Arbeitsbericht Nr. 15/2006
Hrsg.: Matthias Schumann

Thomas Diekmann / Svenja Hagenhoff

Ubiquitous Computing-Technologien zur Integration der realen Welt in betriebliche Informationssysteme
## Inhaltsverzeichnis

<table>
<thead>
<tr>
<th>Kapitel</th>
<th>Titel</th>
<th>Seite</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Einleitung</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Ausgewählte Ubiquitous Computing-Technologien für den Einsatz im betrieblichen Umfeld</td>
<td>2</td>
</tr>
<tr>
<td>2.1</td>
<td>Embedded Devices</td>
<td>2</td>
</tr>
<tr>
<td>2.2</td>
<td>RFID</td>
<td>6</td>
</tr>
<tr>
<td>2.3</td>
<td>Basisfunktionalitäten von Embedded Devices und RFID</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>Technische Überlegungen zur Integration von Ubiquitous Computing-Technologien in betriebliche Informationssysteme</td>
<td>14</td>
</tr>
<tr>
<td>3.1</td>
<td>Datenorientierte Integration von Ubiquitous Computing-Technologien</td>
<td>14</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Speicherung in Netzwerken</td>
<td>16</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Objektbegleitender Datentransport</td>
<td>19</td>
</tr>
<tr>
<td>3.1.3</td>
<td>Integrierte Sichtweise</td>
<td>23</td>
</tr>
<tr>
<td>3.1.4</td>
<td>Zusammenfassung und Beurteilung</td>
<td>25</td>
</tr>
<tr>
<td>3.2</td>
<td>Funktionsorientierte Integration von Ubiquitous Computing-Technologien</td>
<td>25</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Web Services zur Integration von Embedded Devices</td>
<td>27</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Eignung ausgewählter Web Service Standards</td>
<td>29</td>
</tr>
<tr>
<td>3.2.2.1</td>
<td>SOAP</td>
<td>29</td>
</tr>
<tr>
<td>3.2.2.2</td>
<td>WSDL</td>
<td>30</td>
</tr>
<tr>
<td>3.2.2.3</td>
<td>UDDI</td>
<td>31</td>
</tr>
<tr>
<td>3.2.2.4</td>
<td>BPEL4WS</td>
<td>32</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Zusammenfassung und Beurteilung</td>
<td>32</td>
</tr>
<tr>
<td>4</td>
<td>Zusammenfassung und Ausblick</td>
<td>35</td>
</tr>
<tr>
<td>Literaturverzeichnis</td>
<td>36</td>
<td></td>
</tr>
</tbody>
</table>
Abbildungsverzeichnis

Abbildung 2-1: Grundaufbau eines Embedded Device ............................................................4
Abbildung 2-2: Meilensteine der RFID-Entwicklung .................................................................6
Abbildung 2-3: Grundaufbau eines RFID-Systems .................................................................8
Abbildung 2-4: Aufbau des EPC .............................................................................................10
Abbildung 2-5: Taxonomie für Embedded Devices .................................................................11
Abbildung 2-6: Basisfunktionalitäten von RFID und Embedded Devices ..............................13
Abbildung 3-1: Abbildung der realen Welt in ein Informationsystem ......................................15
Abbildung 3-2: EPC-Infrastruktur ..........................................................................................18
Abbildung 3-3: Architektur des objektbegleitenden Datentransports ...................................22
Abbildung 3-4: Architektur zur Integration von Daten aus Ubiquitous Computing-Systemen 24
Abbildung 3-5: Vorgehen bei der Entwicklung einer SOA ......................................................26
Abbildung 3-6: Einordnung von Web Service Standards in die WSA ....................................29
Abbildung 3-7: Integration von Embedded Devices in eine SOA mittels Web Services ........33
### Abkürzungsverzeichnis

<table>
<thead>
<tr>
<th>Abkürzung</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALE</td>
<td>Application Level Events</td>
</tr>
<tr>
<td>BPEL4WS</td>
<td>Business Process Execution Language for Web Services</td>
</tr>
<tr>
<td>CGI</td>
<td>Common Gateway Interface</td>
</tr>
<tr>
<td>DARPA</td>
<td>Defense Advanced Research Projects Agency</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name System</td>
</tr>
<tr>
<td>EAN</td>
<td>European Article Number</td>
</tr>
<tr>
<td>EEPROM</td>
<td>Electrically Erasable Programmable Read-Only Memory</td>
</tr>
<tr>
<td>EPC</td>
<td>Electronic Product Code</td>
</tr>
<tr>
<td>ESP</td>
<td>Elektronisches Stabilitätsprogramm</td>
</tr>
<tr>
<td>FRAM</td>
<td>Ferroelectric Random Access Memorie</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>HTML</td>
<td>HyperText Markup Language</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IrDA</td>
<td>Infrared Data Association</td>
</tr>
<tr>
<td>ONS</td>
<td>Object Naming Service</td>
</tr>
<tr>
<td>PAN</td>
<td>Personal Area Network</td>
</tr>
<tr>
<td>RFID</td>
<td>Radio Frequency Identification</td>
</tr>
<tr>
<td>SMTP</td>
<td>Simple Mail Transfer Protocol</td>
</tr>
<tr>
<td>SOA</td>
<td>Service Oriented Architecture</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>SRAM</td>
<td>Static random access memory</td>
</tr>
<tr>
<td>UDDI</td>
<td>Universal Description, Discovery and Integration</td>
</tr>
<tr>
<td>UPC</td>
<td>Universal Product Code</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
<tr>
<td>WSA</td>
<td>Web Service Architecture</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Service Description Language</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
1 Einleitung


Den Abschluss der Untersuchung bilden eine Zusammenfassung der Ergebnisse der Untersuchung und ein Ausblick auf zukünftige Forschungsfragen.

Ausgewählte Ubiquitous Computing-Technologien für den Einsatz im betrieblichen Umfeld


2.1 Embedded Devices

Vernetzung der Embedded Devices untereinander erfolgen kann. Als besonders geeignete Technologien werden in der Literatur der Bluetooth- und der IrDA-Standard angeführt, die eine Vernetzung über relativ kleine Distanzen (< 1m bei IrDA und 10 bzw. 100 m bei Bluetooth) ermöglichen. Als besonderer Vorteil werden die große Verfügbarkeit, die geringen Kosten, die kleine Baugröße und der vergleichsweise geringe Energieverbrauch genannt. Für Vernetzungen über größere Distanzen eignen sich die verschiedenen IEEE 802.11x-WLAN-Standards, die allerdings einen relativ hohen Energieverbrauch haben (vgl. Want/Borriello/Farkas 2002, S. 37 f.).


Abbildung 2-1 fasst die grundlegenden Komponenten eines Embedded Device zusammen.

- **Preis:** Da Embedded Devices massenhaft eingesetzt werden sollen, dürfen diese nicht zu teuer sein.
- **Größe:** Um Embedded Devices quasi unsichtbar in die Umwelt integrieren zu können, müssen sie sehr klein sein.
- **Energieversorgung:** Aufgrund der inhärenten Mobilität von Embedded Devices, müssen sie über eine autarke Stromversorgung verfügen. Herkömmliche Energiespeicher (Batterien etc.) sind wegen ihrer geringen Energiedichte i. d. R. nicht geeignet.


- „Static Maxi Devices“ sind ortsgebunden und hinsichtlich der Rechenleistung und der Stromversorgung nicht eingeschränkt. Es handelt sich bei diesen Geräten beispielsweise um Server oder aber auch um andere Infrastruktur wie Bildschirme.
2.2 RFID


Abbildung 2-2: Meilensteine der RFID-Entwicklung


- Optical Character Recognition-Verfahren
- Biometrik
- Chipkarten
- Barcode-Systeme


In RFID-Systemen werden verschiedene Frequenzspektren verwendet, die zumeist weltweit oder regional lizenzfrei genutzt werden können. Das verwendete Frequenzspektrum ist eines der Hauptdeterminanten für die Reichweite der Funkübertragung. Darüber hinaus schränkt die zumeist gesetzlich regulierte Sendeleistung die Reichweite ein. Tabelle 2-1 fasst die wichtigsten Frequenzen, ihre Anwendung und die typische Reichweite zusammen (in Anlehnung an Bitkom 2005, S. 14 und Lampe/Flörkemeier/Haller 2005, S. 78).

<table>
<thead>
<tr>
<th>Frequenz</th>
<th>Anwendung</th>
<th>typische Reichweite</th>
</tr>
</thead>
<tbody>
<tr>
<td>135 kHz (Niederfrequenz, LF)</td>
<td>weltweit standardisierte und freigegebene Frequenz, die zumeist für passive RFID-Tags zur Tieridentifikation genutzt wird</td>
<td>&lt; 1,5 m</td>
</tr>
<tr>
<td>13,56 MHz (Hochfrequenz, HF)</td>
<td>weltweit standardisierte und freigegebene Frequenz, die für passive RFID-Tags zur Identifikation von einzelnen Objekten genutzt wird</td>
<td>&lt; 1,0 m</td>
</tr>
</tbody>
</table>
| 868 MHz (Ultrahochfrequenz, UHF) | in Europa standardisierte und freigegebene Frequenz für active und passive RFID-Tags in der Logistik                                                                                                   | < 3 m (bei erlaubten 0,4 Watt Sendeleistung)  
3-5 m (bei (geplanten) 2 Watt Sendeleistung) |
| 915 MHz (Ultrahochfrequenz, UHF) | analog verwendete Frequenz in den USA. RFID-Tags unterstützen i. d. R. beide Frequenzen um globale Logistikantwendung zu ermöglichen                                                                    | 5-7 m (bei erlaubten 4 Watt Sendeleistung) |
| 2,45 GHz (Mikrowelle, MW) | weltweit freigegebenes lizenz- und anmeldefreies Frequenzband für Industrie, Wissenschaft und Gesundheit. Wird für active Transponder (GPS, Temperatursensoren) eingesetzt                                         | keine Angabe möglich                  |

Tabelle 2-1: RFID-Frequenzbereiche und ihre Anwendung


2.3 Basisfunktionalitäten von Embedded Devices und RFID

In Kapitel 2.1 wurde eine Systematisierung von Embedded Devices in drei Gerätekategorien vorgestellt, die auf die Gerätemerkmale Größe, Energieversorgung und Mobilität beruht. Wie die Ausführungen zu Embedded Devices aber gezeigt haben, resultiert die angesprochene Gerätediversifikation aus wesentlich mehr Gerätemerkmale. Im Folgenden soll eine Taxonomie vorgestellt werden, die eine Systematisierung, die wesentlich mehr Gerätemerkmale berücksichtigt, ermöglichen soll.

Die Gerätemerkmale ergeben sich direkt aus den Ausführungen in Kapitel 2.1 und sollen hier nicht noch einmal erläutert werden. Lediglich der Aspekt der eindeutigen Identifizierung eines Embedded Device wurde, wie in der gängigen Literatur zu Embedded Devices auch, stark vernachlässigt. Wie die Ausführungen zu RFID aber gezeigt haben, ist ein Identifikationsmerkmal ein sehr wesentlicher Aspekt. Verfügt ein Embedded Device über ein eindeutiges Identifikationsmerkmal, so kann auch das dazugehörige Objekt eindeutig identifiziert werden. In dem Klassifizierungsrahmen soll deshalb unterschieden werden, ob das Identifikationsmerkmal nur lokal, also innerhalb einer bestimmten Domäne oder global, wie bei dem im Kapitel 2.2 erläuterten EPC, gültig ist.
Abbildung 2-5: Taxonomie für Embedded Devices


- haben ein global eindeutiges Identifikationsmerkmal,
- werden mit physischen Objekt kombiniert,
- sind über eine kabellose Netzwerkschnittstelle erreichbar und
- werden induktiv (passive Transponder) oder intern (aktive Transponder) mit Energie versorgt.

Optional verfügen sie über einen veränderbaren Datenspeicher und Sensorik.

- automatische Identifikation
- Ortsverfolgung
- Überwachung
- Notifikation
- Aktion


Um diese Basisfunktionalitäten erfüllen zu können, müssen Embedded Devices über bestimmte Fähigkeiten verfügen. Für die Identifikationsfunktion muss die Möglichkeit bestehen, auf dem Embedded Device ein Identifikationsmerkmal zu speichern. Um ein Objekt orten zu können (Fremdortung), muss es über ein Identifikationsmerkmal und eine kabellose Netzwerkschnittstelle verfügen. Eine Überwachung von Umweltzuständen setzt Sensoren und ggf. Datenspeicherungskapazitäten voraus. Um Notifikationsnachrichten generieren zu können, muss ein Embedded Device über einen Datenspeicher für die Regeln und eine Datenverarbeitungskomponente für die eigentliche Generierung verfügen. Aktionen können über Aktuatoren durchgeführt werden. Abbildung 2-6 fasst die technischen Voraussetzungen der Basisfunktionalitäten zusammen.

Abbildung 2-6: Basisfunktionalitäten von RFID und Embedded Devices
3 Technische Überlegungen zur Integration von Ubiquitous Computing-Technologien in betriebliche Informationssysteme


3.1 Datenorientierte Integration von Ubiquitous Computing-Technologien


Durch die Speicherung von Daten in Informationssystemen entsteht ein abstraktes Modell der Realität. Objekte der Realwelt werden in Informationssystemen durch Symbole (Namen, Bezeichnungen) abstrahiert. Die Abstraktion führt dazu, dass die Realwelt nicht in ihrer Gesamtheit, sondern nur die relevanten Realitätsausschnitte gespeichert werden. Die offensichtliche Schwierigkeit besteht darin, dass die laufenden Veränderungen der Realwelt in dem abstrakten Modell abgebildet werden müssen,

![Abbildung 3-1: Abbildung der realen Welt in ein Informationsystem](image)


1. **Speicherung in Netzwerken (Data-on-Network):** Ubiquitous Computing-Technologien wie RFID und Embedded Devices erlauben eine Echtzeitabbildung der realen Welt in die digitale Welt (vgl. Heinrich 2005, S. 24 ff.). Aufgrund der hohen Datenvolumen werden die gesammelten Daten in dezentralen Datenbanken gespeichert und verarbeitet. Wie auf diese Daten
transparent in Netzwerken zugegriffen werden kann, soll am Beispiel der EPC-Infrastruktur im Kapitel 3.1.1 dargestellt werden.

2. Objektbegleitender Datentransport (Data-on-Tag): Die Grenzen zwischen der realen Welt und den digitalen Daten verschwimmen zunehmend. Die digitale Welt wird zunehmend in die reale Welt integriert, was Hinsichtlich der o. a. Datentransparenz verschiedene Implikationen mit sich zieht. So werden die Daten, die zur Bildung des abstrakten Modells im Informationssystem benötigt werden, nicht zwangsläufig „online“ gesammelt. Sie werden vielmehr am „Ort des realen Geschehens“ gesammelt, der nicht unbedingt in Reichweite von Netzwerken sein muss. Auch ist es nicht immer möglich die Änderungen des abstrakten Modells „online“ in die reale Welt zu übertragen. Die zur Änderung der realen Welt benötigten Daten müssen u. U. physisch an dem Ort, an dem die Aktion in der realen Welt durchgeführt werden soll, vorhanden sein. Es bietet sich also an die Daten an die Objekte, die am „Geschehen“ beteiligt sind, zu binden. Der sog. objektbegleitende Datentransport soll in Kapitel 3.1.2 dargestellt werden.

Den beiden angeführten Datenspeicherungskonzepten liegen völlig gegensätzliche Herangehensweisen zugrunde. Sie sind trotzdem nicht als konkurrierende, sondern als komplementäre Konzepte zu betrachten. In Kapitel 3.1.3 wird beschrieben, wie die beiden Konzepte kombiniert werden können.

3.1.1 Speicherung in Netzwerken


Um die gespeicherten Informationen über den EPC referenzieren zu können, wurde der sog. Object Naming Service (ONS) entwickelt. Der ONS ist sehr stark an den vom Internet bekannten Domain Name System (DNS) angelehnt. Eine ONS-Anfrage liefert zu einem EPC eine oder mehrere Internetadressen (URLs), unter der/den Informationen zu dem entsprechenden Objekt zu finden ist/sind. Der ONS basiert, somit auf den DNS. Neben den o. a. EPC Information Services können auch herkömmliche HTML-Seiten referenziert werden. Da der potenzielle Adressraum des EPC (96 Bit)

2 Zurzeit liegt noch keine endgültige Spezifikation vor.
wesentlich größer ist, als der Adressraum des Internets (32 Bit), waren bei der Entwicklung des ONS insbesondere Skalierungsfragen von großer Relevanz (vgl. Engels et al. 2001, S. 76 f.).

Abbildung 2-1 fasst die Komponenten der EPC-Infrastruktur zusammen (vgl. Flörkemeier 2005, S. 89).

Abbildung 3-2: EPC-Infrastruktur


den objektbezogenen Daten werden deshalb i. d. R. mit einem spezifischen Zugriffsmechanismus versehen werden. So kann über den EPC zwar die Datenbanken mit den entsprechenden Informationen gefunden werden, der Zugriff auf die Daten kann aber verwehrt werden. Sind diese Zugriffsmechanismen nicht einheitlich gestaltet, so wird die automatische Nutzung der objektbezogenen Daten erschwert. Es ist daher zu überlegen, ob es nicht sinnvoller ist, Daten am Objekt zu belassen und somit dem Besitzer des Objektes zentral zur Verfügung zu stellen (s. Kapitel 3.1.2).

### 3.1.2 Objektbegleitender Datentransport

In der Literatur zu RFID dominiert der Ansatz Daten - wie im vorherigen Abschnitt dargestellt - im Netzwerk zu speichern (vgl. Henriici/Müller 2004, S. 223 ff.). Zumeist wird dies damit begründet, dass die RFID-Tags möglichst einfach gehalten werden sollen, damit die Kosten für sie möglichst gering sind (vgl. z. B. Sarma 2004, S. 52). Es ist aber zu erwarten, dass die Technologie weiterreift und die Kosten für komplexere RFID-Tags mittelfristig fallen werden. Es ist also zu untersuchen, an welchen Stellen eine Speicherung von Daten, die über die reine Identifikation hinausgehen, auf dem RFID-Tag sinnvoll ist.


- Class 2-Transponder sind passiv und haben einen Speicher mit Sicherheitsfunktionen,
- Class 3-Transponder sind semiaktiv und verfügen ebenfalls über einen Speicher mit Sicherheitsfunktionen,
- in Class 4-Transponder sind zusätzlich Sensoren integriert.


Ein objektbegleitender Datentransport impliziert aber auch, dass die am Objekt gespeicherten Daten nur zur Verfügung stehen, wenn das korrespondierende Objekt greifbar, d. h. in Reichweite einer
Technische Überlegungen zur Integration von Ubiquitous Computing-Technologien in betriebliche Informationssysteme


- durch das Objekt erfasst werden, wenn sich das entsprechende Objekt außerhalb der Reichweite einer Netzwerkinfrastruktur befindet (Nutzdaten) oder
- die benötigt werden, um Aktionen, die das Objekt betreffen außerhalb der Reichweite einer Netzwerkinfrastruktur durchführen zu können (Steuerungsdaten).

Es stellt sich nun die Frage, wie die Daten in den zentralen Datenbeständen mit den zwischengespeicherten Daten konsistent gehalten werden können. Dazu soll kurz das Konzept des „Personal Servers“ vorgestellt werden, in dem eine sehr ähnliche Problemstellung vorliegt.


Grundsätzlich sind für die beiden o. a. Datenkategorien (Nutzdaten, Steuerungsdaten) folgenden Abgleichregeln anzuwenden:

- Daten, die ausschließlich vom Objekt gesammelt werden und später grundsätzlich nicht geändert werden, können von dem Objekt direkt in die zentralen Systeme übernommen werden.

- Temporäre Daten, die vom Objekt gesammelt werden, können aus dem objektbegleitenden Datenspeicher gelöscht werden (z.B. Fehlerdaten, die nach einer erfolgreichen Nacharbeit und Qualitätssicherung nicht mehr am Objekt benötigt werden).

- Daten, die benötigt werden, um Aktionen am Objekt durchführen zu können (Steuerungsdaten), basieren i. d. R. auf Entscheidungen, die auf Basis der zentralen Daten gefällt wurden. Die Steuerungsinformationen entstehen also in zentralen Systemen und sollten in die objektbegleitenden Datenspeicher übernommen werden.

Diese Abgleichregeln sind nicht als Manifeste, sondern vielmehr als Heuristiken zu verstehen; im Einzelfall kann es sinnvoll sein, von ihnen abzuweichen. Abbildung 3-3 fasst die Grundelemente der skizzierten Architektur des objektbegleitenden Datentransports zusammen.
3.1.3 Integrierte Sichtweise

Folgende Tabelle fasst die wesentlichen Charakteristika der beiden in den vorangegangenen Ausführungen dargestellten Speicherungsansätze zusammen (vgl. Tabelle 3-1).

<table>
<thead>
<tr>
<th></th>
<th>Speicherung in Netzwerken</th>
<th>objektbegleitender Datentransport</th>
</tr>
</thead>
<tbody>
<tr>
<td>Konzept</td>
<td>Abkopplung der Daten vom Objekt</td>
<td>Integration der Daten mit dem Objekt</td>
</tr>
<tr>
<td>Voraussetzung für Datenzugriff</td>
<td>Netzwerk-Infrastruktur</td>
<td>Präsenz des Objekts</td>
</tr>
<tr>
<td>Speicherort der Objekt-Daten</td>
<td>zentral (Datenbanken)</td>
<td>dezentral (Objekt)</td>
</tr>
<tr>
<td>Inhalt der Daten auf dem Tag</td>
<td>ID (EPC)</td>
<td>objektbezogene Daten</td>
</tr>
<tr>
<td>Art der Daten auf dem Tag</td>
<td>statisch</td>
<td>dynamisch</td>
</tr>
<tr>
<td>Zusatzfunktionalitäten</td>
<td>nicht möglich</td>
<td>möglich</td>
</tr>
<tr>
<td>Erforderliche Speicherkapazität auf dem Tag</td>
<td>gering</td>
<td>hoch</td>
</tr>
<tr>
<td>Transponderkosten</td>
<td>gering</td>
<td>hoch</td>
</tr>
<tr>
<td>Standards</td>
<td>EPC class 0 und 1</td>
<td>EPC class 2, 3, 4</td>
</tr>
<tr>
<td>Datensicherheit</td>
<td>Zugriffsmechanismen in DB</td>
<td>Verschlüsselung auf dem Tag</td>
</tr>
</tbody>
</table>

Tabelle 3-1: Charakteristika der Speicherungsansätze


In den aus der Literatur bekannten Architekturmodellen sollen einfache RFID-Transponder mit denen Objekte über den EPC eindeutig identifiziert werden können dazu beitragen, dass das Abbild der realen...
Welt in den Informationssystemen aktueller und genauer ist. Dazu müssen die Rohdaten der RFID-Lesegeräten zu aussagekräftigen Daten aggregiert werden und sie müssen über die Auflösung des EPC einem Datenobjekt in dem Informationssystem zugeordnet werden. Die aggregierten Daten können entweder in zentralen Systemen gespeichert werden oder direkt an die Geschäftslogikschicht weitergegeben werden.

Bei dem objektbegleitenden Datentransport werden die Nutzdaten quasi asynchron mittels eines Zwischenspeichers erhoben. Die Nutzdaten werden am Objekt gesammelt und sobald eine Netzwerkinfrastruktur in Reichweite ist, mittels einer Synchronisationskomponente in die zentralen Datenbestände übernommen.

Auf Basis der Daten, die in der zentralen Datenhaltung zur Verfügung stehen, können die Geschäftslogik bzw. die Entscheider Entscheidungen treffen, die in Steuerungsdaten resultieren. Die Steuerungsdaten werden an Terminals weitergeleitet, die beispielsweise papierbasierte Anweisungen generieren, Anweisungen am Bildschirm anzeigen oder Steuerungsdaten an Betriebsmittel (Maschinen etc.) weitergeben. Bei dem objektbegleitenden Datentransport können diese Steuerungsdaten am Objekt zwischengespeichert werden und stehen somit auch zur Verfügung, wenn eine Netzwerkinfrastruktur nicht zur Verfügung steht.

Werden komplexe Embedded Devices verwendet, so ist es auch denkbar, dass diese direkt mit der Geschäftslogik kommunizieren. Dieser Ansatz wird in Kapitel 3.2 dargestellt.

Abbildung 3-4: Architektur zur Integration von Daten aus Ubiquitous Computing-Systemen

3.1.4 Zusammenfassung und Beurteilung


3.2 Funktionsorientierte Integration von Ubiquitous Computing-Technologien


1. Identifizieren der Klassen und Objekte
2. Zuweisen von Attributen und Methoden
3. Identifizieren der Beziehungen zwischen Klassen und Objekten
4. Implementieren des Entwurfs
In dem Paradigma der Objektorientierung konzentriert man sich also vorwiegend auf die Frage aus welchen Objekten eine Anwendung zu bestehen hat. Dem gegenüber steht die prozessorientierte Sicht heutiger Unternehmen, die nicht an der Struktur des Unternehmens interessiert sind, sondern vielmehr daran wie es „funktioniert“. Die SOA soll das Bindeglied zwischen der objektorientierten Sicht der DV-Abteilungen und der prozessorientierten Sicht der Fachabteilungen sein. Sie baut auf die objektorientierte Architektur auf, indem sie den vorhandenen objektorientierten Komponenten die Möglichkeit gibt ihre Funktionalität zu veröffentlichren und Funktionen anderer Komponenten zu finden und zu nutzen. Funktionalitäten können als so genannte Services über Netzwerke jedermann, egal wo er sich befindet, angeboten werden. In den Services einer SOA wird in der Regel keine Funktionalität neu implementiert, sie fungieren vielmehr als Wrapper, die bei Aufruf die Funktionalität bereits bestehender Applikationen aufrufen, die Ergebnisse sammeln und schließlich zurückgeben (vgl. Alonso et al. 2004, S. 144). Die Applikationslogik, die die Services kapseln, muss dabei nicht zwangsläufig objektorientiert programmiert sein, es ist auch eine Kapselung bestehender Legacy-Systeme, wie sie noch in vielen Betrieben vorhanden sind, möglich.


![Abbildung 3-5: Vorgehen bei der Entwicklung einer SOA](image)

Das Konzept der SOA zeichnet sich also durch eine strikte Aufgabenteilung aus, bei der Geschäftsprozesse im Vordergrund stehen und die technischen Details immer mehr in den Hintergrund rücken. Der Top-Down-Ansatz soll garantieren, dass die Geschäftsprozesse die Services bestimmen und die Services die konkrete Implementierung. Man erhofft sich, dass dadurch das Business Process Reengineering flexibler und effizienter gestaltet werden kann (vgl. Burkhard/Laures 2003, S. 16 f.).
Erst in jüngster Zeit hat die Diskussion über die SOA mit der Einführung von Web Services als Technologie zur Umsetzung von SOAs einen neuen Aufschwung erfahren. Das World Wide Web Consortium (W3C) entwickelt zurzeit eine Web Service Architecture (WSA), die eine konkrete Umsetzung der SOA auf Basis von Web Services darstellt. Das W3C versteht unter der SOA „a specific type of distributed system in which the agents are „services“. […] a service is a software agent that performs some well-defined operation and can be invoked outside of the context of a larger application. […] the users of that server need to be concerned only with the interface description of the service. Furthermore, most definitions of SOA stress that “services” have a network-addressable interface and communicate via standard protocols and data formats“ (Booth et al. 2003).


3.2.1 Web Services zur Integration von Embedded Devices


- Minimierung von Speicheroperationen
- Minimierung von Speicherbedarf
- Effizientes XML-Parsen
- Plattformunabhängigkeit

Beachtet man diese Designkriterien spricht aus Sicht der Web Service Implementierung nichts gegen die Portierung auf Embedded Devices. Es sind mittlerweile auch schon Web Service Implementierungen für Smartphones und Personal Digital Assistants (PDAs) entwickelten worden, die den obigen Anforderungen genügen.


- lose Kopplung
- Plattform- und Programmiersprachenunabhängigkeit
- Offenheit bezüglich der Standards

Trotzdem bleibt zu zeigen, dass die Web Service Standards auch mit den Besonderheiten von Embedded Devices, wie temporäre Erreichbarkeit, wechselnde Netzwerkadressen zurechtkommen. In Bezugnahme auf die oben beschriebene SOA für betriebliche Anwendungssysteme wird dies im nächsten Abschnitt geschehen.

3.2.2 Eignung ausgewählter Web Service Standards

Die o. a. WSA definiert eine SOA, die auf XML basiert. Sie legt sich aber nicht auf bestimmte Standards wie beispielsweise SOAP oder WSDL fest. Trotzdem haben sich für die verschiedenen Problemstellungen des Service Models verschiedene Standards etabliert. Abbildung 3-6 ordnet die bekanntesten Standards in das Service Model der WSA ein.

Abbildung 3-6: Einordnung von Web Service Standards in die WSA

3.2.2.1 SOAP

Mit der Wahl von XML als Standard für die Datenstrukturierung, ist der Einsatz von generischen Parsern möglich, die die Kosten senken und eine höhere Robustheit gewährleisten. Es bedeutet aber auch, dass man einen hohen Performance-Overhead in Kauf nimmt, der mit dem Wunsch nach Generalität einhergeht. Aber gerade in ubiquitären Systemen ist die Performance ein kritischer Faktor, da sie zum einen durch die Baugröße und zum anderen durch die geringe Batteriekapazität von Embedded Devices eingeschränkt ist.


Zusammenfassend lässt sich feststellen, dass SOAP grundsätzlich für die Portierung auf Embedded Devices geeignet ist, es aber Probleme im Bereich der Performance und der Adressierung geben kann.

### 3.2.2.2 WSDL

Die Web Service Description Language (WSDL) ist vergleichbar mit Interface Definition Languages in konventionellen Middleware-Plattformen. Neben der einfachen Beschreibung der Service Schnittstelle (Service Name, Input- und Output-Parameter) wie sie konventionelle IDLs machen, muss WSDL beispielsweise auch Angaben zum Transportprotokoll machen, da die Nutzung von SOAP nicht an ein bestimmtes Protokoll gebunden ist (s. Kapitel 3.2.2.1). Auch die Adressierung des Services ist Teil eines WSDL-Dokuments und führt somit zu der schon bei SOAP beschriebenen Problemen mit Embedded Devices. Da es bei Web Services keine zentrale Instanz gibt, die wie bei konventionellen Middleware-Plattformen Interaktionsmuster für Web Service Aufrufe mit mehreren (evtl. sogar asynchronen) Nachrichtentransaktionen vorgibt, definiert WSDL eine Reihe von möglichen Interaktionsmustern.

Die Stärken und Schwächen von WSDL ähneln denen von SOAP. WSDL eignet sich einerseits durch seine große Flexibilität gut für Embedded Devices in betrieblichen Anwendungssystemen, andererseits gibt es Probleme im Bereich der Performance (da auch WSDL XML nutzt) und im Bereich der Adressierung.

3.2.2.3 UDDI


Bei Ubiquitous Computing Anwendungen kann die Beschreibung von Web Services noch wesentlich komplizierter werden. Wenn man zum Beispiel einen Web Service sucht, der etwas druckt, so möchte man nicht irgendeinen Web Service mit der gewünschten Funktionalität haben, sondern einen, der einen Drucker in der Nähe ansteuert. Die Ortstransparenz, die man bei herkömmlichen verteilten Systemen und somit auch bei Web Services als erklärtes Ziel hat, muss bei Ubiquitous Computing also aufgegeben werden.

Wenn man sich Szenarien vorstellt, in denen sich Gegenstände, die Ihre Ressourcen über Web-Service-Schnittstellen zur Verfügung stellen, spontan zur Bewältigung einer Aufgabe vernetzen sollen, so muss die Unterstützung durch UDDI-Verzeichnissen grundlegend erweitert werden. UDDI eignet sich in seiner jetzigen Form nur für Szenarien, die ex ante bekannt sind und somit die benötigten Web Services vor der Nutzung manuell gesucht werden können. Dies ist, um auf betriebliche Anwendungssysteme zurückzukommen, nur bei Geschäftsprozessen der Fall, die planbar und beschreibbar sind. Dies ist beispielsweise bei produktionsnahen Geschäftsprozessen oftmals der Fall. Bei Geschäftsprozessen, die die Ideenfindung beschreiben, hat man mit der Formalisierbarkeit in der Regel aber Probleme.
3.2.2.4 BPEL4WS


Verwendet man BPEL4WS für die Implementierung von betrieblichen Anwendungssystemen, so kann man analog zur Geschäftsprozessmodellierung verschiedene Abstraktionsniveaus für die Web Services verwenden. Komplexe Geschäftsprozesse können in kleine, handhabbare Geschäftsprozesse herunter gebrochen werden und kleine Geschäftsprozesse können zu komplexen Geschäftsprozesse zusammengefasst werden.

Es ist allerdings zu beachten, dass man die so gewonnene verbesserte Handhabbarkeit der Komplexität möglicherweise durch einen großen Performanceverlust bei der Ausführung erkauft, da ein erheblicher Kontroll-Overhead durch die Komposition entsteht.

Geht man davon aus, dass die anderen Standards (SOAP, WSDL, UDDI) für Embedded Devices in betrieblichen Anwendungssystemen geeignet sind oder entsprechend angepasst wurden, so ist BPEL4WS für ebenfalls uneingeschränkt einsetzbar.

3.2.3 Zusammenfassung und Beurteilung

3 Technische Überlegungen zur Integration von Ubiquitous Computing-Technologien in betriebliche Informationssysteme


### Technische Überlegungen zur Integration von Ubiquitous Computing-Technologien in betriebliche Informationssysteme

<table>
<thead>
<tr>
<th>Standard</th>
<th>Stärken</th>
<th>Schwächen</th>
</tr>
</thead>
</table>
| SOAP       | - Problem der Adressierung wird Transportprotokoll überlassen  
- Flexible Kommunikationspattern (synchron/asynchron)                                                                                     | Die bei SOAP berücksichtigten Transportprotokolle lösen das Adressierungsproblem z. T. nicht         |
| WSDL       | Unterstützt nahezu beliebige Interaktionsmuster                                                                                                                                                        | Keine Lösung des Adressierungsproblems                                                               |
| UDDI       | Ermöglicht die Katalogisierung der im Betrieb verfügbaren Web Services                                                                                                                                 | - Unzureichende Kategorisierungsmöglichkeiten  
- „Semantik-Problem“                                                                                                                            |
| BPEL4WS    | - Komplexitätsreduzierung  
- Wenn SOAP, WSDL und UDDI einsetzbar, dann ist auch BPEL4WS einsetzbar                                                                                                          | Erhöhter Kontroll-Overhead                                                                            |
| Alle       | - Plattform- und Programmiersprachen-unabhängig  
- Einfache Verarbeitung durch generische Parser, da XML-basiert                                                                                      | Durch die Verwendung von XML entstehen Performanceverluste                                             |

Tabelle 3-2: Stärken und Schwächen der Web Service-Standards

Insgesamt haben die Standards zwar Schwächen, grundsätzlich sind sie aber für die Portierung auf Embedded Devices geeignet. Die Verwendung von XML lässt eine plattform- und programiersprachenübergreifende Verwendung der Standards zu, was bei der hohen Diversifikation hinsichtlich Hard- und Software bei Embedded Devices einen entscheidenden Faktor darstellt.
4 Zusammenfassung und Ausblick


Die Untersuchung erfolgt insgesamt aus einer technik-orientierten Perspektive und auf einem sehr abstrakten Niveau. Um Unternehmen Ubiquitous Computing für den betrieblichen Einsatz näher zu bringen, muss die Darstellung des Beitragspotenzials dieser Technologien in verschiedenen Anwendungsgebieten konkreter erfolgen. Auch muss eine Bewertung des Einsatzes erfolgen, um der Gefahr die Technologie zum reinen Selbstzweck einzusetzen, entgegenzuwirken.
Literaturverzeichnis


