

Fachseminar Mobile Computing

„Darstellungskomponente und Szenarien für die Symbolische Lupe“

SS 2001



Tobias Wallin, ERASMUS-Student 00-904-078, wallint@student.ethz.ch
Prof. Dr. Friedemann Mattern, Assistent Michael Rohs
ETH Zürich [31.07.2001]

Abstract

Alltagsdinge lassen sich mit Online-Ressourcen im Internet verknüpfen, die virtuelle Entsprechungen dieser Alltagsdinge darstellen. Diese virtuellen Gegenstücke können den Zustand des physischen Objektes widerspiegeln, Informationen über dessen Eigenschaften geben oder sogar dessen Funktionalität erweitern.

Ein einfaches Anwendungsbeispiel ist die Verknüpfung von Produkten zu Online-Ressourcen beim Hersteller. Dies kann z.B. als Marketinginstrument benutzt werden, indem Produktinformationen, Werbung oder Grunddienste des Produktes bereitgestellt werden. Für den Kunden wird so der Umgang mit dem Produkt vereinfacht und unterstützt.

Voraussetzung dafür ist die automatische Identifizierung der Objekte, die z.B. durch RFID-Marken (Radio Frequency IDentification) oder Barcodes erreicht werden kann.

Die symbolische Lupe ist ein kleines tragbares Gerät, das physische Objekte identifiziert, die ihnen zugeordneten Ressourcen drahtlos überträgt und sichtbar macht. So wie eine reale Lupe eine detaillierte Ansicht eines realen Objektes bietet, zeigt die symbolische Lupe eine symbolische Ansicht desselben.

Die aktuelle Realisierung besteht aus einem Hitachi Handheld PC und einem PC-Card Barcode-Scanner bzw. einem RFID-Leser. Die Objektinformationen werden über WLAN an das Gerät übertragen. Als Anwendungsprotokoll wird HTTP (HyperText Transfer Protocol) verwendet. Die Objektdaten werden als HTML-Seiten (HyperText Markup Language) angezeigt.

Ich habe mich in meiner Arbeit mit der Implementierung des Backends beschäftigt, das die Objektdaten verwaltet. Dazu wird der Tomcat HTTP Server und Java Servlets verwendet. Als Beispielszenario möchte ich die Verwendung von mit Barcodes markierten Orten zur Navigation zeigen.

Diese Arbeit umfasst eine Darstellung der wesentlichen Komponenten, eine Zusammenfassung von möglichen Szenarien und Anwendungen für das System und deren schriftliche und mündliche Präsentation.

Inhaltsverzeichnis

<i>Abstract</i>	2
<i>Inhaltsverzeichnis</i>	3
<i>Problembeschreibung</i>	4
<i>Die Darstellungskomponente</i>	5
Erweiterung der Aufgabe mit XML-Support	5
Die Systemarchitektur.....	6
<i>Industrielle Barcodeleser</i>	7
Existierende Arbeitsgebiete	7
In Tageszeitungen.....	8
Auf Lebensmittelpackungen.....	8
In der Logistik.....	8
<i>Neue Szenarien für die Symbolische Lupe</i>	9
Verbesserung des Logistikszenarios	9
Verbesserung des Verpackungsszenarios.....	9
Gebäudeorientierungsszenario.....	10
<i>RFID Leser und Marker</i>	11
<i>Fazit</i>	12
Schwierigkeiten.....	12
Vorteile meiner Lösung.....	12
Danken.....	12
<i>Literaturverzeichnis</i>	13

Problembeschreibung

Die symbolische Lupe ist ein in der Hand zu haltendes Gerät, mit dem virtuelle Gegenstände (zum Beispiel eine Webseite) zu physischen Objekten sichtbar gemacht werden können. Die physischen Objekte, zum Beispiel Konsumartikel in einem Supermarkt, sind mit einer RFID Marke versehen, welche die Adresse des virtuellen Gegenstands enthalten. Die Symbolische Lupe ist mit einem kleinen RFID Leser ausgestattet, mit dem die RFID Marke sowohl ausgelesen, als auch beschrieben werden kann. Die Information zu einem Objekt wird auf einem stiftbasierten Hitachi Handheld PC dargestellt. Das Betriebssystem ist Windows CE 2.11. Die Objektinformationen werden über eine PCMCIA WLAN Karte an das Gerät übertragen. Als Anwendungsprotokoll wird HTTP verwendet.

Die Objektdaten werden als HTML Seiten angezeigt, die zum Beispiel auch HTML Formulare enthalten können. Windows CE 2.11 bietet „HTML Viewer Control“ an, welches das Layout von HTML Seiten übernimmt. Dieses lässt sich in Visual C++ programmieren. Das Laden der HTML Seite über HTTP muss auch programmiert werden, zum Beispiel als einen Java Servlet oder Perl CGI-Script.

Im Prinzip ist die Handhabung des Viewer Controls wie folgt: Der HTML Code wird dem tragbaren PC übergeben. Dieser stellt die HTML Seite dar. Tauchen auf der Seite zum Beispiel Bilder auf, wird man über ein Windows-Ereignis darüber informiert. Das Bild muss dann über HTTP geladen und dem Viewer Control übergeben werden. Klickt der Benutzer einen Verweis an, so wird ebenfalls ein Ereignis erzeugt. Dieses enthält die URL (Uniform Resource Locator) der neuen Seite und muss dann geladen werden. Das Viewer Control übernimmt also nur die Darstellung, aber nicht das Laden des Inhalts. Windows CE bietet Funktionen an, die den Umgang mit dem HTTP-Protokoll wesentlich vereinfachen (HTTP WinInet Funktionen).

Die Hauptaufgabe besteht darin, mögliche Anwendungsszenarien für ein derartiges Gerät zu überlegen. Wo, wann und wie könnte es eingesetzt werden? Dazu sollte eine Demonstrationsversion von dem gesamten System implementiert werden. Dafür existiert bereits Code, der die Nutzung des HTML Viewer Controls demonstriert. Was noch zu tun verbleibt, ist die Implementierung des Backends durch ein Java Servlet und Integration dieser Komponenten.

Ein sekundäres und persönliches Ziel dieser Arbeit ist, die mündliche und schriftliche Darstellung eines Projektes auf Deutsch zu üben.

Die Darstellungskomponente

Die zentrale Komponente des Backendes ist das Java Servlet, das mit dem portablen RFID Leser kommuniziert. Die Aufgabe des Servlets ist, als eine Zwischenstufe für den RFID Leser und Internet oder die Datenbank zu agieren. Das Servlet wird entweder mit einer RFID Identifikationsnummer oder einen URL durch einen HTTP GET Request aufgerufen, und holt eine entsprechende Seite.

Die andere Komponente zur Darstellung der Informationen ist der HTML Viewer, der in dem Handgerät sitzt. Die Applikation bekommt 48 Bytes Indaten, entsprechend dem Informationsinhalt der Marke, vom RFID-Leser durch den COM-Anschluss. Diese Daten werden transformiert zu einem HTTP GET Request der zu dem Servlet durch eine WLAN Verbindung geschickt wird.

Das Java Servlet kann auf einem PC oder einer UNIX Station irgendwo im Internet laufen, in meinem Demonstration habe ich rif33 unter Solaris benutzt. Der HTTP Server, oder Webserver, ist Tomcat 3.2.1 von Apache.

Erweiterung der Aufgabe mit XML-Support

Ich habe in dieser Architektur vorausgesetzt, dass die Plattform von der Client schon bekannt ist. Eine Erweiterung der Aufgabe wäre, der Ouput von Server unabhängig von Clienttyp zu machen. Der Verlauf würde dann wie folgt aussehen:

- Das Servlet wird mit einer RFID Identifikationsnummer durch einen HTTP GET Request aufgerufen. Der Request wird als ein AQL (Access Query Language) Anfrage zur Datenbank gesendet (diese Query Language ist spezifisch für OMS).
- Wenn das Servlet mit einer Identifikationsnummer aufgerufen wird, die keine virtuelle Gegenstücke in der Datenbank haben, wird die Datenbank nicht involviert. Dann schickt das Servlet stattdessen einen GET Request zum entsprechenden Webserver auf dem Internet. In diesem Fall werden Daten immer in HTML Format retourniert, und keine XSLT (eXtensible Stylesheet Language Transformer) Transformation ist notwendig.
- Wenn die Identifikationsnummer schon in Datenbank existiert, wird die Antwort einen Text String in XML-Format (eXtensible Markup Language).
- Die XML Datei wird zu einem HTML Datei transformiert [1].

Die letzte Transformation geschieht mittels einer XSL (eXtensible Stylesheet Language) Datei¹. Diese Datei gibt Regeln an welche die beschreiben die Präsentation (das Layout). In der XML Datei sind die tatsächlichen Daten (Text, Bilder, Sound etc.) gespeichert.

Es ist also diese Transformation die beliebige passende Formate als Output zu generieren ermöglicht. Die Voraussetzung ist, dass die Daten schon in Datenbank in XML Format

¹ Einer CSS (Cascading Style Sheet) Datei wäre auch möglich

gespeichert sind, und dass eine spezifische XSL Datei für jeden möglichen Clienttyp für den XSLT Prozessor (in diesem Fall Xalan 2.1.0 [6]) in dem Webserver vorhanden ist.

Als Datenbank habe ich das System OMS Pro 1.3 gewählt, weil ich mich damit ein bisschen auskenne, und das System den Vorteil hat, Daten in XML-Format zurückzugeben. Ein anderes Datenbanksystem wie zum Beispiel Oracle oder Hypersonic hätte sicher mindestens gleich gut sein können. Der Database Server gehört eigentlich zu dem Servlet, und stellt die Schnittstelle zur Datenbank dar.

Die Systemarchitektur

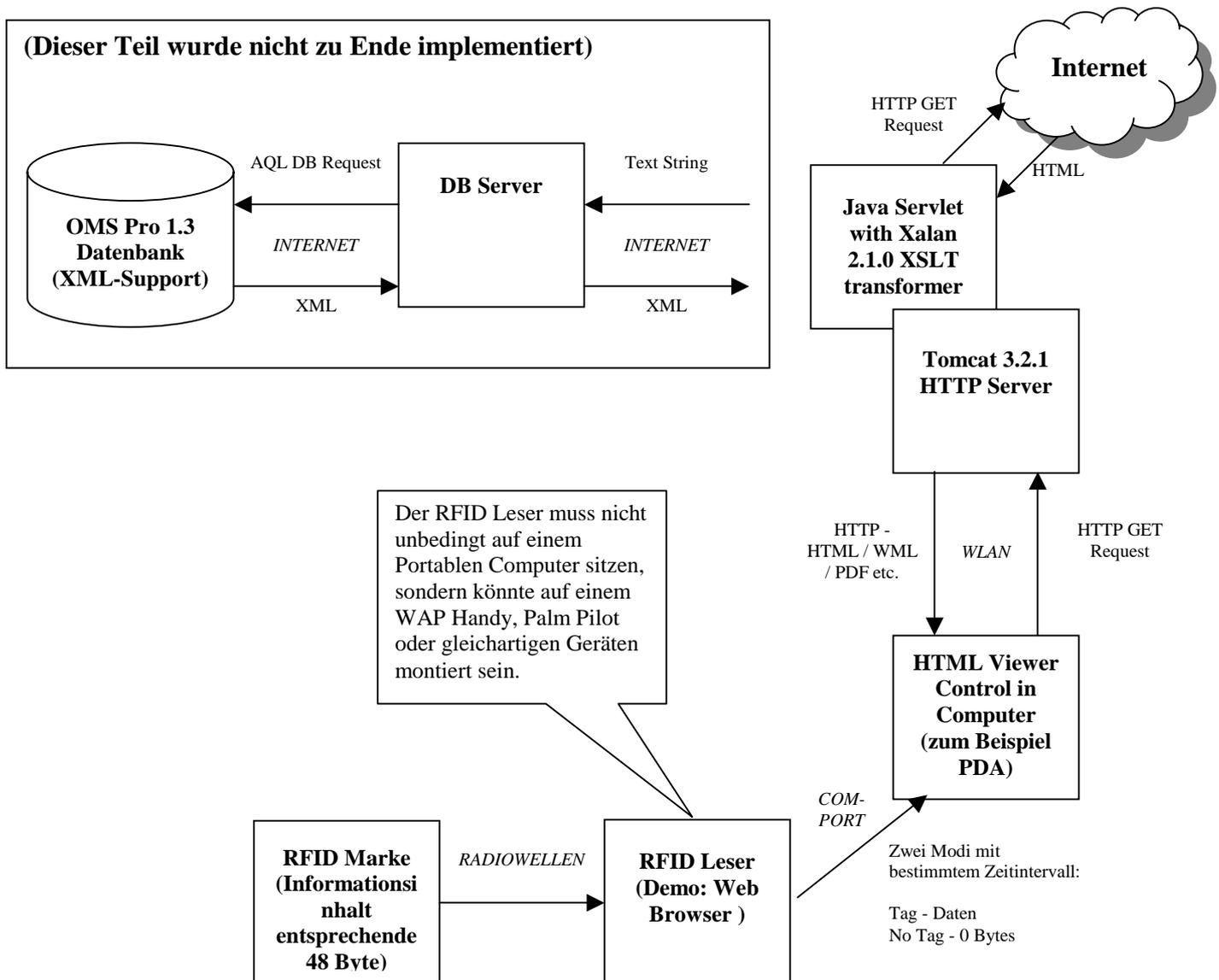


Abbildung 1 Die Systemarchitektur

Industrielle Barcodeleser

Barcode ist eine alte Technik, um Daten mit Produkten zu verknüpfen, die sogenannte automatische Identifikation. Ein Barcode hat den großen Vorteil, dass man ihn schnell und einfach drucken und ablesen kann. Die Entwicklung dieser Leser geht heute sehr schnell voran, und es gibt schon Modelle die auf dem Handy montierbar sind (Abbildung 2). Das Handy kann über WAP (Wireless Application Protocol) mit der Datenbank kommunizieren.



Abbildung 2
Barcodeleser von

Ein Barcode repräsentiert einen Text String, der normalerweise unter dem Barcode zu lesen ist. Dieser String ist auch der Output vom Leser (oder Scanner), wenn er den Barcode liest. Der Leser besteht aus einem Laser, einem Rezeptor und einem Mikrocomputer. Der Laser sendet einen Strahl ab, welchen der Barcode reflektiert, abhängig davon wie breit die verschiedenen Linien sind. Der Mikrocomputer analysiert das Resultat und produziert einen Code daraus.

Eine andere Technologie mit ähnlichen Eigenschaften sind die RFID Marken, die nicht statisch, sondern mittels eines RFID Schreibers programmierbar sind, obwohl sie heute nur relativ begrenzten Speicherplatz enthalten. Ein Nachteil ist, dass sie einen RFID Leser erfordern. Ich werde dieses Thema weiter unten in einem separaten Abschnitt beschreiben.

Existierende Arbeitsgebiete

Das Unternehmen AirClic Inc. [2] in Stockholm (ehemalige Connect Things AB) hat eine ähnliche Lösung wie meine entwickelt. Ihre Lösung baut auf traditionellen Barcodes auf, die von einem Handy oder PDA gelesen werden. AirClic führt ein öffentliches Register der Barcodes mit URLs verbindet. Das Unternehmen verdient Geld durch Gebühren von den Werbeanbietern, die durch Zugriffe auf das Register erhoben werden.

Die Software in diesem System integriert den Leser mit Windows, und wenn ein Barcode gelesen wird öffnet sich automatisch ein Web Browser. Die Software unterstützt viele verschiedene Barcode Leser, nicht nur den von AirClic. Der Benutzer kann die Software auch ohne Leser benutzen, indem er die Barcodenummer auf AirClics Webseite numerisch (als String) eingibt. (Abbildung 3)



Abbildung 3 Web-
Schnittstelle zur
AirClic Datenbank

Die Lösung von AirClic bietet also die folgende Funktionen:

- Mobiler Zugriff auf Applikationen
- Eine sehr umfangreiche zentrale Datenbank
- Einfache Schnittstelle
- Software in Windows integriert

Dazu kann man die Hardwareteile in Abbildung 2 einfach an alle neue Handymodelle von Ericsson montieren. Es gibt auch ein Stand-Alone Version namens „AirClicker“ der sich zum Computer oder PDA durch Kabel anschließen lässt.

In Tageszeitungen

Ein mögliches Szenario ist, mittels Barcode im Telefonbuch auf die entsprechende Website zuzugreifen. Auf der Website kann man zwischen verschiedenen Produkten wählen und entsprechende Preise und Lieferungszeit auslesen. Für diesen Zweck sind Barcodeleser an Handys [2] besonders geeignet. (Abbildung 4)



Abbildung 4 Barcode in der Zeitung

Auf Lebensmittelpackungen

Das Unternehmen Beeline Shopper [4] hat ein anderes Verwendungsgebiet gefunden; sie wollen, dass Kunden ihre Barcodeleser zum Einkaufen benutzen können. Der Nutzen für die Kunden ist günstiger, schneller und gesünder einkaufen können. Wenn die Kunden den Barcode für eine Ware lesen, erhalten sie, wenn möglich, Vorschläge für alternative Produkte die die erwünschten Eigenschaften besser erfüllen. Zum Beispiel könnte der Kunde einen Diätplan angeben, und eine Auswertung und Abgleichung dieses Plans mit der Inhaltsstoffen jedes spezifische Produkts bekommen.

Es ist auch möglich, Informationen über Sonderangebote und „weekly specials“ in einem besonderen Geschäft zu erhalten. Erinnerungen an gescannte Coupons, Zugriff auf Datenbanken mit den Rezepten von allen möglichen Gerichten passend zu den Lebensmitteln auf der Liste und schlussendlich personalisierte Einkaufslisten, mit Coupons eingeschlossen, wenn vorliegend, nach bestimmten Bedingungen sind andere Anwendungsgebiete.



Abbildung 5 Barcode auf dem Lebensmittel

Heutet bietet Beeline nur eine Lösung für stationäre Computer an. Zusammen mit dem am Handy montierten Barcodeleser von AirClic wäre es aber möglich, direkt beim Einkauf die oben erwähnten Fähigkeiten zu ermöglichen. Diese Lösung setzt aber voraus, dass der Kunde Zugriff zu einem WAP-Handy hat.

In der Logistik

Beim Cern in Genf ist man gerade daran, einen neuen Teilchenbeschleuniger zu bauen (LHC, Large Houdron Collider). Dieser besteht aus über 100 Millionen verschiedenen Bauteilen aus der ganzen Welt. Um alle Teile organisieren und katalogisieren zu können, wird ein Datenbanksystem namens MP5 verwendet. Der Abruf von Daten eines Produktes an seinem Standort soll möglich sein, deshalb ist ein mobiler Datenbankzugriff geplant. Seine wichtigsten Funktionen sind das Lesen von spezifischer Bauteilinformation und das Schreiben von notwendigen Daten, wie Zustand und Ortsangabe.

Eine Auswertung, die Anfang dieses Jahres durch eine Diplomarbeit von Lars Roth [5] gemacht wurde, hat als Resultat ergeben, dass eine Lösung, die auf traditionellen Barcodes aufbaut, einer zentralen Datenbank, XML und Zugriff auf diese Datenbank durch Webbrowser in mobilen Terminals, sogenannte „thin clients“, am günstigsten für ein Projekt dieser Größe wäre.

Neue Szenarien für die Symbolische Lupe

Oben wurden einige interessante Szenarien erwähnt. Es gibt dies bezüglich noch viele andere existierende Arbeitsgebiete. Es scheint, dass fast nur die Phantasie die Grenzen von möglichen Applikationen setzt. Nun stellt sich die Frage, was man mittels eines RFID Lesers besser machen könnte.

Eigenschaft	Barcode	RFID Marke
Preis	Tief	Hoch
Lebenszeit	Kurz	Lang
Empfindlichkeit	Ziemlich empfindlich	Relativ robust
Informationsinhalt	Statisch, nur lesen, nur wenige Bytes Speicherplatz	Lese- und Schreibmöglichkeiten, bis ca. 100 Bytes
Sicherheit	Jeder kann es lesen	Jeder kann lesen, aber schreiben kann gesperrt werden
Lesanforderungen	Es dürfe keine visuelle Hindernisse zwischen Leser und Marke vorhanden sein	Es dürfe keine Metalle zwischen Leser und Marke vorhanden sein
Andere Eigenschaften	Möglich zu reproduzieren ohne zusätzliche Hardware	Braucht mehr Platz und wiegt mehr als einen Barcode

Tabelle 1 Vergleich Barcode vs. RFID Marke

Verbesserung des Logistikszenarios

Im vorigen Abschnitt wurde ein existierendes Logistikszenario beschrieben. Ein mögliches Szenario wäre, eine solche logistische Ablaufverfolgung mittels Transpondern zu verbessern. Güter könnten mit Transpondern versehen werden, und überall wo die Güter durchlaufen, würde es möglich sein, die Marke zu lesen. Mit Hilfe einer zentralen Datenbank könnte man dadurch die Güter den Reiseweg entlang verfolgen. Es wäre auch möglich direkt in die RFID Marke hineinzuschreiben, welche Ort der gerade durchgelaufen wurde, und wann. Diese Information könnte, wenn nötig, vor Wiederbeschreiben geschützt werden.



Abbildung 6 Transponders

Verbesserung des Verpackungsszenarios

Künftig wird es vielleicht möglich, jede einzelne Verpackung in einem Laden mit einer RFID Marke zu versehen. Das setzt natürlich voraus, dass der Preis für RFID Marken für diesen Zweck noch relativ stark sinken muss um interessant zu werden. Die Größe der Marke muss auch vermindert werden. Die Verpackungen mit den RFID Tags sollten natürlich wiederverwertbar sein, so dass der Preis und die Umweltbelastung minimiert werden.

Der Nutzen ist, unter anderen, dass die Kunden Zeit sparen, weil sie nur mit den Verpackungen an einem RFID Leser vorbeigehen müssen. Die Gesamtkosten werden automatisch ausgerechnet, und der Kunde könnte mittels Kreditkarte in einem Automat bezahlen. Es wäre nicht mehr nötig, dass ein Verkäufer oder eine Verkäuferin anwesend ist.

Andere Vorteile wären, dass man schnell eine neue Identifikationsnummer oder Daten zum Produkt geben könnte. Zum Beispiel könnten Frischprodukte im Migros Supermarkt nach ein paar Tagen im Schrank automatisch einen tieferen Preis zugeteilt bekommen.

Nachdem ich mit Kronegger Informationssystem AG [3] gesprochen habe, scheint es leider nicht voraussichtlich, dass die Marke in der nahen Zukunft kräftig in Preis sinken werden. Deshalb ist dieses Szenario wahrscheinlich erst im langen Sicht möglich.

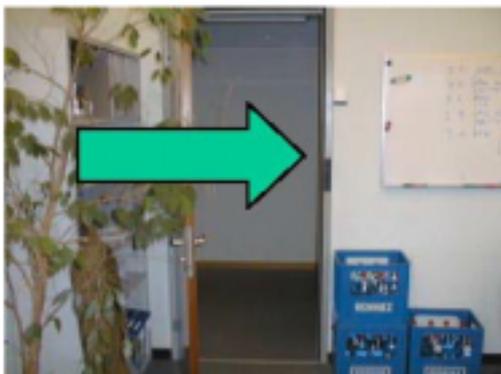
Gebäudeorientierungsszenario

Die Orientierung innerhalb eines Gebäudes könnte mit Hilfe von RFID Tags gemacht werden. Die Person wählt oder erhält eine Startposition, und bekommt von dort auf ihrem PDA-Bildschirm einem graphischen Hinweis in welche Richtung sie laufen sollte. Der PDA ist mit einem lokalen Netzwerk durch WLAN verbunden (künftig wäre auch UMTS möglich). Den Weg entlang sind RFID-Tags angehängt, und jedes mal, wenn die Person an einem solchen Tag vorbeiläuft, wird ein neues Bild auf ihrem PDA angezeigt.

Ein solches Experiment gibt es tatsächlich schon an der ETH Zürich, aber mit Barcodes. Der Unterschied ist, dass der Benutzer den PDA explizit an der Barcode richten muss, wenn er vorbeiläuft. Der Vorteil mit RFID Tags statt Barcodes wäre, dass es genug ist, wenn der Benutzer sich innerhalb einer Distanz von ungefähr einem Meter vom Tag entfernt befindet.

Dieses Szenario könnte auch auf eine Stadt erweitert werden. Der Benutzer könnte zum Beispiel eingeben, welche Sehenswürdigkeiten er besuchen möchte, oder welche Produkte er einkaufen will. Das Handy oder der PDA kann dann mit einer zentralen Datenbank kommunizieren und den kürzesten Weg zu diesem Ort herausfinden. Das setzt natürlich voraus, dass der Benutzer schon einen oder mehrere RFID Tags passiert hat und dass dadurch seine Position ungefähr bekannt ist. Weil RFID Marken heute verhältnismäßig billig sind, sollte es nicht unmöglich sein, solche Tags mit ziemlich geringem Abstand in der Stadt zu platzieren.

Von hier zur Informatikbibliothek



75 Schritte

[Ziel ändern](#) [Übersichtsplan](#)

RFID Leser und Marker

Weil die meisten portable Datensammelungsgeräte nicht von Anfang an für RFID Leser gestaltet wurden, gibt es selten Platz für die Montierung derartiger Geräte. Künftig wird das Problem nur schlimmer, weil die Elektronik dazu tendiert dauernd kompakter zu werden. Deshalb sind Miniaturleser entwickelt worden, und genau einen solchen habe ich in meiner Arbeit verwendet. Der Leser kommt von Kronegger Informationssysteme AG [3], und heißt Microengine. Dieses Gerät kann über eine serielle Schnittstelle einfach am Computer angeschlossen werden.

Der Leser hat die Masse 25 x 30 mm, und Stromverbrauch liegt bei nur bei etwa 120 mA wegen begrenzter Batteriekapazität. Die Frequenzen, die der Hersteller bietet, sind entweder 125 kHz oder 13.56 MHz. Der Leser besteht aus einer Antenne und einem elektronischen Behälter. Die Konstruktion ist so gemacht, dass sie im größten möglichen Umfang unabhängig von künftigen Veränderungen von den Transpondern sein sollte. Die Unterschiede zwischen 125 kHz und 13.56 MHz können folgendermaßen dargestellt werden:

Eigenschaft	125 kHz	13.56 MHz
Preis pro Marke	Mittlere Preis	Mittlere Preis
Datenübertragungsgeschwindigkeit	8 Kilobyte in 100 ms	80 Kilobyte in 10 ms
Rauschempfindlichkeit	Ziemlich hoch	Tieferer, weil Elektrorauschen normalerweise mit der Frequenz sinkt
Lesanforderungen	Unempfindlich	Empfindlich

Tabelle 2 Vergleich zwischen 125 kHz und 13.56 MHz Leser

Die Kommunikation zwischen Leser und Transponder geschieht über ein induziertes magnetisches Feld. Der Transponder besteht aus einem Chip mit einem Datenspeicher für Identifikationswert, eine Antennenrolle (Kupperleitung) und Verpackung. Der Transponder enthält keine Batterie, Energie wird durch das magnetische Feld geliefert.

Die Konstruktion ist sehr robust, und es ist nicht wahrscheinlich, dass Daten während des Transportes verloren gehen.



Abbildung 7 Transponder

Fazit

Ich habe etwa 60 bis 70 Stunden Arbeit in diesem Projekt investiert. Hauptsächlich habe ich während dieser Zeit programmiert, und über RFID Marke und Marker, OMS und Servlets gelesen. Das Resultat und Erfahrungen von dieser Arbeit werden hier unten zusammengefasst.

Schwierigkeiten

Unter der Implementationsphase habe ich vor allem Problem mit den Erweiterung des Projekts gehabt. Es war nicht so schwierig zu herausfinden wie ein Servlet funktioniert, und das HTTP Protokoll und dessen Headers sind einigermaßen erfassbar. In dem Erweiterungsteil hat mir vor allem OMS Mühe gemacht und wie man XML zusammen mit XSL transformieren lassen.

Die Integration des Servlets mit dem HTML Viewer Controller in der Lupe ging relativ problemlos. Weil mein Assistent diesen Teil programmiert hatte, hatte ich aber vor der Integration nicht genau im Detail verstanden wie es funktionierte.

Die deutsche Sprache hat mir während der Rapportphase ab und zu Mühe gemacht, und ich habe während der Arbeit sehr viele neue Wörter gelernt, und auch schreiben trainiert. Der kleine Vortrag war auch nicht einfach, trotz viel Training war es schwierig, die richtigen Formulierungen zu finden.

Vorteile meiner Lösung

Meine vorgeschlagene Lösung bietet die gleichen Vorteile wie diejenige von AirClic, zum Beispiel mobiler Zugriff, einfache Schnittstelle und schnelle Integration mit existierenden Betriebssystemen durch Java Technologie. Dazu gehören die folgenden Vorteile:

- Programmierbare RFID Marke, die größere Möglichkeiten für zum Beispiel Transportgüter bieten.
- Plattformunabhängigen Output vom Servlet durch die Verwendung von XML. Für jeden neuen Clienttyp muss man nur eine XSL-Datei programmieren.
- Mit einer eigenen Datenbank muss nicht der Anwender sich auf den Lieferer verlassen, sondern kann es selber administrieren. Sollte der Bedarf an einer generellen Datenbank mit einem vollständigen Verzeichnis von Barcodes oder RFID Identifikationsnummern entstehen, ist es einfach, die Software dafür anzupassen.

Danken

Schlussendlich möchte ich meinem Assistenten Michael Rohs für seine Geduld und Hilfsbereitschaft danken. Michael hat den Quellcode für den HTML Viewer Control für dieses Projekt zur Verfügung gestellt, und dazu auch meinen Bericht von sprachlichen Fehlern korrigiert.

Literaturverzeichnis

- [1] Über XML, XSL und XSLT, <http://www.w3c.org/>
- [2] AirClic Inc. <http://www.airclic.com/>
- [3] Kronegger Informationssysteme AG, <http://www.kronegger.com/>
- [4] BeeLine Shopper Inc. <http://www.beelineshopper.com/>
- [5] Lars Roth, Diplomarbeit, “Strategies for Mobile Services in the Engineering Data Management System at CERN”, http://edmsoraweb.cern.ch:8001/cedar/doc.info?document_id=306075&version=1.
- [6] Xalan Java version 2.1.0 <http://xml.apache.org/xalan-j/index.html>