

Mobiles Computer Aided Facility Management

Olaf Buck
PietschCONSULT GmbH
Osterweide 8
23562 Lübeck
olaf.buck@pietsch-luebeck.de

Robert Gajcy
Resco GmbH
Millerntorplatz 1
20359 Hamburg
robert.gajcy@resco.de

Volker Linnemann
Universität zu Lübeck
Institut für
Informationssysteme
Ratzeburger Allee 160
23538 Lübeck
linnemann@ifis.uni-luebeck.de

Abstract: Das Gebiet des Computer Aided Facility Management (CAFM) beschäftigt sich mit der Unterstützung der Verwaltung und Bewirtschaftung von Gebäuden, Anlagen und Einrichtungen in ihrer Nutzungsphase durch die Informationstechnik. Während nicht-mobile CAFM-Systeme inzwischen weit verbreitet sind, befindet sich der Bereich der mobilen CAFM-Systeme noch weitgehend im Anfangsstadium. Mobilität bedeutet in diesem Zusammenhang, dass Liegenschaftsdaten nicht nur von einem stationären PC aus abrufbar und änderbar sind, sondern auch von einem mobilen Gerät. Dieses kann direkt im betrachteten Gebäude von Personen, die mit der Bewirtschaftung und Verwaltung eines Gebäudes betraut sind, verwendet werden. In dieser Arbeit werden die Entwicklung von Benutzerschnittstelle und Architektur eines mobilen CAFM-Systems beschrieben, das im Rahmen einer Diplomarbeit als Erweiterung des seit langem im Einsatz befindlichen CAFM-Systems **consultware®** der Firma PietschCONSULT GmbH in Lübeck realisiert wurde

1 Einleitung

Die GEFMA (German Facility Management Association), der Deutsche Verband für Facility Management, definiert Facility Management wie folgt: „*Facility Management ist ein unternehmerischer Prozess, der durch die Integration von Planung, Kontrolle und Bewirtschaftung bei Gebäuden, Anlagen und Einrichtungen (Facilities) und unter Berücksichtigung von Arbeitsplatz und Arbeitsumfeld eine verbesserte Nutzungsflexibilität, Arbeitsproduktivität und Kapitalrentabilität zum Ziel hat.*“ [GEF]. Ein CAFM – System [Nä00] ist ein Werkzeug für das Facility Management mit einem durchgängigen Informationssystem und einer weitgehend redundanzfreien Speicherung. CAFM-Systeme mit einem zentralen Server, auf die nur über stationäre PCs zugegriffen werden kann, sind inzwischen weit verbreitet. Im Gegensatz dazu ist das Gebiet der mobilen CAFM-Systeme weitgehend im Anfangsstadium. Mobilität bedeutet hier, dass die Daten eines CAFM-Systems auch von einem mobilen Gerät (z.B. ein PDA) abrufbar und veränderbar sind. Beispiele für solche mobil abrufbaren Daten sind Raumgröße oder Daten über Gefahrenstoffe, die sich in einem Raum befinden.

In dieser Arbeit werden grundsätzliche Architekturfragen sowie die Benutzerschnittstelle eines mobilen CAFM-Systems beschrieben, das im Rahmen einer Diplomarbeit als Erweiterung des seit langem im Einsatz befindlichen CAFM-Systems **consultware®** realisiert wurde.

2. Das CAFM – System **consultware®**

consultware® ist ein CAFM-System, das vom Lübecker Unternehmen PietschCONSULT GmbH entwickelt wurde. **consultware®** wird unter anderem am Universitätsklinikum Schleswig-Holstein eingesetzt.

consultware® beinhaltet die Funktionsbereiche *Fläche*, *Reinigung*, *Schlüssel*, *Miete*, *Technik*, *Gefahrstoff* und *Brandschutz*.

Fläche bewältigt alle Aufgaben des modernen Flächenmanagements. Es können klassische Raumdaten, normierte Nutzungsdaten und frei definierbare Parameter verwaltet werden. Zur sinnvollen Verwaltung der Liegenschaft können Zeichnungen in gängigen Formaten integriert werden.

Reinigung stellt eine moderne Umsetzung des Gesamtprozesses Unterhaltsreinigung mit all seinen Facetten dar. Integrierte Prozesse wie die ergebnisorientierte Qualitätssicherung, die taggenaue Abrechnung von Normal- und Sonderleistungen, die Revierplanung oder das Auftrags- und Reklamationswesen bieten in der Praxis erprobte und funktionelle Lösungen.

Schlüssel realisiert die Verwaltung von Zutrittsberechtigungen auf grafischer Basis. Hierzu werden alle benötigten Daten wie Räume, Türen, Personen, Schlüssel, Zylinder und Lesegeräte in einem geschlossenen System geführt. Es werden die an Personen ausgegebenen Schlüssel verwaltet, so dass sich auf einen Blick feststellen lässt, wer welchen Schlüssel besitzt und welche Räume durch welche Personen geöffnet werden können.

Miete behandelt die Kosten für die Nutzung von Räumen. Der Anwender wird unterstützt, den Einsatz vorhandener Flächen zu optimieren und so die Wirtschaftlichkeit zu verbessern. Leerstehende und ungenutzte Räume können abgefragt und bei Bedarf schnell anderen Nutzern zur Verfügung gestellt werden.

Technik ist speziell auf die Anforderungen der Betriebstechnik zugeschnitten. Technische Informationen zu Gebäuden und Räumen, zum Inventar und zu Anlagen werden verwaltet. Wartungen, Reparaturen und Aufträge lassen sich effizient planen und abwickeln und das integrierte Störmanagement hilft bei der Verbesserung der Servicequalität. Die Struktur der zu verwaltenden Daten kann selbst bestimmt werden, so dass ein auf spezielle Anforderungen zugeschnittenes System entsteht.

Gefahrstoff schafft eine Verbindung zwischen CAFM und Arbeitssicherheit. Es ermöglicht die raumbezogene Führung und Kontrolle eines Gefahrstoffkatasters und bietet die Möglichkeit, die relevanten Rechtsvorschriften sicher umzusetzen. In Verbindung mit *Brandschutz* können Informationen über Gefahrstoffe zur Erstellung von Feuerwehrplänen genutzt werden. Zusätzlich kann *Gefahrstoff* als zentrale Informationsquelle für Arbeits- und Gefahrstoffe eingesetzt werden. Zu diesem Zweck ist eine umfangreiche Stoffdatenbank integriert.

Brandschutz verwaltet Brandschutzeinrichtungen und erlaubt es, Brandschutzpläne vorschriftsmäßig und kostengünstig selbst zu erstellen. Die Verwaltung von Lösch- und Schutzeinrichtungen basiert auf einer zentralen Datenbank, die die technische Ausstattung wie Feuerlöscher und Brandmelder für den Brandschutz enthält.

Alle Funktionen des Systems werden durch Grafiken unterstützt. Unter anderem bietet *consultware*® eine detaillierte Gebäudeübersicht, deren Einzelheiten hier nicht erläutert werden, da sie für das Verständnis eines mobilen CAFM-Systems nicht wesentlich sind.

3 Grundsätzliche Architekturfragen für ein mobiles CAFM-System

In diesem Abschnitt sollen einige grundsätzliche Fragestellungen in Bezug auf mobiles CAFM angesprochen werden. Außerdem wird erläutert, welche Vorteile ein mobiles CAFM prinzipiell bieten kann.

3.1 Warum mobiles CAFM ?

Stationäre Systeme sind normalerweise nicht vor Ort direkt im verwalteten Gebäude nutzbar, weil dort in der Regel kein Zugriff auf das CAFM-System besteht. Ein Zugriff über das WWW scheidet meistens aus, da die Kunden, die das Gebäude nutzen, die Verwendung ihrer PCs zu derartigen Zwecken meist nicht gestatten. Um CAFM wirklich mobil im verwalteten Gebäude nutzen zu können, ist daher ein System notwendig, welches den Zugriff zu den Daten über mobile Geräte wie PDAs gestattet.

Natürlich sind bei einem mobilen CAFM-System die Beschränkungen zu berücksichtigen, die ein mobiles Gerät wie ein PDA gegenüber einem stationären PC hat. Beispielsweise ist eine grafische Gebäudeübersicht in der Regel nicht möglich. Hier muss man sich auf die Anzeige von wichtigen Daten beschränken. Hinzu kommt noch, dass nicht alle Informationen, die ein stationäres System liefert, mobil benötigt werden. Diese im mobilen Einsatz unerheblichen Informationen würden die beschränkten Kapazitäten eines mobilen Gerätes unnötig belasten.

Die folgenden beiden Unterabschnitte sollen 2 grundsätzliche Alternativen für die Architektur eines mobilen CAFM-Systems betrachten. Diese betreffen die Frage der Aufteilung zwischen mobilem Gerät (Client) und dem zentralen Teil des CAFM-Systems (Server).

3.2 Thin-Client

Das Konzept der „Thin-Client-Architektur“ bedeutet, dass der Client seine Daten möglichst vollständig und bei jedem Einsatz neu von einem Server bezieht. Die funktionale Ausstattung der Client-Anwendung beschränkt sich im Wesentlichen auf die Ein- und Ausgabe, wobei die Einsatzumgebung auf dem Client durch einen konventionellen Browser gegeben ist.

Der große Vorteil einer Thin-Client-Architektur ist, dass eine Veränderung der Daten auf dem Server bei allen Clients direkt sichtbar wird. Insofern ist ein solches System

plattformunabhängig, wartungsarm und hochgradig skalierbar. Allerdings ist die Komplexität der Benutzeroberfläche bei einer browserbasierten Lösung durch die eingeschränkten Möglichkeiten von HTML beschränkt. Hinzu kommt noch, dass der Server von außen stets erreichbar sein muss, was beispielsweise in der consultware®-Architektur nur im LAN der Fall ist, weil sich dort die Datenbank befindet. Eine Freigabe für das WLAN oder das Internet würde Sicherheitsrisiken mit sich bringen, die nicht tolerabel erscheinen. Nicht zu vernachlässigen sind die Kosten, die für den Client entstehen. Existiert in der Einrichtung ein WLAN, so kann dieses meist für das mobile CAFM-System kostenlos mitbenutzt werden. Wenn die Verbindung über GMS oder UMTS erfolgt, können die Kosten schnell explodieren. Neben den Kosten stellt das zukünftige Einsatzgebiet des Systems ein großes Problem dar. Die Benutzung von Mobiltelefonen ist beispielsweise in Krankenhäusern sehr sensibel, weil dadurch Störungen an Medizingeräten (z. B. an Beatmungsgeräten) verursacht werden können. Dies hat zur Folge, dass in sehr vielen Krankenhausbereichen der Betrieb von Mobilfunk untersagt wurde.

3.3 Fat-Client

Das Konzept des „Fat Client“, auch „Rich-Client“ genannt, bedeutet, dass die eigentliche Verarbeitung der Daten vor Ort auf dem Client vollzogen wird. Dieser stellt dafür eine grafische Oberfläche zur Verfügung. Die wesentlichen Vorteile sind:

Möglichkeit der Offline-Anwendung (keine Verbindungskosten, Einsatz in „Funklöchern“) sowie die lokale Datenhaltung und damit eine schnelle Verarbeitung ohne lange Wartezeiten. Diesen Vorteilen stehen folgende Nachteile gegenüber. Um die Konsistenz der Daten zu gewährleisten, muss ein Synchronisations- und Transaktionskonzept entworfen werden. Bei einer Weiterentwicklung der Anwendung muss bedacht werden, dass die Clients nicht automatisch über die Neuheiten verfügen und evtl. einzeln manuell auf den neuesten Stand gesetzt werden müssen.

Folgende Tabelle zeigt eine Gegenüberstellung der beiden Ansätze:

	Vorteile	Nachteile
Thin-Client	<ul style="list-style-type: none"> - Plattformunabhängig (es wird lediglich ein Browser benötigt) - keine Update-Probleme - wenig Ressourcen auf dem Client notwendig 	<ul style="list-style-type: none"> - Gestaltung der Benutzeroberfläche durch HTML eingeschränkt - ständige Verbindung erforderlich
Fat-Client	<ul style="list-style-type: none"> - Umfangreiche Benutzeroberfläche - Offline-Verwendung - lokale Behandlung der Interaktionen 	<ul style="list-style-type: none"> - evtl. Update-Probleme - Synchronisation erforderlich - Transaktion erforderlich

Das Einsatzgebiet spielt eine entscheidende Rolle bei der Wahl der Architektur. Eine Architektur, die eine ständige Verbindung zum Server erfordert, ist in vielen Einsatzgebieten nicht realisierbar, einerseits wegen der Kosten, andererseits weil kein Mobilfunknetz oder WLAN-Netz zur Verfügung steht.

3.4 Auswahl der Rauminformationen

Die Ortung bzw. Auswahl eines Raumes sollte einfach gehalten werden. Der Benutzer, der vor Ort mit einem PDA mit dem System arbeitet, soll möglichst unmittelbar die Informationen zum gewünschten Raum geliefert bekommen. Eine Ortung über GPS scheidet wegen der Ungenauigkeit von GPS sowie wegen des teilweise ungenügenden Empfangs in Gebäuden aus. Stattdessen können mit einem Barcode versehene Türschilder verwendet werden. Die eindeutigen Strichcodes können für jeden Raum in einer relationalen Datenbank hinterlegt werden. Diese Lösung wurde für das realisierte mobile CAFM-System verwendet, da in consultware® bereits entsprechende Funktionen zur Generierung und Verwaltung der Barcodes zur Verfügung gestellt werden.

4 Ein mobiles CAFM-System

In diesem Abschnitt wird ein mobiles CAFM-System beschrieben, welches auf dem System consultware® aufsetzt. Um ein möglichst universell einsetzbares mobiles CAFM-System zu realisieren, wurde für das hier beschriebene System eine Fat-Client-Lösung realisiert, d.h. der Client (z.B. ein PDA) realisiert die mobil benötigten Funktionen selbstständig und kann offline ohne Verbindung zum Server betrieben werden. Zu diesem Zweck ist es möglich, die für eine konkrete Verwendung (ein Gebäude oder eine Gruppe von Gebäuden) relevanten Daten auf ein PDA herunterzuladen und mit diesen Daten zu arbeiten und sie auch im PDA zu verändern. Die veränderten Daten werden später auf den Server hochgeladen, sobald eine Verbindung existiert. Hierbei ergibt sich das Problem der Datenkonsistenz, wenn mehrere Clients gleichzeitig mit den gleichen Daten arbeiten.

Synchronisation von Daten bedeutet immer, eine Strategie zu entwickeln, um Inkonsistenzen zu vermeiden. Eine Lösung wäre es, die heruntergeladenen Daten auf dem Server zu sperren, so dass niemand anderes mit diesen Daten arbeiten und sie verändern kann. Eine solche Lösung ist jedoch zu unflexibel. Die Anwendungen haben gezeigt, dass Konflikte praktisch nie auftreten, weil es in der Praxis sehr selten mehr als eine Person gibt, die gleichzeitig die gleichen Daten verändert. Es wurde daher eine Lösung ähnlich der Lösung von CVS [CVS] gewählt, die es prinzipiell erlaubt, dass mehrere Personen gleichzeitig Daten verändern, die aber versucht, die Konflikte aufzulösen, jedoch dem Anwender beim Einchecken der Daten auf dem Server die Möglichkeit gibt, manuell zu überprüfen, ob die Konfliktauflösung sinnvoll ist und den tatsächlichen Gegebenheiten entspricht. Die Interaktion des Benutzers wurde hierbei bewusst gefordert. So verfügt er über große Beeinflussungsmöglichkeiten und ist jederzeit über den Abgleichvorgang informiert. Die bisherigen Erfahrungen mit dem System haben gezeigt, dass eine solche pragmatische Lösung sehr sinnvoll ist, weil sie die Anwender nicht so stark einschränkt wie ein strikter Checkin/Checkout – Mechanismus dies tun würde.

4.1 Benutzerschnittstelle

Zur Realisierung der Benutzerschnittstelle wurde ein handelsübliches PDA verwendet. Die Anschaffungskosten inklusive des Scanaufsatzes betragen derzeit weniger als 300 €

4.1.1 Arbeitsweise des mobilen Gerätes

Die Arbeitsweise des mobilen Gerätes wird in Abb. 1 demonstriert.

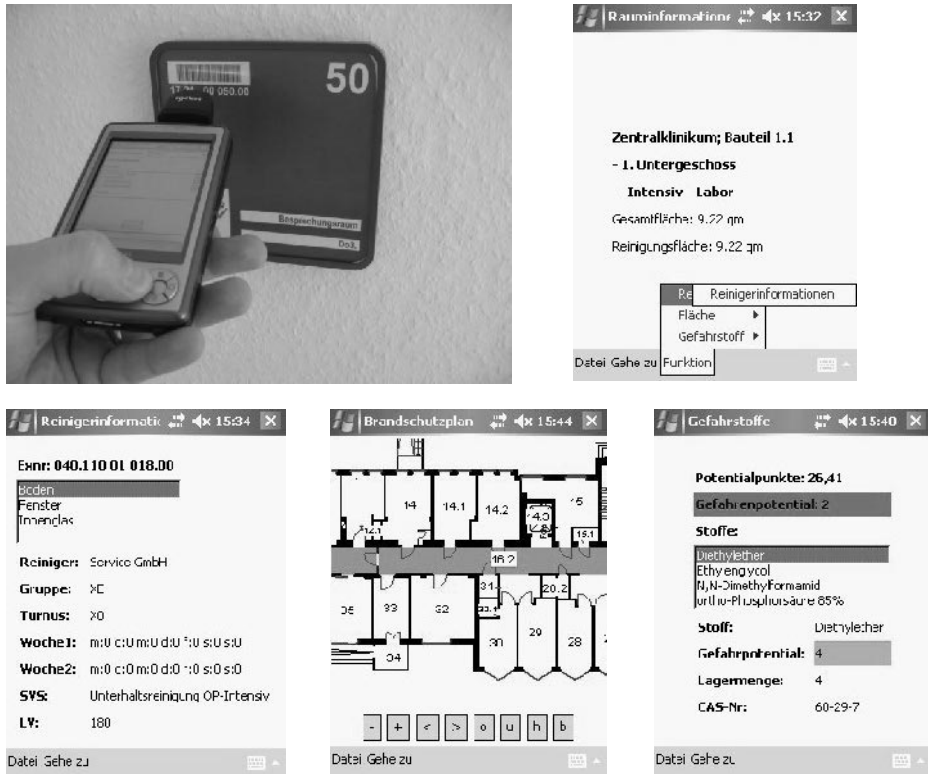


Abbildung 1: Arbeitsweise des mobilen Gerätes

Das linke Bild in der oberen Reihe von Abb. 1 zeigt die Erfassung des Barcodes eines Raumes. Anhand der gespeicherten Daten kann das System direkt den Raum erkennen und Basisinformationen wiedergeben (rechtes Bild in der oberen Reihe von Abbildung 2). Das rechte Bild zeigt außerdem ein Menü zur Auswahl weiterer Informationen. Beispiele für solche Informationen gibt die untere Reihe von Abb. 1.

4.1.2 Arbeitsweise des Systemclients zur Datensynchronisation

Zur Sicherstellung der Konsistenz müssen die Daten der mobilen Geräte mit dem Hauptsystem synchronisiert werden. Die Synchronisation der Daten wird über einen separaten Systemdialog gemäß Abb. 2 durchgeführt.

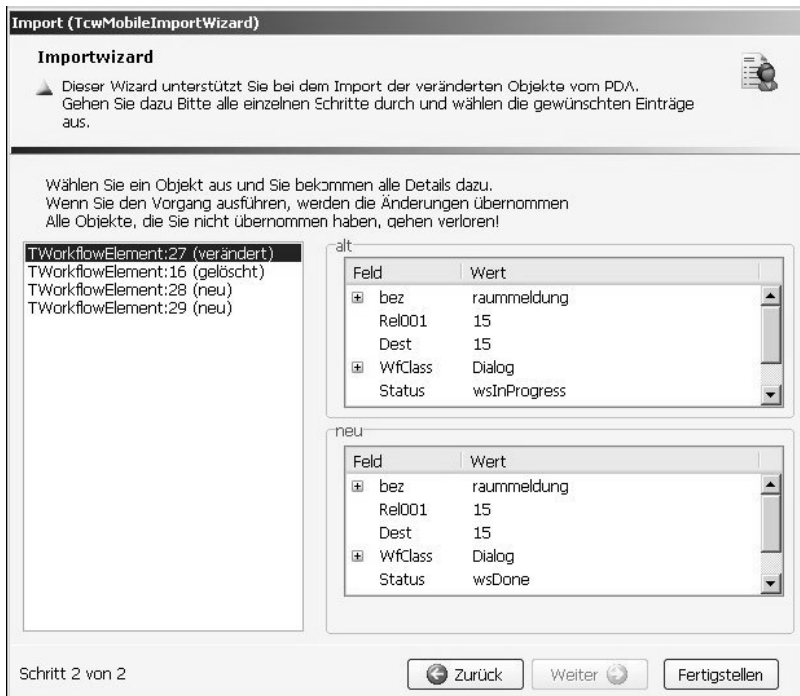
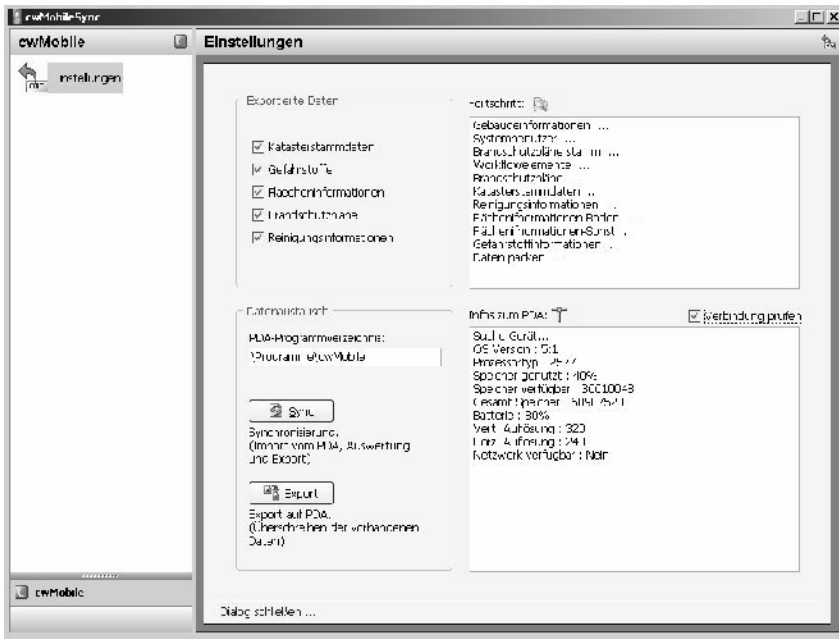


Abbildung 2: Export auf den PDA und Import vom PDA (Synchronisation)

4.1.3 Anwendungsbeispiele

Beispiel 1: Der Beauftragte macht seinen wöchentlichen Rundgang. Er entdeckt dabei diverse Mängel, die vom Hausmeister beseitigt werden sollen. Der Beauftragte erzeugt neue Störungsmeldungen, die in das Workflow-Management einfließen. Nach der Rückkehr erfolgt ein Datenabgleich mit der Datenbank und der Hausmeister wird beim nächsten Systemstart über neue Nachrichten informiert.

Beispiel 2: Der Hausmeister ist nach einem dreiwöchigen Urlaub wieder auf seinem Arbeitsplatz. In der Zwischenzeit haben sich mehrere Störungsmeldungen angesammelt. In einigen Räumen sind neue Glühbirnen notwendig und ein Raum muss nach einem Arbeitsunfall komplett renoviert werden. Nach einem Export der Daten auf das mobile Gerät geht der Hausmeister in das Gebäude. Aus dem Gerät können die Orte sowie die Störungsnachrichten abgelesen werden. Der Hausmeister beseitigt diese gemeldeten Mängel vor Ort, antwortet eventuell an die Sender und kennzeichnet die Meldungen als erledigt. Nach der Rückkehr von seinem Rundgang werden die Daten abgeglichen, so dass die Datenbank auf dem aktuellen Stand ist.

Beispiel 3: Der Brandschutzbeauftragte erzeugt Brandschutzpläne für alle Ebenen der Einrichtung. Die zuständige Feuerwehr wird mit dem mobilen System ausgestattet und kann im Brandfall bereits auf dem Weg zum Einsatzort die betroffenen Gebäude begutachten. So können schon unterwegs wichtige Entscheidungen und Vorbereitungen getroffen werden (z.B. die Ortung der Hydranten oder die Ansicht der Gefahrenpotentiale).

Beispiel 4: Der Hausmeister bekommt den Auftrag, alle Räume eines Gebäudes neu zu vermessen. Er nimmt das Gerät mit und vermisst alle Räume nacheinander. Die ermittelten Daten werden in das System eingegeben und beim nächsten Synchronisationsvorgang für alle berechtigten Systembenutzer zur Verfügung gestellt.

4.2 Implementierung

Die Entwicklungsumgebung für das aktuelle consultware®-System ist „Borland Delphi 6“, welches „Object Pascal“ verwendet. Derzeit wird eine Migration nach Java durchgeführt. Als Datenbankserver kommt z.Zt. der von Sybase vertriebene „Advantage Database Server“ (ADS) zum Einsatz. Dieses Produkt ist ein relationales Datenbank Management System, welches eine Client/Serverarchitektur ermöglicht. Für die Persistenz der Objekte ist ein OR-Mapping-Framework zuständig, das im Rahmen von Studienarbeiten [GB02],[Ga04] entstanden ist und kontinuierlich weiterentwickelt wurde. Dieses Framework hat sich während der Systementwicklung als sehr hilfreich erwiesen, direkte Datenbankzugriffe werden nur in Ausnahmefällen verwendet. Es schafft einen übersichtlichen Quellcode sowie eine getrennte Schicht, die in eigene Klassen gekapselt für die Persistenz zuständig ist.

Da es auf der Ebene der mobilen Clients mehrere Betriebssysteme gibt, nämlich in erster Linie Palm OS und Windows Mobile, war es ein wesentliches Ziel, die Software für die mobilen Clients plattformunabhängig zu erstellen. Daher fiel als Programmiersprache die Wahl auf Java mit der Plattform Java Micro Edition (J2ME), da hier eine virtuelle Maschine für alle relevanten Betriebssysteme existiert. Dadurch kann die Implementierung relativ unabhängig von der zukünftigen Entwicklungsumgebung stattfinden.

Der Datenexport auf das mobile Gerät geschieht im XML-Format. Hierfür wird eine bereits existierende XML-Exportschnittstelle des consultware®-Systems verwendet. Die XML-Daten fließen mit Hilfe des Castor-Frameworks [Ca] in die Anwendung ein. Dieses erlaubt es, Daten aus XML-Dokumenten in Java-Objekte bzw. Java-Objekte in XML-Dokumente zu überführen. Zur Verringerung des zu übertragenden Datenvolumens werden die XML-Daten komprimiert. Auf das verwendete Komprimierungsverfahren kann hier aus Platzgründen nicht eingegangen werden.

Damit bei Synchronisationsvorgängen eine richtige Zuordnung der bearbeiteten Objekte hergestellt werden kann, wurde für die Organisation dieser Informationen eine eigene Verwaltungsklasse eingeführt. Eine Instanz dieser Klasse entspricht der Darstellung einer Objektänderung. Zu den Informationen, die abgelegt werden, gehören die Klassenbezeichnung, die Relation (Primärschlüssel), die Änderungsart sowie der Name der XML-Datei, die alle weiteren Informationen enthält. Veränderungen werden somit auf dem mobilen Gerät aufgezeichnet und können später am Systemclient abgeglichen werden.

Für die Darstellung von grafischen Komponenten kommt die Abstract Window Toolkit (AWT)-Bibliothek von Java zum Einsatz. Es gibt zwar vereinzelt Java Virtual Machines für mobile Geräte, die Swing unterstützen, was jedoch eher die Ausnahme als die Regel ist.

Eine erste Prototypversion des mobilen CAFM wurde im Rahmen einer Diplomarbeit realisiert, die bei der Firma PietschCONSULT GmbH erstellt wurde [Ga06].

5. Zusammenfassung und Ausblick

In dieser Arbeit wurden die Architektur und der Prototyp eines mobilen Computer Aided Facility Management Systems vorgestellt. Obwohl dieser Prototyp die wesentlichsten Funktionen eines mobilen CAFM-Systems bereits enthält, sind doch wichtige Funktionen noch zu realisieren, bevor ein solches System bei Kunden eingesetzt werden kann. Bei diesen Funktionen handelt es sich vor allem um

Schutz der Daten. Zur Zeit liegen alle XML-Dokumente in einem ZIP-Archiv. Sie werden auf dem PDA zur Laufzeit daraus entnommen. Erfahrene Benutzer könnten einzelne Daten extrahieren, ansehen und ggf. verändern. Um dies zu vermeiden, ist geplant, die Bouncy Castle Crypto API [Bo] zu verwenden. Hierbei handelt es sich um Implementierungen von kryptographischen Algorithmen, die von der "Legion of the Bouncy Castle" erstellt wurden. Die Besonderheit liegt in der leichtgewichtigen Implementierung, so dass diese API ohne Probleme im Rahmen von J2ME verwendet

werden kann. Dies ist der wesentliche Unterschied zu der „Java Cryptography Extension“ (JCE), welche seit der Version 1.4 im SDK von Sun enthalten ist [Zi03].

Vereinfachung des Exports der Daten. Es ist prinzipiell nicht notwendig, die Stammdaten (z.B. Gebäude) jedes Mal neu zu exportieren. Das kostet unnötig Zeit und sollte nur bei Bedarf durchgeführt werden. Die Realisierung dieses Konzepts erfordert eine Trennung der entsprechenden Daten in den XML-Darstellungen, so dass eine getrennte Exportierung und Datenübertragung möglich wird.

Literaturverzeichnis

- [Bo] The Legion of the Bouncy Castle, siehe <http://www.bouncycastle.org/>
- [Ca] The Castor Project, siehe <http://www.castor.org/>
- [CVS] Concurrent Version System, <http://www.nongnu.org/cvs/>
- [GEF] German Facility Management Association – Deutscher Verband für Facility Management, <http://www.gefma.de>
- [GB02] André Gode, Olaf Buck: Object Mapping von relationalen Datenbanken. Studienarbeit am Institut für Informationssysteme der Universität Lübeck, durchgeführt bei der Firma PietschCONSULT, 2002
- [Ga04] Robert Gajcy: Refactoring und Verbesserung eines Objekt-Mapping-Frameworks. Studienarbeit am Institut für Informationssysteme der Universität Lübeck, durchgeführt bei der Firma PietschCONSULT, 2004
- [Ga06] Robert Gajcy: Mobiles Computer Aided Facility Management, Diplomarbeit am Institut für Informationssysteme der Universität Lübeck, durchgeführt bei der Firma PietschCONSULT, 2006
- [Nä00] Jens Nävy: Facility Management – Grundlagen, Computerunterstützung, Einführungsstrategie, Praxisbeispiele. 2. Auflage. Springer-Verlag Berlin Heidelberg New York, 2000
- [Zi03] Zimmermann T.: JCA & Bouncy Castle. Fachhochschule Aargau: Seminar Enterprise Computing, 2003