

Reihe: Telekommunikation @ Mediendienste · Band 14

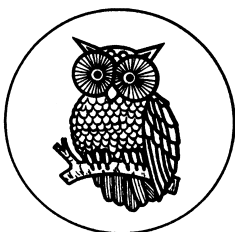
Herausgegeben von Prof. Dr. Dr. h. c. Norbert Szyperski, Köln, Prof. Dr. Udo Winand, Kassel, Prof. Dr. Dietrich Seibt, Köln, Prof. Dr. Rainer Kuhlen, Konstanz, Dr. Rudolf Pospischil, Brüssel, Prof. Dr. Claudia Löbbecke, Köln, und Prof. Dr. Christoph Zacharias, Köln

PD Dr.-Ing. habil. Martin Engelen
Dipl.-Inf. Jens Homann (Hrsg.)

Virtuelle Organisation und Neue Medien 2002

Workshop GeNeMe2002
Gemeinschaften in Neuen Medien

TU Dresden, 26. und 27. September 2002



JOSEF EUL VERLAG
Lohmar · Köln

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Virtuelle Organisation und Neue Medien 2002 / Workshop GeNeMe 2002 – Gemeinschaften in Neuen Medien – TU Dresden, 26. und 27. September 2002. Hrsg.: Martin Engelen ; Jens Homann. – Lohmar ; Köln : Eul, 2002

(Reihe: Telekommunikation und Medienwirtschaft ; Bd. 14)

ISBN 3-89936-007-9

© 2002

Josef Eul Verlag GmbH

Brandsberg 6

53797 Lohmar

Tel.: 0 22 05 / 90 10 6-6

Fax: 0 22 05 / 90 10 6-88

<http://www.eul-verlag.de>

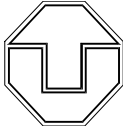
info@eul-verlag.de

Alle Rechte vorbehalten

Printed in Germany

Druck: RSP Köln

Bei der Herstellung unserer Bücher möchten wir die Umwelt schonen. Dieses Buch ist daher auf säurefreiem, 100% chlorfrei gebleichtem, alterungsbeständigem Papier nach DIN 6738 gedruckt.



Technische Universität Dresden
Fakultät Informatik • Institut für Angewandte Informatik
Privat-Dozentur Angewandte Informatik

PD Dr.–Ing. habil. Martin Engelen

Dipl.–Inf. Jens Homann

(Hrsg.)

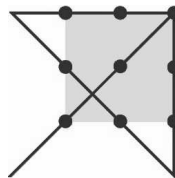


an der

Fakultät Informatik der Technischen Universität Dresden

in Zusammenarbeit mit der
Gesellschaft für Informatik e.V.,
GI-Regionalgruppe Dresden

gefördert von der Klaus Tschira Stiftung
gemeinnützige Gesellschaft mit beschränkter Haftung



am 26. und 27. September 2002

in Dresden

<http://pdai.inf.tu-dresden.de/geneme>

Kontakt: Thomas Müller (geneme@pdai.inf.tu-dresden.de)

G.2. Entwicklung eines dynamischen WAP-Interface am Beispiel der Geschäftspartnerverwaltung des Dokumentenmanagementsystems DokWorks der Firma Phoenix EDV- Systemtechnik GmbH, Itzehoe

Mathias Schnoor,

Prof. Dr. Jörg Raasch

Hochschule für Angewandte Wissenschaften Hamburg

Abstract

Seit 1999 ist es möglich, mit einem Mobiltelefon auch das World Wide Web zu benutzen. Die Technologie, die in Europa zur Verbindung von Mobiltelefon und Internet eingesetzt wird, nennt sich WAP (Wireless Application Protocol) [5], [6] und wurde als eigene, auf XML basierende Seitenbeschreibungssprache, die WML (Wireless Markup Language) [5], [6], entwickelt, die weitgehend an den HTML Standard (Hyper Text Markup Language) [8] angeglichen wurde.

Die Arbeit behandelt die im Rahmen einer Studienarbeit entstandene Programmierung von dynamischen WAP-Seiten mit dem Framework Visual Age Smalltalk Web Connection der Firma IBM.

Es wird die Möglichkeit aufgezeigt, mit einer integrierten Entwicklungsumgebung in einem adäquaten Umfang dynamische WAP-Seiten zu entwickeln, wenn die Standard-Entwicklungsumgebung IBM Visual Age Smalltalk auch für eine unter dieser Entwicklungsumgebung entwickelten Applikation (hier ein Dokumentenmanagementsystem) eingesetzt wird.

Hierzu wird eine WAP-Anwendung vorgestellt, bei der über ein Mobiltelefon aus einer Unternehmensdatenbank Informationen über Adressenangaben von Geschäftspartnern abgerufen werden, wobei insbesondere die abgerufenen Telefonnummern direkt vom Mobiltelefon übernommen und gewählt werden können. Die hierbei im Zusammenhang mit WTAI (Wireless Telephonie Applications Inteface) [9] zu lösenden Probleme werden dargestellt.

1. Einleitung

Ein Kunde aus der Baubranche, der das Dokumentenmanagementsystem DokWorks der Firma Phoenix EDV- Systemtechnik GmbH, Itzehoe, einsetzt, möchte per Mobiltelefon auf Informationen zugreifen können, die in seinem Dokumentenmanagementsystem (DMS) gespeichert sind.

Aus dieser Anforderung heraus wurde eine Studienarbeit formuliert, deren Ergebnis einen telefonischen Zugriff via Mobiltelefon auf die Geschäftspartnerverwaltung des DMS ermöglichen sollte, um dem Kunden die gewünschten Informationen zur Verfügung stellen zu können. Insbesondere wünschte sich der Kunde den mobilen Zugriff auf die Adreßdaten seiner Geschäftspartner, wenn er sich z.B. auf einer der von seinem Unternehmen betreuten Baustelle, aufhielte. Das Mobiltelefon bietet hierfür den schnellsten und einfachsten Kommunikationsweg.

Da das DMS DokWorks schon in der Entwicklungsumgebung Visual Age Smalltalk der IBM Corp. entstanden war, wurde für die Implementierung des benötigten WML-Codes das Framework Web Connection, das mit der Entwicklungsumgebung mitgeliefert wurde, eingesetzt.

2. Das Projekt

Vor der endgültigen Definition des Projektes mit seinen Entwicklungs- und Programmierarbeiten wurden noch einige Vorüberlegungen angestellt.

2.1 Vorarbeiten

Folgende Fragen mußten beantwortet werden:

- Ist die Geschäftspartnerverwaltung des DMS DokWorks als Untersuchungsobjekt für die Anbindung von Mobiltelefonen geeignet ?
- Welche zusätzlichen Funktionalitäten im Vergleich mit dem bestehenden Windows-Client des DMS DokWorks sollte diese Anwendung beinhalten?
- Welche Funktionalitäten im Vergleich mit dem bestehenden Windows-Client müssen weggelassen werden?

- Ist es möglich, mit einem HTML-basierten Framework, eine Applikation zur Anwendung auf einem Mobiltelefon zu realisieren (Web Connection ist eigentlich dafür ausgelegt, dynamische Webseiten zu entwickeln)?
- Welche Zusatztechnologien müssen für die Realisierung eines eigenen WAP-Portals eingesetzt werden?

Erst nach Klärung dieser Fragen, wurde das Projekt genauer definiert.

2.2 Ziel des Projektes

Es sollte ein dynamisches WAP-Interface für die Geschäftspartnerverwaltung des Dokumentenmanagementsystems (DMS) DokWorks entwickelt werden, um die in der Geschäftspartnerverwaltung enthaltenen Adressinformationen über juristische und natürliche Personen über ein Mobiltelefon zur Verfügung zu stellen. Die Informationen beinhalten Firmen- und Mitarbeiternamen, Anschriften, Telefoneinträge, Email-Adressen, usw.

2.3 DMS Funktionalitäten

Da Dokumentenmanagementsysteme recht weit gefächerte Leistungsspektren bieten, soll hier nicht weiter auf die Funktionalitäten von DMS [7] eingegangen werden. Es sei nur angemerkt, dass es sich bei dem DMS DokWorks um ein projektorientiertes DMS handelt, in dem nicht nur passiv Dokumente abgelegt werden, sondern auch aktiv Dokumente erzeugt werden können. Weitere Informationen können bei der Phoenix EDV-Systemtechnik GmbH¹, Itzehoe, angefordert werden.

3. Die Vorgaben

Die einzelnen Bausteine, die der Erreichung des Projektzieles dienen, werden im Folgenden dargestellt.

¹ Phoenix EDV Systemtechnik GmbH, Fraunhoferstrasse 3, 25524 Itzehoe (URL: <http://www.phoenix-edv.com/>)

3.1 Dynamisches WAP-Interface

Da das verwendete DMS als ein dynamisches System realisiert worden ist, das heisst die Einträge innerhalb dieses DMS können und werden im laufenden Betrieb zeitnah aktualisiert, sollte auch ein zu realisierendes WAP-Interface diese Eigenschaft besitzen. Um deshalb sicher zu gehen, dass beide Applikationen auf einen äquivalenten Datensatz zugreifen, sollten beide Applikationen ihre Daten von ein und der selben Datenbank beziehen.

Um diese Vorgabe zu realisieren, sollten folgende Werkzeuge eingesetzt bzw. weggelassen werden:

- WML - Wireles Markup Language [5], [6] wird verwendet zur Implementierung der benötigten Seiten
- keine Verwendung WML-Skript [5], [6] (entspricht HTML, Javaskript), da die Dynamik mit Hilfe des verwendenden Frameworks Web Connection erreicht werden sollte.
- keine Verwendung der Graphikelemente, da der Einsatz für dieses Projekt redundant schien.

Die verwendete Seitenbeschreibungssprache Wireles Markup Language (WML) [5], [6] wird zu einem späteren Zeitpunkt ausführlicher erläutert werden.

3.2 Read-Only-Nachbildung der Geschäftspartnerverwaltung des Dokumentenmanagements DokWorks

Das entstehende WAP-Interface sollte als Read-Only Nachbildung des oben aufgeführten DMS entstehen. Es sollten also nur lesende Zugriffe auf die zu verwendende Datenbank erlaubt werden.

3.2.1 Geringere Funktionalitäten als der Windows-Client

Die Anwendung sollte auf einem Mobiltelefon abgebildet werden. Mobiltelefone sind für manuelle Texteingaben nur bedingt tauglich, weil im Normalfall eine vollständige Tastatur fehlt und somit die Eingaben recht umständlich sind. Daher würde es wenig Sinn machen, auch schreibende Zugriffe zu erlauben. Auch sollte die Datenänderung

dem Windows-Client vorbehalten bleiben. Aus diesen Gründen sind in der vorliegenden Entwicklung nur Read-Only-Zugriffe erlaubt. Eine Erweiterung auf eine Read-Write-Funktionalität wäre aber jederzeit realisierbar.

3.2.2 Erweiterte Funktionalitäten zum Windows-Client

Die entstehende Applikation sollte auch erweiterte Funktionalitäten beinhalten, um sich von dem bestehenden Windows-Client abzugrenzen. Hierbei geht es in erster Linie um Funktionalitäten, die ein Mobilfunkgerät zur Verfügung stellen könnte.

Da in der Geschäftspartnerverwaltung des DMS zu jeder eingetragenen natürlichen oder juristischen Person auch der Eintrag von Telefonnummern vorgesehen ist, soll dieser Eintrag vom Mobiltelefon übernommen werden. Es soll ein selbständiger Rufaufbau zu dieser gewählten Telefonnummer durch das Mobiltelefon erfolgen.

Außerdem soll das Senden von Emails ermöglicht werden. Allerdings soll diese Funktionalität und die damit verbundenen Kosten nicht über den Mobilfunkanbieter geschehen, sondern über das DMS Dokworks abgewickelt werden, da das DMS Dokworks schon eine solche Funktionalität beinhaltet.

3.3 Einführung eines Sessionmanagement

Eine ganz wichtige Vorgabe ist die Einführung eines Sessionmanagements. Das DMS arbeitet als Mehrbenutzersystem benutzerorientiert. Das bedeutet, dass jeder potentielle Benutzer sich vorher mit seinem Benutzernamen und seinem Passwort authentifizieren muss. Die Applikation wiederum kommuniziert über das Wireless Application Protokoll, das zustandslos ist, wie das Internetprotokoll HTTP [10].

Um die gleiche Abstraktion des DMS auch innerhalb einer WAP-Applikation zur Verfügung zu stellen, muss ein Management eingeführt werden, das jeden Benutzer mit nur für ihn relevanten Daten arbeiten lässt und diese ihm zur Visualisierung zu Verfügung stellt. Hierfür gibt es das sogenannte Sessionmanagement.

Beim Sessionmanagement sollte man unterscheiden:

3.3.1 Statische Datenhaltung

Die statische Datenhaltung betrifft hier Daten, die während der gesamten Session gültig sind und ggf. immer wieder abgefragt werden können.

Zu den Statischen Daten gehören:

- Die Benutzerkennung zur eindeutigen Authentifizierung
- Das Passwort zur eindeutigen Authentifizierung
- Der Sessionkey, um die einzelnen Anmeldungen in einem Mehrbenutzerbetrieb zu kapseln
- Die Sessiontime, um ein kontrolliertes Ende der Sitzung herbeiführen zu können, wenn es der User nicht schon manuell beendet hat

3.3.2 Dynamische Datenhaltung

Unter der dynamischen Datenhaltung versteht man den Transfer von sich ändernden Applikationsdaten innerhalb der WML-Seiten, die sich im Laufe einer Session ansammeln oder ändern können. Hier sei das Beispiel eines virtuellen Warenkorbes angeführt. Ein User bewegt sich während seiner Sitzung über mehrere Seiten hinweg und sammelt virtuelle Gegenstände ein, z.B. für einen eventuellen späteren Kauf. Diese Ansammlung kann er dann während seiner Sitzung beliebig erweitern oder verringern, er nimmt also Gegenstände in seinen Warenkorb hinein oder heraus. In dieser Anwendung kann in ähnlicher Weise die Nutzung von Adressdaten der Geschäftspartnerverwaltung koordiniert werden.

4. Die eingesetzten Werkzeuge

Die folgenden Werkzeuge waren zur Entstehung des dynamischen WAP-Interface notwendig.

(a) *Visual Age Smalltalk [2],[3],[4]*

Der Hersteller dieser Plattform ist die Firma IBM Corp.. Smalltalk ist eine mächtige und echte Objektorientierte Programmiersprache (echt bedeutet: alles ist ein Objekt. Effekte sind nur erzielbar, indem Nachrichten an Objekte gesendet werden). Sie ist entstanden aus der Ableitung von Lisp und Simula und behauptet sich trotz der derzeitigen Java Euphorie weiterhin als sehr effektive und mächtige Programmiersprache.

(b) *Visual Age Smalltalk Web Connection [1]*

Web Connection ist ein Framework zur Erstellung von dynamischen Webseiten. Es ist in die Entwicklungsumgebung von Visual Age Smalltalk integriert.

(c) *Dokumentenmanagementsystem DokWorks*

Dieses Dokumentenmanagementsystem der Firma Phoenix Edv- Systemtechnik GmbH (<http://www.phoenix-edv.com/>) bot sich zur Vorgabe an, da es ebenfalls in Visual Age Smalltalk entwickelt ist und weiter entwickelt wird. Da beide Applikationen in Visual Age Smalltalk entwickelt werden sollten oder entwickelt worden sind, bieten sich bei der Modulintegration entscheidende Vorteile.

(d) *Apache Webserver*

Als Anbieter von anwendungsbezogenen Web-Dienstleistungen benötigt man einen Webserver, der die von Browsern eingehende Requests unter Nutzung von Schnittstellen der Anwendung in Responses umsetzen kann. Zur Entwicklung des dynamischen WAP Interfaces wurde der Apache Webserver verwendet.

(e) *Yourwap Wapsimulator*

Wenn man eine Applikation zur Anwendung auf einem mobilen Endgerät (Mobiltelefon) erstellt, so muss man diese aus Kostengründen zunächst als emuliert testen können. Hierfür bot sich der Wapsimulator von der Firma Yourwap an. Diesen gibt es als kostenlosen Download im Internet unter <http://www.yourwap.com>.

(f) *Nokia Toolkit 3.0*

Dieses Toolkit ist nach einer Registrierung unter www.nokia.com, frei im Internet verfügbar. Hiermit sind alle für die Aufgabenstellung relevanten Eigenschaften der Anwendung emulierbar. Es ist Java-basiert, dadurch ist es allerdings recht langsam in der Ausführung.

(g) *Mobile Gateway*

Dieses Gateway kam als kostenlose 90 Tage Version von der Firma Checkcom GmbH und wird zur Realisierung eines eigenen WAP - Portals benötigt. Das Gateway im Allgemeinen wird noch genauer unter Punkt 5.5 behandelt, da der Einsatz eines eigenen

Mobilen Gateways von unterschiedlicher Bedeutung zum Erreichen des Zieles dieser Projektes war.

5. Die Realisierung des WAP-Interfaces

In diesem Kapitel wird die Realisierung des dynamischen WAP - Interfaces behandelt. Die benötigten Technologien und die eingesetzten Werkzeuge werden dabei geschildert.

5.1 Die Architektur

Die Bild 1 zeigt die Architektur des WAP-Interfaces. Die Anbindung des Mobiltelefons erfolgt über Mobilfunkanbieter, WAP-Gateway und Proxy. Der Informationsverlauf wird über das Common Gateway Interface (CGI) gesteuert. Durch den Webserver wird eine Execute-Datei (Abtwasc.exe) gestartet. Diese Datei wird durch Visual Age Smalltalk Web Connection standardmässig mitgeliefert. Es ist in der Programmiersprache C geschrieben und kommuniziert mit der Anwendung PhoenixWAP, die das DMS DokWorks enthält. Ergebnisse werden über Stdout der Execute-Datei wieder dem Webserver zugänglich gemacht. Dem Webserver werden damit die Http-Requests zugeführt (Funktion Get), er erarbeitet die anwendungsspezifische Antwort, bereitet diese als WML-Seite auf und gibt sie als Response (Funktion Post) an das Handy (WAP-Browser) zurück.

Hier ist durchaus noch eine andere Architektur vorstellbar, in der nicht mit dem CGI, sondern mit Servlets gearbeitet wird. Da die WAP-Applikation aber nicht für eine Massenanwendung konzipiert wurde, reicht das CGI vorläufig vollkommen aus.

Endgeräte zur Verfügung gestellt werden. Nahezu alle grösseren Hardware-, Software und Gerätehersteller gehören mittlerweile dazu, ausserdem die wichtigsten Träger- und Telekommunikations-Anbieter.

Die Spezifikation definiert verschiedene Eckwerte, wie mobile Geräte auf Datendienste wie das Internet zugreifen können. So werden z.B. das Übertragungsprotokoll vom Gerät zum Internet oder die Wireless Markup Language (WML), mit dem ähnlich wie bei der Hypertext Markup Language (HTML) die WAP-Seiten programmiert werden, festgelegt. Das WML ist dem HTML angelehnt worden, um Umsteigern das Erlernen dieser neuen Sprache einfacher zu machen. WML basiert aber im Gegensatz zu HTML auf XML.

Die aktuelle WAP-Version ist zur Zeit das Release 1.3. Allerdings sollte aus Gründen der Kompatibilität mit älteren Mobiltelefonen eher auf die Spezifikation 1.2 zurückgegriffen werden. Aber auch eine neuere Version WAP 2.0 steht schon kurz vor der Einführung.

WAP wurde u.a. entwickelt, um die geringen Datenübertragungsraten des GSM-Netzes effizient auszunutzen. Diese betragen 9600 Bit/s und bedeuten für einen Internetzugang eine unzumutbare Geschwindigkeit. Um mit dieser Bandbreite effektiver umzugehen, komprimiert der WAP-Standard die Daten.

Außerdem haben mobile Geräte relativ kleine Displays. Aus diesem Grund sind normale HTML-Seiten, welche auf große Bildschirme angepasst sind und eventuell Javascript und andere Funktionen benutzen, nicht so ohne weiteres auf einem Mobiltelefon darstellbar. Hinzu kommt das Fehlen einer ergonomischen Tastatur. Daher ergab sich die Notwendigkeit einer neuen Sprache, welche speziell an die Beschränkungen mobiler Geräte angepasst ist.

WAP eignet sich also in erster Linie für mobile Geräte wie Mobiltelefone und PDAs, die über den GSM Standard oder dessen Nachfolger wie GPRS oder UMTS mit dem Internet kommunizieren. Jedes mobile Netz kann damit genutzt werden.

(c) *Wireless Markup Language (WML) [5],[6]*

WML ist eine HTML-ähnliche Sprache (Hypertext Markup Language) [12], die wie jene als XML-Dokumenttyp (Extensible Markup Language) definiert ist. Die Anlehnung an HTML trat allerdings erst später ein, um Programmierern den Umstieg

von HTML nach WML zu erleichtern. WML ist für kleine Bildschirme und wenig Speicherbedarf von mobilen Endgeräten optimiert.

WML basiert auf der HDML Version 2.0 (Handheld Device Markup Language), wurde aus o.g. Gründen jedoch überarbeitet, um HTML weitgehend zu gleichen. In WML werden Metaphern Card (Karte) und Deck (Kartenstapel) beschrieben. Diese entsprechen den Seiten eines HTML-Dokuments.

Wie HTML verwendet WML Formatierungsbefehle, um das Aussehen von Texten, die Positionierung von Hyperlinks, Bildern und Formularen sowie die Hierarchie von Cards und Decks festzulegen. Die in WML eingebaute Hierarchie erlaubt es, dasselbe Dokument wahlweise auf einem Mobiltelefon oder einem Organizer (PDA) anzuzeigen.

Ein WML-Dokument wird als Deck bezeichnet, wenn es aus einer oder mehreren Cards besteht, die den darzustellenden WML-Code enthalten. Jedes einzelne Deck entspricht einer HTML-Seite und wird über eine URL-Adresse (Uniform Resource Locator) angesprochen. Ein Deck oder die Card eines Decks können Dienste eines Servers ansprechen, um bestimmte Aufgaben zu erfüllen. Die Aufgaben können auch mit Hilfe von WML-Script-Funktionen erledigt werden.

Wie bei HTML sind WML-Dateien reine Textdateien. WML ist eine Formatierungssprache. Dies bedeutet, dass der Browser endgültig festlegt, wie das Dokument auf dem Display (Bildschirm) erscheint. Formatierungsbefehle dienen dazu Texteigenschaften wie Fettdruck oder Kursivschrift usw. sowie Zeilenwechsel, Bildplatzierungen und weiteres vorzugeben. Wie bei den ersten HTML-Versionen sollten diese Anweisung jedoch eher als Vorschläge für den Browser betrachtet werden.

(d) Das Wap-Gateway

Beim WAP-System liegt zwischen Client und dem Server ein Gateway. Dieses ist für den nahtlosen Übergang von der mobilen Übertragung zum Internet zuständig.

Diesem Gateway (auch WAP-Proxy genannt) fällt die Aufgabe zu, zwischen dem Telefonnetz und dem Internet zu übersetzen. Vom Gateway zum Internetserver werden hier die Protokolle auf HTTP und TCP/IP umgesetzt und die dann erhaltenen Daten vom Gateway zum Mobiltelefon binär codiert und komprimiert, um die Datenmengen zu verkleinern und Bandbreite zu sparen. In der WAP-Architektur kann ein Endgerät niemals direkt eine Verbindung mit dem Webserver aufnehmen. Es ist immer ein zwischengeschaltetes WAP-Gateway notwendig, welches die Protokollumsetzung

vornimmt. Dieses WAP-Gateway kann physikalisch wahlweise bei dem Mobilfunkanbieter (D1, D2, E-Plus, usw.), bei einem Serviceprovider oder bei dem den Inhalt liefernden Unternehmen installiert sein.

(e) Das Framework Web Connection [1]

Das Framework Web Connection ist ein Werkzeug der Firma IBM Corp. zur Erstellung eines dynamischen Webauftrittes. Es erfordert die Entwicklungsumgebung Visual Age Smalltalk. Folgende Komponenten sind in diesem Framework integriert:

- Der WebFormPart
- Der WebConnection Part

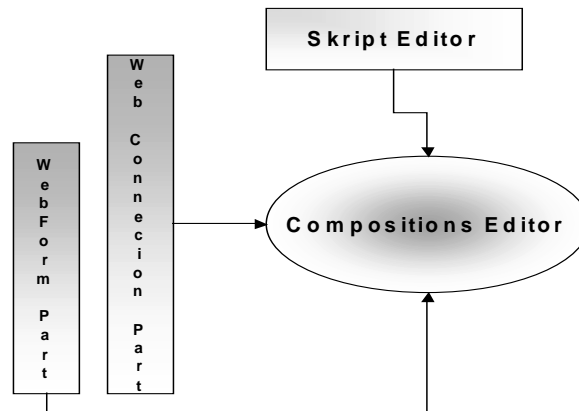


Abb. 2: Zusammenwirken der einzelnen Web Connection Werkzeuge

- Der Kompositioneditor

Das Framework sollte auch dafür benutzt werden, statt HTML-Code nun auch WML-Code zu interpretieren. Damit wird es für die Erstellung der benötigten WML-Seiten verwendbar. Daher war eine Erweiterung und Änderung des Frameworks erforderlich.

5.3 Der Aufbau eines Standard WAP-Portals

Bild 3 zeigt den Kommunikationweg mit dem Einsatz des WAP-Gateways. Dieses Gateway ist bei dem Telefonanbieter oder dem Serviceprovider installiert und wird über das mobile Endgerät angesprochen. Diese Architektur konnte für die Entwicklung eines eigenen dynamischen WAP-Interfaces nicht gewählt werden, da sonst ein Nutzer des entwickelten Systems (im Regelfall eine Firma) eine Standleitung ins Internet haben müsste, oder das System mit samt dem dazugehörigen DMS sowie der Datenbank müsste bei dem Telefonanbieter, respektive beim Serviceprovider installiert werden.

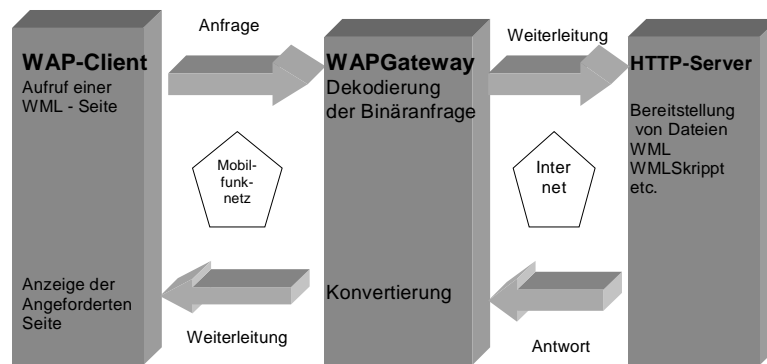


Abb. 3: Übertragungsweg innerhalb eines Standard-WAP-Portals

Deshalb musste für die Realisierung eines eigenen dynamischen WAP-Interfaces eine andere leicht abgewandelte Architektur gewählt werden, in der das schon angesprochene Mobile Gateway zum Einsatz kommt.

5.4 Aufbau eines eigenen WAP-Portals

Wie in Bild 4 ersichtlich, liegt der Unterschied in dieser Architektur im Gegensatz zu der unter Bild 3 aufgezeigten Architektur darin, dass das benötigte Gateway nicht mehr

beim Mobilfunk- oder Serviceprovider alloziert ist, sondern bei dem potentiellen Nutzer (eine Firma) eines solchen WAP-Interfaces. Diese Architektur hat den Vorteil, dass beim Provider nur noch das Mobilfunknetz genutzt werden muss. Um diese Architektur anwenden zu können, muss parallel im mobilen Endgerät eine Konfiguration vorgenommen werden, da jetzt z.B. nicht mehr der Mobilfunkanbieter, sondern eine speziell konfigurierte Telefonnummer des Nutzers angewählt werden muss.

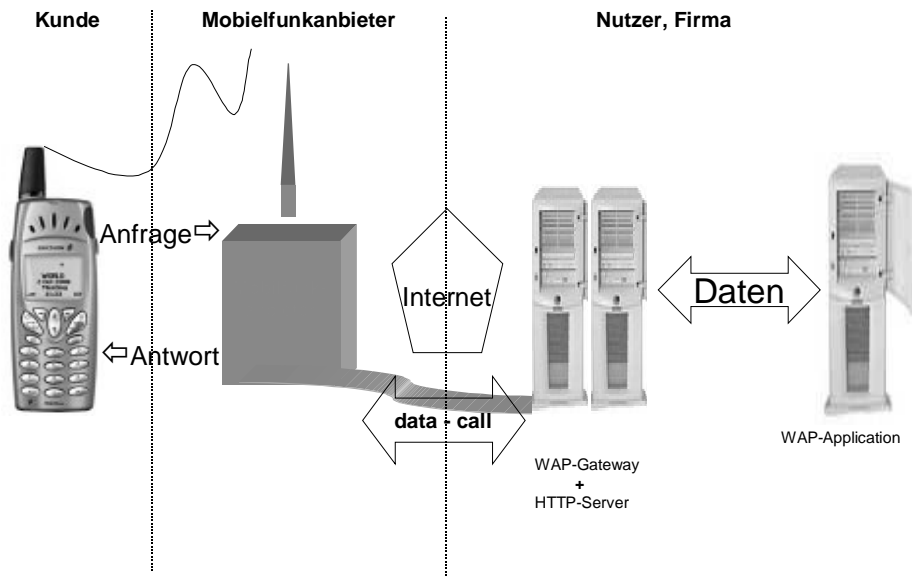


Abb. 4: Übertragungsweg innerhalb eines eigenen WAP-Portals

5.5 Die Mobiltelefonkonfiguration (am Beispiel des Nokia 6310)

- Startseite: <http://servername/cgi-bin/abtwsac.exe/PHWapLogIn>
- Verbindungstyp: Temporaer
- Verbindungssicherheit: Aus
- Übertragungsart: Daten
- Anwahlnummer: default number user
- IP-Adresse: Gateway IP



-
- Authentisierungstyp: Normal
 - Datenanruftyp: ISDN
 - Übertragungsgeschw.: 9600 Baud
 - Benutzername: optional
 - Kennwort: optional

Folgende Optionen müssen individuell eingestellt werden:

- Startseite: es muss die aktuelle URL der Applikation eingetragen werden.
- Anwahlnummer: es muss ein speziell dafür vorgesehener Telefonanschluss eingetragen werden
- IP-Adresse: es muss individuell die IP-Adresse des mobilen WAP- Gateways eingetragen werden

6. Zusätzlich realisierte Funktionen

Unter Kapitel 2 und Kapitel 3 ist schon erwähnt worden, dass eine besondere Mobiltelefonfunktionalität innerhalb des WAP-Interfaces realisiert werden sollte. Beim Erscheinen einer Telefonnummer innerhalb eines WAP-Auftritts, sollte das Mobiltelefon in die Lage versetzt werden, durch Aktivieren dieser Nummer selbständig diese Nummer zu wählen und einen Rufaufbau vorzunehmen, ohne dass eine weitere Aktion durch den User auszuführen ist. Der Mobiltelefonnutzer kennt so etwas aus der Telefonbuchfunktion seines Mobiltelefons. Hier wählt er sich auch aus diesem einen Gesprächspartner heraus, und durch Aktivieren einer Taste wird die oben beschriebene Funktionalität hergestellt.

Im Vorfeld des hier geschilderten Projektes wurde in Lehrveranstaltungen an der HAW-Hamburg eine ähnliche Nutzung von WAP-Mobiltelefonen angestrebt. Diese konnte allerdings ohne die im Folgenden zu beschreibende WTAI-Funktionalität damals nicht vollständig realisiert werden.[13]

6.1 Wireless Telefonie Application Interface (WTAI)

Ab WAP 1.2 gibt es auch für das Wireless Application Protokoll eine erweiterte Funktionalität, das Wireless Telephony Application Interface (WTAI). Diese ist mit WML und WML-Skript für die Anwendung auf einem mobilen Endgerät implementierbar.

Das WTAI erlaubt es, typische Funktionen eines Mobiltelefons per WML oder WML-Script aufzurufen.

WTAI umfasst Zugriffe auf:

- Anruffunktionen
- Text Nachrichten
- Telefonbuch/Adressbuch

und bringt seine eigenen Bibliotheken mit:

(a) Public WTA Function Libraries

Bibliothek	Name	Beschreibung
Public WTAI	Wp	Allgemein zugängliche Funktionen

(b) Network Common WTA Function Libraries

Voice Call	Vc	Rufnummern wählen, Status abfragen
Network Text	Nt	Textnachrichten SMS
Phonebook	Pb	Telefonbuch
Call Logs	Cl	Anruf(er)-Listen
Miscellaneous	Ms	Verschiedenes

Das bedeutet, dass sich zukünftig alle zusätzlichen Mobiltelefonfunktionen mit der Bibliothek Public WTA Funktion Libraries unter WML oder WML-Skript implementieren lassen. Um dieses sicherzustellen, müssen die Mobiltelefonhersteller jedoch diese Funktionalität in ihren mobilen Endgeräten bereitstellen. Bis zum Abschluss des Projektes war aber leider von den Mobiltelefonherstellern keine

offiziellen Statements bezüglich WTAI Implementation zu bekommen und nur durch Umwege war in es Erfahrung zu bringen, dass in einigen Geräten die Anruffunktion (makecall) implementiert worden ist. Erst dadurch konnte diese Funktionalität makecall in dem dynamischen WAP-Interface mit aufgenommen werden und die gestellte Vorgabe erfüllt werden. Verifiziert worden ist dieses bei den Geräten:

- Siemens 35i
- Nokia 6310

6.2 Codebeispiel für WTAI (makecall)

```
<?xml version="1.0"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">

<wml>

<card id="anrufcard" title="Anruf">

    <do type="accept" label="Anrufen">

        <go href="wtai://wp/mc;0123456789"/>

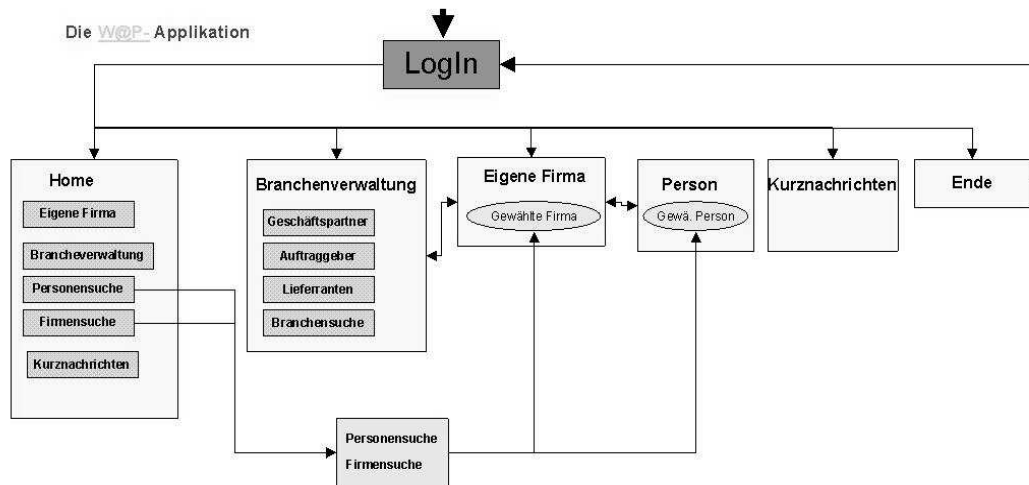
    </do>

</card>

</wml>
```

7. Die WAP Applikation

7.1 Die Dialogstruktur



A

bb. 5: Die Dialogstruktur

Als Ergebnis dieses Projektes ist ein dynamisches WAP-Interface entstanden. Dieses ist unter dem Namen PoenixWAP am Markt verfügbar. PhoenixWAP kann bei der Firma Phoenix Edv- Systemtechnik GmbH in Itzehoe zusammen mit dem bei dieser Firma entwickelten DMS käuflich erworben werden.

7.2 Die Emulation der Applikation PhoenixWAP

Emuliert wurde die Applikation PhoenixWAP mit den unter Punkt 3. aufgeführten Tools. Dabei konnte mit dem Nokia Tollkit 3.0 sogar das verwendete mobile Gateway sowie die WTAI Anruffunktion emuliert werden.



Abb. 6: Handyemulation

7.3 Die Applikation unter Alltagsbedingungen

Getestet wurde die Applikation PhoenixWAP unter Alltagsbedingungen mit den mobilen Endgeräten Nokia 6210, Nokia 6310 und Siemens 35i.

- Das Nokia 6210 ist trotz des implementierten WAP 1.2 Standard nicht WTAI fähig.
- Das Nokia 6310 konnte als für diese Applikation voll funktionstüchtig bewertet werden. Es gab bei den Tests auch im Hinblick auf die WTAI (makecall) Funktionalität keinerlei Beanstandungen.
- Das Siemens 35i ist zwar WTAI (makecall) fähig, doch leider hat dieses Mobiltelefon einen eingebauten Softwarefehler, der nur durch Umprogrammierung in der Applikation behoben werden konnte, um das Mobiltelefon zu testen.

8. Zusammenfassung

Durch das geschilderte Projekt konnte ein Beleg für eine wirtschaftlich sinnvolle WAP-Anwendung erbracht werden.

Auch konnte gezeigt werden, dass ein solches Interface in der objektorientierten Entwicklungsumgebung Smalltalk (hier Visual Age Smalltalk) entwickelt werden kann. Der Vorteil dieser eingesetzten Entwicklungsumgebung liegt darin, dass sowohl die

während dieses Projektes entstandenen Applikation PhoenixWAP als auch das als Vorgabe dienende DMS DokWorks in Visual Age Smalltalk entwickelt wurden.

Die WAP- und WML-Technologien sind gerade im Bezug auf die Seitenbeschreibungssprache WML sicher noch ausbaufähig, was nicht alleine daraus resultiert, dass die Hersteller von mobilen Endgeräten die Beachtung der Definitionen der WAP-Forum-Implementationstandards vermissen lassen. Gleicher Code wird von Mobiltelefon zu Mobiltelefon anders interpretiert oder ist überhaupt nicht darstellbar. Hier muß mit jedem Mobilfontyp der einzelnen Hersteller die WAP-Tauglichkeit überprüft werden.

Die Übertragungsgeschwindigkeit durch GSM von 9600 Bauds erscheint für die reine Textübermittlung durchaus vertretbar. Bei der zusätzlichen Übertragung von Bildelementen wird man sicherlich schlechtere Ergebnisse erzielen.

9. Danksagung

Besonderer Dank gilt der Firma Phoenix EDV-Systemtechnik GmbH in Itzehoe im Allgemeinen und Herrn Marten Feldmann im Besonderen, für die permanente Unterstützung und die Bereitstellung sämtlicher Hard und Software, sowie der benötigten Literatur.

Besonderen Dank gilt der Firma Mobilty United für die Überlassung eines mobilen Gateway in einer 90 Tage Testlizenz der Firma Checkcom.

Besonderen Dank Herrn Timm von der Firma Talkline in Elmshorn, für die Herausgabe einer Liste mit WTAI fähigen Mobiltelefons.

Einen Besonderen Dank gilt den Firmen Herbert Freiberg und Karstadt AG in Itzehoe, für die kostenlose Bereitstellung eines WTAI-fähigen Mobiltelefons.

10. Literatur

- [1] IBM VisualAgeSmalltalk WebConnection UserGuide 5.5, 2000
- [2] IBM VisualAgeSmalltalk UserGuide 4.0, 1997
- [3] IBM VisualAgeSmalltalk Programmer's Reference 4.0, 1997
- [4] IBM Smalltalk The Language, David N. Smith, Benjamin/ Clummings Publishing Company, Inc, 1994

-
- [5] WAP, WML und WMLScript developer's guide, Volkmar Gronau, Markt+Technik Verlag München, 2001
 - [6] WAP- Programmierung, Pekka Nisanen, WebBook SYBEX-Verlag GmbH, Düsseldorf, 2001
 - [7] www.dog.de, DOQ-Magazin für Dokument/ Content/ Knowledge-Management, H&T- Verlagsgesellschaft mbH & Co KG, München, 2002
 - [8] <http://selfhtml.teamone.de/index.htm>, Stefan Münz, Version 8, 2001
 - [9] www1.wapforum.org/tech/documents/wtai-30-apr-98.pdf , Version 1998
 - [10] <http://www.informatik.uni-trier.de/~tiscrypt/%20ep/html/ps-pdf/ep-2-2.pdf>
 - [11] <http://wap-forum.org>
 - [12] <http://www.t-lan.de/GLOSSAR/begriffe/hdml.htm>
 - [13] Mobiltelefone und Organizer als Zugangsmedien zu Informationssystemen, K. Cords, L. Gerken, K. Panier, N. Thyssen, J. Raasch. In: Martin Engelin, Jens Homan (Hrsg): Virtuelle Organisation und Neue Medien 2001, Tagungsband TU Dresden 27./ 28.09.2001, Josef Eul Verlag, 2001, S 279- 306

