklungen in Betrieb genommen. Dabei festgestellte Probleme wurden in der letzten Phase der Überarbeitung behoben. Nähere Informationen zum Projektablauf sind in der Ausarbeitung von [Kruse \(2008\)](#), der die Aufgabe als Projektleiter übernommen hat, zu finden.

2.2. Aufgaben

An dieser Stelle gibt es einen kurzen Einblick in die Aufgaben der anderen Teams.

Framework Es gab verschiedene Ziele des Frameworks. Ein Hauptbereich war es, durch die Abstraktion von den Hardwarekomponenten, entsprechende einfach zu nutzende Services anzubieten. Die Einbindung von Komponenten war ein weiteres Ziel, um häufig benötigte Funktionalitäten nur einmal entwickeln zu müssen (siehe [Abbildung A.1](#)). Die Strukturierung durch ein MVC-Konzept sowie das Einbinden eines Eventhandlings wurden ebenso

umgesetzt (siehe Abbildung A.2). Der letzte wichtige Bereich war die Schaffung einer einfachen Konfigurationsmöglichkeit der Anwendungen, welche durch eine XML-Struktur ermöglicht wurde. Weitere Informationen sind in den Ausarbeitungen von [Tutzschke \(2008\)](#), [Mas \(2008\)](#) und [Dreyer \(2008\)](#) zu finden.

PS Orientierungseinheit Es handelt sich um eine mit moderner Technik unterstützte Schnitzeljagd. Jeder Benutzer kann eine erstellen oder einer beitreten. Am Ende wird der Gewinner ermittelt. Eine Grafik über den Ablauf ist im Anhang B.1 zu finden. Die detaillierte Beschreibung ist in der Ausarbeitung von [Hartmann und Schönherr \(2008\)](#) nachzulesen.

2.3. Bluetooth

Eine der Anforderungen an den PS Community Messenger war die Adhoc Kommunikation mit Bluetooth. Dabei beschäftigt sich die Arbeit von [Vollmer \(2008\)](#) mit der unteren Ebene, der Protokollebene. Die Arbeit von [Dreyer \(2008\)](#) baut auf diese Schicht auf und stellt die Verbindung zum Framework mit dem Ziel her, dass die Anwendung den Service leicht benutzen kann. Bei der Bluetoothkommunikation traten nicht vorhersehbare und nicht lösbare Probleme auf. Als bisher Unbeteiligter versuchte ich die beiden zu unterstützen, um das Problem in den Griff zu bekommen. In einem ersten Ansatz hat Sven Vollmer versucht die Kommunikation auf objektorientierte Art und Weise zu lösen. Dies scheiterte. In einem zweiten Ansatz versuchten wir dies über einen Zustandsautomaten zu lösen und in einem dritten Ansatz versuchten wir anstelle des OBEX Protokolls das L2CAP Protokoll zu nutzen - mit gleichem Ergebnis.

Zum Schluss des Projektes fanden wir heraus, warum diese nicht deterministischen Fehler auftraten: Ziel war es in regelmäßigen Abständen nach neuen Geräten in Reichweite zu suchen und unbekannte Geräte miteinander bekannt zu machen, zu überprüfen, ob beide den gleichen Service benutzen und dann miteinander in Kommunikation zu treten. Das Problem ist einfach zu erklären - die Technologie ist dafür nicht vorgesehen. Um Geräte zu suchen gibt es den Inquiry-Modus. Dies ist allerdings ein Sondermodus, der zur Konsequenz hat, dass Geräte, die sich in diesem Modus befinden, selber nicht gefunden werden können. Da in unserem Szenario regelmäßig gesucht werden soll, schließen sich diese Geräte gegenseitig aus. Unser Szenario ist mit der derzeitigen Bluetoothtechnologie nicht umsetzbar. Eine andere Lösung mit leichten Einschränkungen ist denkbar. Beispielsweise könnte man unter dem „Always On“-Paradigma die Nähe zweier Teilnehmer auch über GPS feststellen und die Kommunikation über den Server laufen lassen. Man ist erst einmal auf Outdoorszenarien beschränkt, aber unter der Voraussetzung, dass Annahmen getroffen werden, ist dies wieder erweiterbar. Bei Verlust des GPS-Signals könnte man annehmen, dass man sich in einem Gebäude befindet. Bei solchen Annahmen ist Vorsicht bezüglich der Richtigkeit geboten.

3. PS Community Messenger

Dieses Kapitel beschäftigt sich mit den wichtigen Aspekten bei der Entwicklung des PSCM.

3.1. Analyse

Zu einer Projektdurchführung gehört eigentlich in der Analyse die Erstellung eines klassischen Lasten- und Pflichtenheftes. Auf diese wurde bewusst verzichtet, da die Erstellung unseren Annahmen entsprechend zu viel Zeit in Anspruch genommen hätte. Es war wichtiger sich erst einmal mit dem für alle Beteiligten neuen Thema überhaupt auseinanderzusetzen und schnell erste Schritte zu machen. Im Kapitel Realisierung [3.3](#) wird das Vorgehen genauer beschrieben. An dieser Stelle wird erst die Anwendung beschrieben und dann die Anforderungen definiert.

3.1.1. Anwendung

Es wurde ein Community Messenger erstellt, der sich im Wesentlichen durch seine pervasiven Eigenschaften von herkömmlichen Messengern unterscheidet. So besteht nicht nur die Möglichkeit sich virtuell zu treffen, sondern auch durch Ortsinformationen - über die eigene Position und die der Freunde - reale Treffen zu ermöglichen.

Dabei soll es zwei Möglichkeiten geben: Zum einen soll man benachrichtigt werden, wenn sich Freunde, die bereits in der Freundschaftsliste abgebildet sind, zufällig in der Nähe befinden. Zum anderen sollen sich unbekannte Menschen, die sich innerhalb einer gewissen Reichweite befinden und gleiche Interessen haben, spontan treffen können.

Eine weitere Besonderheit ist, dass man Orte mit virtuellen Zusatzinformationen versehen kann. Auf diese Informationen wird, sobald man sich in deren Nähe befindet, hingewiesen. Diese Informationen sollen von jedem Benutzer (der Community) erstellt, gelesen und erweitert werden können.

Der grobe Ablauf des PS Community Messengers ist in der Abbildung C.1 im Anhang anhand von Screenshots skizziert.

3.1.2. Anforderungen

Für die Analyse wurde sich auf die Erstellung von Anforderungen beschränkt, die auf Basis dieses Szenarios aber auch der PS Orientierungseinheit entstanden sind. In das Framework sollten mindestens die Anforderungen integriert werden, die sich bei beiden Szenarien überschneiden und zwingend notwendig sind, da diese somit als grundlegende Funktionalitäten identifiziert wurden. Im Folgenden wird unterschieden zwischen fachlichen und technischen Anforderungen.

Fachliche Anforderungen

- Benutzer- und Spielverwaltung (z.B. Anmeldung, Spiel beitreten, Highscore, usw.)
- Objektverwaltung auf dem mobilen Endgerät (z.B. Orte/Ziele eingeben, bearbeiten, löschen)
- Kommunikation mit anderen Spielteilnehmern (z.B. Chat, Nachrichten schicken)
- Erkennen von bestimmten Objekten in Abhängigkeit vom Ort.
- Das Spiel soll unterbrechbar sein und irgendwann fortgesetzt werden können.
- Das Spiel soll zu jeder Zeit ausführbar sein.
- Validierung, ob bzw. wenn ein gesuchter Ort erreicht wurde.
- 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/Spielverwaltung auf einem Server über Serverkommunikation.
- Daten müssen auf dem mobilen Gerät und auf dem Server persistent gehalten werden können.
- Kommunikation (Client/Server und P2P).
- Online- und Offlinefunktionalität.

- Möglichkeiten zur Lokalisierung.

3.1.3. Fazit

Da der PSCM unter dem Hauptthema Pervasive Gaming entwickelt wurde, gibt es auch besondere Eigenschaften, die diesen auszeichnen. Das erste grundlegende Charakteristikum ist, dass der PSCM auf einem natürlichen mobilen Gerät (Handy, Smartphone oder PDA) benutzt werden kann. Aufbauend auf der gewonnenen Mobilität wurde sich diese Eigenschaft zu Nutze gemacht, indem Ortsdienste bzw. Ortsinformationen in den PSCM integriert wurden. Ein weiteres Merkmal ist, dass der PSCM jederzeit und überall nutzbar sein soll. Dafür wurde eine online sowie offline Funktionalität benötigt. Die Onlinefunktionalität beinhaltet eine Serverkomponente. Auf der Serverkomponente befindet sich unter anderem die zuletzt bekannte eigene Position, die erstellten Informationen zu Lokationen, etc. Nähere Informationen zu der Serverkomponente sind in der Arbeit von [Kruse \(2008\)](#) zu finden. Unter Offlinefunktionalität ist die Clientanwendung ohne Serverkomponente zu verstehen.

3.2. Design

3.2.1. Benutzerführung

Das folgende Zustandsdiagramm (Abbildung 3.1) veranschaulicht die Benutzerführung.

Dabei entsprechen die grauen Zustände dem, was in der endgültigen Version, abweichend von den ursprünglichen Anforderungen, aus zeitlichen Gründen nicht mehr umgesetzt werden konnte. Dies sind „Freundschaftseinladungen annehmen“, „neue Freunde suchen“, „Suchergebnisse anzeigen“ und gegebenenfalls „ein Suchergebnis hinzufügen“.

Das Schwarze wurde wie folgt realisiert: 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 oder sich die Profile der Freunde anschauen und diesen eine Nachricht über den Server zukommen lassen.

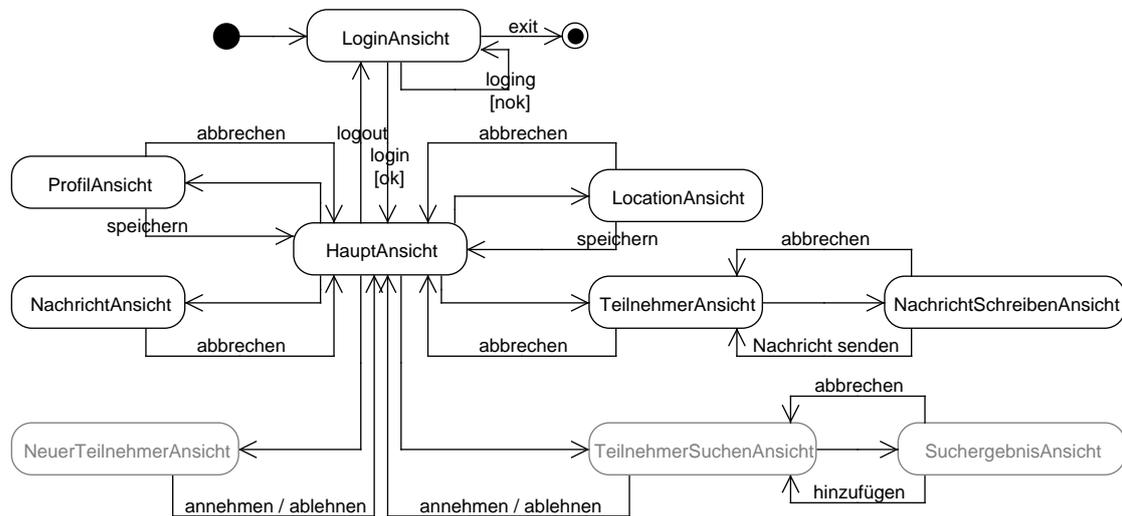


Abbildung 3.1.: Zustandsdiagramm: Benutzerführung

3.2.2. Pattern

Bei der Umsetzung der Anforderungen des PSCM ist als Pattern das Singleton für den Tickerholder umgesetzt worden. Hintergrund ist der, dass es zwar immer neue News geben kann, die aber in einer Message Queue gespeichert werden. Das heißt, es gibt nur genau einen Tickerholder, der in diesem Fall maximal drei News anzeigen kann. Weitere Pattern kamen nicht explizit zum Einsatz. Dabei ist zu erwähnen, dass der Fokus auch nicht auf die Anwendung besonderer Pattern gelegt wurde.

3.3. Realisierung

3.3.1. Prototyping

Als Methode der Softwareentwicklung wurde das Prototyping (siehe [Lichter u. a. \(1993\)](#)) angewendet, da im Bereich der mobilen Entwicklung noch keine Erfahrungen vorhanden waren und es auf Grund der begrenzten Zeit wichtig erschien am Anfang schnell Ergebnisse

zu erzielen, um gegebenenfalls frühzeitig auf Probleme reagieren zu können. Nachfolgend werden die verschiedenen Phasen kurz beschrieben.

Horizontales Prototyping In dieser ersten Phase wurde die Funktionalität erst einmal außen vor gelassen. Die Arbeit konzentrierte sich darauf eine GUI zu schaffen, die die Benutzerführung widerspiegelt. Die Benutzerführung (Kapitel 3.2.1) enthält implizit die Anforderungen bzw. die Funktionalitäten, die umgesetzt werden sollten.

Vertikales Prototyping Die weiteren Phasen entsprechen dem vertikalen Prototyping. Die einzelnen Funktionalitäten wurden implementiert.

1. **Lokale Anwendung** In diesem ersten Schritt wurden die Funktionalitäten ohne Serverkomponente implementiert, um die Komplexität langsam zu erhöhen. Mit Evaluationsdaten war der PSCM soweit funktionsfähig. Zu diesem Zeitpunkt wurden alle geänderten Daten persistiert. Um mit anderen in Kontakt treten zu können und den pervasiven Gedanken sinnvoll zu integrieren wurde in einem nächsten Schritt die Serverkomponente entwickelt.
2. **Server Anwendung** Durch die Serverkomponente konnten die Testdaten durch reale Daten ersetzt werden und die Anwendung konnte wie vorgesehen genutzt werden. Genauere Informationen zu diesem Abschnitt sind in [Kruse \(2008\)](#) zu finden. Aus zeitlichen Gründen fehlt seit der Integration des Servers die Persistierung.

3.3.2. Struktur der Anwendung

Durch die Nutzung des Frameworks wurde eine gewisse Struktur bereits vorgegeben. Dies veranschaulicht das folgende Paketdiagramm (Abbildung: 3.2). Es gibt eine flache Hierarchie, die klar zwischen View, Action und EventAction aus dem Framework trennt. Der Bereich Services beschäftigt sich mit den Webservices. Für genauere Informationen wird auf die Ausarbeitung von [Kruse \(2008\)](#) verwiesen. Der Inhalt des DTO wird durch das Klassendiagramm (Abbildung: 3.3) verdeutlicht.

Ausgehend von der Klasse `AbstractCommElem`, die aus den Eigenschaften besteht, die alle anderen Elemente auch besitzen gibt es als weitere abstrakte Klasse die Klasse `AbstractCommElemPerson`, die ihre Ausprägungen in den Klassen `CommElemBuddy`, `CommElemInvitation` und `CommElemOwnProfile` findet. Weiterhin sind konkrete Implementationen von `AbstractCommElem` die Klassen `CommElemMessage` und `CommElemLocation`.

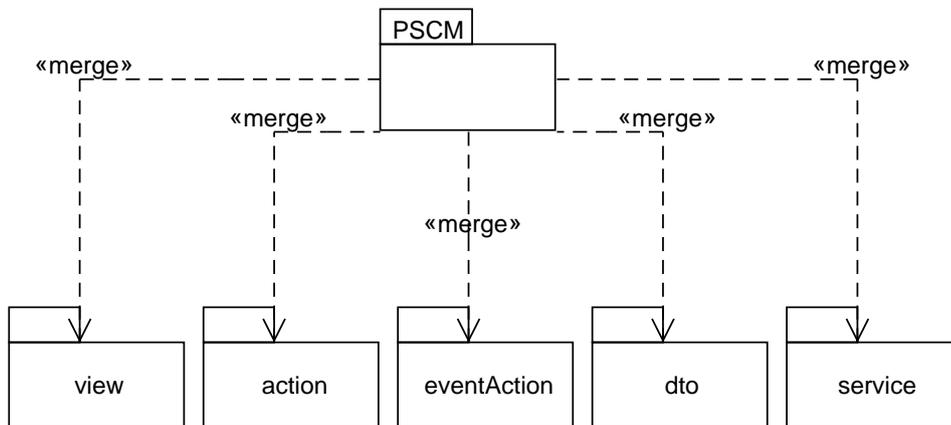


Abbildung 3.2.: Paketdiagramm des PS Community Messengers

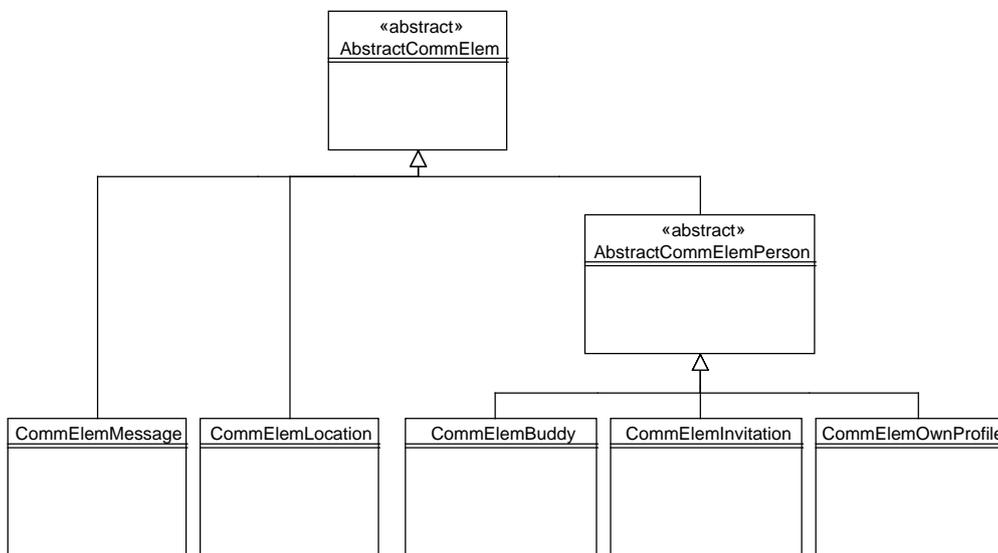


Abbildung 3.3.: Klassendiagramm der Elemente des PS Community Messengers

3.3.3. Location-based Services

Es waren zwei Arten von Ortsinformationen vorgesehen. Zum einen durch absolute Positionsbestimmung mit GPS und zum anderen durch die relative Positionsbestimmung zu anderen Personen per Bluetooth.

GPS Die Positionsbestimmung per GPS ermöglicht es festzustellen, ob Freunde oder besondere Orte, die mit Zusatzinformationen versehen sind, sich aktuell in der Nähe befinden.

Bluetooth Mit dieser Technologie sollte festgestellt werden, wenn sich jemand, der sich noch nicht in der Freundesliste befindet, aber den PSCM benutzt, in der Nähe aufhält. So sollten spontane Interaktionen ermöglicht werden. Wie dem Kapitel 2.3 zu entnehmen ist scheiterte diese Realisierung.

3.4. Test

Tests wurden immer in zwei Schritten durchgeführt, wenn eine Komponente fertiggestellt wurde. Erst wurde im Emulator und dann auf dem Handy getestet. In der im Projektplan vorgesehenen Testphase wurde zwar verstärkt getestet, allerdings lag der Fokus nicht so stark auf dem Testen, wie es in der Phase hätte sein sollen, da bis zu dieser Phase noch nicht alle Funktionalitäten umgesetzt werden konnten, die elementar waren. Ein wenig mehr Systematik beim Testen sollte bei einem erneuten Projektdurchlauf etwa in Form von Unit Tests berücksichtigt werden. Getestet wurde im Emulator mit Debugausgaben. Auf dem Handy gestaltete sich dies auf Grund der kleinen Displays schwierig. Erst zu spät wurde über die Möglichkeit nachgedacht die Testergebnisse per UMTS an den PC zu übertragen und dort auszuwerten.

4. Zusammenfassung

Abschließend werden ein kurzes Fazit über das Projekt und ein Ausblick auf die Masterarbeit gegeben.

4.1. Fazit

Beginnend mit dem Negativen kann man sagen, dass es eine Reihe von Problemen bei der Arbeit mit J2ME und den mobilen Geräten gab. Da ist zum einen die schlechte Dokumentation von J2ME zu erwähnen und zum anderen der Emulator. Wurde ein Test erfolgreich auf dem Emulator ausgeführt, bedeutete dies nicht zwangsläufig, dass dieser Test auch auf dem mobilen Gerät erfolgreich ist. Nicht gesondert behandelt wurden auf Grund der begrenzten Zeit Probleme wie beispielsweise die besondere Bedienbarkeit wegen der kleinen Displays etc. Wir haben in dem Projekt viel dazu gelernt und würden daher bei einem Neubeginn teilweise andere Designentscheidungen treffen. Ein Beispiel dafür ist die Bluetoothkommunikation, die anders gelöst werden müsste. Trotz gegebener Schwierigkeiten ist ein sehenswertes und funktionsfähiges Ergebnis entstanden. Die Projektarbeit in diesem Team hat viel Spaß gemacht, war harmonisch und produktiv, was durch ein überaus starkes Engagement aller widergespiegelt wurde.

4.2. Ausblick

Für die Masterarbeit soll der grobe Kontext der gleiche bleiben. Allerdings gibt es Gründe die Plattform zu wechseln. Zum einen gab es die beschriebenen Probleme mit J2ME, zum anderen soll J2ME nicht weiter entwickelt werden. Zusätzlich veröffentlichte Google das Android SDK und schuf damit eine neue Alternative. Android basiert auf einem Linux Kernel mit darüber liegendem Java und einem für mobile Anwendungen relativ mächtigen Framework. Besonders interessant ist Android, weil Google dies als Open Source Projekt zur Verfügung stellt. Einziger Nachteil ist, dass es vor Mitte des Jahres keine Endgeräte geben wird. Als provisorische Lösung gibt es einen Emulator, der ersten Tests zu Folge vielversprechend scheint.

Literaturverzeichnis

- [Dreyer 2008] DREYER, Markus: *Pervasive Spine - Ein Framework für mobile Anwendungen*, HAW Hamburg, Projektbericht, 2008
- [Gamma u. a. 1995] GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph ; VLISSIDES, John: *Design Patterns*. Addison-Wesley Professional, January 1995. – ISBN 0201633612
- [Hartmann und Schönherr 2008] HARTMANN, Leif ; SCHÖNHERR, Jan: *Pervasive Spine - Ein Framework für mobile Anwendungen*, HAW Hamburg, Projektbericht, 2008
- [Herglotz 2007] HERGLOTZ, Andreas: *Pervasive Gaming im Kontext des Frankfurter Flughafens*, HAW Hamburg, Seminararbeit, 2007
- [Herglotz 2008] HERGLOTZ, Andreas: *Community Driven Development*, HAW Hamburg, Seminararbeit, 2008
- [Kruse 2008] KRUSE, Ralf: *Pervasive Spine - Ein Framework für mobile Anwendungen*, HAW Hamburg, Projektbericht, 2008
- [Lichter u. a. 1993] LICHTER, Horst ; SCHNEIDER-HUFSCHEIDT, Matthias ; ZÜLLIGHOVEN, Heinz: Prototyping in industrial software projects bridging the gap between theory and practice. In: *ICSE '93: Proceedings of the 15th international conference on Software Engineering*. Los Alamitos, CA, USA : IEEE Computer Society Press, 1993, S. 221–229. – ISBN 0-89791-588-7
- [Mas 2008] MAS, Alexander: *Pervasive Spine - Ein Framework für mobile Anwendungen*, HAW Hamburg, Projektbericht, 2008
- [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
- [Tutzschke 2008] TUTZSCHKE, Jan P.: *Pervasive Spine - Ein Framework für mobile Anwendungen*, HAW Hamburg, Projektbericht, 2008
- [Vollmer 2008] VOLLMER, Sven: *Pervasive Spine - Ein Framework für mobile Anwendungen*, HAW Hamburg, Projektbericht, 2008

A. Framework

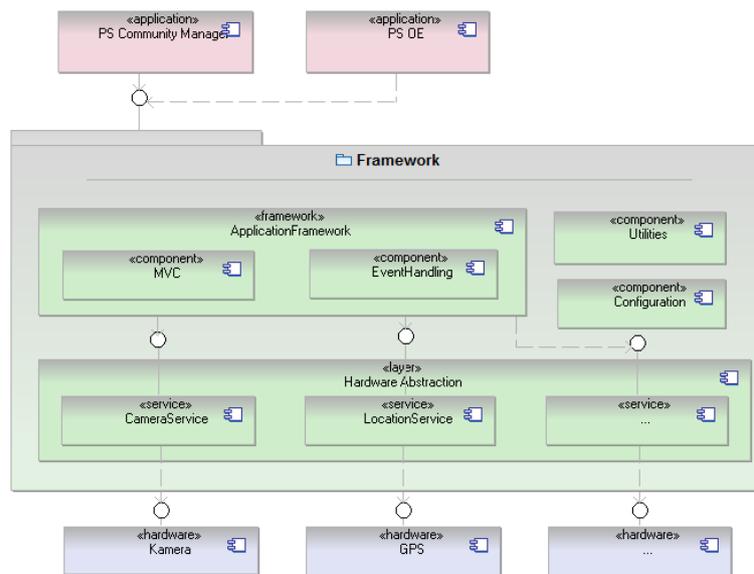


Abbildung A.1.: Architektur des Frameworks

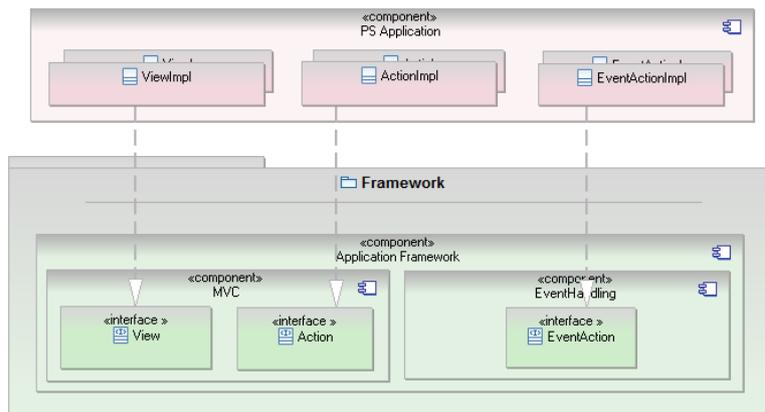


Abbildung A.2.: MVC des Frameworks

B. PS Orientierungseinheit

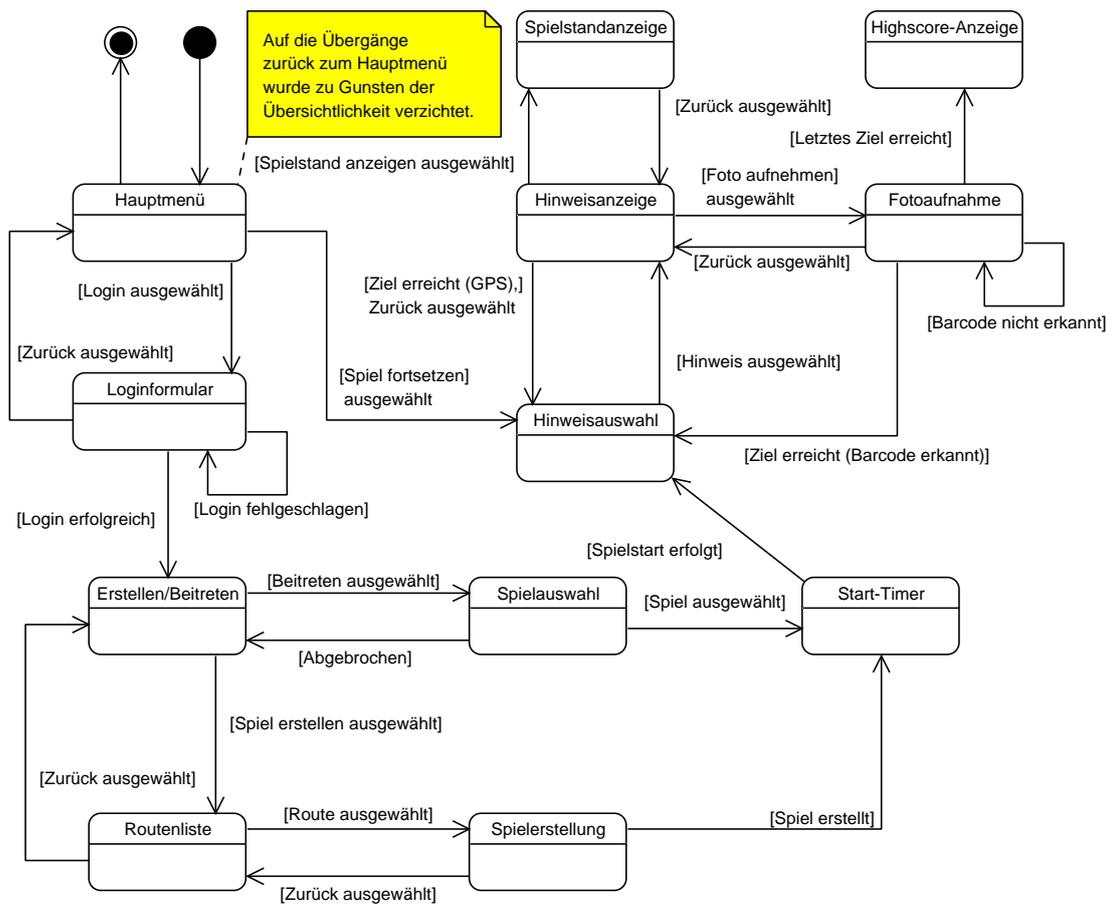


Abbildung B.1.: Zustandsdiagramm: OE-Spiel

C. PS Community Messenger

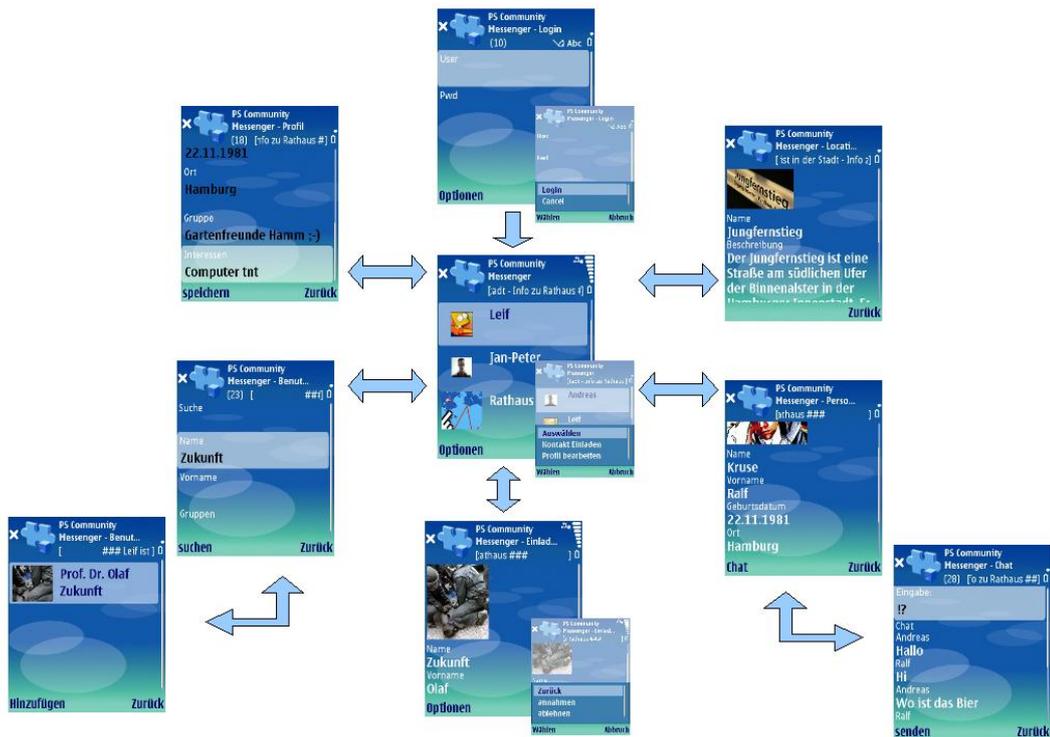


Abbildung C.1.: Ablaufskizzierung des PS Community Messengers