



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung Seminar

Maik Weindorf

Context-Aware Mobile Computing

Inhaltsverzeichnis

| | |
|----------------------------------------------|-----------|
| Inhaltsverzeichnis | I |
| Abbildungsverzeichnis | I |
| 1 Einleitung | 1 |
| 2 Grundlagen | 2 |
| 2.1 Context | 2 |
| 2.2 Context Beispiele | 3 |
| 2.3 Entwicklung | 4 |
| 3 Aktuelle Arbeiten | 5 |
| 3.1 Hydrogen Context-Framework | 5 |
| 3.2 Sensor-based Context-Awareness | 5 |
| 3.3 Weitere interessante Arbeiten | 6 |
| 4 Thesis Outline | 7 |
| 4.1 Potentielle Themen | 7 |
| 4.2 Thema der Masterarbeit | 8 |
| 5 Fazit | 12 |
| Literaturverzeichnis | 13 |

Abbildungsverzeichnis

| | | |
|-----|-----------------------------------------------------------|----|
| 3.1 | Hydrogn Context Framework [Hofer et al., 2003] | 6 |
| 4.1 | Pervasive Gaming Framework (Client Architektur) | 10 |
| 4.2 | Pervasive Gaming Framework (Server Architektur) | 11 |

Kapitel 1

Einleitung

Mit der zunehmenden Verbreitung leistungsfähiger mobiler Geräte, ergeben sich zusätzliche Anforderungen an Computersysteme. Dies gilt insbesondere für sog. ultra mobile Geräte, wie z.B. PDAs und Smartphones, die sich in vielen Aspekten deutlich von voll ausgestatteten Laptops oder PCs unterscheiden. Diese Geräte haben aufgrund ihrer begrenzten Akku-Leistung eine eingeschränkte Laufzeit. Ihre Bedienbarkeit ist durch kleine Tastaturen oder Touch-Bildschirme ebenfalls eingeschränkt. Außerdem verfügen sie noch immer über recht begrenzte Rechenleistung und Arbeitsspeicher, im Vergleich zu aktuellen PCs. Hinzu kommt eine eingeschränkte Konnektivität durch langsame oder instabile Netzwerkverbindungen. Abgesehen von diesen technischen Einschränkungen ergeben zusätzliche Herausforderungen, durch wechselnde und dynamische Rahmenbedingungen, unter denen ultra mobile Geräte verwendet werden. [Hofer et al., 2003] [Yan und Sere, 2004]

Der hohe Mobilitätsgrad dieser Geräte in Kombination mit wechselnden Rahmenbedingungen, ermöglicht aber auch eine völlig neue Art von Anwendungen und bietet viel Potential für eine deutliche Verbesserung der Usability.

Das Schlüsselwort in diesem Zusammenhang lautet "Context-Awareness".

Laut [Schmidt et al., 1998] beschreibt Context¹:

"...a situation and the environment a device or user is in."

Context-Awareness bedeutet somit das Bewusstsein über diese Situation und Rahmenbedingungen. In Kapitel 2.1 folgt eine nähere Betrachtung des Begriffs "Context". Diese Ausarbeitung beschäftigt sich speziell mit "Context-Aware Mobile Computing". Das Gebiet der Context-Awareness ist sehr weit. Im Rahmen dieser Ausarbeitung kann daher nicht auf alle Teilaspekte eingegangen werden.

¹In dieser Ausarbeitung wird durchgängig der Begriff "Context" verwendet. Von einer Verwendung des deutschen Begriffs "Kontext" wird aus Konsistenzgründen abgesehen.

Kapitel 2

Grundlagen

In diesem Kapitel wird zunächst der Begriff "Context" näher betrachtet. Des Weiteren werden Einsatzmöglichkeiten von Context-Awareness in mobilen Umgebungen aufgezeigt.

2.1 Context

"There is more to Context than Location" [Schmidt et al., 1998]. Context-Awareness in mobilen Umgebungen wird häufig gleichgesetzt mit Location-Awareness. Es gibt jedoch wesentlich mehr verwertbare Context-Informationen über die reine Location Information hinaus. In [Schmidt et al., 1998] wird eine Unterteilung in menschliche Faktoren und physikalische Umgebung vorgenommen. Menschliche Faktoren beschreiben dabei z.B. Benutzereinstellungen, soziales Umfeld und die konkrete Aufgabe eines Nutzers (Task). Die physikalische Umgebung hingegen beschreibt unter Anderem Umgebungsbedingungen, wie z.B. Licht, Temperatur und Beschleunigung, sowie Infrastruktur Informationen und natürlich auch die Location. All diese Faktoren lassen sich messen, bzw. ermitteln und unterschiedlich granular betrachten. So kann man beispielsweise Intensität und Wellenlänge als messbare Faktoren von Licht auffassen. Hofer [Hofer et al., 2003] hingegen nimmt eine Unterteilung in physikalischen und logischen Context vor. Dabei beschreibt der physikalische Context "low level" Informationen, wie z.B. eine GPS-Position oder eine Temperatur. Der logische Context hingegen beschreibt "high level" Informationen, wie beispielsweise Straßennamen oder Temperatur-Umschreibungen, wie "heiß" und "kalt". Somit wird schnell klar, dass die Definitionen von Context weder einheitlich noch eindeutig sind.

Es drängt sich die Frage auf, welches Ziel mit der Einbeziehung von Context in die Logik von Anwendungen erreicht werden soll. Eine aktuelle Studie des BSI [BSI, 2006] nennt als Folge von Context-Awareness einen erhöhten Komfort für die Nutzer und auf längere Sicht einen weiteren Schritt in Richtung Autonomie.

(vgl. Autonomic Computing [Ganek und Corbi, 2003] [IBM, 2001])

2.2 Context Beispiele

Im Folgenden werden einige Anwendungsbeispiele für Context-Awareness in mobilen Umgebungen aufgezeigt.

GPS Navigationsgeräte Navigationsgeräte sind typische, wenn auch recht einfache Anwendungsbeispiele von Context-Awareness. Sie beziehen hauptsächlich die gegenwärtige Position in ihre Applikationslogik ein und verbinden diese mit einem logischen Context (z.B. einer Landkarte). Abgesehen davon verwenden moderne Navigationsgeräte jedoch auch noch weniger offensichtliche Context-Informationen. Vor allem die aktuelle Geschwindigkeit wird meist als zusätzliche Information genutzt. So ist es beispielsweise möglich, dem Benutzer Ansagen bei hoher Geschwindigkeit früher mitzuteilen, als bei geringer Geschwindigkeit. Außerdem kann die Lautstärke der Sprachausgabe an die gegenwärtige Geschwindigkeit angepasst werden. Dabei wird davon ausgegangen, dass die Umgebungsgeräusche mit zunehmender Geschwindigkeit ebenfalls zunehmen.

Spamfilter Das Thema Spam, bzw. Spambekämpfung wird zunehmend wichtiger. Einfache regelbasierte Spamfilter sind inzwischen kaum noch in der Lage, Spam Mails wirkungsvoll zu unterdrücken. Moderne Verfahren setzen unter Anderem auf Context-Awareness. Dabei wird beispielsweise berücksichtigt, ob die Mail von einem Bekannten kommt. Wenn ja, wird die Mail positiv bewertet und es bedarf weit mehr einschlägiger Schlüsselwörter und anderer Merkmale, bis die Mail als Spam eingestuft wird. In diesem Beispiel wird also der soziale Context mit einbezogen.

personalisierte Suche Diese Art von Suche, wie man sie z.B. aus Onlineshops wie Amazon kennt, ist eines der bekanntesten Anwendungsgebiete von Context-Awareness. Im Gegensatz zu anderen Anwendungsgebieten von Context-Awareness (wie z.B. Spamfilter) ist es in diesem Fall ganz offensichtlich für die Anwender, dass Context-Informationen genutzt werden.

Loadbalancer Ein weniger offensichtliches Anwendungsgebiet von Context-Awareness sind Loadbalancer für Web- oder DB-Server. Loadbalancer gibt es schon deutlich länger als den Begriff der Context-Awareness und man findet die beiden Begriffe selten in direkter Verbindung. Jedoch nutzen Loadbalancer verschiedene Context-Informationen, wie z.B. die aktuelle Last im System, die aktuelle Anzahl an Clients und darüber hinaus "Erfahrungswerte" aus der Vergangenheit.

2.3 Entwicklung

Der Begriff Context-Awareness gewinnt zunehmend an Bedeutung. Wie schon beschrieben (2.1), ist seine Definition nicht eindeutig. Es entsteht leicht der Eindruck, dass es sich bei "Context-Awareness" um ein völlig neues Thema handelt. Der Begriff wurde jedoch schon Mitte der Neunziger Jahre geprägt. Einige der unter 2.2 beschriebenen Beispiele für Context-Awareness sind sogar bedeutend älter als der Begriff selbst. Es stellt sich also die Frage, warum dieses Thema momentan so viel Beachtung findet. Die Beispiele unter 2.2 zeichnen sich alle dadurch aus, dass jeweils nur sehr begrenzte Context-Informationen genutzt werden. Wenn man die aktuellen Entwicklungen betrachtet (siehe 3), wird deutlich, dass der Trend im Bereich der Context-Awareness dahin geht, viele verschiedene Context-Informationen zu nutzen. Dazu werden die verschiedenen Informationen meist akkumuliert und in abstraktere Informationen überführt. Aufschwung bekommt das Thema zusätzlich durch die zunehmende Verbreitung leistungsfähiger mobiler Geräte.

Kapitel 3

Aktuelle Arbeiten

In diesem Kapitel werden einige Konzepte vorgestellt, die das Ziel verfolgen, Aspekte von Context Awareness zu verwirklichen. Da diese Ausarbeitung als Schwerpunkt die in Kapitel 4 beschriebene Thesis Outline hat, werden die verschiedenen Konzepte nur kurz vorgestellt. Für eine nähere Betrachtung sei an dieser Stelle auf die AW2 Ausarbeitung verwiesen (in Kürze zu finden unter [UbiComp]).

3.1 Hydrogen Context-Framework

Abbildung 3.1 zeigt die Grobarchitektur des Frameworks [Hofer et al., 2003]. Es handelt sich bei dem Hydrogen Context-Framework um ein Java basiertes Framework. Es ist auf eine modulare Anbindungen verschiedenster Sensoren (bzw. Informationsquellen) und Anwendungen ausgelegt. Zentraler Punkt des Frameworks ist ein Context Server, der die verschiedenen Informationen der Sensoren akkumuliert und den Anwendungen in abstrahierter Form zur Verfügung stellt.

3.2 Sensor-based Context-Awareness

...for Adaptive PDA User Interfaces [Schmidt et al., 1998]. Diese Arbeit beschäftigt sich, mit den Einsatzmöglichkeiten von Sensoren zur automatisierten Anpassung von PDA Displays, bzw. der darauf dargestellten Benutzerschnittstellen. Als Beispiele werden Lichtsensoren genannt, anhand derer eine automatische Anpassung der Display Helligkeit an das Umgebungslicht ermöglicht wird. Zusätzlich werden Lagesensoren betrachtet, mit deren Hilfe sich z.B. die Darstellung auf einem Display stets "richtig" herum drehen lässt. D.h. wenn ein Benutzer einen PDA dreht, muss er die Darstellung nicht manuell anpassen.

Kapitel 4

Thesis Outline

In diesem Kapitel wird beschrieben, welche Teilbereiche des Themas "Context-Aware Mobile Computing" sich im Allgemeinen für eine Masterarbeit anbieten und welcher Ansatz im Konkreten weiter verfolgt werden soll.

4.1 Potentielle Themen

Context-Wahrnehmung durch Nutzer Dieser Ansatz wäre mehr auf Usability und Untersuchung von Mensch-Maschine Interaktion ausgelegt, als auf technische Aspekte. Interessante Fragen in diesem Zusammenhang wären, ob ein Benutzer eine Anwendung, die sich unter verschiedenen Rahmenbedingungen unterschiedlich verhält eher als verwirrend oder als nützlich empfindet, bzw. was eine Context sensitive Anwendung erfüllen muss, um nicht auf Ablehnung zu stoßen.

Context-Repräsentationssprache Context Informationen so zu repräsentieren, dass sie sinnvoll für Anwendungen genutzt werden können, ist eine nicht triviale Aufgabe. Im Bereich der Context-Awareness werden Context Informationen häufig abstrakt und natürlich-sprachlich beschrieben. Für konkrete Anwendungen sind solche Beschreibungen eher ungeeignet. Es stellt sich daher die Frage, wie diese Informationen standardisiert und formal beschrieben werden können. Rein regelbasierte Ansätze helfen dabei nur bedingt weiter. Für ernsthafte Lösungen müsste man sich mit Theorien der Wissensrepräsentation, Ontologien und KI Ansätzen beschäftigen. Dies könnte zu einem erhebliche Aufwand führen, der den Rahmen einer Masterarbeit sprengen würde.

Evaluierung bestehender Context Manager und Frameworks Eine reine Evaluierung bestehender Lösungen wäre theoretisch die einfachste Art, sich mit dem Thema

”Context-Aware Mobile Computing” zu beschäftigen. Da es sich bei diesem Thema um ein offenes Forschungsgebiet handelt, finden sich jedoch kaum einsatzfähige Lösungen, die man vergleichen könnte. Vielfach handelt es sich lediglich um Konzepte oder Prototypen. Hinzu kommt, dass man zwar viele Veröffentlichungen findet, jedoch meist so gut wie keine Möglichkeit besteht an die dort beschriebene Prototypen oder Produkte zu kommen.

Konkretisierung: spezielle Einsatzgebiete für Context Eine Betrachtung von ”Context-Aware Mobile Computing” im Zusammenhang mit anderen Themen, wie z.B. BPEL oder Spiele, hätte den Vorteil, dass der Problembereich stark eingegrenzt würde. Somit könnte ein Einsatzgebiet sehr genau betrachtet werden, was bei einem generellen Ansatz nicht immer möglich ist.

Generalisierung: Context als Spezialfall von anderen Themen Context findet sich als Teilbereich in vielen anderen Themengebieten, wie beispielsweise Autonomic Computing, Self Management und einigen Teilbereichen der KI, wieder. Eine Masterarbeit, in der eine Betrachtung von Context als Spezialfall von anderen Themen erfolgen würde, hätte mit hoher Wahrscheinlichkeit einen eher theoretischen Charakter.

Entwurf eines Context Servers (Dienst-Anbieter) Der Entwurf eines Context Servers erfordert vor allem eine saubere Definition der Schnittstellen zu Sensoren und Anwendungen (vgl. 3.1). Eine Besondere Herausforderung wäre bei diesem Ansatz die Akkumulation und Abstraktion von Informationen, sowie die Context-Repräsentation (vgl. 4.1).

Entwurf eines Context Frameworks siehe 4.2

4.2 Thema der Masterarbeit

Aufgrund persönlicher Interessenschwerpunkte und den Erfahrungen aus dem AW2 Projekt ”Pervasive Gaming Framework” (Bericht in Kürze zu finden unter [UbiComp]) kommen für die Masterarbeit nur die Themenbereiche ”Context-Server” oder ”Context-Framework” infrage. Zum jetzigen Zeitpunkt erscheint dabei ein Framework als interessanter. Ein Context-Framework bietet generell mehr Möglichkeiten und Freiheiten als ein Context-Server, da nicht nur eine Kommunikation über externe Schnittstellen erfolgt, sondern vom Framework ein Kontrollfluss vorgegeben werden kann und eine engere Kopplung der einzelnen Komponenten möglich ist. Eine enge Kopplung ist im Allgemeinen aus Software Engineering Sicht unerwünscht, für Context-Awareness jedoch nicht immer zu vermeiden. Dies haben nicht zuletzt die Erfahrungen aus dem AW2 Projekt gezeigt. Die folgenden Erläuterungen beziehen sich daher hauptsächlich auf den Framework-Ansatz.

Um möglichst viel Nutzen aus dem AW2 Projekt für die Masterarbeit ziehen zu können, wird die Implementierung im Rahmen der Arbeit mit dem Microsoft .Net Compact Framework und C# erfolgen. Die Abbildungen 4.1 und 4.2 zeigen eine grobe Architektur des AW2 "Pervasive Gaming Framework". Einige Komponenten wie z.B. Communication-Layer, Location-Layer und Context-Manager werden sich für die Masterarbeit zumindest teilweise wiederverwenden lassen.

Der Schwerpunkt wird auf der Gesamtarchitektur des Frameworks liegen. Die Implementierung wird hauptsächlich der Evaluierung der theoretischen Überlegungen dienen und ist derzeit als Prototyp geplant. Sowohl bei der Architektur, als auch bei der Implementierung, wird der Fokus voraussichtlich auf der Client-Seite liegen. Aus Sicht von "Context-Aware Mobile Computing" sind die "mobilen" Komponenten, also die Clients wesentlich interessanter, als die Server-Komponenten. Weitere Aspekte der Arbeit werden der Entwurf von Definierten Schnittstellen für externe Agenten und Sensoren sein (vgl. 3.1). Wie schon beschrieben ist das Thema der Kontextrepräsentation nicht trivial. Für eine Implementierung ist eine Repräsentation von Context jedoch unumgänglich. Um das Risiko und den Aufwand zu begrenzen ist derzeit eine einfache Lösung vorgesehen, die im Wesentlichen auf die Verwendung von erweiterbaren Context-Objekten setzt. Dieser Ansatz wurde bereits im AW2 Projekt verfolgt und wird im Projektbericht näher beschrieben.

Die Beschreibung der Pläne für die Masterarbeit war bisher eher abstrakt. Um zu verdeutlichen, was durch die zu entwerfende Lösung erreicht werden soll, werden im Folgenden einige Anwendungsbeispiele genannt.

Location Awareness Die Informationen über Positionen soll vom Framework verwaltet werden. Dabei soll von der Lokalisierungs-Technologie wie GPS, IMAPS oder Cricket abstrahiert werden. Anwendungen, die mit dem Framework realisiert werden, müssen sich dann nicht mit den technischen Eigenheiten der verschiedenen Technologien beschäftigen und können außerdem durch Eventhandling über Positionsänderungen, Signalqualität usw. informiert werden. Somit müssen Anwendungen nur reaktiv auf Änderungen eingehen.

Client-Server Wie in Kapitel 1 beschrieben, ist die Konnektivität von mobilen Geräten nicht durchgehend und nicht mit konstanter Qualität gegeben. Aus Sicht von Context-Awareness bedeutet dies, dass Informationen über Dienstgüte, Stabilität von Verbindungen, aber auch weniger offensichtliche Informationen wie die verbleibende Laufzeit, Rechenleistung und Display-Eigenschaften von mobilen Geräten für eine Beeinflussung von Client-Server Kommunikation genutzt werden können. Es ist beispielsweise denkbar, dass ein Client bei der Kommunikation mit einem Server mitteilt, wie "schnell" seine aktuelle Verbindung ist, wie viele Farben sein Display darstellen kann und welche Auflösung sein Display hat. Der Server könnte anhand dieser Informationen entscheiden, ob er dem Client zu seiner angefragten Information über reinen Text hinaus noch Bilder oder Videos schicken kann. Dies ist nur eines von vielen denkbaren Szenarien. Ein Framework kann in diesen

Fällen natürlich nicht die Entscheidungen für anwendungsspezifische Informationen und Inhalte treffen, jedoch könnte ein Framework von konkreten Technologien und technischen Informationen abstrahieren und diese auch akkumulieren.

Client-Architektur

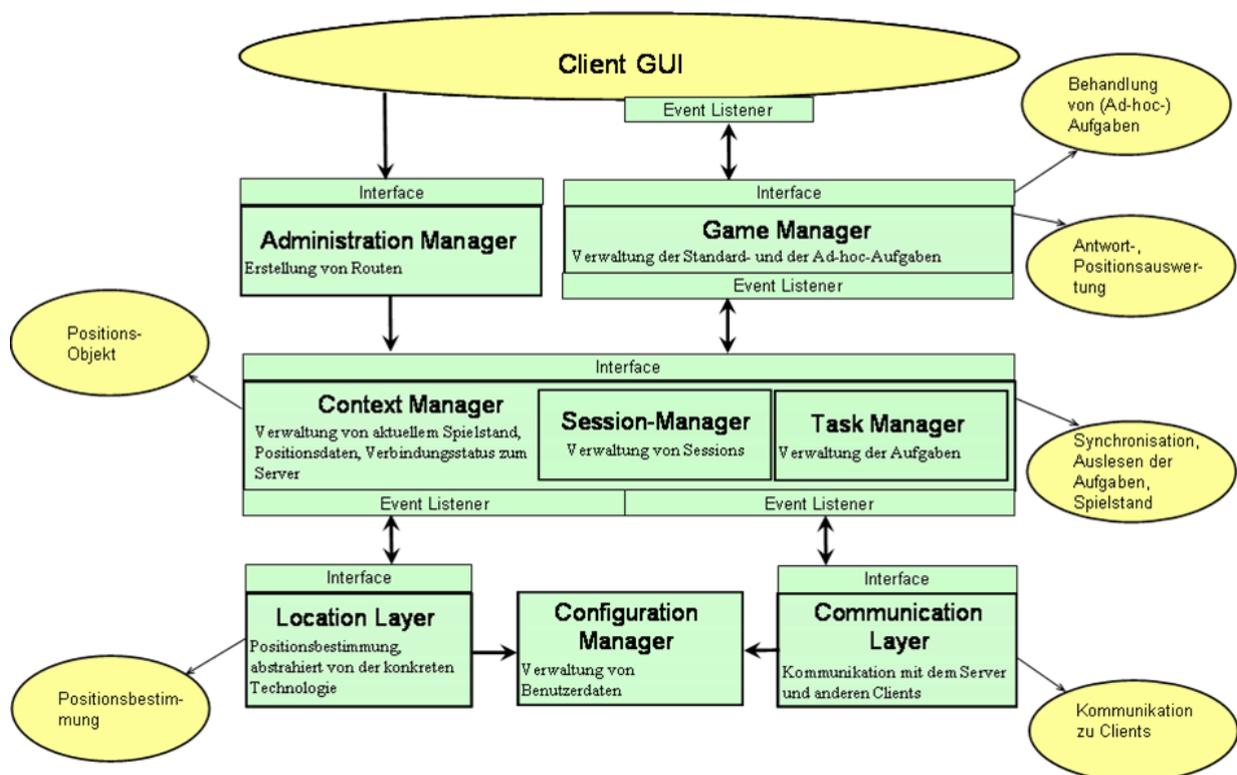


Abbildung 4.1: Pervasive Gaming Framework (Client Architektur)

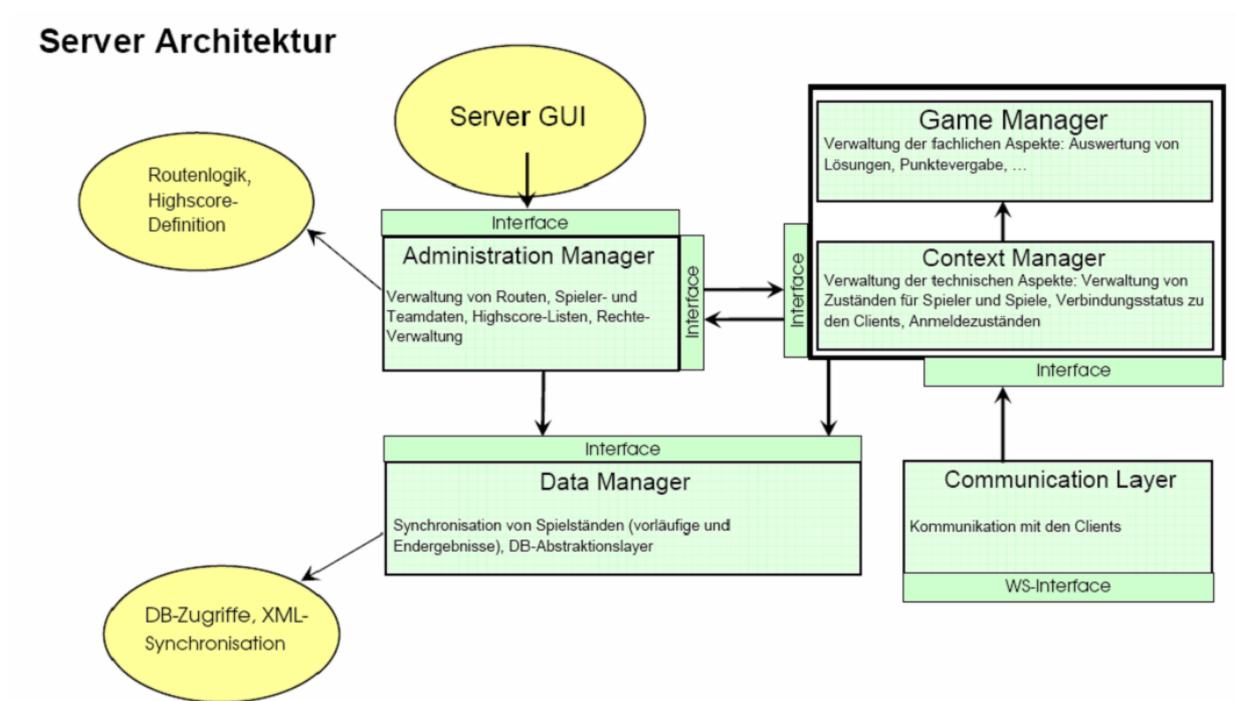


Abbildung 4.2: Pervasive Gaming Framework (Server Architektur)

Kapitel 5

Fazit

Context-Awareness wird zunehmend relevant. In Kapitel 4 wurde gezeigt, dass das Thema "Context-Aware Mobile Computing" viele spannende Aspekte für eine Masterarbeit bietet. Die Entwicklung eines Context-Frameworks für mobile Umgebungen bietet sich aufgrund der Erfahrungen aus dem AW2 Projekt an.

Risiken Wie schon in Kapitel 4 beschrieben ist Context-Repräsentation nicht trivial. Im Rahmen der Masterarbeit wird sich dieser Themenbereich jedoch nicht völlig umgehen lassen und stellt daher sowohl eine Herausforderung, als auch ein Risiko dar.

Ein generelles Problem von Context-Awareness ist, dass das Testen unter Umständen extrem aufwändig wird. Für "normale" Anwendungen, die ihr Verhalten nicht aufgrund von Context-Informationen verändern, gibt es anerkannte Verfahren, um diese effektiv testen zu können. Durch Context-Awareness erhöht sich die Anzahl der potentiell zu testenden Szenarien unter Umständen erheblich, da die regulären Testfälle theoretisch unter allen möglichen Kombinationen von relevanten Context-Informationen überprüft werden müssen. Dies stellt ein Risiko im Rahmen der Masterarbeit und für Context-Awareness im Allgemeinen dar.

Darüber hinaus besteht bei Context-Awareness immer die Gefahr der Anwenderverwirrung durch aus Anwendersicht nicht-deterministisches Verhalten [Barkhuus und Dey, 2003]. Daher darf beim Arbeiten im Umfeld von Context-Awareness nie vergessen werden, dass Context-Awareness die Anwender nicht verwirren, bevormunden oder in irgend einer Form stören, sondern unterstützen soll.

Literaturverzeichnis

- [Ganek und Corbi, 2003] A.G. Ganek, T.A. Corbi: **The dawning of the autonomic computing era**, IBM Systems Journal, 42(1), (2003).
- [Schmidt et al., 1998] A. Schmidt, M. Beigl, H.-W. Gellersen: **There is more to Context than Location**, IEEE (1998).
- [Hofer et al., 2003] T. Hofer et al.: **Context-Awareness on Mobile Devices the Hydrogen Approach**, IEEE (2003).
- [Yan und Sere, 2004] L. Yan, K. Sere: **A Formalism for Context-Aware Mobile Computing**, IEEE (2004).
- [Barkhuus und Dey, 2003] L. Barkhuus, A. Dey: **Is Context-Aware Computing Taking Control Away from the User?**, Proceedings of UbiComp (2003).
- [Chen und Kotz, 2000] G. Chen, D. Kotz: **A Survey of Context-Aware Mobile Computing Research**, IEEE (2000).
- [Chen et al., 2004] G. Chen et al.: **Design and Implementation of a Large-Scale Context Fusion Network**, IEEE (2004).
- [Tan et al., 2003] C.-P. Tan et al.: **Context-aware Service Protocol**, IEEE (2003).
- [BSI, 2006] Bundesamt für Sicherheit in der Informationstechnik: **Pervasive Computing: Entwicklungen und Auswirkungen**, BSI (2006). - ISBN 3-922746-75-6
- [IBM, 2001] ibm.com: **Autonomic Computing**, (2001). Im Internet zu finden unter http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf (Februar 2007)
- [UbiComp] UbiComp @ Informatik.HAW Hamburg. Im Internet zu finden unter <http://users.informatik.haw-hamburg.de/ubicomp/projects.html> (Februar 2007)