

# Interaktion als Dienst in Mundo

Elmar Braun, Erwin Aitenbichler

Fachbereich Informatik, Fachgebiet Telekooperation  
Technische Universität Darmstadt  
Alexanderstraße 6  
64283 Darmstadt  
{elmar, erwin}@tk.informatik.tu-darmstadt.de

**Abstract:** Beim Ubiquitous Computing sollen unzählige verschiedene Geräte in spontaner Kooperation Dienste auf intelligente und benutzerfreundliche Weise erbringen. Ein Dienst könnte z.B. je nach Situation des Benutzers mal ein Display und mal Sprache zur Interaktion nutzen. Solch intelligentes Verhalten sollte automatisch bereitgestellt werden, anstatt jeden Entwickler von neuem mit der Komplexität der Interaktionsinfrastruktur zu konfrontieren. Wir zeigen, wie eine Middleware für Interaktion in einer solchen Umgebung aussehen kann, und wie man mit einfachen Mitteln eine existierende Webapplikation ubiquitär verfügbar machen kann.

## 1 Einführung

Ubiquitous Computing verspricht mühelose und ungezwungene Benutzung von Computern. Überall in die Umgebung integrierte Ein- und Ausgabegeräte erlauben Interaktion jederzeit, anstatt sie auf den Schreibtisch oder ein kleines Mobilgerät zu beschränken. Allgegenwärtige Sensoren nehmen die Bedürfnisse der Benutzer wahr und erlauben intelligente Anpassung daran. Aber wie setzt man diese heterogene und komplexe Umgebung mit ihren zahllosen Geräten für die Benutzerinteraktion ein? Welche Geräte befinden sich gerade in dessen Reichweite? Wie wählt man das für eine Interaktion am besten geeignete Gerät? Wie soll man ein Benutzerinterface entwickeln, wenn von mal zu mal über ganz andersartige Geräte (etwa Touchscreen, Sprachsteuerung, etc.) interagiert wird?

Man kann nicht jedem Entwickler einer interaktiven Applikation aufbürden, jedes Gerät oder gar jede mögliche Gerätekombination zu handhaben. Angesichts der Fülle von Geräten wäre das zu aufwendig. In MUNDO, unserem in Kapitel 2 kurz besprochenen Projekt zu Ubiquitous Computing, entwickeln wir Middleware und Abstraktionen, die Heterogenität und Komplexität solcher Umgebungen zu beherrschen helfen. Dabei ähneln die hier behandelten Probleme bei der Interaktion denen, die für Vernetzung und Kommunikation unter Geräten bereits gelöst wurden. Darauf basierend wird in Kapitel 3 beschrieben, wie unsere Middleware Dienste zur Interaktion mit dem Benutzer bereitstellen kann. In Kapitel 4 wird dann erörtert, wie eine existierende Webapplikation mit wenig Aufwand ubiquitär verfügbar gemacht werden kann.

## 2 Mundo und MundoCore

MUNDO ist ein Konzept für Ubiquitous Computing, das auf der vorhandenen ubiquitären Netzwerkstruktur des Internets aufbaut [HAA<sup>+</sup>02]. Dabei handelt es sich um eine Menge von Diensten, Protokollen, Terminal-Konzepten, Abrechnungs- und Kooperationsmodellen, die inkrementell eingeführt werden können. Das Akronym MUNDO spielt auf das spanische Wort Mundo an, das Welt bedeutet. Es steht aber auch als Abkürzung für "Mobile and Ubiquitous Networking via Distributed Overlay cells".

MundoCore stellt die Kommunikations-Kernkomponente von MUNDO dar. Es implementiert die für Dienste notwendigen Kommunikationsschnittstellen und stellt einheitliche Abstraktions- und Programmiermodelle auf allen unterstützten Plattformen zur Verfügung. Aus unseren Erfahrungen bei der Implementierung von MUNDO und MundoCore haben wir unter anderem die folgenden Anforderungen an eine Kommunikations-Middleware für Ubiquitous Computing identifiziert:

**Hohe Mobilität.** Vertikale und horizontale Zellenwechsel müssen effizient unterstützt werden, ohne bestehende Verbindungen abubrechen. Auf heutigen Systemen führt ein Zellenwechsel typischerweise zum Abbruch aller offenen TCP-Verbindungen. Die entsprechende Fehlerbehandlung bleibt Aufgabe der jeweiligen Anwendungen. Die grundlegende Kommunikations-Abstraktion in MundoCore ist Publish/Subscribe.

**Heterogenität.** Aus der Entwicklung der letzten Jahrzehnte erkennen wir, daß sich in absehbarer Zeit keine einheitliche Plattform materialisieren wird. Eine Middleware für Ubiquitous Computing muß die Kommunikation angefangen von kleinen Sensoren bis hin zu großen Server-Clustern ermöglichen.

MundoCore existiert gegenwärtig als C++ und als Java-Version. Die C++ Version ist auf Desktop/Server Windows, Windows CE, Mac OS/X, Linux und uClinux verfügbar. Eine minimale Installation, bestehend aus einem textbasierten Chat-Dienst, Message-Broker, XML-Parser und IP-Transportdienst, benötigt derzeit 98KB unter Windows CE.

## 3 Interaktion als Dienst der Infrastruktur

Die Portierung eines Programms auf ein Mobilgerät illustriert, welche Probleme mit den Kommunikationsabstraktionen von MundoCore lösbar sind, und wie man ähnliche Konzepte für Interaktion einsetzt. Die Portierung eines Programms von der statischen Umgebung eines Desktops auf ein Mobilgerät führt zu zwei Problemen. Erstens können sich durch Ortswechsel die Netzadressen ändern und offene Verbindungen unterbrochen werden. Zweitens gibt es viele verschiedene Plattformen und Netzwerktechniken für Mobilgeräte (WLAN, GSM, UTMS, etc.). Bei einer Portierung ist man also eventuell gezwungen, die Bindung an das Netzwerk komplett neu zu schreiben.

MundoCore unterstützt die Portierung ohne Änderungen am Netzwerkcode, indem es für eine Vielzahl von Plattformen dieselben Funktionen bereitstellt. Neben den Eigenheiten der Netzwerkanbindung sind aber auch die verschiedenen Interaktionsmöglichkeiten zu

behandeln. So könnte es notwendig sein, ein Desktop-GUI auf ein Mobiltelefon oder auf eine Sprachsteuerung abzubilden. Um den Aufwand für eine manuelle Portierung zu vermeiden sollte man von Interaktionsmodalitäten abstrahieren, so wie MundoCore von Aspekten des unterliegenden Netzwerkes abstrahiert. Es gibt bereits mehrere Ansätze für eine solche geräteunabhängige Beschreibung von Benutzerinterfaces [APB<sup>+</sup>99].

Auch das Problem sich ändernder Netzwerkadressen und unterbrochener Verbindungen tritt in abgewandelter Form auf. Wenn sich ein Benutzer in einer intelligenten Umgebung bewegt, ändert sich die Menge der sich in Reichweite befindlichen Ein- und Ausgabegeräte. Anstatt der Frage, hinter welcher Adresse sich ein bestimmtes Gerät befindet, stellt sich nun die Frage, hinter welchem Gerät sich ein bestimmter Benutzer verbirgt. Applikationen sollten aber nicht auf die Interaktion über z.B. Mobiltelefon beschränkt sein, während der Nutzer direkt vor einem großen Display steht. Aus der Menge der Geräte in Reichweite müssen automatisch die gewählt werden, die das Interface in der höchsten Qualität präsentieren können. Dies kann eine Kombination von mehreren Geräten sein, um z.B. durch Verknüpfung eines Displays aus der Infrastruktur mit einem privaten sprachbasierten Gerät ein multimodales Interface zu erzeugen.

MundoCore bietet einen Publish/Subscribe-Kommunikationsdienst an. Damit sendet man Events über mit Namen versehene Kanäle. Beim Senden wird nicht an eine Netzwerkadresse gesendet, sondern an den Namen eines Kanals. Das Routing von MundoCore schickt das Event dann an alle Clients, die vorher diesen Kanal subskribiert haben. Verbindungsunterbrechungen und sich ändernde Netzwerkadressen werden vom Routing gehandhabt. Sie müssen daher von der Applikation nicht beachtet werden, da sich der Name des Kommunikationskanals nicht ändert.

Eine ähnliche Vorgehensweise kann zur Interaktion mit einem Benutzer über das ihm nächste Gerät verfolgt werden. Anstatt ein konkretes Interface an ein bestimmtes Gerät zu schicken, sollte man eine geräteunabhängige Interfacebeschreibung an den Benutzer adressieren. Aber wer abonniert den entsprechenden Kanal für den Benutzer? Ein Ansatz wäre, alle Geräte, die sich in Reichweite eines Benutzers befinden, automatisch diesen Kanal subskribieren zu lassen. Ohne weitere Koordination würden dann aber alle Geräte in der Umgebung jedes Interface, das an den Benutzer geschickt wird, redundant präsentieren.

In MUNDO gibt es für jeden Benutzer genau einen Agenten, der darauf wartet, daß ihm eine Aufforderung zur Interaktion geschickt wird. Dieser Agent ist auch zuständig für die Buchführung über die in Benutzerreichweite vorhandenen Interaktionsmöglichkeiten und deren Eigenschaften und Status (etwa ob sie bereits belegt sind). Sobald der Agent aufgefordert wird, eine Interaktion mit dem Benutzer durchzuführen, wählt er die geeigneten Geräte aus, und schickt diesen das Benutzerinterface oder (bei mehreren kooperierenden Geräten) Teile davon. Dies ist vergleichbar mit einem Fenstermanager, der versucht, neue Fenster auf einem Desktop sinnvoll zu plazieren. Anstelle des festen 2D-Raumes eines Desktops steht nun der veränderliche und heterogene Raum aller Geräte in Benutzerreichweite zur Verfügung.

## 4 Ubiquitäre Webapplikationen

Umfangreich mit Sensoren und Ein- und Ausgabegeräten ausgerüstete intelligente Umgebungen sind bislang kaum außerhalb der Forschung anzutreffen. Dabei ist die große Altlast nicht an solche Umgebungen angepaßter Programme ein Hemmnis für breitere Akzeptanz. Daher gilt es zu untersuchen, wie man eine Applikation nachträglich anpassen kann, ohne sie komplett auf eine Ubiquitous Computing Middleware portieren zu müssen.

Relativ einfach kann man dies bei Webapplikationen erreichen, die auf einem Server im Netz laufen und durch ein HTML-Frontend bedient werden. In zwei Punkten entsprechen sie bereits dem Konzept aus Kapitel 3. Erstens wird die Präsentation und Interaktion mit dem Benutzer von einem Agenten (dem Browser) übernommen. Zweitens ist HTML zumindest in gewissem Maße bereits geräteunabhängig. Zwar kann man es nicht einfach in ein Sprachinterface übersetzen, aber immerhin kann es mit verschiedenen Hardwareplattformen (sofern dort ein Browser existiert) und ungewöhnlichen Bildschirmgrößen umgehen, auch wenn auf kleinen Displays die Bedienbarkeit meist stark leidet.

In einer Umgebung, in der MUNDO Netzwerkdienste bereitstellt und die Annäherung von Benutzern an Displays erkennt, kann man Webapplikationen mit geringem Aufwand ubiquitär verfügbar machen. Nur Applikationen, die von sich aus Interaktion einleiten (z.B. ein Messagingprogramm, das Benutzer beim Erhalt einer Nachricht informiert), brauchen eine Erweiterung, die entscheidet, wann Interaktion mit dem Nutzer initiiert wird.

Abbildung 1 zeigt, wie eine solche Applikation Interaktion einleitet. Die Assoziation zwischen Geräten und Benutzern wird kontinuierlich von Sensoren erfaßt, welche den Agenten des Benutzers über Geräte in dessen Nähe informieren (1). Unter anderem benutzen wir dafür Infrarot-Tags an Geräten, die eine ID an das Badge eines Benutzers senden, wenn dieser vor einem Gerät steht. Sobald die kontextbewußte Erweiterung der Webapplikation (durch ein Kontext-Event (2) oder aus der Applikationslogik) entscheidet, daß sie mit einem Benutzer interagieren will, benachrichtigt sie dessen Agenten (3), welcher die URL der Applikation an ein Display beim Benutzer schickt (4). Da Browser im allgemeinen keinen Push-Betrieb unterstützen, muß auf dem das Display steuernden Rechner eine Komponente zur Automation (Fernsteuerung) des Browsers laufen, die diesen die empfangene URL abrufen läßt (5). Wir haben für unser Projekt den Internet Explorer automatisiert.

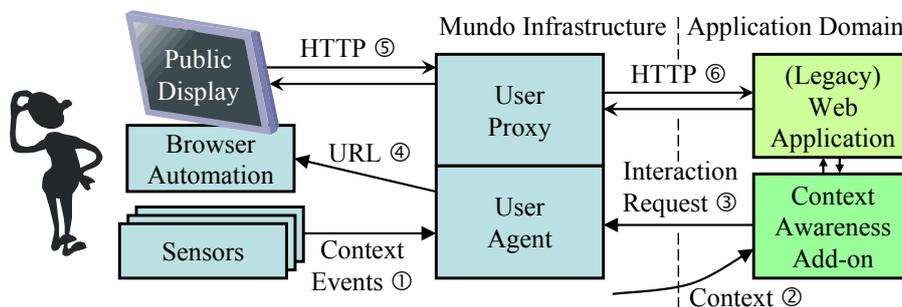


Abbildung 1: Webbasierte Interaktion in MUNDO

In Abbildung 1 läuft die Kommunikation zwischen Browser und Applikationsserver über einen Proxy. Solange nur ein Gerät eine Seite anzeigt, kommt man auch ohne diesen aus, da das Gerät die Seite direkt abrufen kann, wie es in unserer ersten Implementierung auch geschieht. Bei Verbänden mehrerer Geräte, die eine Seite falls angebracht gemeinsam darstellen, muß man das HTML-Dokument in mehrere Teile transkodieren. So könnte man etwa bei Kombination eines Displays ohne Eingabemittel mit einem PDA auf diesem ein Extrakt der Links und interaktiven Elementen der Seite anzeigen. In umgekehrter Richtung muß der Proxy die verteilten Eingaben synchronisieren. Wird auf dem PDA ein Link aktiviert, muß der Benutzeragent auch auf dem Display die neue Seite öffnen.

Bisher wurde nur der Sonderfall diskutiert, daß Applikationen die Interaktion selbst initiieren. Eine Anwendung dafür ist Instant Messaging. Die Implementierung sendet ein Webinterface an das aktuelle Gerät des Benutzers, sobald eine Nachricht für ihn eintrifft. Damit erreicht man geräteunabhängiges Messaging, das den Benutzer auch dann erreichen kann, wenn dieser seinen persönlichen Rechner verläßt. Dabei muß auf dem anzeigenden Rechner kein gesondertes Messagingprogramm installiert sein.

Was ist aber mit der Mehrzahl der Applikationen, die normalerweise vom Benutzer und nicht automatisch gestartet werden? Wenn unser System erkennt, daß sich ein Benutzer mit einem Gerät assoziiert, während keine Anwendung mit ihm interagiert, kann man als eine Art Desktop eine Seite mit Links zu den Anwendungen und Dokumenten des Benutzers zeigen. Von dort erreichte Webseiten und -applikationen können dann die Funktionen unseres Systems nutzen, wie z.B. die auf mehrere Geräte verteilte Präsentation oder das Verfolgen eines mobilen Benutzers über Geräte hinweg. Enthält diese Seite auch Links zu persönlichen Dateien, so kann man die durch Vernetzung mögliche allgegenwärtige Verfügbarkeit von Dateien auch auf eine benutzerfreundliche Weise an die Nutzer weitergeben. Der Vorgang des Startens einer Präsentation auf einem fremden Rechner reduziert sich so etwa auf das Zugehen auf den Rechner und das Aufrufen des Links.

## **5 Verwandte Arbeiten und Ausblick**

Die Idee der Applikationen, die ihren mobilen Benutzern folgen und sich dynamisch auf Geräten in deren Umgebung präsentieren, ist unter dem Namen Teleporting bekannt [BRH94]. Dabei wird das Interface durch Umleitung der Graphikausgabe auf einem anderen Gerät unverändert (wie der Name Teleporting ja suggeriert) reproduziert. Das ist aber nur innerhalb einer Modalität möglich; die Übersetzung eines graphischen Interfaces in Sprache ist nicht möglich. Beachtenswert ist trotzdem, daß ebenfalls eine bereits existierende Applikation ohne Veränderung mobil verfügbar gemacht wurde.

Auch Räume für computergestützte Teamarbeit sind häufig reichhaltig mit verschiedenen Ein- und Ausgabegeräten ausgerüstet. Dabei wird aber meist von einer vorgegebenen statischen Rauminfrastruktur ausgegangen, so daß Interaktion fest programmiert werden kann, anstatt sie dynamisch zu erzeugen. Trotzdem ist die Betrachtung ihrer Software interessant. So nutzt der Stanford iRoom [FJHW00] ebenfalls eine eventbasierte Middleware, die auch dazu genutzt wird, per Event eine Webseite auf einem Display anzuzeigen.

Unserem Projekt am ähnlichsten sind Small Screen/Composite Device (SS/CD) [PSG00] und der UbiCompBrowser [BSLG98]. Bei beiden werden von einem PDA Multimediainhalte abgerufen und auf Geräten in der Nähe des Benutzers präsentiert. Bei SS/CD werden auch mehrere Geräte gruppiert, wenn kein einzelnes akzeptable Qualität liefern kann. Beide Projekte unterscheiden sich durch den Fokus auf Multimedia in zwei Punkten von unserem. Erstens ist die Wahl eines Ausgabegeräts erheblich leichter, da vom Typ der Medien eingeschränkt. Zweitens ist Interaktion erheblich komplexer als Medienwiedergabe, weil neben der Aus- auch die Eingabe von der Umgebung gehandhabt werden sollte. SS/CD und der UbiCompBrowser beschränken die Eingabe auf den PDA.

Unsere Ideen sind ein erster Schritt auf dem Weg, Interaktion mit dem Computer allgegenwärtig verfügbar zu machen. Neben der technischen Machbarkeit, mehrere Geräte koordiniert einzusetzen, gilt es außerdem menschliche Faktoren zu berücksichtigen. Erstens ist die Aufmerksamkeit des Menschen eine begrenzte Ressource. Und zweitens ist es bei Benutzerschnittstellen wichtig, daß sie sich für den Nutzer konsistent und vorhersagbar verhalten. Daher wollen wir in Zukunft untersuchen, wie man einen Überfluß an Interaktionsmöglichkeiten behutsam und intelligent koordiniert einsetzt. Außerdem stellt sich die Frage, wie man dies in einer Mehrbenutzerumgebung so implementiert, daß nur ein Mindestmaß an menschlicher Administration nötig ist, Benutzer sich nicht gegenseitig stören, Ressourcen gerecht geteilt werden, und nicht mißbraucht werden können.

## Literatur

- [APB<sup>+</sup>99] M. Abrams, C. Phanouriou, A. Batongbacal, S. Williams, and J. Shuster. UIML: An Appliance-Independent XML User Interface Language. In *Proceedings of 8th International World-Wide Web Conference (WWW'8)*, pages 1695–1708, Toronto, Canada, 1999.
- [BRH94] F. Bennett, T. Richardson, and A. Harter. Teleporting - Making Applications Mobile. In *Proceedings of 1994 Workshop on Mobile Computing Systems and Applications*, Santa Cruz, USA, 1994.
- [BSLG98] M. Beigl, A. Schmidt, M. Lauff, and H.W. Gellersen. The UbiCompBrowser. In *Proceedings of the 4th ERCIM Workshop User Interfaces for All*, Stockholm, Sweden, 1998.
- [FJHW00] A. Fox, B. Johanson, P. Hanrahan, and T. Winograd. Integrating Information Appliances into an Interactive Workspace. *IEEE Computer Graphics and Applications*, 20(3):54–65, 2000.
- [HAA<sup>+</sup>02] A. Hartl, E. Aitenbichler, G. Austaller, A. Heinemann, T. Limberger, E. Braun, and M. Mühlhäuser. Engineering Multimedia-Aware Personalized Ubiquitous Services. In *IEEE Fourth International Symposium on Multimedia Software Engineering (MSE 2002)*, pages 344–351, Newport Beach, USA, 2002.
- [PSG00] T. Pham, G. Schneider, and S. Goose. A Situated Computing Framework for Mobile and Ubiquitous Multimedia Access Using Small Screen and Composite Devices. In *Proceedings of the 8th ACM international conference on Multimedia*, pages 323–331, Marina del Rey, USA, 2000.