

**DIPLOMARBEIT**

# **PDA-INFO-SYSTEM**

Ausgeführt zum Zweck der Erlangung des akademischen Grades

**Dipl.-Ing. (FH) Telekommunikation und Medien**

am Fachhochschul-Diplomstudiengang Telekommunikation und Medien St. Pölten

unter der Erstbetreuung von

**FH-Prof. Dipl. Ing. Georg Barta**

Zweitbegutachtung von

**Mag. Georg Bernold**

ausgeführt von

Markus Grabner

Matrikelnummer: tm0310038018

Wien, am 3. September 2007

Markus Grabner

# Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Diplomarbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.
- ich dieses Diplomarbeitsthema bisher weder im Inland noch im Ausland einer Begutachterin/einem Begutachter zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Diese Arbeit stimmt mit der von den Begutachtern beurteilten Arbeit überein.

.....

Wien, am 3. September 2007

.....

Markus Grabner

## Vorwort

Kleine mobile Geräte werden sowohl im Geschäfts- als auch im Privatbereich immer bedeutender. Sie werden unter den Begriffen Handheld, Pocket PC, PDA, MDA oder Smart Device angeboten. Im privaten Bereich werden sie häufig als Navigationssystem genutzt, da sie vielfach schon eine GPS-Antenne und die GPS-Software integriert haben. Die Vorteile dieser Geräte liegen auf der Hand: Relative Unabhängigkeit von einer fixen Energieversorgung, klein, leicht tragbar und handhabbar. Infolge des immer flächendeckender werdenden Angebots an mobilen Breitband-Internetdiensten ist auch in Zukunft mit verstärkter Nachfrage an diesen Geräten sowohl im Privat- als auch im Business-Bereich zu rechnen. In naher Zukunft wird es sowohl Handhelds ohne Telefonfunktion (PDA) als auch solche mit Telefonfunktion (MDA) geben. Die derzeitige Situation ist, dass die Mehrheit der Kunden zusätzlich zum gewöhnlichen Mobiltelefon einen PDA verwendet. Mobiltelefon und PDA werden nicht so schnell vollständig durch den MDA ersetzt werden. Gründe sind die mangelnde Subvention der Mobilfunkbetreiber für MDAs mit WLAN-Unterstützung sowie die schlechtere Auflösung und Displaygröße von MDAs gegenüber von PDAs; zur Verwendung des Handhelds als Navigationssystem ist die Displaygröße nicht zu vernachlässigen.

Diese Diplomarbeit beschreibt ein Softwareprodukt, welches sich zur Datenerfassung sowie Informationsbeschaffung auf einem PDA vor Ort beim Kunden oder in der Zentrale eines serviceorientierten Unternehmens der IT-Branche am Desktop-PC eignet. Es werden Techniken, Plattformen und Programmiersprachen beschrieben, mit denen es möglich ist, ein solches System zu entwickeln. Außerdem werden Fragen über die temporäre Speicherung (mobile Datenbank) sowie fixe Speicherung (auf einem Datenbankserver) behandelt. Endprodukt ist ein (neues) weitgehend papierloses Informationssystem für Mitarbeiter und Kunden. Bisher werden Fehlermeldungen und Tätigkeitsberichte des Servicepersonals beim Kunden auf Papier erfasst, diese dann dem Lieferanten zur Abarbeitung gefaxt oder dieser wird telefonisch kontaktiert. In der vorliegenden Arbeit ist dieser Prozess vereinfacht worden.

## Danksagungen

Bedanken möchte ich mich bei allen, die mich beim Schreiben dieser Diplomarbeit unterstützt haben. Dank gilt auch jenen, die mir vor dieser Diplomarbeit schon die Möglichkeit gegeben haben, mein Berufspraktikum bei der Firma Siemens zu absolvieren, sowie im Anschluss eine Firmen-Diplomarbeit zu erstellen:

*Dipl. Ing. Georg Barta* unterstützte mich bei der Korrektur der Diplomarbeit und ist auch mein Erstbegutachter.

*Mag. Georg Bernold* war mein Betreuer im Unternehmen Siemens, und lieferte die Idee zu dieser Diplomarbeit. Er begleitete mich auch während des Berufspraktikums und war Ansprechpartner für fachliche sowie organisatorische Auskünfte.

Ein Dank geht auch an alle Kolleginnen und Kollegen, die mich im Unternehmen sehr zuvorkommend aufgenommen haben.

*Markus Grabner, im September 2007*

## Abstract

Small mobile devices are more and more important for business and private use. They are offered under the terms handheld, pocket PC, PDA, MDA or smart phone. Private operators use them usually as navigation system, because the devices have often integrated a GPS antenna. Advantages of these devices are clear: relatively independent from power supply, small, easy portable and manageable. The coverage of mobile broadband internet services is becoming larger; for that reason, the demand for such devices will go on in private and in business part. In the near future, there will be both handhelds without telephone function (PDA) and such with telephone function (MDA). Currently it seems that most of the customers use a PDA to the common mobile phone additionally; mobile phone and PDA will not completely be substituted fast. Reasons are: the lack of subvention of the mobile providers for MDAs with WLAN support and the worse solution and display dimension of MDAs compared to PDAs. The display dimension is important if it is intended to use the PDA as a navigation system.

This thesis describes a software product, which is suitable for data collection and information search for a field worker on-site. Techniques, platforms and programming languages are characterized, with which it is possible to develop such a system. Furthermore, questions like temporary storage (mobile database) and fix storage (on a data base server) are covered. The final product is a (new) extensively paperless information system for staff and customers. Former, progress and fault reports have been collected on paper by the customers. These slips of paper have been faxed to the vendor or the vendor has been contacted by phone. This thesis identifies possibilities to simplify these processes. Customers will have the opportunity to enter their requests on a web form. The collected data will then be sent to the appropriate administrator.

*Markus Grabner, September 2007*

---

# INHALTSVERZEICHNIS

<b>1 EINLEITUNG</b>	<b>10</b>
1.1 Problemstellung/Projekt/Erkenntnisgegenstand	10
1.1.1 Beschreibung und Funktion des PDA-Info-Systems	10
1.1.2 Begriff: PDA-Info-System	11
1.1.3 Andere Produkte am Markt	11
1.1.4 Benötigte/vorhandene Soft-/Hardware	11
<b>2 ENTWICKLUNG DES PDA-INFO-SYSTEMS</b>	<b>12</b>
2.1 PDA-Info-System: Übersicht, Funktionen	12
2.2 Netzwerktopologie des PDA-Info-Systems	13
2.3 Forschungsmethode und Annäherung an das Problem	13
<b>3 BETRIEBS- UND DATENBANKSYSTEME FÜR MOBILE GERÄTE</b>	<b>15</b>
3.1 Betriebssysteme für Handhelds / PDAs	15
3.1.1 Symbian OS	15
3.1.2 PalmOS	15
3.1.3 Linux Embedded	16
3.1.4 Windows CE	16
3.1.5 Datenbanksysteme für Handhelds	17
3.1.6 Oracle Lite	18
3.1.7 IBM DB2 Everyplace	19
3.1.8 Microsoft SQL Server CE	20
3.2 Gewählte Software	21
3.2.1 Datenbanken für das PDA-Info-System	21
<b>4 MICROSOFT .NET</b>	<b>23</b>
4.1 Einleitung	23
4.2 NET F Begriffe/Technologien	23
4.2.1 ADO.NET	23
4.2.1.1 Data Provider	24
4.2.1.2 DataSet	24
4.2.2 ASP (Active Server Pages)	25
4.2.3 Managed Code	25
4.2.4 CLR (Common Language Runtime)	26
4.2.5 CLI (Common Language Infrastructure)	26
4.2.6 CIL (Common Intermediate Language)	26
4.2.7 CTS (Common Type System)	27
4.3 Kompilierung	28

---

4.4 .NET Sprachen	29
4.4.1 Einleitung	29
4.4.2 Wichtige .NET Sprachen	30
4.4.2.1 Microsoft Visual C#	30
4.4.2.2 Microsoft Visual Basic .NET	30
4.4.2.3 J#	31
4.5 Microsoft .NET Compact Framework	31
4.5.1 Einleitung	31
4.5.2 Erweiterungen zu .NET Compact Framework	31
4.5.2.1 OpenNETCF	32
4.5.2.2 CoreLab MySQLDirect .NET Data Provider	32
4.6 .NET für Microsoft-fremde Umgebungen	32
4.6.1 Mono	32
4.6.2 Portable.NET	33
4.6.3 Microsoft: Project Rotor	33
4.6.4 Red Five Labs .NET Compact Framework für Symbian	33
<b>5 MYSQL</b>	<b>34</b>
5.1 Eigenschaften	34
5.2 Alternativen	35
5.3 Fazit	35
<b>6 DATEN</b>	<b>36</b>
6.1 LOCATION BASED SERVICES	36
6.1.1 Einleitung	36
6.1.2 Location Based Services Funktion mit dem PDA-Info-System	36
6.2 Zu speichernde Daten	38
6.2.1 Feld Art/Kategorie der Anfrage	38
6.2.1.1 Technische Anfrage („Technical Query“)	38
6.2.1.2 Fehlerbericht („Fault Report“)	38
6.2.1.3 Notfall („Emergency“)	38
6.3 Anfragen des Kunden	39
6.3.1 Beispiel eines Webformulars für eine Kundenmeldung	39
6.4 Anpassung der Informationen an Endgeräte	41
6.4.1 Einleitung	41
6.4.2 Begriff der Informationsäquivalenz	41
6.4.2.1 Definition von Informationsäquivalenz	41
6.4.2.2 Informationsäquivalenz im Bezug auf das PDA-Info-System	41
6.4.3 Ursachen der Anpassung	42
6.4.4 Konkrete Anpassungen für das PDA-Info-System	43

---

6.5 Effizientes Suchen mit Hilfe von WebCarousel	45
6.5.1 Begriff	45
6.5.2 Funktionsweise und Entwicklung	45
6.5.3 Ablauf des Auswahlalgorithmus	46
6.5.4 Einbau dieses Systems in das PDA-Info-System	46
6.5.5 WebCarousel Interface	48
6.5.6 Anleitung	48
6.5.6.1 Ähnlichkeit	49
6.5.6.2 Genauigkeit	49
6.5.6.3 Abstände zwischen den Schlüsselwörtern berücksichtigen	51
<b>7 KONFLIKTMANAGEMENT</b>	<b>53</b>
7.1 Transaktionen in MySQL	53
7.1.1 Anwendungsfall einer Transaktion im PDA-Info-System	53
7.2 MySQL Update	56
<b>8 EFFIZIENTE DATENAUFZEICHNUNG UND ANALYSE</b>	<b>59</b>
8.1 Einführung	59
8.2 Mitarbeitertabelle	59
8.2.1 Kundenfeedback	61
8.3 Problembericht – Beschreibung des Problems	61
8.3.1 Probleme klassifizieren	62
8.3.2 Problemstatus	63
8.3.3 Mehrfach vorhandene Fehlereintragungen (Duplikate)	64
8.3.4 Vereinfachung des Problems	65
8.3.4.1 Simplifizierungsalgorithmus	65
8.4 Planung des Desktop Analyse- und Erfassungstools	68
8.4.1 Funktionen des Desktop Analyse- und Erfassungstools	68
8.4.2 Anwendungsfalldiagramm	68
8.4.2.1 Anwendungsfälle	69
8.4.3 Klassendiagramm	72
8.4.4 Objektdiagramme	74
<b>9 BACKUP- UND RESTORE VON DATENBANKEN</b>	<b>75</b>
9.1 Einleitung	75
9.2 ACID-Prinzip	75
9.3 Fehlerbehandlung in Datenbanksystemen	76
9.3.1 Fehlerklassen	76
9.3.1.1 Transaktionsfehler	76
9.3.1.2 Systemfehler	76
9.3.1.3 Externspeicherfehler	76

---

<b>9.4 Backup</b>	<b>76</b>
9.4.1 Unterscheidung nach dem Gegenstand der Sicherung (Granulat)	77
9.4.2 Änderungszustand der zu sichernden Daten	77
9.4.3 Betriebszustand der Datenbank während der Sicherung	78
9.4.4 Paralleles Backup	78
<b>9.5 Restore</b>	<b>78</b>
9.5.1 Unterscheidung nach dem Gegenstand des Restore (Granulat)	78
9.5.2 Art und Weise der Wiederherstellung	79
9.5.3 Betriebszustand der Datenbank während des Restores	79
9.5.4 Paralleles Restore	79
<b>9.6 Alternativen für die Datensicherung und Wiederherstellung</b>	<b>79</b>
9.6.1 Spiegelung	80
9.6.2 RAID	80
9.6.2.1 RAID-Levels für Datenbanken	80
9.6.2.2 RAID-Dreieck	82
<b>9.7 Backup vs. RAID</b>	<b>82</b>
<b>9.8 Backup/Restore und RAID für das PDA-Info-System</b>	<b>83</b>
<b>10 KOSTEN/NUTZEN-ANALYSE</b>	<b>86</b>
<b>10.1 Kosten</b>	<b>86</b>
<b>10.2 Nutzen</b>	<b>86</b>
<b>10.3 Erläuterungen zu den Kostenpositionen</b>	<b>87</b>
10.3.1 Hardwarekosten	87
10.3.2 Lizenzkosten	87
10.3.2.1 Microsoft Lizenzen	87
10.3.2.2 MySQL Lizenzen	88
<b>10.4 Kostenkalkulation für eine/n Mitarbeiter/in inkl. Erstinvestitionen</b>	<b>88</b>
<b>11 ZUSAMMENFASSUNG UND ERWEITERUNGEN</b>	<b>91</b>

# 1 Einleitung

## 1.1 Problemstellung/Projekt/Erkenntnisgegenstand

Möglichst schnelle und zuverlässige Unterstützung und die lückenlose Erfassung der entstandenen Probleme und Lösungen beim Kunden werden angesichts der immer höheren Komplexität der Anwendungssoftware bzw. Hardware immer essentieller. Die Kunden kommen aus der Telekommunikationsbranche (z.B.: Netzbetreiber eines Mobilfunknetzes) und erwerben Produkte von Siemens (z.B.: SimXpress, NTSXpress, VPNXpress oder XBonus)<sup>1</sup>. Aus den oben genannten Gründen muss für ein solches im Feld befindliches Produkt (Soft- oder Hardware), dessen Wartung und Gewährleistung wahrzunehmen ist, abhängig von den Verfügbarkeitsanforderungen, eine 7x24h Hotline für Problemfälle eingerichtet werden.

Beim Kunden auftretende Probleme müssen schnellstens behoben werden. Um nicht immer wieder die gleichen Probleme in den Softwareteilen, Konfigurationseinstellungen bzw. Bedienfehler beheben zu müssen, sollen vorbeugende Maßnahmen gesetzt werden. Dazu ist eine Erfassung und Bewertung aller abgewickelten Geschäftsfälle (Support von Siemens beim Kunden, überregional) erforderlich. Für die Lösung des genannten Problems wurde ein Produkt namens „PDA-Info-System“ entwickelt.

### 1.1.1 Beschreibung und Funktion des PDA-Info-Systems

Der Begriff PDA-Info-System (*vgl. 1.1.2*) bezeichnet ein System, welches Mitarbeitern ermöglicht, direkt während der Behandlung eines Problems beim Kunden, auf einem mobilen Gerät (Bezeichnungen: PDA, Pocket PC oder Handheld) Daten auf einer Oberfläche einzutragen, und diese Daten dann auf einer Datenbank zu sichern.

Der Begriff umfasst auch die theoretische Entwicklung eines Desktop Analyse- und Erfassungstools (DAE) um zukünftige Fehler abschätzen und prognostizieren zu können. Auf dieses Tool wird in *Abschnitt 8.4* eingegangen.

---

<sup>1</sup> Diese Produkte gehören zur ServiceXpress Produktfamilie, die von der Siemens PSE (Programm- und Systementwicklung) entwickelt worden ist. ServiceXpress-Produkte erlauben die schnelle und einfache Administration von Diensten, die bei Mobilfunkbetreibern im Einsatz sind. Sämtliche Benutzeroberflächen werden in handelsüblichen Webbrowsern dargestellt. Mit ServiceXpress kann der Netzbetreiber seine Dienste und deren Anwender einfach und sicher verwalten.

### 1.1.2 Begriff: PDA-Info-System

Der Begriff PDA-Info-System wurde deshalb gewählt, da es sich bei diesem Produkt vorwiegend um ein *Informations-* und *Datensammlungssystem* handelt, und die Möglichkeit bietet, Daten mobil online sowie mobil offline zu erfassen. Bei der Offline-Erfassung müssen die Daten dann zu einem späteren Zeitpunkt mit der Serverdatenbank synchronisiert werden, um diese aktuell zu halten.

### 1.1.3 Andere Produkte am Markt

Da es etwas Vergleichbares, speziell für diesen Anwendungsfall (in *Abschnitt 1.1* erläutert) zugeschnittenes Produkt noch nicht gibt, bzw. von einem externen Unternehmen (z.B. Microsoft oder ein sonstiges spezialisiertes Unternehmen in diesem Bereich) für diesen Fall entwickelt werden müsste, hat sich Siemens für diese Individuallösung entschieden.

Ein ähnliches Produkt wurde von Microsoft für die Telekom entwickelt. [ms2]

Wichtig ist die komplette Offline-Funktionalität, d.h. die Daten können auch offline am PDA beim Kunden erfasst und dort „zwischengeparkt“ werden. Eine spätere Synchronisation mit der Serverdatenbank ist dann erforderlich.

### 1.1.4 Benötigte/vorhandene Soft-/Hardware

Folgende Hardware muss beschafft werden: PDA (Personal Digital Assistent) oder MDA, Windows PC mit Entwicklungsumgebung. (*näheres siehe Kapitel 10 Kosten/Nutzen-Analyse*)

Vorhandene Basis ist ein UNIX-Server mit einem installierten MySQL-Datenbanksystem.

**Software am PDA und MDA:** Windows Mobile 5.0, Microsoft .NET Compact Framework 2.0, OpenNETCF Smart Device Framework (Erweiterung des .NET Compact Frameworks), Microsoft SQL Server CE 2005

**Software am Windows PC:** Entwicklungsumgebung: Microsoft Visual Studio 2005, MySQL Server für Testzwecke.

## 2 Entwicklung des PDA-Info-Systems

### 2.1 PDA-Info-System: Übersicht, Funktionen

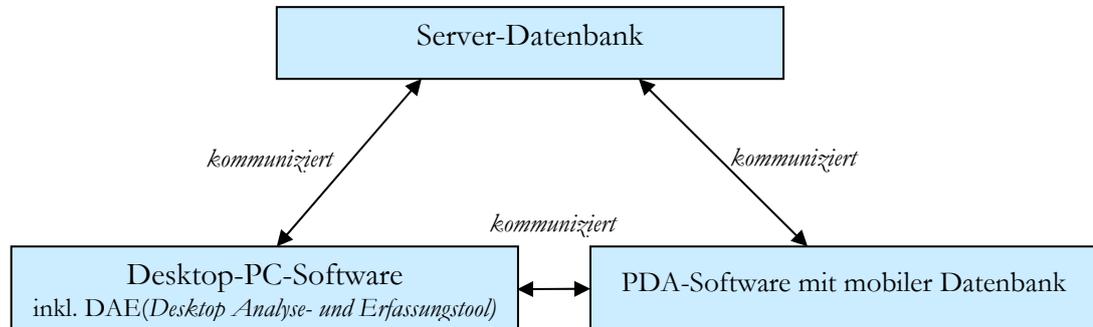


Abb. 1: Komponenten des PDA-Info-Systems

**Server-Datenbank:** Auf der Server-Datenbank werden die Daten, die von einem/einer Siemens Mitarbeiter/in (vor Ort oder in der Zentrale, s. *Abb. 2* Netzwerktopologie) am PDA oder auf der Desktop-PC-Software eingetragen werden, erfasst. In *Abschnitt 3.2.1* wird beschrieben, warum MySQL als Datenbanksystem für das PDA-Info-System verwendet wurde. *Kapitel 5* erklärt die grundlegenden Funktionen von MySQL.

**Desktop-PC-Software:** Die Desktop-PC-Software besteht aus den Funktionalitäten der PDA-Software sowie dem DAE (Desktop Analyse- und Erfassungstool). Auf dieses Tool wird in *Abschnitt 8.4* eingegangen.

**PDA-Software mit mobiler Datenbank:** Die PDA-Software ist eine Anwendung, die auf einem PDA installiert wird und sich zur mobilen Erfassung von Daten direkt vor Ort beim Kunden eignet. Ein/e Servicemitarbeiter/in oder Techniker/in erfasst mithilfe dieser Software die Daten beim Kunden. Der PDA enthält auch eine sogenannte mobile Datenbank. Diese dient zur kurzfristigen Speicherung von Daten.

## 2.2 Netzwerktopologie des PDA-Info-Systems

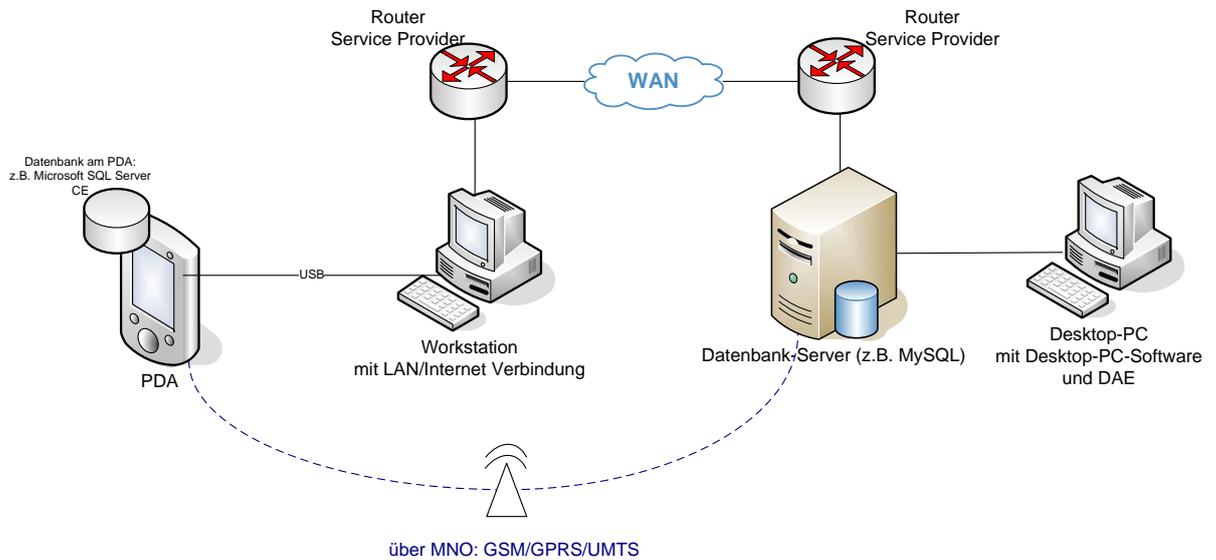


Abb. 2: Topologie des PDA-Info-Systems

Anmerkung zur *Abb. 2*: **MNO**: Ein MNO (Mobile Network Operator) ist ein Mobilfunkbetreiber. **WAN**: Wide Area Network oder Weitbereichsnetze erstrecken sich über einen weiten geographischen Bereich.

Die Datenerfassung kann (wie aus *Abb. 2* ersichtlich) entweder mobil am PDA (*in der Abb. 2 links*) oder am Desktop-PC (*rechts*) in der Zentrale erfolgen. Die Daten werden bei mobiler Erfassung entweder sofort über den MNO (bei Verfügbarkeit), oder zu einem späteren Zeitpunkt über ein WAN in der Zentrale, in der sich der Datenbankserver befindet, gespeichert.

## 2.3 Forschungsmethode und Annäherung an das Problem

Zu Beginn wurden verfügbare mobile Datenbanken verglichen. Auf Basis dieses Vergleiches wurde jene ausgewählt, die für das PDA-Info-System am besten geeignet ist. (Microsoft SQL Server CE) (*s. auch Abschnitt 3.2.1*). Als Vorgabe galt, das Entwicklungswerkzeug Visual Studio 2005 zu verwenden. Ebenfalls Vorgabe war die Verwendung von MySQL am Server. Entwickelt wurde auf Basis von .NET mit der Programmiersprache C#. C# ist eine professionelle Programmiersprache für einen breiten Anwendungsbereich. Die Grundzüge von .NET werden im *Kapitel 4* beschrieben. Im Berufspraktikum bei Siemens wurde bereits die Oberfläche der PDA-Software geplant und später wurde sie entwickelt. Probleme ergaben sich in der An-

bindung des PDAs an eine MySQL-Datenbank. *Abschnitt 3.2.1* geht auf dieses Problem ein. Die effiziente Anzeige von Suchergebnissen am PDA wird in *Abschnitt 6.4* besprochen. Die Suche stellt ein wichtiges Instrument dar, um Servicetechniker/innen vor Ort benötigte Informationen zur Verfügung zu stellen. (s. *Kapitel 6.5*) Schließlich wird noch auf das Thema der effizienten und ausfallsicheren Archivierung der Daten auf der Server-Datenbank eingegangen. Das *Kapitel 9* widmet sich diesem Thema.

## 3 Betriebs- und Datenbanksysteme für mobile Geräte

### 3.1 Betriebssysteme für Handhelds / PDAs [Höpf, Kapitel 2]

In diesem Abschnitt werden die bedeutendsten Betriebssysteme, die zurzeit für PDAs verfügbar sind, erklärt. Es handelt sich hierbei um:

- Symbian OS
- PalmOS
- Linux Embedded
- Windows CE

Im *Abschnitt 3.1.4* wird die Verwendung von Windows CE für das PDA-Info-System begründet. Im *Abschnitt 3.2* wird die Frage beantwortet, welche Datenbank für das PDA-Info-System verwendet wird.

#### 3.1.1 Symbian OS

Symbian OS entstand 1998 durch eine Allianz der Unternehmen Ericsson, Nokia und Motorola mit Psion (Hersteller der gleichnamigen Handhelds mit dem Betriebssystem EPOC).

Symbian OS setzt auf offene Standards und unterstützt Technologien wie Java, WAP, HTML, Bluetooth, GSM/GPRS/EDGE und UMTS. Es kann mehrere Threads gleichzeitig ausführen. Dieses Betriebssystem wird sehr häufig in sogenannten Smart Phones (Telefon und Organizer in einem) eingesetzt.

#### 3.1.2 PalmOS

Geräte die mit PalmOS arbeiten, existieren schon etwas länger, nämlich seit 1996. PalmOS-Geräte sind meist stiftbasiert. Die Firmen Sony und Handspring (seit 2004 palmOne) sowie die Firma Palm Inc. stell(t)en Geräte her, die mit PalmOS arbeiten. Sony hat mit seiner Produktfamilie „Clie“ die Herstellung solcher Geräte Anfang 2004 eingestellt. Applikationen sind plattformunabhängig auf unterschiedlichen PalmOS-Geräten lauffähig. Das aktuelle Betriebs-

system ist PalmOS 6 (Cobalt). Es hat sich aber bisher noch kein Hersteller gefunden, Geräte mit dieser Version auszuliefern. [web1]

### 3.1.3 Linux Embedded

Bei Linux Embedded handelt es sich um die Portierung des Linux Betriebssystems auf mobile Hardware. Hier wird der Standard Linux Kernel auf das mobile System portiert, somit steht die gesamte Funktionalität des Desktops auch für den PDA zur Verfügung. Wie bei Linux üblich, gibt es mehrere grafische Oberflächen für Embedded Linux wie z.B. Qtopia von Trolltech und OPIE (Open Palmtop Integrated Environment). Linux Embedded kann in den vergangenen Jahren auf ein starkes Wachstum zurückblicken. Vorteil ist seine Kostengünstigkeit, da keine teuren Entwicklungsumgebungen erworben werden müssen. Eine solche freie Entwicklungsumgebung (ohne der komplizierten Abstimmung von Compiler, Linker und Debugger) ist z.B. ELinOS von Sysgo.

### 3.1.4 Windows CE

Für Windows CE existieren je nach Version viele unterschiedliche Bezeichnungen. Es gibt die Bezeichnung Microsoft Windows CE, Microsoft Pocket PC, Microsoft Windows Mobile (für Pocket PC) oder ohne den Zusatz „Pocket PC“. In dieser Arbeit werden die Bezeichnungen Windows CE sowie Windows Mobile verwendet. Diese Bezeichnungen beschreiben dieselbe Betriebssystemfamilie, jedoch nicht eine bestimmte Version des Betriebssystems. Wird eine Versionsnummer oder Jahreszahl hinzugefügt, handelt es sich um die jeweilige Version.

Windows CE gibt es nicht nur für PDAs sondern auch für sogenannte *eingebettete Systeme* (engl. *embedded systems*). Eingebettete Systeme sind Systeme, die sehr hardwarenah konstruiert sind, wobei die Software speziell auf die Hardware abgestimmt sein muss. Sie kommen z.B. in Flugzeugen, Autos, Spielkonsolen und auf allen möglichen Geräten der Unterhaltungselektronik vor. Bei Autos und Flugzeugen vernetzt das eingebettete System verschiedene Teilsysteme untereinander. Durch die immer größer werdende Bedeutung von Elektronik in früher einfach konstruierten Geräten, ist für solche Betriebssysteme ein hohes Potential gegeben. Konkurrenzprodukte zu Microsoft in diesem Anwendungsbereich sind: VxWorks, OSEK-OS, Linux-Derivate, NetBSD. Außerdem gibt es für solche Systeme viele unterschiedliche CPU-Architekturen: ARM, MIPS, PowerPC usw. [web2]

Windows CE gewann in den letzten Jahren immer mehr an Bedeutung, vor allem auf den Handhelds. Ein Grund hierfür ist, dass sein großes Vorbild (die gesamte Windows-Familie) Marktführer bei PC-Betriebssystemen ist, und sich Windows CE sehr stark an das Desktop Vorbild anlehnt. Für die Applikationsentwicklung gibt es Ereignisse, Buttons, fertige Open-File-Dialoge und Save-As-Dialoge, Check-Boxen, Optionsfelder, Datenbankanbindungen, Tabellen usw. Die Einarbeitungszeit ist relativ kurz und gestaltet sich einfach. Windows CE ist Multi-Prozess- und multithreadingfähig, und es laufen auch ressourcenintensive Programme wie Tabellenverarbeitung mit PocketExcel und Textverarbeitung mit PocketWord.

Nachteile von Windows CE: Windows CE Geräte sind teurer als Konkurrenzprodukte, sind aber auch leistungsfähiger hinsichtlich Hardware (Prozessor, Speicher). Ein weiterer Nachteil ist die nach Tests festgestellte Instabilität von Applikationen die auf MDAs<sup>1</sup> laufen. Wenn eine Applikation nicht mehr reagiert, ist manchmal nur noch ein Reset und zurücksetzen auf die Werkseinstellungen möglich, um das Gerät wieder zum laufen zu bringen.

Es ist zu erwarten dass der Marktanteil von Windows CE aufgrund der intuitiven Bedienung in Zukunft noch steigen wird. Auch die nahe Verwandtschaft (aber nur hinsichtlich der Bedienung, nicht in der Architektur) zu den Windows PC Systemen, trägt zu diesem Trend bei.

Für dieses Projekt wurde ein PDA von Fujitsu Siemens mit installiertem Windows Mobile verwendet.

### **3.1.5 Datenbanksysteme für Handhelds**

In diesem Kapitel werden zuerst die Unterschiede der gängigen Datenbanken dargestellt. Danach wird auf Fragen wie:

- Welche Datenbank soll am PDA verwendet werden?
- Welche Datenbank läuft zentral am Datenbankserver?
- Wie soll ein Abgleich zwischen PDA und zentralen Datenbankserver erfolgen?
- Wie können Daten bei Ausfall (also redundant) zur Verfügung gestellt werden?

eingegangen.

---

<sup>1</sup> MDA: Ein MDA (Mobile Digital Assistent) ist ein Gerät wie ein PDA jedoch mit der zusätzlichen Möglichkeit mobil zu telefonieren.

Im Folgenden wird eine Übersicht über die wichtigsten mobilen Datenbanksysteme auch im Hinblick auf eigene Entwicklungen gegeben. Es wird für detaillierte Informationen auf [Höpf, ab Seite 261] verwiesen.

### 3.1.6 Oracle Lite

Oracle Lite unterstützt mit JDBC und ODBC zwei standardisierte SQL-Schnittstellen für die Anwendungsprogrammierung. Aus C/C++-Anwendungen kann mit OKAPI (Object Kernel API) zugegriffen werden.

#### Architektur von Oracle Lite [Höpf]

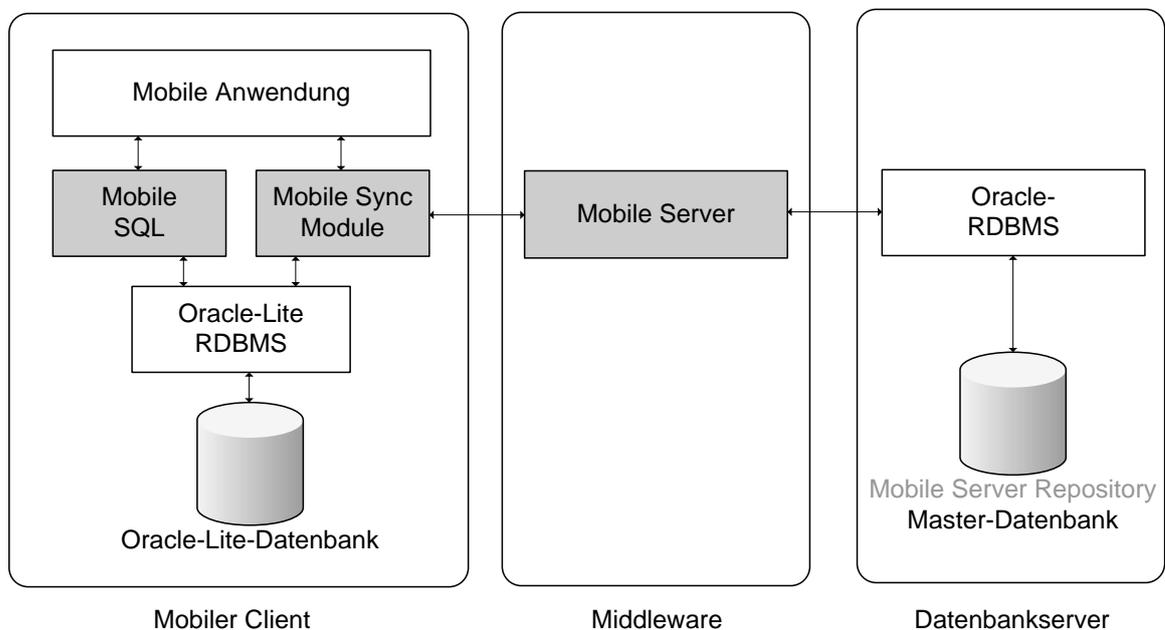


Abb. 3: Architektur von Oracle Lite

#### Relevante Komponenten von Oracle Lite (s. Abb. 3)

*Mobile SQL:* Mobile SQL ist die interaktive SQL-Schnittstelle zu einer Oracle-Lite-Datenbank. Hiermit können SQL Abfragen getätigt werden. Der Sprachumfang ist eine Erweiterung von SQL-92, besitzt aber bei weitem nicht den Umfang von Oracle-SQL. Auf der Windows-32-Plattform arbeitet Mobile SQL ähnlich, wie das kommandozeilenbasierte Oracle-Tool SQL\*Plus. Es gibt auch eine grafische Benutzerschnittstelle für Palm-OS und Windows-CE.

*Mobile Sync Module:* Das Mobile Sync Module läuft am mobilen Client und erledigt die Synchronisation zwischen Oracle Lite und der Master-Datenbank.

*Mobile Server:* Der Mobile Server überprüft bei der Synchronisation die Authentizität des Nutzers und übergibt Änderungen, die in der Oracle-Lite-Datenbank vorgenommen wurden, in die Master-Datenbank.

*Mobile Server Repository:* Das Mobile Server Repository verwaltet spezifische Informationen in der Master-Datenbank, die für den Mobile Server notwendig sind.

### 3.1.7 IBM DB2 Everyplace

Mobile Version des DB2-Datenbanksystems von IBM. IBM DB2 Everyplace ist unter anderem verfügbar für PalmOS, Windows CE/Pocket PC, Linux sowie für Win32-basierte Plattformen. DB2 unterstützt Entwicklungen von Anwendungen in C/C++ und Java auf der Basis von ODBC und JDBC.

DB2 funktioniert wie Oracle Lite auf einer Client-Middleware-Server-Architektur.

#### Architektur von DB2 Everyplace [Höpf]

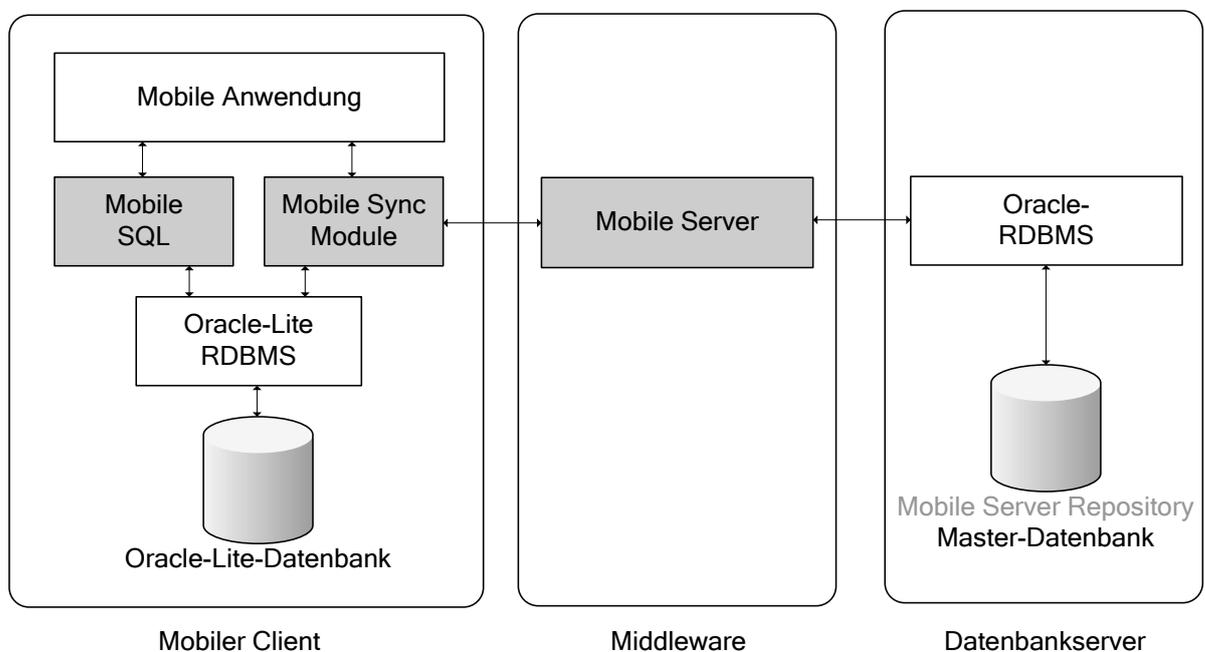


Abb. 4: Architektur von DB2-Everyplace

### Relevante Komponenten von DB2 Everyplace (s. Abb. 4)

*QBE (Query By Example)* ist eine grafische Benutzerschnittstelle und ermöglicht Abfragen der Datenbank. QBE läuft am mobilen Client.

*Sync Client:* Der Sync Client ist ebenfalls eine Anwendung am mobilen Client. Er erledigt die Synchronisation der DB2-Everyplace- und der Masterdatenbank. Hierbei erfolgt der Datenaustausch mit dem Sync-Server.

*Sync Server:* Der Sync Server ist eine Middleware-Anwendung. Er übernimmt die Synchronisation der Daten zwischen Master Datenbank am Server und der mobilen DB2 Everyplace Datenbank.

### 3.1.8 Microsoft SQL Server CE

Microsoft SQL Server CE läuft nur auf dem Betriebssystem Windows CE. Als Entwicklungsumgebung wird „Microsoft Visual Studio .NET“ unterstützt. SQL Server CE ist Bestandteil von Microsoft's .NET Produkten. Das .NET Compact Framework muss für die Entwicklung von Datenbank Anwendungen installiert sein.

### Architektur von Microsoft SQL Server CE [Höpf]

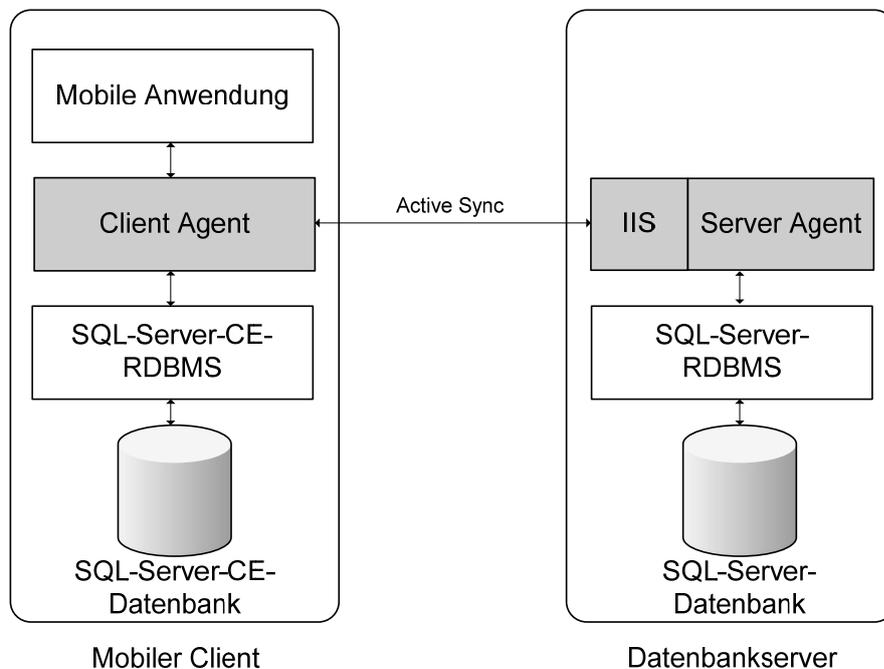


Abb. 5: Architektur von Microsoft SQL Server CE

## Relevante Komponenten von SQL Server CE (s. Abb. 5)

*SQL Server CE* ist die eigentliche Datenbank, die am mobilen oder eingebetteten Gerät läuft.

*Client Agent*: Der Client Agent kommuniziert mit dem Server Agent und stößt die Synchronisation zwischen der mobilen und der Master-Datenbank an. Synchronisiert wird auf Basis von Tabellen.

*Active Sync* führt den nachrichtenbasierten Synchronisationsvorgang durch, d.h. es erfolgt mit Active Sync die Kommunikation zwischen mobilen Client und Backend-Server.

*Server Agent*: Der Server Agent ist ein Modul des Microsoft Internet Information Service (IIS) und läuft auf dem Backend-Server. Der Ablauf ist folgender: Die Datenbankanweisungen kommen über ein IP Paket an, werden dann entpackt und an die Masterdatenbank gesendet.

*IIS*: Das Internet Information Service ist ein Webserver von Microsoft der im Windows Server 2003 integriert ist.

## 3.2 Gewählte Software

### 3.2.1 Datenbanken für das PDA-Info-System

Welches Datenbanksystem für das PDA-Info-System am besten geeignet ist, ist eine nicht ganz leicht zu beantwortende Frage. Eine optimale Lösung wäre natürlich, Produkte eines Herstellers sowohl am Server, als auch am Client zu verwenden, da alle Synchronisierungs- und Replizierungstechniken bereits fertig angeboten werden. Für Replizierung und Synchronisation wird auf [Höpf] verwiesen. Wird mit Windows CE gearbeitet und mit Visual Studio .NET entwickelt, wie im vorliegenden Fall, bieten sich natürlich SQL Server CE und Microsoft SQL Server für die Server-Datenbank an. Am PDA wird eine Datenbank benötigt, weil *Caching* erfolgen soll, d.h. die Daten weisen Redundanz auf, um z.B. auch Zugriff auf einen Teil der Daten außerhalb des Empfangsbereiches des Servers zu haben, oder um einen schnelleren Zugriff und eine schnellere Abfrage von Datenbeständen zu gewährleisten.

Im zugrunde liegenden Projekt für diese Arbeit wurde jedoch ein anderer Ansatz gewählt. Da am Server eine MySQL Datenbank (s. Kapitel 5) verwendet werden musste (sämtliche Datenbestände liegen auf einer MySQL Datenbank und eine Umstellung/Portierung der Datenbestände auf z.B. Microsoft ist vom Projektauftraggeber nicht geplant), war der komfortablere Weg des mobilen Datenbanksystems mit fertigen Synchronisations- und Replikationstechni-

ken ausgeschlossen. Das heißt, ein neuer Lösungsansatz musste gefunden werden, da es von MySQL kein mobiles Datenbanksystem gibt.

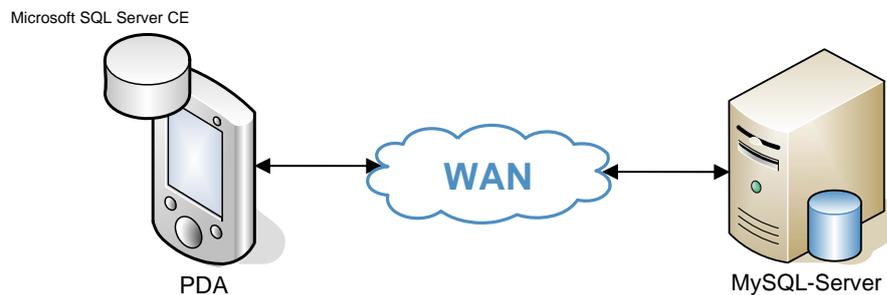


Abb. 6: PDA-Server-Infrastruktur

Wie in *Abb. 6* ersichtlich, wird am PDA eine SQL Server CE Datenbank verwendet und am Server eine MySQL-Datenbank. Da mit .NET entwickelt wurde, wurde die Microsoft SQL Server CE Datenbank am PDA verwendet. Im .NET Framework existieren Klassen und Methoden für die Anbindung der SQL Server CE Datenbank an die PDA-Software. (*Abschnitt 4.2.1.1*)



Abb. 7: PDA-Server-Infrastruktur

*Abb. 7* zeigt, wie Microsoft SQL Server CE und MySQL zusammenarbeiten. Das Bindeglied ist die PDA-Software. Die Abfrage oder die Speicherung von Datensätzen auf der SQL Server CE Datenbank erfolgt mit Klassen und Methoden der Bibliothek „System.Data.SqlServerCe.dll“. In die andere Richtung geschieht das mit der Bibliothek „CoreLab.MySql.dll“, s. *Abschnitt 4.5.2.2*. Die PDA-Software fungiert als „Übersetzerin“. Bsp.: Ergebnisse aus einer SQL Server CE Abfrage werden in einer Variablen zwischengespeichert und dann bei Bedarf in die MySQL Datenbank übertragen und/oder umgekehrt.

## 4 MICROSOFT .NET [vgl. web3]

### 4.1 Einleitung

Da sämtliche Entwicklungen dieser Arbeit auf .NET beruhen, erfolgt an dieser Stelle eine Einführung zu diesem Thema. .NET ist eine noch relativ junge Technologie, deren Zukunftspotential als sehr vielversprechend angesehen werden kann. Microsoft hat einen hohen Marktanteil bei Betriebssystemen; deshalb sind .NET Programme auf vielen Rechnern lauffähig. Im neuen Betriebssystem „Windows Vista“ wird das .NET Framework standardmäßig mitinstalliert.

Microsoft .NET deckt einen großen Bereich von Produkten ab. In diesem Kapitel wird auf das Microsoft .NET Framework (im folgenden .NET F genannt) eingegangen, das für diese Arbeit benötigt wird.

### 4.2 NET F Begriffe/Technologien

#### 4.2.1 ADO.NET

Stellt Klassen und Methoden für den Datenbankzugriff zur Verfügung. ADO.NET wird oft als Erweiterung von ADO (ActiveX Data Objects) verstanden, wurde aber so stark verändert, dass es als neues Produkt betrachtet werden kann.

ADO.NET besteht aus zwei Teilen:

- 1) Data Provider
- 2) Data Sets

#### 4.2.1.1 Data Provider

Der Data Provider stellt das Bindeglied zwischen der Applikation und der Datenbank bzw. dem Datenbankmanagementsystem dar. Er besteht aus einer Vielzahl unterschiedlicher C#-Klassen. Solche Klassen sind z.B.:

*SqlConnection*: Diese Klasse ist für die Verbindung zum Server zuständig, d.h. Instanzen dieser Klasse haben Eigenschaften, die eine Verbindung beschreiben. (z.B. IP-Adresse, Name der Datenbank, User, Passwort, Datenbankport)

*SqlCommand*: Diese Klasse beschreibt ein SQL-Kommando. Dieses Kommando kann gleich dem Konstruktor beim Erstellen einer Instanz der SqlCommand-Klasse übergeben werden.

*Beispiel*

```
SqlCommand cmd = new SqlCommand(„INSERT INTO table1 (Name) VALUES  
( 'ServiceProvider' )“ , con);  
cmd.ExecuteNonQuery();
```

Als zweiter Parameter muss die SqlConnection (das SqlConnection Objekt) übergeben werden (in obenstehenden Beispiel: „con“). Die Methode *ExecuteNonQuery()* angewandt am SqlCommand-Objekt führt dann diesen SQL-Befehl an der Datenbank aus. Es wird der Datensatz in die Datenbank eingetragen.

*SqlDataAdapter*: Der SqlDataAdapter dient zum Füllen eines DataSet. Ein DataSet ist ein „in-memory“ Cache einer Datenquelle. (s. Abschnitt 4.2.1.2) Es enthält z.B. die Datensätze einer Select-Abfrage. Diese Datensätze können dann in der GUI mit der *DataGridView* Klasse dargestellt werden.

#### 4.2.1.2 DataSet

DataSets sind Klassen, die eine relationale Datenbank beschreiben. Bei Betrachtung dieser Klassen wird klarer, worum es geht:

*DataTable*: Eine Tabelle, die aus Namen, Zeilen (Rows) und Spalten (Columns) besteht.

*DataGrid*: Beschreibt eine anzeigbare Tabelle, die bestimmte Möglichkeiten aufweist. (z.B. sortieren)

*DataGridView* ersetzt DataGrid in den neueren Versionen.

*DataColumn* repräsentiert eine Spalte des DataGridView.

*DataRow* repräsentiert eine Zeile des DataGridView.

Es gibt eine Vielzahl weiterer Klassen und Methoden. Hier wurden nur die wichtigsten erläutert. Es wird auf die .NET CF Bibliothek von MSDN verwiesen. *Weitere Informationen unter [ms1]*

Fertige direkte .NET Data Provider<sup>1</sup> gibt es von Microsoft nur für den Microsoft SQL Server und Microsoft SQL Server CE. Für andere Produkte müssen die Datenbankhersteller selbst Treiber für ADO.NET zu Verfügung stellen. Für MySQL gibt es z.B. eine frei verfügbare Version eines Connectors (CONNECTOR/.NET) für .NET F, allerdings nur für das Microsoft .NET Framework, nicht für das .NET Compact Framework das im *Abschnitt 4.5* erläutert wird.

#### **4.2.2 ASP (Active Server Pages)**

Serverseitige Scriptsprache, die für die Entwicklung von Webseiten verwendet wird. ASP.NET ist ein Teil von .NET.

#### **4.2.3 Managed Code**

Managed Code wird zu einem Zwischencode kompiliert, der aber nicht direkt auf der Maschine ausgeführt werden kann. Dieser Code läuft in der Laufzeitumgebung von .NET (CLR), siehe *Abschnitt 4.2.4*

---

<sup>1</sup> Ein Data Provider (oder Connector) stellt Klassen und Methoden zur Verfügung, um von einer Programmiersprache aus auf eine Datenbank zuzugreifen.

#### 4.2.4 CLR (Common Language Runtime)

CLR ist die Microsoft-Implementation von CLI. CLI (Common Language Infrastructure) definiert eine Laufzeitumgebung für Programmcode. Der Ablauf sieht folgendermaßen aus: Der Entwickler schreibt seinen Programmcode in einer Hochsprache wie z.B. C# (ausgesprochen „si-Scharp“). Der Compiler der Sprache konvertiert diesen Code in die CIL (Common Intermediate Language, früher MSIL (Microsoft Intermediate Language)), das CLR konvertiert dann in einen nativen Hardware-Code.

#### 4.2.5 CLI (Common Language Infrastructure)

Die CLI ist ein Standard, der Systeme spezifiziert, die sprach- und plattformneutrale Anwendungsentwicklung und –ausführung ermöglichen. [web26]

#### 4.2.6 CIL (Common Intermediate Language)

*Oder früher: MSIL (Microsoft Intermediate Language)*

Ist die niedrigste vom Menschen noch lesbare Sprache des CLI (Common Language Infrastructure).

CIL kann von jeder Plattform auf der .NET F läuft ausgeführt werden. CIL stellt Sicherheit während der Laufzeit und höhere Zuverlässigkeit als nativ kompilierte Binaries zur Verfügung. Um geistiges Eigentum zu schützen, ist die leichte Zugänglichkeit und Verfügbarkeit insbesondere von Disassemblern ein Problem, da die erstellte exe-Datei als CIL vorliegt und mit dem mit Visual Studio mitgelieferten Tool „.NET Framework IL Disassembler“ betrachtet werden kann. Microsoft begegnet diesem Problem teilweise mit dem „Dotfuscator“<sup>1</sup>. Der Code wird mit dem Dotfuscator verschleiert und teilweise unverständlich. Eine hundertprozentige Sicherheit gewährt dieses Tool aber nicht. Eine Rückgewinnung in einen lesbaren Zustand wäre aber mit einem erheblichen, zeitmäßig unwirtschaftlichen Aufwand verbunden. Dieser Aufwand würde meist mehr Zeit erfordern, als dies bei einer Neuentwicklung der Fall wäre. Eine weitere Möglichkeit wäre, den Code als Binaries für eine bestimmte Hardware zur

---

<sup>1</sup> Dotfuscator: Der Microsoft Dotfuscator ist ein Tool, das es ermöglicht, Code vor dem unberechtigten Lesen zu schützen (Schutz vor Decompilierung). Die Vollversion ist nicht in Visual Studio integriert, jedoch eine Light Version: Dotfuscator Community Edition

Verfügung zu stellen. Dies widerspricht allerdings wieder dem Vorteil von .NET, nämlich der Plattformunabhängigkeit. Der beste Schutz vor unberechtigtem Lesen des Quellcodes ist, diesen Quellcode gar nicht erst am Zielrechner ausführen zu lassen, sondern z.B. als Webservice. Dem Zielrechner werden nur die Ergebnisse mitgeteilt.

Ein großer Vorteil ist aber – wie vorher schon erwähnt – die Ausführbarkeit dieses Codes auf jeder Plattform auf der .NET läuft. Zusätzlich liefert jede .NET Sprache genau den gleichen CIL-Code.

#### 4.2.7 CTS (Common Type System)

CTS wird in jeder Sprache, die auf .NET aufsetzt, verwendet. CTS stellt eine gemeinsame Sammlung von Typen zur Verfügung, die in sehr vielen unterschiedlichen Sprachen verwendet werden können. Beinahe jede .NET Sprache im speziellen (C# und VB .NET) verlässt sich sehr stark auf CTS.

CTS stellt Basisdatentypen zur Verfügung. Bestimmte andere Sprachen wie C# verwenden Aliase z.B.: „int“ für Integer; „int“ ist in CTS als die Struktur System.Int32 definiert.

Es könnte eine Variable in C# folgendermaßen definiert werden:

```
System.Int32 value = 18;
```

*oder aber auch kürzer:*

```
int value = 18;
```

„int“ ist eine Abkürzung, ein Alias, der dasselbe bedeutet wie System.Int32.

Die Funktion von CTS ist, dass viele unterschiedliche Sprachen (die alle Aliase oder unterschiedliche Typen haben) integriert werden können. Weiters besteht Typsicherheit für den Speicher und Objektorientierung. CTS definiert außerdem Regeln, wie Objekte, die in unterschiedlichen Sprachen entwickelt wurden, miteinander interagieren.

### 4.3 Kompilierung

Abb. 8 zeigt die Übersetzung/Kompilierung eines einfachen C# Programms.

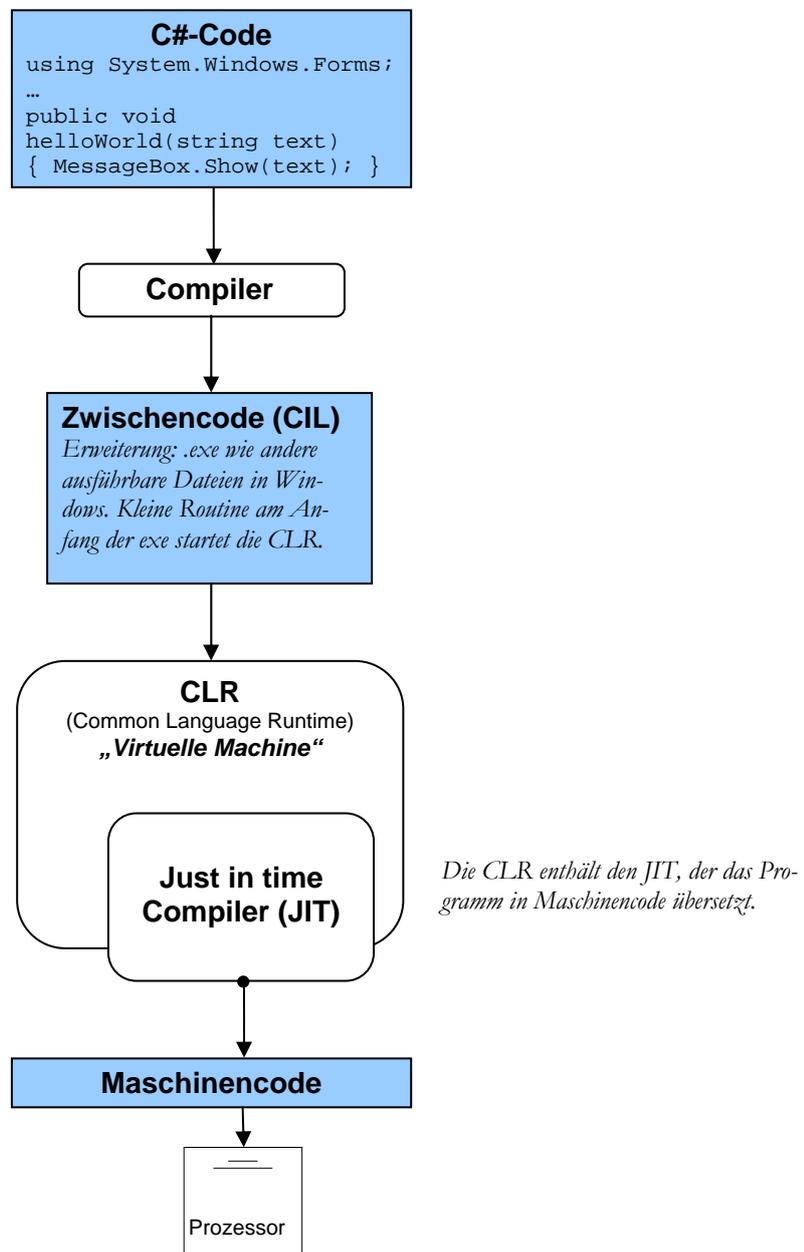


Abb. 8: Übersetzungsvorgang von .NET.

In den Abbildungen (Abb. 9 und Abb. 10) wird ein Stück Code verglichen. Abb. 9 zeigt C#-Code, und Abb. 10 zeigt den Zwischencode (CIL) nach Übersetzung. (s. auch Abb. 8 Übersetzungsvorgang).

## 4.4 .NET Sprachen [web4]

### 4.4.1 Einleitung

Das Resultat beim Kompilieren<sup>1</sup> eines Programms in einer .NET Sprache (egal welche) ist immer das gleiche, nämlich ein Programm in der CIL (Common Intermediate Language). Dieses Programm kann nun auf jeder Plattform, auf der .NET Framework installiert ist, ausgeführt werden. Nach dem Kompilieren entsteht mit dem Microsoft Visual Studio 2005 IDE<sup>2</sup> eine exe-Datei, die dann ausgeführt werden kann.

Wird diese exe-Datei gestartet, so wird sie vom CLR virtuell ausgeführt.

```
/* Program.cs
 * Hauptprogramm, enthält Methode Main.
 *
 * AUTHOR: Markus Grabner
 * DATE: 2006-08-01 */

using System;
using System.Collections.Generic;
using System.Windows.Forms;
using System.Threading;
using System.Runtime.InteropServices;
using OpenNETCF.Diagnostics;
using System.Diagnostics;

namespace ServiceXpress
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [MTAThread]
        static void Main()
        {
            //Splash-Screen laden
            SplashScreen.ShowSplashScreen();
        }
    }
}
```

Abb. 9: Code-Snippet von C#-Code eines Teiles der entwickelten Software für diese Arbeit

<sup>1</sup> Kompilieren = Übersetzen

<sup>2</sup> IDE: Integrated Development Environment (*d.h.* integrierte Entwicklungsumgebung) Die IDE ist ein Programm zur Entwicklung von Software. Sie besteht meist aus einem Texteditor, einem Compiler, einem Debugger, einem Linker und diversen Quelltext-Formatierungsfunktionalitäten und im Falle von Visual Studio noch vielen weiteren Tools.

```

.method private hidebysig static void Main() cil managed
{
    .entrypoint
    .custom instance void [mscorlib]System.MTAThreadAttribute::.ctor() = ( 01 00 00
00 )
    // Code size          8 (0x8)
    .maxstack 8
    IL_0000: nop
    IL_0001: call          void ServiceXpress.SplashScreen::ShowSplashScreen()
    IL_0006: nop
    IL_0007: ret
} // end of method Program::Main

```

Abb. 10: Code-Snippet von CIL (MSIL)-Code eines Teiles der entwickelten Software für diese Arbeit

## 4.4.2 Wichtige .NET Sprachen

### 4.4.2.1 Microsoft Visual C#

Das C#-Symbol steht für das musikalische Zeichen eines überhöhten Tones. [vgl. web5]

C# ist eine professionelle Sprache für die Entwicklung komplexer Aufgabenbereiche. Sie kann als eine Mischung (von der Syntax her) zwischen C++ und Java betrachtet werden. Vor allem Java-Programmierern dürfte ein Erlernen dieser Sprache keine größeren Probleme bereiten. C# bietet Unterscheidung von Groß-/Kleinschreibung, „nicht gesicherter Code“<sup>1</sup>, überladen von Operatoren, integrierte Versionserstellung und, im Vergleich zu Visual Basic, bessere Unterstützung von Inline-Dokumentation [ms2]

Alle Softwareprodukte, die dieser Arbeit zugrunde liegen, wurden in C# entwickelt. C# ist objektorientiert und benötigt das .NET Framework, das in der aktuellen Version 3.0 vorliegt.

### 4.4.2.2 Microsoft Visual Basic .NET

Microsoft Visual Basic .NET ist eine komplett neue Version von Visual Basic für .NET, ebenfalls objektorientiert, weist aber nicht die vielseitigen Möglichkeiten von C# auf.

VB .NET ist für Visual Basic Programmierer gedacht und weist z.B. keine Unterscheidung von Groß- und Kleinschreibung auf. Kompilierung im Hintergrund und Option für die spätere Bindung sind vorhanden. [web5]

<sup>1</sup> Nicht gesicherter Code bedeutet, dass auf Bereiche im Speicher direkt zugegriffen werden kann. Das Schlüsselwort „unsafe“ leitet solche Code-Abschnitte ein. (vgl. C++ „Pointer“) Beispiel einer unsafe-Methode in C#:  
`unsafe static void x (int *p)`

### 4.4.2.3 J#

J# ist die Microsoft Sprache, um für Java Entwickler einen leichten Einstieg in die .NET-Umgebung zu schaffen. Der Grundsyntax von Java und J# ist gleich, jedoch wird Java im sogenannten JRE (Java Runtime Environment) eingesetzt, während C# in CLR (Common Language Runtime) verwendet wird. [web6]

## 4.5 Microsoft .NET Compact Framework

### 4.5.1 Einleitung

Beim Microsoft .NET Compact Framework handelt es sich um eine abgespeckte Version des normalen .NET F. Das .NET CF wurde speziell für die Entwicklung von Anwendungen für den Einsatz sogenannter Smart Devices (wie z.B. Pocket PCs und Mobiltelefone) entwickelt, da diese weniger Speicherplatz aufweisen. Viele Klassen, die in .NET F vorhanden sind, fehlen im .NET CF komplett oder es fehlen Teile davon. Die Entwicklungsumgebung Visual Studio enthält das .NET CF, das z.B. auf einen PDA installiert werden kann. Außerdem enthält es noch SDE (Smart Device Extensions), mit deren Hilfe Geräte am PC emuliert werden können.

Das .NET CF kann genauso wie .NET F gratis von der Microsoft Website heruntergeladen werden.

### 4.5.2 Erweiterungen zu .NET Compact Framework

Da leider sehr viele Klassen im .NET CF fehlen (aus Gründen der Ressourcenbeschränkung von mobilen Geräten), werden oft Funktionalitäten benötigt, die in CF nicht enthalten sind. Werden solche Funktionen dennoch benötigt, können sie über das API auf Betriebssystemebene aufgerufen werden. Compact Framework unterstützt – wie auch die Desktopversion – den Plattformauffrufrdienst P/Invoke. Hiermit kann Code, der sich in dlls befindet, aufgerufen werden.

Eine weitere Möglichkeit ist die Verwendung externer Erweiterungen. Für diese Arbeit wurden folgende Erweiterungen verwendet:

#### 4.5.2.1 OpenNETCF

OpenNETCF erweitert das .NET CF mit nützliche Funktionen wie z.B. die kryptographischen Funktionen (*OpenNETCF.Security.Cryptography*). Auch sehr nützlich ist die Klasse *OpenNETCF.Desktop.Communication* mit deren Hilfe man auf ein angeschlossenes Device über *ActiveSync*<sup>1</sup> zugreifen kann (Klasse: *OpenNETCF.Desktop.Communication.RAPI*<sup>2</sup>), um z.B. Dateien zu kopieren oder Prozesse auf dem Smart Device zu starten.

#### 4.5.2.2 CoreLab MySQLDirect .NET Data Provider

Stellt Klassen und Bibliotheken zur Verfügung, um über ein Smart Device direkt auf eine MySQL-Datenbank zuzugreifen. Dieses Produkt ist kostenpflichtig und ist nach Durchführung von einigen Tests empfehlenswert. Einen freien Connector (basierend auf ADO.NET) gibt es nur für das normale .NET F von MySQL. [mysql1]

### 4.6 .NET für Microsoft-fremde Umgebungen

#### 4.6.1 Mono

Mono ist eine Entwicklungs- und Laufzeitumgebung für .NET. Es ist ein Open-Source-Projekt, und wird hauptsächlich von der Firma Novell betrieben. Mono hat das Ziel, eine NET kompatible Entwicklungsumgebung und einen C#-Compiler unter Windows, Linux, verschiedene UNIX-Derivate und Mac OS X zu schaffen. [web7] *Abb. 11* zeigt das Logo des Mono-Projektes.



Abb. 11: Logo des Mono-Projektes [web7]

<sup>1</sup> ActiveSync: Bei ActiveSync handelt es sich um ein Softwareprodukt von Microsoft, dass die Kommunikation zwischen einem mobilen Gerät und einem Desktop-PC via USB, COM und Infrarotport ermöglicht.

<sup>2</sup> RAPI: (*Remote Application Programming Interface*) ist eine Klasse, die es ermöglicht, mit dem Desktop PC bestimmte Funktionen des mobilen Gerätes (wie z.B. Batterieladezustand, Prozessorarchitektur) „remote“ also von der Ferne (via USB-Schnittstelle) abzufragen. Weiters können unter anderem auch Dateien auf den Handheld kopiert und Prozesse am PDA gestartet werden

#### 4.6.2 Portable.NET

Ziel von Portable.NET ist die Erstellung freier Software-Werkzeuge für die Ausführung und die Entwicklung von Programmen für die CLI. Dieses Produkt war früher nur unter GNU/Linux verfügbar, jetzt aber auch schon für Windows und MAC OS X. [web8]

#### 4.6.3 Microsoft: Project Rotor

Rotor ist eine Implementierung des CLI-Standards (standardisiert in der ECMA<sup>1</sup>) von Microsoft für verschiedene BSD<sup>2</sup>-Varianten. Der Quellcode ist öffentlich, und unterliegt der sogenannten *Shared Source Lizenz*<sup>3</sup>. [web9]

#### 4.6.4 Red Five Labs .NET Compact Framework für Symbian

Das .NET Compact Framework für Symbian wurde von der Firma Red Five (Sitz: Südafrika) entwickelt, um mobile Anwendungen (geschrieben für .NET) auf Symbian OS basierende Geräte lauffähig zu machen.

Es stellt die Zwischenschicht zwischen der .NET Mobilanwendung und dem Symbian Betriebssystem dar. [web10] Symbian ist ein Betriebssystem für Mobiltelefone und MDAs. Es hat große Verbreitung, da praktisch alle großen Mobiltelefonhersteller es verwenden. Auf Symbian baut auch das von Nokia entwickelte S60 (Series 60) auf. Praktisch fast alle großen Mobiltelefonhersteller verwenden oder verwendeten S60 wie z.B.: Siemens, Motorola, Samsung, Panasonic, Lenovo [web11]. Eine Ausnahme stellt Sony Ericsson (SE) dar, die teils eigene entwickelte Produkte einsetzen. Auf den MDAs von SE wird Symbian und als Benutzeroberfläche UIQ (User Interface Quartz) verwendet. UIQ gehörte früher zu Symbian, wurde jedoch von SE gekauft. [web12]

---

<sup>1</sup> ECMA ist eine private Normungsorganisation zur Standardisierung von Informations- und Kommunikationssystemen. [w17]

<sup>2</sup> BSD Berkley Software Distribution ist ein UNIX-Betriebssystem

<sup>3</sup> Sourcen stehen für Forschung und Lehre zur Verfügung. [ms2]

## 5 MySQL

### 5.1 Eigenschaften

Über MySQL gibt es sehr viele unterschiedliche Meinungen. Viele schwören auf MySQL und sagen dieses Produkt hänge in keinem Bereich kommerziellen Systemen hinterher. Andere bezweifeln sogar die Eigenschaft, dass MySQL eine relationale Datenbank sei.

Es stimmt, dass MySQL von immer mehr Anwendern/Anwenderinnen verwendet wird, obwohl noch immer wichtige Eigenschaften fehlen, die in anderen (kommerziellen) Systemen schon als selbstverständlich gelten. Die Konklusion hieraus ist, dass es darauf ankommt, was jemand benötigt, d.h. kann auf bestimmte Eigenschaften, die MySQL nicht bietet, verzichtet werden. Außerdem bietet MySQL Charakteristika wie Kostengünstigkeit, OpenSource und durch die Beliebtheit einen großen Umfang an Hilfestellungen in Foren. Im Webbereich hat MySQL einen sehr hohen Marktanteil.

#### MySQL... [vgl. Kofl]

- ...ist ein relationales Datenbanksystem,
- ...hat Client/Server Architektur d.h. es gibt einen Datenbankserver und beliebig viele Clients. Diese Clients (mit den Anwendungsprogrammen) können sich extern, z.B. irgendwo im WAN, aber auch direkt auf dem MySQL Server befinden. MySQL stellt hiermit *keine* Ausnahme dar. Alle gängigen DB-Systeme sind C/S-Systeme.
- ...unterstützt als Abfragesprache SQL (structured query language). Es gibt mehrere SQL-Dialekte. MySQL gehorcht momentan dem SQL:2003 Standard mit einigen Einschränkungen und eigenen Erweiterungen.
- ...unterstützt SubSelects ab Version 4.1.
- ...unterstützt Views ab Version 5

Des Weiteren können noch Stored Procedures, Trigger, Unicode, Replikationen, Transaktionen und viele weitere Funktionen verwendet werden.

Von MySQL werden bestimmte Funktionen nicht unterstützt, wie z.B. XML, benutzerdefinierte Datentypen, Echtzeitsysteme (OLAP), Hot Backup usw.

## 5.2 Alternativen

Kommerzielle Alternativen zu MySQL sind z.B. Microsoft SQL Server, Oracle, Informix, IBM DB2.

Es gibt auch freie Alternativen wie PostgreSQL (bietet einen höheren Funktionsumfang als MySQL, ist aber auch instabiler lt. [Kofl]). Für sehr kleine Datenbanken gibt es SQLite, und eine ehemals kommerzielle Lösung stellt Firebird (früher Interbase) dar.

## 5.3 Fazit

MySQL ist ein relationales Datenbanksystem, das als sehr leistungsfähig und relativ stabil angesehen werden kann. Besonders im Webbereich hat es sich schon als Quasi-Standard etabliert. Ein weiterer Vorteil ist die große Verbreitung und infolgedessen die gute Dokumentation sowie rasche Versionsupdates. MySQL kann auch mit einer großen Anzahl an Kundenreferenzen aufwarten: wie z.B. Dell, Sony Deutschland, SAS, Siemens, Alcatel, Cisco Systems, Sun Microsystems usw. Eine komplette Liste ist auf [web14] veröffentlicht.

## 6 Daten

Dieses Kapitel beschäftigt sich mit der Darstellung von Daten, der effizienten Suche, der Abfrage von Daten und mit Kundenfehlermeldungen und Anfragen.

### 6.1 Location Based Services

#### 6.1.1 Einleitung

Der Begriff „Location Based Services“ wird des Öfteren in Verbindung mit Systemen, die den Aufenthaltsort bestimmen und zum jeweiligen Aufenthaltsort Informationen liefern, verwendet.

*Beispiel:* Ein/e Tourist/in befindet sich in einer fremden Stadt und führt seinen/ihren PocketPC mit. Je nachdem wo er/sie sich befindet, erhält er/sie Informationen zum aktuellen Aufenthaltsort. Solche Informationen können Museen, Theater oder Plätze von touristischem sowie öffentlichem Interesse sein.

In diesem Kapitel wird die Verwendbarkeit dieser Services anhand des PDA-Info-Systems dargestellt. Die Frage, wie solche Services genutzt werden können, wird beantwortet.

#### 6.1.2 Location Based Services Funktion mit dem PDA-Info-System

Der/Die Mitarbeiter/in begibt sich zum Kunden und das mobile Gerät „weiß“ automatisch wo sich diese/r befindet.

Es gibt für diesen Fall mehrere Möglichkeiten den Standort zu bestimmen. Entweder über GPS, was allerdings hinsichtlich des GPS-Empfanges in geschlossenen Räumen problematisch sein kann, da Satelliten bedingt durch die geradlinigen, wenig durchdringenden Hochfrequenzen nur bei „freier Sicht“ erreicht werden können.

Die zweite Möglichkeit ist die Verwendung eines Dienstes eines Mobilfunkbetreibers. Solche Dienste werden unter dem Namen „Handy-Lokalisierung“ und dergleichen angeboten. Das Problem hierbei ist allerdings die Ungenauigkeit. Eine exakte Positionsbestimmung wie bei

GPS ist hierbei nicht gegeben. Die Ortung eines mobilen Gerätes über einen Mobilfunkbetreiber kann nur anhand der sogenannten CellID erfolgen. Jedes Mobilfunknetz ist in sogenannte Zellen aufgeteilt. In jeder Zelle befinden sich meist mehrere oder viele User. Der Mobilfunkbetreiber protokolliert in welcher Zelle sich der/die jeweilige Benutzer/in gerade befindet. Wenn sich sehr viele Kunden in demselben Stadtteil/Gebiet befinden und es wenige Zellen gibt, ist eine Zuordnung schwierig, da wie erwähnt, nur zellenweise die Position bestimmt werden kann.

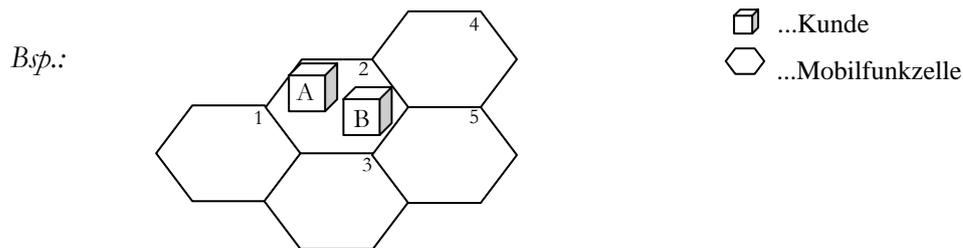


Abb. 12: Mobilfunkzellen mit Kundenstandorten

Nach *Abb.12* ist es hier nicht möglich festzustellen, welcher Kunde (*Kunde A* oder *Kunde B*) gemeint ist, da sich beide in derselben Zelle befinden. (Zelle 2)

Ein solcher Dienst muss beim Provider angemeldet und ein Vertrag unterschrieben werden, um dies ausdrücklich zu erlauben. Bei Missbrauch drohen Strafen im Sinne des Telekommunikationsrechts. Einen solchen Dienst bietet in Österreich z.B. das Mobilfunkunternehmen T-Mobile unter dem derzeitigen Namen „T-Mobile Fleet Manager“ an. Dieser Fleet-Manager gibt dann die Daten in Längen- und Breitengraden aus.

Anhand dieser Daten kann nun der jeweilige Kunde ermittelt werden. Die Lokalisierungsdaten (Längen- und Breitengrade Plus Minus Toleranz) sind mit den anderen Daten in der Datenbank gespeichert. Somit erfolgt eine exakte Zuordnung eines Geschäftsfalles zu einem Kunden.

Des Weiteren werden sämtliche Anlagen des Kunden aufgelistet. Der/Die Mitarbeiter/in braucht nur noch die defekte Anlage aus der Auswahl der verfügbaren Anlagen für den betreffenden Kunden auswählen. Wichtig ist hier natürlich eine sorgfältige Pflege des Datenbestandes. Wie die Aufzeichnungen hinsichtlich der Datenbank erfolgen, wird in *Abschnitt 6.2* behandelt.

## **6.2 Zu speichernde Daten**

### **6.2.1 Feld Art/Kategorie der Anfrage**

Um die Art der Berichte einzugrenzen, ist ein Feld mit dem Namen „Art des Fehlers/Berichtes“ vorgesehen. Es gibt verschiedene Möglichkeiten z.B.: Technical Query, Fault Report und Emergency)

#### **6.2.1.1 Technische Anfrage („Technical Query“)**

Handelt es sich um eine technische Anfrage, erfolgt diese wahrscheinlich telefonisch oder per Mail und kann dann vom/von der jeweiligen Innendienstmitarbeiter/in bearbeitet werden, da dies in den meisten Fällen nicht so dringend erfolgen muss. Der/Die Innendienstmitarbeiter/in erfasst die Daten auf der PC-Software in seinem/ihrem Büro oder am PDA.

#### **6.2.1.2 Fehlerbericht („Fault Report“)**

Bei dieser Art des Berichtes ist eine genaue Erfassung der Daten nach einem bestimmten Schema erforderlich, damit später eine Analyse der Häufungen von Fehlern einfacher erfolgen kann. Hierzu werden genau definierte Felder eingerichtet wie z.B. Kunde, Name/Art der Anlage, verwendete Software, bei welcher Aktion ist der Fehler aufgetreten und ist der Fehler reproduzierbar d.h. tritt er jedes Mal bei ein und derselben Aktion auf.

#### **6.2.1.3 Notfall („Emergency“)**

Ein Notfall kann eintreten, wenn beim Kunden ein Totalausfall der Anlage bzw. von bestimmten Features droht, oder dieser Ausfall bereits eingetreten ist. Hierbei ist eine sofortige Unterstützung und Behebung des Problems notwendig. Man stelle sich nur vor, was es für einen Mobilfunkbetreiber zum Beispiel bedeuten würde, würde das Charging<sup>1</sup> eines SMS-Dienstes österreichweit nur für ein paar Sekunden ausfallen. Massivste Umsatz- und Gewinneinbußen wären die Folge. Um bei solchen Fällen sofort zu reagieren, ist die Einrichtung einer 24 Stunden Hotline notwendig.

---

<sup>1</sup> Charging bezeichnet die Erfassung von Abrechnungsdaten eines Kunden. Anhand dieser Abrechnungsdaten erfolgt dann die Rechnungslegung.

### 6.3 Anfragen des Kunden

Dieser Abschnitt klärt die Frage, wie Kunden Fehler melden und Anfragen durchführen.

Ideal wäre ein System, in dem der Kunde direkt Eingaben tätigen kann und so den Fehler meldet. Der Kunde sieht eine Weboberfläche und führt seinen Fehlerbericht durch. Diese Weboberfläche kann auch auf der fehlerhaften Maschine laufen. (für den Fall eines Totalabsturzes der Maschine auch als Backup auf einem anderen Server) Der Kunde loggt sich auf diesem Webserver ein, sieht sein Benutzerkonto mit sämtlichen Maschinen und wählt jene aus, bei der der Fehler aufgetreten ist, und füllt noch sonstige Formulardaten aus. Nachdem der Kunde die Formulardaten abgeschickt hat, wird ein/e Mitarbeiter/in in Bereitschaft automatisch informiert. Die Daten, die der Kunde ausgefüllt hat, wie z.B. der Fehlerbeschreibung sowie die Marke/Type und Seriennummer des Gerätes werden in der Datenbank des PDA gespeichert. Sobald der/die Mitarbeiter/in zum Kunden fährt, sind bereits alle Daten am PDA eingetragen. Es kann eine Auswertung der Fehlerbeschreibung erfolgen, um dem/der Mitarbeiter/in gleich Lösungsansätze für dieses Problem anzuzeigen, falls genau dieses Problem zuvor schon aufgetreten ist. Wichtig ist hier, dass anhand eines Eintrages des Kunden (in einem Textfeld der Datenbank) die richtige Lösung des Problems aus der Datenbank herausgesucht wird und auch weitere Vergleiche kundenspezifisch durchgeführt werden. Schon während der Anreise zum Kunden kann sich der/die Mitarbeiter/in ein Bild darüber verschaffen und Aktionen überlegen.

In sehr dringenden Fällen kann der Kunde auch einen Button auf der Weboberfläche anklicken und initiiert somit einen Rückruf durch seine/n Betreuer/in.

#### 6.3.1 Beispiel eines Webformulars für eine Kundenmeldung

s. nächste Seite

# Kundenfehlermeldung<sup>1</sup>

Kunde: X

Kundennummer: 999999

Datum des Fehlers  
[JJJJMMTT]

Zeit des Fehlers  
[hh:mm]

**Aktuelle Zeit/Datum**

Anlage/Software

Ihr aktueller Servicetechniker ist:

Fehlerbeschreibung

[<Name des Technikers>](#)

Einen sofortigen Rückruf dieses Technikers einfordern hier klicken

**Rückruf**

Bei welchem Vorgang ist der Fehler aufgetreten?

**Absenden**

Abb. 13: Beispiel einer Kundenfehlermeldung - Webformular

<sup>1</sup> Fehlermeldungen werden auch häufig als „Bug Reports“, „Problem Reports“ oder „Change Requests“ bezeichnet.

## 6.4 Anpassung der Informationen an Endgeräte

### 6.4.1 Einleitung

Dieses Kapitel beschreibt die Möglichkeiten der Darstellung von Daten auf mobilen Geräten. Wegen der Begrenztheit der Darstellungsmöglichkeiten von mobilen Kleingeräten kann es schwierig werden, Daten – insbesondere Statistiken – gut aufbereitet und aussagekräftig auf einem PDA anzuzeigen.

Die meisten größeren und leistungstärkeren mobilen Geräte (Mobiltelefone werden hier nicht betrachtet, mit Ausnahme von MDAs) können z.B. Texte, Bilder und grafische Statistiken anzeigen. [Höpf]

### 6.4.2 Begriff der Informationsäquivalenz

#### 6.4.2.1 Definition von Informationsäquivalenz [Höpf, Veij]

„Ein Benutzer erhalte bei Verwendung von Schnittstelle  $S_i$   $I_i$  Einheiten Information in Zeit  $T_i$ . Eine zweite Schnittstelle  $S_j$  ist

- *äquivalent* zu  $S_j$  wenn  $I_i = I_j$  und  $T_i = T_j$ , also im selben Zeitraum dieselbe Information ausgetauscht wird,
- *schwach äquivalent* zu  $S_i$ , wenn zu jedem  $T_i$  ein  $T_j$  existiert, so dass  $I_i = I_j$ ; es wird also dieselbe Information ausgetauscht, dieser Austausch kann aber länger oder kürzer dauern,
- *stark äquivalent*, wenn sie äquivalent ist und der Benutzer beide Schnittstellen für gleichwertig hält.“

#### 6.4.2.2 Informationsäquivalenz im Bezug auf das PDA-Info-System

Nach dieser abstrakten Definition stellt sich die Frage nach dem Zusammenhang zwischen dieser Begriffsbestimmung und dem PDA-Info-System. Dieselben Daten sollen sowohl auf Desktop-PCs und Laptops als auch auf PDAs angezeigt werden können. Um dies zu ermögli-

chen ist *Informationsanpassung*<sup>1</sup> notwendig, da diese Geräte im Vergleich z.B. unterschiedliche Leistungen, Speicher, Bildschirm-, Displaygröße und Energieversorgung haben. Es ist auch erwähnenswert, dass es beim Vergleich des Informationsgehaltes um dieselbe Person geht. Der Grund hierfür ist folgender: Unterschiedliche Personen nehmen Informationen unterschiedlich auf, da jede Person eine andere Erfahrung, anderes Wissen, eine andere Vorbildung, andere Interessen usw. hat.

Zusammenfassend kann gesagt werden, dass immer darauf geachtet werden muss, dass Informationen auf unterschiedlichen Geräten *immer* den äquivalenten Informationsgehalt besitzen. Es ist vielleicht die Schriftgröße oder eine Grafik kleiner, aber die Informationen, die die Benutzer/innen daraus ziehen, sind dieselben. [vgl. web15]

#### **6.4.3 Ursachen der Anpassung [Höpf, Seite 126 u. 127]**

Bei einer Anfrage, in unserem Fall einer Suche eines Datensatzes oder Anzeigen einer Statistik oder Grafik, müssen die Möglichkeiten einer schnellen Erfassung der Daten durch den/die Benutzer/in gewährleistet werden. Auch ist die Verfügbarkeit von GSM oder UMTS eine wichtige Frage. Es stellen sich Fragen wie: Wie werden Daten dargestellt, wenn beispielsweise nur GSM verfügbar ist? Die Datenraten sind geringer, deshalb sollte bei Bildern und Tabellen die Auflösung geringer sein.

Ein weiterer wichtiger Aspekt ist der aktuelle Benutzerkontext [Höpf, Seite 127]. Außendienstmitarbeiter vor Ort bei einem bestimmten Kunden sind nur an Daten der installierten Anlagen bei genau diesem Kunden interessiert und nicht an anderen Daten.

---

<sup>1</sup> Informationsanpassung bedeutet hier, dass Informationen, die für einen Desktop-PC oder Laptop bestimmt sind, so umgewandelt werden, dass diese auch für einen PDA verwendet werden können.

#### 6.4.4 Konkrete Anpassungen für das PDA-Info-System

Da das Ziel möglichst *eine* Anwendung für alle Geräte (ob PDA, oder Laptop und dergleichen) ist, ist folgender Ansatz denkbar: Es werden in der Server-Datenbank eigene Spalten angelegt, die festlegen, ob bestimmte Daten am PDA, am Desktop oder an beiden Geräten angezeigt werden. Solche Daten sind z.B. Bildschirm-Hardcopies. Die Pfade dieser Hardcopies werden auch in dieser Tabelle gespeichert.

(z.B. in MySQL ein bit(1)-not null-Feld mit '0', oder '1' belegt) 0 für „nur PDA Anzeige“ und 1 für beide (sowohl Desktop als auch PDA)). Diese Einträge werden dann von der PDA-Software als Parameter verwendet.

Des Weiteren wird es notwendig sein, Suchanfragen an den aktuellen Aufenthaltsort (Kunden) zu koppeln, um zu viele Suchergebnisse zu vermeiden. Es könnten z.B. auch weiter zurückliegende Suchergebnisse nur verkürzt angezeigt werden. Unter „verkürzt“ versteht man einfach einen Teil der *Description*<sup>1</sup> eines Problemfalles wegzulassen. Problematisch wird es nur, wenn wichtige Teile der Description weggelassen wurden. In *Abb.14* ist ein längerer Beispieltext aus einer Datenbank angegeben.

```
To be forwarded to Signalware @ !! Goal is to get a SW that will
include the entry      N3 0 INF DEF 4711 1521 ln_daemon -hwq 1024 -
hrq 1024 -f /export/home/omni/conf/nodeConf.C7.99      => While
performing a scheduled activity (synchronizing IN nodes time),
which include restarting of both IN nodes, application was not
coming up and hence RTP still down.      We opened an emergency
based on Update Team recommendation and tac3 restarted the appli-
cation, and call started to work.      After we got the following
response from Germany.      Ulticom: ===== I have found the
cause of the problem. In fact it is due to a configuration error.
In the nodestart.N2.99 and nodestart.N3.99 files on chin2 the
ln_daemon command line is incorrect. It is:      N3 0 INF DEF 4711
1521 ln_daemon -hwq 1024 -hrq -f
/export/home/omni/conf/nodeConf.C7.99      instead of      N3 0 INF
DEF 4711 1521 ln_daemon -hwq 1024 -hrq 1024 -f
/export/home/omni/conf/nodeConf.C7.99      In other words, you are
missing the value of the read queue watermark. This is causing
ln_daemon on these two nodes to fail to start, since it misinter-
prets the command line. Consequently the node never becomes
\"ready\". Consequently the cestart file is not read by PM, since
as per the pm(1s) manpage it waits for the logical node to be
\"ready\" before starting processes in the cestart file.      This
```

<sup>1</sup> Die Description-Spalte speichert die Beschreibung eines Problems. Ausführlich wird darauf in *Abschnitt 8.3* eingegangen.

would not have been an issue when starting Signalware as Standby on chin2, since it was already running as Active on chin1. The problem occurs on chin1 after stopping and starting Signalware on this CE. I have run tests to double check this, and by changing the nodestart files on one CE to be incorrect as above, then after stopping and restarting Signalware on the other CE (which was originally Active), the applications in my cestart file are no longer started. The DF \"BAD\" messages are nothing to worry about. When one CE is stopped, in the meantime DF files on the other CE continue to be updated. When the CE, and thus DFdaemon, are started again, DFdaemon checks all the DF files to find which have changed and thus need to be synchronised. It prints out this message when it finds a file which needs to be synchronised. Thus this is an informational message and not an error message.

additional info from Ulticom: =====  
 \"I ought to mention that, after fixing the nodestart parameters on chin2 and restarting Signalware on this CE, you will still see the problem (that is, your applications will not start and you will be obliged to restart them with START-PROCESS). This is due to parameters being passed from the active CE, chin1. The problem will only go away once you have stopped and restarted Signalware also on chin1. I wonder if it may be possible to avoid the restarts by use of the STOP-NODE and START-NODE commands, or perhaps a SWITCH-CE, but in any case these commands are not recommended since they may have unpredictable side-effects. Clean restarts of the systems would be better. Do let me know if you need any more information on this. Meanwhile I will leave the SR in the \"Solution Provided\" state.\"

What to do now: =====  
 Please care for a correction of the mentioned files on the system like described above, afterwards both nodes should be restarted (one after the other) to be sure that the changes are available and active on both nodes. This should be done within an officially agreed and announced action using the project info process. Normally the missing entries are inserted during the first installation of the system (from Install Server) but in this case may there are historical reasons that they are missing.

===== Now we need to pass this information for Update team because we have a scheduled update on 15-16.November , so Node 1 is update according to above notes...

Abb. 14: Datenbankausschnitt eines MySQL-Datenbank-Feldes

Da dieser Text ziemlich lang und sehr unkomfortabel an einem Handheld zu betrachten ist, müssen Möglichkeiten geschaffen werden, die wichtigsten Textpassagen zu extrahieren. Diesem Problem widmet sich der folgende Abschnitt. (*Abschnitt 6.5*)

## 6.5 Effizientes Suchen mit Hilfe von WebCarousel

### 6.5.1 Begriff

Bei WebCarousel handelt es sich ursprünglich um eine Technik um Web-Suchanfragen effizient auf einem mobilen Gerät darzustellen. Grundproblem: Stellt jemand eine Suchanfrage auf einem Desktop-PC mit einer von ihm bevorzugten Suchmaschine (z.B. Google) zu einem populären Thema, so gibt diese Suchmaschine sehr viele Webseiten-Links als Ergebnis zurück. Da es sehr unübersichtlich ist, mit einem mobilen Gerät eine Vielzahl von diesen Seiten zu laden und zu betrachten, müssen andere Möglichkeiten geschaffen werden. Genau hier setzt WebCarousel an. Voraussetzung: Es sind hohe Übertragungsraten an das mobile Gerät möglich (3G-Netze<sup>1</sup>). WebCarousel setzt bei der Auswahl- und Darstellungsart an.

### 6.5.2 Funktionsweise und Entwicklung [vgl. web17]

WebCarousel wurde an der Universität Kobe in Japan von Akiyo Nadamoto, Hiroyuki Kondo und Katsumi Tanaka entwickelt. [web16]

Der User stellt eine Suchanfrage anhand der von ihm/ihr eingegebenen Schlüsselwörter. Ergebnis ist ein Karussell von Webseitenausschnitten, die nacheinander wiederholt angezeigt werden. Jeder Abschnitt besteht aus entweder einem repräsentativen Bild bzw. Video, dem Header der Website und einem Textausschnitt, oder aus allen drei Elementen.

Der User kann nun angeben wie sehr sich die gefundenen Informationen mit den vom User gewünschten Informationen decken. Der User gibt via Dialog an, ob die Informationen komplett unzutreffend für ihn/sie sind, ob er/sie auf der Suche nach differenzierteren Informationen ist, oder ob die angezeigten Informationen sehr ähnlich den gesuchten sind. Anhand dieser Angaben wird ein neues Karussell gebildet, und der Vorgang beginnt wieder von Neuen.

---

<sup>1</sup> 3G-Netz: Bei einem 3G-Netz handelt es sich um ein Mobilfunknetz der 3. Generation. UMTS (Universal Mobile Telecommunications System) ist ein solches 3G Netz. 3G Netze ermöglichen eine höhere Datenübertragungsraten als herkömmliche Netze wie z.B. GSM.

### 6.5.3 Ablauf des Auswahlalgorithmus [vgl. web17]

Es gibt *drei Komponenten* (Variablen  $C_H$ ,  $C_V$  und  $C_{PV}$ ) für den WebCarousel Algorithmus:

Suche im HTML-Code den Header (<TITLE ... </TITLE) der Webseite und speichere diesen in der Variablen  $C_H$ .

Bestimme kleinstes zusammenhängendes Textsegment, das alle zur Suche verwendeten Schlüsselwörter enthält. Dieses Textsegment kann dann als Stimmenwiedergabe ausgegeben werden. (Variable  $C_V$ , v steht für Voice).

Bestimme Bild/Video (falls vorhanden), das sich im kleinsten zusammenhängenden Textsegment befindet. Speichere Bild/Video in  $C_{PV}$ , PV steht für Picture/Video.

Jetzt werden die zehn besten Ergebnisse der Suchmaschine angezeigt. Danach tritt die Verfeinerung der Suche in Aktion. Es gibt jetzt mehrere Optionen die der Benutzer angeben kann: Benutzer möchte Seiten sehen die: a) ähnlich, b) unähnlich, c) genauer (also mehr in die Tiefe), d) oder ungenauer sind.

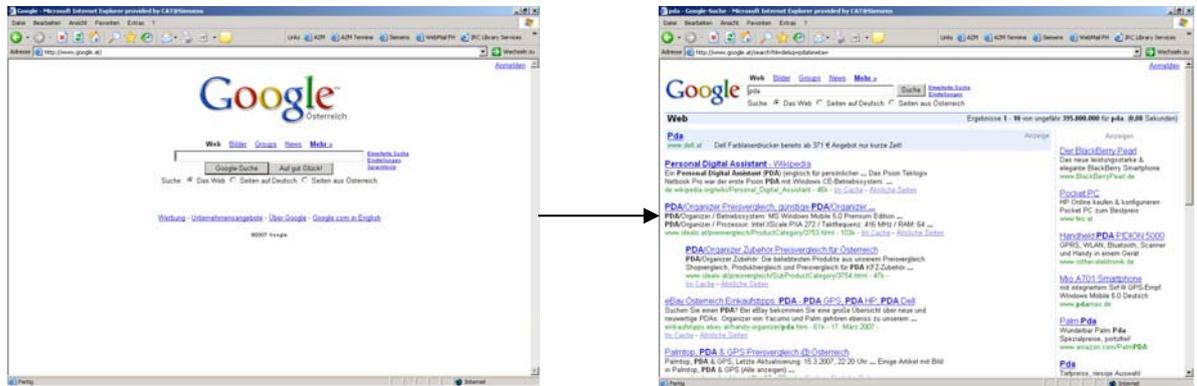
### 6.5.4 Einbau dieses Systems in das PDA-Info-System

Die Suche im PDA-Info-System funktioniert im Prinzip genauso wie eine Web-Suche, mit dem Unterschied, dass nicht im Web gesucht wird, sondern in der Datenbank des PDA-Info-Systems.

Die Software verbindet sich programmtechnisch über einen Connector (im getesteten Fall zu einer MySQL Datenbank über den CoreLab MySQL .NET Direct Data Provider) der auf ADO.NET Klassen beruht. (ADO.NET s. *Abschnitt 4.2.1*)

Es werden einfach SQL-SELECT Statements an die Datenbank gesendet und als Ergebnis werden sehr viele Datensätze zurückgeliefert. Diese Datensätze bzw. eine genau definierte Spalte einer Datensatzzeile wird jetzt durch das WebCarousel Interface am PDA behandelt und es kann dann die Ausgabe PDA-angepasst (mit den zehn Suchergebnissen) usw. erfolgen. Anstatt des PDA könnte WebCarousel auch am Server laufen. Die Daten werden gleich am Server aufbereitet.

Die Suchergebnisse werden dann vom WebCarousel-Interface in html-Code konvertiert und können dann auf einem Webbrowser am PDA angezeigt werden.



beliebige Suchmaschine

liefert Suchergebnisse

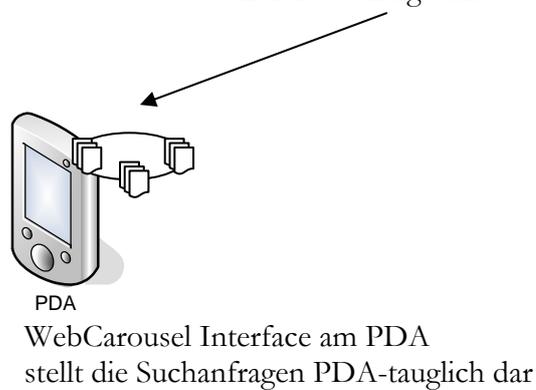


Abb. 15: WebCarousel Ablauf für Websuche

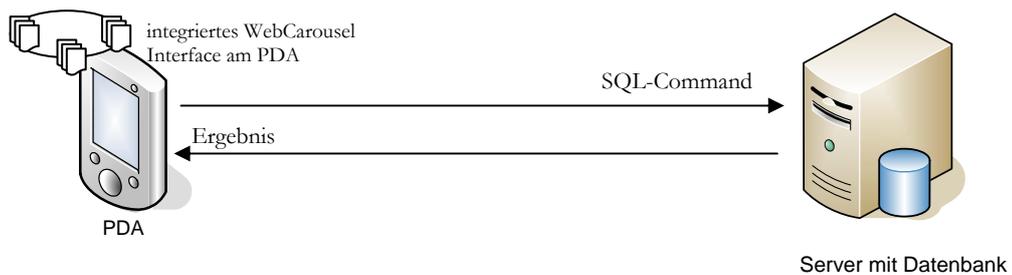


Abb. 16: WebCarousel Ablauf für die Suche in einer beliebigen Datenbank, WebCarousel Software/Interface am PDA

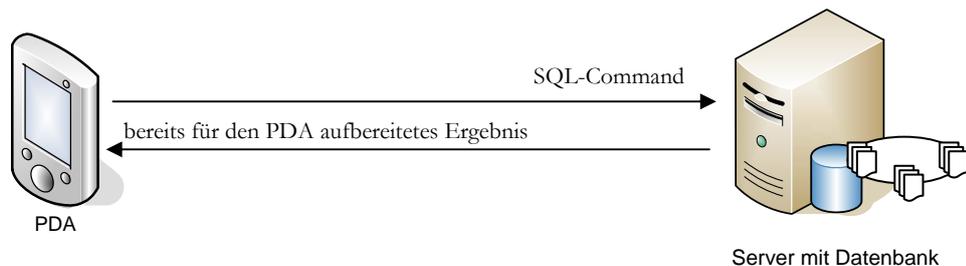


Abb. 17: WebCarousel Ablauf für die Suche in einer beliebigen Datenbank, WebCarousel Software/Interface am DB-Server

### 6.5.5 WebCarousel Interface

Ein solches Interface existierte meines Wissens zum Zeitpunkt der Erstellung dieser Arbeit noch nicht im Web.

### 6.5.6 Anleitung

- 1) Es wird ein SQL-Select Kommando abgesetzt, das mehrere Schlüsselwörter enthält, nach denen gesucht wird.
- 2) Das Ergebnis sind jene Datensätze, die die Schlüsselwörter enthalten.
- 3) Die Komponenten (s. *Abschnitt 6.5.3*) werden bestimmt, für die Datenbanksuche nur zwei, da es keine Bilder als Suchergebnisse gibt. Diese zwei Komponenten sind das Thema  $C_H$  (=Subject) und die Beschreibung  $C_V$  (=Description). Bei der Beschreibung wird die kleinste zusammenhängende Textpassage angezeigt, die alle angegebenen Schlüsselwörter enthält.
- 4) Die besten zehn Suchergebnisse werden angezeigt.
- 5) Nach Durchsicht der zehn Ergebnisse gibt es einen Dialog mit dem/der Benutzer/in. Der/Die Benutzer/in gibt an, wie die Auflistung für die Suche übereinstimmt. (s. *Abschnitt 6.5.3*)
- 6) Jetzt muss eine Messzahl mit folgender Formel berechnet werden:

$$F(R) = \sum_{k=1}^n \frac{1}{N} \cdot (f_k)$$

*F(R) = Messzahl für die Qualität der Suchergebnisse*

*R = laufende Nummer des Suchergebnisses*

*N = Anzahl der Wörter des gesamten Textes*

*f<sub>k</sub> = Häufigkeit des Schlüsselwortes*

*k = laufende Nummer des Schlüsselwortes*

*n = Anzahl der gesamten Schlüsselwörter, nach denen gesucht wird*

Formel F1: Vergleichsmesszahl berechnen

Mit dieser Formel erhält man nun einen Wert (F(R)) für den Vergleich von Suchergebnissen. Diese Formel beruht auf [w8, Folie 12]

### 6.5.6.1 Ähnlichkeit

Jetzt können zwei Suchergebnisse nach Ähnlichkeit verglichen werden:

$$\lambda = |F(R_1) - F(R_2)|$$

Formel F2: Ähnlichkeit berechnen

Für die Ähnlichkeit gibt es zwei Auswahlmöglichkeiten, nämlich „ähnlich“ oder „unähnlich“.

Wenn der/die Benutzer/in nun ein Suchergebnis betrachtet, und wählt „*unähnlich*“ aus, werden alle anderen (oder ein Teil) der anderen Suchergebnisse mit diesem verknüpft, und es werden jene Seiten ausgewählt bei denen  $\lambda$  möglichst nahe bei eins liegt, d.h. die Abweichung beider Suchergebnisse ist relativ groß. Wählt der Benutzer „*ähnlich*“ aus, dann muss  $\lambda$  möglichst nahe bei null liegen, wie aus den Formeln F1 und F2 hervorgeht.

### 6.5.6.2 Genauigkeit

Im vorigen Abschnitt wurde dargestellt, welche Formel man verwenden kann um die Ähnlichkeit von zwei Suchergebnissen zu bewerten.

Dieser Abschnitt beschäftigt sich mit dem Thema, wie man die Genauigkeit quantitativ vergleichen kann. Mit Genauigkeit ist die Tiefe gemeint, also wenn ein Schlüsselwort öfters vorkommt, kann daraus geschlossen werden, dass sich ein Ergebnis genauer mit dem gewünschten Schlüsselwort beschäftigt, s. folgendes Beispiel.

*Beispiel*

Suchwörter:

*Angabe:* Die Wortanzahl von Datensatz 1 beträgt 300, die von Datensatz 2 beträgt 433, Datensatz 3 beträgt 303.

Es soll entschieden werden ob Datensatz 2 oder Datensatz 3 angezeigt wird, nachdem der/die Benutzer/in angegeben hat, er/sie möchte Quellen sehen, die genauer als 1 sind. Auf Basis

von Datensatz 1 (in *Tab. 1* und *Tab. 2* hellgrau hinterlegt) wählt der/die Benutzer/in zwischen „genauer“ und „ungenauer“. Ist Datensatz 2 oder 3 genauer als 1?

Es wird nach folgenden Wörtern gesucht:

Wort 1, Wort 2, Wort 3

Vorkommen im Datensatz 1:	Vorkommen im Datensatz 2:	Vork. im Datensatz 3:
Wort 1: 5x	Wort 1: 11x	Wort 1: 6x
Wort 2: 1x	Wort 2: 1x	Wort 2: 1x
Wort 3: 7x	Wort 3: 19x	Wort 3: 8x

Tab. 1: Beispiel: Wortvorkommen in einem Text, WebCarousel Interface

Nach Formel F1 wird berechnet:

**F(R) für Datensatz 1:**

$$\frac{1}{300} \cdot 5 + \frac{1}{300} \cdot 1 + \frac{1}{300} \cdot 7 = 0,0433$$

$n=3$  (3 Schlüsselw.),  $N=300$ ,  $f_k=5, 1$  und  $7$

**F(R) für Datensatz 2:**

$$\frac{1}{433} \cdot 11 + \frac{1}{433} \cdot 1 + \frac{1}{433} \cdot 19 = 0,07159$$

$n=3$  (3 Schlüsselw.),  $N=433$ ,  $f_k=11, 1$  und  $19$

**F(R) für Datensatz 3:**

$$\frac{1}{303} \cdot 6 + \frac{1}{303} \cdot 1 + \frac{1}{303} \cdot 8 = 0,0495$$

$n=3$  (3 Schlüsselw.),  $N=303$ ,  $f_k=6, 1$  und  $8$

Laut Formel F2 müsste dem Datensatz 3 der Vorzug gegeben werden, wenn ähnliche Datensätze (zu Datensatz 1) gesucht werden, da:  $|0,0495-0,0433| = 0,0062$  und  $|0,07159-0,0433| = 0,02829$ . Die Subtraktion von Datensatz 3 - Datensatz 1 liefert einen Wert der näher bei Null ist, als Datensatz 2 – Datensatz 1.

Da jedoch angegeben wurde, dass die Inhalte mehr in die Tiefe gehen (Auswahl „genauer“), werden die Datensätze hinsichtlich folgender Frage verglichen: Welches Suchergebnis liefert die meisten Worte, bei denen die Wortanzahl im Text am höchsten ist.

Es wird hierfür folgende Formel benötigt:

$$R(n) = \frac{f}{N}$$

$f$  = Suchwörter im Text

$N$  = Anzahl der Wörter des gesamten Textes

Formel F3: Verhältnis  $R(n)$  zwischen Suchwortanzahl im Text zu der Gesamtzahl der Wörter

	Wort 1 R(n)	Wort 2 R(n)	Wort 3 R(n)	N
Datensatz 1	5x 0,01667	1x 0,00333	7x 0,02333	300
Datensatz 2	11x <b>0,02540</b>	1x 0,00231	19x <b>0,04388</b>	433
Datensatz 3	6x 0,01980	1x <b>0,00330</b>	8x 0,02640	303

Tab. 2: Berechnung von  $R(n)$

Datensatz 2 weist bei Wort 1 und bei Wort 3 den höchsten Wert für  $R(n)$  auf. Datensatz 3 nur bei Wort 2. Es kann deshalb gesagt werden, dass Datensatz 2 bei zumindest zwei Suchwörtern mehr in die Tiefe geht, Datensatz 3 nur bei einem. Datensatz 1 ist der Ausgangszustand.

Dieser Ansatz ist eine Adaptierung von [web18]. In diesem Ansatz wird der Abstand der Schlüsselwörter untereinander (noch) nicht berücksichtigt, d.h. wieviele Nicht-Schlüsselwörter stehen zwischen den Schlüsselwörtern. Dies macht bei einer Websuche die Suchmaschine. Da hier jedoch keine Suchmaschine involviert ist, müssen geeignete Techniken selbst entwickelt werden.

Wie diese Abstände theoretisch berücksichtigt werden könnten, beschreibt der nächste Abschnitt.

### 6.5.6.3 Abstände zwischen den Schlüsselwörtern berücksichtigen

Es werden sämtliche Wörter, die zwischen den Schlüsselwörtern liegen, gezählt. Jene Ergebnisse bei denen die wenigsten Nicht-Schlüsselwörter dazwischen sind, werden höher gereiht. Zusätzlich könnte noch der direkte Zusammenhang in einem Satz oder Absatz bevorzugt werden, d.h. größere Abstände (ungefähr größer als 25), werden nicht mehr

berücksichtigt, sondern nur kleinere Abstände. Es kann davon ausgegangen werden, dass ab einem gewissen Abstand die Schlüsselwörter nicht mehr im **direkten** Zusammenhang stehen.

z.B. *Schlüsselwörter*: Unix, Dateisystem, Datenbank, Error

2 *Suchergebnisse*:

1. Suchergebnis:					
Unix	.....Datenbank.....	Unix	Dateisystem.....	Unix.....	Error
	25	3	0	90	40
2. Suchergebnis:					
Datenbank...	.....	Unix	Dateisystem...	Datenbank...	Error.....Error
	212	0	2	3	7

Tab. 3: Beispiel: Abstände zwischen Suchworten, WebCarousel Interface

Anm.: Die Zahlen zwischen den Wörtern in *Tab. 3* geben die Anzahl der Wörter die zwischen den Suchwörtern (=Schlüsselwörtern) liegen wieder.

**Abstände Text 1:** 90, 40, 25, 3, 0      durchschn. Anzahl der Zwischenwörter = 31,60  
 $(90+40+25+3+0)/5$

**Abstände Text 2:** 212, 7, 3, 2, 0      durchschn. Anzahl der Zwischenwörter = 44,80  
 $(212+7+3+2+0)/5$

Text 2 wird höher gereiht, obwohl zwischen Datenbank und Unix ein Abstand von 212 besteht, und dadurch der Durchschnittswert höher liegt als bei Text 1. Es wird der Abstand zwischen „Datenbank“ und „Unix“ (212 Nicht-Schlüsselwörter) nicht berücksichtigt, da der Abstand der Schlüsselwörter laut unserer Festlegung größer ist als 25. Diese Festlegung könnte der Web-Carousel Algorithmus berücksichtigen.

Anhand der Ergebnisse der Abstandsberechnung können neben dem Ähnlichkeits- und Genauigkeitskriterium die Suchergebnisse zusätzlich verglichen werden.

*Anmerkung:* Es könnten noch viele andere Faktoren (wie z.B. Textanalysen nach Sprachen und Häufigkeiten, Analysen nach Satzzeichen, Wortvorkommen in gleichen Absätzen) durchgeführt werden.

## 7 Konfliktmanagement

In diesem Kapitel wird eine Lösung dargestellt, um Datenbanken von zwei unterschiedlichen Herstellern (wie hier der Fall) untereinander zu synchronisieren.

### 7.1 Transaktionen in MySQL

Mithilfe von Transaktionen wird sichergestellt, dass mehrere SQL-Kommandos, die an einen MySQL-Server übermittelt werden, entweder vollständig (alle Kommandos) oder gar nicht (kein Kommando) ausgeführt werden.

#### 7.1.1 Anwendungsfall einer Transaktion im PDA-Info-System

Im Folgenden wird erklärt, wo eine Transaktion im PDA-Info-System benötigt wird. Wie bereits erläutert, müssen sich alle Geschäftsfälle (=Problemfälle/Fehlerfälle beim Kunden) spätestens bei Abschluss dieses Geschäftsfalles auf der MySQL-Datenbank befinden. Teilweise können sich noch neue Datensätze (= noch nicht auf der MySQL-Server Datenbank vorhanden) oder heruntergeladene alte Datensätze (= schon auf der Server Datenbank vorhanden) am PDA befinden.

CNumber	CaseName	Subject	Description	State	Type	Date	MySQL-CNumber	Version
56	Customer A	Query 1	Description 1	OPENED	TECQ	10-05-2005		1
... [weitere Zeilen ausgelassen]								
70	Customer L	Fault Report 47	Description 55	OPENED	FREP	26-03-2007		1

Tab. 4: Datenbanktabelle einer PDA-Datenbank



PDA Datensätze der MySQL DB hinzufügen

CNumber	CaseName	Subject	Description	State	Type	Date	Version
52683	Customer A	Query 1	Description 1	OPENED	TECQ	10-05-2005	1
... [weitere Zeilen ausgelassen]							
52697	Customer L	Fault Report 47	Description 55	OPENED	FREP	26-03-2007	1

Tab. 5: Datenbanktabelle einer MySQL-Datenbank

Nachdem die INSERTs auf MySQL durchgeführt wurden, muss die SQL Server CE DB am PDA auch aktualisiert werden, um sich die Datensätze auch am PDA zu merken, die bereits in MySQL vorhanden sind. (s. Tab.6)

CNumber	CaseName	Subject	Description	State	Type	Date	MySQL-Number	Version
56	Customer A	Query 1	Description 1	OPENED	TECQ	10-05-2005	52683	1
... [weitere Zeilen ausgelassen]								
70	Customer L	Fault Report 47	Description 55	OPENED	FREP	26-03-2007	52697	1

Tab. 6: PDA-Datenbank-Update nach der Änderung in MySQL

Die einzige Änderung, die in diesem Fall erfolgt, ist der Eintrag in der Spalte MySQLCNumber. Hier merkt sich die PDA-Datenbank die CNumber der MySQL Datenbank, um den Datensatz später in der MySQL-Datenbank wieder zu finden.

Der folgende beispielhafte Ablauf zeigt, wozu beim PDA-Info-System eine Transaktion benötigt wird:

- 1) Ein/e Servicemitarbeiter/in befindet sich beim Kunden und speichert auf der Microsoft SQL Server CE Datenbank am PDA einen Datensatz.
- 2) Jetzt möchte er/sie den Datensatz auf der MySQL Server DB eintragen. Hiefür wählt er/sie in der PDA-Software den Menüpunkt „Synchronize“.
- 3) Der Datensatz wird auf die MySQL-Server Datenbank übertragen.
- 4) Die MySQL-Datenbank liefert die *MySQLCNumber* nach dem Eintrag zurück.
- 5) Jetzt ereignet sich ein Fehler. Der PDA erhält die MySQLCNumber nicht. Sie kann am PDA in der Datenbank nicht eingetragen werden. (s. Abb. 18)
- 6) Der/Die Mitarbeiter/in tippt am PDA nochmals auf „Synchronize“.
- 7) Durch den im Punkt 5) ereigneten Fehler würde bei einer weiteren Synchronisation der Datensatz ein zweites Mal in der MySQL Server DB eingetragen, da die PDA-Software nicht „weiß“ – wegen der fehlenden MySQLCNumber – dass der Datensatz bereits am Server eingetragen wurde. Eine Transaktion vermeidet einen solchen Fehler.

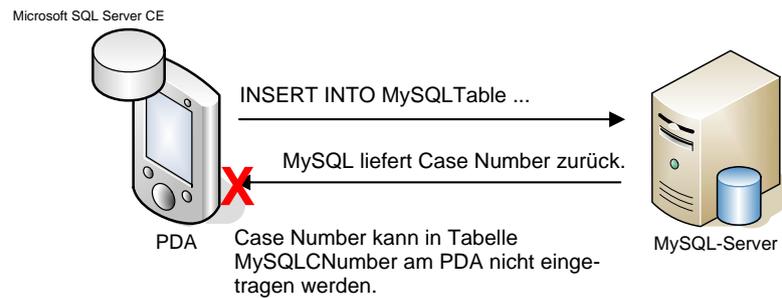


Abb. 18: Beispiel eines Fehlers bei der Synchronisation

Eine solche Transaktion zeigt *Abb. 19* in Form eines C# Code-Ausschnittes. *[ausgelassen]* bedeutet in *Abb.19*, dass bestimmte Code-Teile aus Gründen der Übersichtlichkeit weggelassen wurden.

```
using [ausgelassen] //Klassenimporte (z.B.: Windows.System.Forms...)

//Dieser Methode werden die ConnectionStrings für beide Datenbanken als
//Parameter übergeben. ConnectionStrings enthalten verbindungsrelevante
//Daten für die Datenbank wie z.B. Server-IP, Username, Passwort
private void synchronizeMySQL(string mySqlConnStr,
    string sqlCeConnStr)
{
    //Verbindung zur MySQL-DB erzeugen und öffnen
    MySqlConnection con = new MySqlConnection(string mySqlConnStr);
    con.Open();

    //Verbindung zur Datenbank am PDA (SqlServerCe) erzeugen und öffnen
    SqlCeConnection conCe = new SqlCeConnection(string sqlCeConnStr);
    conCe.Open();

    //Transaktionen beginnen
    MySqlTransaction tn = con.BeginTransaction();
    SqlCeTransaction tnCe = conCe.BeginTransaction();

    try {
        //Hier stehen die Anweisungen, die komplett ausgeführt
        //werden müssen
        MySqlCommand com = new MySqlCommand("INSERT INTO MySQLTable
            [ausgelassen]", con)
```

```
//Kommando ausführen
com.ExecuteNonQuery();

SqlCeCommand comCe = new SqlCeCommand("UPDATE SqlCeTable SET
    MySqlCNumber=[ausgelassen]", conCe);

//Kommando ausführen
comCe.ExecuteNonQuery();

//Ende der Transaktionsausführung
tn.Commit();
tnCe.Commit(); }

catch (Exception ex) {
    //Bei einem Fehler werden die Transaktionen sowohl auf der
    //MySQL-Datenbank als auch auf der SqlCe-Datenbank am PDA
    //rückgängig gemacht.
    tn.Rollback();
    tnCe.Rollback();

    //Der Fehler wird in Form einer Windows MessageBox
    //angezeigt,
    MessageBox.Show("Error: PDA-Info-System", "Synchronization "
        + "failed!", MessageBoxButtons.OK, MessageBoxIcon.Error,
        MessageBoxDefaultButton.Button1);

    //und noch die Exception ausgegeben
    MessageBox.Show(ex.ToString()); }
```

Abb. 19: C# Code-Snippet: Transaktion zur Synchronisation der MySQL-Datenbank mit der PDA-Datenbank

Mit dem Code aus *Abb.19* ist sichergestellt, dass entweder *tn* und *tnCe* oder keines der beiden ausgeführt wird.

## 7.2 MySQL Update

Dieser Abschnitt erklärt, wie gemeinsame Zugriffe und Änderungen auf einen Datensatz behandelt werden. Anhand eines Anwendungsfalles lässt sich dies am einfachsten erläutern.

**Anwendungsfall, Ausgangssituation:**

*Mitarbeiter/in A* lädt sich einen bereits auf dem Server vorhandenen Datensatz auf den PDA zur weiteren Betrachtung und Nachbearbeitung. *Mitarbeiter/in B* tut das gleiche mit derselben Intention. Beide Mitarbeiter/innen haben exakt denselben Stand („Stand X“) vom Server. Sie führen nun die von ihnen gewünschten Änderungen durch. *Mitarbeiter/in A* ändert Stand X auf *Stand A*, und *Mitarbeiter/in B* ändert Stand X auf *Stand B*. Beide speichern diese Stände nun auf ihre jeweiligen PDAs.

Dieser Fall wirft folgende Frage auf: Wie soll bei einem Update auf die MySQL Server Datenbank verfahren werden?

*A) Ablauf:*

Klickt *Mitarbeiter/in A* früher auf „Update“ als *Mitarbeiter/in B*, wird auf jeden Fall einmal der Stand des/der Mitarbeiters/Mitarbeiterin A auf dem Server gespeichert. Klickt dann *Mitarbeiter/in B* auf „Update“, bekommt diese/r eine Meldung, dass sein/ihr heruntergeladener *Stand X* nicht mehr aktuell ist. Er/Sie muss sich dann den aktuellen Stand (*Stand A*) auf den PDA laden. Damit die vom/von der Mitarbeiter/in B erfolgten Änderungen („*Stand B*“) nicht verloren sind, werden diese Änderungen in einem Textfile am PDA gespeichert. Erst jetzt kann *Mitarbeiter/in B* (auf Basis des Standes A von Mitarbeiter/in A) Änderungen am Datensatz durchführen. Wenn gewünscht, kann Mitarbeiter/in B den Text des Textfiles („Stand B“) mit Hilfe eines Menüpunktes einfach in den Datensatz einfügen oder „Stand A“ neu bearbeiten.

Diese Vorgangsweise wird einfach anhand der Spalte „Version“ auf der Server-DB ermöglicht. Erfolgt ein MySQL-Update wird die Versionsnummer um eins erhöht. Stimmt die Versionsnummer der Server-DB nicht mehr mit jener der PDA-DB überein, hat zwischenzeitlich schon ein/e andere/r Mitarbeiter/in den Stand des betreffenden Datensatzes aktualisiert hat, wie im vorliegenden Beispiel dargestellt. Bei Änderungen am PDA wird die Versionsnummer in der Datenbank am PDA nicht erhöht. (s. nachfolgendes Beispiel)

*B) Beispiel:*

Der/Die Mitarbeiter/in...

...speichert den *Datensatz 1* am PDA mit der *Versionsnummer 1*.

...synchronisiert. *Datensatz 1*. In der Server DB steht jetzt *Datensatz 1* mit der *Versionsnummer 1*.

Nach einiger Zeit fährt der/die Mitarbeiter/in wieder zum Kunden und möchte Änderungen vorerst nur am PDA vornehmen.

Der/Die Mitarbeiter/in...

...ruft *Datensatz 1* zur Nachbearbeitung auf und macht diverse Änderungen.

...speichert diese Änderungen vorerst nur am PDA. Die Versionsnummer am PDA wird in diesem Fall **NICHT** erhöht. Die MySQL DB bleibt unbeeinflusst.

Am Ende des Tages tippt der/die Mitarbeiter/in am PDA in der PDA-Software auf „Synchronize“. *Datensatz 1* wird auf der MySQL Server DB aktualisiert und die *Versionsnummer* wird auf 2 erhöht. Auf der PDA DB wird die *Versionsnummer* nach der Synchronisation ebenfalls auf 2 erhöht. Nach der Synchronisation hat der *Datensatz 1* in beiden Datenbanken dieselbe Versionsnummer. Unterschiede aus den Versionsnummern können sich nur ergeben, wenn ein/e weitere/r Mitarbeiter/in den Datensatz in der MySQL Server DB aktualisiert hat. (s. *Punkt A*) in diesem Abschnitt).

## 8 Effiziente Datenaufzeichnung und Analyse

### 8.1 Einführung

Dieses Kapitel widmet sich der Vorgangsweise um eine effiziente Datenaufzeichnung zu ermöglichen. Ein großes Problem stellen Duplikate, also doppelte Einträge die denselben Fehler behandeln, dar. Weiters wird eine Möglichkeit dargestellt, wie die am besten geeigneten Mitarbeiter/innen für die Bearbeitung eines Fehlerfalles ausgewählt werden können.

Am Ende dieses Kapitels wird noch ein Desktop Analyse- und Erfassungstool skizziert.

### 8.2 Mitarbeitertabelle

Ein Teil der Datengrundlage ist der/die *Mitarbeiter/in*, der/die den Fall bearbeitet hat. Zu diesem Zweck gibt es eine eindeutige *Mitarbeiterkennung* am PDA, mit der sich jede/r Mitarbeiter/in am System identifiziert.

Als nächstes werden Daten für die *genutzte Zeit* bis zur Lösung des Falles benötigt. Diese Daten müssen vom/von der Mitarbeiter/in oder dessen/deren Vorgesetzten festgestellt werden.

Ziel ist, eine Art Bewertungssystem einzuführen, mit welchem der/die Mitarbeiter/in subjektiv bewertet, wie einfach oder schwierig die Lösung des Problems für ihn/sie war. In diesem Bewertungssystem gibt es drei Stufen: „schwierig, neutral oder einfach“. Der/Die Mitarbeiter/in bewertet seinen/ihren abgewickelten Fall selbst. Für diese Bewertung fließen folgende Punkte ein: Zeitaufwand für die Problemlösung, Vorbereitungszeit, Fehlersuche und Komplexität des Problems. (Wieviel fachspezifisches Know-How benötigt jemand für die Lösung dieses Problems).

*Fehler, die ein solches Bewertungssystem zwangsläufig mitbringt:* Anhand dieser Daten könnten auch Auswertungen hinsichtlich des Know-Hows und der Produktivität des/der Mitarbeiters/Mitarbeiterin durchgeführt werden. Dies könnte dazu führen, dass er/sie hinsichtlich der Beurteilung des Problems nicht ehrlich ist, d.h. er/sie gibt zu häufig die Bewertung „einfach“ ab (um anzuzeigen, wie viel Know How er/sie hat), bzw. ein/e andere/r Mitarbeiter/in gibt für ein ähnliches Problem die Beurteilung „sehr schwierig“ ab (um seine/ihre Notwendigkeit

in einem bestimmten Projekt zu suggerieren, und vorzugeben, dass lange Schulungszeiten für ein bestimmtes System für neue Mitarbeiter/innen notwendig wären), obwohl dies für ihn/sie einfach zu lösen war. *Lösung:* Sämtliche Werte der *Mitarbeiter-Case-Tabelle* werden verschlüsselt abgelegt. Vorgesetzte oder unbefugte Personen können so die Werte nicht interpretieren.

*Die Mitarbeiter-Case-Tabelle besteht aus folgenden Spalten:*

z.B.:

Mitarbeiter-Nr.	Anlage/Software	Modulnummer	Sollzeit[d]	Istzeit [d]	Bewertung MA	Beurteilung Kunde	Ranking
134	354	664	3,5	5	1	satisfied	0,5
246	354	664	3,5	4,5	1	dissatisfied	-1

Tab. 7: Mitarbeitertabelle für zwei Mitarbeiter

Ranking bei Mitarbeiter-Nr. 134: -1,5 für die Abweichung Soll-Ist-Zeit. +1 für die Bewertung des/der Mitarbeiters/Mitarbeiterin als einfach (Skala von -1 bis +1, +1 ist einfach 0 ist neutral, -1 ist schwierig), und +1 für die Kundenzufriedenheit. (satisfied=1, dissatisfied=-1). Anm.: Wieso bewertet der/die Mitarbeiter/in 134 den Fall als einfach, liegt aber trotzdem über der veranschlagten Zeit? Es könnte die Sollzeit einfach zu kurz veranschlagt worden sein, oder er/sie hatte eine falsche Selbsteinschätzung.

Nach obiger Tabelle wird dem/der Mitarbeiter/in 134 der Vorzug für diese Anlage und dieses Modul gegeben, obwohl diese/r einen halben Tag länger für die Lösung benötigt hat. Somit entsteht mit dem Wachsen dieser Tabelle eine immer breitere Basis für die Entscheidungsfindung. Es könnte nach einiger Zeit z.B. wieder einen Eintrag des/der Mitarbeiters/Mitarbeiterin 134 geben, wo diese/r sehr schlecht abschneidet, und dann unter Umständen ein/e andere/r Mitarbeiter/in zum Zug kommt. Es wird bei mehreren Einträgen derselben Anlage und desselben Moduls der *Durchschnittswert* des *Ranking-Wertes* verwendet. Hat Mitarbeiter/in 134 einmal Ranking 0,5 dann Ranking -4,5, ergibt das ein Ranking von -2. Somit wäre Mitarbeiter/in 246 dann besser und würde für diesen Fall bevorzugt.

Voraussetzung für die Mitarbeiter/innenvorauswahl ist, dass Daten wie Anlage/Software und die Modulnummer zur Verfügung stehen.

Je mehr Daten erfasst wurden, desto leichter ist es, eine/n Mitarbeiter/in für einen bestimmten Geschäftsfall einzusetzen. *Fazit:* Das Bewertungssystem dient als Entscheidungsgrundlage, um Mitarbeiter/innen besser für einen bestimmten Geschäftsfall auszuwählen.

Interpretiert werden die Daten folgendermaßen: Je öfter der/die Mitarbeiter/in bei einer bestimmten Anlage gearbeitet hat, je öfter er/sie die Arbeit laut seiner/ihrer eigenen Beurteilung als einfach klassifiziert hat und diese auch erfolgreich abgeschlossen hat, je öfter der Kunde damit zufrieden war, desto eher wird er/sie genau für diesen Kunden und diese Anlage wieder ausgewählt werden.

### **8.2.1 Kundenfeedback**

In der Praxis ist es schwierig, ein Kundenfeedback zu erhalten, da es für den Kunden mühsam ist, Fragebögen auszufüllen oder etwa telefonische Fragen zu beantworten. Der Kunde nimmt sich meist nicht die Zeit. Der Zeitaufwand muss daher so gering wie möglich gehalten werden. Der Kunde soll nach Abschluss eines Geschäftsfalles (=Case) einfach eine Gesamtbeurteilung über ein Webformular durchführen. Diese Gesamtbeurteilung ist möglichst einfach zu gestalten. (eventuell vier Stufen: äußerst zufrieden, zufrieden, mäßig zufrieden, nicht zufrieden). Alle hier angeführten Informationen werden auf der Datenbank in der sogenannten *Mitarbeiter-Case-Tabelle* abgelegt: Mitarbeiternummer, zugeordneter Case, Sollzeit für die Fehlerlösung, Istzeit für die Fehlerlösung, Bewertung des/der Mitarbeiters/Mitarbeiterin, Beurteilung des Kunden).

### **8.3 Problembereich – Beschreibung des Problems**

Der Text des Problembereiches befindet sich im Feld „Description“ in der Datenbank. Die Logfiles<sup>1</sup> können von der fehlerhaften Software generiert oder vom/von der Mitarbeiter/in oder vom Kunden (Website, Kundenfehlermeldung) in die Datenbank eingetragen werden.

---

<sup>1</sup> In Logfiles (=Protokoll) werden Ereignisse eines Softwareproduktes aufgezeichnet.

Jeder Problebericht sollte folgende Punkte beinhalten: [Zeller]

- *Produkt-Release*: Das ist die Versionsnummer des Produktes oder ein anderer Identifikator.
- *Das Betriebssystem* inkl. der Version.
- *Die Systemressourcen*: Zu den Systemressourcen gehören die Größe des RAM-Speichers und die Kapazität der Festplatte(n).
- *Der Problemverlauf*: Der Problemverlauf gibt an, wie das Problem entstanden ist. Es werden nur jene Schritte angegeben, die unmittelbar zum Fehler führten.
- *Eine Beschreibung, was geschehen hätte sollen*: Beispiel: Das Programm sollte die Rufnummer der Person X ausgeben.
- *Eine Beschreibung des Fehlereignisses*: Beispiel: Das Programm verursachte folgenden Fehler: `MySQLException at Line 126 in class Connector`.
- *Eine einzeilige Zusammenfassung*: Beispiel: Dieser Programmabsturz entstand durch die Rufnummernabfrage eines Kunden bei der Eingabe des Namens.

Es ist darauf zu achten, dass die Privatsphäre des Anwenders nicht angetastet wird.

### 8.3.1 Probleme klassifizieren

Es gibt folgende Kriterien [Zeller]:

*Wichtigkeit und Priorität des Problems*: Dies wird anhand der Berichtsarten (*Abschnitt 6.2*) gespeichert.

*Identifikator*: Der Identifikator ist die Case Number.

*Comments*: Der Administrator oder Kunde kann dem Bericht Kommentare hinzufügen. Entweder unter „Description“ oder es wird ein weiteres Datenbankfeld angelegt.

*Notification:* Ein/e Administrator/in oder Kunde kann Mail-Adressen hinzufügen. Jedes Mal wenn sich der Status des Problembereichs ändert (s. *Abschnitt 8.3.2*), bekommt der Kunde eine Mail mit dem aktuellen Bericht, z.B. nach Behebung des Fehlers.

### 8.3.2 Problemstatus

Ein Status bezeichnet den aktuellen Lebenszyklus des Problems.

Folgende Bezeichnungen werden für das PDA-Info-System verwendet: [Zeller]

- UNCONFIRMED: Das Problem wurde erfasst (vom Kunden oder einem/einer Mitarbeiter/in), aber noch nicht zur Lösung freigegeben.
- ASSIGNED\_NEW: Das ist ein neues Problem, kein dupliziertes, bereits bekanntes Problem. Es wurde in diesem Status bereits dem/der Fachmitarbeiter/in zur Lösung übergeben.
- ASSIGNED\_DUPLICATE: Ein Problem wurde erfasst, das es so schon einmal gab.
- WONTFIX: Dieses Problem wird nicht gelöst werden, da es ein integraler Bestandteil des Systems ist. Es handelt sich um ein „Feature“ nicht um ein „Failure“.
- FIXED: Das Problem wurde vom/von der Tester/in oder Administrator/in als gelöst befunden.
- VERIFIED: Das Problem ist gelöst und wurde auch als gelöst vom Kunden bestätigt.
- CLOSED: Das Problem wurde durch ein neues Software-Release gelöst. Dieses Release wurde dem Kunden schon zugestellt. Nach der Zustellung des Produktes kann dieses Problem des alten „Releases“ auf CLOSED gesetzt werden.

### 8.3.3 Mehrfach vorhandene Fehlereintragungen (Duplikate)

Sind Produkte lange im Einsatz, entstehen eine unüberschaubare Anzahl an Problemfällen, mit sehr umfangreichen Log-Texten und zusätzlichen Anmerkungen von Mitarbeitern/Mitarbeiterinnen und Kunden. Problemfälle wiederholen und überschneiden sich teilweise, da Fehler häufig mehrmals auftreten. Es entstehen sogenannte *Duplikate*.

Eine teilweise Lösung dieses Problems ist, Duplikate zu identifizieren. Der Kunde oder der/die Mitarbeiter/in sucht zuerst in der Datenbank nach Stichworten. Erst wenn kein entsprechender Eintrag gefunden wurde, legt der Kunde oder der/die Mitarbeiter/in einen neuen Eintrag an. Beide haben außerdem die Möglichkeit, Duplikate zu kennzeichnen. Leider ist es in der Praxis nicht immer einfach, doppelte Einträge zu identifizieren. Der Schlüssel liegt hier in der Vereinfachung des Problems (*engl. Simplifying Problems, s. 8.3.4*). Es soll ein detaillierter Problembericht (der viele unterschiedliche Faktoren enthält) in einen *Testfall* (*engl. test case*) übergeführt werden. Ein Testfall beinhaltet immer nur die relevanten Details. Ein Detail ist relevant, wenn durch dieses Detail das Problem entsteht. [Zeller] *Beispiel*: Simulation eines neuen Autos an einem Simulator. Das Auto kommt ohne ersichtlichen Grund zum Stehen. Es werden Tests, unter geänderten Umständen (*engl. circumstances*) durchgeführt: 1) Die Sitze werden ausgebaut und die Passagiere müssen aussteigen. Selbes Ergebnis: Das Auto kommt zum Stehen. 2) Der Motor wird ausgebaut. Das Auto startet erst gar nicht und kann den Parkplatz nicht verlassen. *Schlussfolgerung*: Die Sitze sowie die Passagiere sind für die Lösung des Problems irrelevant. Der Motor ist offensichtlich relevant! Der nächste Schritt wäre nun, sich mit dem Motor zu beschäftigen.

Andreas Zeller<sup>1</sup> führt in seinem Buch „Why Programs fail“ folgenden Satz an, der dieses Problem beschreibt:

*„For every circumstance of the problem, check whether it is relevant for the problem to occur. If it is not, remove it from the problem report or the test case in question.“* [Zeller]

---

<sup>1</sup> **Andreas Zeller** (geb. 28. Oktober 1965 in Hanau) ist Professor und Leiter des Lehrstuhls für Softwaretechnik an der Universität des Saarlandes in Saarbrücken.

Jeder Umstand des Problems wird überprüft, ob er relevant für das entstandene Problem ist. Wenn er dies nicht ist, entferne ihn aus dem Problembereich oder dem Testfall. Ein Umstand ist jeder Aspekt, der das Problem beeinflussen könnte.

### 8.3.4 Vereinfachung des Problems

Im Abschnitt 8.3.3 wurde erklärt, dass für die Suche nach Duplikaten detaillierte Problemfälle in sogenannte Testfälle übergeleitet werden. In diesem Abschnitt geht es um die Art und Weise, wie dies erfolgen kann.

Es gibt grundsätzlich zwei Arten:

- Manuelle Simplifizierung (*engl. Manual Simplification*): Bei der manuellen Vereinfachung wird der Log-Text Zeile für Zeile „händisch“ durchgegangen.
- Automatische Simplifizierung (*engl. Automatic Simplification*): Die Vereinfachung erfolgt durch Tools und bestimmte Algorithmen. (s. Abschnitt 8.3.4.1)

#### 8.3.4.1 Simplifizierungsalgorithmus

Ziel dieses Algorithmus ist, die minimale Anzahl von Umständen festzustellen, bei welchen ein Fehler bei einem Programmablauf auftritt. [Zeller, ab Kapitel 5)

Ein solcher Algorithmus existiert unter dem Namen *dadmin* entwickelt am Lehrstuhl für Softwaretechnik der Universität des Saarlandes. *dadmin* gehört zu *delta debugging*. *delta debugging* ist ein Ansatz, um Fehlerursachen durch Unterschiede (*deltas*) nach dem *trial and error* Prinzip zwischen Programmläufen festzustellen.

*Funktion des Algorithmus:*

Es ist eine *Testfunktion* (*test c*) mit einem oder mehreren Parameter(n) *c* gegeben. Es werden der Reihe nach die zu testenden Parameter (Benutzereingaben) der Testfunktion übergeben. Als Ergebnis gibt es drei Möglichkeiten: der Test ist fehlgeschlagen („*fail*“), der Test war erfolgreich („*pass*“) oder der Test hat ein undefiniertes Ergebnis zurückgegeben („*unresolved*“).

Erläuterungen:

. („\“ ... Komplementärmenge, gegeben sind zwei Mengen A und B.

$A \setminus B = \{ x \in A \mid x \text{ ist nicht Element von } B \}$  )

$c$  wird in zwei Teile geteilt ( $c_1$  und  $c_2$ ).  $c_g$  ist die Gesamtmenge der Parameter.

- wenn  $test(c_g \setminus c_1 = „fail“)$  dann arbeite mit  $c'_g = c_g \setminus c_1$  ( $c'_g$  ist der Wert für  $c_g$  im nächsten Schritt.)
- wenn  $test(c_g \setminus c_2 = „fail“)$  dann arbeite mit  $c'_g = c_g \setminus c_2$  ( $c'_g$  ist der Wert für  $c_g$  im nächsten Schritt.)
- wenn  $test(c_g \setminus c_1 = „pass“)$  **und** ( $c_g \setminus c_2 = „pass“$ ) dann erhöhe die Granularität. Granularität erhöhen bedeutet, die Anzahl der Teile erhöhen. (s. Tab. 8, Zeile 16) sowie Abb. 20

Gesamtmenge, geteilt in 2 Teilmengen      Teilung in 4 Teilmengen

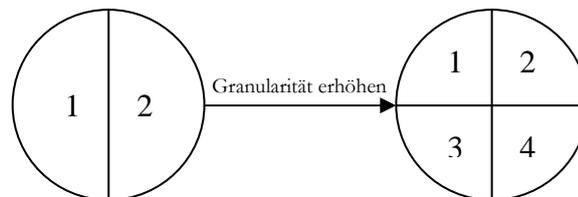


Abb. 20: Granularität erhöhen

Beispiel: Die Eingabe von 2 gefolgt von 4 verursacht den Fehler. Dieser Fehler soll gefunden werden.

Der unter [wpf] veröffentlichte Java-Code kann nach der Ausführung folgende Ausgabe bewirken:

Zn	Ausgabe	Erklärung
1	[1, 2, 3, 4]	Zu testende Eingaben
2	Running ddmin	ddmin-Algorithmus wird gestartet
3	ddmin([1, 2, 3, 4])...	Eingabewerte für ddmin
4	test([])...	leere Eingabe wird überprüft...
5	test([]) = PASS	<b>PASS</b>
6	test([1, 2, 3, 4])...	Test bei der Eingabe aller Zahlen
7	test([1, 2, 3, 4]) = FAIL	<b>FAIL</b>
8	split([1, 2, 3, 4], 2) = [[1, 2], [3, 4]]	Aufteilung auf zwei Teile
9	ddmin: testing subsets	
10	minus([1, 2, 3, 4], [1, 2]) = [3, 4]	Komplementärmenge bilden
11	test([3, 4])...	
12	test([3, 4]) = PASS	<b>PASS</b>
13	minus([1, 2, 3, 4], [3, 4]) = [1, 2]	Komplementärmenge bilden
14	test([1, 2])...	Ergebnisse werden getestet
15	test([1, 2]) = PASS	<b>PASS</b>
16	ddmin: increasing granularity	Granularität wird erhöht, da Zeile 12 und 15 auf PASS steht, d.h. kein Fehler gefunden wurde

17	<code>split([1, 2, 3, 4], 4) = [[1], [2], [3], [4]]</code>	Aufteilung in vier Teile
18	<code>ddmin: testing subsets</code>	
19	<code>minus([1, 2, 3, 4], [1]) = [2, 3, 4]</code>	Komplementärmenge bilden
20	<code>test([2, 3, 4])...</code>	Testen
21	<code>test([2, 3, 4]) = FAIL</code>	<b>FAIL</b>
22	<code>split([2, 3, 4], 3) = [[2], [3], [4]]</code>	Aufteilung in drei Teile
23	<code>ddmin: testing subsets</code>	
24	<code>minus([2, 3, 4], [2]) = [3, 4]</code>	Komplementärmenge bilden
25	<code>test([3, 4])...</code>	Testen
26	<code>test([3, 4]) = PASS</code>	<b>PASS</b>
27	<code>minus([2, 3, 4], [3]) = [2, 4]</code>	Komplementärmenge bilden
28	<code>test([2, 4])...</code>	Testen
29	<code>test([2, 4]) = FAIL</code>	<b>FAIL</b>
30	<code>split([2, 4], 2) = [[2], [4]]</code>	Aufteilung in zwei Teile
31	<code>ddmin: testing subsets</code>	
32	<code>minus([2, 4], [2]) = [4]</code>	Komplementärmenge bilden
33	<code>test([4])...</code>	Testen
34	<code>test([4]) = PASS</code>	<b>PASS</b>
35	<code>minus([2, 4], [4]) = [2]</code>	Komplementärmenge bilden
36	<code>test([2])...</code>	Testen
37	<code>test([2]) = PASS</code>	<b>PASS</b>
38	<code>ddmin([1, 2, 3, 4]) = [2, 4]</code>	<b>Lösung: 2 und 4</b>

Tab. 8: Ausgabe des ddmin-Tools

Vorgangsweise:

Die Zahlen 1 bis 4 werden eingegeben. Eine bestimmte Kombination dieser Zahlen verursacht einen Fehler. Durch schrittweise Einkreisung soll(en) diese Zahl(en) gefunden werden.

Die leere Eingabe verursacht offensichtlich keinen Fehler (*Zeile 4 und 5*) Bei der Eingabe von allen Zahlen entsteht ein Fehler. (*Zeile 6 und 7*) Es soll nun herausgefunden werden, welche Zahlen verantwortlich für diesen Fehler sind. Nach jedem entstandenen Fehler (*wie in Zeile 7*) wird die Gesamtmenge der Zahlen (also in *Zeile 8* die Zahlen 1 bis 4) geteilt. („split“) Danach wird die Komplementärmenge zwischen der vorherigen Gesamtmenge (Zahlen 1 bis 4) und eines übrig gebliebenen Teiles aus der Teilung (Zahlen 1 und 2 *Zeile 10*) gebildet. Entsteht ein Fehler muss wieder geteilt werden. Im oben stehenden Beispiel entsteht noch kein Fehler, (*Zeile 12*), deshalb wird der zweite übrig gebliebene Teil aus der Teilung (Zahlen 3 und 4 *Zeile 13*) für die Bildung der Komplementärmenge herangezogen. Die Eingaben sind wieder korrekt (*Zeile 14*) *Wäre diese Komplementärmenge nicht korrekt, müsste die Menge wieder gleichmäßig aufgeteilt werden. Übrig bleiben würden die Zahlen 1 und 2. Die Menge aus diesen beiden Zahlen würde geteilt. Übrig bleibt dann die Menge aus der Zahl 1 und die Menge aus der Zahl 2. Danach wird die Komplementärmenge zwischen 1 und 2 mit 1, und danach mit 2 gebildet. Einer dieser beiden Zahlen einzeln – oder die Eingabe von 1 gefolgt von 2 – verursacht den Fehler.* Da in diesem Schritt alle Komplementärmengen getestet, und kein Fehler gefunden wurde, muss die Granularität erhöht werden, d.h. es wird nicht wie zuvor in zwei Teile sondern in vier Teile geteilt. (*Zeile 16 und 17*) Der weitere Ablauf erfolgt wie vor der Granularitätserhöhung beschrieben. Die grau hinterlegte *Zeile 29* verursacht

den Fehler, also die Eingabe von 2 gefolgt von 4. Die Eingabe von ausschließlich 4 (*Zeile 34*), sowie von ausschließlich 2 (*Zeile 37*) führt zu *keinem* Fehler.

Dieser Algorithmus kann im Bereich Systemtest/Softwaretest verwendet werden. Das Testen von Modulen mit diesem Algorithmus hilft doppelte Einträge in der Datenbank zu vermeiden. Einträge, die nur relevante Information (*vgl. Abschnitt 8.3.3*) enthalten, können untereinander leichter verglichen werden. Fehler werden so eingekreist und leichter gefunden. Es ist denkbar, dass der/die Tester/in, der/die den Fehler bearbeitet, auch die Datenbank aktualisiert.

## 8.4 Planung des Desktop Analyse- und Erfassungstools

Zur Planung des Desktop Analyse- und Erfassungstools (*im folgenden DAE genannt*) wird UML (*Unified Modeling Language*) verwendet. UML ist eine standardisierte Sprache zur Modellierung von Software.

### 8.4.1 Funktionen des Desktop Analyse- und Erfassungstools

Das DAE wird auf einem Desktop PC installiert und hat mehrere Aufgaben, s. Anwendungsfälle *Abschnitt 8.4.2.1*

#### 8.4.2 Anwendungsfalldiagramm

„Anwendungsfalldiagramme sind beim Entwurf eines Systems Ausgangspunkt der Analysephase. ... Sie werden durch *Assoziationen* ergänzt und mit *Akteuren* verbunden, um Nichtfachleuten wie Managern und Endbenutzern die Gesamtstruktur und Verfügbarkeit in einem System zu veranschaulichen. Anwendungsfälle können dazu verwendet werden, um den Hauptverlauf der Ereignisse in einem System darzustellen, wenn keine Fehler auftreten.“ [Roff] Die Anwendungsfälle werden in Form von Ellipsen dargestellt.

*Assoziationen*: Assoziationen sind Beziehungen zwischen den Anwendungsfällen. Es gibt drei Arten: Die *include*, *extend* und die *Generalisierungsbeziehung*. Die *include*-Beziehung wird verwendet, wenn verschiedene Anwendungsfälle dieselbe Teilfunktionalität beinhalten. Die *extend*-Beziehung beschreibt die Erweiterung eines Anwendungsfalles. (*s. Abb. 21*) Die *Generalisierungsbeziehung* konkretisiert eine allgemeinere durch eine speziellere Aufgabe. *Bsp. aus dem Bestellwesen*: Die Aufgaben „Monatsrechnung erstellen“ und „Kreditkarte belasten“ konkretisie-

ren die Aufgabe „Forderung stellen“. Die Generalisierungsbeziehung wird durch einen Pfeil mit geschlossener, nicht ausgefüllter Spitze dargestellt. [web27] Sie wurde für das DAE nicht benötigt. *Akteure*: Akteure sind die Benutzer des Softwareproduktes.

#### 8.4.2.1 Anwendungsfälle

- Erfassung von Fehler- und Problemfällen,
- Anzeigen von Fehler- und Problemfällen,
- Status des Fehler- oder Problemfalles setzen,
- Anzeigen von Statistiken,
- Planung und Prognostizierung von zukünftigen Fehlern auf Basis dieser Statistiken. (z.B. Erfassung neuer Testfälle für zukünftige Entwicklungen auf Basis von Problemen, die vom Kunden gemeldet wurden.)

*Detaillierte Anwendungsfälle: Sea Level:*<sup>1</sup>

#### I. Erfassung von Fehler- und Problemfällen:

Der/Die „Verwaltungs“-Mitarbeiter/in...

1. ...gibt die Daten des Problems in die MySQL-Datenbank des PDA-Info-Systems ein. Bei diesem Interface handelt es sich um ein ähnliches wie jenes, das am PDA installiert wurde. Es ist Teil des DAE.
2. ...gibt das zugrunde liegende Softwareprodukt sowie weiters benötigte Daten ein. (z.B. Status, Kunde, Art des Fehlers (*s. Abschnitt 6.2*)).
3. ...gibt den Kundennamen ein.
4. Das System speichert die eingegebenen Daten in der Server-Datenbank des PDA-Info-Systems.
5. Das System beendet den Vorgang.

Der Kunde...

- 1a ...gibt die Daten des Problems in ein Web-Formular (*Abschnitt 6.3.1*) ein.
- 2a ...weiter bei I. Punkt 4.

<sup>1</sup> Nach **Cockburn** (Informatiker): Sea Level: beschreiben „eine klar abgegrenzte Interaktion zwischen Hauptakteur und System“ im Anwendungsfalldiagramm [vgl: Wolf, ab Folie II-36]

## II. Anzeigen/Ändern von Fehler- und Problemfällen:

Der/Die „Verwaltungs“-Mitarbeiter/in...

1. ...öffnet den zur Änderung oder zur Anzeige gewünschten Fehler-/Problemfall.
2. ...System überprüft, ob dieser Eintrag vorhanden ist.
3. ...ändert den Fehler-/Problemfall. (nicht bei Anzeige)
4. System speichert die Angaben. (nicht bei Anzeige)

## III. Status des Fehler- oder Problemfalles setzen: *(wie II.)*

## IV. Anzeigen von Statistiken:

Jede/r Benutzer/in des Systems mit Ausnahme des Kunden...

1. ...erstellt einen neuen Bericht
2. ...wählt die Anlage(n)/das(die) Softwareprodukt(e), indem er diese(s) in einem Dialog sucht oder gleich selbst per ID-Eingabe findet.
3. ...fügt diese(s) durch Klicken auf OK in den Bericht hinzu
4. ...kann die gewählten Anlagen (wenn es mehrere sind) durch Einstellung im Kontext-Menü auf Basis bestimmter Parameter (z.B. Fehleranzahl der letzten sechs Monate, Bearbeitungszeit usw.) vergleichen
5. Grafische Darstellung wird ausgegeben
6. Tabelle/Text wird ausgegeben
7. Bericht wird automatisiert erstellt und – wenn gewünscht – als .pdf gespeichert bzw. ausgedruckt.

Der/Die Benutzer/in des Systems mit Ausnahme des Kunden möchte die Daten für eine Anlage anzeigen, ohne Vergleiche durchzuführen

- 1a...kann Daten pro Anlage anzeigen lassen (rechte Maustaste, auf die hinzugefügte Anlage). (Fehlerliste, Administrator der Anlage, usw...)
- 2a...weiter zu IV Punkt 6

## Grafische Darstellung der Anwendungsfalldiagramme

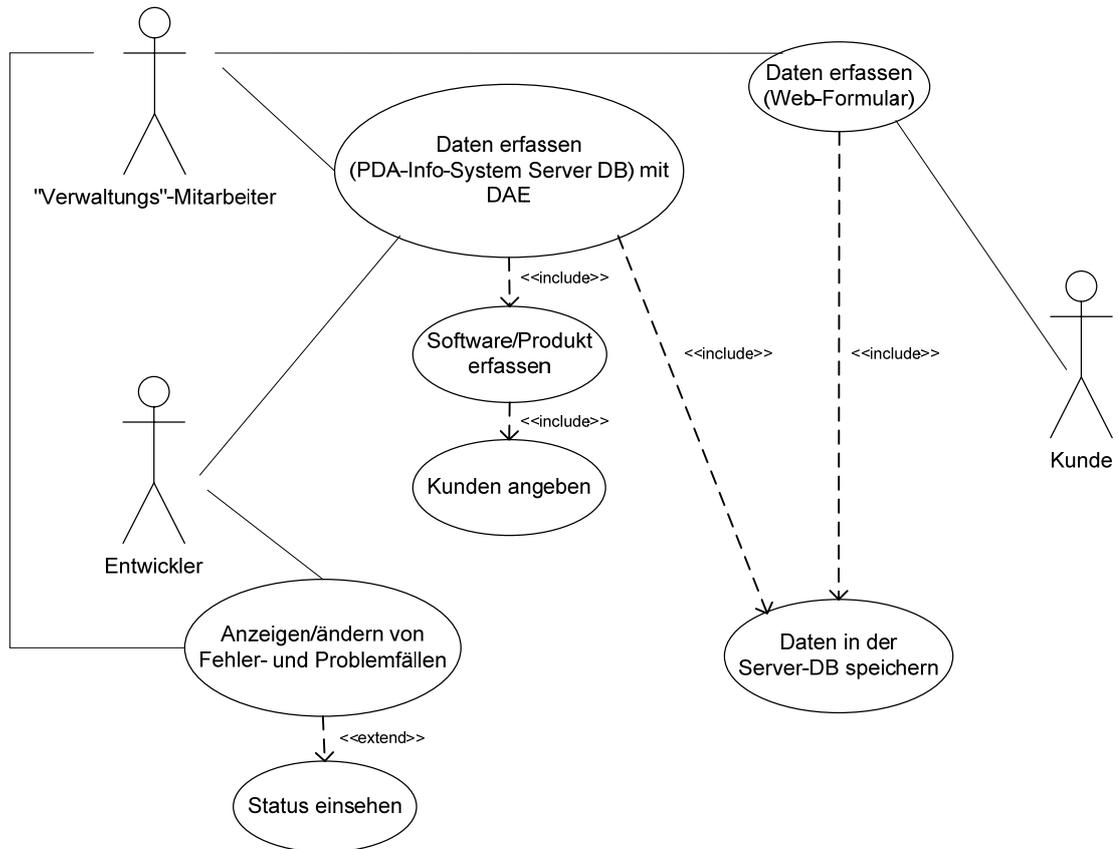


Abb. 21: Anwendungsfalldiagramm zu I, II. und III. Akteure sind Entwickler, Kunde und „Verwaltungs“-smitarbeiter

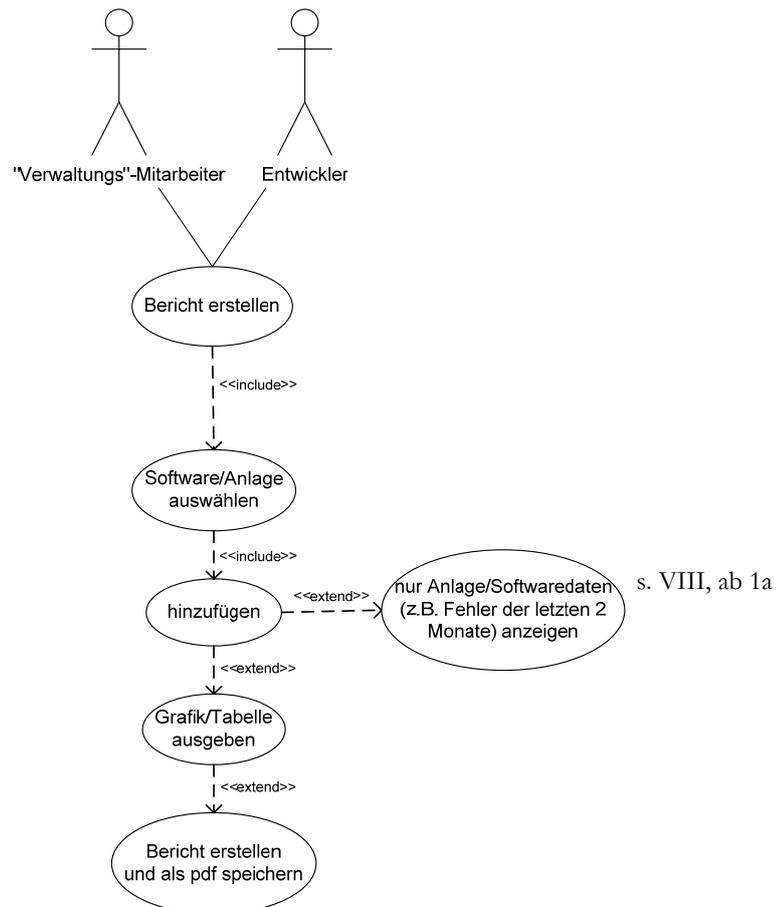


Abb. 22: Anwendungsfalldiagramm zu IV, Akteure sind „Verwaltungs“-Mitarbeiter und Entwickler

### 8.4.3 Klassendiagramm

„Klassendiagramme repräsentieren verschiedene Teile (Klassen) eines Systems, ihre Beziehungen untereinander und ihre Zugehörigkeit zu speziellen Subsystemen. Klassendiagramme umfassen Attribute und Operationen sowie zahlreiche Typen von Rollen und Assoziationen.“ [Roff] Ein neuer Begriff in Klassendiagrammen ist der Begriff *Multiplizität*. Es geht um die Angabe, wie viele Objekte einer Klasse am „anderen Ende“ einer Assoziationslinie für ein Objekt der betrachteten Klasse stehen können [Wolf], s. Abb. 23 und 24. In Abb. 24 ist ersichtlich, dass es noch die sogenannte *Aggregation* und die *Komposition* als spezielle Form der Assoziation bei Klassendiagrammen gibt. Aggregationen werden in Form von nicht ausgefüllten Rauten, Kompositionen in Form von ausgefüllten Rauten an einem Ende der Assoziationslinie dargestellt. Eine Aggregation kennzeichnet eine „Ganzes - Teil von“ - Beziehung. [Wolf]. Die Komposition beschreibt eigentlich das Gleiche wie die Aggregation nur mit dem Unterschied, dass Teile der Komposition keine Daseinsberechtigung haben, d.h. geht das Ganze unter, gehen auch die Teile unter. Bsp. aus dem Bestellwesen: Wird eine Bestellung des Kunden

gelöscht, hätte es keinen Sinn mehr, die einzelnen Artikelpositionen dieser Bestellung noch weiter aufzuheben. Die Positionen gehen mit der Löschung der Bestellung verloren. [Wolf]

### Grafische Darstellung der Klassendiagramme

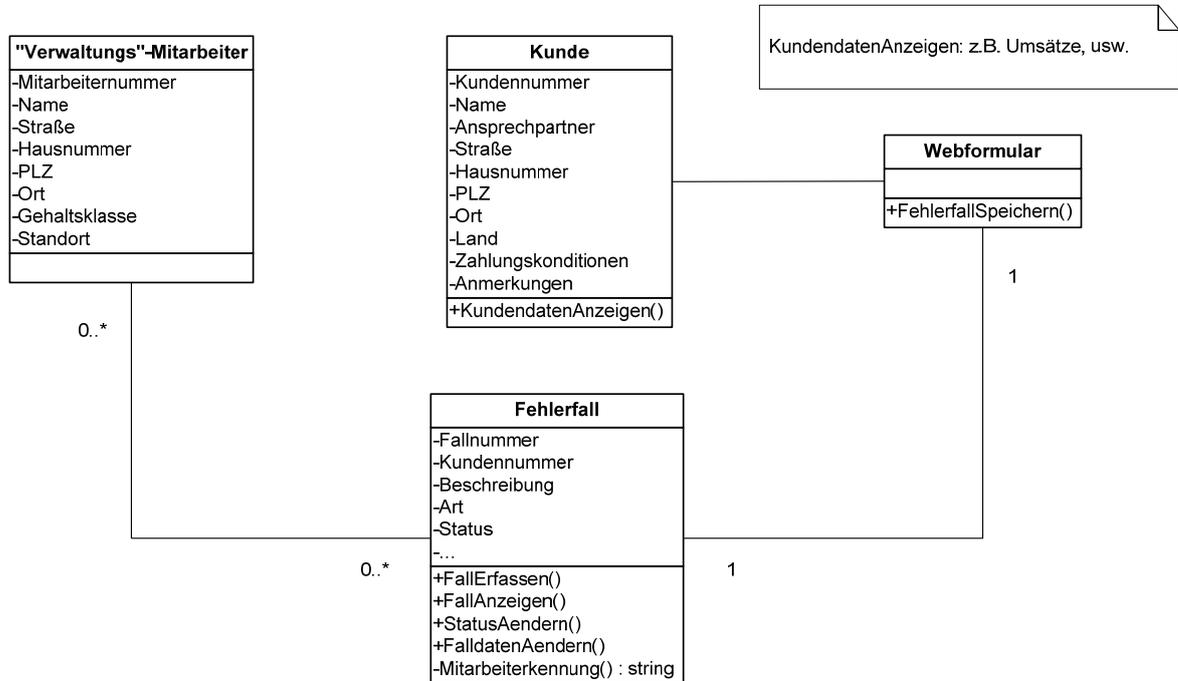


Abb. 23: Klassendiagramm zu I., II. und III.

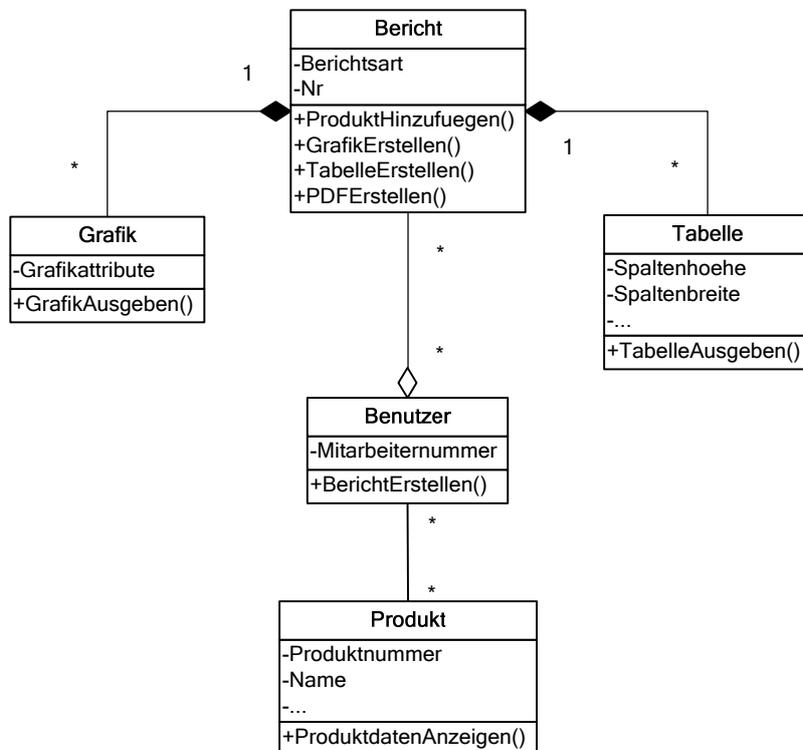


Abb. 24: Klassendiagramm zu IV.

### 8.4.4 Objektdiagramme

Objektdiagramme geben einen Zustand von Objekten zu einem bestimmten Zeitpunkt wieder. Von der Darstellung ähneln Objektdiagramme Klassendiagrammen.

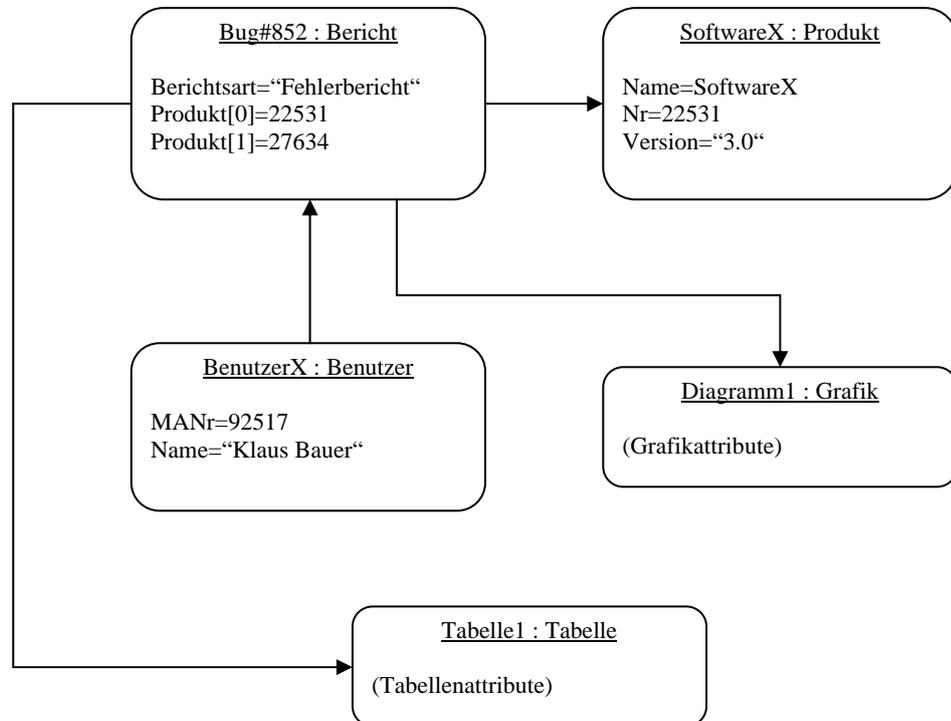


Abb. 25: Beispiel eines Objektdiagrammes zu IV.

*Erläuterung zu Abb. 25:* Der Benutzer/in erstellt ein Objekt mit dem Namen „Bug#852“ der Klasse Bericht. Danach fügt er/sie Produkte (zur Betrachtung) hinzu. Hiefür erstellt das Objekt „Bug#852“ ein Objekt der Klasse Produkt, nämlich „SoftwareX“. Nach dem Ausführen einer Methode mit z.B. dem Namen „BerichtErstellen“ (s. *Klassendiagramm IV.*) erstellt das Objekt „Bug#852“ Objekte der Klasse Diagramm und/oder Tabelle.

Des Weiteren definiert UML noch etliche weitere Diagrammtypen, wie z.B. Sequenzdiagramme und Kommunikationsdiagramme. Diesbezüglich wird auf die Literatur [Roff] verwiesen.

## 9 Backup- und Restore von Datenbanken

### 9.1 Einleitung

Ein wichtiges Kriterium bei der Planung und beim Einsatz eines Datenbankmanagementsystems<sup>1</sup>, im folgenden DBMS genannt, ist die Überlegung, welche Backup- und Restoremöglichkeiten es gibt, und welche dieser Möglichkeiten für den speziellen Einsatzzweck als die sinnvollste erscheint. Zu diesem Zweck müssen zuerst Grundlagen erläutert werden, um die Entscheidung am Ende begründen zu können.

Ziel dieses Kapitels ist es, Sicherungs- und Wiederherstellungstechniken für ein DBMS darzustellen. Am Ende wird ein Backup- und Restoresystem für die Serverdatenbank des PDA-Info-Systems geplant.

### 9.2 ACID-Prinzip

[störl]

Das ACID Prinzip (*ACID=Atomicity, Consistency, Isolation, Durability*) ist ein wichtiges Kriterium bei Datenbanken. Es besagt folgendes:

*Atomarität (Atomicity)*: Eine Transaktion (s. Abschnitt 7.1) wird entweder vollständig oder gar nicht ausgeführt. Kommt es zu einem Fehler während der Durchführung einer Transaktion, müssen bereits zuvor gemachte Änderungen wieder rückgängig gemacht werden.

*Konsistenz (Consistency)*: Eine Datenbank, die vor der Transaktion konsistent war, muss dies auch nach der Transaktion sein. Konsistenz bezeichnet die Widerspruchsfreiheit einer Datenbank.

*Isolation*: Transaktionen laufen als „ein einziges Kommando“ (=isoliert), voneinander ab. Bsp.: Transaktion B sieht Änderungen von Transaktion A erst dann, wenn Transaktion A vollständig abgearbeitet wurde.

*Dauerhaftigkeit (Durability)*: Transaktionen müssen nach erfolgreicher Durchführung dauerhaft in der Datenbank erhalten bleiben.

---

<sup>1</sup> Ein DBMS ist ein System, zur Verwaltung von Datenbeständen. Das Datenbankmanagementsystem ist Teil des Datenbanksystems. Weiters besteht das Datenbanksystem noch aus der Datenbank selbst. In der Datenbank werden die Datensätze gespeichert.

## 9.3 Fehlerbehandlung in Datenbanksystemen

[Störl]

### 9.3.1 Fehlerklassen

Grundsätzlich werden in Datenbanksystemen drei Fehlerklassen unterschieden. Es gibt *Transaktionsfehler*, *Systemfehler* und *Externspeicherfehler*. Nach dem Auftreten eines solchen Fehlers muss die Datenbank wieder in den sogenannten ACID-Zustand zurückgeführt werden. (s. *Abschnitt 9.2*)

#### 9.3.1.1 Transaktionsfehler

Transaktionsfehler treten auf, wenn Transaktionen abgebrochen werden. Zu Transaktionen s. *Kapitel 7 Konfliktmanagement*. Ein Transaktionsabbruch kann durch einen Abbruch des/der Benutzers/Benutzerin oder durch das DBMS selbst erfolgen. Transaktionsabbrüche des DBMS entstehen z.B. durch Deadlocks<sup>1</sup> oder durch Fehler im Anwendungsprogramm.

#### 9.3.1.2 Systemfehler

Systemfehler entstehen durch den Verlust der Daten im RAM-Speicher bei einem Stromausfall.

#### 9.3.1.3 Externspeicherfehler

Externspeicherfehler bezeichnen Fehler auf Festplattenspeichern, die zu einem teilweisen oder einem totalen Verlust von Datenbeständen führen.

## 9.4 Backup

[Störl]

Unter Backup einer Datenbank versteht man das Anlegen einer Kopie der gesamten DB oder von Teilen davon [web19]

---

<sup>1</sup> Deadlock: Ein Deadlock entsteht, wenn zwei Transaktionen einander gegenseitig blockieren.

### 9.4.1 Unterscheidung nach dem Gegenstand der Sicherung (Granulat)

In diesem Abschnitt geht es um die Frage, welcher Teil der Datenbank gesichert wird. Es können logische Strukturen (Tabellen, Tupel<sup>1</sup>) oder physische Strukturen (Table Spaces<sup>2</sup>, Dateien) gesichert werden.

Es gibt folgende Backup-Arten:

*Komplett-Backup*: Wie der Name schon sagt, handelt es sich hier um die Sicherung der gesamten Datenbank, d.h. mit sämtlichen Datensätzen, Dateien, Tabellen, Logfiles.

*Partielles Backup*: Bei einem partiellen Backup werden bestimmte Teile, entweder logische (Tabellen oder Tupel) oder physische Strukturen (Table Spaces oder Dateien) gesichert.

### 9.4.2 Änderungszustand der zu sichernden Daten

Die Sicherung der seit einem bestimmten Zeitpunkt veränderten Teile der Datenbank bezeichnet man als *inkrementelles Backup*. Solche Zeitpunkte sind entweder das letzte komplette oder inkrementelle Backup, oder auch nutzerdefinierte Zeitpunkte. Werden die veränderten Teile zu groß, wird ein Komplett-Backup durchgeführt. Ab diesem werden nur die veränderten Teile gesichert bis zum nächsten Komplett-Backup. Dies ist schneller als eine Komplettsicherung.

---

<sup>1</sup> Tupel: Ein Tupel ist ein Datensatz und somit eine Zeile einer Datenbanktabelle.

<sup>2</sup> Table Space: bezeichnet den Speicherort, in den Tabellen, Indizes und andere Datenobjekte geschrieben werden. [w9]

### 9.4.3 Betriebszustand der Datenbank während der Sicherung

Es gibt zwei Zustände:

*Offline-Backup:* Beim Offline-Backup sind während der Sicherung keine Veränderungen durch Transaktionen an der Datenbank möglich.

*Online-Backup:* Beim Online-Backup ist vollständiger oder teilweiser schreibender Zugriff auf die Datenbank während der Sicherung möglich.

### 9.4.4 Paralleles Backup

Unter parallelem Backup versteht man die Verwendung von mehreren Speichermedien gleichzeitig, um Performance-Gewinne zu erzielen. Die Prozessortaktraten steigen vergleichsweise zu den Festplattenzugriffsgeschwindigkeiten in einem höheren Ausmaß, insbesondere durch den Einsatz von Multiprozessorsystemen. Die Leistung solcher Systeme durch schnellere Prozessoren wird durch langsame Festplatten verringert. Durch die Verwendung von mehreren Festplatten kann diese Entwicklung entschärft werden.

## 9.5 Restore

[vgl. Störl]

Unter Restore einer Datenbank versteht man die Wiederherstellung derselben mithilfe einer Sicherungskopie (Backup). Dieses Backup wurde vor dem Restore erstellt.

Hier gibt es die Unterscheidung zwischen dem Gegenstand der Wiederherstellung („was“) und nach der Art und Weise wie die Wiederherstellung erfolgen soll. („wie“).

### 9.5.1 Unterscheidung nach dem Gegenstand des Restore (Granulat)

Es können sowohl logische (Tabellen, Tupel) als auch physische Strukturen (Table Spaces, Dateien) der Datenbank wiederhergestellt werden.

*Komplett-Restore:* Erfolgt eine vollständige Wiederherstellung der Datenbank in den ACID-Zustand (s. Abschnitt 9.2), wird dies als Komplett-Restore bezeichnet.

*Partielles Restore:* Sehr große Datenbanken mit vielen Tausenden oder Millionen Datensätzen werden nicht auf ein und derselben Platte gespeichert, bzw. ist dies auch überhaupt nicht möglich. In so einem Fall erfolgt ein Partielles Restore, d.h. es wird z.B. nur ein Teil *einer* oder *mehrerer* Platte(n) wiederhergestellt.

### 9.5.2 Art und Weise der Wiederherstellung

*Point-In-Time-Restore:* Point-In-Time-Restore bezeichnet die Wiederherstellung der Datenbank in den ACID-Zustand zu einem früheren Zeitpunkt.

### 9.5.3 Betriebszustand der Datenbank während des Restores

Online-Restore ermöglicht den *schreibenden Zugriff* auf die Datenbank während des Restore-Vorganges. Vorteil ist die raschere Verfügbarkeit der Datenbank bzw. Teilen davon.

### 9.5.4 Paralleles Restore

Analog zu dem in *Abschnitt 9.4.4* besprochenen parallelen Backup gibt es auch ein paralleles Restore. Es ist auch hier sinnvoll, sich die Daten für die Wiederherstellung von mehreren Medien gleichzeitig zu holen.

## 9.6 Alternativen für die Datensicherung und Wiederherstellung

In diesem Kapitel wird darauf eingegangen, welche Möglichkeiten es gibt, Daten zu schützen. Grundsätzlich ist für den Schutz von Daten immer redundante Information notwendig. Redundanz bedeutet in der Informatik die Existenz von identen Informationen. Wichtig ist zu bemerken, dass Redundanz möglichst effizient und sparsam eingesetzt werden muss, um Speicher und Kosten zu sparen.

### 9.6.1 Spiegelung

Bei der Spiegelung werden Informationen einer Magnetplatte auf eine oder mehrere andere gespiegelt (kopiert).

### 9.6.2 RAID

Das Akronym RAID steht für *Redundant Array of Independent Disks*, oder früher *Redundant Array of Inexpensive Disks*. Das Thema RAID ist sehr umfangreich. Für nähere Informationen wird auf [Störl, ab Seite 37] verwiesen.

Bei RAID gibt es sechs Levels. (Level 1-6), außerdem gibt es noch RAID 0. Bei RAID 0 handelt es sich um kein richtiges RAID, da dies keine Redundanz aufweist, jedoch die Zugriffsgeschwindigkeit durch Aufteilung der Datenblöcke auf mehrere Platten erhöht. Dies gewährt eine Zugriffsparallelität und steigert die Geschwindigkeit.

Aus den RAID Levels kann keine bestimmte Reihenfolge oder Rangordnung abgeleitet werden. Diese RAID Levels beschreiben jedoch immer eine bestimmte Technik. Zusammengefasst ist RAID ein Verbund aus mehreren physischen Platten, die als ein logisches Laufwerk organisiert sind. Redundanz wird hier gezielt erzeugt. [web21]

Im Folgenden werden jene Levels dargestellt, die speziell für die Sicherung im Datenbankbereich verwendet werden.

#### 9.6.2.1 RAID-Levels für Datenbanken

[web21]

#### RAID Level 1

Bei RAID-Level 1 handelt es sich um Spiegelung, siehe *Abschnitt 9.6.1*

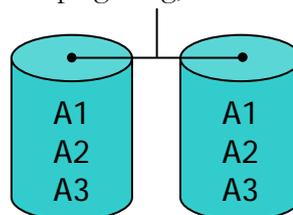


Abb. 26: Darstellung von RAID-Level 1 [web21]

Zu Abb. 26: Der Buchstabe *A* bezeichnet den Datenblock, die Nummer bezeichnet die Ordnungsnummer auf der Festplatte.

### RAID Level 0+1

RAID Level 0+1 kombiniert die Vorteile der Redundanz von RAID-Level 1 und die Erhöhung der Performance durch Level 0. RAID Level 0+1 ist ideal für Fileserver im Web, bei denen die Geschwindigkeit sehr wichtig ist, und wenige schreibende aber viele lesende Zugriffe (File-Download) erfolgen.

### RAID Level 5

RAID 5 benötigt mindestens drei Festplatten, meistens werden drei bis fünf eingesetzt. Bei RAID 5 werden die Nutzdaten blockweise verteilt. Dieser Level verwendet jedoch keine eigene Platte für die Paritätsinformation<sup>1</sup>. Paritätsinformationen werden gebildet, indem Blöcke mit XOR verknüpft werden.

Die XOR-Verknüpfung (auch Exklusiv-Oder,  $\oplus$ ) in der Datenverarbeitung wird am besten anhand einer Wahrheitstabelle (Tab.9) mit der Verknüpfung zweier Bits erläutert:

Bit 1	Bit 2	Verkn.	Ergebnis
0	0	$\oplus$	0
0	1	$\oplus$	1
1	0	$\oplus$	1
1	1	$\oplus$	0

Tab. 9: Wahrheitstabelle für eine XOR Verknüpfung zweier Bits

Dies bedeutet, dass eine Bedingung dann wahr ist (Konvention: Bit mit dem Wert 1 ist wahr), wenn *genau eine* Aussage wahr ist. Sind beide Aussagen wahr oder beide falsch so ist das Ergebnis der XOR-Verknüpfung falsch. (hier Wert 0).

<sup>1</sup> Paritätsinformationen werden dazu verwendet, um Fehler zu erkennen. Bei genügend Paritätsbits können Fehler auch automatisch korrigiert werden.

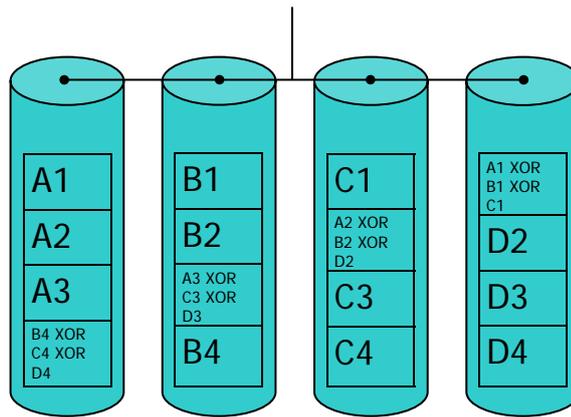


Abb. 27: Darstellung von RAID-Level 5 [web21]

### 9.6.2.2 RAID-Dreieck

[web22]

Das RAID-Dreieck hilft bei der Entscheidung, welcher RAID-Level für einen bestimmten Anwendungsfall am besten geeignet ist:

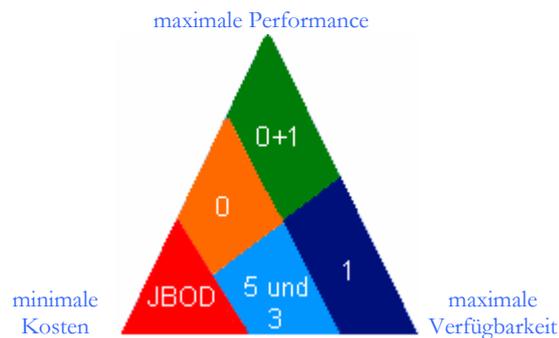


Abb. 28: RAID-Dreieck

JBOD: *Just a Bunch of Disks*: Festplatten werden nicht als RAID, sondern einzeln betrieben. [web21]

## 9.7 Backup vs. RAID

Datenbankbackups werden benötigt, um Datenbanken in einen bestimmten (früheren) Zustand wiederherzustellen. RAID-Systeme schützen vor Datenverlust durch Plattenausfälle. Datenverluste können durch Ereignisse höherer Gewalt (z.B. Blitzschlag, Überflutung, Head-Crash<sup>1</sup> usw.) entstehen.

<sup>1</sup> mechanische Beschädigung einer Festplatte

## 9.8 Backup/Restore und RAID für das PDA-Info-System

Für die Wahl der benötigten Komponenten und der angewandten Verfahren zur Datenbank-sicherung und Wiederherstellung werden im Folgenden die Anforderungen an die Datenbank festgelegt.

- Datensicherheit und Verfügbarkeit sind wichtiger als Zugriffsgeschwindigkeit,
- hohe Performance bei schreibendem Zugriff ist nicht so relevant, da eine Synchronisation der Außendienstmitarbeiter/innen mit der Server-Datenbank im Normalfall nur einmal täglich erfolgt.
- zu fixen Terminen erfolgen Komplettsicherungen, zwischen diesen Komplettsicherungen werden die SQL-Commands in Logfiles gesichert.
- der Backup-Server (enthält Logfiles und die Komplettsicherung der Datenbank) steht örtlich getrennt vom Datenbankserver.

Abb. 29 zeigt die Darstellung des Datenbanksystems.

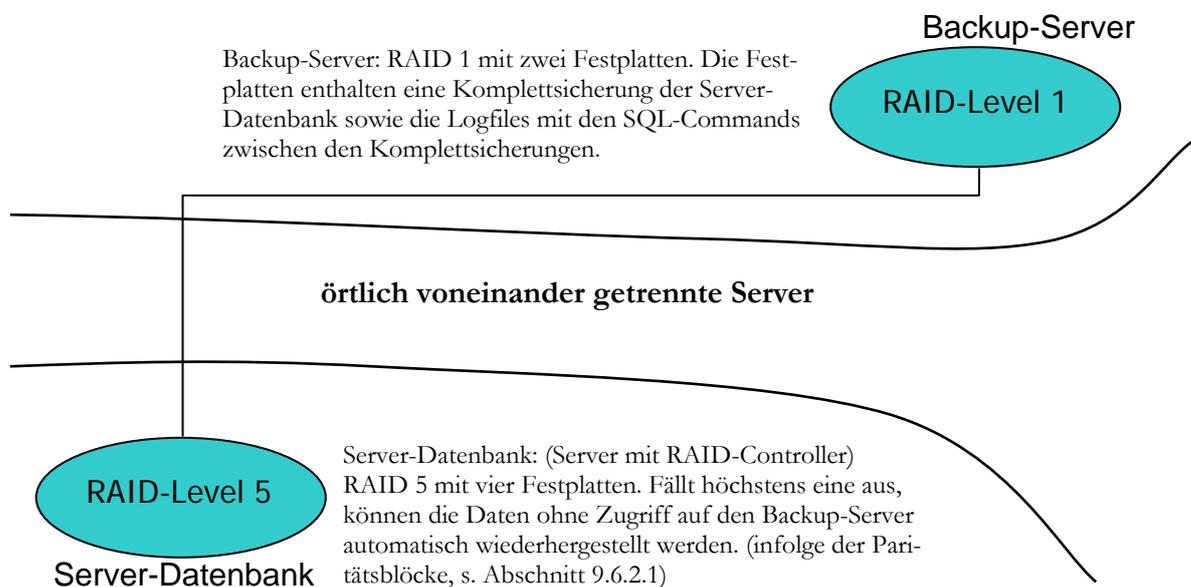


Abb. 29: Datenbanken des PDA-Info-Systems

Abb. 30 zeigt ein Beispiel für den zeitlichen Verlauf eines Backups:

*Beispiel: Jeden Montag erfolgt eine Komplettsicherung der Datenbank. Der Restore zwischen einer und der nächsten Komplettsicherung erfolgt anhand der Logfiles.*

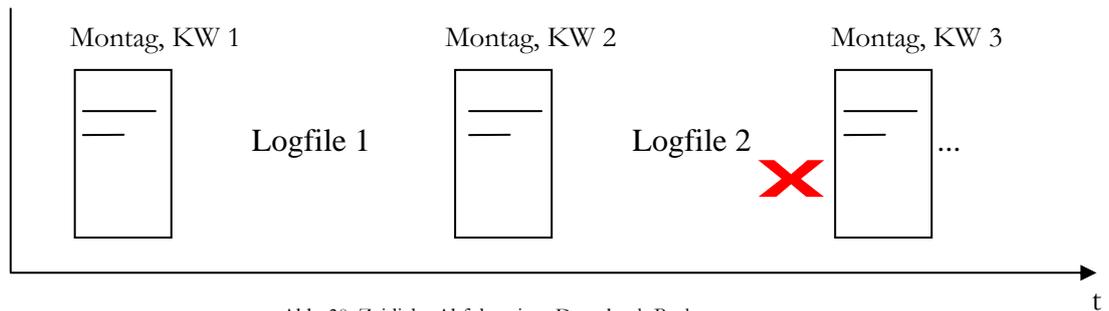
**Backup:**

Abb. 30: Zeitliche Abfolge eines Datenbank-Backups

Zum Zeitpunkt  $\times$  ist, infolge eines Blitzschlages, der Inhalt einer Festplatte des RAID 5 Systems verloren. Durch das RAID 5 System auf der Serverdatenbank können die Daten automatisch wiederhergestellt werden. Ein Zugriff auf den Backup-Server ist nicht erforderlich. Würde mehr als eine Platte oder die komplette Server-Datenbank zerstört, könnte ein Restore mithilfe des Backup-Servers durchgeführt werden. Bei inkonsistenten Datenbeständen muss für ein Restore der Datenbank auch auf den Backup-Server zugegriffen werden.

Für das Komplettbackup kann das Tool `mysqldump` verwendet werden (s. MySQL-Dokumentation). Logfiles können mithilfe von MySQL erstellt werden, oder sie werden programmtechnisch generiert.

**Restore:**

Die Wiederherstellung läuft in der folgenden Reihenfolge ab:

- 1) Einspielen des Komplettbackups,
- 2) Einspielen der Logfiles.

Beispiel für ein MySQL Logfile

```
#PDAInfoSystem_2007-06-12.sql
#2007-06-12 13:12
#
INSERT INTO ...
DELETE FROM ...
SELECT FROM ...
#[ usw. ]
```

*nächstes File*

```
#PDAInfoSystem_2007-06-13.sql  
#2007-06-13 15:49  
  
INSERT INTO....  
#[ usw. ]
```

Es könnte z.B. *jeden Tag* ein neues Logfile erstellt werden, um die Dateigrößen der Logfiles klein zu halten. Während eines Tages wird der neue SQL-Text einfach an die Datei angehängt. (*APPEND*). So ist jederzeit ein Restore der Datenbank mithilfe der Komplettsicherungen und den Logfiles möglich.

Unter Berücksichtigung der Festplattengröße des Backup-Servers werden nach einer bestimmten Zeit alte Komplet-Backups und Logfiles gelöscht.

## 10 Kosten/Nutzen-Analyse

### 10.1 Kosten

Kosten, die bei der Einführung des PDA-Info-Systems entstehen:

- Personalkosten für Test und Implementation des Systems,
- Personalkosten für die Wartung, Weiterentwicklung und Instandhaltung,
- Kosten für die Beschaffung der benötigten Hardware-Komponenten,
- Anfängliche (und folgende) Schulungsmaßnahmen für Mitarbeiter/innen,
- Lizenzkosten für MySQL-Server und IDE-Lizenzen für Visual Studio

### 10.2 Nutzen

Zusatznutzen, die sich aus dem PDA-Info-System ergeben:

- Erhöhung der Kundenzufriedenheit und Steigerung der Bindung des Kunden an ein Produkt durch effizienteres Problemmanagement.
- Die erhöhte Bindung der bestehenden Kunden verringert die Kosten für die Akquisition neuer Kunden um bestimmte Erfolgs-/Umsatzziele zu erreichen. Die Aufrechterhaltung bestehender Kundenbeziehungen kostet bedeutend weniger, als die Gewinnung neuer Kunden
- Kostensenkung durch zentralisierte Erfassung der Daten
- Zentrale Abfragemöglichkeit der Informationen
- „Befreiung“ der Datenbank vor irrelevanten Informationen durch automatisierten Softwaretest nach der Feststellung und Reproduktion des Fehlers(*s. Abschnitt 8.3.4.1*)
- Infolge des teilweisen Ersatzes von Laptop und Mobiltelefon durch Handhelds entstehen weniger Kosten. Handhelds werden auch für die Telefonie sowie für die Terminverwaltung verwendet.

## 10.3 Erläuterungen zu den Kostenpositionen

### 10.3.1 Hardwarekosten

Kostenpositionen für Hardware:

- Anschaffung von Handhelds für die Mitarbeiter/innen
- Infrarot-Projektions-Tastatur für die rasche Dateneingabe vor Ort
- Datenbankserver

### 10.3.2 Lizenzkosten

Microsoft Visual Studio 2005 Professional

MySQL Enterprise Server

#### 10.3.2.1 Microsoft Lizenzen [web23]

Für diese Preiskalkulation wurde eine Einzellizenz für Microsoft Visual Studio 2005 angenommen, da es auf viele verschiedene Faktoren ankommt, welche Lizenz die richtige ist, und wieviele Entwickler/innen sich mit dem Produkt „PDA-Info-System“ schließlich beschäftigen werden.

Als Referenz werden hier die unterschiedlichen Lizenzierungsmöglichkeiten in einem groben Überblick dargestellt:

#### *Lizenzen zum Kaufen*

Die von Microsoft angebotene *OPEN License* richtet sich an Klein- und Mittelbetriebe die 5 bis 249 PCs haben, den Lizenzbetrag sofort entrichten möchten, sowie das dauerhafte Nutzungsrecht erwerben möchten. Demgegenüber steht die *Select License*: Sie richtet sich an Betriebe mit mehr als 249 PCs. Bei dieser Lizenz erfolgt der Erwerb durch Einmalzahlung oder durch laufende Zahlung während der Lizenzlaufzeit. Die Nutzung des Produktes ist zeitlich unbefristet.

### *Weitere Möglichkeiten*

Weiters gibt es noch die Möglichkeit Lizenzen auf Raten zu kaufen, Lizenzen zu leasen und Lizenzen zu mieten.

### *Software Assurance*

Zusätzlich zu all diesen Lizenzierungen gibt es noch die Möglichkeit der *Software Assurance*. Software Assurance unterstützt ein Unternehmen während des Lebenszyklus eines Softwareproduktes. Leistungen sind z.B. Rechte auf neue Produktversionen, Microsoft eLearning, Home Use Program, erweiterter technischer Support, um nur einige zu nennen. Für weitere Informationen wird auf die Website von Microsoft [web23] verwiesen.

### **10.3.2.2 MySQL Lizenzen**

MySQL Community Server unterliegt der sogenannten GNU GPL (GNU General Public License). Bestimmungen der GNU sind, dass diese Produkte frei ohne jede Einschränkung verwendet werden dürfen. Jede Person darf Kopien verteilen, ist jedoch nicht dazu verpflichtet. Programme dürfen verändert und angepasst werden. Wird ein Produkt unter der GPL Lizenz für eigene Entwicklungen verwendet, so unterliegt auch das eigene Produkt der GPL-Lizenz. Für MySQL bedeutet das z.B.: Verwendet ein Entwickler den freien MySQL Community-Server für seine eigenen Entwicklungen, so müsste der Entwickler auch den Quelltext seines eigenen Produktes offenlegen. Da dies aber in den meisten Fällen nicht gewünscht ist, gibt es den sogenannten MySQL Enterprise Server. Dieses Produkt inkludiert des Weiteren auch Beratungs- und Problemlösungsunterstützung. [web24]

### **10.4 Kostenkalkulation für eine/n Mitarbeiter/in inkl. Erstinvestitionen**

In diesem Absatz werden die ungefähren Kosten an einem Beispiel dargestellt. Unternehmens- und Mengenrabatte werden nicht berücksichtigt. Bei der Software wird vom Kauf eines einzelnen Produktes ausgegangen. Die Preise verstehen sich exkl. USt.

<b>Kostenposition</b>	<b>Nettopreis</b>
<b>a) Erstinvestition in Hardware</b>	
a1) Server (Fujitsu-Siemens PRIMERGY RX300S3-903DE) Datenbankserver	2.770,00 EUR
a2) 4x Festplattenerweiterung (SATA 1TB) für DB-Server	1.200,00 EUR
a3) Server (Fujitsu-Siemens PRIMERGY RX300S3-903DE) Backup-Server	2.770,00 EUR
a4) 2x Festplattenerweiterung (SATA 1 TB) für Backup-Server	600,00 EUR
<b>b) Software</b>	
b1) Microsoft Visual Studio 2005	500,00 EUR
b2) MySQL Enterprise Server Basic	495,00 EUR
b3) CoreLab MySQLDirect .NET Data Provider Mobile ca. Team License (für max. 4 Entwickler): 349,95 USD USD/EUR Kurs vom 11. Juni 2007 1 USD = 0,75 EUR	262,00 EUR
<b>c) Hardwarekosten pro Mitarbeiter</b>	
c1) PDA (Fujitsu-Siemens Pocket LOOX T810)	542,00 EUR
c2) Infrarot-Projektions-Tastatur (Celluon CL800BT)	140,00 EUR
<b>d) Personalkosten</b>	
d1) Schulungsaufwand für eine/n Außendienstmitarbeiter/in 10h pro neue/n Mitarbeiter/in, 20 EUR/Stunde und Mitarbeiter/in ( <i>interner Stundensatz</i> )	200,00 EUR
d2) Systementwicklung: Test- und Entwicklungspersonal rd. 350h für einen Entwickler, à 40 EUR ( <i>interner Stundensatz</i> )	14.000,00 EUR
<b>Gesamtkosten bei Erstininstallation für eine/n Mitarbeiter/in</b>	<b><u>23.479,00 EUR</u></b>

Bei den Preisangaben in Kursivdruck handelt es sich um Schätzwerte. Weiters entstehen noch laufende Kosten für die Weiterentwicklung der Software sowie für die Einschulung neuer Mitarbeiter/innen. Für zusätzliche Informationen wird auf die Fujitsu-Siemens [fsc] sowie auf die MySQL Website [mysql2] verwiesen.

## 11 Zusammenfassung und Erweiterungen

Die Wartung, Instandsetzung und Aufrechterhaltung von großen Softwareprojekten, die bereits beim Kunden implementiert sind, gestaltet sich in der Regel schwierig. Diese Arbeit stellt eine Basis dar, um ein unternehmensweites Informationssystem zur Unterstützung bei diesen Aufgaben zu entwickeln.

Im ersten Teil wird auf unterschiedliche mobile Hardware sowie verfügbare Betriebssysteme für diese Hardware eingegangen. Dies soll als Entscheidungsgrundlage für die unternehmensweite Hardwarebeschaffung bei der Einrichtung eines firmenweiten (mobilen) Informationssystems dienen. Des Weiteren werden auch verfügbare mobile Datenbanken für Handhelds beschrieben.

Der zweite Teil beschreibt Programmiertechniken, insbesondere die Microsoft .NET-Umgebung.

Das dritte Kapitel beschäftigt sich mit MySQL. Es beschreibt, worauf bei der Verwendung von MySQL zu achten ist.

Die weiteren Kapitel beschäftigen sich schließlich mit der Datenspeicherung, der effizienten Suche von Daten, der Planung eines Analysetools für Auswertungen, der Anpassung der Informationen an Endgeräte, Informationsäquivalenz sowie Transaktionen und Konfliktmanagement. Ein Thema ist auch die automatisierte Vereinfachung von Problembeschreibungen, sowie automatisierter Softwaretest.

### **Mögliche Ausbaustufen:**

- Erweiterung und Implementation des diskutierten Desktop Analyse- und Erfassungstools. (s. *Abschnitt 8.4*) *Abb. 31* zeigt ein GUI Beispiel des DAE-Tools.

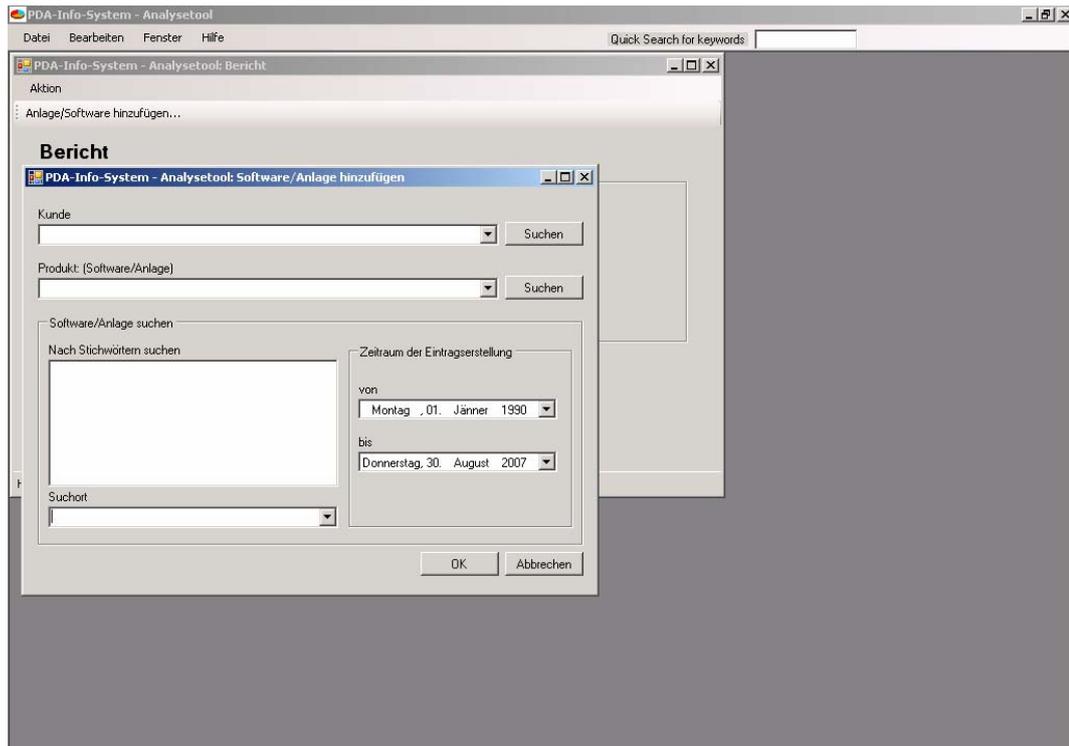


Abb. 31: GUI-Beispiel des DAE-Tools

- Implementation eines Suchalgorithmus in die PDA-Software wie in *Abschnitt 6.5* beschrieben, *Abb. 32* zeigt die derzeitige Implementierung der PDA-Software. (Startfenster)



Abb. 32: PDA-Software des PDA-Info-Systems, Startfenster

- Entwicklung eines Webportals mittels ASP. Das Webportal beinhaltet das Kundenfehlermeldungsformular. Weiters kann die PDA-Software als Webanwendung implementiert werden. Wenn Siemens Mitarbeiter/innen keinen PDA mitführen, können sie beim Kunden die Webseite aufrufen und auf die Serverdatenbank zugreifen. Der Login auf das Webportal (s. Abb. 33) erfolgt rollenspezifisch. Es gibt die Rolle „Customer“ und die Rolle „Admin“. Als weitere Hierarchieebene gibt es die User. Beim Customer sind die User die einzelnen Kunden, beim Admin sind die User der/die Mitarbeiter/innen.

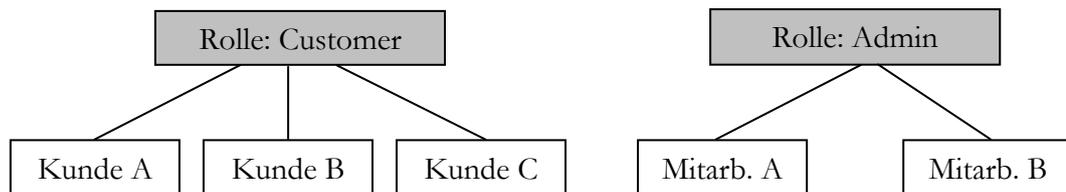


Abb. 33: Rollenverteilung für das Webportal Log-in

- Implementation der Location Based Service Funktion (s. Abschnitt 6.1)

## Literaturverzeichnis

### URL-Liste

- [web1] PalmOS: Wikipedia  
<http://de.wikipedia.org/wiki/Palmos>  
23. Februar 2007
- [web2] Eingebettetes System  
<http://de.wikipedia.org/wiki/Palmos>  
23. Februar 2007
- [web3] Microsoft .NET  
[http://en.wikipedia.org/wiki/Microsoft\\_.NET](http://en.wikipedia.org/wiki/Microsoft_.NET) *einchl. Folgelinks auf dieser Website*  
7. Februar 2007
- [web4] .NET Sprachen  
[http://en.wikipedia.org/wiki/Microsoft\\_.NET\\_Languages](http://en.wikipedia.org/wiki/Microsoft_.NET_Languages)  
7. Februar 2007
- [web5] .NET Sprachen, Unterschiede  
<http://www.microsoft.com/germany/msdn/csharp/2003/faq.msp#E3G>  
8. Februar 2007
- [web6] J#  
[http://de.wikipedia.org/wiki/J\\_Sharp](http://de.wikipedia.org/wiki/J_Sharp)  
8. Februar 2007
- [web7] Mono-Projekt  
<http://de.wikipedia.org/wiki/Mono-Projekt>  
4. Juni 2007

- [web8] Portable.NET  
<http://de.wikipedia.org/wiki/Portable.NET>  
4. Juni 2007
- [web9] Microsoft Project Rotor  
<http://www.microsoft.com/germany/msdn/library/net/ProjektRotorMicrosoftsNETPlattformFuerUnixDerivate.aspx?mfr=true>  
4. Juni 2007
- [web10] RedFive Labs  
*.NET für Symbian OS*  
<http://www.redfivelabs.com/>  
4. Juni 2007
- [web11] S60 (Nokia Series 60)  
<http://de.wikipedia.org/wiki/S60>  
4. Juni 2007
- [web12] UIQ (User Interface Quartz von Sony Ericsson)  
<http://de.wikipedia.org/wiki/UIQ>  
4. Juni 2007
- [web13] ECMA  
[http://de.wikipedia.org/wiki/Ecma\\_International](http://de.wikipedia.org/wiki/Ecma_International)  
4. Juni 2007
- [web14] MySQL Customers by Industry  
<http://www.mysql.com/customers/>  
30.3.2007
- [web15] Informationsanpassung für mobile Geräte  
[www.ipd.uni-karlsruhe.de/~oosem/mobil/resources/koch/Ausarbeitung.doc](http://www.ipd.uni-karlsruhe.de/~oosem/mobil/resources/koch/Ausarbeitung.doc)  
19. März 2007

- [web16] Akiyo Nadamoto, Hiroyuki Kondo, Katsumi Tanaka  
*Graduate School of Science and Technology, Kobe University*  
*1-1 Rokkodai, Nada, Kobe, Hyogo 657-8501, Japan*  
WebCarousel: Restructuring Web Search Results for Passive Viewing in Mobile Environments  
<http://csdl2.computer.org/comp/proceedings/dasfaa/2001/0996/00/09960164.pdf>  
19. März 2007
- [web17] WebCarousel, ab Folie 6  
<http://www.ipd.uka.de/~koenig/MODIS/infoanpassung.pdf>  
19. März 2007
- [web18] WebCarousel, Berechnungen  
<http://www.ipd.uka.de/~koenig/MODIS/infoanpassung.pdf>  
19.4.2007
- [web19] Datensicherung (Backup und Restore) in relationalen und objektorientierten Datenbanksystemen sowie in TP-Monitoren: Forschungspläne, -ziele und -status  
Uta Störl  
<http://www.minet.uni-jena.de/dbis/alumni/stoerl/veroeffentlichungen/grundlagen-ws-95.ps.gz>  
19.4.2007
- [web20] Table Space Definition  
<http://de.wikipedia.org/wiki/Tablespace>  
19.4.2007
- [web21] RAID (Redundant Array of Independent Disks)  
<http://de.wikipedia.org/wiki/RAID>  
23.4.2007

- [web22] RAID-Dreieck  
<http://www.tecchannel.de/storage/grundlagen/401665/index18.html>  
11. Juni 2007
- [web23] Microsoft Visual Studio 2005 Lizenzsystem  
*Open License Merkmale*  
6. Juni 2007
- [web24] MySQL Enterprise  
*Leistungsmerkmale*  
6. Juni 2007
- [web25] Typsicherheit  
[http://msdn2.microsoft.com/de-de/library/hbzz1a9a\(VS.80\).aspx](http://msdn2.microsoft.com/de-de/library/hbzz1a9a(VS.80).aspx)  
10 März 2007
- [web26] CLI  
[http://de.wikipedia.org/wiki/Common\\_Language\\_Infrastructure](http://de.wikipedia.org/wiki/Common_Language_Infrastructure)  
30. Juli 2007
- [web27] UML, Anwendungsfalldiagramm  
[http://www.dpunkt.de/leseproben/2082/Kapitel\\_12.pdf](http://www.dpunkt.de/leseproben/2082/Kapitel_12.pdf)

## Monographien und Lehrbücher

- [Höpf] Hagen Höpfner, Can Türker, Birgitta König-Ries  
Mobile Datenbanken und Informationssysteme  
*Konzepte und Techniken*  
Dpunkt Verlag, 2005
- [Veij] J. Veijalainen und T. Gross  
Mobile Wireless Interfaces: In Search for the Limits.  
In: B. König-Ries, K. Makki, S. A. M. Makki, N. Pissinou und P. Scheuermann  
(Hrsg.), Revised Papers from the NSF Workshop on Developing an Infrastructure for  
Mobile and Wireless Systems, Lecture Notes in Computer Science, Band 2538, S.153-  
163  
Springer-Verlag, London, UK, 2001
- [Kofl] Michael Kofler  
MySQL 5  
*Einführung, Programmierung, Referenz*  
Addison-Wesley, München, 2005, 3. Auflage
- [Wolf] Matthias Wolf  
Software Engineering & UML  
Skriptum für die Fachhochschule St.Pölten, Studiengang Telekommunikation und  
Medien, Informatik  
*Grundlagen der objektorientierten Softwareentwicklung*  
2005
- [Störl] Uta Störl  
Backup und Recovery in Datenbanksystemen  
Teubner Verlag, 2001

- [Zeller]    Andreas Zeller  
              Why Programs fail  
              *A Guide to Systematic Debugging*  
              Dpunkt Verlag, 2006
- [Roff]        Jason T. Roff  
              UML  
              Mitp-Verlag, 2004

## Weiterführende Links

- [cr] CoreLab Software Development  
[www.crlab.com](http://www.crlab.com)
- [ms1] MSDN Library  
<http://msdn2.microsoft.com/en-us/library/ms310245.aspx>,
- [ms2] Bericht: .NET-Technologie steigert Effizienz der Servicetechniker von Telekom Austria  
<http://www.microsoft.com/austria/enterprise/telekom.msp>
- [mysql1] MySQL ADO.NET Treiber  
<http://www.mysql.com/products/connector/>
- [mysql2] MySQL Website  
<http://www.mysql.com>
- [wpf] <http://www.whyprogramsfail.com/resources.php>  
Java-Beispielprogramm für die Implementierung des *ddmin*-Algorithmus  
Datei: *wpf-code-examples.zip*, Datei: *DD.java*
- [fsc] <http://www.fujitsu-siemens.at>  
Fujitsu-Siemens Website

## Glossar und Abkürzungsverzeichnis

- .NET Compact Framework**, Technologie zur Softwareentwicklung für mobile Geräte
- ACID**, Atomicity, Consistency, Isolation, Durability
- ADO.NET**, ActiveX Data Objects
- API**, Application Programming Interface, Schnittstelle einer Anwendung, die anderen Anwendungen zur Verfügung steht. (z.B., Betriebssystemschnittstellen)
- Append**, Log-File Text an eine bestehende Datei anhängen
- ARM**, Advanced RISK Machine, Prozessorarchitektur
- ASP**, Active Server Pages, Serverseitige Scriptsprache von Microsoft. Vergleichbar z.B. mit PHP
- Binaries**, bei Binaries handelt es sich um von der Hardware ausführbaren, zuvor kompilierten Code. (→ kompilieren)
- Bluetooth**, Übertragungsstandard zur drahtlosen Datenübertragung über kurze Distanzen
- BSD**, Berkley Software Distribution
- C#**, Programmiersprache, Hochsprache
- C++**, Programmiersprache, häufig verwendet für maschinennahe Programmierung. Beherrscht aber auch Objektorientierung.
- Caching**, Daten für einen schnellen Zugriff bereitstellen.
- CellID**, Cell Identifier
- CIL**, Common Intermediate Language
- CLI**, Common Language Infrastructure
- CLR**, Common Language Runtime
- committed**, Abschluss einer Transaktion. Änderungen sind ab diesem Zeitpunkt dauerhaft
- Compiler**, wandelt Programmcode in ausführbaren Maschinencode bzw. Zwischencode um
- Connector**, ermöglicht eine Verbindung zu einer Datenbank
- CoreLab MySQL .NET Data Provider**, → Connector auf Basis des → .NET Compact Frameworks für mobile Geräte
- CTS**, Common Type System
- DAE**, Desktop Analyse- und Erfassungstool
- Datenbank**, → DB
- Datenbankmanagementsystem**, → DBMS
- Datenbankserver**, Hardware, auf der das → DBMS läuft
- Datensammlungssystem**, System zur Speicherung von Daten in einer Datenbank
- DB**, der Begriff Datenbank bezieht sich auf die gespeicherten Daten in einer Datenbank
- DBMS**, Datenbankmanagementsystem, Verwaltungsprogramm für Datenbanken → DB
- DBS**, Datenbanksystem, besteht aus dem → DBMS und der → DB
- Deadlock**, Ein Deadlock entsteht, wenn zwei → Transaktionen einander gegenseitig blockieren.
- Disassembler**, ermöglicht die Wiederherstellung von Maschinencode in lesbaren Code.
- Dotfuscator**, Microsoft Programm, das es ermöglicht, Quelltext von Software zu verschleiern, um Urheberrechte zu schützen.
- ECMA**, European Computer Manufacturers Association, seit 1994 Ecma Europe
- EDGE**, Enhanced Data Rates for GSM Evolution, ist ein Verfahren zur Erhöhung der Datenrate in GSM-Netzen und wird mit relativ wenig Aufwand beim Mobilfunkbetreiber realisiert.
- Eingebettete Systeme**, hardwarenah konstruierte Systeme
- ELinOS**, freie Entwicklungsumgebung von der Firma Sysgo zur Entwicklung von →Linux Embedded Anwendungen

**exe-Datei**, ausführbare (executable) Datei in Windows-Betriebssystemen

**Firebird**, Ist ein → DBS

**fixe Speicherung**, hier langfristige, dauerhafte Archivierung von Daten in einer Datenbank  
→ Datenerfassung

**GNU** GNU's not → Unix

**GNU GPL**, GNU GENERAL PUBLIC LICENSE

**GPRS**, General Packet Radio Service, GSM-Erweiterung zur paketorientierten Datenübertragung

**GPS**, Global Positioning System, satellitengestütztes System zur Positionsbestimmung, verwendet die NAVSTAR-Satelliten des US-Verteidigungsministeriums.

**GSM**, Global System for Mobile Communication, Standard für digitale Mobilfunknetze

**GUI**, Graphical User Interface, Erlaubt die Kommunikation zwischen einem Benutzer und einer Maschine mittels grafischen Komponenten (Mauszeiger, Icons usw.)

**Handheld**, Begriff umfasst mobile Geräte wie z.B. →PDA und →MDA

**HTML**, Hypertext Markup Language, textbasierende Auszeichnungssprache für Internetinhalte

**IBM DB2**, ist ein → DBS für → PDAs

**IDE**, Integrated Development Environment (*dt.* integrierte Entwicklungsumgebung) Die IDE ist ein Programm zur Entwicklung von Software. Sie besteht meist aus einem Texteditor, einem Compiler, einem Debugger, einem Linker und diversen Quelltext-Formatierungsfunktionen und im Falle von → Microsoft Visual Studio noch vielen weiteren Tools.

**IIS**, Internet Information Service

**Informationssystem**, System, das zur Speicherung und Informationsbeschaffung sowohl auf/mit mobilen Geräten, als auch auf/mit Desktop PCs ermöglicht.

**Informix**, Ist ein → DBS

**Java**, Programmiersprache

**JBOD**, Just a Bunch of Disks

**JDBC**, Java Database Connectivity, Datenbankschnittstelle (von Sun), bietet eine einheitliche Schnittstelle für verschiedene Datenbanken. JDBC benötigt einen Treiber vom Datenbankhersteller

**kompilieren**, Programmcode übersetzen

**LAN**, Local Area Network

**Laufzeitumgebung**, Zwischenschicht, die zwischen der Anwendungs- und der Betriebssystemschicht liegt. Programme können so plattformunabhängig (bezogen auf Hardware und Betriebssystem) laufen. Beispiele, → .NET Framework, Java-Applets.

**Linux Embedded**, → Handheld-Betriebssystem auf Basis des Linux-Kernels

**Logfile**, In Logfiles (=Protokoll) werden Ereignisse eines Softwareproduktes aufgezeichnet

**Log-Text**, Text von → Logfiles

**MAC OS X**, Macintosh Operating System X, Betriebssystem für Macintosh-Rechner

**MDA**, Mobile Digital Assistant, PDA, der Telefonfunktionen inkludiert.

**Microsoft Pocket PC** → Microsoft Windows Mobile

**Microsoft SQL Server CE**, ist ein → DBS für → PDAs

**Microsoft SQL Server CE**, mobile Datenbank, wird am mobilen Gerät installiert. Die Datensätze werden in Dateien mit der Dateinamenerweiterung sdf gespeichert.

**Microsoft Visual Studio 2005**, ist eine integrierte Entwicklungsumgebung (→ IDE) Hat Funktionen wie Kompilierung, Syntax-Hervorhebung

**Microsoft Windows CE**, → Microsoft Windows Mobile

**Microsoft Windows Mobile**, Betriebssystem für mobile Geräte

**MIPS**, Microprocessor without interlocked pipeline stages, Prozessorarchitektur

**MNO**, Mobile Network Operator, Mobilfunkbetreiber

- mobil online**, Die erfassten Daten werden direkt auf der Server-Datenbank (über die Luftschnittstelle, oder USB) gespeichert.
- MSDN**, Microsoft Developer Network, beinhaltet Artikel und Referenzen für Softwareentwickler
- MSIL**, Microsoft Intermediate Language
- Multiprozessor-System**, Ein Computer der mehrere Prozessoren besitzt.
- multithreadingfähig**, Threads laufen nebeneinander
- MySQL**, Unternehmen, das das gleichnamige Datenbanksystem herstellt
- mysqldump**, Tool von → MySQL. Ermöglicht die Erstellung eines Datenbankbackups.
- NetBSD**, → Unix Derivat. → BSD
- ODBC**, Open Database Connectivity, Datenbankschnittstelle (von Microsoft), bietet einheitliche Schnittstelle für verschiedene Datenbanken. → JDBC benötigt einen Treiber vom Datenbank hersteller
- Offline-Funktionalität**, Bei der Datenspeicherung muss keine Netzwerkverbindung des mobilen Gerätes zum fixen Server bestehen. Die Daten werden vorübergehend temporär gespeichert. → temporäre Speicherung
- OKAPI**, Object Kernel API
- OLAP**, Online Analytical Processing
- OPIE**, Open Palmtop Integrated Environment
- Oracle Lite**, ist ein → DBS für → PDAs
- OSEK-OS**, Betriebssystem für eingebettete Komponenten, Handelszeichen der Siemens AG
- PalmOS**, Palm Operating System
- PDA**, Personal Digital Assistant, kleiner, tragbarer Computer, hat Funktionen wie z.B. Terminplaner, Tabellenkalkulation, Textverarbeitung, → WLAN, → Bluetooth, → GPS
- Pocket PC**, Pocket Personal Computer, PDA, auf dem ein Windows Betriebssystem läuft
- portieren**, das Lauffähigmachen eines Programmes auf einem anderen System
- PostgreSQL**, Ist ein → DBS
- PowerPC**, Prozessorarchitektur
- PSE**, Program and System Engineering, Programm- und Systementwicklung ist ein Unternehmen des Siemens Konzerns
- QBE**, Query by Example
- RAID**, Redundant Array of Independent Disks
- RAID**, Redundant Array of Independent Disks
- RAM**, Random Access Memory
- redo**, nochmals tun
- redundant**, → Redundanz
- Redundanz**, Informationen die mehrfach vorhanden sind
- SDE**, Smart Device Extensions
- SE**, Sony Ericsson. Joint Venture der Firmen Sony und Ericsson
- Sprachsyntax**, → Syntax bezogen auf die Programmiersprache
- SQL**, Structured Query Language, strukturierte Abfragesprache für Datenbanken
- SQL-Kommando**, Kommandos, die an eine Datenbank zur Abarbeitung gesendet werden.
- SQL-Statement**, → SQL-Kommando
- Stored Procedure**, In Stored Procedures können mehrere → SQL-Kommandos zusammengefaßt und am → Datenbankserver gespeichert werden.
- Support**, Unterstützung des Kunden durch den Lieferanten bei Problemen
- Symbian OS**, Symbian Operating System
- Synchronisation**, Abgleichen
- Syntax**, Die Grammatik einer Programmiersprache

- temporäre Speicherung**, hier zur Zwischenspeicherung von Daten in einer Datenbank
- Thread**, Teil eines Prozesses
- Transaktion**, Werden benötigt, um mehrere SQL-Kommandos vollständig oder gar nicht auszuführen
- Trigger**, Führen in Datenbanken bei Auftreten bestimmter Ereignisse bestimmte Aufgaben durch
- Tupel**, Ein Tupel ist ein Datensatz und somit eine Zeile einer Datenbanktabelle
- Typsicherheit**, Code, der nur auf Speicheradressen zugreifen darf, für die er die Berechtigung hat.  
[web25]
- UIQ**, User Interface Quartz
- UML**, Unified Modeling Language
- UMTS**, Universal Mobile Telecommunications System, Teil der 3. Mobilfunkgeneration 3G.  
Ermöglicht höhere Datenraten als → GSM, → GPRS oder → EDGE
- undo**, annullieren
- Unicode**, Zeichenkodierung
- Unix**, Betriebssystem
- VxWorks**, Proprietäres Echtzeitbetriebssystem von der Firma Wind River Systems
- WAN**, Wide Area Network, Weitverkehrsnetz
- WAP**, Wireless Application Protocol, Protokolle und Techniken, um auf langsamen Leitungen  
Internetinhalte verfügbar zu machen
- Windows Vista**, neuestes Windows Betriebssystem
- WLAN**, Wireless Local Area Network, drahtloses lokales Funknetz
- XML**, Extensible Markup Language, Auszeichnungssprache
- XOR**, Exklusiv-Oder-Verknüpfung
- Zwischencode**, Ist jener Code, der von einer Laufzeitumgebung ausgeführt werden kann.