



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Projektbericht

Jan-Peter Tutzschke

Pervasive Spine

Jan-Peter Tutzschke

Thema des Projektberichts

Pervasive Spine

Stichworte

Pervasive Computing, Pervasive Gaming, Mobile Computing

Kurzzusammenfassung

In diesem Projektbericht wird das Konzept und Teile der Realisierung des *Pervasive Spine* Frameworks für mobile ortsabhängige Spiele vorgestellt.

Inhaltsverzeichnis

1	Einleitung und Motivation	1
2	Grundlagen	2
3	Projekt	3
3.1	Vision	3
3.2	Ablauf	3
3.3	Szenarien	4
3.3.1	PS Community Manager	4
3.3.2	PS Orientierungseinheit	5
3.4	Anforderungen	5
3.4.1	Fachliche Anforderungen	5
3.4.2	Technische Anforderungen	6
4	Konzept	8
4.1	Architektur des Frameworks	8
4.2	Realisierung	9
4.2.1	Infrastruktur	10
4.2.2	Framework	10
4.2.3	LocationService	11
4.2.4	Configuration	12
4.2.5	Utilities	12
5	Zusammenfassung und Ausblick	13
	Abbildungsverzeichnis	14
	Literatur	15

1 Einleitung und Motivation

Die steigende Leistungsfähigkeit moderner mobiler Endgeräte ermöglicht die Entwicklung von immer neuen Spielideen und -konzepten, die nach dem Paradigma des *Pervasive Gaming* die reale Welt mit der virtuellen verbinden (Schneider und Kortuem, 2001). Eine Implementierung dieser neuen Spielideen und -konzepte ist durch die technischen Möglichkeiten unter Umständen zeitaufwendig und komplex. In der Projektveranstaltung im Rahmen des Masterstudiengangs Verteilte Systeme an der HAW Hamburg wurde aus diesem Grund das Framework *Pervasive Spine* entwickelt und durch die Applikationen *PS Community Manager* und *PS Orientierungseinheit* validiert. *Pervasive Spine* stellt für die Applikationen ein Grundgerüst zu Verfügung, welches die Entwicklung beschleunigen soll.

In diesem Projektbericht wird das Konzept und Teile der Realisierung des *Pervasive Spine* Frameworks vorgestellt. Da das Projekt mit einem Projektteam von acht Projektmitgliedern durchgeführt wurde, wird hauptsächlich der Aufgabenbereich des Autors vorgestellt. Für einen tieferen Einblick in die Aufgabenbereiche der anderen Projektmitglieder wird an dieser Stelle auf die Projektberichte von Alexander Mas (Mas, 2008) (Framework), Markus Dreyer (Dreyer, 2008) (Framework + P2P-Kommunikation), Sven Vollmer (Vollmer, 2008) (Bluetooth-Kommunikation), Andreas Herglotz (Herglotz, 2008) und Ralf Kruse (Kruse, 2008) (*PS Community Manager*) und Leif Hartmann und Jan Schönherr (Hartmann und Schönherr, 2008) (*PS Orientierungseinheit*) verwiesen.

2 Grundlagen

Für ein besseres Verständnis werden grundlegende Begriffe wie *Pervasive Computing*, *Pervasive Gaming* und *Framework* kurz beschrieben und auf weiterführende Literatur verwiesen.

Pervasive Computing - Beim *Pervasive Computing* wird mit der verfügbaren Informations- und Kommunikationstechnologie eine Durchdringung des Alltages angestrebt, bei der Informationen und Dienste jederzeit für den Benutzer verfügbar sind. Dieser Begriff wurde erstmalig von Marc Bregman in einem Interview (Bregman) beschrieben.

Pervasive Gaming - *Pervasive Gaming* bezeichnet einen Teilbereich des *Pervasive Computings* bei dem Spiele in der realen Welt durch Informations- und Kommunikationstechnologien angereichert werden. Auf diese Weise wird eine Brücke zwischen der virtuellen und der physischen Welt geschlagen (Schneider und Kortuem, 2001). Dazu werden Technologien aus dem Bereich des *Pervasive und Mobile Computings* unterstützend in traditionelle Spiele integriert, ein Spiel mit neuen Komponenten angereichert oder aber ganz neue Spiele entwickelt. Im Gegensatz zu klassischen Computerspielen, die in einer virtuellen Welt stattfinden, sind beim *Pervasive Gaming* physikalische Bewegungen und soziale Interaktion mit anderen Benutzern in der physikalischen Welt nötig (Magerkurth u. a., 2005).

Framework - Ein *Framework* bestimmt die Softwarearchitektur einer Applikation und stellt ein Grundgerüst von wiederverwendbaren Klassen und Komponenten zur Verfügung, wobei der Kontrollfluss beim *Framework* liegt. Die Klassen eines *Frameworks* arbeiten über definierte Schnittstellen zusammen und stellen dadurch einen wieder verwendbaren Entwurf dar (Johnson und Foote, 1988).

Besonders die Begriffe *Pervasive Computing* und *Pervasive Gaming* werden in der Forschung teilweise unterschiedlich betrachtet. Ein Überblick und eine Diskussion über die unterschiedlichen Definitionen wird in Nieuwdorp (2007) gegeben.

3 Projekt

In diesem Kapitel wird zu Beginn die Vision des Projektes vorgestellt. Anschließend wird auf den Ablauf des Projekts eingegangen und dann zwei Szenarien präsentiert, aus denen sich Anforderungen an ein Framework für *Pervasive Spiele* ableiten lassen.

3.1 Vision

Frameworks sollen die Entwicklung beschleunigen und ein Grundgerüst für die Entwicklung von konkreten Applikationen bereitstellen. Im Bereich des *Pervasive Gaming* existieren derzeit kaum Frameworks (vgl. Fetter u. a. (2007)), die als Unterstützung für die Entwicklung von *Pervasiven Spielen* eingesetzt werden können. Dadurch ist eine schnelle Realisierung und Validierung von neuen Spielideen und -konzepten nur schlecht möglich. Wiederkehrende Anforderungen wie z.B. die Ablausteuering der Anwendung oder die Behandlung von Ereignissen in *Pervasiven Spielen* werden bei jeder neuen Applikation zeitintensiv implementiert. Aus diesem Grund soll ein Framework für *Pervasive Spiele* entwickelt werden, welches ein Grundgerüst bereitstellt. Dabei soll das Framework die Integration der Technik der mobilen Endgeräte in ein *Pervasives Spiel* ermöglichen und die Verwendung vereinfachen.

3.2 Ablauf

Vor dem offiziellen Start des Projekts fand bereits eine kurze Vorbesprechung mit einigen Projektteilnehmern während der Semesterferien statt. Bei dieser Vorbesprechung wurden die Interessen der Projektteilnehmer abgestimmt und die Entwicklung eines Frameworks im Bereich des *Pervasive Gaming* beschlossen.

Während der ersten Treffen der Projektmitglieder wurden zunächst Instrumente und Regeln für die Projektorganisation und das Projektmanagement festgelegt und unscharfe Projektziele definiert. Dazu wurden z.B. für das Projektmanagement feste wöchentliche Termine für Statusbesprechungen eingeführt, ein Projektplan angelegt und Meilensteine definiert. Die Rolle des Projektleiters wurde von einem Projektmitglied für die Dauer des Projekts übernommen. Für die Projektorganisation wurden einige Qualitätsmaßnahmen zur Qualitätssicherung beschlossen, z.B. wurden Richtlinien für die Entwicklung schriftlich definiert, ein Dokumentationssystem aufgesetzt und eine Versionsverwaltungs-Software bereitgestellt. Parallel zur Projektorganisation fand gleichzeitig eine Orientierung und Evaluation z.B. der einzusetzenden Programmierplattform oder Hardware statt. Die Ergebnisse wurden dann im Projektteam vorgestellt und auf der Basis Entscheidungen für die

Auswahl getroffen. Da die meisten Projektmitglieder bisher keine Erfahrung mit der Entwicklung mobiler Anwendungen für mobile Endgeräte hatten, wurde außerdem ein Treffen mit den Projektteilnehmern des letzten Jahrgangs organisiert, um Schwierigkeiten zu erfahren und von den Erfahrungen der Kommilitonen zu profitieren.

Nach dem Abschluss der Orientierungsphase wurden die technischen Möglichkeiten der Programmierplattform und der mobilen Endgeräte durch die Projektmitglieder evaluiert und erste Erfahrungen in der Entwicklung mobiler Applikationen gesammelt. Auf dieser Grundlage wurden anschließend die Ziele der Projektveranstaltung durch die Projektmitglieder konkretisiert. Das Framework sollte durch zwei *Pervasive Spiele* validiert werden, die sich jedoch grundlegend in ihrer Funktionalität unterscheiden. Mit diesem Wissen und den unscharfen Anforderungen an ein *Pervasive Gaming Framework* wurde in einem ersten Iterationsschritt ein Frameworkansatz entwickelt, der zunächst grundlegende Funktionalität bereitstellt und die Steuerung einer Anwendung vorgibt. Gleichzeitig wurden die Spezifikationen für die zwei prototypischen *Pervasive Spiele* entwickelt und anschließend implementiert. Daraus haben sich dann weitere Anforderungen (wie z.B. eine allgemeine Ereignisbehandlung) an das Framework ergeben, die iterativ integriert wurden.

Zum Abschluss wurden die prototypischen *Pervasive Spiele* unter realistischen Bedingungen getestet. Die Ergebnisse der Projektveranstaltung wurden in einer Abschlussveranstaltung besprochen und mit den anfänglich definierten Anforderungen abgeglichen.

3.3 Szenarien

Damit ein konkretes Framework für *Pervasive Spiele* entwickelt werden konnte, wurden zwei Szenarien entwickelt, die in den folgenden Abschnitten vorgestellt werden. Aus diesen Szenarien konnten dann fachliche und technischen Anforderungen abgeleitet werden, die als Grundlage für die Entwicklung eines Frameworks für *Pervasive Spiele* herangezogen werden konnten.

3.3.1 PS Community Manager

Die Anwendung *PS Community Manager* soll Anwendern bei der Pflege und der Verwaltung ihrer Kontakte unterstützen und auf interessante Orte (engl. points of interest) in der Umgebung aufmerksam machen. In der Nähe befindliche Anwender werden durch die Anwendung von einander in Kenntnis gesetzt und können dann z.B. Nachrichten austauschen oder Treffen vereinbaren. So sind Treffen von

Anwendern denkbar, die sich sonst unter Umständen in großen Menschenmengen oder in der Bahn nicht getroffen hätten. Zusätzlich sollen die Positionen und Informationen zu Kontakten und interessanten Orten auf dem mobilen Endgerät persistent gehalten und auf eine zentrale Plattform im Internet übertragen werden können. Dadurch entsteht eine Gemeinschaft, die den Inhalt selbstständig erstellt, pflegt und Teil des Inhalts ist.

3.3.2 PS Orientierungseinheit

Das Spiel *PS Orientierungseinheit* basiert auf dem Spielprinzip einer Schnitzeljagd und soll z.B. neuen Kommilitonen das Gelände der HAW Hamburg und wichtige Orte auf dem Campus spielerisch näher bringen. Ein Spieler bekommt dazu auf dem mobilen Endgerät drei zufällige Hinweise angezeigt, die Orte beschreiben, die in der realen Welt gefunden werden müssen. Aus diesen Hinweisen wird einer durch den Spieler ausgewählt. Ob der Ort erfolgreich gefunden wurde, wird nach dem Typ des Hinweises entschieden. Entweder muss das Zielgebiet erreicht werden (Außenbereich) oder es muss ein Semacode gefunden und mit einer Kamera im mobilen Endgerät abfotografiert werden (Innen- und Außenbereich). Der Spieler, der am schnellsten alle Orte durch die Hinweise gefunden hat, gewinnt das Spiel. Während des Spiels soll ein Spieler jederzeit seine Position auf einer Karte ansehen können und bei Orten, die nur durch das Erreichen des Zielgebiets gelöst werden, zusätzlich noch die Richtung zum Zielgebiet (nur im Außenbereich) sehen können. Außerdem werden die Routen für unterschiedliche Spiele auf einer zentralen Plattform im Internet vorgehalten. Auf diese Plattform sollen auch Zwischenergebnisse und das Endergebnis übertragen werden können. So können auch Wettbewerbe durchgeführt werden.

3.4 Anforderungen

In diesem Abschnitt werden die fachlichen und technischen Anforderungen an die Szenarien aus Abschnitt 3.3 identifiziert und als grundlegende Anforderungen an das zu entwickelnde Framework festgeschrieben.

3.4.1 Fachliche Anforderungen

Die fachlichen Anforderungen werden für die Applikationen *PS Community Manager* und *PS Orientierungseinheit* im Folgenden grob skizziert:

PS Community Manager:

- Die Verwaltung der Benutzer und der Orte muss über eine zentrale Plattform im Internet möglich sein.

- Die Verwaltung von Kontaktinformationen und Orten auf dem mobilen Endgerät und das Übertragen auf die zentrale Plattform im Internet muss möglich sein.
- Die Kommunikation mit anderen Benutzern muss möglich sein.
- Die Nähe von bestimmten Orten oder anderen Benutzern muss auf dem mobilen Gerät signalisiert werden.
- Die Anwendung soll unterbrechbar sein und fortgesetzt werden können.

PS Orientierungseinheit:

- Die Verwaltung der Highscorelisten, unterschiedlicher Routen, Spiele und der Spieler muss über einer zentrale Plattform im Internet möglich sein.
- Wird ein gefundener Semacode abfotografiert, muss der Spieler über Erfolg oder Misserfolg unterrichtet werden.
- Wird ein gesuchter Ort erreicht, muss der Spieler benachrichtigt werden.
- Ein Spiel soll unterbrechbar sein und fortgesetzt werden können.
- Der aktuelle Standort des Spielers soll auf einer Karte auf dem mobilen Endgerät visualisiert werden.

Die fachlichen Anforderungen der beiden Szenarien sind teilweise identisch und können bei der weiteren Betrachtung der technischen Anforderungen zusammengefasst werden.

3.4.2 Technische Anforderungen

Aus den fachlichen Anforderungen aus Abschnitt 3.4.1 ergeben sich die technischen Anforderungen, die aufgrund der zahlreichen Gemeinsamkeiten gemeinsam aufgeführt werden und gleichzeitig auch die Anforderungen an das zu entwickelnde Framework darstellen:

- Leicht integrierbare Benutzer- und Spielerverwaltung.
- Die Daten müssen auf dem mobilen Gerät und auf dem Server persistent gehalten werden können.
- Die Kommunikation mit einer zentralen Instanz im Internet und Peer-to-Peer mit anderen Mitspielern muss möglich sein.

- Die aktuelle Position muss verfügbar sein.
- Damit unterschiedliche Applikationen unterstützt werden können, werden einfache Konfigurationsmöglichkeiten benötigt.

Auf Grundlage der identifizierten Anforderungen wird im folgenden Kapitel das Konzept von *Pervasive Spine* erläutert und anschließend Aspekte aus der Realisierung vorgestellt.

4 Konzept

Auf Basis der technischen Anforderungen aus Abschnitt 3.4.2 wird in diesem Kapitel ein Konzept für ein Framework für *Pervasive Spiele* vorgestellt und anschließend die Realisierung beschrieben.

4.1 Architektur des Frameworks

Die Architektur für das Framework ist aus den technischen Anforderungen aus Abschnitt 3.4.2 iterativ entwickelt worden. Eine Übersicht über die Architektur ist in Abbildung 1 dargestellt.

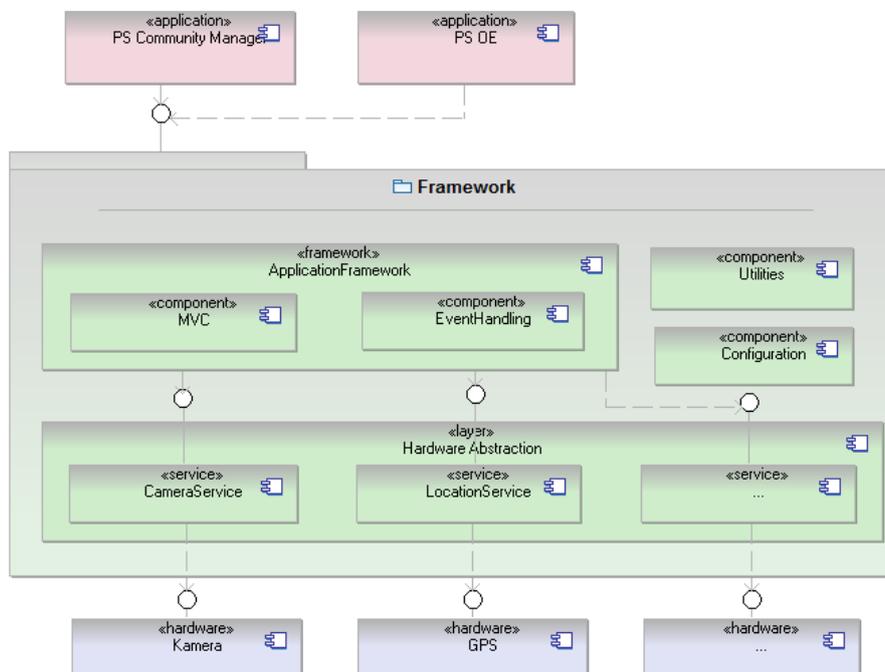


Abbildung 1: Pervasive Spine Architektur

Die zwei skizzierten Szenarien aus Abschnitt 3.3 setzen auf das Framework auf und verwenden die angebotenen Dienste. Das Framework ist zweischichtig aufgebaut und lässt sich in eine Applikations- und eine Hardwareabstraktionsschicht aufteilen. Die Aufgaben der einzelnen Schichten werden im Folgenden kurz erläutert:

Applikationsschicht - In der Applikationsschicht wird die Ablaufsteuerung der Applikation durch eine Realisierung des Model-View-Controller (Krasner und Pope, 1988) Musters vorgegeben. Dadurch können Applikationen nach

dem Baukastenprinzip zusammengesteckt (vgl. Abbildung 2) werden und der Fokus der Entwicklung auf die Problemdomäne gelegt werden.

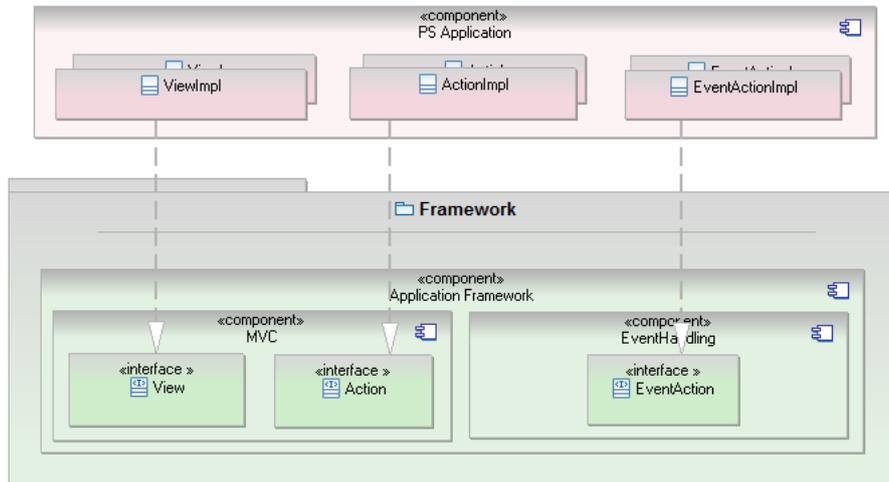


Abbildung 2: Entwicklung von Applikationen

Zusätzlich stellt die Applikationsschicht einen generischen Ansatz für die Registrierung und Verarbeitung von Ereignissen auf dem mobilen Endgerät bereit, der in der Ablaufsteuerung integriert ist.

Hardwareabstraktionsschicht - Die verfügbare Hardware wird in Form von Diensten abstrahiert und dem Applikationsframework oder den konkreten Applikationen transparent bereitgestellt. Dabei orientieren sich die Dienste nicht an der konkreten Technik der mobilen Endgeräte, sondern an der Funktionalität, die bereitgestellt werden soll. Ein Dienst kann so je nach Verfügbarkeit unterschiedliche Technik nutzen, um eine Dienstanfrage zu erfüllen. Zusätzlich kann mögliche Heterogenität der verwendeten mobilen Endgeräte transparent vor den höheren Schichten versteckt werden.

Neben den zwei Schichten werden von dem Framework zusätzlich einige Querschnittsfunktionalitäten bereitgestellt, die Hilfsmittel für die Entwicklung von Applikationen darstellen oder eine Konfiguration einer Applikation durch Parameter außerhalb des Source-Codes ermöglichen.

4.2 Realisierung

In diesem Kapitel wird auf die Realisierung einiger der Komponenten aus der Architekturübersicht aus Abbildung 1 eingegangen, die vom Autor im Projekt umgesetzt worden sind.

4.2.1 Infrastruktur

Pervasive Spine und die Beispielapplikationen wurden auf der J2ME-Programmierplattform auf Nokia E70 Endgeräten realisiert. Dabei wurde auf die Konfiguration *Connected Limited Device Configuration (CLDC)* und dem Profil *Mobile Information Device Profile 2.0 (MIDP)* aufgesetzt. Zusätzlich wurden noch weitere Erweiterungen (Java Specification Requests (JSRs)) aus dem Java Community Process (JCP) eingesetzt, die zusätzliche Funktionalitäten anbieten. Durch den Einsatz dieser Spezifikationen sollte eine höhere Portabilität zwischen heterogenen mobilen Endgeräten erreicht werden. Als integrierte Entwicklungsumgebung wurde EclipseME verwendet und der Emulator aus dem Sun Java Wireless Toolkit 2.5.2 for CLDC integriert. Auf die eingesetzten Technologien für die Serverkomponenten wird an dieser Stelle nicht weiter eingegangen.

4.2.2 Framework

Damit die Ablaufsteuerung einer Applikation durch das *Pervasive Spine* Framework übernommen werden kann, müssen durch das Framework vorgegebene Schnittstellen implementiert werden:

View - *Views* sind für die Darstellung der Benutzeroberfläche verantwortlich und können von anderen *Views*, *Actions* oder *EventActions* aufgerufen werden. Aktionen auf der Oberfläche können wiederum auf andere *Views*, *Actions* weiterleiten.

Action - *Actions* kapseln die Geschäftslogik einer Applikation (z.B. nach dem Auslösen einer Aktion durch das Drücken eines Knopfs auf der Benutzeroberfläche). Nachdem die Geschäftslogik abgearbeitet wurde, kann eine *Action* als Resultat eine *View* festlegen, die als Nächstes angezeigt werden soll.

EventAction - *EventActions* werden beim Auftreten von Ereignissen aufgerufen und kapseln wie *Actions* die Geschäftslogik. Auch hier kann nach Abarbeitung der Geschäftslogik auf eine konfigurierte *View* weitergeleitet werden. Für jedes Ereignis existiert eine spezielle *EventAction* Schnittstelle, die die Ereignisdaten zur Verfügung stellt.

Die einzelnen *Views*, *Actions* und *EventActions* werden in einer zentralen XML-Konfigurationsdatei (der so genannten *spine.xml*) zu einer konkreten Anwendung verwoben. Da keine direkten Abhängigkeiten zwischen den einzelnen Komponenten bestehen, ist eine Wiederverwendung in anderen Applikationen oder eine Mehrfachverwendung möglich.

In der XML-Konfigurationsdatei werden die konkreten *Views* und *Actions* allgemein deklariert und können dann innerhalb von unterschiedlichen *Flowpoints* referenziert werden. *Flowpoints* stellen einen Ablaufpunkt in einer Applikation dar, an dem bestimmte Ereignisse auftreten können, die als *Events* innerhalb der XML-Konfigurationsdatei definiert werden. Zu einem *Event* gehört ein passender *EventHandler*, der sich beim Betreten eines *Flowpoints* für Ereignisse anmeldet oder aber sich von der Benachrichtigung abmeldet. Zusätzlich sind jedem *Event* für die Benachrichtigung der Applikation *EventActions* zugeordnet. Dieses Konzept ermöglicht eine Art regelbasiertes Kontextbewusstsein und vereinfacht dadurch die Applikationsentwicklung.

Für die Behandlung von Ausnahmen kann ein zentraler *ExceptionHandler* für eine Applikation festgelegt werden, der im Fehlerfall aufgerufen wird.

4.2.3 LocationService

Damit die Szenarien aus Abschnitt 3.3 realisiert werden konnten, wurde eine Möglichkeit für die Bereitstellung der aktuellen Position benötigt. Dieser Dienst lässt sich als *LocationService* in die Hardwareabstraktionsschicht in der Architekturübersicht aus Abbildung 1 einordnen.

Zunächst wurde eine Schnittstelle mit der benötigten Funktionalität spezifiziert und anschließend eine Implementierung auf Basis der J2ME Erweiterung JSR-179 Location API hinzugefügt. Obwohl die J2ME Erweiterung bereits von der konkret verwendeten Hardware abstrahiert, wurde eine weitere Abstraktionsschicht eingeführt, die individuelle Anpassungen ermöglicht. Das Konzept der zusätzlichen Abstraktionsschicht hatte sich bereits im Verlauf der Projektveranstaltung bewährt, da die J2ME Erweiterung einige Teile der Programmierschnittstelle gar nicht implementiert hat (z.B. der Näherungsmechanismus an einen Ort) oder die Implementierung Abstürze auf den mobilen Endgeräten verursacht hat. Die Anpassungen waren dann für die höheren Schichten transparent durchführbar.

Neben dem *LocationService* wurden noch weitere Dienste durch andere Projektmitglieder realisiert wie z.B. ein *AudioService*, ein *CameraService*, ein *BarcodeService*, ein *PersistenceService* für die Speicherung von Daten auf dem mobilen Endgerät, ein *TimerService* und der *PeerCommunicationService* für die Peer-to-Peer-Kommunikation.

4.2.4 Configuration

Um die Konfiguration von Frameworkkomponenten und Applikationen zu vereinfachen wurde eine *Configuration* Schnittstelle bereitgestellt, die Konfigurationsparameter aus einer Datei ausliest und dadurch flexible Anpassungen von Parametern ohne Änderungen am Source-Code erlaubt. Die Schnittstelle orientiert sich an dem *Apache Commons Configuration* Projekt für Java SE, wurde aber für den Einsatz auf mobilen Endgeräten mit J2ME optimiert.

4.2.5 Utilities

Zur Unterstützung der Entwicklung wurden zu Beginn des Projekts einige Hilfsmittel implementiert, die nachfolgend kurz aufgeführt werden:

AbstractFactory - Die *AbstractFactory* instanziiert Klassen auf Basis des Klassennamens und kapselt die Verarbeitung von Ausnahmen, die bei der Instanziierung auftreten können. Die Dienste der Hardwareabstraktionsschicht stellen über konkrete *Factories* ihre Schnittstellen zur Verfügung und ermöglichen auf diese Weise eine flexible Konfiguration.

Logging - Als zusätzliche Unterstützung bei der Fehlersuche im Emulator und auf dem mobilen Gerät wurde für das Projekt eine vereinfachte Version des *Apache Commons Logging Framework* entwickelt und eingesetzt.

XML Parser - Da die Ablaufsteuerung der Applikationen und die Ereignisbehandlung über eine XML-Datei konfiguriert werden, wurde ein allgemeiner Parser für XML-Dokumente als Hilfsmittel zur Verfügung gestellt. Dafür wurde wie bei den Diensten der Hardwareabstraktionsschicht eine Schnittstelle nach dem Push-Prinzip entwickelt, da J2ME und die Konfiguration CLDC diese Funktionalität nicht anbieten. Im Rahmen des Projekts wurde die Schnittstelle durch den SAX-Parser aus der J2ME Erweiterung JSR 172 J2ME Web Services implementiert, der für die Kommunikation über WebServices eingesetzt wird. Alternativ gibt es für J2ME noch weitere XML-Parser, die aber aufgrund des vorhandenen Standards nicht weiter betrachtet wurden und zusätzlich weiteren Speicherplatz auf dem mobilen Endgerät verbraucht hätten.

Der Einsatz der Hilfsmittel war nicht auf das Framework begrenzt, sondern hat auch die Applikationen umfasst. So wurden grundlegende Mechanismen zentral bereitgestellt.

5 Zusammenfassung und Ausblick

Im Rahmen der Projektveranstaltung wurde das Framework *Pervasive Spine* entwickelt und die Funktion durch die Applikationen *PS Orientierungseinheit* und *PS Community Manager* validiert. Die Projektveranstaltung kann somit als erfolgreich bewertet werden. Die Anforderungen an die Szenarien aus Abschnitt 3.4 wurden bis auf die Peer-to-Peer-Kommunikation umgesetzt. Aufgrund des begrenzten Zeitraums konnten aber nicht alle Aspekte in das Framework integriert werden. So ist ein einheitlicher Middleware-Ansatz für die Kommunikation mit einer zentralen Internetplattform bisher nur auf Applikationsebene umgesetzt worden. Auch wurden bisher keine fachlichen Komponenten in das Framework integriert, die z.B. Komponenten für die Benutzer- oder Spielverwaltung umfassen und von den prototypischen Applikationen einfach verwendet werden können. Das vorgestellte Konzept der *Flowpoints* aus Abschnitt 4.2.2 ermöglicht zwar eine Art Kontextbewusstsein der Applikationen, ist aber nur eingeschränkt erweiterbar (z.B. wird bisher nur auf definierte Ereignisse reagiert, auf andere Änderungen im Kontext des Benutzers jedoch nicht) und sollte durch einen allgemeineren Ansatz abgelöst werden. Zusätzlich ist bei der Entwicklung auf den mobilen Endgeräten verstärkt die Problematik des Testens aufgefallen, die sich aus dem pervasiven Ansatz und dem Einsatz in mobilen Umgebungen ergibt. Auch der Schritt aus der Emulatorumgebung auf das konkrete mobile Endgerät hat häufig zu Problemen geführt, die zeitintensiv gelöst werden mussten.

Die Arbeitsatmosphäre innerhalb des Projektteams war über die gesamte Projektlaufzeit hinweg sehr konzentriert und zielorientiert. Architektur- und Projektentscheidungen wurde im Projektteam und unter den Projektmitgliedern konstruktiv diskutiert und dadurch von allen Projektmitgliedern getragen. Dadurch wurde wenig Zeit für unnötige Diskussionen aufgewendet und es konnten viele der Anforderungen umgesetzt werden.

Durch das Projekt konnten die meisten Projektmitglieder erste Erfahrung in der Entwicklung *Pervasiver Spiele* in mobilen Umgebungen machen und Anregungen für mögliche Abschlussarbeiten sammeln.

Abbildungsverzeichnis

1	Pervasive Spine Architektur	8
2	Entwicklung von Applikationen	9

Literatur

- [Bregman] BREGMAN, Marc: Interview With Marc Bregman. The Convenience of Small Devices: How Pervasive Computing Will Personalize EBusiness. In: *IBM online Think Research*
- [Dreyer 2008] DREYER, Markus: Pervasive Spine, Hochschule für Angewandte Wissenschaften Hamburg, 2008
- [Fetter u. a. 2007] FETTER, Mirko ; ETZ, Markus ; BLECHSCHMIED, Heiko: Mobile chase - towards a framework for location-based gaming. In: *GRAPP (AS/IE)*, INSTICC - Institute for Systems and Technologies of Information, Control and Communication, 2007, S. 98–105
- [Hartmann und Schönherr 2008] HARTMANN, Leif ; SCHÖNHERR, Jan: Pervasive Spine, Hochschule für Angewandte Wissenschaften Hamburg, 2008
- [Herglotz 2008] HERGLOTZ, Andreas: Pervasive Spine, Hochschule für Angewandte Wissenschaften Hamburg, 2008
- [Johnson und Foote 1988] JOHNSON, R. ; FOOTE, B.: Designing reusable classes. In: *Journal of Object-Oriented Programming*, 1988, S. 22–35
- [Krasner und Pope 1988] KRASNER, Glenn E. ; POPE, Stephen T.: A cookbook for using the model-view controller user interface paradigm in Smalltalk-80. In: *J. Object Oriented Program.* 1 (1988), Nr. 3, S. 26–49. – ISSN 0896-8438
- [Kruse 2008] KRUSE, Ralf: Pervasive Spine, Hochschule für Angewandte Wissenschaften Hamburg, 2008
- [Magerkurth u. a. 2005] MAGERKURTH, Carsten ; CHEOK, Adrian D. ; MANDRYK, Regan L. ; NILSEN, Trond: Pervasive games: bringing computer entertainment back to the real world. In: *Comput. Entertain.* 3 (2005), Nr. 3, S. 4–4
- [Mas 2008] MAS, Alexander: Pervasive Spine, Hochschule für Angewandte Wissenschaften Hamburg, 2008
- [Nieuwdorp 2007] NIEUWDORP, Eva: The pervasive discourse: an analysis. In: *Comput. Entertain.* 5 (2007), Nr. 2, S. 13
- [Schneider und Kortuem 2001] SCHNEIDER, J. ; KORTUEM, G.: How to Host a Pervasive Game - Supporting Face-toFace Interactions in Live-Action Role-playing, 2001

Literatur

[Vollmer 2008] VOLLMER, Sven: Pervasive Spine, Hochschule für Angewandte
Wissenschaften Hamburg, 2008