



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Seminararbeit

Jaroslav Urich

Context-Aware Services:  
Multimedia-Unterstützung im Flugzeug

Jaroslav Urich  
Context-Aware Services: Multimedia-Unterstützung  
im Flugzeug

Seminararbeit im Rahmen der Veranstaltung Anwendungen 1  
im Studiengang Informatik (Master of Science)  
am Studiendepartment Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

bei Prof. Dr. Kai von Luck

Abgegeben am 26. Juli 2007

**Jaroslav Urich**

**Thema der Seminararbeit**

Context-Aware Services: Multimedia-Unterstützung im Flugzeug

**Stichworte**

Context, Context-Awareness, Context-Aware Services, Service Discovery, Flughafen, Flugzeugkabine

**Zusammenfassung**

Diese Ausarbeitung beschäftigt sich mit Context-Aware Services. Anhand eines Anwendungsbeispiels werden Grundkomponenten eines Context-Aware Systems identifiziert. In diesem Kontext werden die wichtigsten Definitionen erläutert und einige relevante Technologien vorgestellt.

Die Ergebnisse dieser Ausarbeitung dienen als Grundlage für das im nächsten Semester anstehende Projekt.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>5</b>
1.1	Motivation . . . . .	5
1.2	Zielsetzung . . . . .	6
1.3	Gliederung . . . . .	6
<b>2</b>	<b>Analyse</b>	<b>7</b>
2.1	Beispielszenario . . . . .	7
2.2	Identifikation der Kernkomponenten . . . . .	8
2.2.1	Beurteilung des aktuellen Zustandes . . . . .	8
2.2.2	Auffindung / Erkennung von Diensten . . . . .	9
2.2.3	Inanspruchnahme von Diensten . . . . .	9
2.3	Context-Awareness . . . . .	10
2.3.1	Context . . . . .	10
2.3.2	Context-Awareness . . . . .	11
2.4	Service Discovery . . . . .	11
<b>3</b>	<b>Fazit und Ausblick</b>	<b>14</b>
3.1	Bewertung und Zusammenfassung . . . . .	14
3.2	Ausblick auf das Projekt . . . . .	14
	<b>Literaturverzeichnis</b>	<b>16</b>

# 1 Einleitung

Mobile Geräte werden immer kleiner und leistungsfähiger. Sie sind in der Lage unterschiedliche Aufgaben zu erfüllen, für die früher verschiedene Geräte verwendet werden mussten. So kann ein modernes mobiles Gerät nicht nur die Funktion eines Handys übernehmen (wie z.B. Telefonieren oder Versenden von Kurznachrichten wie SMS), sondern bietet es unter anderem die Funktionalität eines kleinen PCs (z.B. Erstellung und Bearbeitung von Dokumenten oder Abspielen eines Filmes). Die meisten Geräte verfügen in der Regel über eine hohe Kommunikationsfähigkeit (z.B. über GPRS, Bluetooth oder WLAN). Dies ermöglicht eine wachsende Anzahl mobiler Benutzer neue Anwendungen in Anspruch zu nehmen.

Die heutigen mobilen Geräte sind relativ leistungsfähig, so dass sie viele Aufgaben ohne Hilfe weiterer Geräte bewältigen können. Dennoch gibt es viele Situationen, in denen der Benutzer auf die Hilfe weiterer Geräte angewiesen ist (z.B. Drucken eines Geschäftsdokumentes). Ein Zusammenspiel unterschiedlicher Geräte ohne aufwändiges Anschließen und Konfigurieren wäre wünschenswert.

In den folgenden Unterkapiteln werden die Motivation (1.1) und die Zielsetzung (1.2) dieser Ausarbeitung dargelegt. Das Unterkapitel 1.3 stellt eine kurze Übersicht der gesamten Ausarbeitung dar.

## 1.1 Motivation

Bei der Entwicklung immer neuer Geräte für die Realisierung bestimmter Aufgaben wird erwartet, dass diese als Hilfsgeräte für die Lösung komplexerer Aufgaben eingesetzt werden können. Das Zusammenspiel verschiedener Geräten ist häufig mit relativ komplexen und aufwändigen Konfigurationen verbunden. Eine automatische Erkennung von Geräten und deren Zusammenarbeit wäre hilfreich.

Die Motivation dieser Ausarbeitung besteht in der Konzeption eines Systems, das ermöglicht mehrere Geräte (bzw. Dienste) für eine bestimmte Aufgabe in Anspruch zu nehmen,

ohne dass der Benutzer viele Konfigurationen vornehmen muss, bzw. dass dies automatisch geschieht.

## 1.2 Zielsetzung

Das Ziel dieser Arbeit ist die Analyse der Funktionalität von Context-Aware Services am Beispiel eines Anwendungsfalles und die Identifikation der einzelnen Komponenten für die Entwicklung. Dadurch soll ein Überblick über die konzeptionellen und technischen Anforderungen geschaffen werden.

## 1.3 Gliederung

Im Kapitel 2 (Analyse) wird ein Beispielszenario für das im Unterkapitel 1.1 besprochene System skizziert. Anhand dieses werden die Kernkomponenten identifiziert und einige Technologien für die Umsetzung vorgestellt.

Im 3. Kapitel (Fazit und Ausblick) werden die Analyseergebnisse bewertet und es wird ein Ausblick auf das im nächsten Semester geplante Projekt gegeben.

## 2 Analyse

Dieses Kapitel beschäftigt sich mit der Analyse eines Systems, das ein automatisches Zusammenspiel von verschiedenen Diensten (bzw. Geräten) ermöglicht (vgl. Unterkapitel 1.1). Um konkrete Anhaltspunkte für die Identifikation der Kernkomponenten eines solchen Systems zu geben, wird ein Anwendungsbeispiel genommen.

### 2.1 Beispielszenario

Die Metapher *Flughafen* bietet sich als exemplarisches Anwendungsumfeld besonders gut an: *«Während einer Flugreise besucht ein Reisender drei Aufenthaltsorte: der Start- und der Ziel-Flughafen und die Flugzeug-Kabine. An diesen Orten können unterschiedliche Dienste zur Verfügung stehen. Diese müssen von dem mobilen Gerät des Fluggastes erkannt und falls gewünscht in Anspruch genommen werden können.»*.

Im Folgenden wird das Beispielszenario konkreter beschrieben.

Ein Fluggast befindet sich im Flughafenterminal. Er hat bereits den Check-In durchgeführt und wartet darauf ins Flugzeug einzusteigen. Da die meisten Reisenden schon 60 bis 90 Minuten vor dem Abflug das Flughafengelände betreten, sind die Fluggäste gezwungen eine relativ große Zeit zu warten.

Der Fluggast hat ein mobiles Gerät bei sich (wie z.B. ein PDA<sup>1</sup>), das ihm ermöglicht diese Wartezeit zu verkürzen, indem er sich z.B. einen Film ansehen kann. Der PDA registriert einen Dienst, der gegen Entgelt Filme zum Download anbietet. Der Reisende nimmt das Angebot an und lädt einen Film herunter.

---

<sup>1</sup>Ein Personal Digital Assistant (englisch für *persönlicher digitaler Assistent*) ist ein kleiner tragbarer Computer mit eigener Stromversorgung, der neben vielen anderen Programmen hauptsächlich für die persönliche Kalender-, Adress- und Aufgabenverwaltung benutzt wird (Quelle: Wikipedia)

Der Fluggast startet den Film. Nach einiger Zeit wird der Reisende aufgefordert sich in das Flugzeug zu begeben. Er unterbricht den Film und geht zum Flugzeug. Sein PDA merkt sich die Stelle, wo der Film unterbrochen wurde.

Während des Fluges entschließt sich der Reisende den Film weiter zu sehen. Er schaltet sein PDA wieder ein und startet das Programm, welches Filme abspielen kann. Nach dem Programmstart ist eine Meldung zu sehen, die besagt, dass es möglich ist den Bildschirm des Flugzeuges zu benutzen. Der Reisende stimmt zu, startet die Filmwiedergabe und sieht den Film auf diesem Bildschirm weiter.

In dem folgenden Unterkapiteln werden die Bestandteile des Systems identifiziert und anhand des Beispielsszenarios veranschaulicht.

## 2.2 Identifikation der Kernkomponenten

Im Rahmen dieser Ausarbeitung wurden für dieses Beispielsszenario<sup>2</sup> folgende Kernkomponenten identifiziert:

- Beurteilung des aktuellen Zustandes
- Auffindung / Erkennung von Diensten
- Inanspruchnahme von Diensten

Im Folgenden werden die einzelnen Komponenten näher erläutert.

### 2.2.1 Beurteilung des aktuellen Zustandes

Die Beurteilung des aktuellen Zustandes ist eine wesentliche Aufgabe des Gesamtsystems. Das System soll in der Lage sein seinen momentanen Zustand zu analysieren und daraus mögliche Folgerungen zu ziehen. So hat z.B der PDA, wie im Beispielszenario beschrieben<sup>3</sup>, seinen Zustand und seine Möglichkeiten analysiert und ist zum Ergebniss gekommen nach einem besseren Darstellungs-Gerät in der Umgebung zu suchen, um dem Fluggast ein besseres Filmerlebnis zu ermöglichen.

---

<sup>2</sup>siehe Unterkapitel [3.2](#)

<sup>3</sup>vgl. mit Unterkapitel [3.2](#)

### 2.2.2 Auffindung / Erkennung von Diensten

In dem Beispielszenario aus dem Unterkapitel 3.2 hat der PDA erkannt, dass in der Umgebung sich ein größerer Bildschirm befindet und dessen Dienste in Anspruch genommen werden können. Dies war das Ergebnis seiner Suche nach besseren Diensten. Der PDA hat die Umgebung untersucht und die verfügbaren Dienste lokalisiert. Der PDA hat berücksichtigt, dass nur der Bildschirm, der dem Sitz des Fluggastes zugeordnet ist, angesprochen werden darf und beispielsweise nicht der seines Nachbarn.

### 2.2.3 Inanspruchnahme von Diensten

Diese Komponente beschäftigt sich mit der Benutzung eines Dienstes. So wird im Anwendungsbeispiel<sup>4</sup> der Film auf dem Bildschirm der Flugzeug-Kabine dargestellt. Dies ist keineswegs eine triviale Aufgabe, da hierfür die Geräte eine 'gemeinsame Sprache' verstehen müssen. Das bedeutet, dass die Dienste über eine standardisierte Schnittstelle ansprechbar sein müssen, damit auch unterschiedliche Geräte (bzw. Dienste) sie benutzen können.

An dieser Stelle soll noch erwähnt werden, dass bei der Dienst-Nutzung sowohl die Qualität des Dienstes als auch der aktuelle Zustand des Dienst-Anbieters<sup>5</sup> berücksichtigt müssen. So kann es beispielsweise entschieden werden, ob dieser Dienst für die Aufgabe geeignet ist<sup>6</sup> oder auf welche Weise der Dienst genutzt wird<sup>7</sup>.

Die oben beschriebenen Anforderungen und damit verbundenen Probleme werden im Umfeld des Context-Awareness behandelt. Die folgenden Unterkapitel erklären den Begriff 'Context-Awareness' und geben eine kurze Übersicht über die Technologien, die für die Umsetzung solcher Systeme relevant sind.

---

<sup>4</sup>siehe Unterkapitel 3.2

<sup>5</sup>hier ist das Gesamtsystem (bzw. das Gerät) gemeint ist, das den Dienst anbietet

<sup>6</sup>an dieser Stelle ist die generelle Qualität des Dienstes gemeint

<sup>7</sup>Oft gibt es mehrere Möglichkeiten einen Dienst in Anspruch zu nehmen. So kann beispielsweise ein Film komplett auf ein anderes Gerät kopiert und von da aus abgespielt werden oder der Film kann auf vom PDA auf das Ausgabegerät gestreamt werden (vgl. Unterkapitel 3.2).

## 2.3 Context-Awareness

Dieses Kapitel beschäftigt sich mit der Begriffsklärung des Context-Awareness und erläutert die Zusammenhänge, die damit verbunden sind.

Bevor der Begriff 'Context-Awareness' erläutert wird, soll zuerst geklärt werden was 'Context' bedeutet.

### 2.3.1 Context

In der Literatur findet sich eine Vielzahl von Definitionen, die den Begriff 'Context'<sup>8</sup> erklären. Die meisten sind relativ umfangreich und haben eher einen Aufzählungscharakter. In dieser Ausarbeitung wird eine Definition verwendet, die relativ allgemein ist und somit alle andere Definitionen einschließt.

'Context' wird definiert als «jede Information, die benutzt werden kann, um die Situation einer *Entity* zu charakterisieren». Eine *Entity* ist in dem Zusammenhang «eine Person, ein Ort oder ein Objekt, welche als relevant für die Interaktion zwischen dem Anwender und der Applikation (einschließlich des Anwenders und der Applikation selbst) angesehen werden kann» (Dey, 2001).

Als eine der schwierigsten Aufgaben in diesem Zusammenhang erweist sich die Modellierung von *Context*. Die Lösung dieser Aufgabe besteht in der Beantwortung der folgenden Fragen:

- Welche Informationen sind für die jeweilige Interaktion relevant?
- Wie können diese Informationen gewonnen werden?
- Wie werden diese Informationen zu einem *Context* zusammengefasst, verwaltet und gespeichert?

Eine Erläuterung der Lösungen für diese Probleme würde den Rahmen dieser Ausarbeitung sprengen. Es wird schließlich auf die folgende Literatur verwiesen:

- In (PreBur, 2003) wird ein Konzept des *Activity-Centric Context* beschrieben, in dem der *Context* erst zu existieren beginnt, wenn eine Aktivität ausgeführt wird.

---

<sup>8</sup>Hier wird der englische Begriff 'Context' im Zusammenhang mit 'Context-Awareness' verwendet, damit es keine Verwechslung mit dem deutschen Wort 'Kontext' entsteht, obwohl eine gewisse Parallele in der Bedeutungen dieser Wörter erkennbar ist.

- Das *Schichtenmodell* für die *Context*-Modellierung wird in (Aust, 2004) erläutert.
- *Ontologie-basierte Context-Modelle* werden in (BuTM, 2005) behandelt.

### 2.3.2 Context-Awareness

Unter '*Context-Awareness*' wird die Fähigkeit (einer Anwendung oder eines Dienstes) verstanden, *Context* wahrzunehmen und einzusetzen<sup>9</sup>.

Dadurch kann die Anwendung (bzw. der Dienst) die Erwartung des Anwenders in größerem Maße erfüllen, als wenn die *Context*-Informationen nicht berücksichtigt werden. So kann das System besser auf die Wünsche des Anwenders unter dem Bezug auf die aktuelle Situation eingehen (bzw. solche erfüllen).

Unter Berücksichtigung der oberen Definition ist ein '*Context-Aware Service*' ein Dienst, der Informationen über den *Context* benutzt, um sein Verhalten an die jeweilige Situation anzupassen.

Für die *Context-Aware*-Systeme sind nicht nur der *Context* und die damit verbundenen Entscheidungen relevant, sondern auch die Fähigkeit, andere Dienste zu lokalisieren und deren Funktionalität in Anspruch zu nehmen, spielt eine nicht weniger wesentliche Rolle. Das Auffinden bzw. die Erkennung der in der Umgebung sich befindenen Dienste und deren Nutzung werden im folgenden Unterkapitel kurz erläutert.

## 2.4 Service Discovery

Dieses Unterkapitel beschäftigt sich mit der Problematik der Dienstsuche und deren Nutzung und stellt einige Lösungsansätze kurz vor.

Zuerst soll geklärt werden, was unter dieser Problematik zu verstehen ist.

Mit der Lokalisierung, Suche und Nutzung eines Dienstes beschäftigt sich ein Teilgebiet der Informatik, das unter dem Namen *Service Discovery* bekannt ist. Es untersucht, wie die Dienste (*Services*) bekannt gemacht werden können, und relativ schnell und effizient von einem anderen Dienst (bzw. von einer Applikation) gefunden werden können, um ihre Funktionalitäten in Anspruch zu nehmen.

---

<sup>9</sup>vgl (Dey, 2001) und (DeyAbowd, 2001)

Es gibt bereits eine Reihe von Lösungsansätzen, deren Umsetzungen schon heute relativ gute Ergebnisse liefern. Im Folgenden werden einige davon kurz dargestellt<sup>10</sup>.

### **Jini**

Jini ist ein Java-Framework, das von Sun Microsystems entwickelt wurde. Es ermöglicht eine relativ einfache und komfortable Programmierung von verteilten Anwendungen. Die Architektur von Jini ist dienstorientiert, d.h. die Anwendungen (bzw. Systeme), die mit Jini gebaut werden, basieren auf der Interaktion (bzw. Zusammenspiel) von den zur Verfügung gestellten Diensten<sup>11</sup>. So kann beispielsweise ein Drucker seinen Dienst bei einem so genannten Lookup-Service anmelden, wo er von den anderen Diensten (bzw. Applikationen) gefunden und in Anspruch genommen werden kann.

Die Interaktionen zwischen den Diensten verlaufen über das Netzwerk. Jini setzt voraus, dass ein Netz schon aufgebaut ist. Eine weitere Einschränkung des Frameworks ist, dass alle Geräte, die Jini einsetzen, eine Java Virtual Machine besitzen müssen.

Für weitere Informationen über dieses Framework wird auf <http://www.sun.com/software/jini/> verwiesen.

### **UPnP**

Universal Plug and Play (UPnP) wurde ursprünglich von der Firma Microsoft eingeführt. UPnP basiert auf einer Reihe von standardisierten Netzwerkprotokollen und Datenformaten. Dies ermöglicht eine herstellerübergreifende Ansteuerung von Geräten (bzw. Diensten)<sup>12</sup>.

Im Unterschied zu Jini existiert in UPnP keine gemeinsame Stelle, wo alle verfügbaren Dienste registriert sind. Sobald ein Dienst gestartet wird, meldet er sich per Multicast bei allen Kontrollpunkten im Netz an. Es ist ebenso möglich nach Diensten im Netzwerk zu suchen. UPnP ist in der Lage IP-Adressen automatisch zu vergeben, falls die Geräte noch keine IP-Adresse besitzen oder sich kein DHCP-Server in der Nähe befindet.

Eine genauere Beschreibung der Arbeitsweise von UPnP kann unter <http://www.upnp.org/> eingesehen werden.

---

<sup>10</sup>Eine gründlichere Behandlung dieser Problemstellung würde den Rahmen dieser Ausarbeitung überschreiten.

<sup>11</sup>vgl. (OaksWong, 2000)

<sup>12</sup>vgl. (TaiJerWeast, 2003)

## Bonjour

Bonjour ist eine Implementierung des Zeroconf-Systems<sup>13</sup> von Apple. Es beschreibt eine Technik, mit der eine automatische Erkennung von Diensten in IP-Netzen möglich ist.

Bonjour setzt nicht voraus, dass das Netzwerk vollständig aufgebaut ist. Es ist in der Lage Geräte (bzw. Dienste) ohne Hilfe von DNS- oder DHCP-Server automatisch zu erkennen. Bonjour stellt u.a. Mittel zur Verfügung, um Dienste zu lokalisieren oder solche bekannt zu machen.

Weitere Informationen zu Bonjour findet sich unter <http://www.apple.com/macosx/features/bonjour/>.

---

<sup>13</sup>Zeroconf oder Zero Configuration Network ist eine Sammlung von Techniken, mit deren Hilfe ein IP-Netzwerk automatisch aufgebaut werden kann, ohne jegliche Konfigurationen oder Verwendung von speziellen Servern. (CheStein, 2006)

## **3 Fazit und Ausblick**

In diesem Kapitel werden zunächst die Ergebnisse dieser Ausarbeitung präsentiert (siehe Unterkapitel ). Abschließend wird dann im Unterkapitel ein Ausblick auf das anstehende Projekt gegeben.

### **3.1 Bewertung und Zusammenfassung**

Das Ziel dieser Ausarbeitung war, anhand eines Beispielszenarios die wichtigsten Aspekte eines Context-Aware Service (bzw. eines Context-Aware-Systems) zu identifizieren. Hierbei wurden die wesentlichen Begriffe in diesem Kontext erläutert und ein kleiner Ausblick auf die heutige Technologie gegeben, mit deren Hilfe die Realisierung eines solchen Systems vorstellbar wäre.

Das Anwendungsbeispiel spielte hier eine wesentliche Rolle. Es diente nicht nur der Identifikation der Kernkomponenten eines Context-Aware-Systems, sondern bietet unter anderem eine gute Möglichkeit für die Realisierung eines solchen Systems, um die gravierendsten Probleme bei der Entwicklung von Context-Aware Services zu analysieren.

Das folgende Unterkapitel gibt einen kurzen Ausblick auf das weitere Geschehen in diesem Umfeld.

### **3.2 Ausblick auf das Projekt**

Im Rahmen des Projektes, das im nächsten Semester durchgeführt wird, sollte das Beispielszenario (siehe Unterkapitel ) für Forschungszwecke umgesetzt werden. Hierbei sollen die folgenden Punkte erarbeitet werden.

- *Definition von System-Anforderungen*  
Anhand des Beispielszenarios sollen die konkreten Anforderungen an das System gestellt werden. Hierbei sollen die allgemeinen Anforderungen an ein Context-Aware System von den Anforderungen an das konkrete System unterschieden werden.
- *Design und Architektur des System*  
Mit Rücksicht auf die gewonnenen Anforderungen soll die Architektur des Systems konzipiert werden.
- *Analyse der heutigen Technologien*  
Bevor die modellierte Architektur implementiert wird, sollen zunächst die heute verfügbaren Technologien analysiert werden, die für die Realisierung in Frage kommen könnten. Hierbei soll untersucht werden, ob überhaupt und mit welchem Aufwand eine Realisierung des Systems möglich wäre.
- *Entwicklung eines Prototyps*  
Die erarbeitete Architektur soll mit den gewählten Technologien prototypisch implementiert werden. Der Prototyp soll zunächst nicht den vollen Umfang der Funktionalitäten des Systems beinhalten. Die Sicherheitsaspekte werden zuerst nicht berücksichtigt, und der Prototyp soll in der Anfangsphase nur wenige Context-Aware Services beinhalten.

Mit dieser Ausarbeitung wurden zunächst die Grundsteine für eine erfolgreiche Durchführung des anstehenden Projektes gelegt. In welchem Maße die oben vorgestellten Ziele auch tatsächlich erreicht werden, bleibt noch abzuwarten.

# Literaturverzeichnis

- [Aust 2004] AUSTALLER, G. (Hrsg.): *Web Services als Bausteine für kontextabhängige Anwendungen*. 2004
- [BuTM 2005] RENATO BULCÃO NETO, Maria da Graça Campos P. (Hrsg.): *A Semantic Web-Based Infrastructure Supporting Context-Aware Applications*. 2005
- [CheStein 2006] CHESHIRE, Stuart ; STEINBERG, Daniel H.: *Zero Configuration Networking*. 1. Auflage. O'Reilly Media, Inc., 2006. – ISBN 0-596-10100-7
- [Dey 2001] DEY, Anind K. (Hrsg.): *Understanding and Using Context*. 2001. – URL <http://www.cc.gatech.edu/fce/ctk/pubs/PeTe5-1.pdf>
- [DeyAbowd 2001] DEY, Anind K. (Hrsg.) ; ABOWD, Gregory D. (Hrsg.): *Towards a Better Understanding of Context and Context-Awareness*. 2001. – URL <http://www.cc.gatech.edu/fce/ctk/pubs/PeTe5-1.pdf>
- [OaksWong 2000] OAKS, Scott ; WONG, Henry: *Jini in a Nutshell*. 1. Auflage. O'Reilly Media, Inc., 2000. – ISBN 978-3-89721-194-0
- [PreBur 2003] PREKOP, Paul (Hrsg.) ; BURNETT, Mark (Hrsg.): *Activities, Context and Ubiquitous Computing*. 2003. – URL <http://arxiv.org/ftp/cs/papers/0209/0209021.pdf>
- [TaiJerWeast 2003] TAI, Charlie ; JERONIMO, Michael ; WEAST, Jack: *UpnP Design by Example*. Intel Press.,US, 2003. – ISBN 978-0-9717-861-10