FAKULTÄT FÜR !NFORMATIK

Die Communication Center Management Plattform

Eine Webapplikation zur Hardwaresteuerung/-überwachung

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Magister der Sozial- und Wirtschaftswissenschaften

im Rahmen des Studiums

Wirtschaftsinformatik

ausgeführt von

Martin Kulnigg
Matrikelnummer 9925684

am:
Institut für Informationssysteme

Betreuung:
Betreuer/Betreuerin: Univ.-Prof. Dr. Georg Gottlob
Mitwirkung: Univ.-Ass. Marcus Herzog

Wien, 15.08.2008

_____ (Unterschrift Verfasser/in)

_____ (Unterschrift Betreuer/in)

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Anwendungsfalldiagrammverzeichnis

Zusammenfassung

Das Communication Center ist ein Fernüberwachungs- und Steuerungsgerät mit eingebauten GSM Modem, das zum Messen von Daten und zum Steuern von Geräten eingesetzt werden kann. Die Verwaltung der Geräte erzeugt in Szenarien, in denen mehrere Communication Center dezentral eingesetzt werden, einen erheblichen Aufwand. Um die Geräteeinstellungen zu ändern, müssen sie entweder vor Ort aufgesucht oder über das GSM Modem fernkonfiguriert werden. Die gemessenen Daten können nicht kumuliert aus den Geräten ausgelesen werden und somit ist die Beobachtung der Daten über einen längeren Zeitraum schwierig. Es soll eine Webapplikation entwickelt werden, die diese Probleme löst und die Verwaltung der Geräte vereinfacht. Die Geräte sollen über ein benutzerfreundliches Webinterface wartbar sein und sollen aus der Applikation heraus automatisiert konfiguriert werden. Um das Problem der Datenaufzeichnung zu lösen, soll eine Funktionalität zum automatischen Auslesen und Speichern der aufgezeichneten Daten geschaffen werden. Diese Daten sollen in ein Standardformat wie Excel exportierbar sein und statistisch ausgewertet werden können. Die vorliegende Arbeit beschäftigt sich mit der Entwicklung der Applikation, sie beschreibt diesen Prozess von den technischen Grundlagen, über die Analyse bis hin zur fertigen Applikation.

Abstract

The Communication Center is a remote monitoring and control system with a built-in GSM modem which can be used for measuring data and for controlling the devices. Administration of these devices can mean a considerable load of work in scenarios in which several remote Communication Centers are used. To change the set-up of devices they must be visited on site or configured by remote control via a GSM modem. The measured data cannot be read out of the devices in a cumulated way. So the observation of data over a longer period is difficult to manage. A web-application is to be developed which solves those problems and makes administration of the devices easier. The devices are to be maintained via an easy-to-use web-interface and are to be configured automatically by the application. To solve the problem of data logging a function for automatic reading and storing of the registered data is to be realized. These data are to be exported into a standard format like Excel and to be processed statistically. This work deals with the development of the application, describes the process starting from the technical basis to the analysis and finally to the executable application.

1 Einleitung

Die vorliegende Arbeit befasst sich mit der Entwicklung von CCMP, der Communication Center Management Plattform. CCMP ist eine Webapplikation, die die gesamte Funktionalität zur Verwaltung von Communication Centern(CC) anbieten soll. Das Communication Center ist ein Mess- bzw. Überwachungsgerät mit eingebautem GSM Modem, das, je nach angeschlossenen Sensoren, verschiedenste Werte messen oder Steuerfunktionen übernehmen kann.

Der einleitende Teil soll die Problemstellung erläutern und anhand eines kurzen Beispiels veranschaulichen. Daraus ergibt sich die Motivation für diese Arbeit. Danach wird das Projektumfeld kurz beschrieben. Es folgt die Erörterung der Zielsetzung der Arbeit, welche die erfolgreiche Entwicklung von CCMP ist. Im Anschluss daran befindet sich eine kurze Übersicht über den weiteren Aufbau der Arbeit.

1.1 Problemstellung

Die Problemstellung besteht darin, ein oder mehrere Communication Center, die räumlich verteilt sind, möglichst einfach, effizient und benutzerfreundlich zu verwalten.

In einem typischen Szenario mit mehreren Geräten muss jedes dieser Geräte einzeln konfiguriert werden. Das geschieht über einen Computer mit analogem Modem, mit Hilfe dessen der Benutzer sich mit dem Gerät verbindet. Dieser Prozess ist recht zeitintensiv, da das User Interface der Geräte nicht sehr benutzerfreundlich ist und setzt bei Benutzung eines anlogen Modems gewisse technische Grundkenntnisse voraus. Man kann mit 20 bis 30 Minuten pro Gerätekonfiguration rechnen. Außerdem kann es für den Benutzer sehr frustrierend sein, wenn er bei mehreren Geräten die gleiche Konfiguration immer und immer wieder eingeben muss.

Beim Archivieren der geloggten Daten steht der Benutzer vor einem ähnlichen Problem wie bei der Konfiguration der Geräte. Er muss bei jedem Gerät entweder über SMS oder über einen Verbindungsaufbau die Daten anfragen. Hat ihm das Gerät die Daten geschickt, so muss er sich um die Aufbereitung und Archivierung derselben Daten kümmern. Das kann für den Benutzer relativ zeitaufwendig sein, da die Geräte die Daten praktisch unformatiert schicken und er sich selbst darum kümmern muss, ob eine Datenzeile schon archiviert wurde oder nicht.

Zusammenfassend kann man sagen, dass die Verwaltung des Communication Center aufwendig, technisch anspruchsvoll, aufgrund der langen Verbindungszeiten kostenintensiv, und nicht sehr benutzerfreundlich ist.

1.2 Beispiel

Als Beispiel soll folgendes Szenario dienen: Die Firma Entenhausen Energie möchte das Communication Center zur Überwachung von Muffenbunkern einsetzten. Ein Muffenbunker ist ein Raum, der unter der Erde liegt und nur schwer begehbar ist. Über die Muffenbunker laufen die Hauptstromleitungen von Entenhausen Energie, die die Stadt mit Strom versorgen. Die einzelnen Bunker liegen räumlich etwa 15 – 30 Gehminuten auseinander. Entenhausen Energie möchte nun alle 12 Muffenbunker überwachen und bestellt daher 12 Communication Center bei der Firma Communications Easy. Bei jedem Muffenbunker sollen die Temperatur, möglicher Wasser- oder Gaseintritt sowie der Eingang des Bunkers gegen unbefugten Zutritt überwacht werden.

Sobald die Geräte installiert sind, muss sich ein Techniker der Entenhausen Energie darum kümmern, dass die Geräte richtig konfiguriert werden. Zu diesem Zweck benötigt er einen Computer an dem ein analoges Modem angeschlossen ist. Nun muss er mit Hilfe des Modems zu jedem Gerät eine Verbindung aufbauen und die gewünschte Konfiguration eintragen. Diese ist allerdings bei allen Geräten ident und so führt er dieselbe Arbeit 12x hintereinander aus. Will er die Konfiguration ändern steht er vor demselben Problem.

Da Entenhausen Energie die gemessenen Werte auch archivieren will, muss der Techniker in regelmäßigen Abständen die geloggten Daten von den Geräten abfragen. Das kann entweder über einen Verbindungsaufbau mit dem Modem geschehen oder indem er eine SMS an jedes Gerät schickt. Das Gerät übermittelt ihm dann die Daten, welche vom Techniker aufzubereiten und zu archivieren sind.

In diesem Szenario, in dem 12 Geräte verwaltet werden, lässt sich ein nicht unbeträchtlicher Aufwand für den Techniker feststellen, sofern das System ordnungsgemäß konfiguriert und die geloggten Daten in regelmäßigen Abständen archiviert werden.

1.3 Motivation

Als Motivation für dieses Projekt stehen auf der einen Seite die Gründe der Firma LeP Lehotzki electronic Products GmbH, die wirtschaftlich und kundenorientiert motiviert sind und auf der anderen Seite der Wunsch des Entwicklers sich auf technisches Neuland zu begeben.

Die Firma Lehotzki will das Communication Center auch für technisch nicht versierte Kunden bedien- und wartbar machen und somit eine größere Kundengruppe erreichen. Durch die Plattform will man sich außerdem gegenüber Konkurrenzprodukten differenzieren, die nicht automatisiert konfigurier- und steuerbar sind.

Für den Entwickler stellt dieses Projekt die Möglichkeit dar, eine sehr spezielle, in sich einzigartige Software zu entwickeln, die einige essentielle Probleme des CCs lösen soll. Die technische Herausforderung, die vor allem im Design des User Interfaces und der Kommunikationsschnittstelle liegt, ist für ihn die Hauptmotivation.

Der größte gemeinsame Motivationsfaktor für beide Parteien ist es ein Produkt zu schaffen, das am österreichischen Markt bei Projektstart in dieser oder ähnlicher Form noch nicht existiert hat und das in vielen verschiedenen Szenarien einsetzbar ist.

1.4 Das Projekt und sein Umfeld

Das Projekt wird im Dezember 2004 gestartet. Auftraggeber ist die LeP Lehotzki electronic Products GmbH, die das Communication Center entwickelt hat und es verkauft. Auf der anderen Seite steht die Firma Metamagix Software & Consulting GmbH als Auftragnehmer. Martin Kulnigg, Entwickler bei Metamagix soll die Entwicklung der Plattform umsetzen.

Die Projekt Dauer ist anfangs für ca. 6 Monate angesetzt, es werden aber am Ende etwas mehr als 2 Jahre daraus werden, da es im Laufe des Projekts zu Problemen mit dem Modem und der Kommunikationsschnittstelle kommt, die im Vorfeld nicht abzusehen waren.

Das erste Projekt wurde gemeinsam mit der Firma Wienstrom 2005 umgesetzt. Bei diesem Projekt werden 12 CCs eingesetzt und die CCMP Plattform läuft auf einem eigens dafür angeschafften Server. In diesem Projekt wird CCMP sowohl zur Verwaltung der Geräte als auch zum Archivieren der Messdaten eingesetzt.

2006 belegt CCMP beim E-Biz Award des Report Verlages in Wien den dritten Platz.

1.5 Ziel der Arbeit

Das Ziel der Arbeit ist die Beschreibung des Softwareentwicklungsprozesses, der zu Erstellung von CCMP geführt hat, mit Hilfe von anerkannten und weitläufig eingesetzten Methoden und Werkzeugen des Software Engineering. Es soll gezeigt werden, wie es von einer konsistenten und sauberen Analyse zur Implementierung der Software kommt, die dann verschiedene Tests durchlaufen muss, um zu einer fertigen Applikation zu reifen.

Der technische Aspekt der Arbeit soll soweit gehen, dass ein grundlegendes Verständnis für die notwendigen technischen Grundlagen zur Umsetzung eines solchen Projektes geschaffen wird. Es soll prinzipiell möglich sein, mit den hier behandelten Technologien und Werkzeugen eine ähnliche Applikation wie CCMP zu entwickeln.

Der praktische Teil dieser Arbeit ist die Entwicklung von CCMP, der Communication Center Management Plattform und hat die Fertigstellung der Applikation zum Ziel. CCMP soll die komplette Verwaltung des Communication Centers abdecken.

Es soll gezeigt werden, dass es möglich ist das Communication Center oder ähnliche Geräte über eine Webplattform zu verwalten. Der Benutzer soll sich nicht mehr um technische Details zum Kommunikationsaufbau kümmern müssen, sondern soll eine möglichst einfache und gut zu bedienende Benutzerschnittstelle zur Verfügung gestellt bekommen.

Ein weiteres Ziel von CCMP ist es, die Prozesse zur Initialisierung bzw. Änderung der Gerätekonfiguration sowie dem Datenlogging zu automatisieren , d.h. der Benutzer muss nur noch auf einen Knopf drücken oder den gewünschten Zeitpunkt für den Start des Prozesses eingeben und der Prozess läuft dann eigenständig ab. Diese Automatisierung soll dazu führen, dass die Kommunikationskosten pro Gerät erheblich gesenkt werden.

Daneben soll eine benutzerfreundliche Schnittstelle zur Auswertung der geloggten Daten geschaffen werden, die es dem Benutzer ermöglicht die Daten in ein gängiges Format (Excel, CSV, etc.) zu exportieren und statistisch auszuwerten.

1.6 Aufbau der Arbeit

Die Arbeit gliedert sich in insgesamt 6 Kapitel.

1. Einleitung
2. Technische Grundlagen
3. Anforderungsanalyse
4. Die Applikation
5. Testen
6. Ausblick

Die Einleitung beschreibt das Problem, die Motivation und Zielsetzung sowie das Projektumfeld der Arbeit. Das Kapitel „Technische Grundlagen“ befasst sich mit technischen Bereichen, die für die Entwicklung von CCMP relevant sind und daher bekannt sein müssen. Im Kapitel „Anforderungsanalyse“ geht es vor allem um die Erarbeitung und Dokumentation der

Anforderungen für die Erstellung der Plattform. Dieses Kapitel ist Voraussetzung für das Kapitel „Applikation“, das die eigentliche Umsetzung der Plattform beschreibt. Es befasst sich mit der Architektur der Applikation und es werden die einzelnen Bereiche bzw. Funktionalitäten der Software im Detail betrachtet. Das fünfte Kapitel „Testen“ dokumentiert die im Zuge der Fertigstellung von CCMP durchgeführten Tests und beschreibt einzelne Testmethoden im Detail. Im letzten Kapitel „Ausblick“ werden Weiterentwicklungen und mögliche zukünftige Projekte angeführt.

2 Technische Grundlagen

Dieses Kapitel beschäftigt sich mit den technischen Bereichen, die die Grundlage für die Entwicklung von CCMP bilden. Diese sind das Communication Center, das TC-35 Terminal von Siemens, das pppd Protokoll und eSSENCE™, ein Java Framework. Es soll eine grundlegende Wissensbasis zu diesen Themen vermittelt werden, die für die Entwicklung der Webplattform notwendig ist.

2.1 Das Communication Center

*„Das **Communication Center (CC)** ist ein Fernüberwachungs- und Fernsteuergerät für kleine dezentrale Anlagen. Digitale und analoge Eingangssignale können überwacht werden und senden im Störungsfall SMS oder E-Mails, digitale Ausgänge können per SMS geschalten werden. Die (Fern-) Parametrisierung erfolgt über einen geräteinternen Webserver - ohne zusätzlich zu installierende Software.“* [LEPF07, Seite 3]

Ein Communication Center funktioniert wie folgt: Für jedes CC kann eine bestimmte Anzahl von Empfängern festgelegt werden. Weiters hat jedes CC hat eine bestimmte Anzahl von analogen und digitalen Ein- und Ausgängen, sogenannten I/Os, an die verschiedene Sensoren zum Messen von Daten angeschlossen werden bzw. können mittels der Ausgänge andere Geräte geschaltet werden. Für die I/Os können Meldungen angelegt werden, die an Bedingungen gekoppelt sind. Sobald eine der Bedingungen erfüllt ist, wird die entsprechende Meldung per SMS oder Email über das GSM Modem an die angegebenen Empfänger geschickt. Dafür benötigt jedes Communication Center eine eigene SIM-Karte.

Das Communication Center steht im Focus des dieser Arbeit zugrundeliegenden Projekts und im Folgenden werden die verschiedenen Verbindungsmöglichkeiten, die Parametrisierung am Gerät, die verschiedenen Einsatzszenarien sowie die verschiedenen Modelle des Communication Center betrachtet.

2.1.1 Der Verbindungsaufbau mit dem Communication Center

Um das Communication Center zu parametrisieren, muss zuerst eine Verbindung mit dem Gerät hergestellt werden. Hier gibt es die drei folgenden Möglichkeiten:

- **Direktverbindung:** Bei der Direktverbindung werden der Computer und das CC mittels eines ausgekreuzten Netzwerkkabels (Ethernet Kabel - Cross Over) verbunden. Beide Geräte müssen dieselbe Subnetmask (255.255. 255.0) verwenden. Die Default-Einstellungen des CC sind Subnetmask: 255.255.255.0 und IP-Adresse:

192.168.1.24. Am PC muss dann für die LAN-Verbindung eine IP-Adresse im Bereich 192.168.1.XXX angegeben werden. Sind die Einstellungen korrekt, kann das Gerät im Internet Browser mit der Adresse http://192.168.1.24 erreicht werden. [vgl. LEPB05, Seite 6]

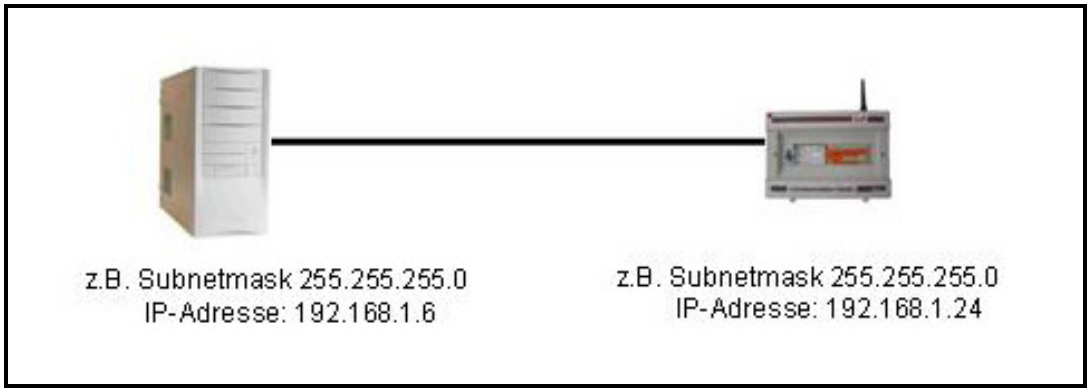


Abbildung 1: Aufbau einer Direktverbindung mit dem Communication Center

- **Verbindung über ein Netzwerk:** Das Communication Center kann über einen Router, Hub oder Switch an ein Netzwerk angeschlossen werden. Verbindet man das Communication Center mit dem Netzwerk, so muss diesem eine fixe IP-Adresse (innerhalb des Netzwerkes) und die Subnetmask des Netzwerkes zugewiesen werden. Zur Verbindung verwendet man ein Standard-Ethernet-Kabel. [vgl. LEPB05, Seite 6]

- **Verbindung über eine DFÜ-Verbindung:** Zum Aufbau der DFÜ-Verbindung benötigt man ein analoges Modem, das am PC angeschlossen wird. Als Telefonnummer wird die Nummer der SIM Karte eingetragen, die sich im Communication Center befindet. Die IP-Adressen sind in diesem Fall festgelegt. Username und Passwort sind ebenfalls voreingestellt (Username: myusername, Passwort: mypassword). Sobald eine Verbindung zwischen dem Modem und dem Communication Center besteht, kann das Communication Center in einem Internet Browser über die Adresse http://190.1.1.2 erreichen werden. [vgl. LEPB05, Seite 7]

Die Direktverbindung ist vor allem bei Erstinbetriebnahme oder bei Störungen der Geräte interessant, da ein Techniker vor Ort die Parametrisierung zum Testen der angehängten Sensoren vornehmen möchte. Die Verbindungsmöglichkeit per Modem ist für die Fernparametrisierung gedacht und ermöglicht die Änderung der Einstellungen von jedem beliebigen PC aus.

2.1.2 Die Parametrisierung

Sobald eine Verbindung mit dem Communication Center erstellt ist, kann es über einen Webbrowser erreicht werden. Man erhält zunächst eine allgemeine Übersicht über die verschiedenen Bereiche des Communication Centers:

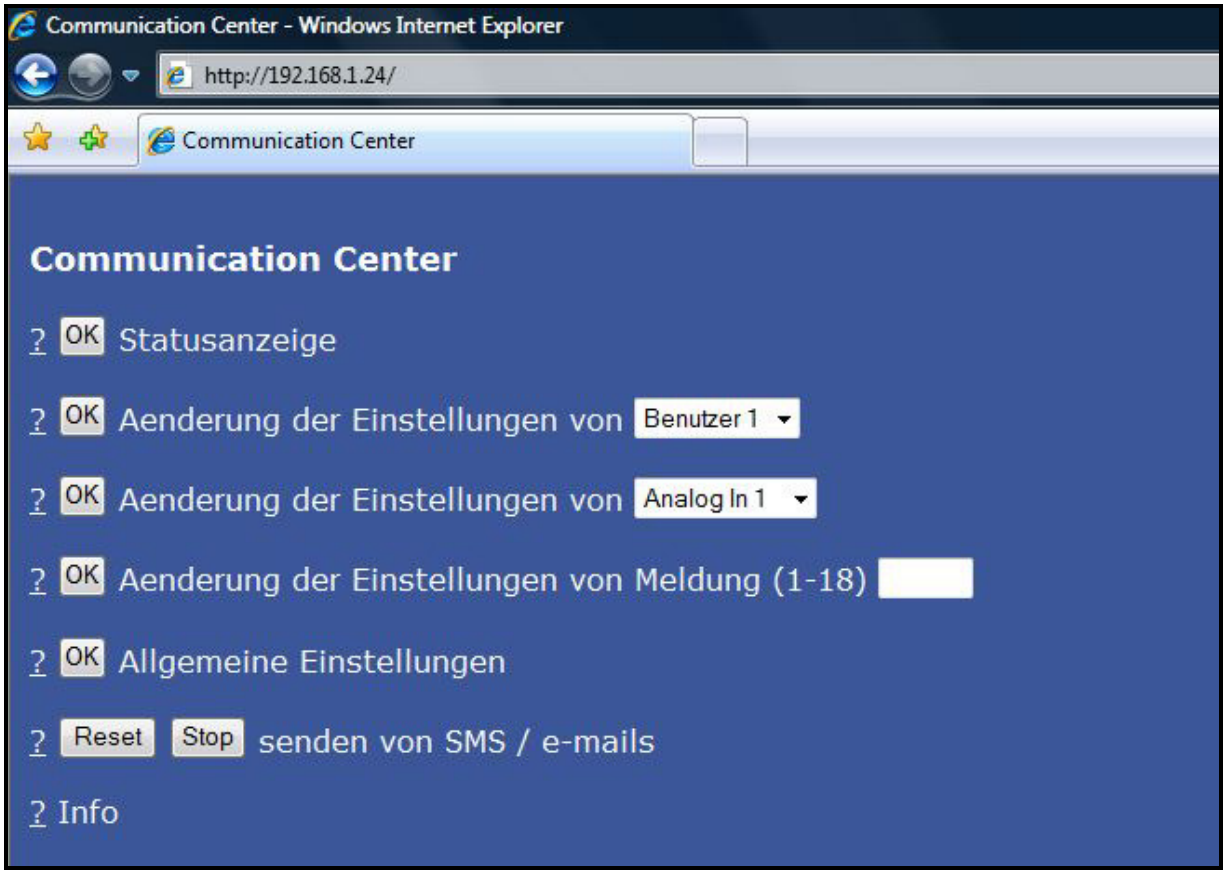


Abbildung 2: Communication Center Userinterface - Allgemeine Übersicht

Es können folgende Einstellungen bzw. Parameter eingeben und verändert werden:

- **Benutzer:** Attribute der SMS/E-Mail Empfänger
- **IO(Ein- und Ausgänge):** Attribute der I/Os
- **Meldungen:** Attribute der Meldungen
- **Allgemeine Einstellungen:** Globale Parameter, die für den Betrieb des CCs notwendig sind.
- **Administrator Einstellungen:** Sicherheitskritische Einstellungen.

Beim Ändern der Parameter werden immer die aktuellen Daten angezeigt. Diese können überschrieben und mit dem „Send“-Button an das Communication Center übertragen werden, welches die Daten entsprechend verarbeitet und speichert. Dann schickt das CC die gespeicherten Daten zurück, welche mit den gesendeten Daten verglichen werden. Stimmen

diese überein, so sind die Daten im CC korrekt gespeichert worden. [vgl. LEPB05, Seite 9,10]

2.1.2.1 Benutzerdaten

Für die Benutzer kann eine Telefonnummer und eine Email Adresse eingegeben werden. Die Telefonnummer sollte folgendem Format entsprechen: +Ländercode, Vorwahl (ohne 0), Nummer. Wird keine Nummer/Email Adresse eingegeben so wird an den Benutzer auch keine SMS/Email gesendet. Pro Gerät können 8 verschiedene Benutzer angegeben werden. [vgl. LEPB05, Seite 10]

2.1.2.2 I/O Daten

Für jeden I/O kann ein Name (max. 9 Zeichen) spezifiziert werden. Wird kein Name angegeben so ist der I/O inaktiv (ausgeschalten) und scheint in keiner Meldung auf. Für analoge Eingänge kann eine Skalierung des Wertebereichs (Untergrenze, Obergrenze) sowie eine Skalierungseinheit definiert werden (z.B. Grad, Volt, etc.). Für Temperatureingänge darf der Wertebereich der Skalierung nicht verändert werden, da diese sonst neu geeicht werden müssen. Digitale I/O können invertiert werden. [vgl. LEPB05, Seite 10]

2.1.2.3 Meldungsdaten

Grundsätzlich unterscheidet man zwischen 2 Arten von Meldungen: Durch Eingänge ausgelöste Meldungen und Statusmeldungen. Es können insgesamt 18 Meldungen pro Gerät festgelegt werden.

Durch Eingänge ausgelöste Meldungen

Alle digitalen und analogen Eingänge können zum Auslösen (Triggern) von Meldungen verwendet werden. Eine detailliertere Ausführung findet sich in [vgl. LEPB05, Seite 11,12]. Folgende Daten werden für eine Meldung eingegeben:

- **Triggerbedingung**: Die Triggerbedingung ist die Bedingung, die für den ihr zugewiesenen I/O eintreten muss, damit eine Nachricht (Email und/oder SMS) verschickt wird. Sie besteht aus dem Identifikationskürzel des Eingangs (z.B. AI1 für Analog Eingang 1 oder DI1 für Digital Eingang 1), einem Operator (<,> bei Analogen Eingängen, = bei digitalen Eingängen) und einem Wert.
- **Verzögerungszeiten:** Diese Zeiten geben an wie lange eine Bedingung bereits gültig/nicht mehr gültig sein muss, damit eine Meldung geschickt wird. Das Minimum ist 10 Sekunden.
- **Meldungstext:** Ein beliebiger Text. (max. 50 Zeichen)

- **Email- bzw. SMS-Empfänger:** Pro Meldung können je 8 Empfänger für den Email- bzw. SMS-Versand angegeben werden.

Statusmeldungen

Statusmeldungen sind Meldungen, die in periodischen Abständen geschickt werden. Für die Triggerbedingung muss hier „Stat“ eingegeben werden. Die Verzögerungszeiten sind fix, der Meldungstext ist wieder beliebig wählbar. Für Statusmeldungen können die Wochentage (1 = Montag, 2=Dienstag…) angegeben werden, an denen sie verschickt werden sollen. Diese werden nur als Email versendet und zwar jeweils um 10:00 Uhr an den ausgewählten Tagen. Für eine Statusmeldung kann eine Liste mit Empfängern definiert werden.

2.1.2.4 Allgemeine Einstellungen

In den allgemeinen Einstellungen können folgende Attribute verändert werden, die herangezogene Quelle ist [vgl. LEPB05, Seite 12]:

- **Kopfzeile**: Der Name der Anlage. Er wird in jeder gesendeten Email und SMS mitgeschickt.
- **Email-Nr.**: Die Nummer des Email Providers.
- **Anzahl Prioritätsmeldungen**: Wird eine Meldung als Prioritätsmeldung deklariert, so wird sie immer sofort übermittelt. Meldungen, die keine Prioritätsmeldungen sind, werden, wenn sie im Zeitraum zwischen Start- und Stoppzeit auftreten, unterdrückt und erst später gesendet. Wird z.B. „3“ eingegeben, so sind die Meldungen 1-3 Prioritätsmeldungen und die Meldungen 4-18 Meldungen mit niedriger Priorität.
- **Startzeit**: Beginn des Zeitraums in dem Meldungen mit niedriger Priorität nicht gesendet werden.
- **Stoppzeit**: Ende des Zeitraums in dem Meldungen mit niedriger Priorität nicht gesendet werden.

2.1.2.5 Administratoren Einstellungen

Die Administratoren Einstellungen beinhalten alle sicherheitsrelevanten Einstellungen und sind von der Basis-Seite aus nicht zu erreichen. Um die Administratoren-Seite zu erreichen muss im Webbrowser http://192.168.1.24/sc?e=yes eingegeben werden.

Folgende Parameter können hier eingestellt werden:

- **Subnetmask des Gerätes** (erfordert einen Neustart)
- **IP-Adresse des Gerätes** (erfordert einen Neustart)

- **3 Basis Nummern des Gerätes:** Es ist möglich drei Telefonnummern am Gerät anzugeben, die sich ohne Authentifizierung einwählen können.
- **Code**: Der Code muss jeder SMS, die an das CC geschickt wird, vorangestellt werden und dient der Authentifizierung.

Außerdem können im Bereich „Administratoren Einstellungen“ die Temperatur Eingänge geeicht werden. [vgl. LEPB05, Seite 13]

2.1.2.6 Statusanzeige

Im Bereich Statusanzeige findet man eine Auflistung aller aktiven I/Os, wobei der I/O Name und der momentane Wert des angezeigt wird. Für digitale Ausgänge können die Werte auch hier verändert werden. [vgl. LEPB05, Seite 13]

2.1.3 Einsatzszenarien für das Communication Center

Das Communication Center bietet aufgrund seiner I/Os eine Schnittstelle für viele verschiedene Sensoren. Es ist sowohl für den Einsatz im Freien als auch in Gebäuden konzipiert und kann mit Hilfe einer eigenen Stromversorgung auch in Szenarien ohne ausreichende externe Infrastruktur eingesetzt werden. Das führt dazu, dass das CC für viele verschiedene Szenarien einsetzbar ist. Einige davon werden im Folgenden kurz ausgeführt.

2.1.3.1 Überwachung und Steuerung von Straßenbeleuchtung

Das Problem ist, dass die Straßenbeleuchtung in vielen österreichischen Gemeinden sehr veraltet ist und einen hohen Energieverbrauch aufweist. Die Gemeinden haben das Potential für Einsparungen erkannt und es wird damit begonnen, die Straßenbeleuchtungen zu erneuern. Das Communication Center kann hier folgendermaßen eingesetzt werden: Es wird in einen zentralen Verteiler eingebaut, der mehrere Straßenzüge steuert. Das CC soll den Ausfall von Lampen und Schaltgeräten melden, den Energieverbrauch der Straßenzüge messen und kontrollieren sowie das automatische Ein- und Ausschalten bzw. das Absenken der Lampen steuern. Die Beleuchtungszeiten können im CC angegeben werden. Das Referenzprojekt für dieses Szenario ist die Gemeinde Felixdorf in Niederösterreich. Nachfolgende Abbildung zeigt die Aufgaben des eingebauten CC: [vgl. LEPF07, Seite 6]

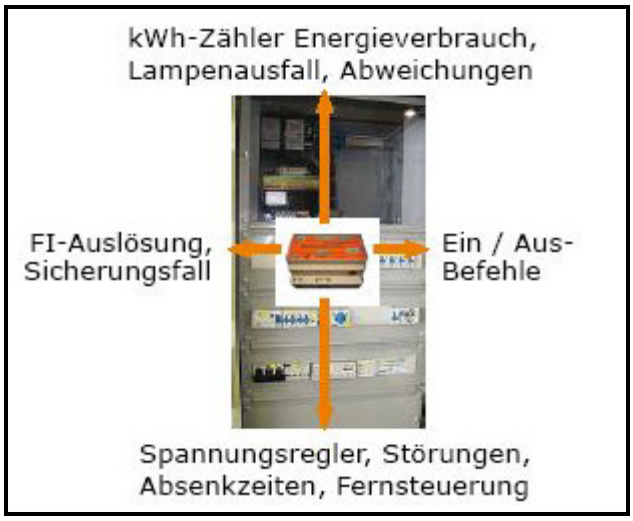


Abbildung 3: Das Communication Center in einem Verteilerkasten

2.1.3.2 Brandmeldung

In diesem Szenario soll das Communication Center als Erweiterung zu einer klassischen Brandmeldezentrale eingesetzt werden. Das CC hat die Aufgabe, die Brandmelder zu überwachen und im Alarmfall das Einsatzpersonal direkt per SMS zu benachrichtigen. Weiters kann es bei einer Störung der Brandmelder das Servicepersonal verständigen. Dadurch wird die Sicherheit der Brandmelde-Anlage und das Servicelevel erhöht. Durch die Funktionalität des Datenloggens werden alle Ereignisse chronologisch aufgezeichnet und können archiviert werden. [vgl. LEPF07, Seite 7]

2.1.3.3 Wasseraufbereitung und Überwachung von Sammelbecken

Wasseraufbereitung erfordert das Messen verschiedenster Werte, wobei Temperatur-,pH-,Lf- und Redoxwerte gemessen werden. Sollten diese Werte gewisse Grenzen überschreiten, so müssen sofort Gegenmaßnahmen ergriffen werden, um die Qualität des Wassers zu sichern. In diesem Fall übernimmt das CC die sofortige Alarmierung der zuständigen Personen. Das CC kann auch zur Fernsteuerung von Pumpen und Ventilen eingesetzt werden. Da der Bestand an Konditionierungsmitteln mit dem CC kontrolliert werden kann, ist es außerdem möglich, automatische Bestellungen an die zuständigen Lieferanten zu schicken. Bei Sammelbecken werden die Niveaumessungen vom CC übernommen und die mühsamen Messungen vor Ort können entfallen. Das CC kann bei fehlender Stromversorgung vor Ort auch mittels einer kleinen Solarzelle und einer Pufferbatterie versorgt werden. [vgl. LEPF07, Seite 8]

2.1.3.4 Überwachung von Serverräumen

In Serverräumen besteht ein hoher Bedarf an Überwachung, da sich auf den Servern oft wichtige und sensible Daten befinden bzw. ein Teil des Kerngeschäftes vieler Unternehmen

über diese Server abgewickelt wird. Gerade in den Sommermonaten kann es zu einem kritischen Temperaturanstieg kommen. Hier wird das Communication Center vor allem zur Überwachung des Raumklimas eingesetzt. Das CC kann zentral aufgestellt und verschiedenste Sensoren daran angeschlossen werden: Bewegungs- und Zutrittsmelder, Temperaturfühler, Feuchtigkeitsmesser, etc. Außerdem kann der Zutritt zu den Serverräumen überwacht werden. Der Vorteil besteht darin, dass das CC von der bestehenden Infrastruktur getrennt und an eine eigene Stromversorgung angeschlossen ist - daher bleibt es auch bei einem allgemeinen Stromausfall noch betriebsfähig. Für dieses Szenario gibt es eine fertige vorinstallierte Variante des Communication Center, bei der zur Inbetriebnahme nur die Sensoren entsprechend platziert werden müssen. [vgl. LEPF07, Seite 10]

2.1.3.5 Überwachung von Kesselanlagen

Die Überwachung von Kesselanlagen ermöglicht eine genau Messung und eine Optimierung des Energieverbrauchs. Es ist möglich, Störungen und Abweichungen vom Normalbetrieb praktisch sofort zu erkennen und entsprechend darauf zu reagieren. Diese können aufgrund der genauen Aufzeichnungen durch das Datenloggen auch im Nachhinein noch rekonstruiert werden. Das führt zu möglichen Einsparungen im Energieverbrauch, der durch schadhafte oder verschmutze Teile bzw. Unregelmäßigkeiten entstehen kann und erhöht außerdem die Wartungsqualität. [vgl. LEPF07, Seite 9]

2.1.3.6 Monitoring von Kühlgut

In Großküchen gelten sehr strenge Vorschriften bezüglich der Überwachung von Kühl- und Gefriereinrichtungen. Es muss vollständige Aufzeichnungen hinsichtlich Temperatur (siehe Abbildung 4) bzw. Luftfeuchtigkeit und Funktionalität geben. Diese Aufgaben kann das CC übernehmen. Außerdem ist es möglich, Störungen und Zutritte zu erkennen und bei Gefahr für die gekühlten Güter können die verantwortlichen Personen rechtzeitig alarmiert werden. Aufgrund der integrierten Funktionalität des Datenloggens können die händischen Aufzeichnungen des Betriebspersonals entfallen und es existieren im Falle von Störungen genaue Aufzeichnungen über das Geschehen. [vgl. LEPF07, Seite 9]

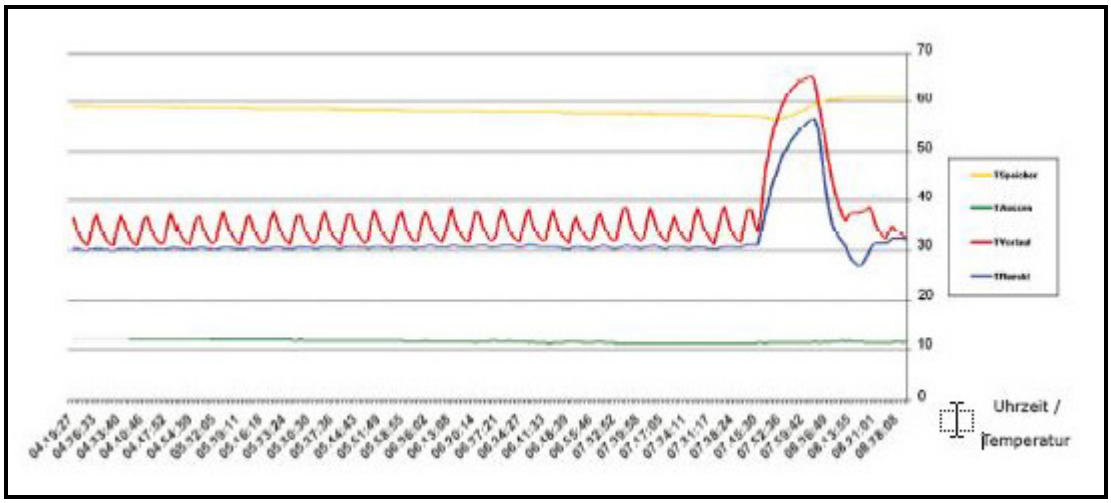


Abbildung 4: Eine Temperaturkurve eines Kühlhauses

2.1.4 Modelle des Communication Center

Zurzeit gibt es vier verschiedene Modelle des Communication Center, die sich entweder in der Anzahl ihrer Ein- und Ausgänge oder aber in der Ausstattung des Gehäuses bzw. der Spannungsversorgung unterscheiden. Das IP65 Gehäuse schützt die Geräte gegen Witterungsverhältnisse. Es ist daher zu empfehlen, wenn die Geräte im Freien eingesetzt werden. Die Auflistung der nachfolgenden Modelle ist [LEPDB05, Seite 2] entnommen:

- **CC2-426:** 6 digitale Eingänge,4 digitale Ausgänge, 2 Temperatureingänge (analog)
- **CC2-266:** 6 digitale Eingänge,2 digitale Ausgänge, 6 Multifunktionseingänge (analog)
- **CC3-426:** wie CC2-426 im IP65 Gehäuse und Spannungsversorgung
- **CC3-266:** wie CC2-266 im IP65 Gehäuse und Spannungsversorgung

2.2 Grundlagen der Kommunikation

Dieser Abschnitt soll einen Überblick über die einzelnen Komponenten geben, die für die Kommunikation mit dem Communication Center benötigt werden. Er umfasst eine Beschreibung des Point-to-Point Protocol, seiner konkreten Implementierung unter Linux sowie eine kurze Beschreibung des verwendeten GSM Modems.

2.2.1 Das Point-to-Point Protocol

„The Point-to-Point Protocol (PPP) provides a standard method for transporting multi-protocol datagrams over point-to-point links.“ [RFC1661]

Im Internet ist das Point-to-Point Protocol(PPP) für verschiedene Zwecke im Einsatz, wie zum Beispiel für die Verbindung zwischen 2 Routern oder zwischen einem Router und einem Modem. Es ist in RCF 1661 definiert und wird durch RFC 1662 bzw. RFC 1663 erweitert. [vgl. Tane03, Seite 268]

2.2.1.1 Merkmale

Das Point-to-Point Protocol ist durch folgende drei Merkmale charakterisiert, welche von Tanenbaum [vgl. Tane03, Seite 268] diskutiert werden:

- **Eine Rahmenbildungsmethode:** Dieses verwaltet die einzelnen Rahmen und kennzeichnet das Ende eines Rahmen und den Anfang des nächsten eindeutig. Die Fehlererkennung, die von PPP unterstützt wird, wird vom Rahmenformat übernommen.
- **Das Link Control Protocol(LCP):** Das Link Control Protocol ist ein Verbindungssteuerungsprotokoll. Es ist für die Aktivierung und das Testen von Leitungen, das Aushandeln von Optionen, sowie das Beenden von Verbindungen verantwortlich. Weiters kann es sowohl mit synchronen als auch asynchronen Leitungen bzw. byte- und bitorientierten Kodierungen umgehen.
- **Eine Gruppe von Network Control Protocols(NCP):** Das Point-to-Point Protocol implementiert eine Methode zum Aushandeln von Optionen auf der Vermittlungsschicht, die vom auf dieser Schicht benutzen Protokoll unabhängig sind. Die Methode ist dafür verantwortlich, dass auf jeder Vermittlungsschicht ein anderes Network Control Protocol läuft.

2.2.1.2 Verbindungsaufbau

IM PPP-Rahmenformat sieht der Ablauf vom Aufbau bis zum Trennen einer Verbindung folgendermaßen aus:

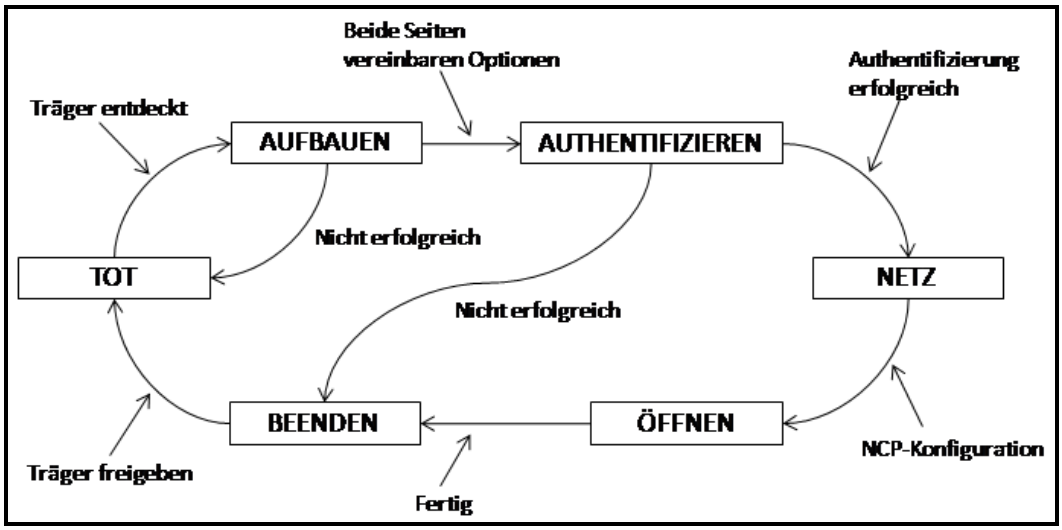


Abbildung 5: Die einzelnen Phasen des Verbindungsaufbaus des Point-to-Point Protocols nach Tanenbaum

Eine Leitung hat den Status „TOT“, wenn kein Träger auf physikalischer Ebene und keine Verbindungen auf der Bitübertragungsschicht existieren. Sobald eine physikalische Verbindung aufgebaut wurde, wechselt der Zustand der Leitung in „AUFBAUEN“. In dieser Phase werden die LCP-Optionen verhandelt. Ein Prozess macht Vorschläge, die der andere akzeptiert oder ablehnt, außerdem kann von LCP die Leitungsqualität festgestellt werden. Falls diese Phase erfolgreich ist, kommt es zur Authentifizierung, falls nicht, wird der Vorgang beendet. Im Status „Authentifizieren“ kommt es zu einer Prüfung der Identitäten der beiden Parteien und sollte diese erfolgreich sein, tritt die Leitung in die Phase „NETZ“ über. Hier wird das passende Network Control Protocol zur Konfiguration der Vermittlungsschicht aufgerufen. Bei einer erfolgreichen Konfiguration wird der Zustand „Öffnen“ erreicht und die Datenübertragung kann beginnen. Ist diese abgeschlossen, wechselt die Leitung in die Phase „BEENDEN“, in der die Verbindung korrekt beendet wird und von dort wieder nach „TOT“, sobald der Träger freigegeben wurde. [vgl. Tane03, Seite 270]

2.2.1.3 Rahmen

Das PPP-Rahmenformat setzt auf das HDLC-Rahmenformat auf, der Unterschied ist, dass PPP zeichenorientiert und HDLC bitorientiert ist.

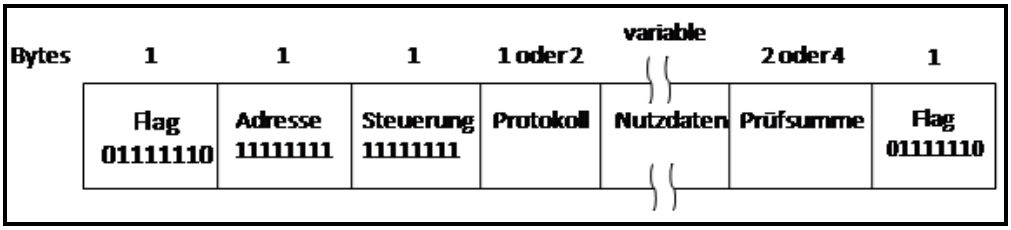


Abbildung 6: Die PPP-Rahmen nach Tanenbaum

Ein Rahmen ist laut Tanenbaum [vgl. Tane03, Seite 269] folgendermaßen aufgebaut:

- **Flag**: Jeder PPP Rahmen beginnt mit dem Flagbyte des HDLC-Standards (01111110).
- **Adresse:** Dieses Feld hat immer den Wert 11111111, was bedeutet, dass der Rahmen von allen Stationen akzeptiert werden soll.
- **Steuerung:** Der Standard Wert dieses Feldes ist 00000011. Er zeigt einen nicht nummerierten Rahmen an. Das gewährleistet keine zuverlässige Übertragung in rauschenden Umgebungen anhand von Folgenummern und Bestätigungen und daher kann bei Bedarf eine zuverlässige Übertragung mittels nummeriertem Rahmen erreicht werden.
- **Protokoll:** Definiert die Paketart, die sich im Feld Nutzdaten befindet.
- **Nutzdaten:** Dieses Feld hat eine beliebige Länge bis hin zu einem vereinbarten Maximum. Die Standardlänge beträgt 1500 Byte.
- **Prüfsumme:** Dieses Feld ist normalerweise 2 Byte, alternativ nach Vereinbarung 4 Byte groß.

Folgende LCP-Rahmen sind nach RFC 1661 definiert:

Bezeichnung	Richtung	Beschreibung
Configure-	I ↹ A	Liste der vorgeschlagenen Optionen und Werte
Configure-ack	I → A	Alle Optionen werden angenommen
Configure-nak	I → A	Einige Optionen werden nicht angenommen
Configure-reject	I → A	Einige Optionen können nicht verhandelt werden
Terminate-	I ↹ A	Anforderung zum Trennen der Verbindung
Terminate-ack	I → A	Verbindung wird getrennt
Code-reject	I → A	Unbekannte Anforderung erhalten
Protocol-reject	I → A	Unbekanntes Protokoll angefordert
Echo-request	I ↹ A	Anforderung, den Rahmen zurückzusenden
Echo-reply	I → A	Rückgabe des Rahmens
Discard-request	I ↹ A	Rahmen verwerfen(für Testzwecke)

Tabelle 1: Die LCP-Rahmen nach Tanenbaum

Die Richtung zeigt immer von welchem Teilnehmer der Impuls ausgeht. „I“ steht für den Initiator des Verbindungsaufbaus und „A“ für den Antwortenden. Die Rahmen vom Typ „Configure“ sind für das Aushandeln der Optionen zuständig, die „Terminate“-Codes dienen zum Beenden der Verbindung, die „Reject“-Rahmen zeigen an, dass der Antwortende etwas erhal-

ten hat, das er nicht versteht, „Echo“-Rahmen werden zum Testen der Leitungsqualität verwendet und der „Discard“-Request dient zu Testzwecken. [vgl. Tane03, Seite 271]

2.2.1.4 Ein Szenario für den Einsatz von PPP

Ein typisches Szenario für den Einsatz von PPP wäre folgendes: Ein privater Internet Nutzer möchte aus seinem PC über einen Anruf bei seinem Internet-Service-Provider einen temporären Internet Host machen. Der Heim PC ruft zuerst über das angeschlossene Modem den Router des Service-Providers an. Sobald das Modem des Routers reagiert und eine physikalische Verbindung aufgebaut hat, beginnt der PC damit, mit dem Router die zu verwendenden PPP-Parameter zu verhandeln. Er sendet ihm dafür mehrere LCP-Pakete und verwendet dafür ein oder mehrere PPP-Rahmen. Sind die Parameter fertig verhandelt, so werden mehrere NCP-Pakete zur Konfiguration der Vermittlungsschicht verschickt. Da der PC ein TCP/IP Protokoll ausführen will, benötigt er eine IP Adresse. Diese weist ihm der Router mittels NCP für die Dauer der Verbindung aus einem Set IP Adressen, die eigens dafür vorgesehen sind, zu. Nun ist der PC in der Lage IP-Pakete zu senden und zu empfangen. Bei Beendigung der Sitzung baut NCP die Verbindung zur Vermittlungsschicht ab, die IP-Adresse wird wieder freigegeben und kann dem nächsten PC zugewiesen werden. Dann beendet LCP die Verbindung auf Ebene der Sicherungsschicht. Als letztes sendet der PC dem Modem den Befehl „Auflegen“, wodurch die Verbindung auf der Bitübertragungsschicht beendet wird. [vgl. Tane03, Seite 268]

2.2.2 pppd

Der folgende Abschnitt beschäftigt sich mit pppd, den pppd Options, dem Authentifizierungsmechanismus von pppd, Chat und dem TC-35 Modem. Es soll ein Verständnis dafür geschaffen werden, wie unter Linux eine Punkt-zu-Punkt Verbindung aufgesetzt wird, welche Komponenten daran beteiligt sind und wie die Kommunikation mit einem Modem prinzipiell funktioniert. Die verwendeten Quellen sind „Linux – Installation, Konfiguration, Anwendung“ von Michael Kofler, die pppd(8)- Linux Man Page, die chat(8)-Linux Man Page, die TC-35 Terminal Command Spezifikation und die Bedienungsanleitung des TC-35 Terminal.

PPPD ist der Point-to-Point Protocol Daemon. Das Point-to-Point Protocol wird, wie in Abschnitt 2.2.1 beschrieben, dazu verwendet, unter Unix-ähnlichen Systemen Internet Verbindungen zwischen Dial-up Modems, DSL Connections und anderen Punkt zu Punkt Verbindungen aufzusetzen. PPPD arbeitet mit dem PPP Driver des Kernels zusammen, um die Verbindung aufzusetzen, zu verwalten und die für die Verbindung benötigten IP-Adressen auszuhandeln. Bei einem Verbindungsaufbau durch pppd, heißt die Seite von der die Kommunikation ausgeht „Client“ und die andere Seite mit der die Verbindung aufgebaut werden soll „Server“. [vgl. PPPD08]

2.2.2.1 Options

PPPD können verschiedene Optionen zur Konfiguration des Verbindungsaufbaus übergeben werden, diese können entweder auf der Command Line oder als File übergeben werden. Das File *options* liegt normalerweise im Verzeichnis etc/ppp/. Im Folgenden werden kurz die für das CCMP Projekt wichtigsten Optionen beschrieben, diese sind der pppd(8)- Linux Man Page [vgl. PPPD08] entnommen (Die kursiv geschriebenen Ausdrücke bezeichnen einen variablen Wert):

- ***ttyname*:** Der Name des seriellen Ports, über den mit dem Server kommuniziert werden soll. Falls *ttyname* nicht mit „/“ beginnt, so wird die Zeichenkette "/dev/" automatisch vor *ttyname* hinzugefügt um einen gültigen Device Namen zu formen.
- ***speed*:** Eine Dezimalzahl, die die gewünschte Baud Rate für das serielle Device angibt. Unter Linux werden nur allgemein verwendete Baud raten unterstützt.
- **connect *script*:** Mit dieser Option kann ein Script angegeben werden, das ausgeführt wird, bevor das PPP Protocol gestartet wird. Normalerweise ist es zu diesem Zeitpunkt notwendig, verschiedene Parameter an den Verbindungsträger zu übergeben, bei einem Dial-up Modem müssen zum Beispiel mehrere Parameter übergeben werden damit es die richtige Nummer wählt. Hierfür eignet sich chat.
- **disconnect *script***: Diese Option veranlasst die Ausführung eines Skripts nachdem PPPD die Verbindung beendet hat. Das kann zum Beispiel eine Reihe von Befehlen sein, die das Modem dazu veranlasst, korrekt zu terminieren und sich zu neu zu starten. Das Skript wird nicht ausgeführt wenn das Modem bereits aufgelegt hat.
- **file *name*:** Mit diesem Parameter kann ein individuelles File *name*, das Optionen enthält, übergeben werden.
- **idle *n*:** Diese Option gibt an, dass pppd beendet werden soll, wenn *n* Sekunden lang keine Datenpakete gesendet oder empfangen werden.
- **maxconnect *n*:** Durch diesen Parameter wird die Verbindung *n* Sekunden, nachdem das erste Netzwerkprotokoll aufgesetzt wurde beendet.
- **nolock:** Die Option nolock gibt an, dass pppd kein UUCP-ähnliches Lock File für das serielle Device anlegt. Das Lock File würde pppd einen exklusiven, alleinigen Zugriff auf das Device garantieren.
- **nopersist:** PPPD wird beendet sobald die Verbindung beendet wurde. Diese Option ist ein standardmäßig gesetzt.

2.2.2.2 Authentifizierung

Als Authentifizierung wird jener Prozess beschrieben, bei dem der Client den Server von seiner Identität überzeugt. Das funktioniert folgendermaßen: Der Client sendet seinen Namen und eine geheime Information, die eindeutig von ihm kommt, an den Server. Der Server antwortet dem Client indem er seinen Namen und einen Beweis, dass er die geheime Information kennt, zurückschickt. Bei der geheimen Information handelt es sich normalerweise um ein Password, das sich der Client und der Server teilen.

Im Moment unterstützt pppd drei Mechanismen zur Authentifizierung: PAP (Password Authentication Protocol), CHAP (Challenge Handshake Authentication Protocol) und EAP (Extensible Authentication Protocol). Wird PAP verwendet so sendet der Client seinen Namen und ein unverschlüsseltes Passwort an den Server, um seine Identität zu beweisen. Wird CHAP eingesetzt, so geht die Authentifizierung vom Server aus, er schickt ein sogenanntes Challenge Packet an den Client, das seinen Namen enthält. Auf dieses Packet muss der Client mit seinem Namen und der geheimen Information, die Client und Server teilen, antworten. EAP unterstützt eine CHAP ähnliche Authentifizierung, die um einen Sicherheitsmechanismus (SRP-SHA1) erweitert ist. Es ist prinzipiell möglich, dass beide Seiten einen unterschiedlichen Authentifizierungsmechanismus verwenden.

pppd speichert die geheimen Informationen je nach Authentifizierungsmechanismus in folgenden Files: /etc/ppp/pap-secrets, /etc/ppp/chap-secrets, /etc/ppp/srp-secrets. Jede Zeile eines solchen Files enthält genau eine geheime Information, die aus drei Feldern besteht: der Namen des Client, der Name des Servers und das Geheimwort. [vgl. PPPD08]

2.2.2.3 chat

Für diesen Abschnitt wurde die chat(8)-Linux Man Page [CHAT08] als Quelle herangezogen. chat ist ein Programm, das von pppd für den Verbindungsaufbau verwendet wird, es definiert eine skriptbasierte Konversation zwischen dem Computer und dem Modem. Primär dient chat dazu, die Verbindung zwischen den pppd Prozessen des Client und des Servers aufzusetzen.

An chat können verschiedene Optionen übergeben werden, die interessantesten für den Einsatz im CCMP Projekt sind (Die kursiv geschriebenen Ausdrücke bezeichnen einen variablen Wert):

- **f *<chat file>*:** Das chat Skript wird aus *<chat file>* gelesen. Es ist darauf zu achten, dass der Benutzer, der das File ausführt, die notwendigen Leseberechtigungen besitzt. Das File darf mehrzeilig sein.

- **t *<timeout>*:** Mit t kann ein Timeout gesetzt werden, das bestimmt wie lange auf eine erwartete Zeichenkette gewartet werden soll. Wenn die Zeichenkette nicht innerhalb des angegebenen Timeouts eintrifft, so wird keine Antwort gesendet. Alternativ kann eine andere Antwort gesendet werden oder das Skript wird abgebrochen.
- **r *<report file>*:** Diese Option gibt an in welches File der Output der verwendeten REPORT Zeichenketten geschrieben werden soll.
- **T *<phone number>*:** Mit diesem Parameter kann eine Telefonnummer angegeben werden, die im Skript für den Platzhalter \T ersetzt wird.

Das chat Skript definiert die Kommunikation mit dem Modem. Es besteht aus einem oder mehreren „expect-send"-Paaren von Zeichenketten, die mit Leerzeichen getrennt sind. Diese „expect-send"-Paaren können AT Befehle und die vom Modem erwartete Antwort sein oder zu folgenden Zeichenketten gehören (Die kursiv geschriebenen Ausdrücke bezeichnen einen variablen Wert):

- **ABORT *<Ausdruck>*:** Viele Modems geben den Status eines Anrufs als Zeichenkette zurück. Diese Zeichenketten können „CONNECTED", „NO CARRIER" oder „BUSY" sein. Ein gültiges „expect-send" Paar wäre „ABORT NO CARRIER" oder „ABORT BUSY". Diese beiden Paare hätten zur Folge, dass das Skript abgebrochen wird, sobald das Modem die Antwort „NO CARRIER" oder „BUSY" gibt.
- **SAY *<Ausdruck>*:** SAY wird verwendet um Zeichenketten am Terminal auszugeben. Ein gültiges Paar wäre „SAY Sie werden soeben mit der Nummer ...".
- **REPORT *<Ausdruck>*:** REPORT wird verwendet um Meldungen in ein Reportfile zu schreiben. Ein gültiges Paar wäre „REPORT Die Verbindung wurde erfolgreich erstellt.".

Ein typisches chat File kann folgendermaßen aussehen:

```
ABORT "NO CARRIER"
ABORT "BUSY"
ABORT "ERROR"
''     AT&F
OK     ATD06507407068
REPORT Nummer gewählt
CONNECT ''
```

Wird dieses File ausgeführt, so passiert folgendes: chat sendet AT&F an das Modem und wartet auf die Antwort „OK“. Sobald das Modem mit „OK“ antwortet, führt chat den Befehl „ATD06507407068“ aus, es weist das Modem mit dem Befehl ATD an, die Nummer 06507407068 zu wählen. Dann schreibt chat „Nummer gewählt“ in das Report File. Sobald das Modem mit „CONNECT“ antwortet ist das Skript erfolgreich beendet. Sollte das Modem während dieser Konversation irgendwann mit „NO CARRIER“, „BUSY“ oder „ERROR“ antworten, wird das Skript sofort beendet.

2.2.3 Das TC-35 Terminal

Das TC-35 Terminal von Siemens ist ein Standard GSM Modem mit einer seriellen Schnittstelle (RS232). Es ist für die Übertragung von Daten, Anrufen, SMS und Fax in GSM Netzwerken geeignet. Aufgrund der Standardschnittstellen und des integrierten SIM-Kartenlesers ist es einfach und schnell zu bedienen. Über die RS232 Schnittstelle kann das TC 35 Terminal an einen PC oder Server angeschlossen werden. Die Applikationen, die das TC 35 Terminal zur Kommunikation benutzen, können es über diese Schnittstelle mit AT Befehlen ansprechen und steuern. [vgl. TC35UG]

2.2.3.1 AT Befehle

Der AT Befehlssatz wurde von der Firma Hayes zur Steuerung von Modems mittels Terminalprogrammen entwickelt. Jeder Befehl beginnt mit „AT“ für „Attention“ und danach folgt der spezifische Befehl. Zu jedem AT Befehl gibt es eine oder mehrere mögliche Antworten von Seiten des Modems. Für die Entwicklung von CCMP waren vor allem folgende Befehle interessant, sie sind aus der TC 35 Terminal Spezifikation [TC35] entnommen (Die kursiv geschriebenen Ausdrücke bezeichnen einen variablen Wert):

- **AT&F:** Alle Parameter des Modems werden auf die Default Einstellungen des Herstellers zurückgesetzt. Die erwartete Antwort auf diesen Befehl ist „OK“
- **AT+IPR=?:** Gibt eine Liste der vom Modem unterstützten Baud Raten zurück.
- **AT+IPR=*<rate>*:** Setzt die Baudrate des Modems auf den mit *<rate>* angegeben Wert. Wurde die Baud Rate korrekt übernommen so lautet die Antwort des Modems „OK“**.**
- **AT+CSQ=?:** Dieser Befehl dient zur Überprüfung der Signalqualität. Das Modem gibt die Signalqualität und abschließend „OK“ zurück.
- **AT+CMGL=ALL:** Auf dieses Command gibt das Modem eine Liste der gespeicherten SMS zurück.
- **AT+CMGR=?**: Gibt an ob SMS am Modem gespeichert sind.
- **AT+CMGR=*<index>*:** Gibt die SMS mit dem angegebenen Index zurück.

- **AT+CMGD=*<index>*:** Löscht die SMS mit dem angegebenen Index.

Prinzipiell funktioniert die Kommunikation zwischen dem Terminal und dem Modem folgendermaßen: Das Terminal schickt einen Befehl, das Modem antwortet darauf. Dann schickt das Terminal den nächsten Befehl und wartet auf eine Antwort.

2.3 eSSENCE™

eSSENCE™ ist ein Java-Framework, das die Entwicklung von Webapplikationen auf Basis von Templates (z.B. HTML, XML aber auch CSV) unterstützt. Es wurde von Martin Gilly und Randolph Kepplinger entwickelt und basiert auf dem MAGNETIC-Konzept. [vgl. Gill00]

2.3.1 Die Komponenten

Das eSSENCE™ Framework besteht aus folgenden Komponenten: Templates, Magic Objects, Agents, eSSENCE™ Core-Klassen und statischen Files. Im folgenden Abschnitt werden die einzelnen Komponenten beschrieben.

2.3.1.1 Templates

Templates werden eingesetzt um das grafische Userinterface der eSSENCE™ Applikationen zu implementieren. Die Templates bestehen aus beliebigen SGML-basiertem Code (z.B. HTML) und einer eSSENCE™-eigenen Syntax zur Darstellung von Variablen und dynamischen Aufrufen, die beim Parsen durch konkrete Werte ersetzt wird. Templates können entweder die Repräsentation eines real-world Objektes sein oder Funktionalität kapseln, die keinem Objekt direkt zuzuordnen ist. Ein Template, das ein Objekt abbildet, kann folgende Informationen und Funktionalitäten enthalten: die Attribute und Relationen des Objektes, den Datenbank Table in dem das Objekt gespeichert wird, sämtliche Benutzerrechte auf das Objekt, verschiedene Darstellungen des Objekts, sogenannte VIEWs und Javascript Funktionen zur Manipulation des Objekts auf GUI Ebene.

Templates, die zu keinem Objekt gehören, werden als sogenannte „DIRECT Templates“ bezeichnet, weil sie nicht zur Definition von Daten haltenden Objekten (z.B. Person) dienen und damit keine real-world Objekte modellieren, sondern dynamische Komponenten zur Realisierung einer Applikation mit eSSENCE™ darstellen. Auch sie können verschiedene Attribute, VIEWs und Javascript Teile enthalten. „DIRECT Templates“ werden zum Beispiel zur dynamischen Anzeige von Listen oder zur Erzeugung von komplexen GUI-Elementen genutzt.

2.3.1.2 Magic Objects

Magic Objects sind Java-Klassen, die eSSENCE™ Objekten zugeordnet sind. Sie dienen dazu, die Business Logik einer Applikation abzubilden. Ein Magic Object kann mit einem Template verknüpft werden, indem der Name der Klasse im Template hinterlegt wird. Die Klasse stellt eine Reihe von Methoden zur Verfügung, welche genau dann aufgerufen werden, wenn das Template bzw. das zugehörige Objekt einen entsprechenden Zustand erreicht, zum Beispiel *preStore()*, *preCreate()* oder *parse()*. Die Methode *parse()* wird immer dann aufgerufen, wenn das Template geparst (d.h. für die Anzeige verarbeitet) wird, *preSto-*

re() und *preCreate()* bevor das Objekt, das durch das Template beschrieben wird, gespeichert oder angelegt wird. Die Methoden dienen sozusagen als Schnittstelle zur Businesslogik und erlauben eine vom Zustand des Objekts abhängige Ausführung des Java Codes.

2.3.1.3 Agents

Agents sind die einzelnen Teile des eSSENCE™-Frameworks; es sind RMI Komponenten, die als eigenständige Prozesse laufen. Es ist möglich, die einzelnen Agents auf unterschiedlichen Servern zu verteilen, das kann zum Beispiel Vorteile im Bereich der Performance haben. In eSSENCE™ hat jeder der Agent eine bestimmte Aufgabe:

- **Navigation Agent:** Der Navigation Agent agiert auf Ebene des Applikation Server und ist für den Empfang, die Aufbereitung und die Verteilung der HTTP-Requests an die übrigen Agenten zuständig.
- **Page Builder:** Der Page Builder ist für den Zusammenbau der Ergebnisseiten verantwortlich. Das dafür benötigte Basiscode (z.B. HTML Tags) aus dem Template und die Objektinformationen mit denen die fertige Ausgabe erzeugt wird, werden ihm von anderen Agents übergeben.
- **Data Agent:** Die Aufgabe des Data Agent ist die Verwaltung von Objekten. Er übernimmt das Anlegen, Ändern und Löschen von Objekten sowie die Durchführung von Abfragen gegen das Object-Repositiory.
- **FileData Agent:** Der FileData Agent übernimmt die Verwaltung der Files.
- **Index Agent:** Der Index Agent ist für die Indizierung der Objekte auf der Datenbank verantwortlich.
- **Thread Agent:** Der Thread Agent erlaubt Threads für die Verarbeitung von Batchprozessen zu starten und verwaltet diese dann unabhängig. . Er startet und beendet sie und stellt auf Anfrage Informationen über einzelne Threads zur Verfügung.

Um ihre Aufgaben zu erfüllen nützen die Agents die von den Core-Klassen zur Verfügung gestellten Kernfunktionalitäten des Systems.

2.3.1.4 eSSENCE™ Core-Klassen

Die eSSENCE™ Core-Klassen implementieren die Kernfunktionalitäten von eSSENCE™. Diese sind die datenbankunabhängige Verwaltung von Objekten und Files, der Zusammenbau der HTML oder XML-Ausgaben aus Templates und Objektinformationen, die Verarbeitung von HTTP-Requests und ihrer Parameter, die Verwaltung von Sessions, Caching, die Ausführung von Magic Klassen und das automatisierte Rendern von Templates.

2.3.1.5 Statische Files

Da für Webapplikationen auch statische Files wie Bilder, Files, etc. benötigt werden, ist es in eSSENCE™ auch möglich diese zu verwalten.

2.3.2 Architektur

Das Herzstück der eSSENCE™ Architektur sind die einzelnen RMI-Agents. In Abbildung 7 ist das Zusammenspiel der Agents bei der Verarbeitung eines HTTP-Requests zu sehen. Schickt der Anwender über seinen Webbrowser einen HTTP-Request ab, so wird dieser HTTP-Request über den Applikation-Server an den Navigation Agent gereicht. Er verarbeitet ihn und leitet die Anfrage entweder an den Page Builder oder an den File Data Agent weiter. Ist der File Data Agent der ausgewählte Agent, so übergibt im der Navigation Agent eine FOID (File OID). In diesem Fall handelt es sich um einen Dateidownload. Der File Data Agent holt das File aus der Datenbank und schickt es über den Applikation-Server direkt an den Browser des Anwenders. Im anderen Fall geht die Anfrage des Navigation Agents an den Page Builder.

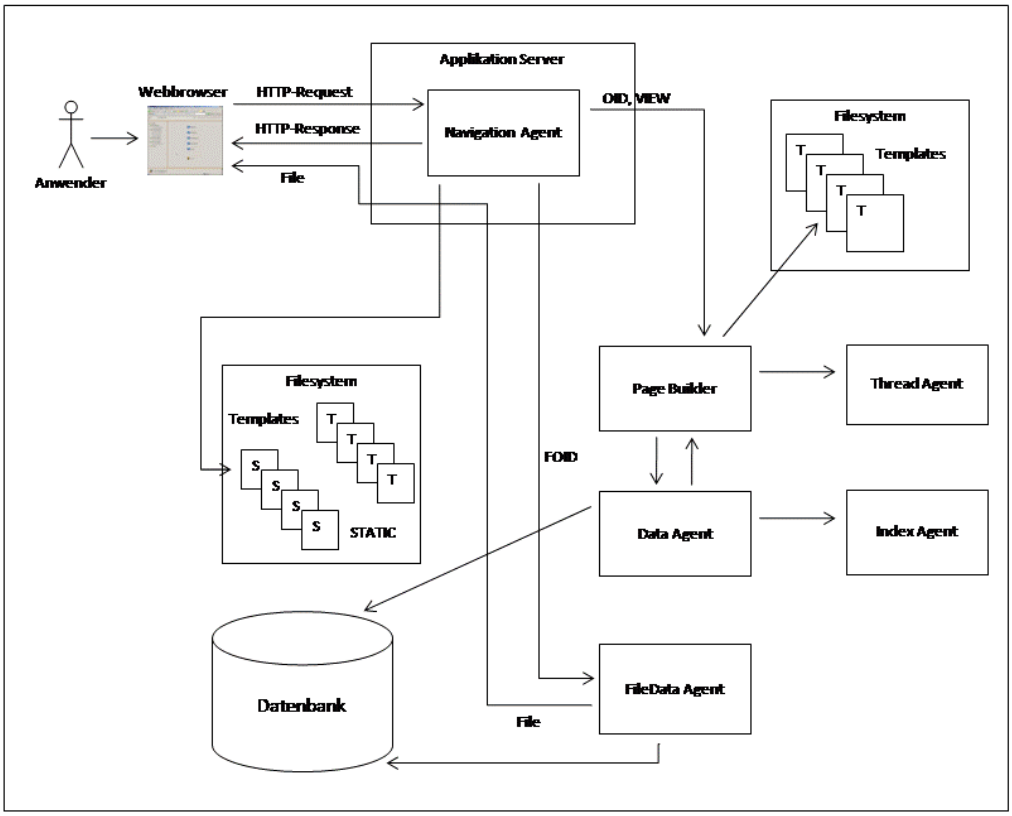


Abbildung 7: Verarbeitung eines HTTP-Requests in eSSENCE™

Der Page Builder bekommt eine OID (Object ID) und einen VIEW übergeben. Die OID leitet er an den Data Agent weiter und mit Hilfe des übergebenen VIEW, kann er das benötigte Stück HTML im Template identifizieren und herauslesen. Der Data Agent holt sich das Objekt mit der übergebenen OID aus der Datenbank und gibt es an den Page Builder zurück. Der fügt das HTML Stück und die Objektinformation zu einer HTML Seite zusammen und reicht sie an den Navigation Agent zurück. Falls vom Navigation Agent eine Anfrage kommt für die ein Thread benötigt wird, so leitet der Page Builder diese an den Thread Agent weiter. Wird ein Objekt angelegt, so wird der Index Agent kontaktiert. Er kümmert sich um die richtige Indizierung des Objektes auf der Datenbank.

2.3.3 Vorteile durch den Einsatz von eSSENCE™

eSSENCE™ wurde entwickelt um Webapplikationen effizienter und einfacher entwickeln zu können und bietet daher für den Entwickler zahlreiche Vorteile[META08]:

- **Plattformunabhängigkeit**: Da eSSENCE™ auf Java basiert, sind die mit eSSENCE™ entwickelten Applikationen plattformunabhängig, das heißt der Entwickler muss sich nicht darum kümmern, welches Betriebssystem auf dem Zielrechner, auf dem die Applikation installiert wird, läuft.
- **Datenbankunabhängigkeit**: Die mit eSSENCE™ entwickelten Webapplikationen sind datenbankunabhängig, das heißt es ist egal ob später eine MySQL, eine Oracle oder eine andere Datenbank unter der Applikation läuft.
- **Wiederverwendbarkeit von Code**: Die Architektur von eSSENCE™ unterstützt die Entwickler dabei, modular zu programmieren, was dazu führt, dass die einzelnen Komponenten stark in sich gekapselt sind und so einfach wiederverwendet werden können.
- **Standardmodule**: eSSENCE™ stellt eine Reihe von Modulen zur Verfügung, die Standardfunktionalitäten wie die Verwaltung von Objekten, die Verwaltung von Files oder ein umfangreiches Usermanagement kapseln.
- **Automatisiertes Template-Rendering**: eSSENCE™ stellt eine Komponente für das automatisierte Rendern von HTML-Templates zur Verfügung. Das ermöglicht eine rasche und effiziente Entwicklung der Applikation. Die Standard HTML Views werden auf Basis von Konfigurationsfiles gerendert und der Entwickler muss sich nur um die Erstellung der Konfigurationsfiles bzw. um spezielle Views kümmern.

- **Einsatz von Standardtechnologien**: eSSENCE™ basiert auf Standardtechnologien wie Java, HTML und javascript. Dadurch ist es möglich, externe Tools und Bibliotheken schnell und einfach in eSSENCE™ einzubinden.

3 Anforderungsanalyse

Der Analyseprozess dient der Erarbeitung und zur Beschreibung der gewünschten Anforderungen an das System. Diese Anforderungen werden in einem Dokument, dem Analysedokument festgehalten. Die Analyse selbst lässt sich in zwei Bereiche gliedern: erstens die Anforderungsanalyse, deren Aufgabe die Beschreibung aller funktionalen und nicht funktionalen Anforderungen ist und zweitens das Analysemodell, das dazu dient die Anforderungsanalyse in Bezug auf Machbarkeit und Vollständigkeit zu überprüfen.

In der Anforderungsanalyse geht es darum, die individuellen Anforderungen an das System zu finden. Das geschieht meist in enger Zusammenarbeit mit dem Kunden, dabei sind oft mehrere Iterationen (Anforderungen dokumentieren – Fragen klären – Dokumentation anpassen) nötig um alle Anforderungen an das zu erstellende System zu definieren. Die Anforderungsanalyse wird in der Sprache des Kunden geschrieben, d.h. es wird nur die Funktionalität des Systems dargestellt, nicht aber ihre Umsetzung. Die Anforderungen werden durch Anwendungsfälle strukturiert, zwischen den Anforderungen darf es zu Redundanzen und Inkonsistenzen kommen. Damit kann die Anforderungsanalyse als gemeinsamer Bezugspunkt zwischen dem Kunden und dem Entwickler gesehen werden. Die Schwierigkeit in dieser Phase liegt im hohen Kommunikationsbedarf, da bereits hier möglichst alle Unklarheiten zwischen dem Kunden und dem Entwickler beseitigt werden müssen, damit sie nicht in den weiteren Projektverlauf miteinfließen. Das ist vor allem anfangs sehr schwierig, da die unterschiedlichen Denkmuster zwischen den beiden Parteien ein Gespräch über einen gemeinsamen Inhalt kompliziert machen können. Es muss erst eine gemeinsame Sprache und vor allem eine gemeinsame Sicht auf das Projekt erarbeitet werden. In dieser Phase des Projektes ist es Ziel, das Anforderungsdokument zu erstellen, dessen Qualität entscheidend für den weiteren Projekterfolg ist, da die gesamte Entwicklung darauf aufbaut. Das Anforderungsdokument sollte aus einer Systembeschreibung, einem Begriffsverzeichnis, einer Aktorenliste, Anwendungsfalldiagrammen, Anwendungsfallbeschreibungen, einem Analyseprototyp und einem Domänenmodell bestehen.

Die Erstellung des Analysemodells ist dann der zweite Schritt der Analysephase. Es geht darum, die in der Anforderungsanalyse erarbeiteten Anforderungen auf eine mögliche Umsetzung und auf ihre Vollständigkeit zu überprüfen. Das Modell zeigt eine interne Sicht auf das System und verwendet die Sprache der Entwickler. Es zeigt den Entwicklern wie das System aufgebaut sein soll und stellt die Umsetzung der Funktionalität dar. Redundanzen und Inkonsistenzen die in der Anforderungsanalyse noch auftreten können, werden beseitigt ,damit eine Struktur aus stereotypen Klassen und Modulen erarbeitet werden kann. Das Ana-

lysemodell dient damit dazu, den Entwicklern zu zeigen wie der Aufbau des Systems sein soll und stellt so einen ersten Schritt in Richtung Entwurf dar. [vgl. Zuse01, Seite 81,116]

Insgesamt ist die Analyse ist eine sehr kritische Phase in einem Projekt, da von ihrer erfolgreichen Umsetzung der gesamte Projekterfolg abhängen kann. Denk- und Konzeptionsfehler, die in dieser Phase begangen werden, werden meist in spätere Projektphasen mithineingetragen, sie sind schwer zu finden und ihre Behebung kann hohe Kosten erzeugen, wenn sie das Grundkonzept der zu erstellenden Software betreffen.

Das folgende Kapitel beinhaltet die Anforderungsanalyse für das CCMP Projekt, die, wie von Zuser [vgl. Zuse01, Seite 81,98] beschrieben, durchgeführt wurde. Sie ist das Ergebnis von mehreren Sitzungen mit den Partnern und Kunden und bietet eine in sich konsistente und klar strukturierte Grundlage für die Entwicklung der Applikation. Die nachfolgenden Punkte umfassen eine Beschreibung des Systems, ein Begriffsverzeichnis, eine Liste der Aktoren, eine Liste der Anwendungsfälle sowie ein Domainmodell. Auf einen Analyseprototyp wurde im Einverständnis mit dem Kunden verzichtet.

3.1 Systembeschreibung

In diesem Abschnitt wird ein kurzer Überblick über die Situation vor dem Projekt, die daraus entstehenden Anforderungen, die Schwachstellen des existierenden Systems, die Abgrenzungen des zu entwickelnden Systems sowie über die mögliche Übernahme existierender Datenbestände gegeben. Diese Beschreibung dient als Grundlage zur Erfassung des Szenarios und der Problemstellung.

3.1.1 Ausgangsituation

Das Communication Center (CC) ist ein Produkt der Firma LEP Lehotzki GmbH. Es kann in den verschiedensten Szenarien zur Fernüberwachung, zur Messung von Umweltparametern und zur Steuerung von Industrieanlagen eingesetzt werden. Dabei können ein oder mehrere Geräte pro Szenario eingesetzt werden. Die Communication Center sind räumlich oft weit verteilt und schwer zugänglich und werden deshalb von einer zentralen Stelle aus verwaltet. Wenn möglich wird nur die erstmalige Inbetriebnahme von einem Techniker vor Ort durchgeführt. Zur Verwaltung der Geräte wird ein PC mit einem Modem benötigt, das dazu verwendet wird sich über eine DFÜ Verbindung in die Geräte einzuwählen.

Über diese Verbindung kann der Benutzer das CC jederzeit umkonfigurieren und auf seine momentanen Bedürfnisse einstellen. Dazu muss er sich in jedes Gerät einzeln einwählen und dort die gewünschten Änderungen durchführen. Diese Art der Geräteverwaltung hat sich vor allem in Szenarien mit mehreren Geräten als schwerfällig und zeitintensiv herausgestellt.

In Szenarien mit nur einem oder sehr wenigen Geräten ist die Anschaffung des Modems ein großer Kostenfaktor. Deswegen hat die Firma LEP Lehotzki GmbH sich dazu entschlossen eine Webapplikation zur Verwaltung der Geräte entwickeln zu lassen, die zusätzlich noch Funktionalität zum Loggen der gespeicherten Daten, sowie zum Archivieren der versendeten SMS anbieten soll. Die Applikation soll auf der einen Seite als komplette Serverlösung für Szenarios mit mehren bis vielen Communication Centern angeboten werden, d.h. eine Firma kauft sich einen Server mit Modem auf dem die Webapplikation mit eigener Datenbank installiert wird und benutzt diesen Server ausschließlich für die Verwaltung ihrer CC´s. Auf der anderen Seite soll es auch eine Lösung für Benutzer mit einem oder wenigen Geräten geben. Für diese Lösung ist ein Server geplant, der für mehrere Benutzer zugänglich ist, die eine monatliche Gebühr für die Benutzung der Webapplikation entrichten.

3.1.2 Anforderungen an die Applikation

Die Applikation soll dem Anwender, die Verwaltung der Geräte durch eine übersichtliche und leicht zu bedienende Benutzeroberfläche wesentlich erleichtern. Durch die Automatisierung der Kommunikation und die dadurch reduzierten Verbindungszeiten pro Gerät sollen die Kommunikationskosten stark gesenkt werden. In großen Szenarien mit gleich oder ähnlich konfigurierten Geräten soll der benötigte Administrationsaufwand durch eine Funktion zum Kopieren der Geräte verringert werden, was sich in geringeren Arbeitskosten zeigen soll. Es soll ein SMS-Archiv geben, das es möglich macht auch nach langer Zeit noch festzustellen ob an gewissen Tagen oder zu gewissen Ereignissen ein oder mehrere SMS geschickt wurden. Daran kann man überprüfen ob sich ein Gerät zu diesen Zeitpunkten korrekt verhalten hat oder nicht. Die Applikation soll eine Funktionalität zum Loggen der auf den Geräten gespeicherten Datensätze bieten, die auch automatisch in bestimmten Intervallen funktionieren soll. Die geloggten Daten sollen in einem Excel exportierbar sein, was eine Auswertung und Analyse der Logdaten über lange Zeiträume möglich macht. Diese Auswertungsmöglichkeit schafft die Voraussetzungen, Schwankungen in den Messdaten schon frühzeitig zu erkennen und so etwaigen Unregelmäßigkeiten oder Vorfällen frühzeitig entgegenzuwirken.

Ziel ist, dass die Geräte immer im selben Zustand sind wie im System, das heißt die Geräte im System und im Einsatz sollen wenn möglich immer synchron sein.

3.1.3 Schwachstellen des momentanen Systems

Im Moment sind die Communication Center nur einzeln konfigurierbar und die Einstellungen eines Gerätes sind nicht auf andere Geräte übertragbar, das heißt, wenn in einem Szenario mit mehreren oder vielen Geräten alle die gleiche Konfiguration bekommen sollen, muss jedes der Geräte einzeln konfiguriert werden. Das ist sehr zeit- und somit kostenintensiv.

Außerdem kann es durch Eingabefehler leichter zu Unterschieden zwischen den Gerätekonfigurationen kommen.

Der hohe Preis für ein Modem wird im jetzigen System vor allem Anwender, die nur ein oder wenige Geräte haben, davon abhalten diese über DFÜ Verbindung zu verwalten und somit verlieren diese Geräte den großen Vorteil der Fernwartbarkeit. Die Benutzer, die sich eine Modem anschaffen, müssen über erweiterte Computerkenntnisse verfügen um das Modem zu konfigurieren und sich eine DFÜ Verbindung einrichten zu können.

Im jetzigen System ist kein Daten Logging vorgesehen. Dadurch ist es sehr schwierig Messdaten über längere Zeiträume zu beobachten, auszuwerten und zu vergleichen. SMS können kaum archiviert werden, was es schwierig macht, festzustellen ob Geräte an gewissen Zeitpunkten oder Tagen in der Vergangenheit funktioniert haben.

3.1.4 Abgrenzung des Systems

Die Applikation soll eine 1:1 Abbildung des oder der verwalteten Communication Center zur Verfügung stellen. Das System soll als Webapplikation von überall zugänglich sein und neben der Verwaltungsmöglichkeit für die Geräte, ein SMS Archiv und eine Funktion zum Datenloggen mit Excel Export anbieten. Die Gerätekonfigurationen sollen als PDF exportiert werden können. Ansonsten sind keine Anbindungen an andere Applikationen vorgesehen.

3.1.5 Datenbestände und Übernahme bestehender Daten

Es existieren im Moment keine Datenbestände. Für Geräte, die sich bereits im Einsatz befinden ist eine Funktionalität vorgesehen mit der die bestehende Konfiguration vom Gerät in die Webapplikation eingespielt werden kann. Die Archivierung von SMS und Logdaten aus dem Zeitraum vor Einführung des Systems ist nicht geplant.

3.2 Begriffsverzeichnis

Das Begriffsverzeichnis dient dazu die Fachsprache des Anwenders und der Anwendungsdomäne zu verstehen und zu dokumentieren. Es ist eine Definition von Begriffen und Abläufen, die im Kontext der Anwendungsdomäne eine spezielle Bedeutung oder nur dort eine Bedeutung haben. Mit Hilfe des Begriffsverzeichnisses sollen Missverständnisse ausgeräumt werden. [vgl. Zuse01, Seite 85]

Communication Center	**Ein Fernüberwachungs- und Messgerät, das von der Firma LEP Lehotzki GmbH hergestellt wird.**
CC	**Abkürzung für Communication Center**
Daten Logging	**Prozess, der das Auslesen der geloggten Daten aus dem Communication Center durchführt.**
Empfänger	**Eine Person, deren Email und/oder Telefonnummer im Gerät eingetragen ist. Diese Person bekommt Meldungen per Email oder SMS vom Communication Center geschickt.**
Fernparametrisierung	**Das Ändern von Konfigurationsdaten auf einem Communication Center über eine DFÜ Verbindung.**
Initialisierungsprozess	**Der Prozess, der das Auslesen von Daten aus einem Communication Center durchführt und die Daten im System abspeichert.**
Meldung	**Eine Nachricht, die in Form einer SMS oder Email bei Eintritt eines bestimmten Ereignisses vom Communication Center an vorher festgelegte Personen verschickt wird**
I/O	**Ein Analoger/Digitaler Eingang oder ein Digitaler Ausgang am Communication Center. An die Eingänge können Sensoren und Messgeräte angeschlossen werden, über die Ausgänge können andere Geräte gesteuert werden.**
Synchronisationsprozess	**Der Prozess, der das Ändern von Daten eines Communication Centers und das Einspielen dieser Daten in das Gerät umfasst.**
Statusmeldung	**Eine Nachricht, die in Form einer Email an bestimmten Wochentagen vom Communication Center an vorher festgelegte Personen verschickt wird und den Status der I/Os übermittelt.**
Triggerbedingung	**Regel, die festlegt aufgrund des Eintretens welcher Ereignisses am Gerät eine Nachricht an einen Empfänger geschickt werden soll.**
triggern	**Das Auslösen einer Meldung am Gerät aufgrund eines bestimmten Ereignisses**
Verzögerungszeit Ein	**Die Zeitspanne die eine Bedingung gültig sein muss, bevor eine Meldung geschickt wird**
Verzögerungszeit Aus	**Die Zeitspanne die eine Bedingung nicht mehr gültig sein muss, damit das Senden von Meldungen beendet wird.**

Tabelle 2: Begriffsverzeichnis

3.3 Aktorenliste

Die nachfolgende Abbildung zeigt eine Auflistung aller Aktoren im System, ihre Darstellung in den nachfolgenden Diagrammen, ihre Rechte im System, eventuelle Anmerkungen und die hierarchischen Beziehungen in denen sie zueinander stehen [vgl. Zuse01, Seite 85,86]:

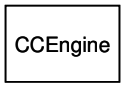
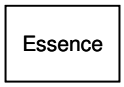
Aktorenname	Symbol	Rechte	Anmerkungen	Hierarchie
Anonym		Besuchen der Informationsseite über CCMP, Registrieren als CCMPuser		
CCMPuser		Anlegen/Ändern/Löschen von Gruppen, Empfängern und Geräten. Suchen von archivierten SMS. Suchen und Auswerten von Logdaten	Nur selbst angelegte Objekte können geändert und gelöscht werden.	
Admin		Anlegen/Ändern/Löschen von CCMPusern.	Ändern und Löschen aller Objekte im System.	
CCEngine		Ändern von Gerätedaten und Anlegen von Empfängern. Kommunikation mit den Geräten. Auslesen und Archivieren von Logdaten aus den Geräten. SMS Archivierung.		
Essence		Anlegen von Benutzern bei der automatischen Registrierung.		

Abbildung 8: Eine Liste der Aktoren im CCMP-System

3.4 Anwendungsfälle

Dieser Abschnitt enthält eine ausführliche Beschreibung der Anwendungsfälle, die in den Sitzungen mit den Partnern erarbeitet wurden sowie die grafische Darstellung dieser Anwendungsfälle.

3.4.1 Allgemein

„Ein Anwendungsfall ist eine abgeschlossene, zusammenhängende Einheit, welche einen Teil der Funktionalität des Systems repräsentiert. Ein Anwendungsfall sollte eine logische zusammengehörige, wiederkehrende Anwendung innerhalb des Systems darstellen.“ [Zuse01, 86]

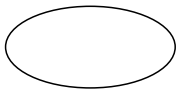
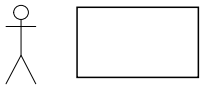
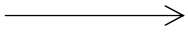
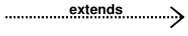
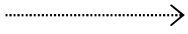
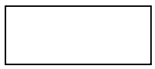
Element	Beschreibung
	**Anwendungsfall**: Ist eine abgeschlossene, zusammenhängende Einheit, welche einen Teil der Funktionalität des Systems repräsentiert. Wenn der Rand des Anwendungsfalles unterbrochen ist, so besteht er aus weiteren.
	**Aktor**: Ist ein bestimmter Benutzer des Systems, der genau definierte Rechte und Aufgaben innerhalb des Systems hat. Ist der Aktor ein anderes System so wird er als Rechteck dargestellt.
	Kommunikationsbeziehung: Eine Kommunikationsbeziehung ist die einzig mögliche Beziehung zwischen einem Aktor und einem Anwendungsfall. Es existiert genau dann eine Beziehung zwischen einem Anwendungsfall und einem Aktor, wenn der Aktor dazu ermächtigt ist den Anwendungsfall auszulösen.
	**Generalisierungsbeziehung**: Diese Beziehung zwischen 2 Aktoren oder 2 Anwendungsfällen bedeutet, dass das Element am Pfeilanfang eine spezielle Form des Elements an der Pfeilspitze ist. Das erste Element kann alles, was auch das Element an der Pfeilspitze kann, hat aber noch zusätzliche Fähigkeiten.
	**Extend-Beziehung**: Diese Beziehung zwischen 2 Anwendungsfällen bedeutet, dass der Anwendungsfall auf den die Pfeilspitze zeigt, von dem Anwendungsfall, welcher Ausgangspunkt des Pfeils ist, erweitert werden kann.
	**Include-Beziehung**: Diese Beziehung zwischen 2 Anwendungsfällen bedeutet, dass der Anwendungsfall, auf den die Pfeilspitze zeigt, von dem Anwendungsfall, welcher Ausgangspunkt des Pfeils ist, benutzt wird.
	**System**: Das System bietet das Verhalten, welches durch einen Anwendungsfälle beschrieben wird, an und realisiert es. Ein System kann aus mehreren Subsystemen bestehen.

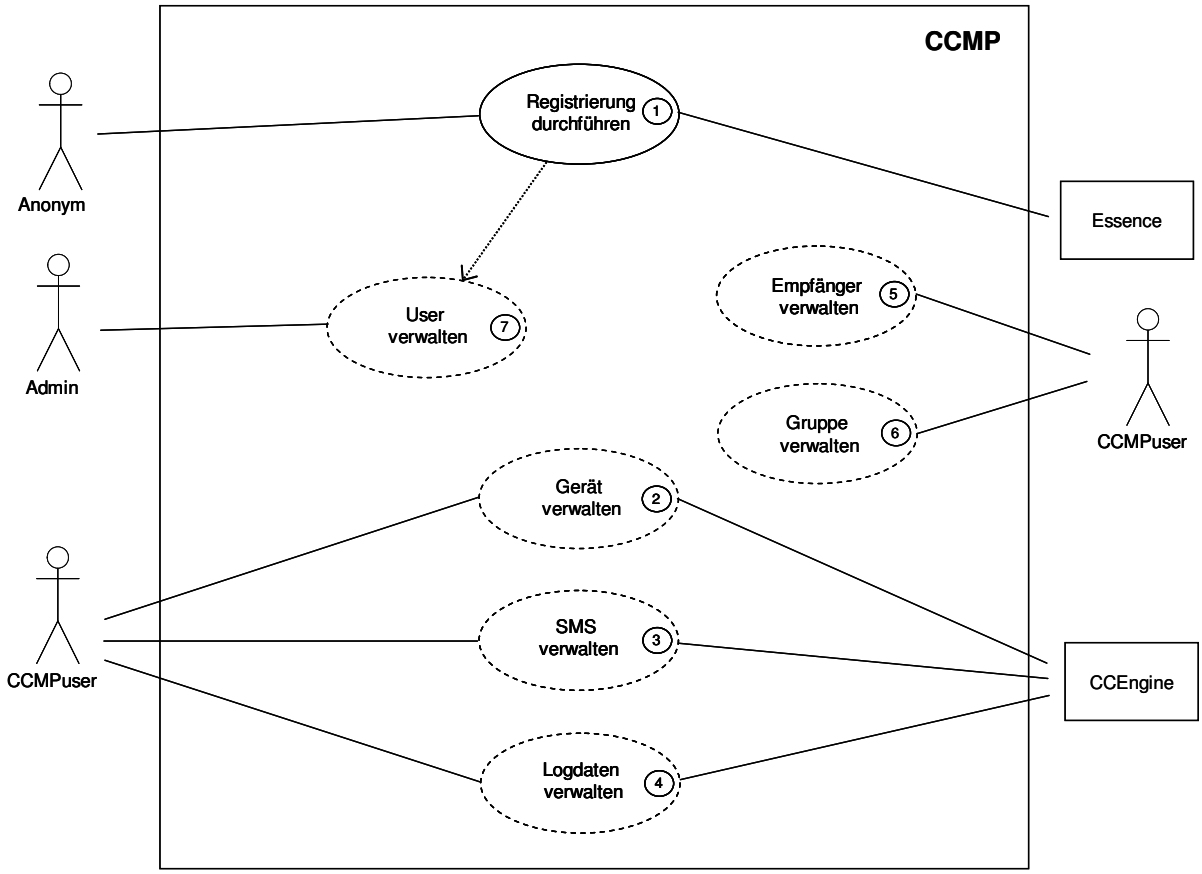
Abbildung 9: Übersicht der Elemente zur Erstellung eines Anwendungsfalles nach Zuser

Ein Anwendungsfalldiagramm zeigt das externe Verhalten des Systems aus der Sicht des Benutzers auf, indem es die Aktoren, die Anwendungsfälle und deren Beziehungen zueinander darstellt. Die Darstellung der Anwendungsfälle folgt im wesentlichen der UML Notation (Unified Modeling Language Notation).

Das folgende Diagramm erklärt die einzelnen Elemente eines Anwendungsfalldiagrammes.

Jedes Anwendungsfalldiagramm ist durch eine Anwendungsfallbeschreibung erklärt. Eine Anwendungsfallbeschreibung besteht aus einer eindeutigen Nummer, einem Titel, einer Kurzbeschreibung, die den Anwendungsfall skizziert, einer Auflistung aller notwendigen Vorbedingungen, einer Detailbeschreibung des Ablaufes, einer Auflistung der Auswirkungen des Anwendungsfalles und eventuellen Anmerkungen. Die Detailbeschreibung ist so aufgebaut, dass es immer ein Ereignis (E1) gibt auf das eine Antwort (A1) folgt. Alternative Ereignisse und deren Antwort wird ein A vorangestellt (AE1) und (AA1). Jedes Ereignis und jede Antwort hat eine eindeutige Nummer innerhalb des Anwendungsfalles. [vgl. Zuse01, Seite 86,90]

3.4.2 Übersicht



Anwendungsfall 1: Anwendungsfalldiagramm „Übersicht“

Nr.: 1
Titel: Registrierung durchführen
Kurzbeschreibung: Ein anonymer User gibt seine Stammdaten ein und registriert sich als CCMPuser.
Vorbedingungen: Keine
Beschreibung des Ablaufs: E1) Der Anwender gibt die URL der CCMP Seite in seinen Browser ein. A1) Das System liefert die Startseite von CCMP mit dem Link zur Registrierung. E2) Der Anwender klickt auf den Registrierungslink.

A2) Das System zeigt das Registrierungsformular an. E3) Der Anwender gibt alle erforderlichen Daten (Vorname, Nachname, Email Adresse, Adresse,…) ein und schickt das Registrierungsformular ab. A3) Das System überprüft die eingegebenen Daten. E4) Die eingegebenen Daten sind vollständig und korrekt. Der User ist noch nicht im System vorhanden. A4) Das System führt den Anwendungsfall 7.1(„User anlegen“) aus und verschickt eine Email Bestätigung mit dem Usernamen und dem Passwort an den Anwender. AE4) Die eingegebenen Daten sind nicht vollständig und/oder nicht korrekt bzw. der User ist schon im System vorhanden. AA4) Eine entsprechende Fehlermeldung wird ausgegeben und der Anwender hat die Möglichkeit, den Fehler zu beheben.
Auswirkungen: Ein CCMPuser wird angelegt
Anmerkungen: Keine

Nr.: 2
Titel: Gerät verwalten
Kurzbeschreibung: Dieser Anwendungsfall besteht aus weiteren Anwendungsfällen.

Nr.: 3
Titel: SMS verwalten
Kurzbeschreibung: Dieser Anwendungsfall besteht aus weiteren Anwendungsfällen

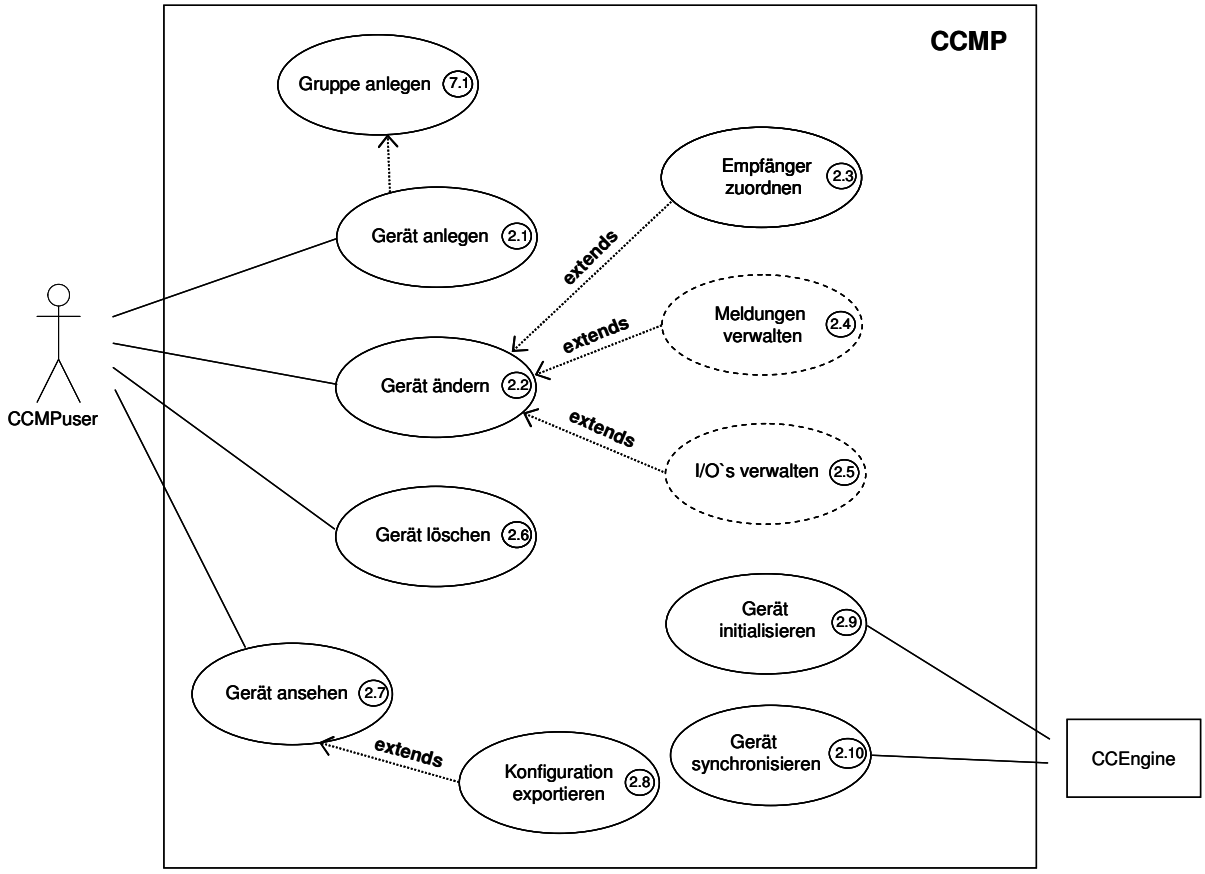
Nr.: 4
Titel: Logdaten verwalten
Kurzbeschreibung: Dieser Anwendungsfall besteht aus weiteren Anwendungsfällen

Nr.: 5
Titel: User verwalten
Kurzbeschreibung: Dieser Anwendungsfall besteht aus weiteren Anwendungsfällen

Nr.: 6
Titel: Empfänger verwalten
Kurzbeschreibung: Dieser Anwendungsfall besteht aus weiteren Anwendungsfällen

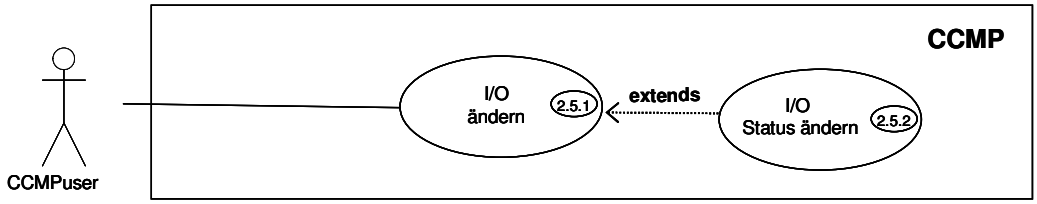
Nr.: 7
Titel: Gruppe verwalten
Kurzbeschreibung: Dieser Anwendungsfall besteht aus weiteren Anwendungsfällen

3.4.3 Gerät verwalten



Anwendungsfall 2: Anwendungsfalldiagramm „Gerät verwalten“

3.4.4 I/O verwalten



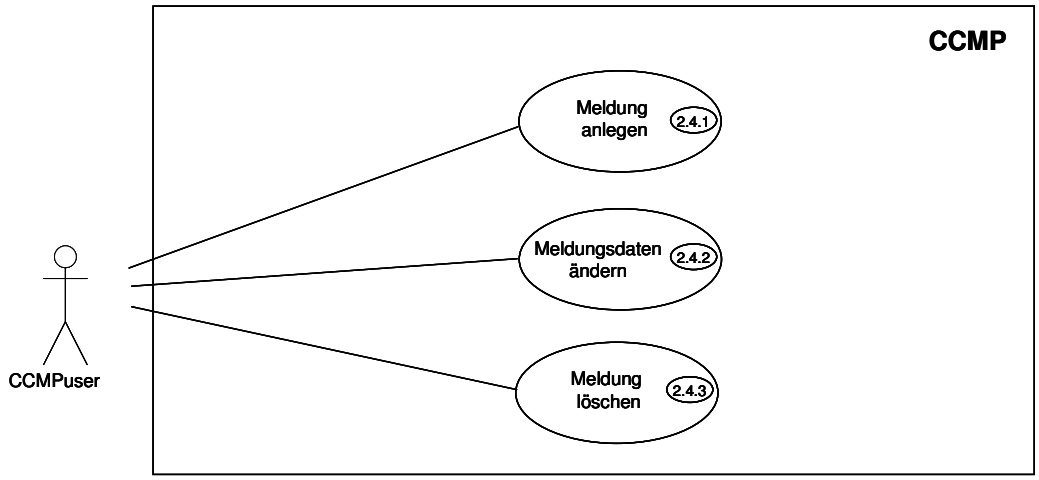
Anwendungsfall 3: Anwendungsfalldiagramm „I/O verwalten“

Nr.: 2.5.2
Titel: I/O ändern
Kurzbeschreibung: Der Anwender ändert die Daten eines I/O
Vorbedingungen: Der Anwender ist mit der Berechtigung CCMPuser eingeloggt und befindet sich bereits in einem bestimmten Gerät
Beschreibung des Ablaufs: E1) Der Anwender wählt den I/O aus, den er ändern will. A1) Das System zeigt ein Formular zur Änderung der I/O Daten an. E2) Der Anwender ändert die Daten. A2) Das System überprüft, ob alle notwenigen Daten eingeben wurden und die Daten korrekt sind. E3) Alle notwendigen Daten wurden eingegeben und die Daten sind korrekt. A3) Das System speichert die geänderten Daten ab.

AE3) Nicht alle notwendigen Daten wurden eingegeben oder die Daten sind nicht korrekt. AA3) Eine entsprechende Meldung wird ausgegeben.
Auswirkungen: Die Daten eines I/O wurden geändert und gespeichert. Das zugehörige Gerät hat den Status „asynchron“ bekommen.
Anmerkungen: Das Gerät sollte nachdem die Daten eines I/O´s geändert wurden, unbedingt synchronisiert werden.

Nr.: 2.5.2
Titel: I/O Status ändern
Kurzbeschreibung: Der Anwender ändert den Status eines I/O
Vorbedingungen: Der Anwender ist mit der Berechtigung CCMPuser eingeloggt und befindet sich bereits in einem bestimmten Gerät
Beschreibung des Ablaufs: E1) Der I/O hat den Status „aktiv“ und der Anwender will den Status ändern. A1) Das System fragt den Anwender, ob er den Status wirklich ändern will. AE1) Der I/O hat den Status „inaktiv“ und der Anwender will den Status ändern. AA1) Das System fragt den Anwender, ob er den Status wirklich ändern will. E2) Der Anwender entscheidet sich die Änderung durchzuführen. A2) Das System ändert den Status des „I/O“. AE2) Der Anwender entscheidet sich die Änderung nicht durchzuführen. AA2) Das System ändert den Status des „I/O“ nicht. E3) Der Status des I/O wurde geändert. A3) Das System setzt den Status des Gerätes auf „asynchron“.
Auswirkungen: Der Status eines I/O wurden geändert und gespeichert. Das zugehörige Gerät hat den Status „asynchron“ bekommen.
Anmerkungen: Das Gerät sollte nachdem der Status eines I/O´s geändert wurde, unbedingt synchronisiert werden.

3.4.5 Meldung verwalten



Anwendungsfall 4: Anwendungsfalldiagramm "Meldung verwalten"

Nr.: 2.6.1
Titel: Meldung anlegen
Kurzbeschreibung: Der Anwender legt eine Meldung an

Vorbedingungen: Der Anwender ist mit der Berechtigung CCMPuser eingeloggt und befindet sich bereits in einem bestimmten Gerät

Beschreibung des Ablaufs:

E1) Der Anwender wählt die Option „Meldung anlegen“ aus.
A1) Das System zeigt ein Formular zur Eingabe der Meldungsdaten an.
E2) Der Anwender gibt die Meldungsdaten ein.
A2) Das System überprüft, ob alle notwenigen Daten eingeben wurden und die Daten korrekt sind.
E3) Alle notwendigen Daten wurden eingegeben und die Daten sind korrekt.
A3) Das System legt eine neue Meldung an.
AE3) Nicht alle notwendigen Daten wurden eingegeben oder die Daten sind nicht korrekt.
AA3) Eine entsprechende Meldung wird ausgegeben.
E4) Die Meldung wurde angelegt.
A4) Das System setzt den Status des Gerätes auf „asynchron“.

Auswirkungen: Eine Meldung wurde angelegt. Das zugehörige Gerät hat den Status „asynchron“ bekommen.

Anmerkungen: Das Gerät sollte nach dem Anlegen einer Meldung, unbedingt synchronisiert werden.

Nr.: 2.6.2

Titel: Meldung ändern

Kurzbeschreibung: Der Anwender ändert die Daten einer Meldung

Vorbedingungen: Der Anwender ist mit der Berechtigung CCMPuser eingeloggt und befindet sich bereits in einem bestimmten Gerät

Beschreibung des Ablaufs:

E1) Der Anwender wählt die Meldung aus, die er ändern will.
A1) Das System zeigt ein Formular zur Änderung der Meldungsdaten an.
E2) Der Anwender ändert die Daten.
A2) Das System überprüft, ob alle notwenigen Daten eingeben wurden und die Daten korrekt sind.
E3) Alle notwendigen Daten wurden eingegeben und die Daten sind korrekt.
A3) Das System speichert die geänderten Daten ab.
AE3) Nicht alle notwendigen Daten wurden eingegeben oder die Daten sind nicht korrekt.
AA3) Eine entsprechende Meldung wird ausgegeben.
E4) Die Meldung wurde geändert.
A4) Das System setzt den Status des Gerätes auf „asynchron“.

Auswirkungen: Die Daten einer Meldung wurden geändert und gespeichert. Das zugehörige Gerät hat den Status „asynchron“ bekommen.

Anmerkungen: Das Gerät sollte nach dem Ändern einer Meldung, unbedingt synchronisiert werden.

Nr.: 2.6.3

Titel: Meldung löschen

Kurzbeschreibung: Der Anwender löscht eine Meldung

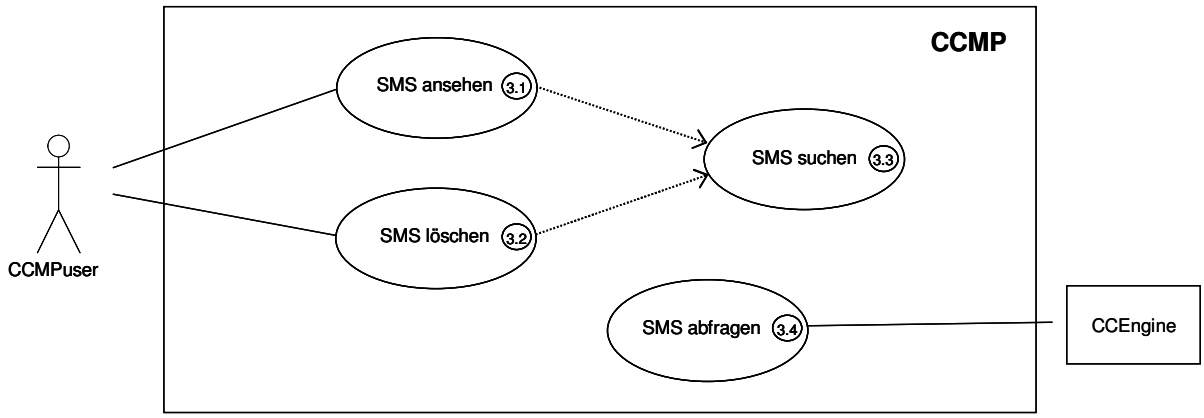
Vorbedingungen: Der Anwender ist mit der Berechtigung CCMPuser eingeloggt und befindet sich bereits in einem bestimmten Gerät

Beschreibung des Ablaufs:

E1) Der Anwender wählt die Meldung, die er löschen will aus.
A1) Das System fragt den Anwender, ob er die Meldung wirklich löschen will.
E2) Der Anwender entscheidet sich dafür die Meldung zu löschen.

A2) Das System löscht die Meldung. AE2) Der Anwender entscheidet sich dafür, die Meldung nicht zu löschen. AA2) Das System löscht die Meldung nicht. E3) Die Meldung wurde gelöscht. AA3) Das System setzt den Status des Gerätes auf „asynchron
Auswirkungen: Eine Meldung wurde aus dem System gelöscht. Das zugehörige Gerät hat den Status „asynchron“ bekommen.
Anmerkungen: Das Gerät sollte nach dem Löschen einer Meldung, unbedingt synchronisiert werden.

3.4.6 SMS verwalten



Anwendungsfall 5: Anwendungsfalldiagramm "SMS verwalten"

Nr.:3.1
Titel: SMS ansehen
Kurzbeschreibung: Der Anwender sieht sich eine SMS an.
Vorbedingungen: Der Anwender ist mit der Berechtigung CCMPuser oder höher eingeloggt
Beschreibung des Ablaufs: E1) Der Anwender wählt die Liste mit allen SMS. A1) Das System zeigt eine Liste mit allen SMS an. AE1) Der Anwender entscheidet sich, eine SMS zu suchen. AA1) Das System führt den Anwendungsfall 3.3 („SMS suchen“) aus. E2) Der Anwender wählt die SMS, die er ansehen will aus. A2) Das System zeigt die SMS mit ihren Attributen an.
Auswirkungen: keine
Anmerkungen: keine

Nr.:3.2
Titel: SMS löschen
Kurzbeschreibung: Der Anwender löscht eine SMS aus dem System.
Vorbedingungen: Der Anwender ist mit der Berechtigung CCMPuser oder höher eingeloggt
Beschreibung des Ablaufs: E1) Der Anwender wählt die Liste mit allen SMS.

A1) Das System zeigt eine Liste mit allen SMS an.
AE1) Der Anwender entscheidet sich eine SMS zu suchen.
AA1) Das System führt den Anwendungsfall 3.3 („SMS suchen“) aus.
E2) Der Anwender wählt die SMS, die er löschen will aus.
A2) Das System fragt den Anwender, ob er die SMS wirklich löschen will.
E3) Der Anwender entscheidet sich dafür die SMS zu löschen.
A3) Das System löscht die SMS.
AE3) Der Anwender entscheidet sich dafür, die SMS nicht zu löschen.
AA3) Das System löscht die SMS nicht.

Auswirkungen: Eine SMS wurde gelöscht

Anmerkungen: keine

Nr.:3.3

Titel: SMS suchen

Kurzbeschreibung:

Vorbedingungen: Der Anwender ist mit der Berechtigung CCMPuser oder höher eingeloggt

Beschreibung des Ablaufs:
E1) Der Anwender wählt ein Gerät aus, für welches er nach SMS suchen will.
A1) Das System wartet auf weitere Eingaben.
E2) Der Anwender wählt einen Zeitraum aus, über den er nach SMS suchen will.
A2) Das System sucht nach SMS, die den eingegebenen Suchkriterien entsprechen.
E3) Es existieren SMS, die den Suchkriterien entsprechen.
A3) Das System zeigt die gefundenen SMS in einer Liste an.
AE3) Es existieren keine SMS, die den Suchkriterien entsprechen.
AA3) Es wird eine entsprechende Meldung angezeigt.

Auswirkungen: Eine SMS wurde gelöscht

Anmerkungen: keine

Nr.:3.4

Titel: SMS abfragen

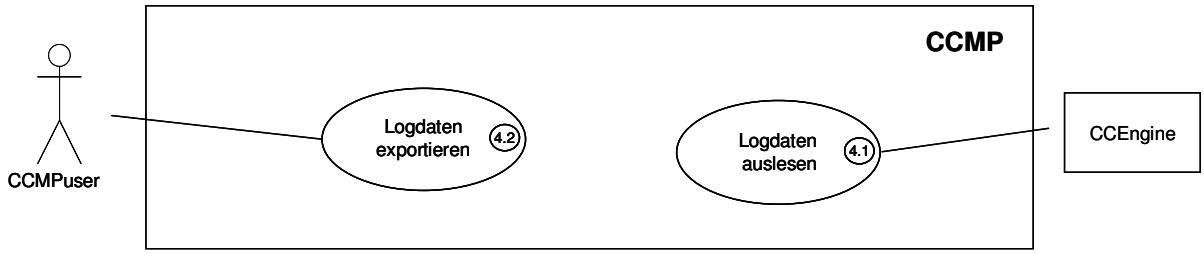
Kurzbeschreibung: Der Anwender(das System) liest alle SMS aus, die sich auf dem Modem befinden und löscht die ausgelesen SMS vom Modem.

Vorbedingungen: keine

Beschreibung des Ablaufs:
E1) Der Anwender fordert eine Liste mit SMS an.
A1) Das System liest die aktuelle SMS Liste vom Modem aus.
E2) Die Liste beinhaltet ein oder mehrere SMS.
A2) Das System filtert die Indices der SMS aus der Liste.
AE2) Die Liste mit SMS ist leer.
AA2) Das System beendet den Prozess zum Auslesen der SMS.
E3) Der Anwender fordert eine konkrete SMS aus der Liste der SMS an.
A3) Das System liest die SMS aus dem Modem aus.
E4) Der Anwender bereitet die SMS zum Speichern auf.
A4) Das System überprüft, ob die SMS ein korrektes Format hat und noch nicht im System angelegt ist
E5) Die SMS hat ein korrektes Format und ist noch nicht im System angelegt.
A5) Das System legt die SMS an.
AE5) Die SMS ist schon im System angelegt.

AA5) Das System legt die SMS nicht an. E6) Alle neuen SMS aus der aktuellen Liste sind im System angelegt. A6) Das System löscht die SMS aus dem Modem.
Auswirkungen: Der Speicher des Modem wird geleert.
Anmerkungen: keine

3.4.7 Logdaten verwalten



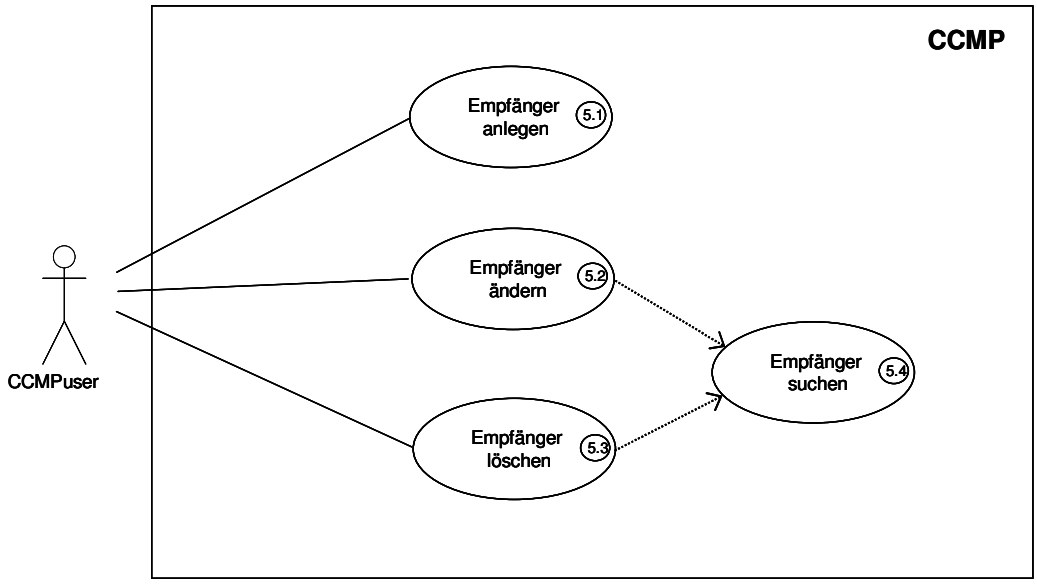
Anwendungsfall 6: Anwendungsfalldiagramm "Logdaten verwalten"

Nr.: 4.1
Titel: Logdaten auslesen
Kurzbeschreibung: Der Anwender(das System) liest Logdaten aus.
Vorbedingungen: keine
Beschreibung des Ablaufs: E1) Die CCEngine findet ein Gerät, für das das nächste Logdatum überschritten ist. A1) Das System baut eine Verbindung mit dem Gerät auf. E2) Es konnte eine Verbindung mit dem Gerät aufgebaut werden. A2) Das System fragt die Logdaten ab. AE2) Es konnte keine Verbindung mit dem Gerät aufgebaut werden. AA2) Das System beendet den Prozess. E3) Die Logdaten sind vollständig ausgelesen. A3) Das System trennt die Verbindung zu dem Gerät und überprüft, ob ein Logfile für das Gerät vorhanden ist und ob es noch nicht zu groß ist. E4) Es ist ein Logfile für das Gerät vorhanden und es ist nicht zu groß. A4) Das System öffnet das Logfile. AE4) Es ist kein Logfile für das Gerät vorhanden oder das Logfile ist zu groß. AA4) Das System legt ein neues Logfile an und öffnet es. E5) Die Logdaten sind bereit zur Überprüfung. A5) Das System überprüft, ob die Logdaten ein korrektes Format haben und keine alten Datumswerte beinhalten. E6) Die Logdaten haben ein korrektes Format und beinhalten keine alten Datumswerte. A6) Das System schreibt die Logdaten in das Logfile. AE6) Die Logdaten haben ein ungültiges Format oder beinhalten alte Datumswerte. AA6) Das System wirft die ungültigen Logdaten weg. E7) Alle Logdaten sind verarbeitet. A7) Das System berechnet ein neues Logdatum und beendet den Prozess.
Auswirkungen: keine
Anmerkungen: keine

Nr.: 4.2

Titel: Logdaten exportieren
Kurzbeschreibung: Der Anwender exportiert die Logdaten in einem CSV File.
Vorbedingungen: Der Anwender ist mit der Berechtigung CCMPuser oder höher eingeloggt
Beschreibung des Ablaufs: E1) Der Anwender will Logdaten exportieren. A1) Das System stellt dem Anwender eine Auswahlmaske zur Verfügung. E2) Der Anwender wählt ein bestimmtes Gerät aus. A2) Das System wartet auf weitere Eingaben. E3) Der Anwender gibt den Zeitraum, für den er die Logdaten exportieren will an. A3) Das System sucht nach Logdaten für die angegeben Suchkriterien. E4) Es wurden Logdaten gefunden, die den angegebenen Suchkriterien entsprechen. A4) Das System stellt ein Excel mit den Logdaten zum Export zur Verfügung. AE4) Es wurden keine Logdaten gefunden, die den angegebenen Suchkriterien entsprechen. AA4) Eine entsprechende Meldung wird angezeigt.
Auswirkungen: Ein CSV File mit den Logdaten wird erzeugt
Anmerkungen: keine

3.4.8 Empfänger verwalten



Anwendungsfall 7:Anwendungsfalldiagramm "Empfänger verwalten"

Nr.:5.1
Titel: Empfänger anlegen
Kurzbeschreibung: Der Anwender legt einen neuen Empfänger mit allen Daten (Vorname, Nachname, Telefonnummer und Email) an.
Vorbedingungen: Der Anwender ist mit der Berechtigung CCMPuser oder höher eingeloggt
Beschreibung des Ablaufs:

E1) Der Anwender gibt die Daten des Empfängers ein.
A1) Das System überprüft, ob alle notwendigen Daten eingegeben wurden, die Daten korrekt sind und der Empfänger noch nicht im System vorhanden ist.
E2) Alle erforderlichen Daten wurden eingegeben, die Daten sind korrekt und der Empfänger ist noch nicht im System vorhanden.
A2) Das System legt den Empfänger an.
AE2) Nicht alle erforderlichen Daten wurden eingegeben, die Daten sind nicht korrekt oder der Empfänger ist schon unter diesem Benutzer im System angelegt.
AA2) Eine Fehlermeldung wird ausgegeben.

Auswirkungen: Ein neuer Empfänger wurde angelegt und gespeichert

Anmerkungen: keine

Nr.: 5.2

Titel: Empfänger ändern

Kurzbeschreibung: Der Anwender ändert die Daten eines Empfängers

Vorbedingungen: Der Anwender ist mit der Berechtigung CCMPuser oder höher eingeloggt

Beschreibung des Ablaufs:
E1) Der Anwender wählt die Liste mit allen Empfängern.
A1) Das System zeigt eine Liste mit allen Empfängern an.
AE1) Der Anwender entscheidet sich einen Empfänger zu suchen.
AA1) Das System führt den Anwendungsfall 5.4 („Empfänger suchen“) aus.
E2) Der Anwender wählt den Empfänger, den er ändern will aus.
A2) Das System zeigt ein Formular mit allen Empfängerdaten an.
E3) Der Anwender ändert die Daten.
A3) Das System überprüft, ob alle notwendigen Daten eingegeben wurden und die Daten korrekt sind.
E4) Alle erforderlichen Daten wurden eingegeben und die Daten sind korrekt.
A4) Das System speichert den geänderten Empfänger ab.
AE4) Nicht alle erforderlichen Daten wurden eingegeben oder die Daten sind nicht korrekt.
AA4) Eine Fehlermeldung wird ausgegeben.

Auswirkungen: Die Daten des Empfängers wurden geändert und gespeichert.

Anmerkungen: Die Daten des Empfängers werden für alle Geräte geändert, denen er zugeordnet ist.

Nr.: 5.3

Titel: Empfänger löschen

Kurzbeschreibung: Der Anwender löscht einen Empfänger aus dem System

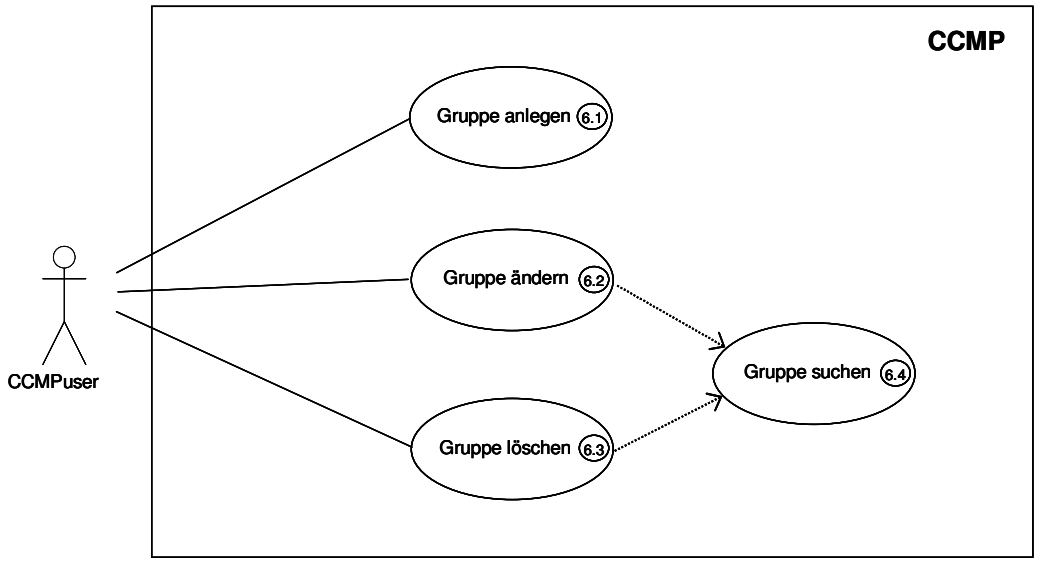
Vorbedingungen: Der Anwender ist mit der Berechtigung CCMPuser oder höher eingeloggt

Beschreibung des Ablaufs:
E1) Der Anwender wählt die Liste mit allen Empfängern.
A1) Das System zeigt eine Liste mit allen Empfängern an.
AE1) Der Anwender entscheidet sich einen Empfänger zu suchen.
AA1) Das System führt den Anwendungsfall 5.4 („Empfänger suchen“) aus.
E2) Der Anwender wählt den Empfänger, den er löschen will aus.
A2) Das System fragt den Anwender, ob er den Empfänger wirklich löschen will.
E3) Der Anwender entscheidet sich dafür den Empfänger zu löschen.

A3) Das System löscht den Empfänger. AE3) Der Anwender entscheidet sich dafür, den Empfänger nicht zu löschen. AA3) Das System löscht den Empfänger nicht.
Auswirkungen: Ein Empfänger wurde gelöscht.
Anmerkungen: Wird ein Empfänger gelöscht, so wird er aus allen Geräten gelöscht und steht in der Gerätekonfiguration nicht mehr zur Verfügung. Die betroffenen Geräte sollten nach dem Löschen eines Empfängers unbedingt synchronisiert werden.

Nr.: 5.4
Titel: Empfänger suchen
Kurzbeschreibung: Der Anwender sucht einen Empfänger anhand seines Namens, der Telefonnummer oder der Email Adresse.
Vorbedingungen: Der Anwender ist mit der Berechtigung CCMPuser oder höher eingeloggt
Beschreibung des Ablaufs: E1) Der Anwender gibt den Begriff für den Empfänger nach dem er suchen will in das Feld für die Volltextsuche ein. A1) Das System sucht nach Empfängern, auf die der angegebene Suchbegriff passt. E2) Es existieren Empfänger, auf die der Suchbegriff passt. A2) Die gefundenen Empfänger werden in einer Liste angezeigt. AE2) Es konnte keine Empfänger gefunden werden, auf die der Suchbegriff passt. AA2) Es wird eine entsprechende Meldung angezeigt.
Auswirkungen: Keine
Anmerkungen: Keine

3.4.9 Gruppe verwalten



Anwendungsfall 8: Anwendungsfalldiagramm "Gruppe verwalten"

Nr.: 6.1
Titel: Gruppe anlegen
Kurzbeschreibung: Der Anwender legt eine neue Gruppe an.

Vorbedingungen: Der Anwender ist mit der Berechtigung CCMPuser oder höher eingeloggt
Beschreibung des Ablaufs: E1) Der Anwender gibt den Namen der Gruppe ein. A1) Das System überprüft, ob der Name der Gruppe eingegeben wurde und korrekt ist. E2) Der Name der Gruppe wurde eingegeben und ist korrekt. A2) Das System legt die Gruppe an. AE2) Der Name der Gruppe wurde nicht eingegeben oder ist nicht korrekt. AA2) Eine Fehlermeldung wird ausgegeben.
Auswirkungen: Ein neue Gruppe wurde angelegt und gespeichert
Anmerkungen: keine

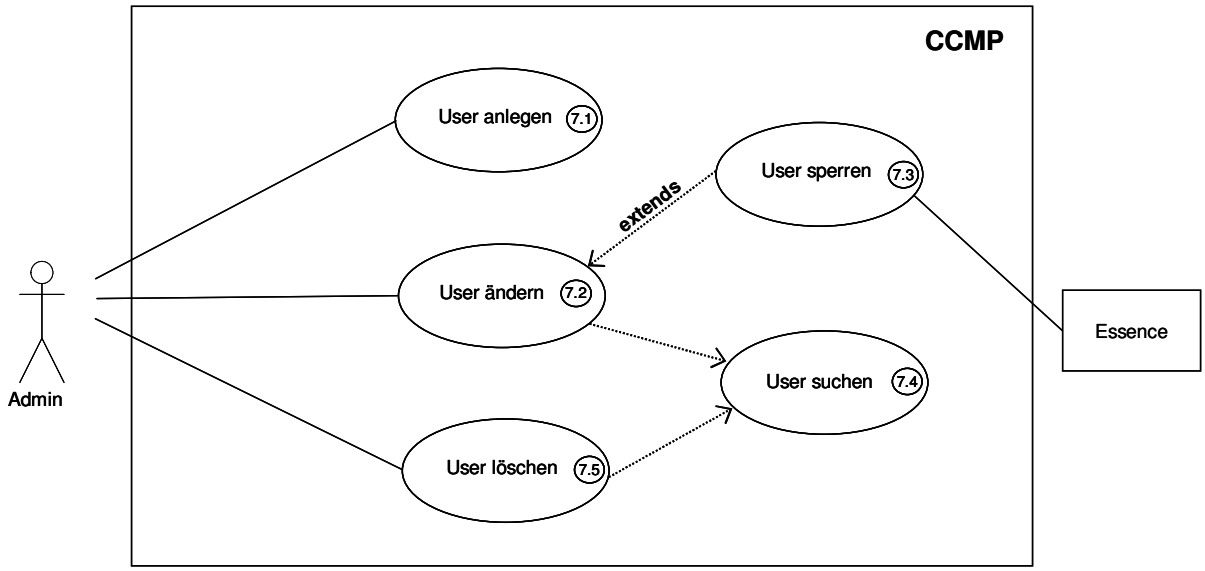
Nr.: 6.2
Titel: Gruppe ändern
Kurzbeschreibung: Der Anwender ändert den Namen einer Gruppe
Vorbedingungen: Der Anwender ist mit der Berechtigung CCMPuser oder höher eingeloggt
Beschreibung des Ablaufs: E1) Der Anwender wählt die Liste mit allen Gruppen. A1) Das System zeigt eine Liste mit allen Gruppen an. AE1) Der Anwender entscheidet sich eine Gruppe zu suchen. AA1) Das System führt den Anwendungsfall 6.4 („Gruppe suchen“) aus. E2) Der Anwender wählt die Gruppe, die er ändern will aus. A2) Das System zeigt ein Formular mit dem Gruppennamen an. E3) Der Anwender ändert den Namen der Gruppe. A3) Das System überprüft, ob der Name der Gruppe eingegeben wurde und korrekt ist. E4) Der Name der Gruppe wurde eingegeben und ist korrekt. A4) Das System speichert die geänderte Gruppe ab. AE4) Der Name der Gruppe wurde nicht eingegeben oder ist nicht korrekt. AA4) Eine Fehlermeldung wird ausgegeben.
Auswirkungen: Die Daten des Empfängers wurden geändert und gespeichert.
Anmerkungen: Keine

Nr.: 6.3
Titel: Gruppe löschen
Kurzbeschreibung: Der Anwender löscht eine Gruppe aus dem System
Vorbedingungen: Der Anwender ist mit der Berechtigung CCMPuser oder höher eingeloggt
Beschreibung des Ablaufs: E1) Der Anwender wählt die Liste mit allen Gruppen. A1) Das System zeigt eine Liste mit allen Gruppen an. AE1) Der Anwender entscheidet sich, eine Gruppe zu suchen. AA1) Das System führt den Anwendungsfall 6.4 („Gruppe suchen“) aus. E2) Der Anwender wählt die Gruppe, die er löschen will aus. A2) Das System fragt den Anwender, ob er die Gruppe wirklich löschen will. E3) Der Anwender entscheidet sich dafür die Gruppe zu löschen. A3) Das System löscht die Gruppe. AE3) Der Anwender entscheidet sich dafür, die Gruppe nicht zu löschen. AA3) Das System löscht die Gruppe nicht.

Auswirkungen: Der Empfänger wurde gelöscht
Anmerkungen: Wird eine Gruppe aus dem System gelöscht, werden alle Geräte, die in dieser Gruppe sind auch gelöscht.

Nr.: 6.4
Titel: Gruppe suchen
Kurzbeschreibung: Der Anwender sucht eine Gruppe anhand ihres Namens.
Vorbedingungen: Der Anwender ist mit der Berechtigung CCMPuser oder höher eingeloggt
Beschreibung des Ablaufs: E1) Der Anwender gibt den Begriff für den Gruppe nach der er suchen will in das Feld für die Volltextsuche ein. A1) Das System sucht nach Gruppen, auf die der angegebene Suchbegriff passt. E2) Es existieren Gruppen, auf die der Suchbegriff passt. A2) Die gefundenen Gruppen werden in einer Liste angezeigt. AE2) Es konnte keine Gruppe gefunden werden, auf die der Suchbegriff passt. AA2) Es wird eine entsprechende Meldung angezeigt.
Auswirkungen: Keine
Anmerkungen: Keine

3.4.10 User verwalten



Anwendungsfall 9: Anwendungsfalldiagramm "User verwalten"

Nr.: 7.1
Titel: User anlegen
Kurzbeschreibung: Der Anwender legt einen neuen User an.
Vorbedingungen: Der Anwender ist mit der Berechtigung Admin eingeloggt
Beschreibung des Ablaufs: E1) Der Anwender gibt die Daten des Users ein und ordnet ihm eine Benutzergruppe zu. A1) Das System überprüft, ob alle notwendigen Daten eingegeben wurden und der User noch nicht im System vorhanden ist.

E2) Alle erforderlichen Daten wurden eingegeben und der User ist noch nicht im System vorhanden.
A2) Das System legt den User an.
AE2) Nicht alle erforderlichen Daten wurden eingegeben oder der User ist schon im System vorhanden.
AA2) Eine Fehlermeldung wird ausgegeben.

Auswirkungen: Ein neuer User wurde angelegt und gespeichert

Anmerkungen: keine

Nr.: 7.2

Titel: User ändern

Kurzbeschreibung: Der Anwender ändert die Daten eines Users

Vorbedingungen: Der Anwender ist mit der Berechtigung Admin eingeloggt.

Beschreibung des Ablaufs:
E1) Der Anwender wählt die Liste mit allen Usern.
A1) Das System zeigt eine Liste mit allen Usern an.
AE1) Der Anwender entscheidet sich einen User zu suchen.
AA1) Das System führt den Anwendungsfall 7.5 („User suchen“) aus.
E2) Der Anwender wählt den User, den er ändern will aus.
A2) Das System zeigt ein Formular mit allen Userdaten an.
E3) Der Anwender ändert die Daten.
A3) Das System überprüft ob alle notwendigen Daten eingegeben wurden und die Daten korrekt sind.
AE3) Der Anwender entscheidet sich den User zu sperren.
AA3) Das System führt den Anwendungsfall „User sperren“ aus.
E4) Alle erforderlichen Daten wurden eingegeben und die Daten sind korrekt.
A4) Das System speichert den geänderten User ab.
AE4) Nicht alle erforderlichen Daten wurden eingegeben oder die Daten sind nicht korrekt.
AA4) Eine Fehlermeldung wird ausgegeben.

Auswirkungen: Die Daten des Users wurden geändert und gespeichert.

Anmerkungen: Keine

Nr.: 7.3

Titel: User sperren

Kurzbeschreibung: Der Anwender sperrt einen User.

Vorbedingungen: Der Anwender ist mit der Berechtigung Admin eingeloggt.

Beschreibung des Ablaufs:
E1) Der Anwender wählt User sperren aus.
A1) Das System fragt den Anwender, ob er den User wirklich sperren will.
E2) Der Anwender entscheidet sich dafür den User zu sperren.
A2) Das System sperrt den User.
AE2) Der Anwender entscheidet sich dafür den User nicht zu sperren.
AA2) Das System sperrt den User nicht.
E3) Der User ist gesperrt.
A3) Das System versendet eine Benachrichtigung an den gesperrten Benutzer.

Auswirkungen: Der User wurde gesperrt und ist inaktiv.

Anmerkungen: Ein gesperrter User kann sich nicht mehr einloggen.

Nr.: 7.4

Titel: User suchen
Kurzbeschreibung: Der Anwender sucht einen User
Vorbedingungen: Der Anwender ist mit der Berechtigung Admin eingeloggt.
Beschreibung des Ablaufs: E1) Der Anwender gibt den Begriff für den User nach dem er suchen will in der Suchmaske ein. A1) Das System sucht nach Usern, auf die der angegebene Suchbegriff passt. E2) Es existieren User, auf die der Suchbegriff passt. A2) Die gefundenen User werden in einer Liste angezeigt. AE2) Es konnte kein User gefunden werden, auf den der Suchbegriff passt. AA2) Es wird eine entsprechende Meldung angezeigt.
Auswirkungen: Keine
Anmerkungen: Keine

Nr.: 7.5
Titel: User löschen
Kurzbeschreibung: Der Anwender löscht einen User
Vorbedingungen: Der Anwender ist mit der Berechtigung Admin eingeloggt.
Beschreibung des Ablaufs: E1) Der Anwender wählt die Liste mit allen Usern. A1) Das System zeigt eine Liste mit allen Usern an. AE1) Der Anwender entscheidet sich einen User zu suchen. AA1) Das System führt den Anwendungsfall 5.4 („User suchen“) aus. E2) Der Anwender wählt den User, den er löschen will aus. A2) Das System fragt den Anwender, ob er den User wirklich löschen will. E3) Der Anwender entscheidet sich dafür den User zu löschen. A3) Das System löscht den User. AE3) Der Anwender entscheidet sich dafür, den User nicht zu löschen. AA3) Das System löscht den User nicht.
Auswirkungen: Keine
Anmerkungen: User sollten nur in seltenen und eindeutigen Fällen gelöscht werden. In allen anderen Fällen sollten sie inaktiv gesetzt werden.

3.5 Domänenmodell

Das Domänenmodell zeigt die Objekte der Anwendungsdomäne und ihre wichtigsten Attribute. Die Darstellung erfolgt mittels eines UML Klassendiagramms. [vgl. Zuse01, Seite 91,92]

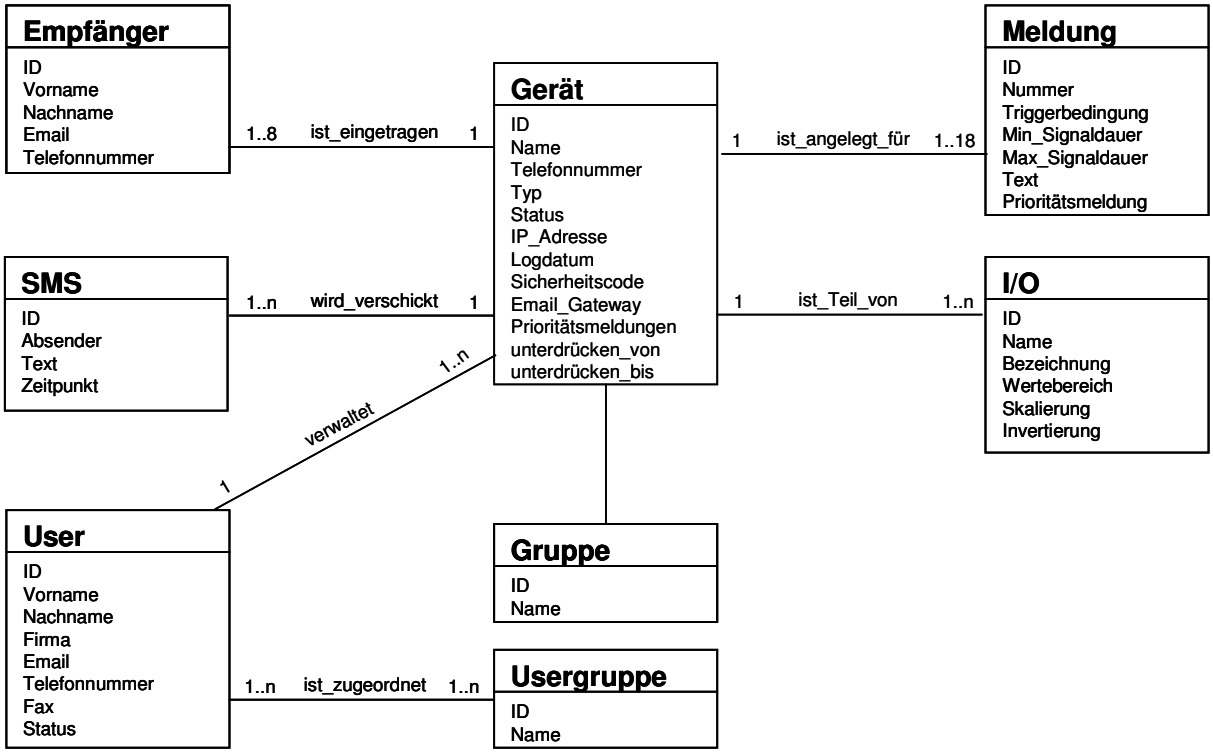


Abbildung 10: Domänenmodell

4 Die Applikation

Diese Kapitel beschäftigt sich mit der Umsetzung der CCMP Applikation. Es werden die grafische Benutzerschnittstelle, das Datenmodell, die Architektur und die wichtigsten Funktionalitäten von CCMP beschrieben.

4.1 Die grafische Benutzerschnittstelle

Dieser Abschnitt beschäftigt sich mit der Umsetzung des grafischen User Interfaces. Es wird im Folgenden anhand von Screenshots erklärt wie die in der Analysephase definierten Anforderungen in der Webapplikation umgesetzt wurden. Die grafische Benutzerschnittstelle wurde für eine einfache und übersichtliche Verwaltung des Communication Center durch den Benutzer konzipiert. Sie setzt sich, wie in Abbildung 11 zu sehen ist, aus vier Frames zusammen: Der Hauptnavigation, dem Menübaum, dem Content-Frame und einem Frame mit den Kontaktdaten von LeP. Der Inhalt des Content-Frame ändert sich laufend, es ist sozusagen die Arbeitsfläche des Benutzers (hat der Benutzer nichts anderes ausgewählt, so wird die CCMP Hilfe im Content-Frame dargestellt). Die anderen Frames ändern ihren Inhalt nicht.

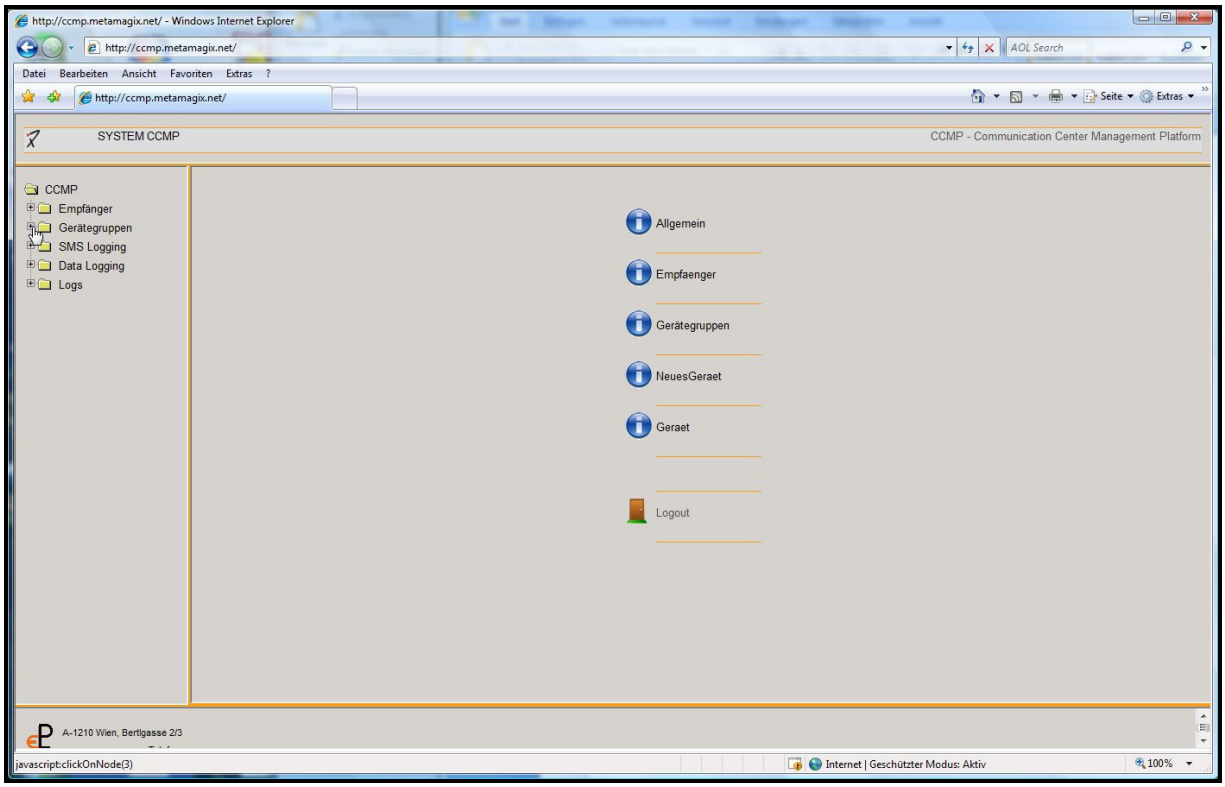


Abbildung 11: Screenshot "Startseite CCMP"

4.1.1 Die Hauptnavigation

Im obersten Frame befindet sich die Hauptnavigation. Diese hat im Moment nur zwei Einträge: „System“ und „CCMP“. Unter „System“ findet der Benutzer eine Möglichkeit die Applikation zu beenden („Logout“). Im Menüpunkt „CCMP“ findet der Benutzer die Möglichkeit vor, den Menübaum aufzurufen. Die Hauptnavigation hat im Moment keine große Bedeutung, sie wurde vor allem in Hinblick auf eine mögliche Erweiterung von CCMP implementiert, da sie dann weitere Menüeinträge enthalten würde.

4.1.2 Der Menübaum

Der Menübaum ist immer zu sehen und gibt dem Benutzer einen kompletten Überblick über die zu verwaltenden Objekte und Funktionalitäten der Applikation. In Abbildung 12 ist der vollständig ausgeklappte Menübaum zu sehen.

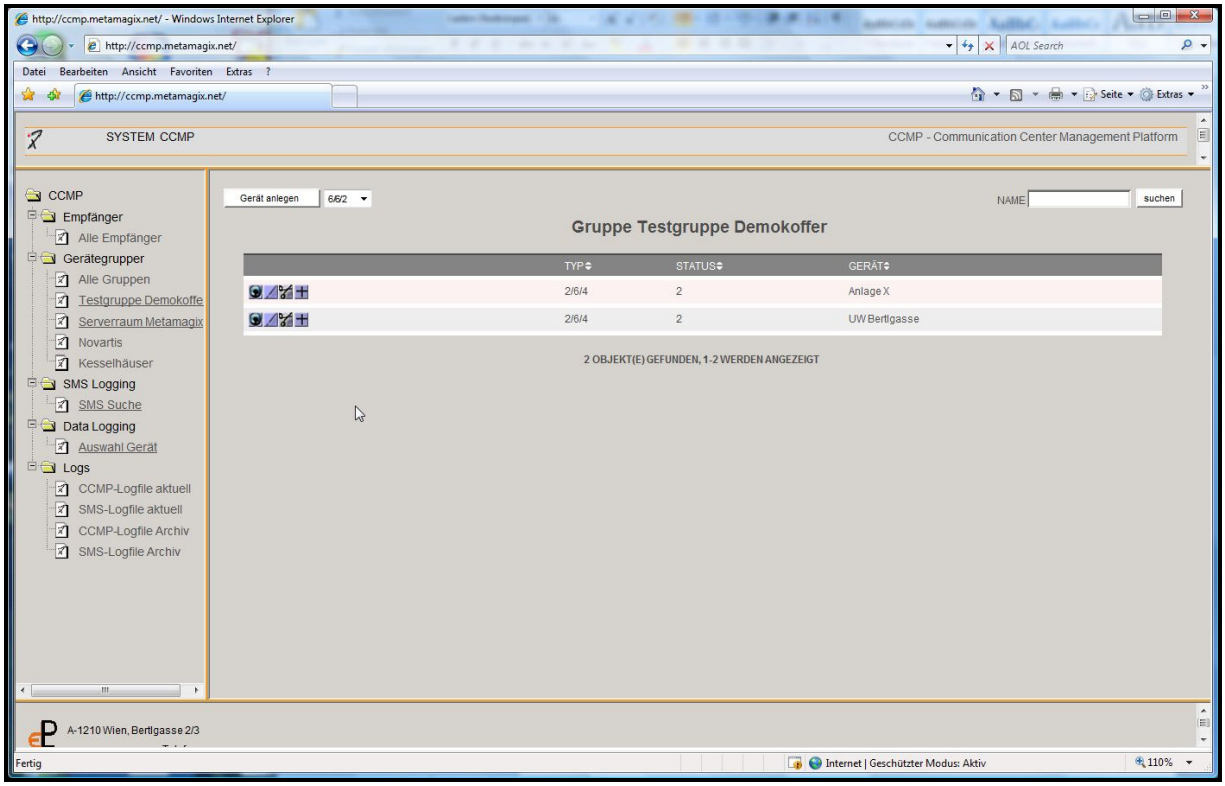


Abbildung 12: Screenshot "Testgruppe Demokoffer"

4.1.3 Der Content Frame

Im Content Frame findet die eigentliche Verwaltung der Objekte, die Darstellung von Objektlisten und die Arbeit mit verschiedenen Funktionalitäten wie „Datenlogs abfragen“ oder „SMS Suche“ statt. Abbildung 12 zeigt die Darstellung der Geräte in der Gruppe „Testgruppe Demokoffer“ in Form einer Liste.

4.1.4 Empfänger verwalten

Die Empfänger können über den Menüeintrag „Empfänger“ verwaltet werden. Über den Untermenüpunkt „Alle Empfänger“ kann sich der Anwender eine Liste aller Empfänger anzeigen lassen. In der Liste kann er Empfänger hinzufügen, ansehen, editieren, löschen und kopieren. Pro Empfänger können der Vorname, Nachname, die Telefonnummer und die Email Adresse eingegeben werden. Empfänger sind die Personen, an die das Communication Center die Meldungen per SMS oder Email versendet.

4.1.5 Geräte und Gruppen verwalten

Über den Menüpunkt „Gerätegruppen“ werden die Gruppen und ihre Geräte (Communication Centers) bearbeitet. Der Benutzer hat die Möglichkeit eine Liste aller Gruppe über den Eintrag „Alle Gruppen“ aufzurufen, in der Liste hat er die Möglichkeit, Gruppen anzulegen, anzusehen, zu ändern, zu löschen und zu kopieren. Die einzelnen Gruppen werden ebenfalls als Menüeinträge im Menüpunkt „Gerätegruppen“ angezeigt (siehe Abbildung 12). Wählt der User eine der Gruppen aus, so werden die zugehörigen Geräte im Content Frame angezeigt. Die Geräte können wiederum über die angezeigte Liste angelegt, angesehen, geändert, gelöscht und kopiert werden.

4.1.5.1 Geräte verwalten

Legt der User ein Gerät an, so wählt der zuerst das Modell aus. In einem nächsten Schritt gibt er die Telefonnummer, die Gruppe, der er das Gerät zuordnen will und das gewünschte Logintervall an. Hat er diese Daten fertig eingegeben, so hat er die Möglichkeit das Gerät entweder zu initialisieren, d.h. die Daten vom Communication Center mit der angegeben Telefonnummer einzulesen oder den Initialisierungsschritt auszulassen und selber die Konfiguration zu beginnen. Für ein Gerät können unter dem Punkt „Gerät ändern“ seine Attribute eingegeben werden und es können ihm Empfänger zugeordnet werden. Die wichtigste Ansicht für den Benutzer ist „Gerät ansehen“. Hier kann er bis zu 18 Meldungen und eine vom Gerätetyp abhängige Anzahl an I/Os verwalten. Die Meldungen und I/Os werden wie in Abbildung 13 gezeigt in einer Liste angezeigt. Desweiteren kann er in der Maske „Gerät ansehen“ den Synchronisationsprozess starten, er hat einen Überblick über den Synchronisationsstatus des Gerätes, hat die Möglichkeit das automatisierte Datenloggen einzustellen und zu starten und kann sich ein PDF mit der vollständigen Geräte Konfiguration downloaden.

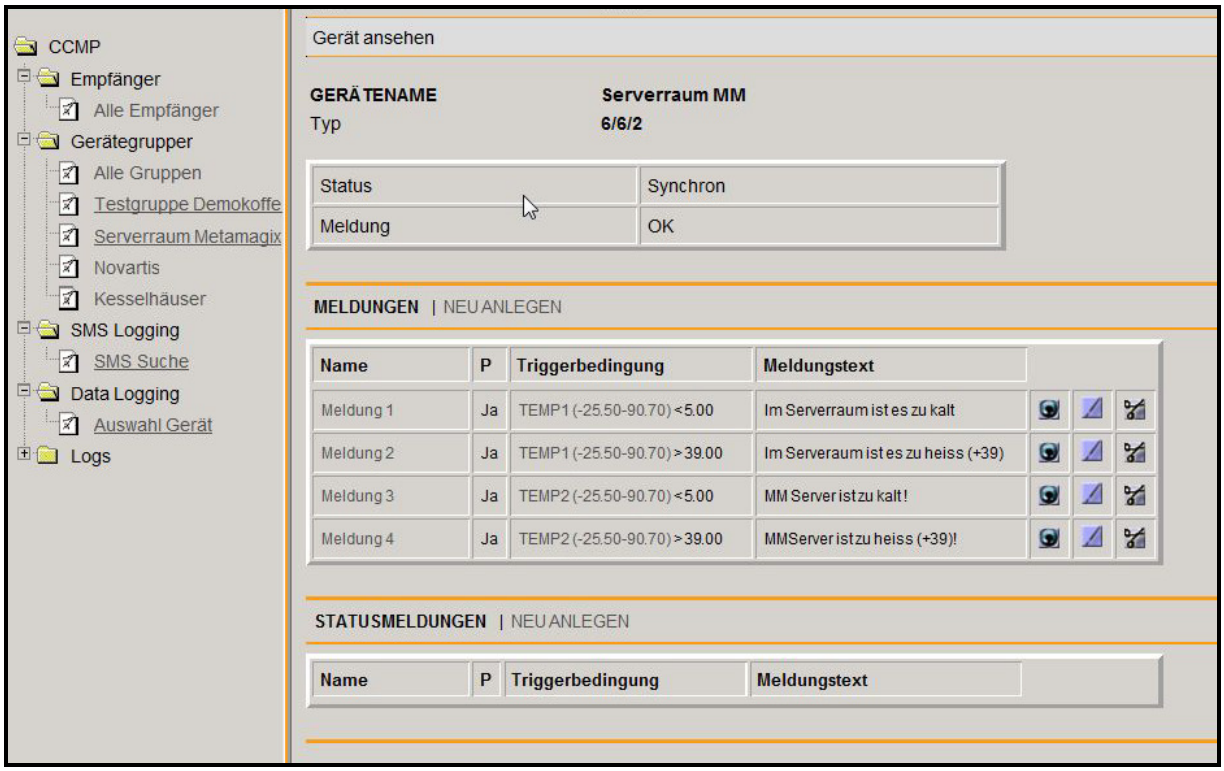


Abbildung 13: Screenshot "Gerät ansehen"

4.1.5.2 Das automatisierte Datenlogging

Das automatisierte Datenlogging kann vom Benutzer für jedes Gerät individuell eingestellt werden. Er hat die Möglichkeit unter „Gerät ansehen“ ein Logintervall in Stunden anzugeben und mit dem Button „Daten Loggen“ kann er das automatisierte Datenlogging jederzeit starten.

4.1.6 SMS Logging

Der Menüpunkt „SMS Logging“ hat den Eintrag „SMS Suche“, der dem User eine Maske zur Suche über die archivierten SMS liefert. Er kann die SMS für ein bestimmtes Gerät und/oder über einen bestimmten Zeitraum suchen. Die Suchmaske ist in Abbildung 14 zu sehen, in diesem Fall hat der Benutzer keine Auswahl getroffen und es werden alle im Archiv vorhandenen SMS angezeigt.

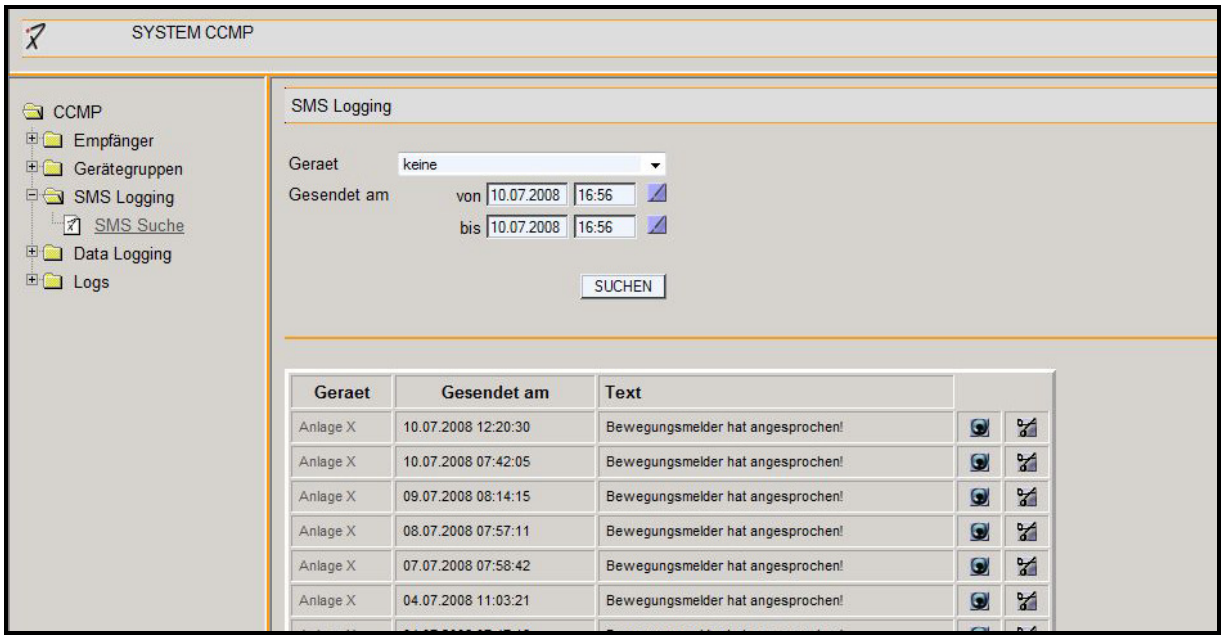


Abbildung 14: Screenshot "SMS Logging"

4.1.7 Data Logging

Der Menüeintrag „Data Logging“ hat einen Eintrag „Auswahl Gerät“. Dieser Eintrag führt den User zu einer Eingabemaske, in der er ein bestimmtes Communication Center auswählen und einen Zeitraum angeben kann. Er bekommt ein Excel File mit den geloggten Daten des Geräts über die gewünschte Zeitspanne zum Download. Die Maske ist in Abbildung 15 zu sehen.

Abbildung 15: Screenshot "Data Logging"

4.1.8 System Logs

Über den Menüeintrag „Logs“ hat der Benutzer die Möglichkeit, sich die aktuellen Logfiles des Systems im Browser anzeigen zu lassen. Er kann zwischen zwei Arten von Logfiles wählen: dem CCMP-Logfile und dem SMS-Logfile. Im CCMP-Logfile wird die Kommunikation zwischen der CCMP Applikation und den Communication Centern aufgezeichnet, im SMS-Logfile das Auslesen der SMS aus dem Modem geloggt. Außerdem gibt es die Möglichkeit

über die archivierten Logfiles zu suchen und sich ein altes Logfiles anzusehen. Diese Funktionalität ist beim Auftreten von Fehlern von Interesse, da diese Logfiles die Grundlage für die Fehlerbehebung sind. Grüne Einträge bedeuten, dass alles ordnungsgemäß abläuft, rote Einträge kennzeichnen Fehler. Schwarze Logeinträge sind von geringem Interesse.

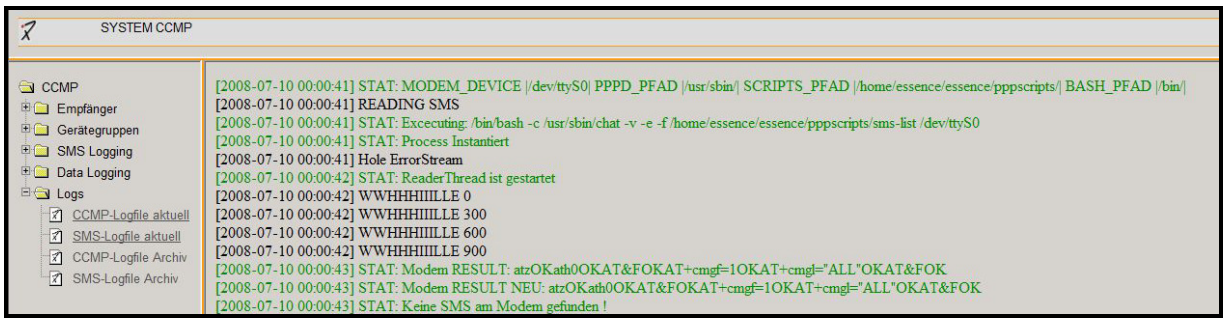


Abbildung 16: Screenshot "Logs-SMS Logfile"

4.2 Das Datenmodell

Im Folgenden wird das Datenmodell mit Hilfe eines EER-Diagrammes dargestellt. Anschließend werden die wichtigsten Entitäten kurz beschrieben. In dieser Detailbeschreibung erfolgt die Benennung der Attribute bereits wie auf Datenbankebene. Es werden jeweils der Attributname, der Datentyp, eine kurze Beschreibung sowie eine Klassifizierung angegeben.

4.2.1 Das EER-Modell

Die Grundlage des EER-Modells ist das in der Anforderungsanalyse erarbeitete Domänenmodell. Dieses wurde ausgearbeitet und in ein EER-Diagramm überführt, welches in Abbildung 17 zu sehen ist. Zur Vereinfachung des Diagramms wurden nicht alle Attribute der Entitäten aufgelistet:

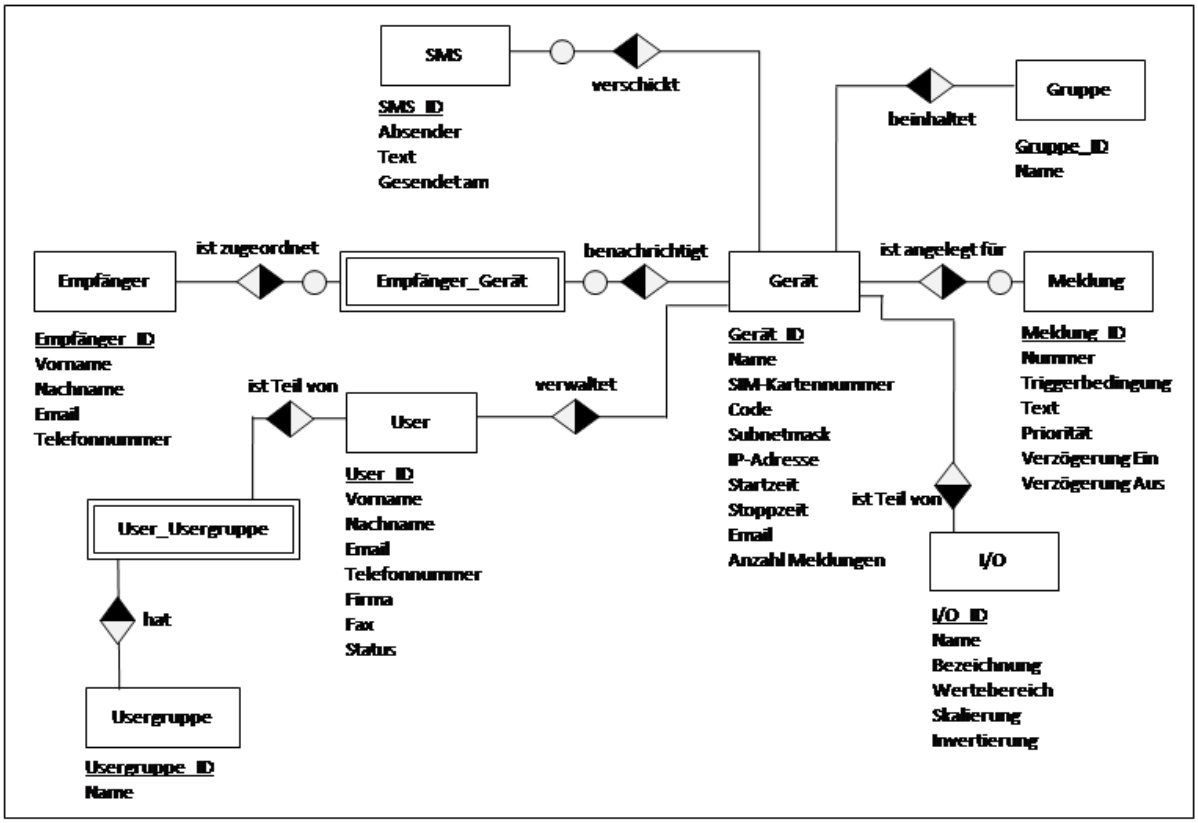


Abbildung 17: EER-Modell

4.2.2 Gerät

Die Entität „Gerät“ ist eine Repräsentation des Communication Centers innerhalb der CCMP Applikation. „Gerät“ bildet die Attribute des Communication Centers ab, zusätzlich gibt es noch Attribute, die für die Applikationslogik benötigt werden.

Name	Datentyp	Beschreibung	Attribut des CC
ID	varchar(50)	Eine eindeutige, im System einmalige ID	Nein
Gruppen_ID	varchar(50)	Die ID der Gruppe, der das Gerät zugeordnet ist	Nein
Name	varchar(50)	Der Name des Geräts	Ja
simnummer	varchar(50)	Die Nummer der Sim-Karte, unter der das Communication Center erreichbar ist.	Ja
email	varchar(50)	Die Telefonnummer eines SMS-to-Email Providers	Ja
anzahl_meldungen	varchar(5)	Die Anzahl der Meldungen, die für das Gerät angelegt werden dürfen	Nein
startzeit	varchar(10)	Die Startzeit der Zeitspanne in der Meldungen mit niedriger Priorität unterdrückt werden.	Ja
stoppzeit	varchar(10)	Die Endzeit der Zeitspanne in der Meldungen mit niedriger Priorität unterdrückt werden.	Ja
code	varchar(4)	Ein vierstelliger Code zur Authentifizierung am Communication Center, wird von CCMP nicht benötigt	Ja
subnet	varchar(20)	Die für das Gerät eingestellte Subnetmask. Dieses Attribut sollte nicht verändert werden.	Ja
ip	varchar(20)	Die IP Adresse des Geräts. Dieses Attribut sollte nicht verändert werden.	Ja
typ	varchar(15)	Das Gerätemodell	Nein
useraccess	varchar(50)	Die ID des Benutzer,s der das Gerät angelegt hat und verwalten darf	Nein
initialisiert	int(1)	Zeigt an ob ein Gerät initialisiert wurde oder nicht	Nein
initialisiertam	timestamp	Der Zeitpunkt an dem das Gerät initialisiert wurde	Nein
lastsyncdate	timestamp	Der letzte Synchronisationszeitpunkt	Nein
status	int(1)	Der Synchronisationsstatus des Gerätes	Nein
nextlogdate	timestamp	Der nächste Logzeitpunkt des Gerätes	Nein
lastlogdate	timestamp	Der letzte Logzeitpunkt des Gerätes	Nein
logintervall	int(5)	Das eingestellte Logintervall	Nein
logstatus	int(1)	Der Logstatus des Gerätes	Nein
logtries	int(1)	Die Anzahl der Logversuche während eines Logvorganges	Nein

4.2.3 I/O

Die Entität „I/O“ entspricht den Ein- und Ausgängen des Communication Centers.

Name	Datentyp	Beschreibung	Attribut des CC
ID	Varchar(50)	Eine eindeutige, im System einmalige ID	Nein
Geraet_ID	varchar(50)	Die ID des Geräts, dem der I/O zugeordnet ist.	Nein

Name	varchar(40)	Der Name des I/Os.	Ja
Art	varchar(50)	Eine eindeutige, nicht veränderbare Bezeichnung für den I/O.	Ja
Untergrenze	varchar(50)	Untergrenze des akzeptierten Wertebereichs für analoge Eingänge.	Ja
Obergrenze	varchar(50)	Obergrenze des akzeptierten Wertebereichs für analoge Eingänge.	Ja
Einheit	varchar(100)	Skalierungseinheit für die zu messenden Werte bei analogen Eingängen.	Ja
Invert	Int(1)	Invertierungsparameter bei digitalen I/Os	Ja

4.2.4 Meldung

„Meldung“ entspricht den vom Communication Center verschickten Nachrichten.

Name	Datentyp	Beschreibung	Attribut des CC
ID	Varchar(50)	Eine eindeutige, im System einmalige ID	Nein
Geraet_ID	varchar(50)	Die ID des Geräts, dem die Meldung zugeordnet ist.	Nein
IO_ID	varchar(50)	Die ID des I/O für den die Meldung ausgelöst wird	Nein
Name	varchar(40)	Die eindeutige Nummer der Meldung.	Ja
verzoegerung_ein	varchar(20)	Die Zeit, die die Triggerbedingung gültig sein muss, bis eine Meldung versandt wird.	Ja
verzoegerung_aus	varchar(20)	Die Zeit, die die Triggerbedingung nicht mehr gültig sein muss, bis das Versenden von Meldungen eingestellt wird.	Ja
text	varchar(255)	Der Meldungstext	Ja
triggeroperator	char()	Der mathematische Operator der Triggerbedingung	Nein
triggerbedingung	varchar(100)	Die Bedingung, die erfüllt sein muss, damit eine Meldung versendet wird.	Nein
prioritaetsmeldung	int(1)	Diese Attribut gibt an, ob eine Meldung eine Prioritätsmeldung ist oder nicht	Nein

4.2.5 Empfänger

Mit der Entität „Empfänger“ werden die vom Communication Center benachrichtigten Personen bezeichnet.

Name	Datentyp	Beschreibung	Attribut des CC
ID	Varchar(50)	Eine eindeutige, im System einmalige ID	ID
Vorname	varchar(30)	Der Vorname des Empfängers	Nein
Nachname	varchar(30)	Der Nachname des Empfängers	Nein
Telefonnummer	varchar(50)	Die Telefonnummer des Empfängers	Ja

Email	varchar(100)	Die Email Adresse des Empfängers	Ja

4.2.6 SMS

Die Entität „SMS“ existiert nur innerhalb der Webapplikation. Sie dient zur Archivierung der versendeten SMS.

Name	Datentyp	Beschreibung	Attribut des CC
ID	Varchar(50)	Eine eindeutige, im System einmalige ID	Nein
Geraet_ID	varchar(50)	Die ID des Geräts, das die SMS versendet hat.	Nein
telefonnummer	Varchar(50)	Die Telefonnummer des CC, das die SMS versendet hat.	Nein
text	varchar(50)	Der Text der SMS im Originalformat.	Nein
webtext	varchar(50)	Der Text der SMS, der für die Weboberfläche formatiert wurde.	Nein
gesendet_am	varchar(50)	Der Zeitpunkt an dem die SMS versendet wurde	Nein

4.3 Die Architektur

In diesem Abschnitt wird der Aufbau der Applikation betrachtet. Zuerst folgt ein allgemeiner Überblick über die Architektur, dann werden die einzelnen Komponenten (Klassen), ihre Funktionalität und ihr Zusammenspiel beschrieben. Desweiteren werden die Kommunikationsabläufe mit Hilfe von Sequenzdiagrammen dargestellt werden. Klassen- und Methodennamen werden im Folgenden *kursiv* geschrieben

4.3.1 Allgemeiner Überblick

Die Architektur der CCMP Applikation besteht aus zwei Hauptkomponenten. Die eine Komponente ist das eSSENCE™ Framework, sie umfasst das grafische User Interface, zur Verwaltung der Geräte, der geloggten Daten, SMS und Logfiles. Das User Interface basiert auf eSSENCE™ und nutzt die von eSSENCE™ zur Verfügung gestellten Agenten und Kernfunktionalitäten um die Objekte in der Datenbank zu verwalten. Auf diesen Teil der Applikation wird im folgenden Abschnitt nicht weiter eingegangen. Die andere Komponente ist die CCEngine, die als eigenständiger Prozess läuft und sich selbstständig um die Initialisierung bzw. Synchronisation der Geräte, das Datenloggen und die Abfrage von SMS am Modem kümmert. Zum Speichern und Abrufen von Objekten nutzt sie Funktionen des eSSENCE™ Frameworks. Dieser Aspekt der Architektur ist in Abbildung 18 dargestellt:

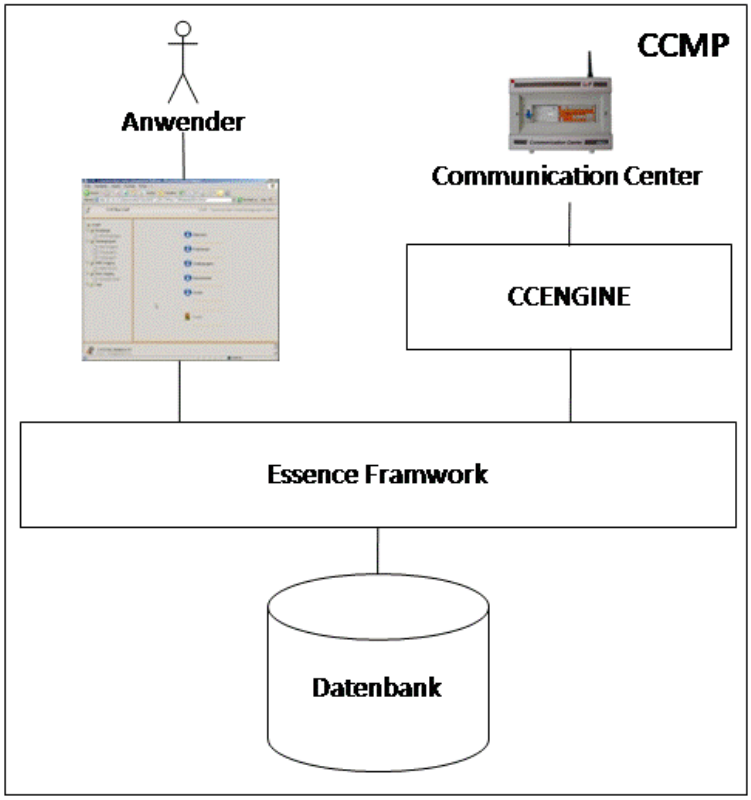


Abbildung 18: Architektur CCMP

4.3.2 CCEngineI

CCEngineI implementiert das Interface *CCEngine*, das eine Erweiterung der RMI Komponente *Remote* ist. *CCEngineI* wird mit eSSENCE™ gemeinsam gestartet und läuft als eigenständiger RMI-Prozess, der als Herzstück von CCMP betrachtet werden kann. Seine Aufgabe ist es, die Kommunikation mit den Communication Centern abzuwickeln und am serverseitigen Modem die hereinkommenden SMS abzufragen. Dazu kann die Instanz der Klasse *CCEngineI* auf ein Set von verschiedenen Klassen zugreifen. Ist der RMI Prozess gestartet, wird, durch einen Timeout Thread geregelt, alle 5 Minuten die Methode *run()* durchlaufen. In dieser Methode werden immer dieselben drei Routinen durchlaufen:

- **Synchronisation von Geräten:** In diesem Teil wird eine Datenbankabfrage gestartet, die nach Geräten sucht, die den Status „zu synchronisieren“ haben. Es werden maximal 10 Geräte pro Durchlauf von *run()* verarbeitet. Ein Gerät nach dem anderen wird entweder synchronisiert oder initialisiert. Dazu wird die Klasse *CCSynchronizer* instanziert und ihre Methode *synchronize()* aufgerufen. Diese Methode gibt nach Abschluss des Kommunikationsprozess, je nachdem, ob das Gerät mit dem dazugehörigen Communication Center synchronisiert werden konnte oder nicht, *true* bzw. *false* zurück. Dann wird das Gerät auf die Datenbank gespeichert.
- **Abfragen von SMS:** Die zweite Routine ist dafür zuständig, dass die vom Communication Center an das Modem gesendeten SMS, vom Modem ausgelesen und in der Datenbank gespeichert werden. Es wird eine Instanz der Klasse *SMSEngine* erzeugt und ihre Methode *readSMS()* aufgerufen. *readSMS()* kapselt die gesamte Logik des Auslesevorgangs. Sobald das Auslesen beendet ist, wird überprüft, ob währenddessen Fehler aufgetreten sind.
- **Abfragen von geloggten Daten:** Die dritte und letzte Routine in *run()* kümmert sich darum, dass die geloggten Daten aus den Communication Centern ausgelesen werden. Dazu wird eine Datenbankabfrage durchgeführt, deren Ergebnis alle Geräte sind, deren nächstes Logdatum (der Zeitpunkt an dem wieder Daten abzufragen sind) älter als das aktuelle Datum ist. Für jedes dieser Geräte passiert folgendes: Es wird eine Instanz von *CCSynchronizer* erzeugt und ihre Methode *readLog()* aufgerufen. In dieser Methode ist wird der gesamte Prozess ausgeführt, am Ende wird je nach Erfolg *true* oder *false* an die CCEngineI zurückgegeben.

Am Ende von *run()* wird der Timeout Thread gestartet und die *CCEngineI* pausiert für 5 Minuten. Dann wird *run()* erneut durchlaufen.

4.3.3 CCSynchronizer

Eine Instanz von *CCSynchronizer* wird von der *CCEngineI* zur Ausführung von Kommunikationsprozessen mit dem Communication Center benötigt. Ihre wichtigsten Methoden sind *synchronize()* und *readLog()* zur Ausführung des Synchronisationsprozesses bzw. des Leseprozesses der geloggten Daten. In *synchronize()* wird entschieden, ob es sich um einen Synchronisationsprozess oder um einen Initialisierung-Prozess handelt, je nachdem wird dann die Methode *initializeCC()* oder *updateCC()* aufgerufen. Diese beiden Methoden kapseln den Ablauf durch den jeweiligen Prozess. Desweiteren gibt es in *CCSynchronizer* Methoden, die die Name–Value Paare für die gesendeten HTTP-Requests zusammenbauen, die das Holen aus und das Speichern der Geräte in die Datenbank übernehmen, die Funktionalität zur Überprüfung der Gerätedaten kapseln, die die Ausführung der einzelnen HTTP-Requests steuern und deren Ergebnisse auswerten, sowie Methoden, die den Ablauf des Auslesens der geloggten Daten kapseln.

4.3.4 SMSEngine

In der Klasse SMSEngine ist die gesamte Funktionalität gekapselt, die benötigt wird um die am Modem gespeicherten SMS auszulesen. In der Methode *readSMS()* wird zuerst ein Skript ausgeführt, dass einen Prozess startet, der eine Liste der am Modem gespeicherten SMS zurückgibt. Aus dieser Liste werden die Indizes der SMS herausgelesen und dann wird die jeweilige SMS vom Modem abgerufen. Das geschieht mittels eines weiteren Skripts. Die verwendeten Skripts enthalten AT Befehle. Es wird überprüft, ob die ausgelesene SMS dem erwarteten Format entspricht, wenn ja, wird sie für die Datenbank aufbereitet (sie wird in einzelne Attribute zerteilt) und gespeichert. Dann wird ein weiteres Skript zum Löschen der verarbeiteten SMS ausgeführt.

4.3.5 CCDataLogger

In der Klasse *CCDataLogger* ist die notwendige Funktionalität zum Verarbeiten und Speichern der ausgelesenen Daten implementiert. Beim Instanziieren der Klasse wird überprüft, ob für das übergebene Gerät schon ein Logfile vorhanden ist und, ob dieses Logfile eine gewisse Größe übersteigt. Falls notwendig wird ein neues bzw. ein weiteres File für das Gerät angelegt. Es wird immer zuerst die Methode *getJunksToLog()* ausgeführt, in der aus den gesendeten Antworten des Communication Centers die relevanten Stücke, die die geloggten Daten enthalten, herausgefiltert und zwischengespeichert werden. Dann wird von der aufrufenden Klasse (*CCSynchronizer*) die Methode *writeJunksToLogFile()* ausgeführt, in der die gespeicherten Teile der Logdaten, in einzelne Zeilen zerteilt, sortiert und in das Logfile gespeichert werden. Die Schwierigkeit besteht darin, aus dem Logfile die letzte gespeicherte Zeile auszulesen und nur Daten mit einem aktuelleren Datum in das Logfile zu speichern. Deswegen ist das Sortieren der Zeilen so wichtig.

4.3.6 HTTPConnector

In der Klasse *HTTPConnector* gibt es Methoden, die zur Kommunikation mit dem Communication Center dienen. Unter anderem eine Methode zum Aufbau der Verbindung. Das geschieht mittels eines Skripts, welches AT Befehle enthält. Ist die Verbindung erfolgreich erstellt, so werden von der Klasse(meist *CCSynchronizer*), die über die Instanz von *HTTPConnector* verfügt, normalerweise unterschiedliche Methoden zum Senden von HTTP-Requests an das Communication Center aufgerufen. Welche Methoden aufgerufen werden, hängt von der Art des Prozesses ab. Am Ende eines Kommunikationsvorganges wird eine Methode zum Beenden der Verbindung ausgeführt. Diese Methode ist besonders wichtig, da der Zustand des Modems nach dem Verbindungsabbruch von ihrer erfolgreichen Ausführung abhängt und somit, in weiterer Folge, der Zustand des Systems. Neben den bereits angeführten Methoden gibt es in *HTTPConnector* noch Routinen, die für die Überprüfung der Antworten des Communication Centers zuständig sind.

4.3.7 Der Synchronisationsprozess

Das folgende Diagramm in Abbildung 19 zeigt den Ablauf eines Synchronisationsprozesses. Zur Vereinfachung wurden manche Methoden mit ähnlicher Funktionalität zu Gruppen zusammengefasst. Der Ablauf ist folgendermaßen: In der *run()* Methode der Klasse *CCEngineI* wird die Methode *synchronize()* des *CCSynchronizers* aufgerufen, die dann die Methode *updateCC()* ausführt. In *updateCC()* wird *getObjectData()* aufgerufen (hier wurden mehrere Methoden *getCommonData(), getMessageData, getPortData()* etc. zusammengefasst), die die Attribute für das zu synchronisierende Objekt aus der Datenbank holt und in einer internen Datenstruktur abspeichert. Als nächstes wird *checkObjects()* (eine Zusammenfassung von Methoden zur Überprüfung der Attribute) der Klasse *ErrorChecker* aufgerufen, danach die Methode *buildObjRequestData()*(eine Zusammenfassung von Methoden), in welcher die Name-Value Pairs für die Requests, die an das CC geschickt werden sollen, zusammengebaut werden. Dann wird eine Verbindung mit dem Communication Center über die Methode *connect()* von *HTTPConnector* erstellt. Hier beginnt die eigentliche Synchronisation mit dem Gerät. Dieser Prozess ist in der Methode *synchronizeObjData()* (abermals eine Zusammenfassung von Objekten) gekapselt, in der noch verschiede andere Methoden aufgerufen werden, die HTTP-Requests mit den zu ändernden Attributen an das Communication Center schicken. Diese Methoden sind in *setObjData()* zusammengefasst. *setObjData()* verwendet die Methode *submit()* von *HTTPRequester* um die HTTP-Requests an das Communication Center zu senden. Die Antworten des Communication Center werden in *transformResult()* transformiert und an die Methode *synchronizeObjData()* zurückgegeben, wo mit Hilfe der Methode *verifyResult()* überprüft wird, ob die Antworten des Communication Center ident mit den gesendeten HTTP-Requests sind (Nur wenn das der Fall ist, kann das Communication

Center als synchron mit seiner Repräsentation in der Applikation betrachtet werden). Ist der Synchronisationsvorgang abgeschlossen, wird in der Methode *updateCC()* die Methode *disconnect()* von *HTTPConnector* aufgerufen und die Verbindung mit dem Communication Center getrennt. Der Synchronisationsstatus des Gerätes (true oder false) wird an die *CCEnginel* zurückgegeben und in einem letzten Schritt wird das Gerät über die Methode *saveObjects()* (eine Zusammenfassung von Methoden) von *CCSynchronizer* gespeichert.

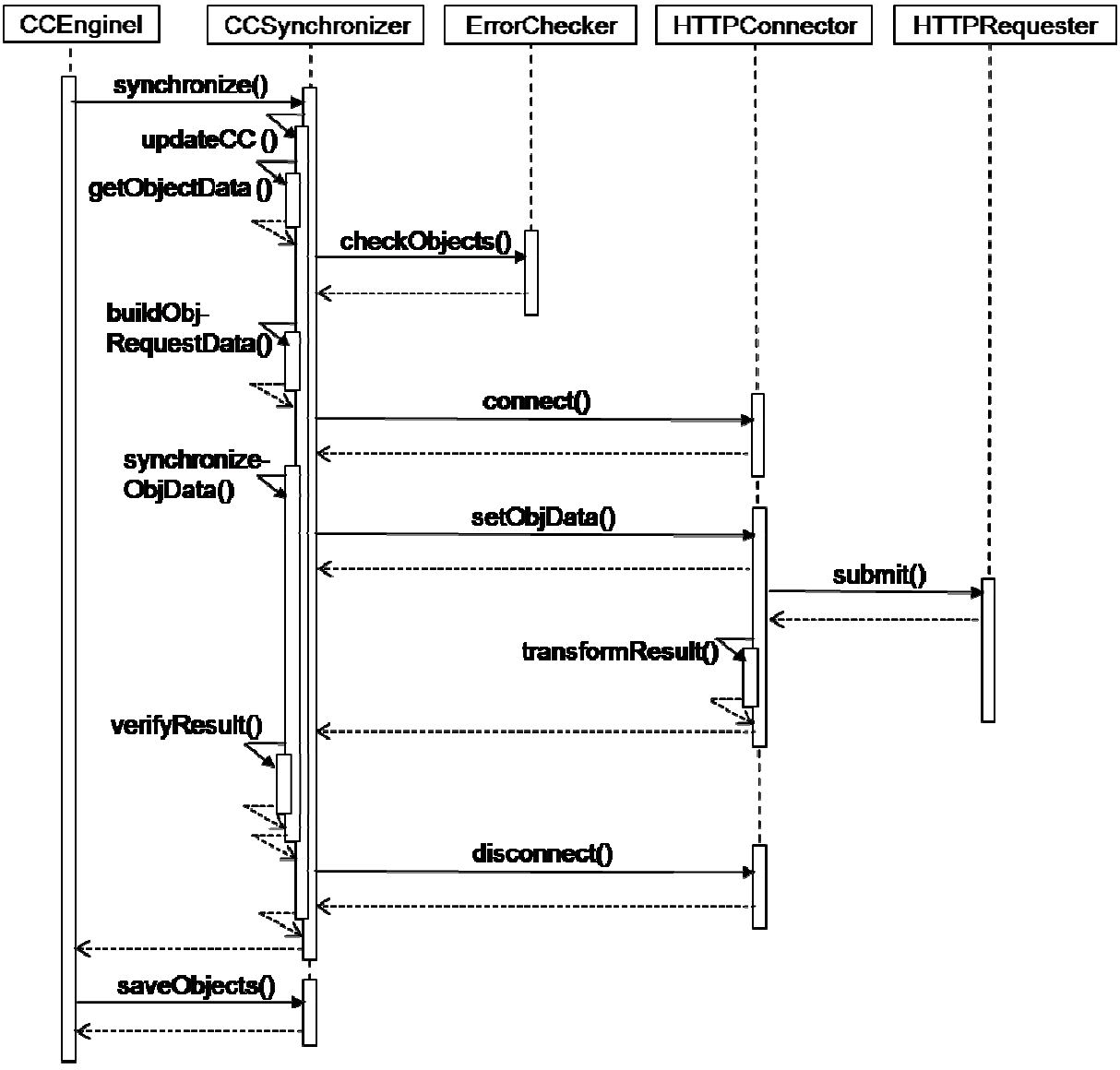


Abbildung 19: Sequenzdiagramm „Synchronisationsprozess“

4.3.8 Abrufen der geloggten Daten

Das folgende Diagramm in Abbildung 20 zeigt den Ablauf des Prozesses, der für das Auslesen der geloggten Daten am Communication Center zuständig ist.

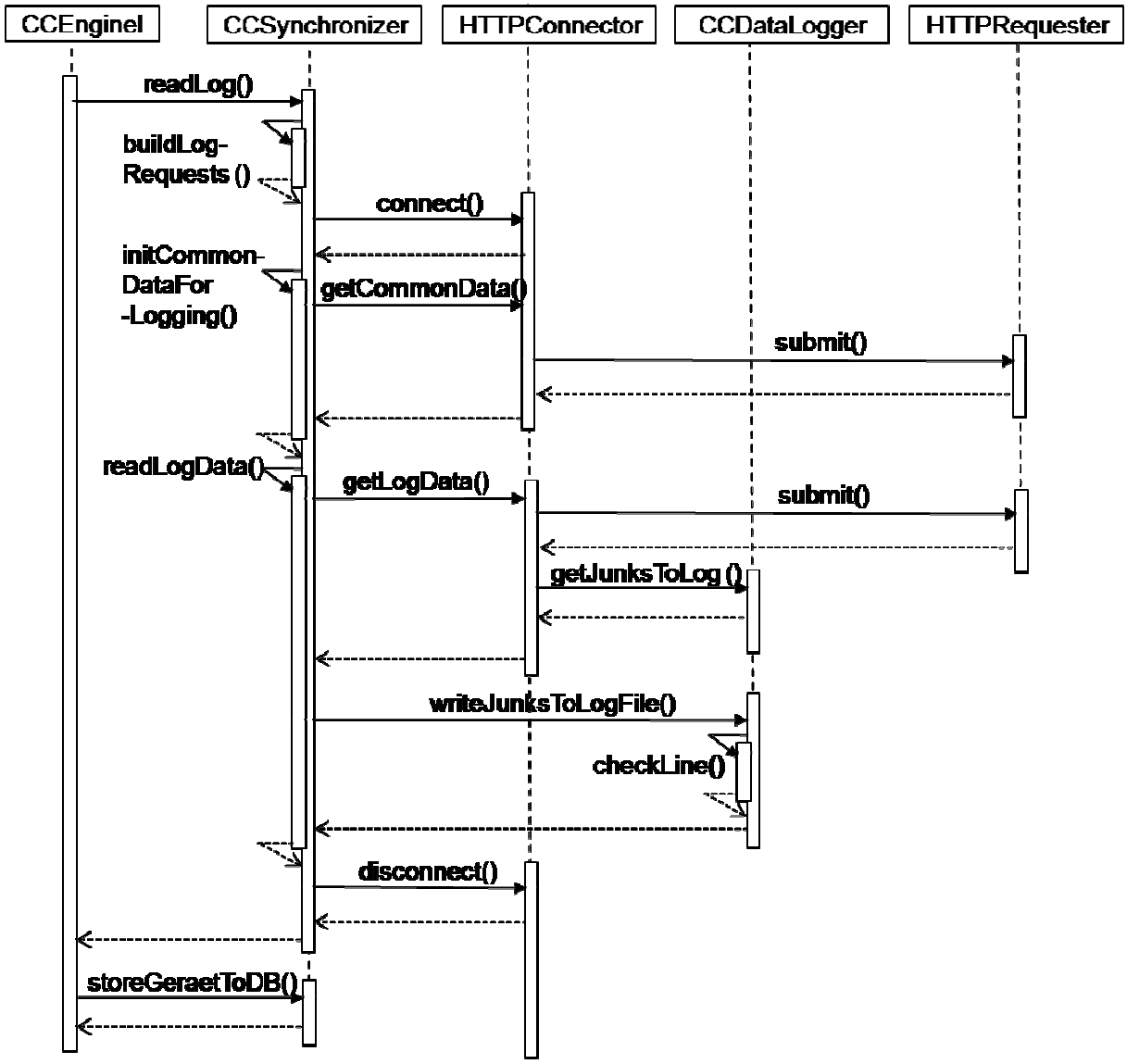


Abbildung 20: Sequenzdiagramm „Auslesen der geloggten Daten"

In der *run()* Methode der Klasse *CCEnginel* wird die Methode *readLog()* des *CCSynchronizers* aufgerufen. In *readLog()* wiederum wird die Methode *buildLogRequests()* ausgeführt. Diese Methode ist für den Zusammenbau der HTTPRequests zuständig, die an das Communication Center geschickt werden. Dann wird in *readLog()* über die Methode *connect()* des *HTTPConnectors* eine Verbindung mit dem Gerät erstellt, für das die Logdaten ausgelesen werden sollen. Als nächstes ruft *CCSynchronizer* die Methode *getCommonData()* des *HTTPConnectors* auf, der dann mittels *submit()* der Klasse *HTTPRequester* den notwendigen Request an das Communication Center schickt. Der gesendete Request bereitet das Communication Center auf das bevorstehende Datenlogging vor. Dann wird im *CCSynchronizer* die Methode *readLogData()* ausgeführt, der Aufruf dieser Methode startet den eigentlichen Prozess des Datenauslesens. Zuerst werden über die Methode *getLogData()* der Klasse *HTTPConnector* die am Ge-

rät gespeicherten Daten vollständig ausgelesen. Die notwendigen Requests werden wieder über die Methode *submit()* der Klasse *HTTPRequester* abgeschickt. Dann wird die Methode *getJunksToLog()* des CCDataLoggers aufgerufen, in der die ausgelesenen Daten ausgewertet und gesäubert werden. Logeinträge, die bereits im Logfile gespeichert wurden, werden eliminiert. In einem weiteren Schritt werden diese gesäuberten Daten in der Methode *writeJunkstoLogFile(),* die wieder aus *readLogData()* von *CCSynchronize*r aufgerufen wird, sortiert und mit der Methode *checkLine()* zeilenweise auf ihre Korrektheit überprüft. Dann werden die Logdaten zeilenweise in das Logfile geschrieben. Nach dem Speichern der Daten beendet die Instanz der Klasse *CCSynchronize*r die Verbindung mit dem Communication Center, in dem die Methode *disconnect()* aufgerufen wird und gibt das Ergebnis des Prozesses an *CCEngineI* zurück. In *CCEngineI* wird das Gerät für das die Daten geloggt wurden mit der Methode *storeGeraetToDB()* in die Datenbank gespeichert. Vor dem Speichern wird noch anhand des für das Gerät eingestellten Logintervalles der nächste Logzeitpunkt bestimmt.

4.4 CCMP Features

Dieser Abschnitt beschreibt die wichtigsten Features der CCMP Applikation. Die folgenden Funktionalitäten der Webapplikation bieten den eigentlichen Nutzen gegenüber einer Verwaltung des Communication Centers über ein Modem und eine DFÜ-Verbindung.

4.4.1 Initialisieren des Communication Centers

Der Benutzer kann das Gerät initialisieren, sobald er das Gerät in der Webapplikation angelegt hat und eine Telefonnummer eingeben hat. Er kann zwischen der Option „Initialisieren“ und „Initialisieren ohne Anruf“ wählen. Beim Initialisieren erstellt die CCMP Applikation eine Verbindung mit dem Communication Center und lässt sich die aktuelle Konfiguration des Geräts schicken. Diese Konfiguration wird dann für das angelegte Gerät abgespeichert. „Initialisieren ohne Anruf“ bedeutet, dass das Gerät eine Standardkonfiguration erhält, der Anruf zum Communication Center entfällt. Diese Funktionalität erlaubt dem Benutzer eine leichte Integration von bestehenden bereits konfigurierten Geräten in die Webapplikation. Er spart Zeit und vermeidet Tippfehler, da er nicht alles von Hand eingeben muss.

4.4.2 Synchronisieren des Communication Centers

Synchronisieren bedeutet das Einspielen einer Gerätekonfiguration aus der Webapplikation in ein Communication Center. Sobald der Benutzer die Daten eines Gerätes über die Webapplikation verändert und speichert, bekommt das Gerät den Status „Asynchron“ da nun die Möglichkeit besteht, das die Konfiguration von der des dazugehörigen Communication Center abweicht. Hat der Benutzer das Ändern der Konfiguration abgeschlossen, so kann er den Synchronisationsprozess starten. Die Applikation wählt sich nun über das Modem in das Communication Center ein, spielt die neue Konfiguration ein und überprüft ob das Gerät richtig konfiguriert wurde. Falls der Vorgang erfolgreich war, wird der Status des Geräts auf „Synchron“ gesetzt, sonst auf „Error“. Der Benutzer kann während des Synchronisationsvorganges andere Tätigkeiten erledigen.

4.4.3 Automatisiertes Datenlogging

Das automatisierte Datenlogging ist ein wesentliches Feature der Webapplikation, da es nur innerhalb der Applikation verfügbar ist. Der Benutzer hat die Möglichkeit für ein Gerät ein bestimmtes Logintervall einzustellen, d.h. die Zeitspanne die zwischen zwei Logvorgängen vergehen darf. Das Logintervall ist in Stunden anzugeben, das Minimum ist stündlich. Sobald er das Logintervall eingestellt hat, braucht sich der Anwender nicht mehr um den Logvorgang zu kümmern, dieser wird von der Applikation durchgeführt. Auch der nächste Logzeitpunkt wird nach einem Logvorgang automatisch berechnet. Dieses Feature garantiert eine lückenlose Aufzeichnung und vor allem Archivierung der geloggten Daten. Der Benutzer kann über die Logfiles suchen, beliebige zeitliche Einschränkungen machen und sich die gewünschten

Ergebnisse in ein CSV File exportieren lassen. Das CSV Format erlaubt ihm, in Excel statistische Auswertungen über die Daten zu legen.

4.4.4 SMS Archivierung

Die SMS Archivierung erlaubt es, vom Communication Center versendete Meldungen über das Webinterface zu suchen und anzusehen. Die Meldungen werden beliebig lange gespeichert, was den Vorteil hat, dass der Benutzer nachprüfen kann, ob ein Gerät zu einem bestimmten Ereignis, auch lange nachdem das Ereignis eingetreten ist, eine Meldung geschickt hat oder nicht. Nicht benötigte SMS können gelöscht werden.

4.4.5 Kopieren von Geräten

Das Kopieren von Geräten erspart dem Benutzer die Eingabe von identischen bzw. ähnlichen Konfigurationen. Er kopiert einfach das Gerät auf der Webplattform, ändert die Telefonnummer und kann den Synchronisationsvorgang für das kopierte Gerät starten. Die Konfiguration wird dann in das Communication Center mit der angegebenen Nummer eingespielt. Das verkürzt den Zeitaufwand für die Konfiguration mehrerer gleich eingestellter Geräte, verringert die Wahrscheinlichkeit für mögliche Fehleingaben und erspart dem Anwender die wiederholte Eingabe der gleichen Gerätekonfiguration.

4.4.6 PDF Export einer Geräte Konfiguration

Falls ein Techniker vor Ort etwas an einem Communication Center einstellen oder es reparieren muss, so kann es von großem Nutzen sein, wenn er die momentanen Einstellungen des Gerätes kennt. Da es nicht immer einfach ist, sich direkt mit dem Gerät per Laptop zu verbinden, wurde ein PDF Export für Gerätekonfigurationen implementiert. Die gesamte Gerätekonfiguration wird auf Anfrage in ein PDF geschrieben und dort übersichtlich dargestellt. Das PDF steht entweder zum Download zur Verfügung oder kann direkt aus der Applikation heraus gedruckt werden.

5 Testen

Der erste Abschnitt dieses Kapitels beschreibt und definiert als Einleitung die Tätigkeit des Software Testens. Anschließend werden im zweiten Abschnitt einige Grundprinzipien zur erfolgreichen Durchführung von Software Tests erläutert. Als Grundlage dafür wurde das Buch „Methodisches Testen von Programmen“ von Glenford J. Myers [vgl. Myer91, Seite 1,15] herangezogen.

Testen von Software wird von Myers als eine kreative, kostenerzeugende und vor allem destruktive Tätigkeit beschrieben. Das Ziel des Testens ist die Werterhöhung des getesteten Programms durch das Auffinden von Fehlern. Diese Werterhöhung soll wiederum die Kosten, die durch das Testen entstehen, decken bzw. übersteigen. Daraus kann man den Schluss ziehen, dass Definitionen wie *„Testen ist der Prozess, der zeigen soll, dass keine Fehler vorhanden sind“* [Myer91, Seite 3] oder *„Der Zweck des Testens ist es zu zeigen, dass ein Programm die geforderten Funktionen korrekt ausführt“* [Myer91, Seite 3], nicht zulässig sind, da sie das Testen als Tätigkeit beschreiben, die offensichtlich nur Kosten erzeugt, aber nicht werterhöhend für die Software ist. Testen lässt sich daher wie folgt definieren:

„Testen ist der Prozess, ein Programm mit der Absicht auszuführen, Fehler zu finden“ [Myer91, Seite 4].

Die Absicht, das Programm auszuführen um Fehler zu finden, zeigt die destruktive Natur die der Arbeit des Software Testers zugrunde liegt und darum ist es sehr wichtig, eine geeignete Person dafür auszuwählen. Die meisten Menschen halten diese Tätigkeit für schwierig, da sie eine konstruktive Einstellung zu ihrem Beruf haben. Für die Planung und Durchführung von Tests mit dem Zweck, Fehler zu finden, ist allerdings eine destruktive Einstellung erforderlich: Es gilt zu zerstören was andere geschaffen haben. Myers führt folgende Prinzipien des Testens an:

Für einen Testfall sollte das Resultats bzw. die erwarteten Werten definiert werden.

Der Programmierer bzw. die Programmierorganisation sollte nicht sein/ihr eigenes Programm testen.

Testergebnisse sollten gründlich überprüft werden.

Testfälle sollten sowohl für ungültige/unerwartete als auch für gültige/erwartete Eingabedaten definiert werden.

Es sollte sowohl getestet werden ob ein Programm nicht tut, was es tun soll als auch ob das Programm tut, was es nicht tun soll.

Wegwerftestfälle sollten vermieden werden.

Kein Testverfahren sollte unter der Annahme geplant werden, dass keine Fehler gefunden werden.

Testen ist eine hoch kreative und intellektuell herausfordernde Tätigkeit

Diese Prinzipien sollten unbedingt befolgt werden, da sie die Grundlage für erfolgreiche Tests sind.

Der Aufgabe des Testens wurde im CCMP Projekt aus zwei Gründen ein besonderer Stellenwert zugewiesen: Die Entwicklung von CCMP hat sich zum Teil auf technischem Neuland abgespielt und allein daraus hat sich die Notwendigkeit ergeben, verstärkt zu testen, da manche Teile der Entwicklung erst durch Tests Schritt für Schritt erschlossen werden konnten. Das betrifft vor allem die Kommunikation zwischen der Applikation und den Communication Centern. Der zweite Faktor, der ein intensives und systematisches Testen notwendig gemacht hat, hat sich aus einer der Kernfunktionalitäten des Communication Center, der Überwachung von Objekten, ergeben. Um eine zufriedenstellende und sichere Überwachung gewährleisten zu können, auf die sich der Anwender verlassen kann, muss sichergestellt sein, dass die Applikation in seiner Grundfunktionalität fehlerfrei läuft.

Deswegen wurde im Rahmen der Entwicklung von CCMP mit dem Testen bereits nach Abschluss des ersten Projektdrittels begonnen. Zu diesem Zeitpunkt wurde das Webinterface getestet. Im weiteren Projektverlauf wurden Tests für das Modem und die Datenverbindung, den Ausfall von Hardwarekomponenten und den Einsatz von SIM Karten unterschiedlicher Anbieter durchgeführt. Außerdem wurden automatisierte Tests erstellt. um die Grundfunktionalität des Systems zu überprüfen.

5.1 Testen des Webinterfaces

Mit dem Testen des Webinterfaces wurde nach etwa 6 Monaten begonnen. Hierfür wurden Black-Box-Tests herangezogen. Black-Box-Tests sind Tests, bei denen das Programm als für den Tester nicht einsichtiges System betrachtet wird und bei dem jene Situationen untersucht werden sollen, in denen das Programm von seiner Spezifikation abweicht. Das Interesse an der Programmstruktur und am inneren Programmverhalten rückt in den Hintergrund. Man nennt diese Tests auch Ein-/Ausgabe-Tests, wobei die Testdaten alleine aus der Programmspezifikation abgeleitet werden. Im Prinzip müsste man einen vollständigen Eingabe Test durchführen um alle möglichen Fehler zu finden. [vgl. Myer91,Seite 6,8]

Das ist in der Praxis natürlich unmöglich, da man bereits für ein vierstelliges Eingabefeld 10000 Testfälle erstellen müsste. Zuser [vgl. Zuse01, Seite 198,203] schlägt folgendes Verfahren vor: Die Eingabedaten werden in Äquivalenzklassen unterteilt, wobei eine Klasse mit solchen Werten gefüllt wird, die ein gleiches Systemverhalten hervorrufen. Es werden sowohl gültige als auch ungültige Eingabedaten herangezogen. Für jede der Äquivalenzklassen wird ein Testfall formuliert, der die korrekte Funktionsweise des Systems für die Eingabe eines Wertes dieser Klasse überprüfen soll. Ein Testfall hat eine Nummer, eine Klasse, die entweder ein Normalfall (eine laut Spezifikation gültige Eingabe) oder ein Fehlerfall (eine laut Spezifikation fehlerhafte Eingabe) ist, eine Beschreibung, ein erwartetes Ergebnis sowie eine Beispiel-Eingabe. Nach der kompletten Auflistung der Testfälle folgt eine kurze Beschreibung und Auswertung der Testergebnisse.

5.1.1 Gerät anlegen

Nr.	Klasse	Beschreibung des Testfalls	Erwartetes Ergebnis	Beispielhafte Eingabe
1	NF	Gültige Telefonnummer eingegeben	Eingabe wird akzeptiert	Telefonnummer = '+436768834552' od. Telefonnummer = '06768834552'
2	FF	Telefonnummer ist leer	Fehlermeldung	Telefonnummer = ''
3	FF	Telefonnummer beginnt mit Zahl > 0	Fehlermeldung	Telefonnummer = '6768834552'
4	FF	Telefonnummer beginnt nicht mit Zahl und nicht mit Zeichen = '+'	Fehlermeldung	Telefonnummer ='-436768834552'
5	FF	Telefonnummer ist länger als 19 Zeichen	Fehlermeldung	Telefonnummer ='+43676883455212345671'
6	FF	Telefonnummer ist kürzer als 10 Zeichen	Fehlermeldung	Telefonnummer ='067612345'
7	NF	Log-Intervall ist größer als 24	Eingabe wird akzeptiert	Log-Intervall ='168'
8	FF	Log-Intervall ist kleiner als 24	Fehlermeldung	Log-Intervall ='12'
9	FF	Log-Intervall ist negativ	Fehlermeldung	Log-Intervall ='-24'
10	FF	Log-Intervall ist keine Zahl	Fehlermeldung	Log-Intervall ='168h'

11	SF	Log-Intervall ist leer	Eingabe wird akzeptiert (Logging deaktiviert)	Log-Intervall =''
12	SF	Log-Intervall ist 0	Eingabe wird akzeptiert (Logging deaktiviert)	Log-Intervall ='0'
13	SF	Log-Intervall ist 24	Eingabe wird akzeptiert (Logging täglich)	Log-Intervall ='24'

Tabelle 3: Testfall - "Gerät anlegen"

5.1.2 Empfänger anlegen/ändern

Nr.	Klasse	Beschreibung des Testfalls	Erwartetes Ergebnis	Beispielhafte Eingabe
1	NF	Gültiger Nachname eingegeben (bel. String)	Eingabe wird akzeptiert	Nachname = 'Schustermeier'
2	SF	Nachname ist leer	Eingabe wird akzeptiert	Nachname = ''
3	FF	Nachname ist länger als 30 Zeichen	Fehlermeldung	Nachname = 'Schustermeiermüller Franciscus der erste'
4	NF	Gültiger Vorname eingegeben (bel. String)	Eingabe wird akzeptiert	Vorname = 'Franciscus'
5	SF	Vorname ist leer	Eingabe wird akzeptiert	Vorname = ''
6	FF	Vorname ist länger als 30 Zeichen	Fehlermeldung	Vorname = ' Franciscus Schustermeiermüller der Erste'
7	NF	Gültige Telefonnummer eingegeben	Eingabe wird akzeptiert	Telefonnummer = '+436768834552' od. Telefonnummer = '06768834552'
8	SF	Telefonnummer ist leer	Eingabe wird akzeptiert	Telefonnummer = ''
9	FF	Telefonnummer beginnt mit Zahl > 0	Fehlermeldung	Telefonnummer = '6768834552'
10	NF	Gültige Email Adresse eingegeben	Eingabe wird akzeptiert	Email ='kutte@aon.at'
11	SF	Email Adresse ist leer	Eingabe wird akzeptiert	Email = ''
12	FF	Email Adresse beinhaltet kein '@'	Fehlermeldung	Telefonnummer ='kutteaon.at'
13	FF	Email Adresse beinhaltet keinen'.'	Fehlermeldung	Telefonnummer ='kutte@aonat'
14	FF	Telefonnummer und Email Adresse sind leer	Fehlermeldung	Telefonnummer ='' Email=''

Tabelle 4: Testfall - "Empfänger anlegen/ändern"

5.1.3 Gerät ändern

Nr.	Klasse	Beschreibung des Testfalls	Erwartetes Ergebnis	Beispielhafte Eingabe
1	NF	Gerätename enthält nur Zeichen aus Text 1	Eingabe wird akzeptiert	Gerätename = 'Serverraum Metamagix'
2	NF	Gerätename ist leer	Eingabe wird akzeptiert	Gerätename = ''
3	FF	Gerätename enthält ein oder mehrere Zeichen, die nicht aus Text 2 sind	Fehlermeldung	Gerätename = 'Überwachung des Serverraumes'
4	FF	Gerätename ist länger als 49	Fehlermeldung	Gerätename = 'Serverraum

		Zeichen		Metamagix in der Hackengasse 27/6 1150 Wien’
5	NF	Gültige Telefonnummer eingegeben	Eingabe wird akzeptiert	Telefonnummer = ’+436768834552’ od. Telefonnummer = ’06768834552’
6	FF	Telefonnummer ist leer	Fehlermeldung	Telefonnummer = ’’
7	FF	Telefonnummer beginnt mit Zahl > 0	Fehlermeldung	Telefonnummer = ’6768834552’
8	NF	Zwischen 1 und 8 Empfänger zuordnen	Eingabe wird akzeptiert	-
9	SF	0 Empfänger zuordnen	Eingabe wird akzeptiert	-
10	FF	Mehr als 8 SMS Empfänger zuordnen	Fehlermeldung	-
11	NF	Sicherheitscode ist eine 4-stellige Zahl	Eingabe wird akzeptiert	Sicherheitscode =’1234’
12	NF	Sicherheitscode ist leer	Eingabe wird akzeptiert	Sicherheitscode =’’
13	FF	Sicherheitscode ist keine 4-stellige Zahl	Fehlermeldung	Sicherheitscode =’12345’
14	NF	Subnet Mask ist eine gültige IP- Adresse	Eingabe wird akzeptiert	Subnet Mask=’ 255.255.255.0’
15	NF	Subnet Mask ist leer	Eingabe wird akzeptiert	Subnet Mask=’’
16	FF	Subnet Mask ist keine gültige IP- Adresse	Fehlermeldung	Subnet Mask=’ 999.995.996.0’
17	NF	IP Adresse ist eine gültige IP-Adresse	Eingabe wird akzeptiert	IP Adresse=’ 255.255.255.0’
18	SF	IP Adresse ist leer	Eingabe wird akzeptiert	IP Adresse=’’
19	FF	IP Adresseist keine gültige IP-Adresse	Fehlermeldung	IP Adresse=’ 255.33D.9:0’
20	NF	Anzahl der Prioritätsmeldungen ist eine Zahl zwischen 0 und 18	Eingabe wird akzeptiert	Anzahl der Prioritätsmeldungen =’5’
21	NF	Anzahl der Prioritätsmeldungen ist leer	Eingabe wird akzeptiert	Anzahl der Prioritätsmeldungen =’’
22	FF	Anzahl der Prioritätsmeldungen ist keine Zahl zwischen 0 und 18	Fehlermeldung	Anzahl der Prioritätsmeldungen =’50’

Tabelle 5: Testfall - "Gerät ändern"

5.1.4 I/O ändern

Nr.	Klasse	Beschreibung des Testfalls	Erwartetes Ergebnis	Beispielhafte Eingabe
1	NF	Portname enthält nur Zeichen aus Text 2	Eingabe wird akzeptiert	Name = ’Port1’
2	FF	Portname enthält ein oder mehrere Zeichen die nicht in Text 2 enthalten sind	Fehlermeldung	Name = ’Port 1!’
3	SF	Portname ist leer	Eingabe wird akzeptiert	Name = ’’
4	FF	Portname hat mehr als 9 Zeichen	Fehlermeldung	Name = ’Port_Temp_1’
5	NF	Untergrenze ist eine Zahl, die der Definition von Zahl entspricht	Eingabe wird akzeptiert	Untergrenze= ’51.12’
6	FF	Untergrenze ist eine Zeichenkette unterschiedlich von der Definition in Zahl	Fehlermeldung	Untergrenze= ’3351.123’

7	SF	Untergrenze ist leer	Eingabe wird akzeptiert	Untergrenze=''
8	SF	Untergrenze ist eine Zahl kleiner -99999	Fehlermeldung	Untergrenze='-100000'
9	SF	Untergrenze ist eine Zahl größer 99999	Fehlermeldung	Untergrenze='100000'
10	NF	Obergrenze ist eine Zahl, die der Definition von Zahl entspricht	Eingabe wird akzeptiert	Obergrenze= '51.12'
11	FF	Obergrenze ist eine Zeichenkette unterschiedlich von der Definition in Zahl	Fehlermeldung	Obergrenze= '3351.123'
12	SF	Obergrenze ist leer	Eingabe wird akzeptiert	Obergrenze=''
13	SF	Obergrenze ist eine Zahl kleiner -99999	Fehlermeldung	Obergrenze='-100000'
14	SF	Obergrenze ist eine Zahl größer 99999	Fehlermeldung	Obergrenze=' 100000'
15	NF	Einheit enthält nur Zeichen aus Text 2	Eingabe wird akzeptiert	Einheit = 'Grad'
16	FF	Einheit enthält ein oder mehrere Zeichen die nicht in Text 2 enthalten sind	Fehlermeldung	Einheit = 'Grad !'
17	SF	Einheit ist leer	Eingabe wird akzeptiert	Einheit = ''
18	FF	Einheit hat mehr als 9 Zeichen	Fehlermeldung	Name = 'Fahrenheit'

Tabelle 6: Testfall - "I/O ändern"

5.1.5 Meldung ändern

Nr.	Klasse	Beschreibung des Testfalls	Erwartetes Ergebnis	Beispielhafte Eingabe
1	NF	Triggerbedingung Wert ist eine Zahl, die der Definition von Zahl entspricht	Eingabe wird akzeptiert	Triggerbedingung Wert = '90.3'
2	FF	Triggerbedingung Wert ist eine Zeichenkette unterschiedlich von der Definition in Zahl	Fehlermeldung	Triggerbedingung Wert = '90,4342'
3	SF	Triggerbedingung Wert ist leer	Eingabe wird akzeptiert	Triggerbedingung Wert =''
4	SF	Triggerbedingung Wert ist eine Zahl kleiner -99999	Fehlermeldung	Triggerbedingung Wert ='-100000'
5	SF	Triggerbedingung Wert ist eine Zahl größer 99999	Fehlermeldung	Triggerbedingung Wert ='100000'
6	SF	Mindestsignaldauer Meldungsauslösung ist leer	Eingabe wird akzeptiert	Mindestsignaldauer Meldungsauslösung =''
7	SF	Mindestsignaldauer Meldungsauslösung ist eine Zahl kleiner 0	Fehlermeldung	Mindestsignaldauer Meldungsauslösung ='-5'
8	SF	Mindestsignaldauer Meldungsauslösung ist eine Zahl größer 16000	Fehlermeldung	Mindestsignaldauer Meldungsauslösung ='17000'
9	FF	Mindestsignaldauer Meldungsrücksetzung ist eine Zeichenkette unterschiedlich von der Definition in Zahl	Fehlermeldung	Mindestsignaldauer Meldungsrücksetzung = '15.5555'
10	SF	Mindestsignaldauer Meldungsrücksetzung ist leer	Eingabe wird akzeptiert	Mindestsignaldauer Meldungsrücksetzung =''
11	SF	Mindestsignaldauer Meldungsrücksetzung ist eine Zahl kleiner 0	Fehlermeldung	Mindestsignaldauer Meldungsrücksetzung ='-5'
12	SF	Mindestsignaldauer Meldungsrücksetzung ist eine Zahl größer	Fehlermeldung	Mindestsignaldauer Meldungsrücksetzung ='17000'

		16000		
13	NF	Meldungstext enthält nur Zeichen aus Text 1	Eingabe wird akzeptiert	Name = 'Die Blumen frieren ab'
14	FF	Meldungstext enthält ein oder mehrere Zeichen die nicht in Text 1 enthalten sind	Fehlermeldung	Name = 'Die Blümchen frieren ab!'
15	SF	Meldungstext ist leer	Eingabe wird akzeptiert	Name = ''
16	FF	Meldungstext hat mehr als 48 Zeichen	Fehlermeldung	Name = 'Das ist eine Meldung mit mehr als 48 Zeichen. Das wird nicht akzeptiert'
17	NF	Zwischen 1 und 8 SMS Empfänger zuordnen	Eingabe wird akzeptiert	-
18	SF	0 SMS Empfänger zuordnen	Eingabe wird akzeptiert	-
19	FF	Mehr als 8 SMS Empfänger zuordnen	Fehlermeldung	-
20	NF	Zwischen 1 und 8 Email Empfänger zuordnen	Eingabe wird akzeptiert	-
21	SF	0 Email Empfänger zuordnen	Eingabe wird akzeptiert	-
22	FF	Mehr als 8 Email Empfänger zuordnen	Fehlermeldung	-

Tabelle 7: Testfall - "Meldung ändern"

5.1.6 SMS Archiv durchsuchen

Nr.	Klasse	Beschreibung des Testfalls	Erwartetes Ergebnis	Beispielhafte Eingabe
1	NF	Gerät auswählen	Liefert alle SMS für das ausgewählte Gerät	Gerät = 'Gerät 1'
2	NF	Kein Gerät ausgewählt	Liefert alle archivierten SMS	Gerät = 'keine'
3	NF	Gesendet am „von" ist ein gültiges Datum mit Uhrzeit	Liefert alle archivierten SMS von dem angegebenen Datum weg.	Gesendet am „von" = ' 11.11.2006 11:50'
4	NF	Gesendet am „bis" ist ein gültiges Datum mit Uhrzeit	Liefert alle archivierten SMS bis zu dem angegebenen Datum.	Gesendet am „bis" = ' 11.12.2006 11:50'
5	NF	Gesendet am „von" und Gesendet am „bis" sind ein gültiges Datum mit Uhrzeit	Liefert alle archivierten SMS von einem angegebenen Datum bis zu einem angegebenen Datum	Gesendet am „von" = ' 11.11.2006 11:50' Gesendet am „bis" = ' 11.12.2006 11:50'
6	NF	Gerät auswählen und Gesendet am „von" und Gesendet am „bis" sind ein gültiges Datum mit Uhrzeit	Liefert alle SMS für das ausgewählte Gerät von einem angegebenen Datum bis zu einem angegebenen Datum	Gerät = 'Gerät 1' Gesendet am „von" = ' 11.11.2006 11:50' Gesendet am „bis" = ' 11.12.2006 11:50'
7	NF	Kein Gerät ausgewählt und Gesendet am „von" und Gesendet am „bis" sind ein gültiges Datum mit Uhrzeit	Liefert alle archivierten SMS von einem angegebenen Datum bis zu einem angegebenen Datum.	Gerät = 'keine' Gesendet am „von" = ' 11.11.2006 11:50'

				Gesendet am „bis“ = ’ 11.12.2006 11:50’

Tabelle 8: Testfall - "SMS Archiv durchsuchen"

5.1.7 Datenlogs suchen

Nr.	Klasse	Beschreibung des Testfalls	Erwartetes Ergebnis	Beispielhafte Eingabe
1	NF	Gerät auswählen	Liefert ein Excel File mit allen Logdaten für das Gerät aus dem aktuellen Logfile.	Gerät = ’Gerät 1’
2	NF	Geloggt am „von“ ist ein gültiges Datum mit Uhrzeit	Liefert ein Excel File mit allen Logdaten für ein Gerät von dem angegebenen Datum weg	Geloggt am „von“ = ’ 11.11.2006 11:50’
3	NF	Geloggt am „bis“ ist ein gültiges Datum mit Uhrzeit	Liefert ein Excel File mit allen Logdaten für ein Gerät bis zu dem angegebenen Datum.	Geloggt am „bis“ = ’ 11.12.2006 11:50’
4	FF	Geloggt am „bis“ ist ein älteres Datum als Geloggt am „von“	Fehlermeldung	Geloggt am „von“ = ’ 12.12.2006 11:50’ Geloggt am „bis“ = ’ 11.11.2005 11:50’
5	NF	Geloggt am „von“ und Geloggt am „bis“ sind gültige Datumswerte mit Uhrzeit	Liefert ein Excel File mit allen Logdaten für ein Gerät von einem angegebenen Datum bis zu einem angegeben Datum.	Geloggt am „von“ = 11.11.2005 11:50’ Geloggt am „bis“ = ’ ’12.12.2006 11:50’
6	NF	Geloggt am „von“ und/oder Geloggt am „bis“ sind ungültige Datumswerte mit Uhrzeit	Liefert ein Excel File mit allen Logdaten für das Gerät aus dem aktuellen Logfile.	Geloggt am „von“ = 11/11-2005 11:50’ Geloggt am „bis“ = ’ ’12.12.Dgfg 11:50’

Tabelle 9: Testfall - "Datenlogs suchen"

5.1.8 Ergebnisse

Das Testen des Webinterfaces anhand der Testfälle wurde insgesamt drei Mal durchgeführt. Nach jedem Durchgang wurden die Ergebnisse dem Entwickler übergeben, dessen Aufgabe es war, die entdeckten Fehler zu beheben. Vor dem ersten Testlauf war keine Überprüfung der Eingabe auf Ebene des Webinterfaces implementiert, es wurden insgesamt 22 Fehler gefunden. Die Eingabedaten wurden erst vor dem Speichern in die Datenbank überprüft. Nach diesem Durchlauf wurden Javascript-Funktionen programmiert, welche die Eingabedaten bereits im Webinterface validieren. In einem zweiten Durchlauf konnten anhand der Testfälle nur vier Fehler entdeckt werden, die nach einer Überarbeitung des Validierungsprogramms im dritten Testlauf vollständig behoben waren. Die Ergebnisse sind im Detail in Tabelle 10 dargestellt. Es werden die gefunden Fehler pro Testlauf für jeden Testfall angezeigt. In der letzten Zeile der Tabelle ist die Summe der Fehler pro Testdurchlauf zu sehen.

	1.Testlauf	2.Testlauf	3.Testlauf
Gerät anlegen	4	0	0
Empfänger anlegen/ändern	0	0	0
Gerät ändern	5	1	0
I/O ändern	3	1	0
Meldung ändern	7	2	0
SMS Archiv durchsuchen	2	0	0
Datenlogs suchen	2	0	0
Gesamt	**22**	**4**	**0**

Tabelle 10: Auswertung Testfälle Webinterface

5.2 Testen des Modems und der Datenverbindung

Da die automatisierte Kommunikation mit dem Communication Center den Kern der CCMP Applikation darstellt, war es wichtig, das Modem und die Datenverbindung eingehend zu testen. Desweiteren wurden Tests von Optimierungsmaßnahmen der Verbindungszeiten durchgeführt.

5.2.1 Testen des Modems unter hoher Belastung

Für CCMP wurde ein Belastungstest des Modems durchgeführt. Ziel war es herauszufinden, wie das Modem auf eine erhöhte Belastung reagiert und ob das System unter solchen Umständen stabil bleibt. In diesem Fall wurde simuliert, dass drei unterschiedliche Benutzer, zur gleichen Zeit im System eingeloggt sind und jeder dieser Benutzer drei Geräte synchronisiert, bzw. von einem Gerät Daten loggt. Dieses Szenario stellt eine im Echtbetrieb fast unrealistische Belastung dar, da die Synchronisation von Geräten ein sehr selten ausgeführter Prozess ist und maximal 80 Geräte pro Server verwaltet werden sollen. Der Test wurde insgesamt zwei Mal durchgeführt, beide Male war der Testlauf fehlerfrei. Auffällig war aber, dass pro Testlauf eine Stunde benötigt wurde, woraus folgt, dass die Benutzer ebenso lange auf ihre Ergebnisse warten mussten. Dieses Problem ist grundsätzlich nicht zu lösen, da das Modem eine Art Engpass darstellt: Es kann immer nur ein Gerät nach dem anderen synchronisiert werden. Um die Wartezeiten zu reduzieren, wurde, wie in 5.2.2 beschrieben, versucht die Verbindungszeiten zu reduzieren.

5.2.2 Optimierung der Verbindungzeiten

Die Optimierung der Verbindungszeiten war ein aus Kostengründen bzw. unter dem Aspekt einer erhöhten Benutzerfreundlichkeit wichtiges Kriterium. Die Ansatzpunkte für diese Problemstellung waren die, nach jedem, vom Modem an das Communication Center gesendeten Request, eingebauten Timeouts. Diese Timeouts sind notwendig, da das Communication Center nach einem Request eine gewisse Zeitspanne (die unbekannt ist) benötigt, bis es für den nächsten Request bereit ist. Bei einem Synchronisations-/Initialisierungsprozess eines Communication Centers werden je nach Modell 41 bzw. 43 Requests geschickt; beim Datenloggen hängt die Anzahl der Requests von der vorhandenen Datenmenge ab. Die Timeouts wurden anfangs auf 5 Sekunden gesetzt. Das ergibt eine reine Timeoutzeit von 205/215 Sekunden pro Gerät und Synchronisationsprozess.

Das Optimieren der Verbindungszeiten wurde gesondert für das Datenloggen und das Synchronisieren/Initialisieren der Geräte durchgeführt, da diese in beiden Fällen unterschiedlich lange brauchen bis sie für den nächsten Request bereit sind. Es wurde folgende Methode angewandt: Für jeden Testdurchlauf wurden die Timeoutzeiten um 0.5 Sekunden heruntergesetzt, dann wurden zwei Geräte (verschiedene Modelle) synchronisiert/initialisiert bzw. es

wurden für diese Geräte die Daten geloggt. Falls keine Fehler aufgetreten sind wurde das Timeout weiter reduziert und der nächste Testlauf durchgeführt. Die Ergebnisse sind in Tabelle 11 zu sehen. Durch die Optimierung konnten bis zu 70 % Timeoutzeit beim Synchronisationsprozess und 90% beim Datenloggen eingespart werden.

	Synchronisation/Initialisierung	Datenloggen
Timeout vor der Optimierung	5	5
Timeout nach der Optimierung	1.5	0.5
Einsparung pro Request in s	3.5	4.5
Einsparung pro Request in %	70 %	90 %
Einsparung gesamt in s	143.5/150.5	variable
Prozessdauer vor der Optimierung	205/215	variable
Prozessdauer nach der Optimierung	61.5/64.5	variable

Tabelle 11: Optimierung der Verbindungszeiten - Auswertung

5.3 SIM Karten Tests

Im Laufe des Projektes wurden SIM Karten von verschiedenen Anbietern getestet, da es wichtig war herauszufinden, durch welchen Anbieter die zuverlässigste Datenübertragung erreicht werden kann. Auf der einen Seite wurden die verschiedenen SIM Karten serverseitig in das Modem eingesetzt und auf der anderen Seite wurden die Geräte mit verschiedenen SIM Karten getestet.

5.3.1 Verschiedene SIM Karten im Modem

Die Zuverlässigkeit der SIM Karte, die auf Serverseite im Modem eingesetzt wird, hat eine hohe Priorität, da es von ihr abhängt ob die Kommunikation mit den Geräten einwandfrei funktioniert oder nicht. Die SIM Karten wurden über einen Zeitraum von 10 Tagen zu verschiedenen Zeitpunkten(8:00, 12:00, 16:00, 20:00, 24:00) getestet. Eine Zuverlässigkeit von „sehr gut“ bedeutet, dass während dieser 10 Tage nicht mehr als 1 Verbindungsfehler auf Seitenn des Modems aufgetreten ist. Bei mehr als 5 Fehlern wurde die SIM Karte, bzw. das Mobilnetz des Betreibers für den Zweck der Applikation, als unzuverlässig eingestuft. Die durchgeführten Tests haben ergeben (siehe Tabelle 12), dass sowohl die SIM Karte von T-Mobile (0 Fehler) als auch A1 (0 Fehler) zu 100 % zuverlässig ist, allerdings hängt die Zuverlässigkeit der Karten stark von den an das Modem übergebenen Konfigurationsparametern ab und es ist nicht möglich eine T-Mobile SIM Karte mit der gleichen Konfiguration wie eine A1 SIM Karte (und umgekehrt) zu betreiben. Das bedeutet, dass das Modem auf die jeweilige SIM Karte eingestellt werden muss. Telering (9 Fehler) hat sich tageweise als unzuverlässig erwiesen und konnte daher für den Einsatz im Echtbetrieb nicht in Betracht gezogen werden. Bei ONE war es überhaupt nicht möglich über das Modem eine Verbindung mit einem Gerät herzustellen.

	T-Mobile	A1	ONE	Telering
Zuverlässigkeit	sehr gut	sehr gut	-	unzuverlässig

Tabelle 12: Auswertung Test - Verschiedene SIM-Karten im serverseitigen Modem

5.3.2 Verschiedene SIM Karten im Communication Center

Das Ziel dieser Tests war es herauszufinden, welche SIM Karten eine zuverlässige Datenübertragung auf Seiten des Communication Center garantieren. Es wurde sowohl in Ballungsgebieten mit einer hohen Netzdichte, als auch in Gebieten mit einer geringeren Netzdichte getestet. Desweiteren wurden die Tests zu verschiedenen Tageszeiten durchgeführt. Die SIM Karten wurden über einen Zeitraum von 10 Tagen zu verschiedenen Zeitpunkten(8:00, 12:00, 16:00, 20:00, 24:00) getestet. Das Ergebnis ist in Tabelle 13 zu sehen:

	T-Mobile	A1	ONE	Telering
Wien	sehr gut	sehr gut	-	unzuverlässig
Eggenburg	sehr gut	sehr gut	-	nicht eingesetzt
Gablitz	sehr gut	sehr gut	-	nicht eingesetzt
Wels	nicht funktioniert	sehr gut	-	nicht eingesetzt

Tabelle 13: Auswertung Test - Verschiedene SIM Karten im Communication Center

„sehr gut“ bedeutet, dass die Geräte zu 100% erreichbar waren. Diese Bestmarke wurde von A1 in jedem Testszenario erreicht, T-Mobile wurde in Wien, Eggenburg und Gablitz als „sehr gut“ eingestuft, in Wels war es nicht möglich mit der T-Mobile Karte eine Verbindung zu dem Gerät zu erstellen. Die SIM Karte von Telering wurde außerhalb des Ballungszentrums Wien nicht eingesetzt, da sie sich schon in Wien als nicht zuverlässig erwiesen hatte. Mit der ONE SIM Karte war es überhaupt nicht möglich eine Verbindung zwischen dem Modem und einem Communication Center zu herzustellen.

5.4 Crash Tests

Mit Abschluss des Projekts wurde eine Reihe von Tests durchgeführt, die im Folgenden als Crash Tests bezeichnet werden. Dabei handelt es sich um Tests, die den Ausfall einer Hardware Komponente im laufenden Betrieb simulieren und die zeigen sollen, wie sich das System in einem solchen Fall verhält. Ziel der Tests war es, aufgrund der Ergebnisse, entsprechende Lösungen für diese Szenarios zu finden, sodass der Normalbetrieb des Systems durch den Ausfall möglichst wenig beeinträchtigt wird.

5.4.1 Ausfall des Servers

Dieser Test wurde mittels eines Scripts durchgeführt, das während des laufenden Serverbetriebes gestartet wurde und dessen Aufgabe es war den Server herunterzufahren. Das Ergebnis war ein kompletter Ausfall von CCMP. In diesem Szenario gab es eine einzige Lösung. Es wurde ein sogenannter Watchdog auf einem anderen Server installiert. Dieses Programm hat die Aufgabe, in regelmäßigen Abständen Anfragen an den CCMP Server zu senden. Falls der Server mehr als 5 Minuten lang nicht antwortet, benachrichtigt der Watchdog den Administrator per SMS und Email. Dieser kann dann die notwendigen Schritte zur Wiederinbetriebnahme des Systems veranlassen.

5.4.2 Ausfall des serverseitigen Modems

In diesem Fall wurde untersucht, wie sich der Ausfall des serverseitigen Modems während eines Kommunikationsprozesses mit einem Communication Center auswirkt. Das Modem wurde vom Strom genommen während es mit dem Gerät verbunden war. Das Resultat war, dass der Kommunikationsprozess in einer Warteposition verblieben ist, da er auf eine Antwort des Modems warten musste. Diese konnte jedoch nicht empfangen werden, da das Modem zu diesem Zeitpunkt bereits ausgeschalten war. Dieses Problem wurde durch einen Timeout Thread gelöst, der den Kommunikationsprozess beendet, sobald er mehr als 1 Minute auf eine Antwort des Modems wartet. In einer zweiten Iteration ist das Problem aufgetreten, dass der pppd nicht mehr funktionsfähig war, nachdem der Kommunikationsprozess vom Timeout Thread beendet worden ist. Dieses Problem konnte nicht gelöst werden, also wurde ein Watchdog-ähnliches Programm geschrieben, das in den Logfiles nach bestimmten Mustern sucht, an denen es erkennt, dass pppd nicht mehr korrekt läuft. Bei einem entsprechenden Ergebnis wird der Systemadministrator per SMS oder Email benachrichtigt.

5.4.3 Ausfall eines Communication Centers

Der Ausfall eines Communication Centers wurde nach dem Ausfall des serverseitigen Modems getestet. Das Communication Center wurde einfach während einer bestehenden Verbindung vom Strom genommen. Das Resultat war ein Ähnliches wie bei Ausfall des Mo-

dems. Da zu diesem Zeitpunkt bereits ein Timeout Thread existiert hat, kam es zu keinen Endlos-Wartezeiten. Das in Abschnitt 5.4.3 beschriebene pppd Problem ist auch in diesem Fall aufgetreten und wurde durch das Watchdog Programm zur Benachrichtigung des Systemadministrators gelöst.

5.5 Automatisierte Tests

Für das automatisierte Testen von CCMP wurde ein spezieller Test entwickelt, der aus drei Komponenten besteht:

- Puretest (HTTP-Recorder)
- XML Konfigurationsfile
- Testprogramm

In diesem Abschnitt folgt eine Beschreibung der oben genannten Komponenten und es wird erläutert, wie mit dieser Testmethode ein automatisierter Test erstellt und durchgeführt wird.

5.5.1 Puretest

Puretest ist ein freies Tool der Firma „Minq Software“ und soll als Werkzeug zum Erstellen der automatisierten Tests eingesetzt werden. Es stellt einen HTTP-Recorder zur Verfügung mit dem es möglich ist, sämtliche HTTP-Requests, die von einem Benutzer im Web Browser ausgeführt werden, aufzunehmen. Die Reihenfolge in der der Benutzer die HTTP-Requests ausführt wird ebenfalls aufgezeichnet. Die Key-Value Pairs, aus denen die HTTP-Requests bestehen, dienen als Grundlage für das Konfigurationsfile.

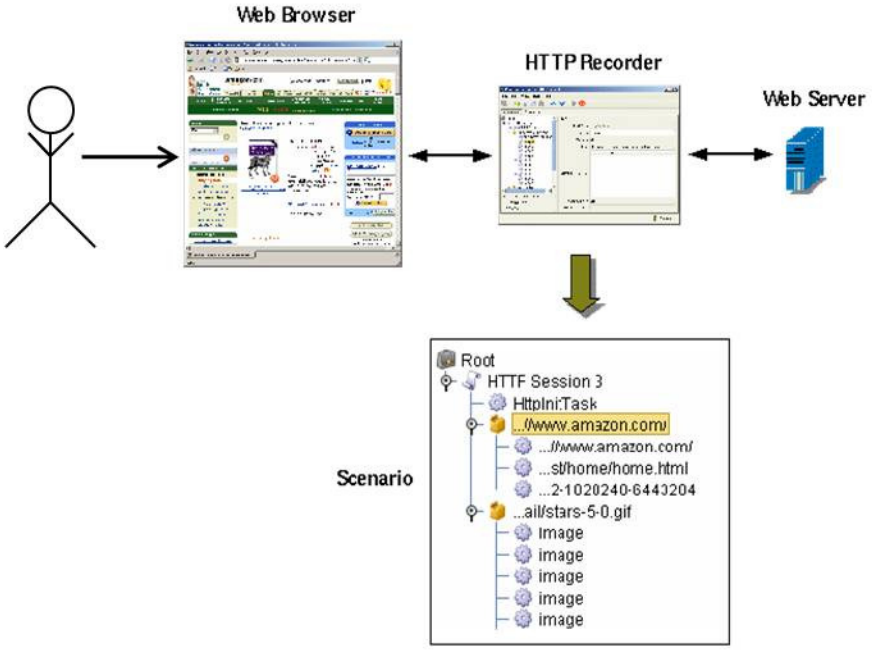


Abbildung 21: Purtest Übersicht

5.5.2 Das Konfigurationsfile

Das Konfigurationsfile bildet die Datenbasis für den automatisierten Test. Der Inhalt der von Puretest erstellten Textdatei wird entsprechend transformiert und in das File eingepflegt. Es beinhaltet alle Informationen, die zum Anlegen von Empfängern, Gruppen, und Geräten benötigt werden, sowie die Zugangsdaten zum Server auf dem die Applikation ausgeführt wird.

5.5.2.1 Empfänger

Innerhalb des Tags *<empfaenger>* können bis zu 8 Empfänger angegeben werden. Die Attribute für die einzelnen Empfänger werden im Tag *<empfaengername>* eingefügt. Für einen Empfänger können der Name, der Vorname, die Telefonnummer und die Emailadresse angegeben werden.

```
<empfaenger>
        <empfaengername name="huber" vorname="peter" telnummer="069914448883"
        email="xx@yy.com"/>
        <empfaengername name="maier" vorname="franz" telnummer="06642343322“
        email="xx@yy.com"/>
</empfaenger>
```

5.5.2.2 Gruppe

Gerätegruppen können mit dem Tag <gruppe> angegeben werden. Innerhalb dieses Tags wird der Gruppenname eingetragen. Es können beliebig viele Gruppen angeführt werden.

```
<gruppe>Testgruppe1</gruppe>
<gruppe>Testgruppe2</gruppe
```

5.5.2.3 Geraet

Es ist möglich, beliebig viele Geräte zu beziffern. Ein Gerät beginnt mit dem Tag *<geraet>*, es hat außerdem die Attribute *simnummer*, *typ*, *initialisiert* und *gruppe*. Das Attribut *simnummer* enthält die Nummer unter der das Gerät erreichbar ist, *typ* gibt das Gerätemodell an, mit *initialisiert* kann unterschieden werden, ob es sich um ein Gerät handelt, das bereits in Betrieb genommen wurde oder nicht und *gruppe* legt die Gruppenzugehörigkeit des Geräts fest.

```
<geraet  simnummer="06502099983" typ="6/6/2" initialisiert="0" gruppe="Testgruppe1"
        <ports>
                <port>
                        <name>Analog1</name>
                        <typ>Analog Eingang 1</typ>
                        <untergrenze>10</untergrenze>
                        <obergrenze>40</obergrenze>
                <einheit>volt</einheit>
                </port>
```

```
            <port>
                <name>Digital 1</name>
                <typ>Digital Eingang 1</typ>
                <invertiert>0</invertiert>
            </port>
        </ports>
…
</geraet>
```

Innerhalb des Tags *<ports>* können die einzelnen I/Os angegeben werden. Ein einzelner I/O wird durch das Tag *<port>* gekennzeichnet, innerhalb von *<port>* werden die einzelnen Attribute des I/Os mittels Tags angegeben. *<name>* spezifiziert den Namen des I/O, <typ> die Bezeichnung des I/O am Gerät, *<untergrenze>,<oberrgrenze>* und *<einheit>* gibt die Attribute für die analogen Eingänge an, *<invertiert>* bestimmt für digitale I/O, ob sie invertiert sind oder nicht.

In einem weitern Schritt ist geplant, das Konfigurationsfile um Meldungen und Statusmeldungen sowie ihre Attribute zu erweitern.

5.5.3 Das Testprogramm

Für das Testprogramm wurde Java als Programmiersprache gewählt. Es besteht aus insgesamt drei Klassen: *CCMPTest*, *CCMPTestConfig* und *CCMPTester*. Diese Klassen werden im Folgenden kurz beschrieben.

5.5.3.1 CCMPTest

CCMPTest ist die Hauptklasse des Programms und der Einstiegspunkt in das Testprogramm. In dieser Klasse befindet sich die Methode zur Steuerung des Programmablaufs. In der Methode wird die Klasse *CCMPTestConfig* erzeugt und ihre Methode zum Einlesen des Konfigurationsfiles ausgeführt. Ist das File erfolgreich eingelesen, wird die Klasse *CCMPTester* instanziiert. Dann werden die Datenstrukturen aus *CCMPTestConfig* geholt und eine nach der anderen an die Klasse *CCMPTester* übergeben. In *CCMPTest* findet das Fehlerhandling statt. Treten während des Durchlaufs keine Fehler auf, so kann der Test als vollständig ausgeführt betrachtet werden.

5.5.3.2 CCMPTestConfig

In dieser Klasse wird das Konfigurationsfile eingelesen und die spezifischen Datenstrukturen aufgebaut. Die Methode zum Einlesen des Konfigurationsfiles bekommt als Argument den Pfad übergeben, unter dem das File zu finden ist. In dieser Methode wird das File eingelesen und sein Inhalt in einen Stringbuffer geschrieben. Die Klasse hat desweiteren eine Methode zum Aufbau der Datenstrukturen für Geräte, Gruppen, Empfänger und I/O. In dieser Metho-

de wird der Stringbuffer mit dem Inhalt des Files durchlaufen, verarbeitet und in eine Hashtable-Struktur geschrieben. Die Klasse kapselt außerdem eine Reihe von getter-Methoden, welche die fertigen Datenstrukturen retournieren.

5.5.3.3 CCMPTester

Diese Klasse übernimmt im Prinzip die Aufgabe des Testers. Sie kapselt Methoden, die als Argumente Hashtables mit den spezifischen Datenstrukturen (Gerät,I/O,Empfänger) übergeben bekommen. Diese Datenstrukturen werden in der Methode transformiert und zu einem HTTP-Request zusammengefügt, den die CCMP Applikation verarbeiten kann. Dann wird der fertige Request an den Server geschickt. Ist der Request erfolgreich, so wird true an die aufrufende Klasse, ansonsten false retourniert. Außerdem befinden sich in dieser Klasse Methoden zum Abschicken von HTTP-Requests.

5.5.4 Erstellen eines Tests

Im ersten Schritt zur Erstellung eines automatisierten Tests entwirft der Tester ein Testszenario, dass heißt er legt genau fest, welche Funktionen er im System testen will, in welcher Reihenfolge er diese ausführen will und wie die Testdatensätze beschaffen sein sollen. Für das Testszenario sollte ein eigenes Dokument erstellt werden.

Als nächsten Schritt startet der Tester Puretest und stellt es so ein, dass es sämtliche HTTP-Requests, die von seinem Webbrowser gesendet werden, aufnimmt. Nun loggt er sich auf dem CCMP Server ein und arbeitet das Testszenario mit Hilfe des vorher erstellten Dokuments ab.

Ist er damit fertig, beendet er die Aufnahme in Puretest und kontrolliert, ob die aufgenommenen HTTP-Requests korrekt sind. Dann speichert er sie in einem Textfile ab.

In einem nächsten Schritt überträgt er die Key-Value Pairs in das Konfigurationsfile. Das fertige XML File hinterlegt der Tester in das entsprechende Verzeichnis aus dem das Testprogramm die Konfigurationsdatei einliest. Der Test kann nun durch das Starten des Testprogramms angestoßen werden. Er läuft vollkommen eigenständig ab und kann beliebig oft durchgeführt werden.

5.5.5 Ablauf eines Tests

Für die Durchführung des Tests wird ein eigener Account angelegt, damit durch den Test keine Produktivdaten manipuliert werden. Der Ablauf eines Tests zum Überprüfen der Grundfunktionen von CCMP kann folgendermaßen aussehen:

Nach einem erfolgreichen Login auf dem entsprechenden Server werden alle existierenden Empfänger gelöscht. Findet das Programm keine weiteren Empfänger, geht es zum nächsten Schritt über und wendet dasselbe Verfahren zuerst bei allen vorhandenen Geräten und anschließend bei allen Gruppen an. Das ist notwendig um eventuelle Inkonsistenzen und Redundanzen zu vermeiden, die dadurch entstehen können, dass Geräte mit der gleichen Telefonnummer mehrmals im System existieren. Die Geräte müssen vor den Gruppen gelöscht werden, da sie sonst für das Testprogramm nicht mehr auffindbar sind.

Sobald alle Datensätze aus dem System gelöscht wurden, beginnt das Testprogramm entsprechend dem Konfigurationsfile Empfänger anzulegen. Nach dem erfolgreichen Anlegen der Empfänger werden Gruppen und im nächsten Schritt die Geräte angelegt, wobei jedes Gerät einer Gruppe zugewiesen wird. Die Geräte werden ohne einen Anruf initialisiert.

Dann werden die I/Os aktiviert, indem ihre Attribute mit den Daten des Konfigurationsfiles befüllt werden.

Der automatisierte Teil des Tests ist damit beendet. Der Tester loggt sich mit dem Test Account ein und kontrolliert mit Hilfe seines Testszenarios, ob alle Empfänger, Gruppen, Geräte und I/Os korrekt angelegt wurden. Ist das der Fall, so kann angenommen werden, dass die Grundfunktionen des Systems ordnungsgemäß funktionieren. Falls nicht, muss der Tester in den Logfiles nach Fehlern suchen und diese beheben. Dann kann der Test erneut durchgeführt werden.

5.5.6 Ziel der automatisierten Tests

Durch die automatisierten Tests sollen Standardfunktionen, die der Tester normalerweise im Webinterface selbst ausführen muss, getestet werden. Zu diesen Standardfunktionen zählen das Anlegen, Ändern, Ansehen und Löschen von Geräten, Empfängern, Gruppen und I/Os. In einer späteren Projektphase soll auch das Anlegen, Ändern, Ansehen und Löschen von Meldungen, das Initialisieren und Synchronisieren von Geräten sowie das Datenlogging automatisch getestet werden. Damit soll bei einer Neuinstallation oder beim Auftreten von Fehlern die Grundfunktionalität der Applikation schnell und effizient getestet werden können. Dem Tester bleibt dadurch die mühsame, immer gleiche und zeitintensive Eingabe von Daten erspart und er kann sich auf die Interpretation der Testergebnisse konzentrieren. Das soll eine Einsparung von Kosten ermöglichen und dem Tester dabei helfen, sich verstärkt auf das Testen von kritischen Bereichen der Applikation konzentrieren zu können.

6 Ausblick

Der Schlussabschnitt der vorliegenden Arbeit enthält einen kurzen Rückblick die das CCMP Projekt und beschäftigt sich dann mit den weiteren Einsatzmöglichkeiten und Erweiterungen von CCMP.

Die dem Projekt zu Grunde liegende Aufgabenstellung kann rückblickend als ausgesprochen anspruchsvoll und interessant bezeichnet werden. Die größte Herausforderung lag in der Entwicklung der Kommunikationsschnittstelle; die Zusammenführung der einzelnen Komponenten (Modem, PPPD, Chat, Java) zu einer robusten und verlässlichen Basis für die automatisierten Kommunikationsprozesse erschien in manchen Phasen des Projektes fast unmöglich, konnte aber am Ende dennoch realisiert werden. Die Konzeption der CCEngine, jener RMI-Komponente, die für Initialisierungs-/Synchronisations-/ und Datenlogging - Prozesses zuständig ist, stellte den mit Sicherheit interessantesten Teil des Projektes dar. Im Oktober 2006 wurde das Projekt mit dem 2ten Platz des ebiz egovernment award Wien ausgezeichnet. Im Moment wird CCMP bei der Firma Wien Energie eingesetzt und hat außerdem eine Reihe von Anwendern, die das Communication Center in einem Szenario mit ein bis zwei Geräten einsetzten.

Im Jahr 2007 gab es die Überlegung das Communication Center und CCMP für die Überwachung von Wohnhausanlagen einzusetzen. In diesem Szenario würde das Communication Center vor allem Zählerkästen und Gemeinschaftsräume wie Garagen, Fahrradräume etc. überwachen und den Stromverbrauch kontrollieren. Bei Unregelmäßigkeiten könnte die zuständige Person benachrichtigt werden. Dieses Projekt wäre vor allem in Bezug auf die Anzahl der eingesetzten Geräte interessant, da überlegt wurde, mehrere hundert Wohnhausanlagen mit dem Communication Center auszustatten.

Von Seiten des Auftraggebers der Applikation gibt es Überlegungen, CCMP funktional und grafisch zu überarbeiten bzw. zu erweitern. Die grafische Überarbeitung soll sich vor allem auf eine Anpassung der Benutzerschnittstelle an Web 2.0 Standards beziehen. Es soll auf der einen Seite eine optische Überarbeitung geben, die das Userinterface intuitiver und benutzerfreundlicher macht. Hier soll ein professioneller Grafiker bemüht werden und es soll eine Anlehnung an das Google Groups Design geben. Auf der anderen Seite sollen Komponenten wie DHTML und Ajax eingesetzt werden, damit die Benutzung der Oberfläche der einer modernen Webapplikation entspricht. Für die funktionale Erweiterung von CCMP gibt es folgende Ideen: ein statistisches Tool und eine GIS Komponente. Das statistische Tool soll zur grafischen Auswertung der geloggten Daten innerhalb der Webplattform dienen. Es

soll zum Beispiel möglich sein, Kurven und Diagramme zu zeichnen, Durchschnittswerte zu bilden, sowie mögliche Ausreißer leicht zu erkennen. Die Auswertungen sollen außerdem gespeichert werden und dem Benutzer zu jeder Zeit zur Verfügung stehen. Damit würde der Benutzer sich die Auswertung der Daten in den Excel Files ersparen. Die GIS Komponente könnte dem Benutzer einen geografischen Überblick über seine Geräte geben. Das ist vor allem in Szenarios mit sehr vielen, beziehungsweise mit weit verteilten Geräten von Vorteil. Fällt ein Gerät aus, so kann sofort festgestellt werden, wo dieses Gerät aufgestellt ist und wie es erreichbar ist.

Literaturverzeichnis

[CHAT08] chat(8) - Linux man page, http://linux.die.net/man/8/chat , 02.07.2008

[DUST01] **Dustin E.**, **Rashka J., Paul J.:** Sofware automatisch testen, 1.Auflage, Springer-Verlag Berlin Heidelberg

[GILL00] **Gilly M.**: MAGNETIC- Design of an Object-Oriented Web Site Management System, TU-Wien 2000

[Kapp04] **Kappel G., Pröll B., Reich S., Retschitzegger W..:** Web Engineering – Systematische Entwicklung von Web-Anwendungen, dpunkt.verlag GmbH, Heidelberg, 2004

[Kofl99] **Kofler M.:** Linux – Installation, Konfiguration, Anwendung, 4.Auflage, Addison Wesley, Bonn, 1999

[LEPB05] **LEP:** Bedienungsanleitung Communication Center (CC), http://www.lehotzki.at/ccbed266v14.pdf. , 21.05.2008

[LEPF07] **LEP:** LeP Anwendungsfolder und Referenzen, http://www.lehotzki.at/LEP_FOLDER_2007.pdf , 21.05.2008

[LEPDB05] **LEP:** Datenblatt Communication Center (CC), http://www.lehotzki.at/ccdatenblatt2005.pdf ,23.05.2008

[META08] http://www.metamagix.net, 14.03.2007

[Myer91] **Myers G.J.:** Methodisches Testen von Programmen, 4. Auflage, Oldenbourg, München, 1991

[PPPD08] **Mackerras P.:** pppd(8) - Linux man page, http://linux.die.net/man/8/pppd , 23.06.2008

[RFC1661] **Simpson, W.A.: The Point-to-Point Protocol (PPP),** http://rfc.net/rfc1661.html, 17.06.2008

[Tane03] **Tanenbaum A.S.:** Computernetzwerke, 4. Auflage, Pearson Studium, München, 2003

[TC35] TC 35 Terminal Spezifikation: http://www.mobiltim.com/images/terminal/tc35terminal/TC35.pdf , 15.04.2008

[TC35UG] TC 35 Terminal User Guide: www.dateline.ru/download/manual/man-**siemens**-gsm-**tc35**t.pdf , 15.04.2008

[Zuse01] **Zuser W., Biffl S., Grechenig T., Köhle M.:** Sofware-Engineering mit UML und dem Unified Process, Pearson Studium, München, 2001