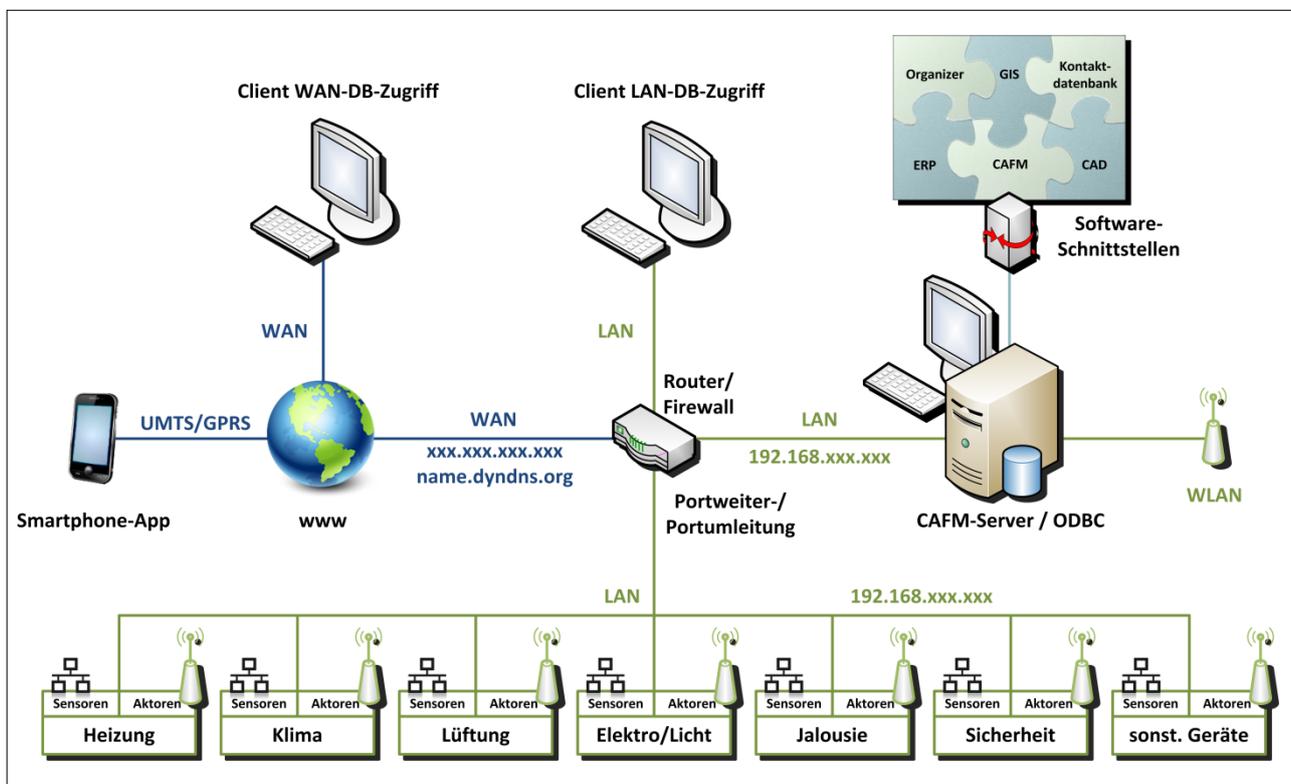


EDV-gestütztes Berechnen, Entwerfen und Konstruieren im Bauwesen

Entwicklung eines Standards für innovative IT-Schnittstellen im Facility Management zur Kopplung webbasierter CAFM-Systeme mit neuen Gebäudeautomationstechnologien



DISSERTATION

Asbjörn Gärtner

Juni 2013



Vom Fachbereich Bauingenieurwesen
der Technischen Universität Kaiserslautern
zur Verleihung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

**Entwicklung eines Standards für innovative
IT-Schnittstellen im Facility Management zur
Kopplung webbasierter CAFM-Systeme
mit neuen Gebäudeautomationstechnologien**

von

Dipl.-Ing. Asbjörn Gärtner

aus Schmalenberg

Dekan:	Prof. Dr.-Ing. Wolfgang Kurz
Prüfungsvorsitzender:	Prof. Dr.-Ing. Karsten Körkemeyer
1. Berichterstatter:	Prof. Dr.-Ing. Klaus Wassermann
2. Berichterstatter:	Prof. Dr.-Ing. Joachim W. Hohmann

Tag der mündlichen Prüfung: 28. Juni 2013

Kaiserslautern 2013

D 386

Vorwort

Die vorliegende Arbeit entstand während meiner Zeit als wissenschaftlicher Mitarbeiter im Fachgebiet Bauinformatik und Facility Management des Fachbereichs Bauingenieurwesen der Technischen Universität Kaiserslautern.

Durch meine Tätigkeit im Lehrbetrieb und operativen Management des neu aufgebauten und ersten universitären Studiengangs Facility Management hatte ich stets einen engen Bezug zu den Themenfeldern IT, Programmierung und CAFM. Zudem konnte ich im Rahmen von Drittmittelprojekten viel Erfahrung auf den genannten Gebieten sammeln. Meine Vorliebe für Technik, insbesondere das Interesse an Automationstechnologien, brachte weitere Impulse zur Themenfindung. Dieser Background führte zusammen mit meinen Beobachtungen in der Entwicklung neuer Gebäudeautomationssysteme zur Erkenntnis der Möglichkeiten und des Bedarfs von Technologiekopplungen und damit zum Thema dieser Arbeit.

Mein besonderer Dank gilt meinem Doktorvater, Inhaber des Lehrstuhls, Initiator und Leiter des Bachelorstudiengangs Facility Management Herrn Prof. Dr.-Ing. Klaus Wassermann für die wissenschaftliche Unterstützung, uneingeschränkte Betreuung und vielseitigen Anregungen in unseren Gesprächen.

Danken möchte ich auch dem mit Herrn Prof. Dr.-Ing. Klaus Wassermann gemeinsamen Initiator des Studiengangs, Herrn Prof. Dr.-Ing. Joachim W. Hohmann, für die Übernahme des Koreferats als zweiter Berichterstatter sowie die fachliche Unterstützung mit vielen hilfreichen Tipps und Ratschlägen.

Dem mit Herrn Prof. Dr.-Ing. Klaus Wassermann gemeinsamen Leiter des Studiengangs Facility Management, Herrn Prof. Dr.-Ing. Karsten Körkemeyer, möchte ich für die spontane Übernahme des Vorsitzes der Promotionskommission sehr herzlich danken.

Mein weiterer Dank gilt den Mitarbeitern des Fachgebiets Bauinformatik und Facility Management Frau Annette Reincke und Frau Christiane Köhler, dem Auszubildenden Herrn Christian Bayer und ehemaligen Auszubildenden Herrn Jens Hellriegel für den technischen Support, dem Inhaber des Fachgebiets Immobilienökonomie Herrn Juniorprof. Dr. Björn-Martin Kurzrock für seine stets motivierenden Worte und Empfehlungen sowie allen Freunden und Kollegen, die mir den Rücken für die erfolgreiche Fertigstellung dieser Arbeit gestärkt haben.

Herrn Jürgen Reimann von der Akktor GmbH und Herrn Christian Genter von der Omnio AG danke ich für den Support und die Bereitstellung von Hardware aus der Ratio©-Produktreihe zu Testzwecken.

Insbesondere meiner Frau Michaela, meiner Tochter Lislely und meinen Eltern Erika und Arne möchte ich für ihre Unterstützung auf meinem Weg sehr herzlich danken.

Kaiserslautern, Juni 2013

Dipl.-Ing. Asbjörn Gärtner

Kurzfassung

Immer mehr Gebäude werden aus Gründen des Komforts, der Sicherheit und der Wirtschaftlichkeit mit fortschrittlicher Automationstechnik ausgestattet. Daher gewinnt Gebäudeautomation zunehmend an Bedeutung. Hierfür sind gebäudetechnische Anlagen in vielen Fällen über Busleitungen mit einem Leitrechner vernetzt und zentral erfasst. Sie werden mit Hilfe von Sensoren und Aktoren gesteuert.

Um Bauwerke effektiv und effizient bewirtschaften zu können, werden zunehmend CAFM-Systeme als unverzichtbares Werkzeug des Facility Managers eingesetzt. Dabei werden heute die gleichen gebäudetechnischen Anlagen ein zweites Mal erfasst, um sie im Rahmen des Anlagen- und Wartungsmanagements verwalten zu können. Eine webbasierte, datenbankgekoppelte Steuerung dieser Anlagen direkt über ein CAFM-System ist derzeit nicht verfügbar. Themenbezogene Normen und Richtlinien geben keinen Aufschluss über konkrete Kopplungsmechanismen von Gebäudeautomation und CAFM zur Steuerung gebäudetechnischer Anlagen. Es fehlt an Netzwerkmodellen und Lösungskonzepten.

Die vorliegende Arbeit setzt an dieser Problematik an. Sie prüft, in welcher Weise gebäudetechnische Einrichtungen hinsichtlich einer komfortablen und zentralen Steuerungsfunktion in CAFM-Systeme eingebunden werden können. Aspekte wie neuartige Vernetzungsvarianten, die Bereitstellung von Anlagendaten mit zugehörigen Importfunktionen sowie die Möglichkeit einer browsergestützten Bedienung stehen dabei im Fokus der Betrachtung. Der Leitgedanke dabei ist, vorhandene Technologien und Standards zu nutzen, um daraus neue Lösungen zu schaffen.

Der vorgestellte Lösungsansatz analysiert, vergleicht und bewertet unterschiedliche Netzstrukturen, dessen Ergebnis als Entscheidungshilfe für die Modellentwicklung dient. Der „Plug and Play“-Gedanke wird durch die Verwendung der bereits standardisierten Netzwerktechnologie UPnP (Universal Plug and Play), realisiert, wofür es der Ausarbeitung und Definition neuer bereitzustellender Datenformate (XML) bedarf. Für die praktische Umsetzung wird eine eigene Datenbankstruktur (MS Access) zur Simulation eines CAFM-Systems entworfen und eine mit VB.NET programmierte Benutzeroberfläche erstellt. Der entwickelte Programmbaustein „FMControl“ demonstriert das webbasierte Schnittstellenkonzept mit Importfunktion und Steuermechanismen.

Die Ergebnisse dieser Arbeit schaffen die Basis für die Optimierung im Bereich der webbasierten und datenbankgekoppelten Gebäudeautomation. Durch die Implementierung der vorgestellten Funktionen des Programmbausteins „FMControl“ können CAFM-Systeme zu leistungsstarken Steuerungsinstrumenten für Gebäude und deren Anlagen ausgebaut werden.

Abstract (English)

A continuously growing number of buildings are today equipped with modern automation technologies, primarily for reasons of comfort and efficiency. As a result Building Automation (BA) has an increasing impact on building construction and operation. Building automation systems are in many cases linked to a central computer via bus cabling and operated centrally. They are controlled with the help of sensors and ASIs.

CAFM systems, the most important IT tools of facility managers, are increasingly being deployed to operate buildings efficiently and securely. In this case the same building automation services are processed a second time to administer them within the scope of an equipment and maintenance management concept. Today web-based, database integrated control of these systems directly through a CAFM system is not available. Relevant standards and guidelines do not provide any information about tangible coupling mechanisms between building automation and CAFM for controlling building automation systems, and there is a lack of network models and concepts for feasible solutions.

The following paper will attempt to address this problem. It will examine how building automation systems and convenient central controls can be integrated into CAFM systems. The focus of the thesis will be on aspects of innovative networking models, deploying system data together with import functions, as well as the feature of a browser-based systems operation. The guiding principle here is to utilize existing proven technologies and standards to create new solutions.

This approach to finding a solution will analyze, compare and evaluate various network configurations, and the results obtained will contribute to decision-making for developing models. The concept of "Plug and Play" is realized by using the already standardized network technology UPnP (Universal Plug and Play), which requires new data formats (XML) for planning and definition. A separate database structure (MS Access) was developed and a user interface designed in VB.NET to simulate CAFM system functionalities. The here presented software module "FMControl" will demonstrate the web-based interfacing concept with import functions and control mechanisms.

The results of this paper will generate a basis for further optimization in the area of web-based and database oriented building automation. Implementation of the presented functions of the "FMControl" software module will allow CAFM systems to be enhanced as high-performance building controls tools.

Inhaltsverzeichnis

Vorwort	3
Kurzfassung	5
Abstract (English)	7
Inhaltsverzeichnis	9
1 Einleitung	13
2 Aufgabenstellung	15
2.1 Ausgangssituation	15
2.2 Problemstellung	18
2.3 Zielsetzung	20
2.4 Vorgehensweise	22
3 Grundlagen	25
3.1 Facility Management	25
3.2 CAFM-Systeme / IWMS.....	30
3.3 Gebäudeautomation.....	38
3.4 Middleware	43
3.5 Betrachtung von Normen und Richtlinien	45
3.5.1 DIN 32736: Gebäudemanagement – Begriffe und Leistungen	45
3.5.2 GEFMA 400: Computer Aided Facility Management (CAFM)	46
3.5.3 GEFMA 410: Schnittstellen zur IT-Integration von CAFM-Software	47
3.5.4 VDI/GEFMA 3814 Blatt 3.1: Gebäudeautomation	50
4 Technologien	53
4.1 Bussysteme	53
4.1.1 KNX (Bussystem der Konnex-Association).....	55
4.1.2 LON (Local Operating Network)	61
4.1.3 LCN (Local Control Network)	66
4.1.4 BACnet	70
4.1.5 digitalSTROM	73
4.1.6 EnOcean.....	76
4.2 Computernetzwerke	82
4.2.1 LAN via Kabel/Ethernet (klassisch)	83
4.2.2 LAN via Stromnetz (PowerLAN, Powerline, dLAN)	85
4.2.3 LAN via Funk (WLAN).....	87
4.2.4 Netzwerkprotokoll TCP/IP	89

5	Modellentwicklung zur Kopplung von CAFM und GA	93
5.1	Varianten der Systemintegration durch hardwareabhängige Schnittstellenmodelle..	94
5.1.1	Netzwerkstruktur eines CAFM-Systems.....	94
5.1.2	Unidirektionale Kopplung des Feldbus über lokale Schnittstellenlösung.....	96
5.1.3	Unidirektionale Kopplung des Feldbus über webbasierte IP-Controller	99
5.1.4	Bidirektionale Kopplung des Feld-/Funkbus über lokalen USB/COM-Port.....	101
5.1.5	Bidirektionale Kopplung netzwerkfähiger Komponenten über lokales LAN.....	104
5.1.6	Bidirektionale Kopplung internetfähiger Komponenten über globales WAN ..	107
5.2	Bewertung und Vergleich unter Beachtung maßgebender Anforderungskriterien...	109
5.2.1	Anforderungskriterien	109
5.2.2	Entscheidungsmatrix	112
5.2.3	Vor- und Nachteile des gewählten Modells	113
5.3	Aufbau des Integrationsmodells und technische Umsetzung	115
5.3.1	Interner Zugriff vs. Fernzugriff	116
5.3.2	Dynamic Domain Name System (DynDNS) bei Fernzugriff	118
5.3.3	Portweiterleitung / Portumleitung bei Fernzugriff	119
5.3.4	Gerätean Kopplung via UPnP (Universal Plug and Play).....	122
5.3.5	XML (Extensible Markup Language)	128
5.4	Workflow-Modelle	137
5.5	Innovative Standards	139
6	Praktische Umsetzung	151
6.1	Aufbau des Datenbankmodells.....	151
6.1.1	Tabellen	152
6.1.2	Relationen.....	156
6.2	XML-Schema für UPnP	160
6.2.1	device.xml.....	160
6.2.2	scpd.xml.....	161
6.3	Programmbaustein „FMControl“	172
6.3.1	Aufbau der Importfunktion	174
6.3.2	Aufbau der Steuerfunktion.....	181
6.3.3	Vorbereitung der Simulation	185
6.4	Facility Automation durch Integration von GA und CAFM	187
6.4.1	Benutzeroberfläche	187
6.4.2	Anlagenimport.....	189
6.4.3	Anlagensteuerung	192

7 Fazit und Ausblick.....	195
7.1 Fazit dieser Arbeit	195
7.2 Ausblick	196
I. Anhang	199
I.a Struktur der relationalen Datenbank, Tabellenbeziehungen	199
I.b FMA.Codex Device Description device.xml	200
I.c FMA.Codex Service Description scpd.xml.....	202
I.d Quellcode FMControl.....	231
II. Literaturverzeichnis.....	269
III. Abbildungsverzeichnis	275
IV. Tabellenverzeichnis	281
V. Abkürzungsverzeichnis	283
VI. Lebenslauf.....	285

1 Einleitung

Wir leben in einer hoch technisierten Welt, unser Alltag ist bestimmt durch Technik, alles ist miteinander vernetzt, das Internet ist längst zum Standard geworden und nicht mehr wegzudenken. Die Entwicklung bleibt nicht stehen, wir sind unaufhaltsam auf der Suche nach neuen, innovativen Technologien.

In der heutigen Zeit ist dieser technische Fortschritt nicht nur in der Computerindustrie, dem Bereich der Unterhaltungselektronik, der Anlagentechnik oder der Automobilindustrie, sondern auch im Segment des Bauwesens und Gebäudemanagements zu finden. Diese technischen Entwicklungen erleichtern uns den Alltag, sowohl im Sinne von Komfort und Sicherheit als auch im Sinne von Wirtschaftlichkeit durch Effektivität und Effizienz. Sie erstrecken sich mittlerweile auf alle Bereiche und Lebenslagen, so auch dem ganzen Sektor der Gebäudeautomation. Waren anfangs nur Industrie- oder Bürogebäude mit einer solchen Technik ausgestattet, so erfreut sich der Einsatz im Privatbereich immer größerer Beliebtheit.

Die Begriffe der Gebäudeautomation (GA) und Gebäudeleittechnik (GLT) wurden bereits in den 1970er Jahren geprägt und waren Jahrzehnte das Maß aller Dinge, um gebäudetechnische Einrichtungen zu messen, zu steuern und zu regeln (MSR-Technik). Dieser Technologie lagen sog. Bus-Systeme zu Grunde, die getrennt von IT-Netzwerken völlig autark „arbeiteten“. Die Anbindung an herkömmliche Computersysteme entwickelte sich erst in den 1990er Jahren, als die Netzwerktechnologie fortgeschritten war und es in diesem Bereich standardisierte Protokolle gab. Der Gedanke, die Informationen der GLT auch über das heutzutage gebräuchliche LAN-Netz abzurufen, prägte die Hardware-Entwicklung in den letzten zehn Jahren zunehmend. Resultierend aus der Vielzahl vorhandener Bus-Systeme und Technologien entwickelte sich parallel auch eine Vielzahl an Modulen, die eine Kopplung an IT-Netzwerke ermöglichen.

Die hier vorliegende Arbeit befasst sich mit der Analyse und Bewertung vorhandener Netzwerktechnologien in der Gebäudeautomation mit Fokus auf der Kombination von Kommunikationswegen, sowohl aus informations- und ingenieurstechnischer Sicht als auch aus einer auf das Gebäude bezogenen ganzheitlichen Betrachtung – dem Facility Management. Dabei werden vorhandene Infrastrukturen aus wirtschaftlichen Gründen berücksichtigt und das Blickfeld in Bezug auf neue technologische Ansätze für bislang ungenutzte Möglichkeiten erweitert.

Kern dieser Arbeit ist der Entwurf eines innovativen Netzwerkmodells zur Einbindung gebäudetechnischer Anlagen in computergestützte Systeme für das Facility Management (CAFM-Systeme) als neuartiges Steuerungsinstrument, um Gebäude aus technischer, kaufmännischer und infrastruktureller Sicht optimal und nachhaltig betreiben zu können.

Mit dem Einsatz neuartiger Vernetzungsvarianten im Bereich der Gebäudeautomation und damit neuen Möglichkeiten der Prozessoptimierung durch aktive Steuerung gewerkebezogener Endgeräte soll den Anforderungen eines wirtschaftlichen, effektiven und effizienten Betriebes Rechnung getragen werden.

Eine besondere Herausforderung ist es, die verschiedenen Bereiche E-Technik, Bauingenieurwesen und Informatik aus FM-Sicht miteinander in Beziehung zu bringen und somit auf der Grundlage fachbereichseigener Technologiestandards mit bereits definierten Standard-Protokollen ein Modell für neue, webbasierte Vernetzungsvarianten zu entwickeln. Daraus leiten sich erstmals zu definierende „FM/IT-Standards“ für Schnittstellen zur Anlagensteuerung zwischen der Gebäudeautomation und CAFM-Systemen ab.

Um diese Herausforderung angehen zu können, bedarf es im Vorfeld einiger Begriffsdefinitionen und Antworten auf grundlegende Fragen. Denn was ist z. B. unter CAFM, GLT, IT-Schnittstellen oder -Standards zu verstehen, dazu noch „webbasiert“? Wie funktionieren Kommunikationssysteme mit KNX, LON oder BACnet? Was sind Netzwerktechnologien? Wie hängen (W)LAN, Ethernet/IP oder UPnP zusammen? Was gibt es schon auf dem Sektor der webbasierten Gebäudeautomation, was zählt bereits zum Standard, wohin soll die Reise gehen und was ist die Vision? Ist die Idee der Steuerung gebäudetechnischer Einrichtungen über CAFM-Systeme bereits in einschlägigen Normen und Richtlinien im Ansatz bereits verankert? Diese Fragestellungen werden zu Beginn dieser Arbeit behandelt und beantwortet, was dem besseren Verständnis der darauf folgenden Kapitel dient.

2 Aufgabenstellung

Zu Beginn dieses Kapitels richtet sich der Blick auf die gegenwärtige Situation im Bereich des Gebäudebetriebs unter dem Aspekt der Anlagenverwaltung und Anlagensteuerung. Daraus wird zunächst die Problemstellung abgeleitet, worauf die Definition der Zielsetzung mit anschließender Beschreibung der angestrebten Vorgehensweise folgt.

Anlass und Motivation für diese Arbeit ist die Erkenntnis aus Beobachtungen der Entwicklung neuer Technologien im Bereich der Gebäudeautomation. Bislang gibt es keine Studien zum Vergleich von Vernetzungsvarianten aus FM-Sicht. Dabei steckt großes Potenzial für CAFM-Systeme in solchen Technologiekopplungen.

2.1 Ausgangssituation

Smart Home, Assisted Living oder das „intelligente Haus“ – wofür stehen diese Begriffe? Meist tauchen sie in den Medien auf, wenn es um Werbung für solche Systeme im Bereich von Wohnhäusern geht. Gebäudeautomation ist nichts Neues, wird aber im Vergleich zu gewerblichen Einrichtungen bislang nur vereinzelt im Privathaushalt eingesetzt. Doch der technische Fortschritt bringt viele neue Funktionen und Möglichkeiten mit sich, die mittlerweile auch für Privatanwender erschwinglich werden. Die Technologien gewinnen zunehmend an Bedeutung und werden durch gute Marketingstrategien verstärkt wahrgenommen.

Diese intelligenten Systeme erwecken oftmals den Eindruck, nur aus Gründen des Komforts eingesetzt zu werden. Spielerei, sagen viele. Doch tatsächlich steckt mehr dahinter. Komplexe Steuerungsmechanismen der GLT reagieren auf Ereignisse oder vordefinierte Szenarien und stellen damit den reibungslosen Ablauf des Gebäudebetriebs sicher. Nicht nur die Sicherheit, sondern auch die Erfüllung der Betreiberverantwortung steht dabei im Vordergrund. Ein weiterer Faktor ist das Energieeinsparpotenzial durch eine intelligente Steuerung, vor allem von Energieverbrauchern, Heizzentralen oder Lüftungszentralen, mit Hilfe der MSR-Technik. Der Einsatz dieser effektiven und effizienten Technologien trägt durch innovative Steuerungsfunktionen hinsichtlich neuer Möglichkeiten der Energieoptimierung zu einer ökonomischen und nachhaltigen Gebäudebewirtschaftung bei, was durch Gütesiegel und Zertifikate wie LEED, BREEAM oder DGNB bescheinigt und dokumentiert werden kann.

Richten wir den Blick auf die beiden für Gebäudebewirtschaftung und Gebäudeautomation verwendeten Systeme und deren Zweck:

CAFM-Systeme werden für die computergestützte Gebäudebewirtschaftung eingesetzt und sind das Handwerkszeug eines Facility Managers. Sie dienen der Unterstützung zur Abwicklung von FM-Prozessen. Dazu gehören die Erfassung objektbezogener Daten für Verwaltung, kaufmännische Abwicklung oder auch Benchmarking und Prozessoptimierungen. Auch wenn der Trend zu rein webbasierten Lösungskonzepten geht, basieren die meisten in Deutschland verbreiteten Systeme auf der klassischen Client/Server-Struktur, bei der ein sog. Helpdesk als webbasierte Erweiterung dient. Oftmals gibt es Datenschnittstellen, beispielsweise zu ERP-Systemen, CAD-Systemen oder Kontaktdatenbanken. Auch zur GA werden Schnittstellen realisiert, aufgrund bisheriger Anforderungen jedoch nur zum unidirektionalen Datenaustausch ohne Steuerungsfunktion.

Als Gebäudeautomation bezeichnet man die gesamte MSR-Technik, mit der gebäudetechnische Einrichtungen gesteuert werden. Die Erfassung von Zählerständen, Betriebsstunden oder Verbrauchsdaten wird oftmals über eine Datenschnittstelle an CAFM-Systeme zwecks Statistiken und Benchmarks realisiert. Die GA beinhaltet auch technische Vorrichtungen für das operative Störfallmanagement. Als Komponenten der Gebäudeautomation sind Geräte (Sensoren und Aktoren) der Gewerke Heiztechnik, Klimatechnik und Lüftungstechnik (kurz HKL) sowie Elektrotechnik, Brandschutztechnik, ggf. auch Sanitärtechnik und weitere gebäudetechnische Sonderanlagen wie Sicherheitstechnik oder Medientechnik zu nennen. Im weiteren Verlauf dieser Arbeit ist die Analyse der Kommunikationsfähigkeit erforderlich.

Das Haupteinsatzgebiet derartiger Systeme sind größere Objekte wie Bürogebäude, Schulen, Krankenhäuser, Produktionsstätten oder Industrieanlagen allgemein. Hier ist der technische und durch Energieeinsparung ökonomische Nutzen am größten. Doch auch für den Hausgebrauch sind bereits Lösungen am Markt, sei es zur Festinstallation am Neu-/Umbau oder flexible Funkschalter zum Nachrüsten. Beides findet man oft schon mit einer produktbezogenen Software, die eine Bedienung über das Internet ermöglicht. Ob man hier schon von „webbasierter Steuerung“ sprechen kann, ist im Folgenden zu klären.

In diesem Zusammenhang ist auch zu erläutern, was unter dem Begriff „webbasiert“ verstanden wird. Webbasiert bedeutet, dass sich Applikationen über das World Wide Web bedienen lassen. Gebäudeautomationstechnologien, wie z. B. KNX, LON oder BACnet, lassen sich durch spezielle Schnittstellenhardware (sog. IP-Controller) an das

Computernetzwerk anbinden. Oftmals ist neben der geräteabhängigen Kopplung auch eine herstellerabhängige Softwareinstallation nötig, um über das Internet arbeiten zu können. Lösungsmodelle zur Fernsteuerung via Remotedesktop, TeamViewer oder RDSPlus, wie sie in der Applikation „myHomeControl“ (Version V1.2.8 SP11) zur Bedienung der Ratio©-Geräte der Omnio AG eingesetzt werden, stehen nicht in Einklang mit dem Gedanken der geräteunabhängigen Steuerung über einen Internetbrowser.

Bezogen auf die Kopplung der beiden Systeme, GA und CAFM, kann man feststellen, dass derzeitige Schnittstellen meist nur für die reine unidirektionale Datenübergabe ausgelegt sind, wie z. B. Zählerstände, Betriebsstunden oder Verbrauchsdaten. Die Frage ist nun, ob sich diese reine Datenübergabe um Steuerungsmechanismen erweitern lässt und was hardware- und softwareseitig dazu notwendig ist. Denn CAFM-Systeme agieren über (W)LAN-Verbindungen in einem reinen Computernetzwerk mit TCP/IP-Protokollen, während die GA üblicherweise über eigene Busleitungen kommuniziert. Es ist zu prüfen, welche Technologien es überhaupt für die Vernetzung von Geräten und Anlagen gibt und welche sich miteinander koppeln lassen. Hieraus leiten sich die Möglichkeiten von Schnittstellen zu CAFM-Systemen für Steuerungsfunktionen ab.

Interessant in diesem Zusammenhang ist eine neue Entwicklung im Bereich der IP-basierten Steuerung. In dem bei „Strompraxis – Sonderheft 2009“ erschienenen Artikel „digitalSTROM - Das Internet der elektrischen Geräte“ von Lukas Widmer wird über neue Ideen zu Vernetzungsvarianten und -technologien berichtet. Dabei geht es um die Entwicklung eines neuartigen Chips, der in Klemmen oder direkt in entsprechenden Endgeräten montiert wird. Dieser Chip hat einen eigenen Prozessor und eine eindeutige Identifikationsnummer, die vergleichbar mit einer IP-Adresse ist. Auch diese Variante wird im späteren Verlauf dieser Arbeit betrachtet, zusammen mit weiteren Systemen wie KNX, LON, LCN, EnOcean oder BACnet. Ob und wie sich diese GA-Technologien mit IP-basierten Computernetzwerken wie LAN (Ethernet/IP), WLAN oder PowerLAN kombinieren lassen, ist ebenfalls Gegenstand dieser Arbeit.

Im Vordergrund der Dissertation stehen Analysen, Vergleiche und Entwicklungen von Modellen, bei denen das Hauptaugenmerk auf der Kopplung von GA- und CAFM-Systemen liegen wird. Eine sinnvolle Kombination macht die Idee aus ganzheitlicher FM-Perspektive sehr interessant, sowohl technisch, kaufmännisch und infrastrukturell, wie auch der Ausblick auf mögliche Zertifizierungen hinsichtlich Green Building Zertifikaten.

2.2 Problemstellung

Aus Beobachtungen der derzeitigen Ausprägungen von Technologien im Bereich der Gebäudeautomation lassen sich konkret folgende Probleme erkennen:

Grundsätzlich fehlt es an methodischen Ansätzen für Kommunikationswege zum Zwecke der Anlagensteuerung aus CAFM-Systemen heraus. Dahinter verbirgt sich ein großes Optimierungspotenzial, das es auszuschöpfen gilt.

GA- und CAFM-Systeme sind sowohl netzwerktechnisch als auch datentechnisch getrennte Systeme. Dieser „Parallelbetrieb“ führt dazu, dass Anlagen- und Geräteattribute bei der Aufnahme in der Praxis mehrfach erfasst werden müssen, sowohl für den Bereich CAFM (kaufmännische Sichtweise) als auch den Bereich GA (technische Sichtweise), was eine redundante Datenhaltung zur Folge hat.

Die getrennten Vernetzungen verursachen einen hohen logistischen und infrastrukturellen Mehraufwand, was auch einen enormen Kostenfaktor darstellt. Eine einzige, einheitliche IT-Infrastruktur würde eine Menge Vorteile bieten, wie geringere Planungskosten, reduzierte Lebenszykluskosten, optimiertes Systemmanagement und einen verbesserten Service.

Im Bereich der Steuerungstechnologien der GA existiert bereits eine Vielzahl unterschiedlicher Systeme, die sich mit den gleichen Zielen parallel zueinander entwickeln. Entsprechende Unterschiede gibt es daher auch bei Busleitungen und Protokollen.

Nach derzeitigem Erkenntnisstand haben Gebäudeautomationssysteme keine konsequente, geräteunabhängige Browser-Ausrichtung. Es handelt sich meist um herstellerabhängige Einzellösungen in Form von lokalen Applikationen, die einige Dienste über das Internet bereitstellen. Folglich werden neue Ansprüche an die Gebäudeautomation gestellt.

Eine zentrale und universelle Steuerung gebäudetechnischer Einrichtungen über CAFM-Systeme ist bislang nicht möglich. Es existieren auch noch keine hersteller- und Gewerke übergreifenden Standards für Kopplungen oder Schnittstellen dieser Art.

Zudem fehlt es an Methoden einer komfortablen Einbindung, Installation und Konfiguration von gebäudetechnischen Anlagen. Auch dieser Punkt soll bei der Erstellung des Lösungskonzepts berücksichtigt werden. Was ist dazu nötig? Was müsste standardisiert werden? Welche Prozesse müssten modelliert und durchlaufen werden?

Der IST-Zustand lässt sich in folgender Grafik darstellen:

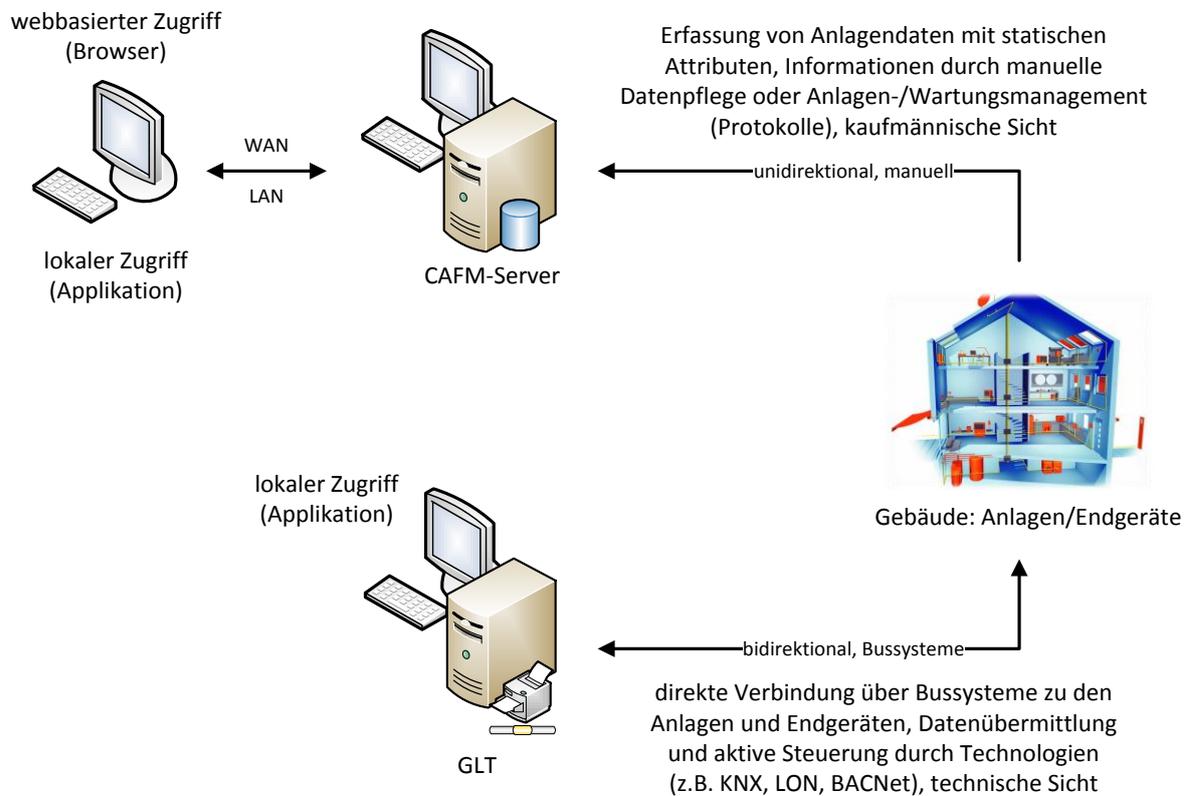


Abbildung 1: IST-Zustand, eigene Darstellung mit Bildmaterial der Initiative Intelligentes Wohnen, www.intelligenteswohnen.com

Der Server eines klassischen CAFM-Systems ist über ein Computernetzwerk mit den Clients verbunden. Der Zugriff erfolgt aus einer Applikation heraus (Client-Server-Variante) oder aus dem Internetbrowser heraus (webbasierte Variante).

Anlagendaten gebäudetechnischer Einrichtungen werden bei der Erstaufnahme manuell erfasst und später durch Informationen aus dem Anlagen- und Wartungsmanagement heraus manuell gepflegt. Bestenfalls gibt es eine Datenschnittstelle zum Übermitteln von Zählerständen, Betriebsstunden, Verbrauchsdaten oder Störmeldungen. Diese Informationen werden alle unidirektional übermittelt, also von der GA zum CAFM-System.

Die Verbindung zwischen gebäudetechnischen Anlagen und den Steuereinheiten bzw. einem Leitreechner als Zentrale zum Messen, Steuern und Regeln (MSR-Technik) erfolgt über separate Busleitungen. Datenaustausch und bidirektionale Kommunikation werden über Technologien wie KNX, LON oder BACnet realisiert.

2.3 Zielsetzung

Aus der Problemstellung heraus ergibt sich die Aufgabe, ein Konzept zu entwickeln, das den beschriebenen Kopplungsprozess von GA und CAFM abbildet, die redundante Datenhaltung auflöst und webbasierte Steuerungen zulässt. Diese Prozesse folgen einer Ablaufkette, wofür ein sog. Workflowmodell festzulegen ist. Daraus müssen ggf. neue Standards formuliert werden, damit eine Geräteanbindung unabhängig von den im Vorfeld betrachteten Netzwerktechnologien möglich ist.

Eine webbasierte Lösung mit Datenbankanbindung und Steuerungsmöglichkeit aus einem CAFM-System heraus ist ein neuartiger und innovativer Lösungsansatz.

Das Konzept soll eine Lösung der komfortablen Einbindung, Installation und Konfiguration gebäudetechnischer Anlagen beinhalten. Geräte sollen automatisch erkannt und in das CAFM-System integriert werden, eine Art „Plug and Play“ auf Netzwerk- bzw. Anlagenebene. Hierbei ist auch zwischen Entwicklungen bzgl. Hardware und Entwicklungen bzgl. Software zu unterscheiden.

Anschließend ist eine praktische Umsetzung durch die Programmierung einer Weboberfläche angestrebt, die diese Schnittstellenfunktionen mit Datenbankanbindung demonstriert.

Damit soll eine Grundlage geschaffen werden, um elektrische Verbraucher (wie beispielsweise Beleuchtungen, Jalousien, Pumpen, Heizungen oder auch Multimediageräte) steuern und bei Verfügbarkeit entsprechender Hardware auch das Raumklima regeln zu können – über ein CAFM-System, webbasiert und datenbankgestützt. Dies lässt sich erweitern auf die Sicherheitstechnik, u. a. die Kontrolle und Überwachung mit Kameras und Meldern, und andere gebäudetechnische Bereiche.

Dieser Lösungsansatz birgt einen weiteren sehr interessanten Aspekt und entscheidenden Vorteil. Mit der Datenbankanbindung kann die Aktivität eines Benutzers bzw. der Status eines Objekts in einem Datensatz mit Zeitstempel erfasst und archiviert werden. Man spricht dabei von einer sog. Historisierung. Dies ist in Bezug auf die Betreiberverantwortung ein sehr wichtiges und wesentliches Funktionsmerkmal. Weiterhin sollen Komfort, Sicherheit und Wirtschaftlichkeit durch Weiterentwicklungen ausgebaut werden.

Das Ziel dieser Arbeit ist also der Entwurf eines innovativen Netzwerkmodells zur Integration gebäudetechnischer Einrichtungen und computergestützter Netzwerke auf Automationsebene (bislang nur Managementebene). Um Anlagen und Geräte ohne separate Bus-Systeme über das Netzwerk webbasiert steuern und an eine CAFM-Datenbank anbinden zu können, bedarf es einer Standardisierung von Protokollen und Automatismen im Netzwerk. Diese sollen einen reibungslosen Datenimport von Geräteeigenschaften und Diensten für eine zentrale und komfortable Anlagenkonfiguration und -steuerung gewährleisten.

Durch diese Kopplung werden CAFM-Systeme zu leistungsstarken Steuerungsinstrumenten, um Gebäude aus technischer, kaufmännischer und infrastruktureller Sicht optimal und nachhaltig bewirtschaften zu können.

Der SOLL-Zustand lässt sich in folgender Grafik darstellen:

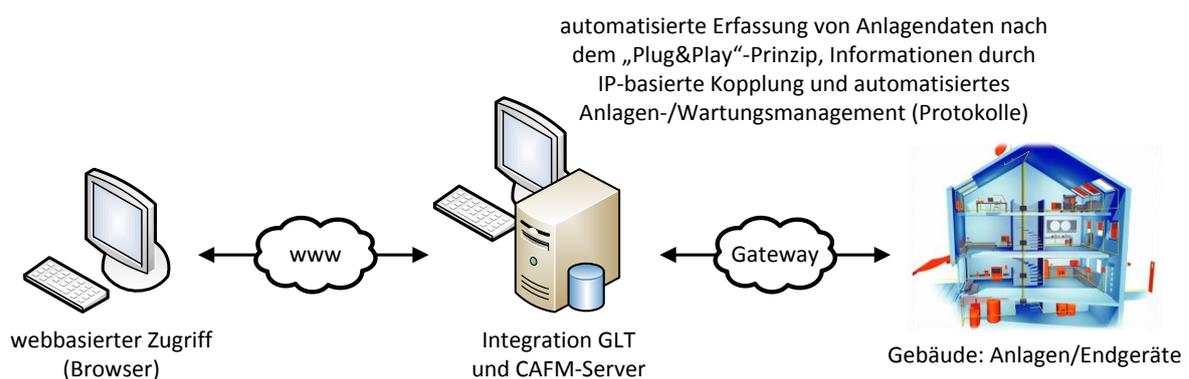


Abbildung 2: SOLL-Zustand, eigene Darstellung mit Bildmaterial der Initiative Intelligentes Wohnen, www.intelligenteswohnen.com

In diesem Modell sind die Funktionen des Leitrechners der GLT mit dem CAFM-System vereint. Dieses ist über ein Computernetzwerk mit den Clients verbunden. Der Zugriff der Clients erfolgt webbasiert über den Internetbrowser.

Anlagendaten gebäudetechnischer Einrichtungen werden bei der Installation nach dem „Plug and Play“-Prinzip automatisch erfasst und später durch Informationen aus der IP-basierten und bidirektionalen Kopplung automatisch aktualisiert und gepflegt. Der aktive Datenaustausch erfolgt über das Computernetzwerk, separate Busleitungen entfallen.

2.4 Vorgehensweise

Die Vorgehensweise wird mit einem Phasenmodell beschrieben, das sich in der Struktur der einzelnen Kapitel widerspiegelt. Sämtliche der sich aus den einzelnen Schritten ergebenden Fragestellungen sollen dabei aus FM-Sicht beantwortet werden. Das Phasenmodell ist wie folgt aufgebaut:

Nach einem kurzen Überblick über die Grundlagen von CAFM-Systemen, der Gebäudeautomation und der in diesem Zusammenhang maßgebenden Entwicklung des Internets stehen themenbezogene Normen und Richtlinien im Fokus der Betrachtung. Sie werden auf Definitionen oder Handlungsempfehlungen hinsichtlich Schnittstellen zur Steuerung gebäudetechnischer Anlagen hin geprüft.

Im nächsten Schritt wird eine Auswahl gängiger Technologien in der Gebäudeautomation und der Datenübertragung in Computernetzwerken vorgestellt. Es werden Grundlagen vermittelt, die zum Verständnis des weiteren Vorgehens notwendig sind. Darüber hinaus wird dargestellt, was in Sachen netzwerkbasierter Bedienung bereits realisiert und technisch umgesetzt wurde.

Anschließend werden zum Zwecke der Modellentwicklung für webbasierte CAFM-Steuerungsmechanismen unterschiedliche Netzstrukturen entworfen und analysiert. Für die Wahl eines optimalen Lösungskonzepts werden diese bewertet und verglichen. Nach Festlegung des hardwareseitigen Aufbaus unter Verwendung bereits standardisierter Netzwerkprotokolle nach dem „Plug and Play“-Prinzip folgt die Beschreibung der Automatismen im Netzwerk samt Workflow zur Umsetzung der softwareseitigen Funktionalitäten.

Zur Demonstration der Anwendbarkeit des webbasierten Schnittstellenkonzepts entsteht der Programmbaustein „FMControl“, wie er in gängigen CAFM-Systemen implementiert werden könnte. Der Entwurf einer Datenbankstruktur und die Ausarbeitung bereitzustellender Datenformate sind ebenso Gegenstand dieses Kapitels wie die Programmierung einer Benutzeroberfläche mit Importfunktion und Steuerungsmechanismen.

Zusammenfassend werden der entwickelte Standard und vorgestellte Programmbaustein sowie deren Nutzen für eine Schlussfolgerung bewertet. Die Arbeit schließt mit einem Fazit sowie dem Ausblick auf weiteren Forschungs- und Entwicklungsbedarf in dieser Hinsicht.

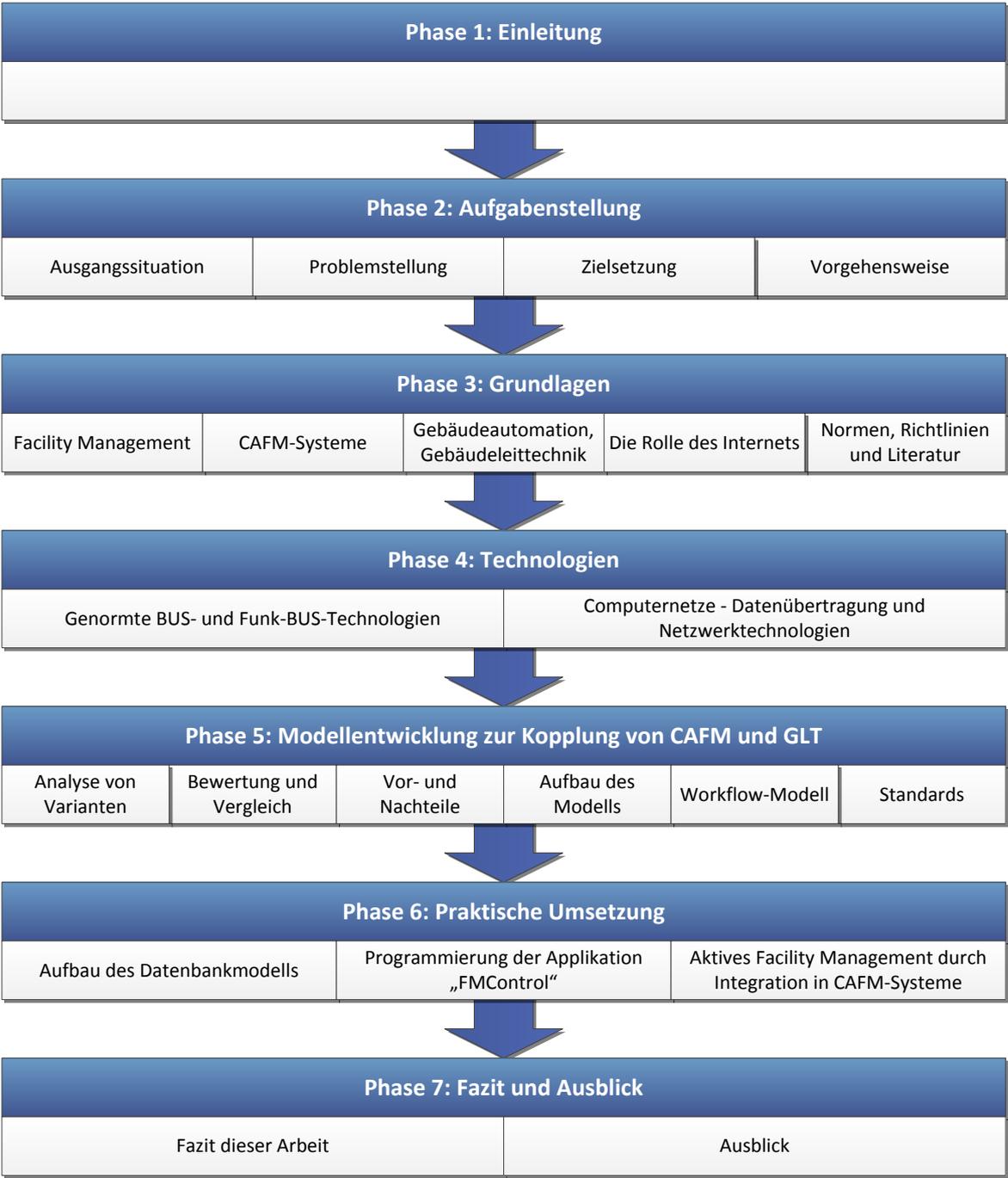


Abbildung 3: Phasenmodell, eigene Darstellung

3 Grundlagen

In diesem Kapitel werden Grundlagen, Begriffe und Definitionen erläutert, deren Bedeutungen im weiteren Verlauf dieser Arbeit als bekannt vorausgesetzt werden.

3.1 Facility Management

Was versteht man unter Facility Management? In der Literatur finden sich zahlreiche Definitionen, was ein Indiz für die Anwendungsvielfalt ist und die verschiedenen Sichtweisen widerspiegelt.

Der Deutsche Verband für Facility Management e.V. (GEFMA) beschreibt den Begriff wie folgt: „Facility Management ist die Betrachtung, Analyse und Optimierung aller kostenrelevanten Vorgänge rund um ein Gebäude, ein anderes bauliches Objekt oder eine im Unternehmen erbrachte (Dienst-)Leistung, die nicht zum Kerngeschäft gehört.“¹

Der internationale Verband IFMA definiert Facility Management als ein interdisziplinäres Berufsfeld, das die Funktionalität der geschaffenen Umwelt durch die Integration von Mensch, Ort, Prozess und Technologie gewährleistet.²

Als drittes Beispiel sei die Definition der DIN EN 15221-1 „Facility Management - Teil 1: Begriffe“ zitiert. Facility Management ist die „Integration von Prozessen innerhalb einer Organisation zur Erbringung und Entwicklung der vereinbarten Leistungen, welche zur Unterstützung und Verbesserung der Effektivität der Hauptaktivitäten der Organisation dienen“.³

Mit eigenen Worten formuliert, bezeichnet Facility Management die ganzheitliche Verwaltung und Bewirtschaftung von Gebäuden, Anlagen und Einrichtungen und es erstreckt sich über alle Bereiche des gesamten Lebenszyklus, von der Planung und Erstellung über die Nutzung bis hin zum Abriss. Den größten Anteil bilden dabei die Organisation und Steuerung sämtlicher Dienstleistungen während der Nutzungsphase. Diese umfassen technische Aufgaben, infrastrukturelle Leistungen und kaufmännische Bereiche.

¹ (German Facility Management Association, GEFMA-Richtlinie 100, 1996)

² (International Facility Management Association, IFMA - What is FM?, 2013)

³ (Deutsches Institut für Normung e.V., DIN EN 15221-1, 2007)

Man kann also von einer interdisziplinären Managementaufgabe sprechen, die „durch ergebnisorientierte Handhabung von Facilities und Services im Rahmen geplanter, gesteuerter und beherrschter Facility Prozesse eine Befriedigung der Grundbedürfnisse von Menschen am Arbeitsplatz, Unterstützung der Unternehmens-Kernprozesse und Erhöhung der Kapitalrentabilität bewirkt. Hierzu dienen die permanente Analyse und Optimierung der kostenrelevanten Vorgänge rund um bauliche und technische Anlagen, Einrichtungen und im Unternehmen erbrachte (Dienst-)Leistungen, die nicht zum Kerngeschäft gehören“.⁴

Die Aufgaben des Facility Managements sind in erster Linie das Bereitstellen von Flächen- und Gebäudedaten, das Instandhalten und Warten von Bauwerken und gebäudetechnischer Anlagen, die Kennzahlenermittlung zwecks Benchmarking, die Gebäudekostenzuordnung zur Kostenverfolgung sowie die Optimierung der Gebäudenutzung.

Der Grundgedanke des Facility Managements entstand Mitte der 1950er Jahre in den USA aus dem Bedürfnis heraus, die Produktivität in Betriebsführung und Instandhaltung zu verbessern, was bis Ende der 1970er konsequent umgesetzt wurde. Zu diesem Zeitpunkt wurden bereits Kongresse zu diesem Themenbereich abgehalten und Arbeitsgruppen zur Analyse der Grundlagen des Facility Managements eingesetzt. Nach Gründung der ersten nationalen und internationalen Verbände, gab es Mitte der 1980er Jahre die ersten Aktivitäten auch in Europa. In Deutschland hat sich Facility Management zu einer der bedeutendsten Dienstleistungsbranchen mit einer Bruttowertschöpfung von 112 Mrd. Euro⁵ entwickelt.

Als wichtigste berufsständische Institutionen und Verbände sind die 1982 gegründete International Facility Management Association (IFMA) und die 1989 in Deutschland gegründete German Facility Management Association e.V. (GEFMA) zu nennen. Weitere Institutionen und Verbände sind EuroFM (European Facility Management Association), RealFM (Association for Real Estate and Facility Managers), DIN (Deutsches Institut für Normung), VDMA (Verein Deutscher Maschinen- und Anlagenbau) und VBI (Verband Beratender Ingenieure), sowie GlobalFM als internationaler Dachverband.

⁴ (German Facility Management Association, GEFMA-Richtlinie 100-1, 2004)

⁵ (Thomzik, FM-Branchenreport, IAI e.V., 2010)

Die GEFMA befasst sich mit der Vereinheitlichung der Begriffe und Aktivitäten im Bereich des FM und hat seit 1996 ein kontinuierlich wachsendes Richtlinienwerk zur Etablierung der Branche geschaffen. Das Richtlinienwerk ist Basis für Branchenkonsens und qualitätsorientierte FM-Dienstleistungen. Es umfasst derzeit 75 Dokumente.

In den Richtlinien 100 (Facility Management – Begriff, Struktur, Inhalte) und 100-1 (Facility Management – Grundlagen) werden Gliederung, Anforderungen, Nutzen, Grundsätze und Prozessorientierung wie folgt dargestellt:

Gliederung des Facility Managements⁶:



Abbildung 4: Gliederung nach GEFMA-Richtlinie 100, eigene Darstellung

⁶ (German Facility Management Association, GEFMA-Richtlinie 100, 1996)

Nutzen des Facility Managements

- „Langfristiger Erhalt und Erhöhung von Vermögenswerten
- Verringerung der gebäude- und servicebedingten Kosten
- Entlastung des Managements und der Mitarbeiter in Sekundärprozessen
- Höhere Zufriedenheit durch optimale Arbeitsplatzgestaltung
- Höhere Produktivität des Unternehmens“⁷

Grundsätze des Facility Managements

- „Überspannung des gesamten Lebenszyklus‘ von Facilities
- Integration einzelner Serviceleistungen zu einem FM-Prozess
- Facility Management als Schnittstelle zum Kerngeschäft“⁷

Prozessorientierung des Facility Managements

- „Festlegung von unmissverständlichen Anforderungen an das FM durch den Kunden
- Vereinbarung von Service-Levels und Messgrößen
- Geplante, gesteuerte und beherrschte Facility-Prozesse durch den Dienstleister
- Ergebnisse entsprechen den Merkmalen der Kundenanforderungen
- Nutzenzuwachs für den FM-Kunden und Steigerung der Kundenzufriedenheit“⁷

Anforderungen an das Facility Management

- „Unterstützung von Kernprozessen
- Erhöhung der Leistungsfähigkeit betrieblicher Arbeitsplätze
- Gewährleistung von Sicherheit und Gesundheitsschutz für die Mitarbeiter
- Erhaltung baulicher und anlagentechnischer Werte
- Einhaltung gesetzlicher Vorschriften
- Erhöhung von Nutzungsqualitäten
- Reduzierung von Nutzungskosten“⁷

⁷ (German Facility Management Association, GEFMA-Richtlinie 100-1, 2004)

Die nachfolgende Darstellung verdeutlicht die unterschiedlichen Zeit-, Objekt- und Organisationshorizonte von Facility- und Gebäudemanagement:

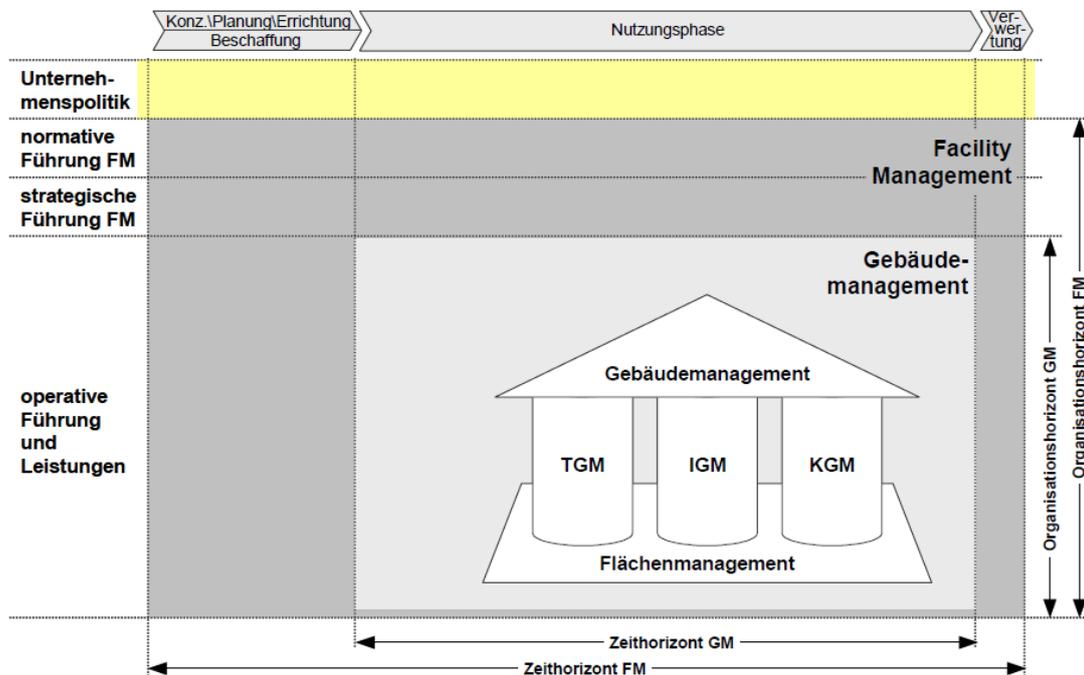


Abbildung 5: Abgrenzung Facility-/Gebäudemanagement, Quelle: GEFMA-Richtlinie 100-1, 2004

Das **Technische Gebäudemanagement (TGM)** „umfasst alle Leistungen, die zum Betreiben und Bewirtschaften der baulichen und technischen Anlagen eines Gebäudes erforderlich sind.“⁸ Dazu zählen Prozesse wie Anlagenmanagement, Wartungs-/Instandhaltungsmanagement, Energiemanagement, IT und Gebäudeautomation sowie Contracting.

Das **Infrastrukturelle Gebäudemanagement (IGM)** „umfasst die geschäftsunterstützenden Dienstleistungen, welche die Nutzung von Gebäuden verbessern.“⁸ Dazu zählen Prozesse wie Flächenmanagement, Umzugsmanagement, Reinigungsmanagement, Schlüsselmanagement, Sicherheit, Grün-/Außenpflege, Winterdienste, Fuhrparkmanagement, Ver-/Entsorgung und Catering.

Das **Kaufmännische Gebäudemanagement (KGM)** „umfasst alle kaufmännischen Leistungen aus den Bereichen TGM und IGM unter Betrachtung der Immobilienökonomie.“⁸ Dazu zählen Prozesse wie Auftragsmanagement, Vertragsmanagement, Buchhaltung, Controlling, Kostenabrechnung und Budgetierung.

⁸ (Deutsches Institut für Normung e.V., DIN 32736, 2000)

3.2 CAFM-Systeme / IWMS

Computer Aided Facility Management (CAFM) bezeichnet den Einsatz von IT-Werkzeugen zur computergestützten Verwaltung, Organisation und Steuerung von Arbeitsprozessen im Facility Management für ein effizientes Informations-, Prozess-, Entscheidungs- und Maßnahmenmanagement.

Eine CAFM-Software besteht aus einer Datenbank und einer Benutzeroberfläche. Diese Software dient zur Unterstützung der FM-Kernprozesse durch Datenerfassung, Datenverwaltung, Prozessoptimierung, transparente Abbildung kaufmännischer Vorgänge und Bereitstellung von alphanumerischen und grafischen Informationen über Gebäude und Objekte.

Doch wo ist CAFM in Bezug auf IT-Ansätze zur Unterstützung des Facility Managements einzuordnen? Alleine können CAFM-Systeme die Prozesse einer ganzheitlichen Gebäudebewirtschaftung nicht abbilden. Üblicherweise sind sie Kern einer IT-Lösung mit Schnittstellen zu beispielsweise Geoinformationssystemen (GIS), CAD-Systemen, ERP-Systemen (Finanzbuchhaltung, Controlling, Personalverwaltung, Dokumentenmanagement etc.) oder sonstigen Office-Anwendungen. Die Verbindung zur Gebäudeautomation wird erst im späteren Verlauf dieser Arbeit betrachtet.

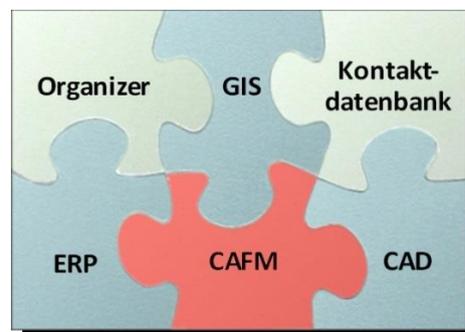


Abbildung 6: Beispiele von üblichen CAFM-Schnittstellen, eigene Darstellung

IT-Lösungen für das Facility Management werden in den USA als „Integrated Workplace Management System“ (IWMS) bezeichnet. Das Marktforschungsunternehmen Gartner definiert IWMS in einer Veröffentlichung zum Thema „Magic Quadrant for Integrated Workplace Management Systems“ (Mai 2012) als interdisziplinäre IT-Lösung mit folgenden fünf Kernbereichen:⁹

⁹ (Gartner, Magic Quadrant for Integrated Workplace Management Systems, 2012)

- Project management
(Projektplanung/-Steuerung von Bauprojekten)
- Space and facilities management (CAFM)
(Schwerpunkt Flächenmanagement)
- Real estate/property portfolio management and lease administration
(Immobilienwirtschaft)
- Computerized maintenance management systems (CMMS)
(IT-gestütztes Wartungs- und Instandhaltungsmanagement)
- Sustainability/facility optimization and compliance
(Optimierung des Energieverbrauchs, Nachhaltigkeit)

Nach dieser Definition ist CAFM nur ein Teil des ganzheitlichen IWMS-Gedanken. Der in Deutschland verbreitete Ansatz basiert auf einer Integrationslösung mit CAFM als Kernapplikation, die im weiteren Verlauf dieses Unterkapitels fokussiert wird.

Die meisten CAFM-Softwareprodukte sind modular aufgebaut. Diese Module sind dabei so strukturiert, dass der Anwender themenbezogen und modulübergreifend arbeiten kann. Inhaltlich orientieren sich die Module an den im Folgenden aufgeführten FM-Kernprozessen.

Auch für diesen Sektor des Facility Managements stellt die GEFMA Richtlinien bereit, welche die Anwendungsbereiche und Leistungsmerkmale von CAFM beschreiben. So werden in der GEFMA-Richtlinie 400 im Rahmen der Definition von Funktionalitätsanforderungen folgende FM-Kernprozesse genannt:¹⁰

- Datenmanagement
- Flächenmanagement
- Inventarmanagement
- Raumreservierung
- Reinigungsmanagement
- Umzugsmanagement
- Energiecontrolling
- Instandhaltungsmanagement
- Schließenanlagenmanagement
- Vertragsmanagement
- Mietmanagement
- Betriebskostenabrechnung
- Controlling
- Sicherheit und Arbeitsschutz
- Umweltschutz

¹⁰ (German Facility Management Association, GEFMA-Richtlinie 400, 2013)

Welchen Nutzen und welche Vorteile hat die Einführung eines CAFM-Systems für den Anwender? Dass professionelles Facility Management aufgrund der großen Datenmengen und komplexen Prozessen nicht ohne den Einsatz von Computern realisierbar ist, liegt auf der Hand. Die GEFMA-Richtlinie 400 erläutert den Nutzen eines CAFM-Systems in Kapitel 3: Die Effizienzsteigerungen und Nutzenpotenziale durch die Implementierung eines CAFM-Systems sind vielseitig. Dazu zählen die Sicherstellung der Funktionsfähigkeit und Verfügbarkeit von Facilities sowie Werterhalt und Wertsteigerung der baulichen Objekte und technischen Anlagen bei absoluter Kostentransparenz. CAFM-Systeme gewährleisten einen hohen Informationstand durch die zentrale Bereitstellung von aktuellen Daten als Entscheidungsgrundlage für das Geschäftsprozessmanagement unter Berücksichtigung geltender Gesetze, Normen und Richtlinien. Der Nutzen ist abhängig von Zustand, Größe und Komplexität der Facilities, Anzahl der eingebundenen Mitarbeiter sowie Nutzerstruktur und Betreiberkonzept. Der GEFMA-Arbeitskreis CAFM verweist zur genaueren Untersuchung der Wirtschaftlichkeit einer CAFM-Einführung oder einem Systemwechsel auf die GEFMA-Richtlinie 460 und das CAFM-Handbuch.¹¹

Ob der Einsatz einer CAFM-Software lohnend ist oder nicht, lässt sich nicht pauschal beantworten. Dem Nutzen sind Aufwand und Kosten gegenüberzustellen. Dabei zählen Aspekte wie Transparenz, Verfügbarkeit, Zeitersparnis, Kostenreduktion, Ressourceneinsparung und Geschäftsfelderweiterung zu den Nutzeffekten eines CAFM-Systems. Dagegen stellt der Aufwand für Datenerfassung, Datenaufbereitung und Datenpflege zusammen mit den Investitionskosten mitunter eine Hürde für den Einsatz dar. Für die Berechnung der Wirtschaftlichkeit gibt es verschiedene Methoden. Die GEFMA-Richtlinie 460 bietet eine Hilfestellung bei der Analyse möglicher Nutzeffekte und Kosten für CAFM-Einführung und -Betrieb sowie bei der Berechnung der Wirtschaftlichkeitskennzahlen.

Ein wichtiger Punkt ist die Betrachtung der bereits genannten Schnittstellen. Im Februar 2013 wurde eine von Prof. Dr.-Ing. Joachim W. Hohmann initiierte und geleitete Trendstudie in Kooperation mit dem GEFMA-Arbeitskreis CAFM veröffentlicht. Interessant waren die Angaben der Umfrageteilnehmer zu der Frage nach realisierten Schnittstellen. Etwa ein Drittel der Unternehmen gibt an, dass ihre CAFM-Systeme eine Schnittstelle zur Gebäudeautomation haben.¹² Dabei ist zu beachten, dass es sich bei diesen Lösungen ausschließlich um reine Datenschnittstellen (unidirektional) handelt, bei denen sich die Gebäudeautomation nicht aktiv steuern lässt.

¹¹ (German Facility Management Association, GEFMA-Richtlinie 400, 2013)

¹² (Hohmann, Fachvortrag CAFM-Trendstudie, 2013)

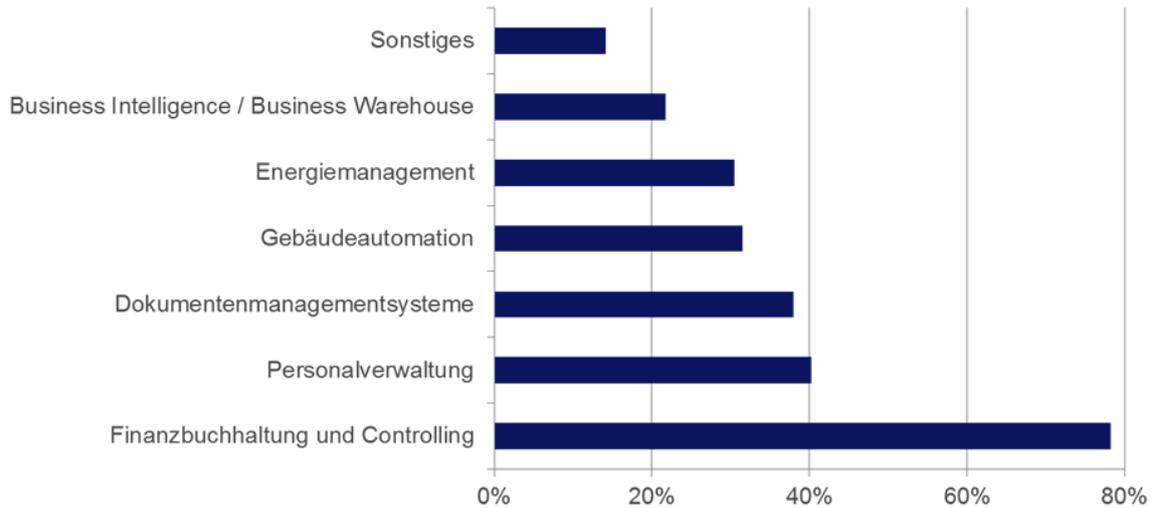


Abbildung 7: Realisierte Schnittstellen von CAFM-Systemen, Quelle: Hohmann, Fachvortrag CAFM-Trendstudie 2013

Ein Viertel der Anwender gibt an, seinen CAFM Einsatz auf den Bereich Energiecontrolling ausweiten zu wollen.¹³ Die Frage nach einer bidirektionalen Schnittstelle zur GA wurde zwar nicht gestellt, doch hätte eine Steuerungsfunktion aus einem CAFM-System heraus einen bedeutenden Mehrwert für den Punkt „Energiecontrolling“. Mit dieser Funktionserweiterung würde man dem Begriff „Controlling“ seiner wortwörtlichen Bedeutung nachkommen: man überwacht nicht nur, man kann auch aktiv steuern.

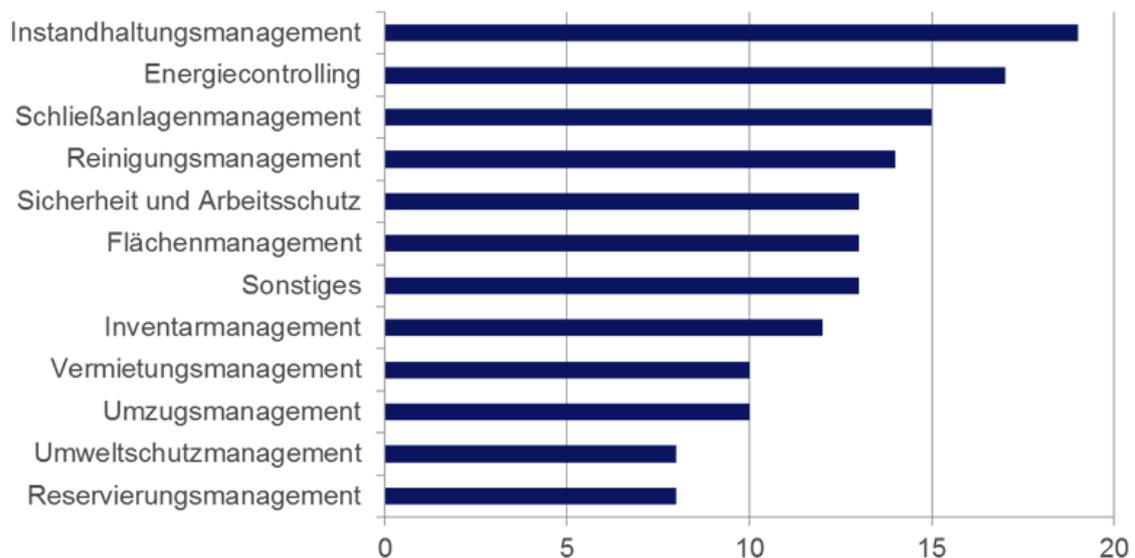


Abbildung 8: Ausweitung der Einsatzbereiche von CAFM, Quelle: Hohmann, Fachvortrag CAFM-Trendstudie 2013

¹³ (Hohmann, Fachvortrag CAFM-Trendstudie, 2013)

In der folgenden Tabelle ist eine Auswahl an CAFM-Herstellern mit ihren Produkten aufgelistet. Die Auswahl beruht auf den nach der GEFMA-Richtlinie 444 derzeit zertifizierten CAFM-Softwareprodukten.¹⁴

Unternehmen	Software
ARCHIBUS Solution Centers Germany GmbH	ARCHIBUS
Axxerion Facility Services	Axxerion
BFM GmbH	KeyLogic
Byron Informatik AG	Byron/BIS
conject AG	conjectFM
EBCsoft GmbH	VITRIcon
decon Deutsche Energie-Consult GmbH	deconFM
eTASK Service Management GmbH	eTASK FM-Portal
EuSIS GmbH	sMotive Web Portal
IBM Deutschland GmbH IC information company	IBM TRIRIGA
iffm Institut für Facility Management GmbH	iffmGIS
IMS Gesellschaft für Informations-und Managementsysteme mbH	IMSSware
Keßler Real Estate Solutions GmbH	FAMOS
Loy & Hutz	visualFM
Loy & Hutz	Waveware
N+P Informationssysteme GmbH	SPARTACUS
Nemetschek Allplan Systems GmbH	Allplan Allfa
pit-cup GmbH	pit FM
Planon GmbH	Planon Enterprise Solution

Tabelle 1: GEFMA-zertifizierte CAFM-Unternehmen und -Software, Quelle: GEFMA-Richtlinie 940

Eine vollständige Auflistung von GEFMA-zertifizierten CAFM-Softwarelösungen ist in der GEFMA-Richtlinie 940 zu finden. Diese Marktübersicht erscheint in einer jährlich veröffentlichten Sonderausgabe des Fachmagazins „Der Facility Manager“.

¹⁴ (German Facility Management Association, GEFMA-Richtlinie 940, 2012)

Richten wir den Blick auf die technischen und infrastrukturellen Eigenschaften von CAFM-Systemen. Die Anwendungen müssen multiuserfähig sein, also einen gleichzeitigen Zugriff mehrerer Benutzer von unterschiedlichen Orten auf die gleiche Datenbasis ermöglichen.¹⁵

Im klassischen Fall ist die Netzwerkstruktur als **Client/Server-Lösung** aufgebaut. Auf dem vom Nutzer betriebenen Server sind Datenbank und Dienste bereitgestellt. Auf den Netzwerk-Clients ist eine entsprechende Applikation lokal installiert. Die Kommunikation mit dem Server, auf dem alle gemeinsam genutzten Komponenten zentral abgelegt sind, wird über benutzerabhängige Zugriffsrechte geregelt. Für Datenbankabfragen sendet die Clientanwendung SQL-Befehle über sog. ODBC-Schnittstellen an den Datenbankserver. SQL (Structured Query Language) ist eine standardisierte Abfragesprache, ODBC (Open Database Connectivity) ist eine „Methode zum Zugriff auf verschiedene Datenbanksysteme über den jeweils passenden Treiber.“¹⁶ Bei Client/Server-Strukturen ist die Ausprägung des Netzwerks nicht begrenzt. Der Einsatz von CAFM-Systemen erfolgt zwar überwiegend im firmeneigenen Netzwerk, doch es lassen sich auch Fernzugriffe, z. B. über VPN-Verbindungen (Virtual Private Network), realisieren.

Die folgende Abbildung zeigt ein Client/Server-Modell in eigener Darstellung.

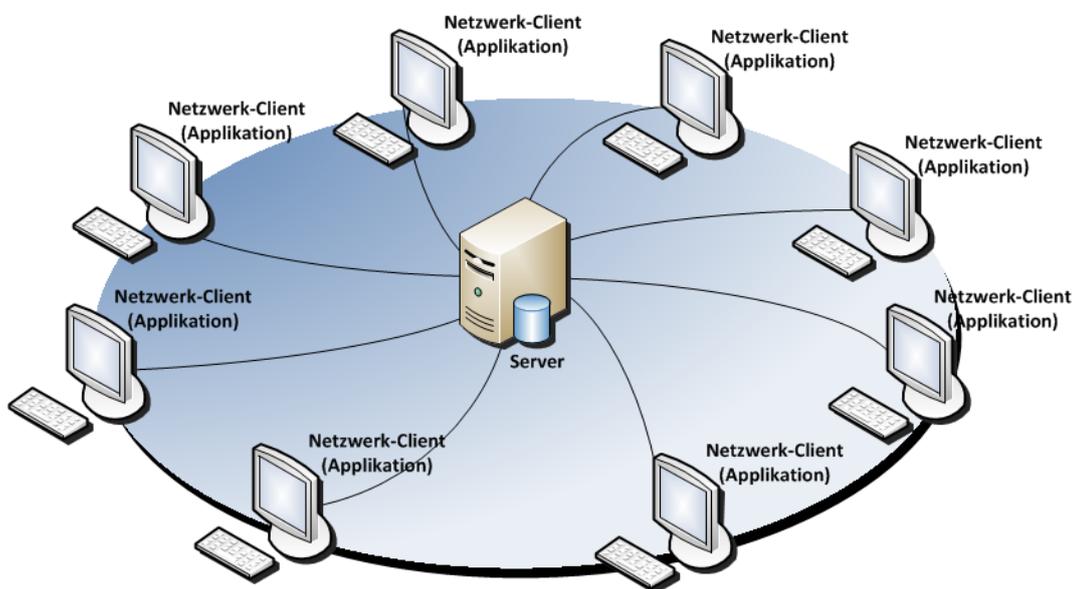


Abbildung 9: Client/Server-Modell, eigene Darstellung

¹⁵ (May, CAFM-Handbuch, 2013)

¹⁶ (Theis, Einstieg in Visual Basic 2008, 2008)

Der Trend folgt den Anforderungen hinsichtlich **webbasierter Lösungen**, wobei es aktuell nur wenige Hersteller gibt, die bereits ein „echtes“ vollständig webbasiertes CAFM-System in Deutschland anbieten. Meist ist nur ein sog. Helpdesk die einzige, über einen Internetbrowser nutzbare Webfunktion. Bei einem webbasierten CAFM-System wird neben der Datenbank lediglich ein Internetbrowser zum Eingeben und Abrufen der Daten benötigt, die lokale Installation einer Applikation auf dem Client entfällt. Durch Eingabe der entsprechenden Web-Adresse (URL) kann die Oberfläche des CAFM-Systems wie eine beliebige Website geöffnet werden. Die Kommunikation erfolgt über ein firmenweites Intranet (Local Area Network) oder über das weltweite Internet (Wide Area Network).

Die folgende Abbildung zeigt eine webbasierte Lösung in eigener Darstellung:

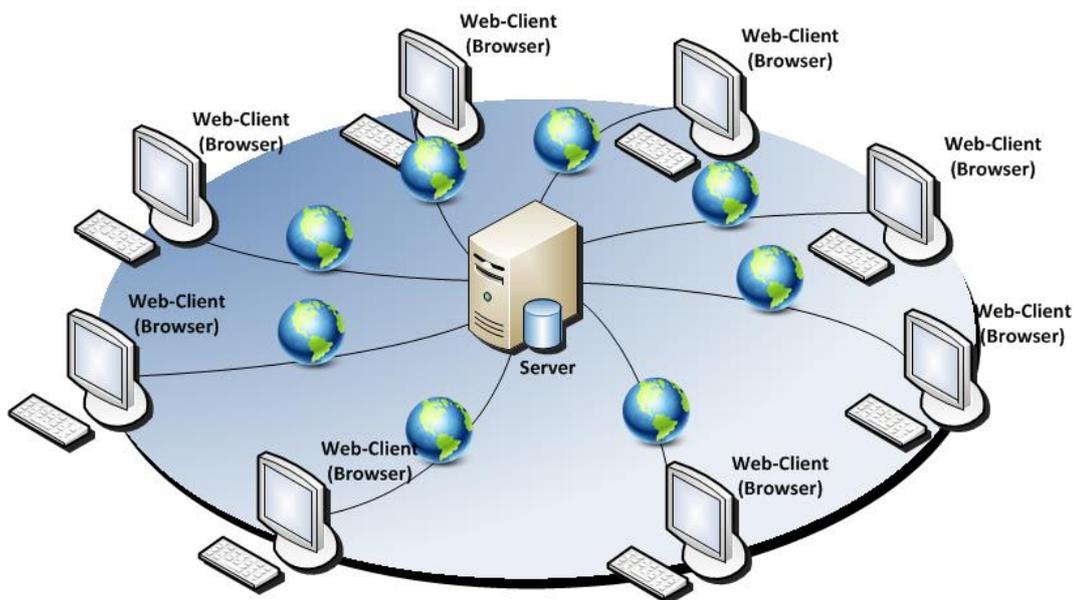


Abbildung 10: Webbasierte Lösung, eigene Darstellung

In jüngster Zeit erfreut sich eine dritte Variante immer größeren Zuspruchs, deren Einfluss auch Auswirkungen auf CAFM-Systeme haben wird – das neue IT-Trendthema „**Cloud Computing**“. Im Vergleich zur webbasierten Lösung werden statt eines vom Nutzer betriebenen Servers komplette und bedarfsorientierte IT-Infrastrukturen in Form virtueller Rechen-, Netzwerk- und Speicherkapazitäten, Software oder Dienstleistungen bereitgestellt.

„Grundgedanke des Cloud Computing ist, dass alle Anwendungen“ und Daten „im Web bereitgestellt werden – ob einzelne Software oder aber ganze Betriebssysteme. Die Cloud ist das World Wide Web, Cloud Computing bedeutet also – frei übersetzt –, das Internet als PC zu nutzen. Die Nutzung von IT-Leistungen erfolgt je nach aktuellem Bedarf über Datennetze ‚in der Wolke‘ anstatt auf lokalen Rechnern.“¹⁷

Bei der Bereitstellung von Hardware, Software und Serviceleistungen über das Netzwerk gibt es verschiedene Ausprägungen: „Während beim sogenannten Application Service Providing (ASP) Anbieter eine auf den jeweiligen Kunden zugeschnittene Applikation offerieren, geht ‚Software as a Service‘ (SaaS) einen Schritt weiter, indem Standard-Applikationen via Internet einem breiten Publikum zur Verfügung gestellt werden. ‚Infrastructure as a Service‘ (IaaS) stellt Rechnerinfrastruktur (Rechenkapazität, Speicherplatz) über das Internet bereit, während ‚Platform as a Service‘ (PaaS) Online-Entwicklungstools zur Verfügung stellt. Cloud Computing ist quasi alles zusammen.“¹⁸ Weitere, detaillierte Informationen liefert der Leitfaden „Cloud Computing“ von BITKOM (Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V.).

Die folgende Abbildung zeigt ein Cloud Computing-Modell in eigener Darstellung:

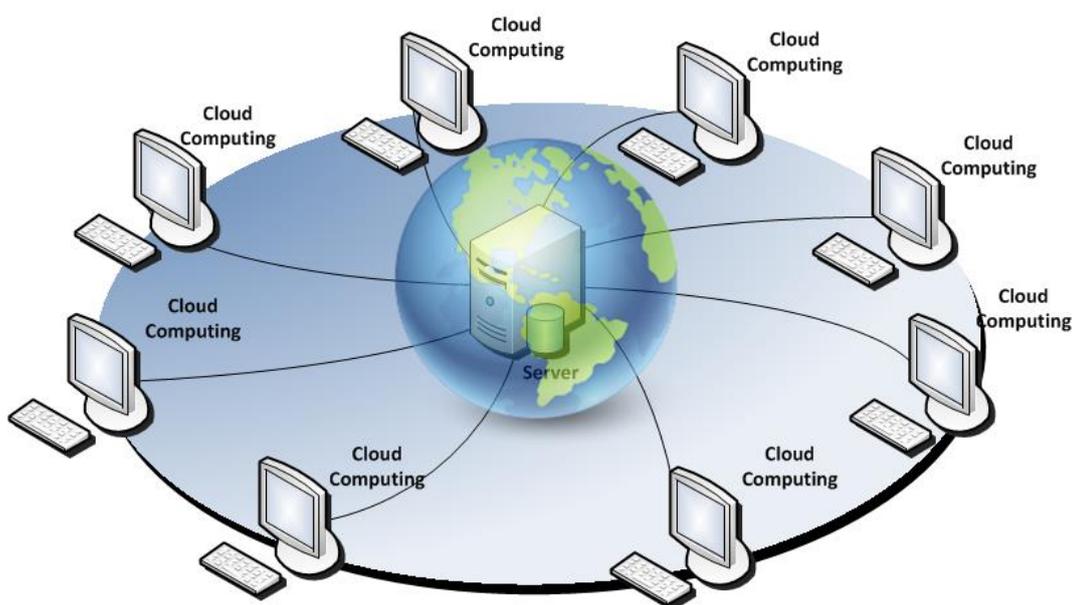


Abbildung 11: Cloud Computing, eigene Darstellung

¹⁷ (Silies, Cloud Computing: Die Zukunft liegt in den Wolken, 2012)

¹⁸ (Behaneck, Cloud Computing - In der Wolke rechnen, 2012)

3.3 Gebäudeautomation

Was bedeutet Gebäudeautomation und mit welchem Ziel wird sie eingesetzt? Dieses Kapitel widmet sich der Definition des Begriffs und stellt die wichtigsten Funktionen dar.

Die technische Ausstattung von Gebäuden steigt stetig. Nicht nur in Summe verbauter Anlagenkomponenten, sondern auch im Grad der Komplexität. Dabei geht es um gebäudetechnische Einrichtungen aus den Bereichen Heiz-, Klima- und Lüftungstechnik (HKL) sowie Elektrotechnik (speziell Beleuchtung/Beschattung) und Brandschutztechnik. Aber auch Sanitärtechnik oder weitere gebäudetechnische Sonderanlagen wie beispielsweise Sicherheitstechnik oder Medientechnik sind im Feld der technischen Gebäudeausrüstung (TGA) zu finden. Diese Gewerke sind in einer anlageübergreifenden Vernetzung durch Bussysteme mit intelligenten Gebäudefunktionen miteinander verbunden.

1965 wurde der Begriff „Gebäudeautomation“ eingeführt. Zwei Jahre später war er auch erstmals in einer deutschen Fachliteratur zu finden, einem HLH-Aufsatz (Fachbeitrag zum Themengebiet Lüftung/Klima, Heizung/Sanitär und Gebäudetechnik) von Dipl.-Ing. Hollmann, Ingenieurbüro, Frankfurt. Knapp 30 Jahre später wurde der Begriff mit der DIN V 32734 (1992) und mit der DIN 276 Kosten im Hochbau, Kostengruppe 480 (1993) offiziell in der Normung eingeführt. Endgültig definiert wurde er in der DIN EN ISO 16484 vom technischen Komitee CEN/TC 247 Gebäudeautomation und Gebäudemanagement.¹⁹

Nach DIN EN ISO 16484-2 bezeichnet die Gebäudeautomation „Einrichtungen, Software und Dienstleistungen für automatische Steuerung und Regelung, Überwachung und Optimierung sowie für Bedienung und Management zum energieeffizienten, wirtschaftlichen und sicheren Betrieb der Technischen Gebäudeausrüstung“.²⁰

In der von VDI und GEFMA gemeinsam verfassten Richtlinie 3814, Blatt 3.1 ist die Gebäudeautomation definiert als „Automatische Regelung, Steuerung, Überwachung, Optimierung, Betrieb sowie manuelle Eingriffe und Management durch Funktionen von Geräten, Stationen, Einheiten und Peripheriegeräten der technischen Gebäudeausrüstung.“²¹

¹⁹ (Siemens Building Technologies, Gebäudeautomation – Begriffe, Definitionen und Abkürzungen, 2003)

²⁰ (Deutsches Institut für Normung e.V., DIN EN ISO 16484-2, 2004)

²¹ (Verband Deutscher Ingenieure e.V., VDI/GEFMA-Richtlinie 3814, Blatt 3.1, 2012)

Die Gebäudeautomation (GA) wird als Gesamtheit der MSR-Technik in Gebäuden verstanden, die von einer zentralen, rechnergestützten Leitstelle aus verwaltet wird und zur Erfassung von Betriebsdaten dient. GA ist heute zu einem unverzichtbaren Werkzeug geworden, die eine immer bedeutendere Schlüsselfunktion im Gebäudebetrieb einnimmt. „Ziel ist es, Abläufe übergreifend selbstständig, nach vorgegebenen Einstellwerten durchzuführen oder deren Bedienung bzw. Überwachung zu vereinfachen.“²²

In welcher Form und zu welchen Zwecken wird die Gebäudeautomation innerhalb des technischen Gebäudemanagements eingesetzt? Die Argumente für den Einsatz solcher Anlagen liegen auf der Hand. Der Komfort wird verbessert (intelligente Steuerungen, automatische Regulierungen, Fernbedienung etc.), die Sicherheit erhöht (Präsenz- und Bewegungsmelder, Maximalwertwächter, Einbruch-, Brand- und Rauchmeldesysteme etc.), der Energieverbrauch reduziert (kontrollierte Be- und Entlüftung, Regelung der Raumtemperatur, tageslichtabhängige Lichtsteuerung etc.) und damit die Wirtschaftlichkeit optimiert.

Diese Ziele, insbesondere eine energieeffiziente Nutzung, können aber nur durch eine intelligente Regelung und Steuerung erreicht werden. Die Gebäudeleittechnik (GLT) ist integraler Teil der GA. Die Benutzeroberfläche der GLT-Software bildet die Schnittstelle zwischen Mensch und technischer Gebäudeausrüstung (TGA) und unterstützt bzw. ermöglicht je nach Grad der Automation folgende wesentliche Leistungen des Technischen Gebäudemanagements (TGM).²³

Diese Leistungen werden in der VDI/GEFMA-Richtlinie 3814 Blatt 3.1 wie folgt beschrieben:

- Betreiben inklusive Instandhaltung, Inspektion, Wartung, Instandsetzung, Verbesserung
- Dokumentieren: Mit der GA werden in Abhängigkeit von den implementierten Messgrößen auch Messwerte erfasst und gegebenenfalls archiviert. Diese Werte stehen als Grundlage für Analysen oder für die Protokollierung im Rahmen von Wartungs- und Inspektionstätigkeiten sowie für die Verarbeitung in umgebenden Systemen zur Verfügung.
- Meldungen und Berichte: Aus den erfassten Messwerten und Daten generiert die GA Meldungen und Berichte zur Auswertung. Dazu zählen das Meldungs

²² (Deutsches Institut für Normung e.V., Dossier: Energieeffizienz, 2013)

²³ (Verband Deutscher Ingenieure e.V., VDI/GEFMA-Richtlinie 3814, Blatt 3.1, 2012)

management zur Veranlassung von Maßnahmen, wie Betriebsmeldungen, Störungsmeldungen oder Alarmer/Alarmverarbeitung, und das Berichtswesen z. B. zum Nachweis der Erbringung von zugesicherten Leistungen (z. B. Anlagenverfügbarkeit im Zusammenhang mit SLA/(K)PI) oder als Entscheidungsgrundlage (z. B. Ersatzinvestition für den Austausch von Anlagen oder Komponenten), wie Verfügbarkeitsberichte oder Energieberichte.

- Energiemanagement: EN 16001 und GEFMA 124 sind Leitfäden für die Einführung bzw. Durchführung eines Energiemanagements. In der GA abgebildete Funktionen des Energiemanagements sind das Lastmanagement und das Energiecontrolling.

„Gebäudeautomation ist aber nicht nur attraktiv weil sie Energiekosten einspart. Dank kurzer Amortisationszeiten zahlen sich die notwendigen Investitionen schnell aus, besonders im Vergleich mit anderen Energieeffizienzmaßnahmen.“²⁴ Folgende Grafik zeigt beispielhaft einige dieser Maßnahmen, in Abhängigkeit der jeweiligen Kategorie.

	Kategorie	Massnahmen, z.B.	Einsparpotential [%]	Amortisation [Jahre]
	Gebäude Automation	<ul style="list-style-type: none"> ▪ Installation und Optimierung der Energiefunktionen ▪ Optimierung im Betrieb durch <ul style="list-style-type: none"> ▪ effizienten Einsatz der GA und Schwachstellenanalyse ▪ Dynamisches Energiemanagement 	5-30	0-5
	Technische Installationen	<ul style="list-style-type: none"> ▪ HLKS, Kälte, Beleuchtung ▪ Steuerungen, Motoren, Antriebe ▪ Stromerzeugung 	10-60	2-10
	Gebäudehülle	<ul style="list-style-type: none"> ▪ Dämmung, Fenster ▪ Wärmebrücken, Bauphysik 	>50	10-60

Abbildung 12: Energieeffizienz durch Gebäudeautomation, Quelle: Siemens Building Technologies

Technisch werden die Komponenten der GA mit Hilfe von Bus-Systemen physikalisch vernetzt und durch die Interaktion mit Kommunikationsprotokollen logisch zu einem Gesamtsystem zusammengefasst. Der grundlegende Aufbau eines Gebäudeautomations-systems gliedert sich in drei logische Funktionsebenen: Managementebene, Automations- und Feldebene. „Die einzelnen Ebenen sind je nach Größe und Komplexität des Objekts bzw. der zu automatisierenden Liegenschaften ausgeprägt. Aufgrund des Fortschrittes in der digitalen Steuerungs- und Regelungstechnik verwischen die Grenzen der einzelnen Ebenen zunehmend. Insbesondere Funktionen der Automationsebene wurden mit

²⁴ (Siemens Building Technologies, Optimierung der Gebäudeenergieeffizienz, 2013)

leistungsfähigeren digitalen Systemen zunehmend dezentralisiert, d.h. in die Feldebene integriert.“²⁵

- Die gebäudetechnischen Anlagen und Steuerungseinheiten sind über Bussysteme mit einer zentralen, rechnergestützten Leitstelle vernetzt. Der sog. Leitrechner bildet die **Managementebene** ab, in der sämtliche Daten zusammenlaufen. Dahinter verbirgt sich ein Server mit Datenbanksystem und einer vom Hersteller der Bedieneinheiten gelieferten Software. Von hier wird die komplette Haustechnik visualisiert, überwacht und gesteuert. Bei dieser automatisierungstechnischen Instrumentarisierung spricht man von der sog. Gebäudeleittechnik (GLT). Sie ist damit integraler Teil der Gebäudeautomation.
- Die **Automationsebene** liegt zwischen der Feld- und Managementebene. Sie regelt den Informationsaustausch zwischen den einzelnen Komponenten der gesamten Anlagentechnik und übernimmt Steuer- und Regelungsaufgaben durch elektronische Baugruppen, der sog. DDC-GA (Direct Digital Control - Gebäudeautomation) bei vorgegebenen Ereignissen, wie z. B. Schaltzuständen, Grenzwert-/Höchstlastüberschreitungen oder Zeitintervallen. Dabei kommunizieren diese Komponenten über herstellerspezifische Bussysteme oder herstellerunabhängige Standards, wie KNX, LON und BACnet. Der Gedanke der Webfähigkeit spiegelt sich in der Entwicklung von BACnet/IP fähigen DDC-GA-Komponenten mit einem integrierten Webserver wider.
- In der **Feldebene** befinden sich die zum Messen, Steuern und Regeln benötigten Feldgeräte. Dabei handelt es sich um sog. Sensoren (Messwert- und Kontaktgeber) und Aktoren (Schalt- und Stellgeräte).²⁶ Sensoren dienen der Erfassung und Übermittlung von Betriebszuständen gebäudetechnischer Anlagen, Messwerten (Temperatur, Druck, Feuchte, Niederschlag, Wind, Bewegung) Zählerständen, Betriebsstunden oder Verbrauchswerten. Aktoren dienen der, teilweise auch messwertabhängigen, Steuerung der anlagentechnischen Komponenten wie Motoren, Pumpen, Ventile, Lüftungsklappen oder Schalterstellungen. Sensor- und Aktor-Funktionen sind in der VDI-Richtlinie 3813, Blatt 2 „Gebäudeautomation (GA) - Raumautomationsfunktionen (RA-Funktionen)“ im Einzelnen spezifiziert. Mit entsprechenden Bedien- und Anzeigeeinheiten können die Betriebsdaten überwacht und gesteuert werden. Da die Webfähigkeit derzeit in den DDC-GA-Komponenten der

²⁵ (BauNetz Media GmbH, Baunetzwissen Elektro - Ebenen der Gebäudeautomation, 2013)

²⁶ (Verband Deutscher Ingenieure e.V., VDI/GEFMA-Richtlinie 3814, Blatt 3.1, 2012)

Automationsebene liegt, ist für die weitere Betrachtung eine Prüfung der webbasierten Kommunikationsfähigkeit auf Feldebene erforderlich.

Die nachstehende Abbildung zeigt den Aufbau der Funktionsebenen in eigener Darstellung:

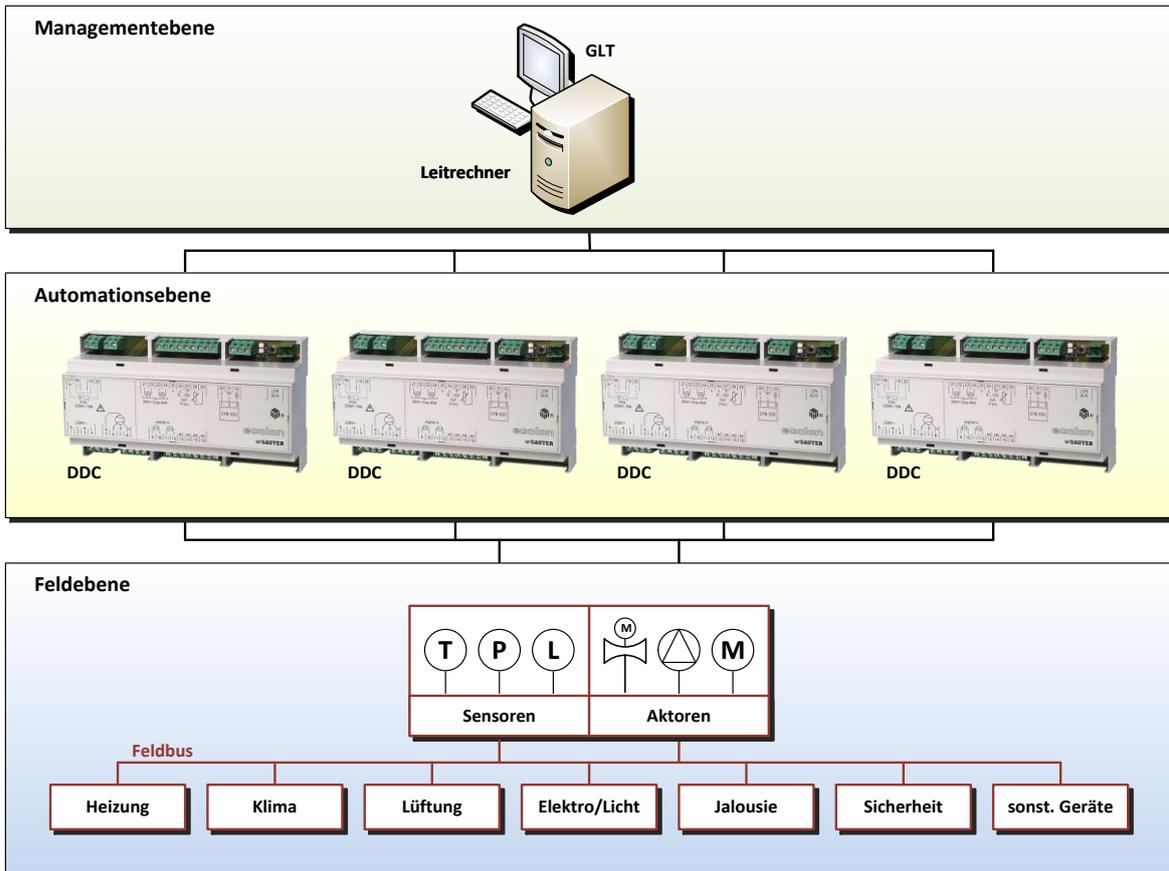


Abbildung 13: Funktionsebenen der Gebäudeautomation, eigene Darstellung mit Bildmaterial aus Online-Produktkatalog Fa. Sauter AG

Nach derzeitigen Definitionen (siehe Normen und Richtlinien in Kapitel 3.5) gibt es nur in der Managementebene, von der GLT aus, Schnittstellen zum (Computer Aided) Facility Management. In der GLT gesammelte Daten werden an CAFM-Systeme übergeben und zur Generierung von Wartungs- oder Serviceaufträgen sowie der Berechnung von Gebäudekosten genutzt. Die bei Abweichungen vom Regelbetrieb automatisiert ausgelösten Störmeldungen und Reparaturaufträge dienen der frühzeitigen Erkennung von Störungen und Fehlern und damit der Zuverlässigkeit und Verfügbarkeit von Anlagen.

Die Ansätze dieser Arbeit werden das Integrationspektrum von der Managementebene auf die Automationsebene erweitern.

3.4 Middleware

Durch die Vielzahl an Softwarelösungen im FM-Bereich und Automationstechnologien, die innerhalb eines Unternehmens zum Einsatz kommen können und miteinander kommunizieren müssen, gibt es eine große Menge an potenziellen Kombinationsmöglichkeiten. Rechnerisch gäbe es $m \cdot n$ Varianten, wenn jede Software eine Schnittstelle zu jedem System der Gebäudeautomation hätte. Mit Hilfe sog. Middleware kann man die Anzahl der Schnittstellen von $m \cdot n$ auf $m+n$ reduzieren.

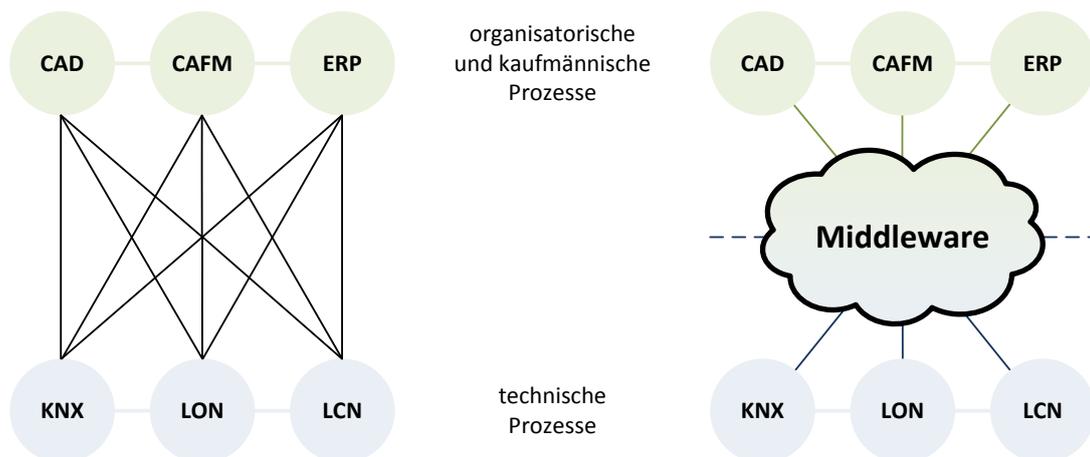


Abbildung 14: Integration technischer und kaufmännischer Prozesse ohne und mit Middleware, eigene Darstellung

Als Middleware bezeichnet man eine zusätzliche Schicht in einer komplexen Anwendungsstruktur. Technisch gesehen handelt es sich um eine Software-Schnittstelle, die Anfragen fremder Software zum Zwecke des Datenaustauschs „übersetzen“ kann. Sie stellt Funktionen und Dienste bereit, sodass die Softwarekomponenten unabhängig von ihrer Infrastruktur und Programmlogik kommunizieren können. Durch diese vereinfachten Zugriffsmechanismen bleiben die jeweilige Infrastruktur und Programmlogik nach außen hin verborgen.²⁷

In der Praxis werden Kopplungen von GA und CAFM (auf Datenbasis) bereits mit einer separaten Middleware realisiert. Informationen der GA werden über einen Leitrechner an den Middleware-Server übertragen und von diesem gefiltert. Sie werden anschließend in geeigneter Form für andere Systemteilnehmer bereitgestellt oder über definierte Schnittstellen versendet.²⁸

²⁷ (DATACOM Buchverlag GmbH, ITWissen - Middleware, 2013)

²⁸ (May, CAFM-Handbuch, 2013)

Als Anwendungsbeispiele seien im Folgenden drei Produkte genannt, die im Bereich Facility Management als Schnittstellenlösungen eingesetzt werden.

- **Hermos FIS#Facility**

Die Gebäudeautomation ist durch gewerkeübergreifende Anlagenstrukturen gekennzeichnet. Hermos FIS#Facility stellt eine Plattform dar, in der verschiedene Anlagen unterschiedlicher Hersteller mit einheitlicher Benutzeroberfläche zentral zusammengeführt werden können. Verbrauchs- und Betriebsdaten sowie Störmeldungen werden in einer Datenbank erfasst. Die Anlagen lassen sich damit effizient überwachen und steuern. In der Datenbank gespeicherte Informationen stehen ERP- und CAFM-Systemen für kaufmännische und strategische Prozesse bereit.²⁹

- **Wonderware**

Mit Hilfe dieser Software lassen sich verschiedene Systeme der Gebäudeautomation zusammenführen. Das Prinzip basiert auf der Verwendung offener Protokolle und Standards. Wonderware bildet damit ein einheitliches Steuerungssystem für gebäudetechnische Einrichtungen. Diese Schnittstelle ermöglicht Verbindungen zu vorhandenen Building Automation Systemen und die Integration mit Geoinformationssystemen (GIS), dem Energiemanagement oder Komponenten wie Löschanlagen, Feuermeldern, Aufzügen etc. zwecks Betriebsdatenerfassung und Reporting.³⁰

- **sMOTIVE**

Die Integrationsplattform sMOTIVE Middleware unterscheidet sich von den zuvor genannten Schnittstellenlösungen in einem wesentlichen Punkt: es geht um die Integration verschiedener Anwendungen im Bereich des Facility Managements ohne Schnittstellen zur Gebäudeautomation. Der Leitgedanke ist die ganzheitliche Abbildung sämtlicher Prozesse über eine einzige, webbasierte Oberfläche. sMOTIVE ist eine modular aufgebaute Workflowmanagement-Software und in der Lage, IWMS-, IPS-, CMMS- und CAFM- Prozesse, die prinzipiell dem gleichem Schema folgen, in einem Portal zu integrieren.³¹

²⁹ (Hermos, Hermos FIS#Facility, 2013)

³⁰ (Invensys Systems GmbH, Wonderware in der Gebäudeleittechnik - Industrial Operations, 2013)

³¹ (sMOTIVE, sMOTIVE Web-Portal, 2013)

3.5 Betrachtung von Normen und Richtlinien

In diesem Kapitel werden einige Normen und Richtlinien aufgeführt, die sich mit den Themen CAFM, IT-Schnittstellen oder Gebäudeautomation beschäftigen. Sie werden speziell auf Kapitel mit der Behandlung von Schnittstellen hin betrachtet und analysiert. Ein besonderes Augenmerk gilt dabei den Ausführungen zur Kopplung von CAFM und GA/GLT, nicht nur auf Managementebene (Datenaustausch), sondern insbesondere auch auf Automationsebene (Steuerungsstrategien).

3.5.1 DIN 32736: Gebäudemanagement – Begriffe und Leistungen

In dieser viel zitierten Norm werden Begriffe definiert und Leistungen des Gebäudemanagements beschrieben. Diese Norm zielt nicht darauf ab, Systeme zu beschreiben oder zu definieren, wer die aufgeführten Leistungen zu erbringen hat. Sie dient dem einheitlichen Sprachgebrauch und der Strukturierung von Leistungen.

Das Gebäudemanagement ist ein Teilbereich des Facility Managements, wie in Abbildung 4: Gliederung nach GEFMA-Richtlinie 100, eigene Darstellung, gezeigt.

An dieser Stelle sei verdeutlicht, dass das Gebäudemanagement als das Organisieren auf Managementebene zu verstehen ist und nicht gleichbedeutend mit der Gebäudeautomation ist. Die Behandlung von Prozessen auf der Automationsebene bleibt in dieser Norm außen vor. Dies ist Schwerpunkt der Richtlinie VDI/GEFMA 3814 Blatt 3.1 „Gebäudeautomation“, die in Kapitel 3.5.4 dieser Arbeit behandelt wird.

DIN 32736 ordnet die Leistungen des Gebäudemanagements in die unterteilten Bereiche „Technisches GM“, „Infrastrukturelles GM“ und „Kaufmännisches GM“. Die als Unterkapitel genannten Aufgaben dieser Teilbereiche decken sich weitestgehend mit den in der GEFMA-Richtlinie 100-1 aufgezählten Punkten, wie in Kapitel 3.1 dieser Arbeit aufgeführt. Unter das Technische Gebäudemanagement fällt auch der für weitere Betrachtungen relevante Bereich „Informationsmanagement“, was im gleichnamigen Kapitel 3.1.4 der DIN 32736 im Detail behandelt wird. Dabei wird die Gebäudeautomation (GA) separat vom Computer Aided Facility Management (CAFM) betrachtet, was ein Indiz für den fehlenden Ansatz einer Kopplung beider Systeme ist.

3.5.2 GEFMA 400: Computer Aided Facility Management (CAFM)

Die neueste Ausgabe der GEFMA Richtlinie 400 von Januar 2013 definiert CAFM als eine Anwendungssoftware zur umfangreichen Unterstützung von FM-Prozessen über den Zeitraum des gesamten Lebenszyklus eines Gebäudes. Wichtige Merkmale dabei sind die Verarbeitung grafischer und alphanumerischer Daten sowie die systematische Steuerung im Sinne eines Workflow-Managements. Diese Merkmale ermöglichen zusammen mit den in dieser Richtlinie beschriebenen CAFM-Funktionalitäten auf Basis einer zentralen Datenbank eine Effizienzsteigerung in sämtlichen FM-Prozessen.

Die Richtlinie nimmt Bezug auf den im englischsprachigen Raum etablierten Begriff Integrated Workplace Management System (IWMS, siehe Kapitel 3.2 dieser Arbeit), wonach er dem Verständnis eines CAFM-Systems nach GEFMA Richtlinie 400 weitgehend entspricht.

Zum Begriff „Gebäudeautomation“ gibt es bereits in der Einleitung der CAFM-Richtliniengruppe einen Verweis zur DIN 32736 „Gebäudemanagement – Begriffe und Leistungen“ (siehe Kapitel 3.5.1). Entsprechend dieser Norm werden CAFM-Systeme klar von der Gebäudeautomation abgegrenzt und gleichzeitig Schnittstellen zu Parallelsystemen als Grunderfordernis deklariert. Auch Prof. Michael May, Leiter des CAFM-Arbeitskreises der GEFMA, schreibt in dem von ihm herausgegebenen CAFM-Handbuch: „Computer Aided Facility Management bildet einen Schwerpunkt im Informationsmanagement von Immobilien und zugehörigen Anlagen bei klarer Abgrenzung zur Gebäudeautomation und anderen gängigen IT-Anwendungen wie Planungssoftware, Office-Lösungen oder kaufmännischer Standardsoftware. Bei aller Abgrenzung gilt es als ein Grunderfordernis des CAFM, beim Aufbau entsprechender nutzerspezifischer Systeme Schnittstellen zu den im Unternehmen bereits eingeführten benachbarten Systemen im notwendigen Umfang zu entwickeln und zu unterhalten.“³²

Insbesondere die „klare Abgrenzung“ zur Gebäudeautomation wird im weiteren Verlauf dieser Arbeit diskutiert werden.

Unter den in Richtlinie 400, Kapitel 4.3 „CAFM Funktionalitäten“ aufgezählten FM-Prozessen ist der Bereich MSR-Technik gemäß strikter Trennung nicht aufgeführt. Lediglich in den Kapiteln 4.3.6 „Energiecontrolling“ und 4.3.7 „Instandhaltungs-

³² (May, CAFM-Handbuch, 2013)

management“ gibt es Empfehlungen für GLT-Schnittstellen zwecks Datenimport (Fernauslesetechnologie) bzw. Anbindung hinsichtlich Störfallmanagement. Eine Kopplung zwecks aktiver Steuerung der Anlagen ist hierbei nicht angedacht.

Es stellt sich die Frage, ob die Steuerung von Anlagen als neuer FM-Prozess definiert werden sollte?

Die in Richtlinie 400, Kapitel 4.4.4 „Schnittstellen“ geforderte Integration von CAFM in vorhandene IT-Landschaften führt zwei Varianten der Kopplung auf. Zum einen die Realisierung von Daten-Schnittstellen (Offline-Schnittstellen) und zum anderen die „auf Basis standardisierter Kommunikationsmethoden und -dienste [...] synchrone oder asynchrone bidirektionale Kommunikation zwischen verschiedenen Anwendungen“³³(Online-Kopplung).

Außer der eher oberflächlichen Empfehlung, derartige Schnittstellen generell zu realisieren, gibt es keine weiteren oder detaillierteren Angaben zur konkreten Umsetzung solcher System-Anbindungen. Des Weiteren wird nicht spezifiziert, welchen Prozess die Schnittstellen abbilden bzw. welchen Anwendungszweck sie verfolgen. Es ist anzunehmen, dass unter einem bidirektionalen Austausch eine Methode zur Abfrage und Übergabe von statischen Werten (Zählerstände, Störungen, Betriebsstunden,...) für Statistiken zu verstehen ist, nicht aber Protokolle zur aktiven Anlagensteuerung.

Fazit: Es bleibt offen, welche Informationen im Detail über die Schnittstelle ausgetauscht werden sollen. Kaufmännische Daten für kalkulatorische Auswertungen, Geräteattribute für das Anlagenmanagement oder Protokolle für eine aktive Steuerung? Konkrete Schritte zur Umsetzung einer bidirektionalen Online-Kopplung werden nicht genannt.

3.5.3 GEFMA 410: Schnittstellen zur IT-Integration von CAFM-Software

Damit eine CAFM-Software in eine bestehende IT-Infrastruktur eingebunden werden kann, ist ein Schnittstellenkonzept zur Interaktion mit anderen Softwareanwendungen notwendig. Dabei ist zwischen CAFM-Software und CAFM-System zu

³³ (German Facility Management Association, GEFMA-Richtlinie 400, 2013)

unterscheiden.³⁴ Die GEFMA Richtlinie 410 gibt Handlungsanweisungen zur Integration verschiedener Systeme mit gleicher Datenbasis. Ob CAFM-, CAD-, GLT-, ERP- oder sonstiger fachspezifischer Software, alle halten größtenteils den gleichen Datenbestand vor.

Folgende Grafik verdeutlicht die gemeinsame Datenbasis verschiedener Softwarekomponenten eines CAFM-Systems, wie sie in Kapitel 3.2 bereits genannt wurden. Hier wird die Bedeutung und Verwendung einer Middleware als Schnittstellenlösungen (Integrationssoftware) noch einmal verdeutlicht.

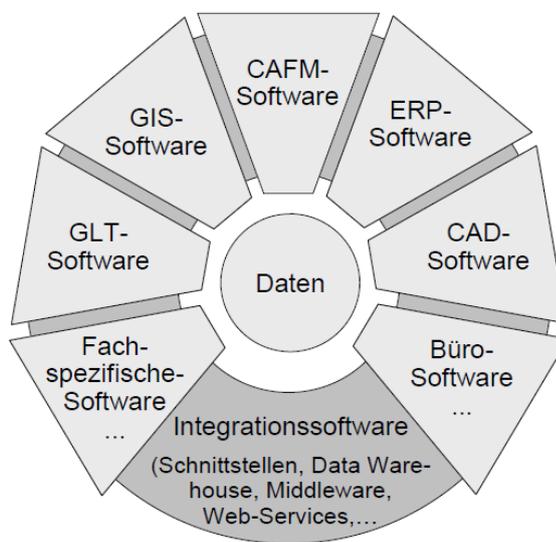


Abbildung 15: Softwarekomponenten eines CAFM-Systems, Quelle: GEFMA-Richtlinie 410

In der Einleitung dieser Richtlinie ist bereits erkennbar, dass die Themenbereiche Informations- und Wissensverwaltung im Fokus stehen. Das Schnittstellenkonzept sieht lediglich den reinen Datenaustausch zwischen Software-Komponenten vor, eine Hardware-Anbindung ist nicht Bestandteil.

Es stellt sich die Frage, ob ein ganzheitliches FM die Steuerung von Anlagen beinhalten sollte?

Die Funktionen einer Schnittstelle werden in Richtlinie 410, Kapitel 2.7 „Schnittstellenprogramm“ wie folgt definiert:

³⁴ (German Facility Management Association, GEFMA-Richtlinie 410, 2007)

- „Auslesen der Daten aus dem Ursprungsort, ggf. in eine Transferdatei
- Überprüfung der Daten am Zielort nach festgelegten Regeln (z. B. Beachtung funktionaler Abhängigkeiten und Plausibilität)
- Einlesen der Daten am Zielort
- Rückmeldung der erfolgreichen oder fehlerhaften Datenübernahme und ggf. Löschung von temporären Transferdateien“³⁵

Richtlinie 410, Kapitel 2.9 „Serielle Schnittstelle“ beschränkt sich ebenfalls auf die Konzepte des reinen Datenaustauschs, wobei gerade hier die technische Voraussetzung, nämlich die direkte Kopplung an das Bus-System der Gebäudeautomation für die Anlagensteuerung gegeben wäre. Zudem gibt es keine weitere Ausführung zum Begriff „standardisiertes Datenübertragungsprotokoll“.

Es gilt zu beantworten, ob es nicht einen Standard hinsichtlich Kopplung zwischen COM-Port und CAFM-System geben sollte, ggf. in Abhängigkeit des angeschlossenen Geräts? Sollte das Blickfeld auf weitere Anschlüsse gerichtet werden?

Richtlinie 410, Kapitel 4 „Überblick zu Schnittstellen“ klassifiziert nochmals die Bereiche Online- und Offline-Schnittstelle sowie Uni- und Bidirektionalität.

Auch hier ist zu diskutieren, ob die Funktionalität der Schnittstelle von der reinen Datenübertragung um Protokolle zur Anlageninstallation, -konfiguration und -steuerung erweitert werden sollte.

Folgender Absatz in Richtlinie 410, Kapitel 4.2 „Übertragung von FM-Daten“ ist im Hinblick auf das Ziel dieser Arbeit besonders hervorzuheben: „Im Bereich der Gebäudeautomationssysteme existieren viele hersteller- und produktabhängige Spezifikationen, sodass hier meist eine Individuallösung gefunden werden muss. Jedoch sind auch hier Standardisierungsbemühungen zu erkennen, sodass beispielsweise für MS-Betriebssysteme die OPC-Schnittstelle (OLE for Process Control) geschaffen wurde, die es erlaubt, Daten herstellerunabhängig zwischen den verschiedenen Systemen auszutauschen. Aber auch hier müssen die GA-Systemhersteller oder Drittanbieter einen entsprechenden OPC-Server zur Verfügung stellen, da dieser hardwareseitig die physische Anbindung an die Feldgeräte (Datenpunkte) realisiert.“³⁵

³⁵ (German Facility Management Association, GEFMA-Richtlinie 410, 2007)

Genau an dieser Problematik setzt diese Arbeit an, um Standardlösungen für die herstellerunabhängige Anbindung von Feldgeräten an CAFM-Systeme zu entwickeln.

Richtlinie 410, Kapitel 6 „Vorgehensweise zur Schnittstellendefinition und -integration“ gibt Handlungsempfehlungen zur Planung und Umsetzung von IT-Schnittstellenprojekten.

Die aufgeführten Schritte stellen den Workflow eines IT-Schnittstellenprojekts dar. Diese Prozessfolge lässt sich analog auf eine hardwaregebundene Schnittstelle zur Steuerung von Anlagen übertragen und wird in späteren Kapiteln dieser Arbeit Anwendung finden.

3.5.4 VDI/GEFMA 3814 Blatt 3.1: Gebäudeautomation

Im September 2012 ist eine neue Richtlinie erschienen, die VDI/GEFMA 3814 Blatt 3.1 "Gebäudeautomation (GA), Hinweise für das Gebäudemanagement, Planung, Betrieb und Instandhaltung, Schnittstelle zum Facility Management". Diese neue Richtlinie ist aus einer Kooperation von der GEMFA und dem VDI entstanden und ersetzt die bis dahin gültige GEFMA-Richtlinie 450.

VDI/GEFMA Richtlinie 3814 Blatt 3.1 hat wohl die größte Affinität zum Thema dieser Arbeit, da sie sich speziell mit der Gebäudeautomation im Facility Management beschäftigt. Die Gebäudeautomation wird zunächst für sich betrachtet. Die Konzepte zur Verknüpfung von GA und CAFM werden im letzten Kapitel angeschnitten.

Gebäudeautomation (GA) wird als „Werkzeug für das Bedienen, Stellen, Schalten, Steuern, Regeln und Überwachen der Technischen Gebäudeausrüstung (TGA)“³⁶ definiert. Für den Austausch technischer Daten der GA mit CAFM-Systemen werden geeignete Schnittstellen gefordert. Die Integration von CAFM-Funktionalitäten in der GA, oder umgekehrt, GA-Funktionen in CAFM-Systemen, wird als bereits angewandte Praxis beschrieben.

Die in GEFMA Richtlinie 400 genannte „klare Abgrenzung“ der beiden Systeme scheint mit dieser Darstellung zu verschwimmen. Auch wenn es bei dieser Form der Schnittstellen oder Integration um die Bereitstellung technischer Betriebsdaten und

³⁶ (Verband Deutscher Ingenieure e.V., VDI/GEFMA-Richtlinie 3814, Blatt 3.1, 2012)

Messwerte geht, darf die Notwendigkeit der klaren Abgrenzung in Frage gestellt werden. Über eine Aufhebung der strikten Trennung und damit Verschmelzung der „Parallelwelten“ muss diskutiert werden. Argumente für diese Zusammenführung sind die Erweiterung von CAFM-Systemen um Anlagensteuerungsmechanismen bei Nutzung der gleichen Datenbasis und vorhandener IT-Infrastrukturen.

Nach einer kurzen Einleitung, der Darstellung des Anwendungsbereichs sowie normativer Verweise, werden in Kapitel 3 „Begriffe“ einige für diese Arbeit relevante Definitionen getroffen. Dazu gehören Datenpunkte, Feldgeräte und die Gebäudeleittechnik.

Ein Datenpunkt „umfasst alle zugeordneten Informationen, die seine Semantik (Bedeutung) vollständig beschreiben. Die Informationen umfassen den Aktualwert und/oder Zustand und Parameter (Eigenschaften und Attribute), z. B. Signalart und Signalkennlinie, Messbereich, Einheit und Zustandstexte. Bei Datenkommunikationsobjekten werden diese Informationen ‚Properties‘ genannt. Ein Datenpunkt wird mittels einer Datenpunktadresse und/oder einer Benutzeradresse identifiziert“.³⁷

Während in GEFMA Richtlinie 450 noch die Feldebene beschrieben war, wird in VDI/GEFMA 3814 Blatt 3.1 jetzt das Feldgerät definiert. Es handelt sich dabei um „Geräte wie Messwert- und Kontaktgeber (Sensoren), Schalt- und Stellgeräte (Aktoren) sowie Ein-/Ausgabemodule für Ein- und Ausgabefunktionen, ebenso die örtlichen Steuer-, Regel- und Überwachungsgeräte, lokale Vorrang-Bedien-/Anzeigeinheiten und direkt wirkende Bedien- und Anzeigeelemente, die in der Feldebene angeordnet sind, sowie dialogorientierte Bedieneinheiten, z. B. Raumbediengeräte“.³⁷

Die Gebäudeleittechnik (GLT) ist definiert als „Technisches System mit Stell- und Anzeigeeinheiten zum Leiten der technischen Gebäudeausrüstung gemäß DIN IEC 60050-351 vorwiegend durch menschlichen Eingriff (Leiten, Bedienen und Beobachten)“.³⁷ Oftmals wird der Begriff Gebäudeautomation als Synonym für die Gebäudeleittechnik verwendet. Doch die GLT ist nur ein Bestandteil der GA, wie in Kapitel 3.3 dieser Arbeit, Abschnitt Managementebene, bereits dargestellt. „Zwischen Gebäudeautomation (GA) und GLT als integraler Teil der GA wird erst seit der Einführung homogener digitaler Komplettsysteme unterschieden. Die GLT wurde im Allgemeinen über einen zentralen Rechner (handelsüblicher PC) mit Tastatur,

³⁷ (Verband Deutscher Ingenieure e.V., VDI/GEFMA-Richtlinie 3814, Blatt 3.1, 2012)

Bildschirm, Protokoll- und Störmeldedruker realisiert und an zentraler Stelle im Gebäude (Leitwarte, Leitstand oder ähnlich) aufgestellt, während die GA umfassender definiert wird.“³⁸

Das letzte Kapitel dieser Richtlinie trägt den Titel „Verknüpfung der Gebäudeautomation mit CAFM-Software“ und greift das Kernthema der vorliegenden Arbeit auf. Eine CAFM-Software wird gleich zu Beginn als ein „Werkzeug zur Abbildung mittelbarer Prozesse“ mit „kaufmännisch-organisatorischem Charakter“ klassifiziert.³⁸ Damit ist schon im Vorfeld zu erkennen, dass die Abbildung eines unmittelbaren Prozesses, wie eine aktive Steuerung, keine Handlungsempfehlung dieser Richtlinie sein wird. Sie gibt Aufschluss über Ansatzpunkte zur Integration auf Managementebene („Zusammenführung technischer und kaufmännisch-organisatorischer Perspektiven“)³⁸, die im Einzelfall zu prüfen seien. Eine konkrete Vorgehensweise der Integration bzw. deren Prüfung wird nicht aufgezeigt. Hauptargument der Integration ist hier die Unterstützung der Gebäudeautomation hinsichtlich einer Dokumentation als Entscheidungshilfe für Instandhaltungsmaßnahmen.

Schließlich werden am Ende der Richtlinie folgende Schnittstellenkonzepte aufgeführt, mit dem Hinweis, dass Lösungsansätze stark variieren und somit kein allgemeingültiges „How-to“-Konzept empfohlen werden kann:

- „intermediäre Datenbank: Datentransfer über eine zwischengeschaltete, von den beteiligten Systemen unabhängige Datenbank
- Systemaufruf ohne Datentransfer
- direkter wechselseitiger Lese- und Schreibzugriff der beteiligten Systeme
- Datentransfer über Webservices
- Übergabe von Dateien“³⁸

Alle genannten Konzepte bewegen sich auf Managementebene. Die Integration auf weiteren Ebenen, insbesondere der Automationsebene unter Nutzung gemeinsamer Ressourcen und einer Einheitlichen IT-Infrastruktur kann zu einer innovativen Kombination der Systeme führen.

³⁸ (Verband Deutscher Ingenieure e.V., VDI/GEFMA-Richtlinie 3814, Blatt 3.1, 2012)

4 Technologien

Für die Realisierung automatisierter Prozesse in der Gebäudeautomation müssen entsprechende Geräte und Anlagen miteinander kommunizieren. Dazu werden Kommunikationssysteme, wie Feldbusse (Bereich Elektrotechnik) und/oder Computernetze (Bereich Informatik) eingesetzt.

Dieses Kapitel widmet sich im ersten Teil den Funktionsweisen und Kommunikationswegen von Bussystemen im Bereich der Gebäudeautomation und stellt die darauf basierenden Feldbus-Technologien und -Protokolle sowie die konkreten Transportmedien vor. Im zweiten Teil werden die Funktionsweisen und Kommunikationswege von Computernetzwerken erörtert, ebenfalls unter Betrachtung der Technologien, Netzwerkprotokolle und Transportmedien.

4.1 Bussysteme

Bussysteme sind als Teil der Gebäudesystemtechnik die technische Grundlage für eine intelligente Gebäudeautomation. Über Bussysteme können getrennte Gewerke wie beispielsweise Elektro (Licht/Beschattung) Heizung, Klima, Lüftung, Sicherheit sowie Schließanlagen eines Gebäudes zusammengeschaltet werden. Ein „Bus“ ist die Zusammenfassung einer Vielzahl paralleler Leitungen, an die mehrere Komponenten oder Funktionsblöcke eines Gerätes angeschlossen sind.

Dieses Unterkapitel behandelt die gängigsten Feldbus-Technologien für die Gebäudeautomatisierung. Dazu gehören KNX (Bussystem der Konnex-Association), LON (Local Operating Network) und LCN (Local Control Network). Die genannten Technologien unterscheiden sich in der Art des Übertragungsmediums und der Übertragungsprotokolle, die für den Datentransport sorgen. Mit BACnet (Building Automation and Control Networks) wurde ein herstellerunabhängiges Netzwerkprotokoll in der Gebäudeautomation geschaffen. Weitere Konzepte sind Powerline und digitalSTROM, bei denen der Datenaustausch über das Stromnetz erfolgt.

Neben diesen kabelgebundenen Systemen gibt es auch funktechnische Systeme, wie beispielsweise ZigBee und Z-Wave. In dieser Arbeit beschränkt sich die Betrachtung von Funksystemen aber auf eine weitere kabellose Variante: die batterielose Funksensorik von EnOcean.

Die konventionelle Elektroinstallationstechnik dient zum einen dem reinen Transport der Energie und zum anderen gleichzeitig der Steuerung aller elektrotechnischen Vorgänge im Gebäude, einfach durch Schließen (Einschalten der Lampe) oder Unterbrechen (Ausschalten) des Energiekreislaufs. Doch mit der Zunahme von Geräten und Funktionen kommt die herkömmliche Methode schnell an ihre Grenzen, da der Bedarf an Kabeln und Leitungen und damit der Material-, Kosten- und Arbeitsaufwand immer höher wird.³⁹

Ein „bahnbrechendes“ Konzept schaffte schließlich die Grundlage für neue Möglichkeiten der Elektroinstallation: Bussysteme. Es beruhte auf der Idee, Energie und Informationen mit getrennten Leitungen zu transportieren.

„Allen Bussystemen gemeinsam ist, dass sämtliche Verbraucher (Aktoren) mit sämtlichen Befehlsgebern (Sensoren) über ein Medium, meist eine 2-adrige, verdrehte Kupferleitung, miteinander verbunden sind.“³⁹



Abbildung 16: Buskabel, Quelle: Online-Produktkatalog Fa. VOKA

Wie funktioniert ein Bussystem? Sämtliche Busteilnehmer, wie Bedienelemente, Sensoren und Aktoren, werden über eine Busleitung miteinander verbunden. Auf dieser Leitung werden Signale zum Schalten, Steuern, Stellen, Regeln, Melden, Messen, Zählen und Überwachen gesendet. Voraussetzung dafür ist, dass die Installationsgeräte busfähig sind. Dazu müssen sie mit einer programmierbaren Steuereinheit (einem sog. Controller) ausgerüstet sein, die Informationen in Form von Datentelegrammen über den Feldbus versenden und empfangen kann. Die Interaktion der gesamten Elektroinstallation erfolgt durch Programmierung dieser Controller.³⁹

Die Struktur von Bussystemen kann dabei linien-, ring-, baum- oder sternförmig sein. Man bezeichnet dies als Netztopologien, die im Folgenden näher erläutert sind:

³⁹ (BauNetz Media GmbH, Baunetzwissen Elektro - Grundprinzip und Elemente von Bussystemen, 2013)

- Linientopologie (Installation in einer Linie ohne Abzweige)
Die Teilnehmer sind direkt an die offene Übertragungsleitung angeschlossen
- Ringtopologie (Installation, bei der Anfang und Ende einer Linie verbunden sind)
Die Teilnehmer sind direkt an die geschlossene Übertragungsleitung angeschlossen
- Baumtopologie (Installation in einer Linie mit Abzweigen)
An einer Übertragungslinie sind weitere Linien angeschlossen
- Sterntopologie (Installation, bei der Abzweige in einem Punkt zusammengeführt sind)
Teilnehmer werden zentral verbunden, beispielweise mit einem Hub oder Switch

Für Bussysteme gibt es eine Vielzahl an Normen und Regelwerken, die teilweise bereits in Kapitel 3.3 zitiert worden sind. Als wichtigste Dokumente sind die deutschen Normen DIN EN ISO 16484, DIN EN 13321, DIN EN 14908 und DIN EN 50090 sowie die VDI Richtlinien 3813 und 3814 zu nennen.

4.1.1 KNX (Bussystem der Konnex-Association)

Im Jahr 1999 ist aus den ehemals drei verschiedenen Standards EIB (Europäischer Installationsbus), BatiBUS und EHS (European Home System) das Bussystem KNX entstanden.⁴⁰ Die Namensgebung resultiert aus dem Zusammenschluss einiger führender Firmen aus der GA-Branche, die ihren Verband bei der Gründung „Konnex-Association“ getauft hatten. Der Installationsbus wird seither unter dem Namen KNX beworben und vertrieben. Die Herstellervereinigung sorgt für die Festlegung technischer Richtlinien und Prüfstandards, die Vorbereitung von Normen und die Überwachung der Konformität und Interoperabilität.

KNX ist ein industrielles Kommunikationssystem, das in der Haus- und Gebäudetechnik zur informationstechnischen Vernetzung von Geräten (Sensoren, Aktoren, Steuer-Regelgeräten, Bedien- und Beobachtungsgeräten) genutzt wird. Sein Einsatz ist auf die Elektroinstallationstechnik abgestimmt, wodurch Funktionen und automatisierte Abläufe in einem Gebäude sichergestellt werden.⁴¹ Das Ziel ist es, die Basis für eine gewerkeübergreifende Kommunikation zu schaffen, unabhängig vom Hersteller der Feldgeräte.

⁴⁰ (KNX Association, KNX Association - Introduction, 2013)

⁴¹ (Zentralverband Elektrotechnik- und Elektronikindustrie e.V., Zentralverband der Deutschen Elektro- und Informationstechnischen Handwerke, Handbuch Haus- und Gebäudesystemtechnik, Grundlagen, 2006)

Nach Angaben der Konnex-Association ist KNX der weltweit einzige international anerkannte und offene Standard im Bereich Haus- und Gebäudesystemtechnik. Er erfüllt die Anforderungen der europäischen Standards CENELEC EN 50090 und CEN EN 13321-1 und der internationalen Norm DIN ISO/IEC 14543-3. Darin sind das KNX-Protokoll wie auch die beiden kabelgebundenen Übertragungsmedien TP (Twisted Pair) und PL (Powerline) sowie die Funk-Variante RF (Radio Frequency) und die Ethernet-Variante IP (Internet Protocol) anerkannt.⁴²

Die am häufigsten eingesetzte Variante ist KNX.TP, bei der das Übertragungsmedium ein Buskabel ist. Die KNX-Association hat hierfür verschiedene KNX-Leitungen zertifiziert, wobei das mit YCYM 2x2x0,8 bezeichnete Kabel die klassische Lösung ist (KNX Association Zertifizierungs-Nr. 1/39/92)⁴³ KNX.TP-zertifizierte Geräte kommunizieren miteinander auf der gleichen Buslinie.

Folgendes Bild zeigt ein KNX-Kabel, wie es als Übertragungsmedium in KNX.TP-Systemen verwendet wird.



Abbildung 17: KNX-Kabel, Quelle: Online-Produktkatalog Fa. VOKA

KNX.TP-Systeme benötigen zwei voneinander getrennte Netze – das Energieversorgungsnetz für die Stromversorgung der Endgeräte und das Kommunikationsnetz zur Datenübertragung. Dabei beschreibt die Topologie die Struktur des Systems im Hinblick auf die kommunikationstechnische Verbindung der enthaltenen Komponenten.⁴³ Beispiele für Netztopologien sind in Kapitel 4.1 beschrieben.

KNX-Anlagen sind in einer Baumstruktur angeordnet. Die Hauptübertragungsleitung wird als „Backbone Line“ oder Bereichsline (BL) bezeichnet, die mehrere Bereiche

⁴² (KNX Association, KNX Standard - Einführung, 2013)

⁴³ (Merz, Hansemann & Hübner, Gebäudeautomation, Kommunikationssysteme mit EIB/KNX, LON und BACnet, 2010)

über deren Hauptlinie (HL) verbindet. Diese verzweigen sich in einzelne Linien (L), an denen die Endgeräte, sog. Teilnehmer, angeschlossen sind. Über Linien- oder Bereichskoppler lassen sich verschiedene Segmente verbinden.

Die Grafik zeigt die beschriebene Struktur eines KNX-Netzwerks:

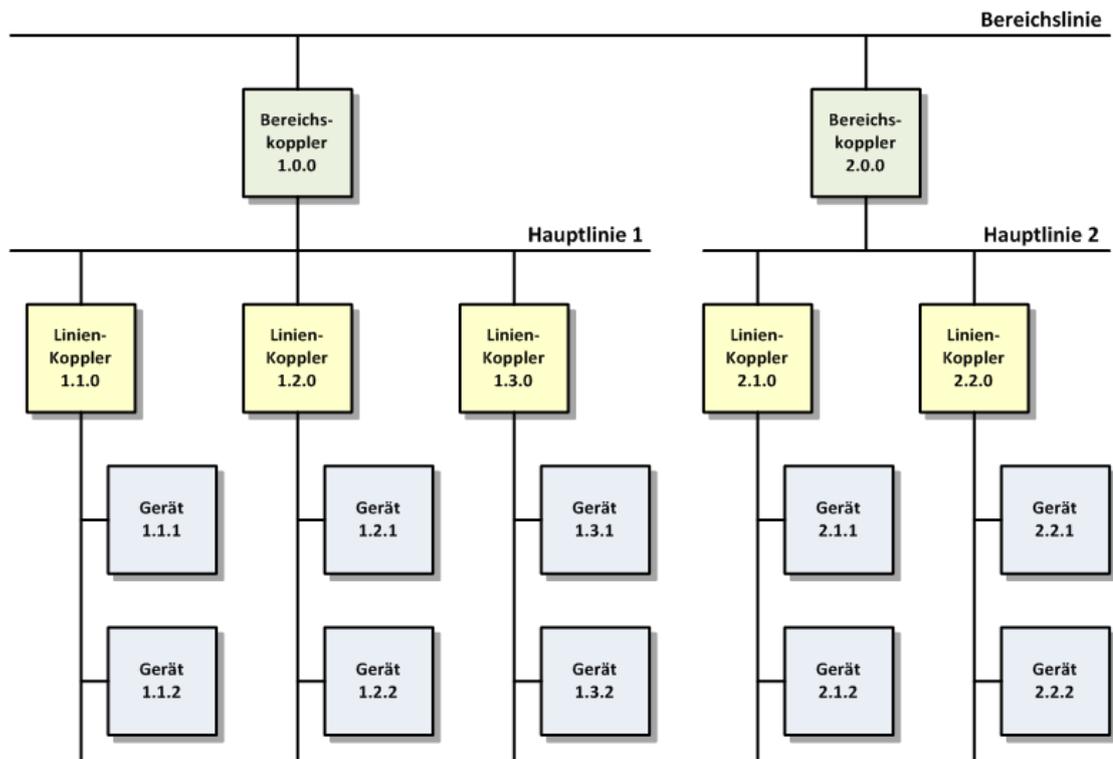


Abbildung 18: Topologie eines KNX-Netzwerks nach DIN EN 13321-2, eigene Darstellung

Alle KNX-Feldgeräte besitzen einen eigenen Mikrocontroller und sind gleichberechtigte Teilnehmer des Bussystems. Zu den Systemkomponenten gehören die Spannungsversorgungen zum Erzeugen der Busspannung und sog. Koppler zum Verbinden von Busabschnitten.⁴⁴

Häufig eingesetzte Busgeräte sind Spannungsversorgungen mit Drossel, Schaltaktoren und Tastensensoren. Die Spannungsversorgung stellt die Eingangsspannung der angeschlossenen Geräte mit 30 VDC sicher. Über den Schaltaktor lassen sich je nach Verdrahtung verschiedene Stromkreise öffnen und schließen. Tastensensoren lösen durch Tastendruck den Schaltvorgang im Aktor aus, in dem ein Datentele-

⁴⁴ (Arbeitskreis Maschinen- und Elektrotechnik staatlicher und kommunaler Verwaltungen (AMEV), Hinweise für Planung, Ausführung und Betrieb der Gebäudeautomation in öffentlichen Gebäuden, 2005)

gramm mit Schaltbefehl gesendet wird. In Abhängigkeit der Länge des Tastendrucks unterscheidet das Telegramm zwischen Ein-/Ausschalten und Dimmen.⁴⁵

Wird ein Datentelegramm gesendet, muss die zu übertragende Bitinformation mit Hilfe eines Transceivers in physikalische Signale (je nach Übertragungsmedium z. B. Spannung, Funk oder Licht) umgewandelt werden. Der Empfänger formt das Signal wieder in die Bitinformation zurück.⁴⁵

Damit gesendete Datentelegramme die Absender und Empfänger identifizieren können, müssen alle angeschlossenen Busgeräte eindeutig adressiert sein, was über die Zuordnung einer physikalischen Adresse sichergestellt wird. In dieser Adressierung spiegelt sich die in Abbildung 18 dargestellte Bustopologie wider. Sie beinhaltet Blöcke mit Angaben zum Bereich, zur Linie und zum Teilnehmer.

Bereich				Linie				Teilnehmer							
B3	B2	B1	B0	L3	L2	L1	L0	T7	T6	T5	T4	T3	T2	T1	T0

Abbildung 19: Aufbau der Adresse eines KNX-Teilnehmers, eigene Darstellung

Die Adresse 1.12.7 würde dabei den 7. Teilnehmer der 12. Linie des 1. Bereichs kennzeichnen. Das Datentelegramm würde folgenden Binärcode beinhalten:

0001 1100 0000 0111

Im Dualsystem können $2^4 = 16$ Bereiche, $2^4 = 16$ Linien und $2^8 = 256$ Teilnehmer beschrieben werden. Das sind in Summe $2^{16} = 65.536$ Geräte.

Die Adresse ist Teil des Datentelegramms, das zur Kommunikation zwischen den Geräten versendet wird. Die Quelladresse ist dabei die physikalische Adresse des Absenders, die Zieladresse kann auch eine logische Gruppenadresse sein. Diese beschreibt in der sog. 2-Ebenen-Adressierung die Haupt- und Untergruppe bzw. in der 3-Ebenen-Adressierung die Haupt-, Mittel- und Untergruppe.

Des Weiteren beinhalten Datentelegramme ein Kontrollfeld, einen Adresstyp, einen Routingzähler, eine Nutzdatenlänge, die Nutzdaten und ein Prüffeld.⁴⁵

⁴⁵ (Merz, Hansemann & Hübner, Gebäudeautomation, Kommunikationssysteme mit EIB/KNX, LON und BACnet, 2010)

Abbildung 20 zeigt die einzelnen Bitfelder eines KNX-Standarddatentelegramms mit Angabe der jeweiligen Länge.

Kontrollfeld	Quelladresse	Zieladresse	Adresstyp	Routingzähler	Länge	Nutzdaten	Sicherung
8 Bit	16 Bit	16 Bit	1 Bit	3 Bit	4 Bit	16-128 Bit	8 Bit

Abbildung 20: Aufbau eines KNX-Standarddatentelegramms, eigene Darstellung

Aus der Zielsetzung dieser Arbeit heraus ist das Protokoll der Ethernet-Variante, KNXnet/IP, im Sinne der Webfähigkeit genauer zu betrachten. Es wird bei der Einbindung von KNX-Geräten in TCP/IP-Netze der Gebäudeautomation eingesetzt, um auch lokale IP-Netzwerke und das Internet für die Übertragung von KNX-Telegrammen benutzen zu können. Das Datentelegramm ist in das TCP/IP-Telegramm eingebettet (detaillierte Informationen zum Netzwerkprotokoll TCP/IP siehe Kapitel 4.2.4).

Auf diese Weise können KNXnet/IP-Router alternativ zu KNX-Linien- oder Bereichskopplern eingesetzt werden, wie Abbildung 21 zeigt. Mit der Nutzung der vorhandenen Ethernet-Verkabelung wird die normale KNX-Bereichsline durch eine schnelle Linie auf Ethernetbasis ersetzt. Das IP-Netzwerk stellt einen Hochgeschwindigkeitsersatz für die Bereichsline dar und verbindet die KNX-Subnetze.^{46,47}

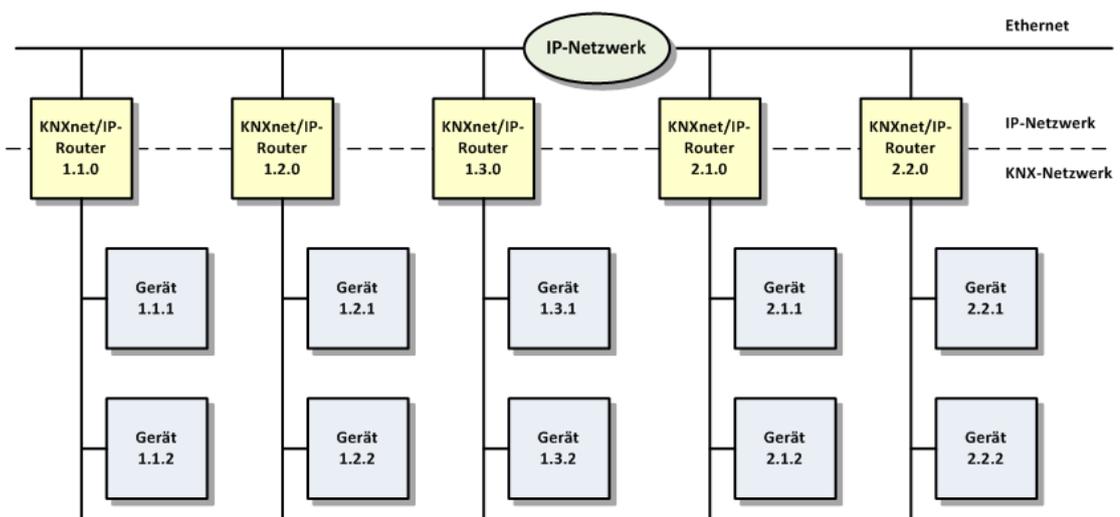


Abbildung 21: Topologie eines KNXnet/IP-Netzwerks nach DIN EN 13321-2, eigene Darstellung

⁴⁶ (KNX Association, KNX Standard - Kommunikationsmedien, 2013)

⁴⁷ (Deutsches Institut für Normung e.V., DIN EN 13321-2, 2012)

In dieser Grafik sind die gleichen Geräte wie in Abbildung 18 mit KNXnet/IP-Routern verbunden, die wiederum alle mit demselben IP-Netzwerk gekoppelt sind. Datentelegramme der Subnetze werden als Teil von TCP/IP-Telegrammen über das IP-Netzwerk zu einem oder mehreren Geräten gleichzeitig übertragen. Sofern die Filterkriterien erfüllt sind, übertragen die KNXnet/IP-Router als Empfänger der IP-Telegramme die Informationen an ihr Subnetz.

In der Praxis werden sog. Feldbus-Controller eingesetzt, mit denen sich ein KNX.TP-Netzwerk an das Ethernet anbinden lässt. Der Feldbus wird über die TP-Klemme angebunden, das Ethernet über die RJ45-Buchse für LAN-Kabel. Folgende Grafik zeigt einen solchen Controller der Firma Wago:



Abbildung 22: KNXnet/IP-Controller aus dem Wago I/O-System, Quelle: Online-Produktkatalog Fa. Wago

Für die Bedienung auf Managementebene bietet das Unternehmen Wago auch eine eigene Software namens „Wago I/O-Pro CAA“ an. Mit ihr lassen sich sämtliche HKL-Anwendungen durchführen, wie zum Beispiel die Regelung von Lüftungsanlagen oder auch Beleuchtungs- und Jalousiesteuerungen. Ein im Controller integrierter Web-Server ermöglicht es, projektspezifische Funktionen der Applikation über das Internet darzustellen. Somit lässt sich die Applikation auch aus der Ferne bedienen.

Seit Februar 2012 existiert eine aktualisierte DIN Norm für die KNXnet/IP-Kommunikation, die DIN EN 13321-2 „Offene Datenkommunikation für die Gebäudeautomation und Gebäudemanagement – Elektrische Systemtechnik für Heim und Gebäude – Teil 2: KNXnet/IP Kommunikation“.

4.1.2 LON (Local Operating Network)

Für umfangreiche Anlagen wird zunehmend auch die LON-Technologie (Local Operating Network) eingesetzt, die ihre Ursprünge in der Industrieautomatisierung hat. LON gibt es seit Anfang der 1990er Jahre und ist eine Entwicklung der Echelon Corporation. Die LonMark Deutschland e.V. ist die Interessenvereinigung der Anwender und Entwickler rund um die LON-Technologie im deutschsprachigen Raum.

Diese Netzwerk-Technologie ermöglicht die Integration von Geräten aus unterschiedlichen Gewerken in einem System. Integration bedeutet in diesem Zusammenhang die gewerkeübergreifende Vernetzung von z. B. Heizung, Klima, Lüftung, Beleuchtung, Beschattung, Zutrittskontrolle, Feuer- und Rauchmeldung etc.

Seit Dezember 2008 ist diese Technologie von der ISO (International Organization for Standardization) und der IEC (International Electrotechnical Commission) als internationale Norm anerkannt und in der internationalen Normenreihe ISO/IEC 14908 dokumentiert.⁴⁸ Die deutsche Ausgabe dieser Norm, DIN EN 14908, wurde letztmalig im Oktober 2012 aktualisiert.

Die Grundidee, den Informationsaustausch zwischen Anlagen und Geräten verschiedener Hersteller unabhängig von den Anwendungen zu ermöglichen, gleicht der in Kapitel 4.1.1 vorgestellten KNX-Technologie. Der wesentliche Unterschied liegt in der technischen Umsetzung. Im Gegensatz zum hierarchisch orientierten KNX-System beruht das LON-Konzept auf einer dezentralen Automatisierung, also einer Verteilung der Intelligenz.

Alle an einem LON-Netzwerk angeschlossenen Sensoren und Aktoren (auch Knoten oder engl. Nodes genannt) können die für das lokale Endgerät bestimmten Informationen selbst verarbeiten, da sie jeweils über einen frei programmierbaren Chip, dem sog. „Neuron“ verfügen (siehe Abbildung 23). Jeder Chip besteht aus drei CPU-Kernen: der Media-Access-CPU zur Kontrolle der physikalischen Verbindung zum Netzwerk, der Network-CPU zur Kodierung und Dekodierung der Netzwerknachrichten und der Application-CPU mit der vom Anwender programmierten Software. In diesen Anwendungsprogrammen steckt die eigentliche „Intelligenz“ eines Knotens.

⁴⁸ (LonMark International, LON Archives ISO/IEC Standardization, 2008)

Die Abbildung zeigt den Aufbau eines Neuron-Chips in eigener Darstellung auf Basis des Fachbuchs „LON-Technologie“ von Dietrich, Loy & Schweinzer⁴⁹.

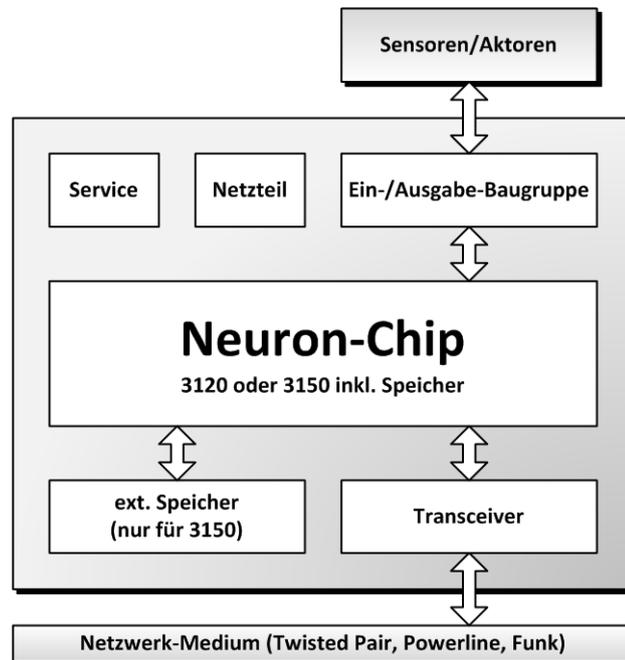


Abbildung 23: Aufbau eines Neuron-Chips nach Dietrich, Loy & Schweinzer („LON-Technologie“,1998), eigene Darstellung

Diese Geräte können über verschiedene Übertragungsmedien wie Zweidrahtleitung, Funk, Lichtwellenleiter, Powerline oder auch über IP miteinander verbunden werden. Als Verbindungskabel für die einzelnen LON-Knoten dient in der Regel ein 4-adriges Kabel vom Typ JY (St) Y 2x2x0,8, wobei zwei Adern für den LON-Bus und zwei weitere für die Spannungsversorgung von passiven LON-Geräten verwendet werden.



Abbildung 24: Raumcontroller mit Klappenstellantrieb, Quelle: Online-Produktkatalog Fa. Schneider Electric Buildings

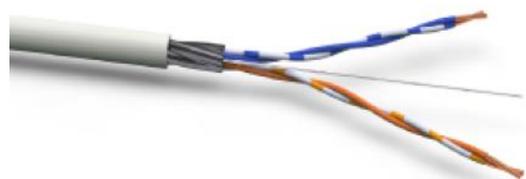


Abbildung 25: LON- Kabel, Quelle: Online-Produktkatalog Fa. VOKA

⁴⁹ (Dietrich, Loy & Schweinzer, LON-Technologie, 1998)

Die Kommunikation in einem LON-Netzwerk erfolgt dezentral, d.h. die Knoten tauschen Informationen direkt untereinander aus. Sie können in beliebigen, auch gemischten Netzwerk-Topologien angeordnet sein. Dabei muss jedes Netzwerksegment einen Abschlusswiderstand mit einer Impedanz von ca. 52 Ohm haben, um eine saubere Datenübertragung zu gewährleisten.

Nachstehende Abbildung zeigt eine gemischte Topologie eines LON-Netzwerks, auch „Free Topology“ genannt, mit Linien-, Ring- und Baumstruktur.

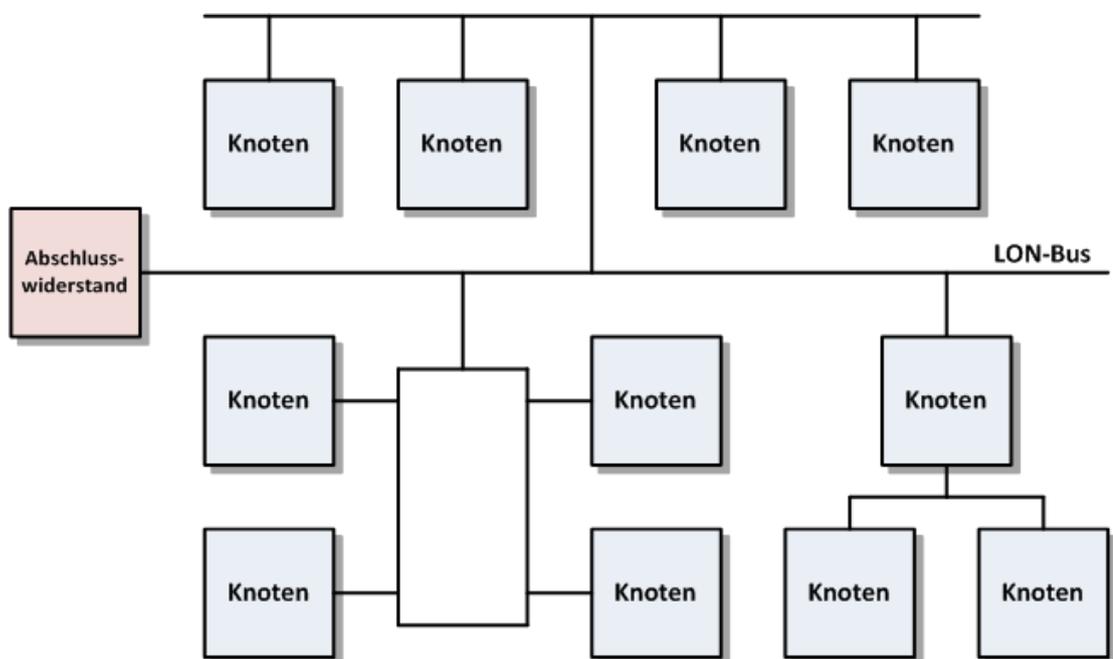


Abbildung 26: Gemischte Topologie eines LON-Netzwerks, eigene Darstellung

LON-Netzwerke sind in ihrer Ausprägung nicht begrenzt. Lokale Netzwerke können zu großen, ortsübergreifenden Netzwerken zusammengefasst werden. Sie können auch über das Internet von entfernten Orten aus kontrolliert und gesteuert werden.

Durch eine weltweit einmalige 48-Bit-Identifikationsnummer (der sog. Neuron-ID) ist jeder Neuron-Chip und damit jeder Knoten im Netz eindeutig identifizierbar. Aktionen können so an jedem beliebigen Ort und bei jedem beliebigen Teilnehmer innerhalb des LON-Netzwerkes ausgelöst werden. Dafür gibt es das gemeinsame Kommunikationsprotokoll „LonTalk“ sowie standardisierte Netzwerkvariablen und Funktionsprofile für unterschiedlichste Anwendungen.

Das LonTalk-Protokoll definiert alle sieben Ebenen des ISO/OSI-Referenzmodells. Für Schicht 1 (Physikalische Schicht) gibt es keine Einschränkung auf ein bestimmtes Übertragungsmedium, es stehen verschiedene Transceiver zur Verfügung, wie zum Beispiel leitungsgebundene Übertragung, Funk, Glasfaser aber auch Powerline-Kommunikation. Die Datenkodierung auf der physikalischen Schicht kann direkt durch den Neuron-Chip oder durch den Transceiver selbst gesteuert werden.⁵⁰

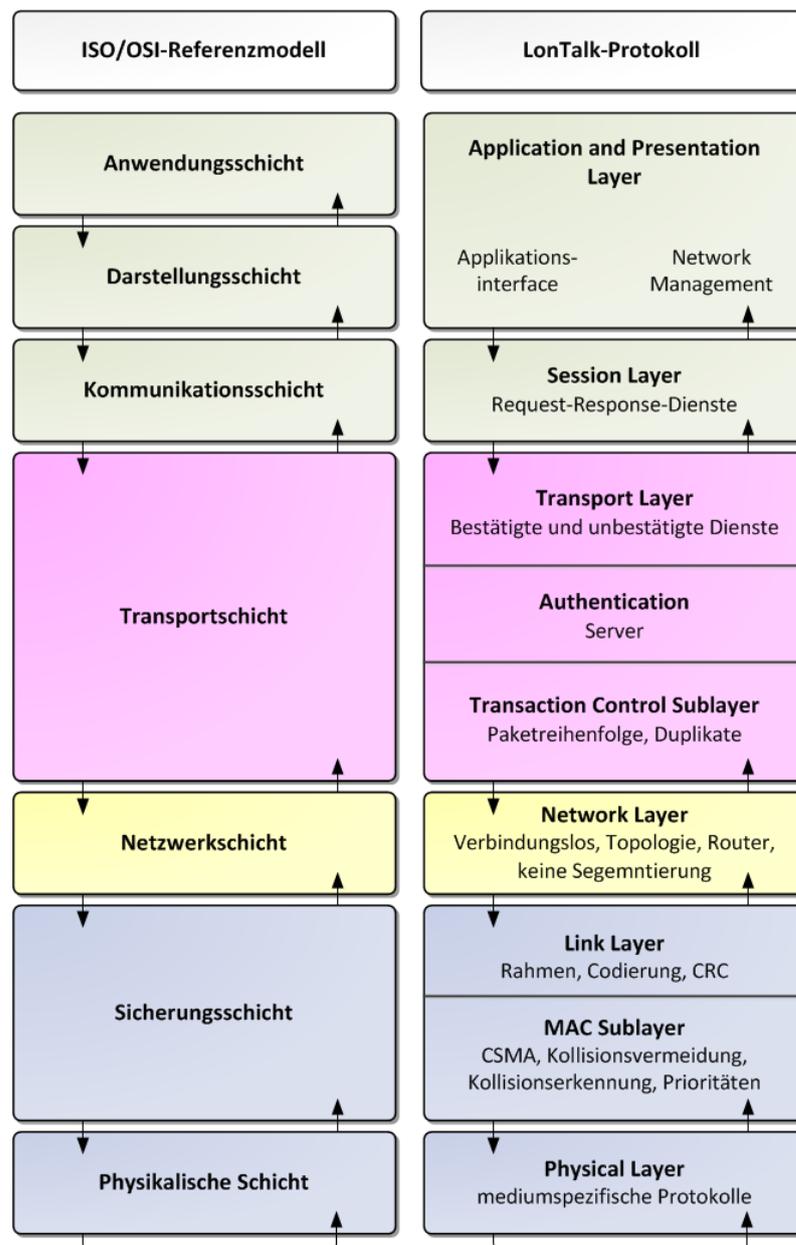


Abbildung 27: Aufbau des LonTalk-Protokolls auf Grundlage des ISO/OSI-Referenzmodells nach Dietrich, Loy & Schweinzer („LON-Technologie“, 1998), eigene Darstellung

⁵⁰ (Dietrich, Loy & Schweinzer, LON-Technologie, 1998)

Eine technische Voraussetzung für ein funktionierendes LON-Netzwerk ist die Interoperabilität (das problemlose Zusammenarbeiten) verschiedener Geräte unterschiedlicher Funktionen und Hersteller, was durch die Organisation LonMark International mit einer Zertifizierung von neuen Geräten sichergestellt wird.

Mit dem Einsatz von LON IP-Routern lassen sich LON-Netzwerke über Ethernet/IP integrieren. Das Ethernet wird so als „Backbone“ (Hauptübertragungsleitung) verwendet, mit dem sich auch ferne Teilnehmer vernetzen lassen. Die Router senden Datenpakete aus LON-Netzwerken über Ethernet/IP-Kanäle.

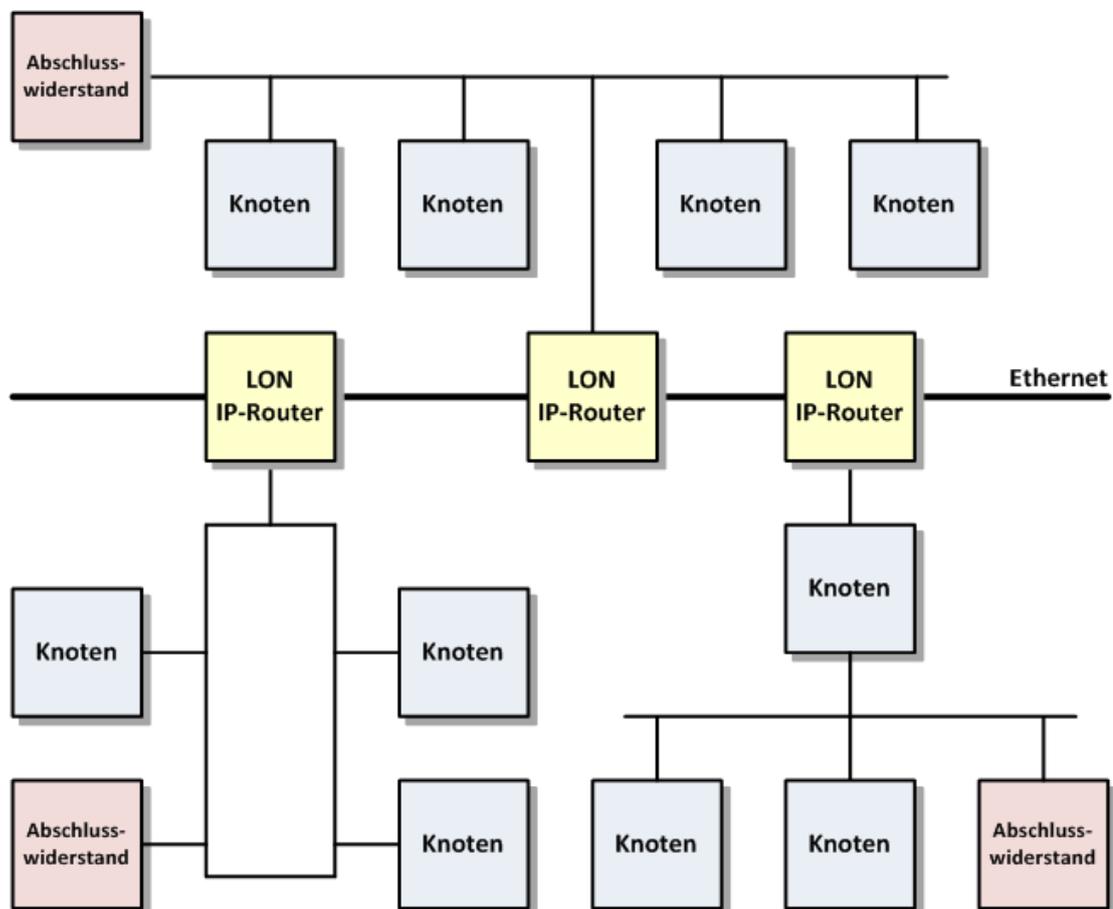


Abbildung 28: LON-Topologie beim Einsatz von IP-Routern, eigene Darstellung

LON IP-Router sind Systemgeräte zur Vermittlung von Paketen zwischen einem Netzwerk-Segment und einem IP-Netzwerk. Einige Modelle haben auch einen eingebauten Webserver zur Konfiguration über das Internet. Es gibt eine Vielzahl an LON-zertifizierten Geräten und Herstellern.

Folgende Abbildung zeigt einen IP-Router der Firma Loytec electronics GmbH:



Abbildung 29: LON IP-Router, Quelle: Online-Produktkatalog Fa. Loytec

Auf Managementebene gibt es für LON-Systeme eine Vielzahl an herstellerabhängigen Softwareprodukten. Als Beispiel sei TAC Vista der Fa. Schneider Automatic Buildings Germany GmbH genannt. Ein modernes Gebäudemanagementsystem, das Systeme für Heizung, Lüftung, Klima, Beleuchtung und Zutritt/Sicherheit in Liegenschaften überwacht und steuert. Mit dieser Software lässt sich beispielsweise das in Abbildung 24 gezeigte Gerät (Raumcontroller mit Klappenstellantrieb) bedienen. Laut Produktbeschreibung des Herstellers läuft TAC Vista als eigenständiges System oder Multi-Computersystem in lokalen oder verteilten Netzwerken. Das System kann sowohl für die Bedienung und Überwachung vor Ort als auch dezentral eingesetzt werden.

4.1.3 LCN (Local Control Network)

Ein mittlerweile ebenfalls weit verbreitetes, einfach zu konfigurierendes System bildet der LCN-Bus. Der Begriff LCN (Local Control Network) ist die Produktbezeichnung des Gebäudeleitsystems des Unternehmens Issendorff KG, das seit 1992 auf dem Markt ist. Das System findet, wie KNX und LON auch, in den Bereichen Lichtsteuerungen, Beschattung, Temperaturregelung oder Sicherheit Anwendung.

LCN ist ein dezentrales System, das im Unterschied zu anderen Bussystemen ganz ohne separate Busleitung auskommt. Der Datenaustausch der Komponenten erfolgt über eine freie Ader der handelsüblichen Elektroinstallationsleitung. Es ist kein separates Busnetz erforderlich. Eine zusätzliche Spannungsversorgung oder zentrale Steuerungseinheiten werden nicht benötigt, die LCN-Module arbeiten autark. Dadurch können sie direkt an das verlegte Unterputzkabel angeschlossen und weiterhin Standard-Installationskomponenten, wie herkömmliche Taster, Bewegungsmelder usw., eingesetzt werden.⁵¹

Derzeit gibt es noch keine Regelwerke wie Normen oder Richtlinien zur LCN-Technologie, die anzuwenden wären. Generell gibt es wenige Planungsregeln zu beachten. Da die Datenader als Netzleitung behandelt wird, darf sie nach VDE mit der 230 V-Installation gemeinsam geführt werden.⁵¹

Die Busmodule enthalten einen Mikroprozessor zum Messen, Steuern und Regeln. Sie verfügen über ein integriertes Netzteil. Die Module gibt es als Bauform für Hutschienen oder als Unterputzvariante. Unterputzmodule lassen sich in UP-Dosen hinter Tastern, Steckdosen und Schaltern sowie in Verteilerdosen einsetzen.



Abbildung 30: Busmodul LCN-UPP für Unterputzdose, Quelle: Online-Produktkatalog Fa. Issendorff KG

Die beschriebenen Module sind Kern des dezentralen Systems. Sämtliche verfügbaren LCN-Module weisen die gleiche Hard- und Software auf und lassen sich deshalb identisch programmieren. Die Standardmodule können sowohl für Sensoren als auch Aktoren eingesetzt werden, da je 2 Schaltausgänge und 2 bzw. 3 Eingänge sowie ein Busankoppler integriert sind. Das Sortiment der Sensoren reicht von Licht-, Wind- und Regensensoren bis hin zu Temperaturfühlern und Tür-/Fensterkontakten.

⁵¹ (Issendorff KG, LCN - Gebäudeleittechnik (Systembeschreibung), 2009)

Sensoren liefern Messwerte, Aktoren führen einen zuvor in Abhängigkeit des Sensorwertes programmierten Befehl des Moduls aus. Diese Befehle müssen dem zunächst funktionslosen Modul bei der Inbetriebnahme zugewiesen werden, um die gewünschten Funktionen und Aktionsmuster abbilden zu können.⁵²

Hinsichtlich der Kommunikation nutzt das System zur Datenübertragung einen freien Datendraht sowie den Nullleiter des herkömmlichen Stromnetzes, was die Verwendung eines mindestens 4- oder 5-adrigen Stromkabels voraussetzt (Standard-NYM-Kabel). Über diese beiden Drähte ist jedes Modul in der Lage, mit dem gesamten Bus über LCN-spezifische Kommandos und Telegramme zu kommunizieren.⁵²



Abbildung 31: NYM-Kabel, Quelle: OnlineProduktkatalog Fa. VOKA

Für den LCN-Bus gibt es keine vorgeschriebene Topologie. Der Bus kann linien-, stern- oder baumförmig aufgebaut sein, wie in der folgenden Grafik dargestellt:

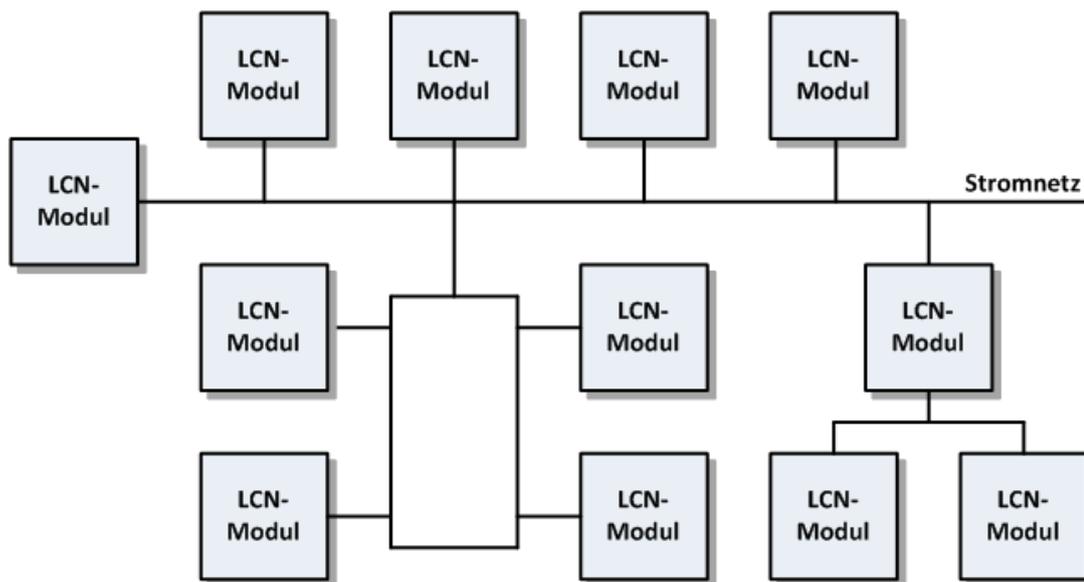


Abbildung 32: Topologie eines LCN-Bus, eigene Darstellung

⁵² (Issendorff KG, LCN - Gebäudeleittechnik (Systembeschreibung), 2009)

Das Unternehmen Issendorff KG wirbt mit der außerordentlichen Flexibilität bzgl. der Ausstattungsgröße, die von kleinen Wohnhäusern bis zu großen Zweckbauten mit einer Dimension von bis zu 30.000 Modulen individuell angepasst werden kann.

LCN-Module kommunizieren über Kommandos, die in Datentelegrammen verpackt über die Datenleitung versendet werden. Datentelegramme beinhalten neben den Absender- und Empfängeradressen ein Infofeld, ein Prüffeld und Felder mit Nutzdaten, z. B. Kommando „Licht an“, Parameter 1 „Helligkeit 80 %“.

Absender	Info	Prüfsumme	Ziel-Segment	Ziel-Modul	Kommando	Parameter 1	Parameter 2
8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit

Abbildung 33: Aufbau eines LCN-Datentelegramms, eigene Darstellung

Die LCN-Module werden über ihre individuelle Adresse (ID) identifiziert, die zuvor über eine Konfigurationssoftware zugewiesen wird. Diese ID ist jedoch nicht gleichbedeutend zur IP-Adresse und somit nicht direkt aus dem Internet ansprechbar. Dazu bedarf es einer Kopplung des LCN-Bus mit einem PC über eine entsprechende RS232-Schnittstelle, den Buskoppler LCN-PK.⁵³



Abbildung 34: Buskoppler LCN-PK (RS232-Schnittstelle), Quelle: Online-Produktkatalog Fa. Issendorff KG

Die Visualisierung, Überwachung und Steuerung über das Internet wird über eine serverbasierte Software, das LCN-GVS (Globales Visualisierungssystem), realisiert. Diese Applikation ist seit Juni 2010 auf dem Markt.

⁵³ (Issendorff KG, LCN - Gebäudeleittechnik (Systembeschreibung), 2009)

4.1.4 BACnet

BACnet steht für „Building Automation and Control network“ und ist im Vergleich zu den gerade vorgestellten Technologien KNX, LON und LCN kein eigenes Bus-System, sondern ein Netzwerkprotokoll für die Gebäudeautomation. Es stellt herstellerunabhängige Kommunikationsmöglichkeiten zwischen Anlagen und Geräten zur Verfügung. Mit BACnet lassen sich z. B. KNX- und LON-Systeme koppeln und über die GA/GLT auf Managementebene überwachen.

BACnet ist die Entwicklung eines 1987 von der American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) gegründeten Komitees. Es hatte als Ziel, einen einheitlichen und firmenneutralen Standard für die Datenkommunikation der Gebäudeautomationssysteme bereitzustellen. Der neue Standard wurde 1995 mit ANSI/ASHRAE-Norm 135 erstmals veröffentlicht, gefolgt von der zweiten Auflage im Jahre 2002.⁵⁴

„BACnet ist 2003 in die internationale Norm als ISO 16484-5 Systeme der Gebäudeautomation: Datenkommunikationsprotokoll sowie bereits in den Jahren vorher in zahlreiche nationale Normen (in Europa als DIN EN 13321 Offene Datenkommunikation für die Gebäudeautomation und Gebäudemanagement - Elektrische Systemtechnik für Heim und Gebäude und DIN V ENV 1805-1 Datenübertragungsprotokoll für die HLK-Gebäudetechnik - Managementebene: Datenkommunikation für Gebäudeautomation (BACnet) aufgenommen worden.“⁵⁵

Was war die Motivation, einen solchen Standard zu entwickeln? Mit dem technischen Fortschritt der Gebäudeautomation gab es zunehmend mehr Hersteller und Produkte auf den Markt, die ihren eigenen Kommunikationsstandard hatten. Es war nicht gewährleistet, dass ein Sensor der Firma A mit dem Aktor der Firma B kommunizieren konnte, auch wenn es sich um dasselbe Gewerk handelte. Das Netzwerkprotokoll sollte aber nicht nur herstellerunabhängig sondern auch offen sein. Dadurch wird jedem Interessierten ermöglicht, im Komitee mitzuarbeiten und die Erweiterungsvorschläge kommentieren und ergänzen zu können.

Was ist Grundlage des Netzwerkprotokolls und wie ist es aufgebaut? BACnet orientiert sich am ISO/OSI-Referenzmodell (siehe Kapitel 4.2.4) und fasst die Schichten 4-7 in der Anwendungsschicht (Application Layer) zusammen.

⁵⁴ (ASHRAE SSPC 135, Overview, 2013)

⁵⁵ (BauNetz Media GmbH, Baunetzwissen Elektro - BACnet-Bussystem, 2013)

Die folgende Grafik zeigt den Aufbau des BACnet-Protokolls auf Grundlage des ISO/OSI-Referenzmodells:

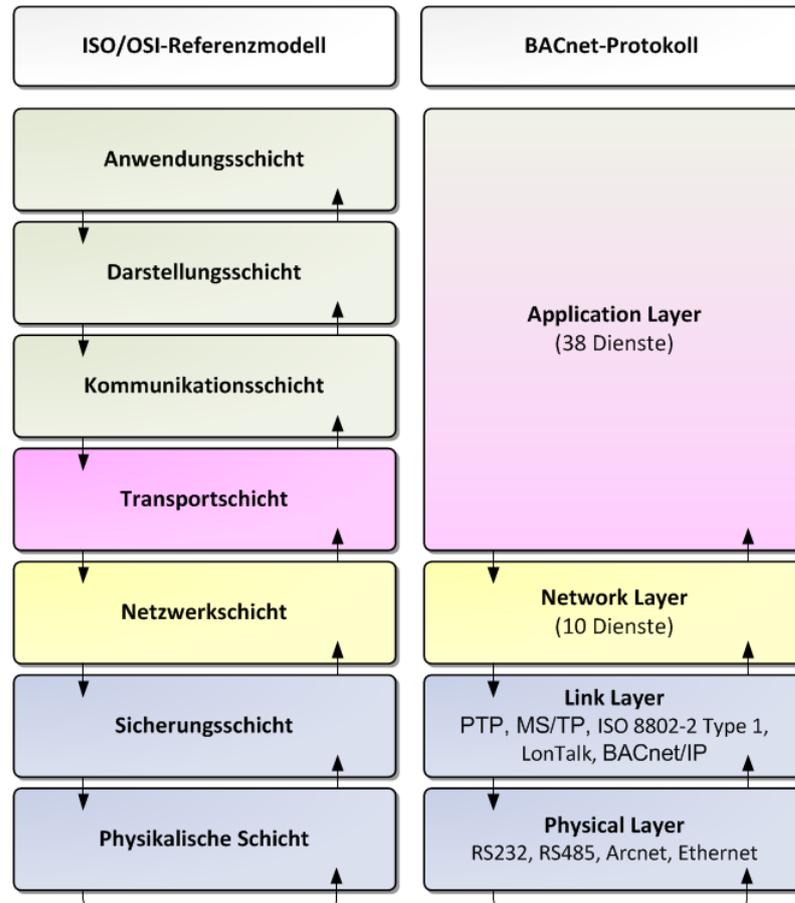


Abbildung 35: Aufbau des BACnet-Protokolls auf Grundlage des ISO/OSI-Referenzmodells, eigene Darstellung

BACnet-Geräte können eine oder mehrere der folgenden physikalischen Übertragungsschichten in Kombination mit der entsprechenden Sicherungsschicht (Physical Layer / Link Layer) verwenden:⁵⁶

- RS232 mit Protokoll PTP (Point-To-Point)
- RS485 mit Protokoll MS/TP (Master-Slave Token Passing)
- ARCnet
- Ethernet nach ISO/IEC 8802-3
- BACnet over LonTalk
- BACnet/IP basierend auf UDP/IP

⁵⁶ (BauNetz Media GmbH, Baunetzwissen Elektro - BACnet-Bussystem, 2013)

„In Europa hat sich der Transport über das Internetprotokoll (BACnet/IP) weitestgehend durchgesetzt. Dabei erfolgt der Transport mit Hilfe des UDP (User-Datagramme Protocol), welches verbindungslos arbeitet und daher einfach zu implementieren ist.“⁵⁷ Die Funktionsweise eines Internetprotokolls ist in Abschnitt 4.2.4 detailliert beschrieben. Kommen mehrere Übertragungsschichten gleichzeitig zum Einsatz, können Router für den Transport der Daten zwischen den verschiedenen Layern eingesetzt werden.

Für die Kommunikation zwischen beispielsweise KNX- und LON-Geräten über das BACnet-Protokoll braucht es Gateways, welche die physische Kopplung realisieren:



Abbildung 36: BACnet/IP-Controller (KNX/LON-Gateway), Quelle: Online-Produktkatalog Fa. Wago

Neben den physikalischen Übertragungsschichten ist auch die Deklaration von Diensten und Objekten für die Kommunikation von BACnet-Geräten notwendig.

Dienste ermöglichen den Zugriff auf Objekte oder die Ausführung von Funktionen. Auf der Netzwerkschicht sind insgesamt 10 Dienste verfügbar. Sie regeln die Weiterleitung von Nachrichten zwischen Geräten, die sich auf unterschiedlichen Netzwerkschichten befinden. Auf der Anwendungsschicht sind insgesamt 38 Dienste verfügbar, mit denen Zugriffsmechanismen und Funktionsaufrufe bereitgestellt

⁵⁷ (BauNetz Media GmbH, Baunetzwissen Elektro - BACnet-Bussystem, 2013)

werden. BACnet-Dienste lassen sich in 5 Gruppen einteilen: Objektzugriff (Lesen und Schreiben), Filetransfer, Alarm- und Event-Funktionen, RemoteDevice-Management und Virtual Terminal.⁵⁸

Objekte bilden die Eigenschaften von Komponenten der Gebäudeautomation ab. Man ordnet jedem Aus- oder Eingang eines BACnet-Gerätes ein Objekt zu. Jedes Objekt besitzt Attribute in Abhängigkeit seiner Verwendung, wie beispielsweise Namen und Nummer des Objektes, die physikalische Einheit, den aktuellen Wert usw. Diese Attribute sind obligatorisch. Optional können Beschreibungstexte oder Grenzwerte für die Alarmierung hinterlegt werden. Objektdaten lassen sich abfragen oder bearbeiten, ohne dass man den genauen Aufbau des Geräts kennen muss. Entscheidend sind nur die Ein- bzw. Ausgänge des Geräts unabhängig von Modell oder Hersteller. Merz, Hansemann und Hübner beschreiben in ihrem Buch „Gebäudeautomation – Kommunikationssysteme mit KNX, LON und BACnet“ insgesamt 25 Standard-Objekte der BACnet-Anwendungsschicht in jeweils eigenen Unterkapiteln. Diese Objekte dienen der Beschreibung von physikalischen Werten (analoge, binäre oder mehrstufige Soll-/Ist-/Mittelwerte und Zähler) oder funktionsorientierten Eigenschaften (z. B. Reglerobjekte, Geräteobjekte, Alarmklassen, Gruppenobjekte, Trendobjekte, Zeitplanobjekte, Kalenderobjekte).⁵⁸

Zusammenfassend lässt sich sagen, dass mit BACnet ein Protokoll für die Gebäudeautomation geschaffen wurde, dessen Aufbau und geräteunabhängige Funktionen mit der Bereitstellung von Objekten und Diensten eine sehr gute Grundlage für die Umsetzung der Ziele dieser Arbeit sein könnten.

4.1.5 digitalSTROM

Eine seit 2011 verfügbare neue Variante der Gebäudeautomation nennt sich „digitalSTROM“ und wird von der deutsch-schweizerischen Aizo AG entwickelt und vermarktet. Das Stromnetz wird dadurch intelligent, was auch als „Smart Grid“ bezeichnet wird. Im Vergleich zu KNX, LON oder LCN beschränkt sich die Steuerung auf das Gewerk Elektrotechnik, mit Fokus auf Licht, Sicherheit und Zugang, dazu noch vorwiegend im Bereich Home Automation. Dennoch ist digitalSTROM im Rahmen dieser Untersuchung zu berücksichtigen, da diese innovative Technologie

⁵⁸ (Merz, Hansemann & Hübner, Gebäudeautomation, Kommunikationssysteme mit EIB/KNX, LON und BACnet, 2010)

erstens noch ganz am Anfang der Entwicklung steht, zweitens eine Art des Powerline-Verfahrens ist und drittens eine Ergänzung zu den schon zuvor genannten Systemen darstellt.

Wie beim Powerline-Verfahren auch, überträgt digitalSTROM die Daten über die 230V-Leitung, ohne eine zusätzliche Busleitung zu benötigen. Der Unterschied liegt in der Kommunikationsweise, da keine höhere Frequenz auf die 50Hz-Grundwelle moduliert (Frequenzüberlagerungsverfahren), sondern die Sinuskurve in der Nähe des Nulldurchgangs manipuliert wird (digitale Kommunikation).

Die Zentrale des gesamten Systems bildet der digitalSTROM-Meter (dSM), eine Art Busmaster, der die Kommunikation zwischen den einzelnen digitalSTROM-Klemmen im Stromkreis über das proprietäre digitalSTROM-Protokoll koordiniert und zusätzlich noch den Stromverbrauch des Raumes bis auf die Geräteebene misst. Er wird auf der Hutschiene des Elektroverteilers eingebaut. Eine weitere, optionale Komponente, die das Funktionsspektrum deutlich erweitert, ist der digitalSTROM-Server (dSS). Durch ihn wird die Anbindung an das Internet über XML (Extensible Markup Language), SOAP (Simple Object Access Protocol) und TCP/IP-Protokolle realisiert. Über den Server können auch Apps für die mobile Steuerung bereitgestellt werden. Ein 16A-Schutzschalter und der digitalSTROM-Filter gegen Störungen im Stromnetz komplettieren das Gesamtsystem, das im Folgenden abgebildet ist.⁵⁹

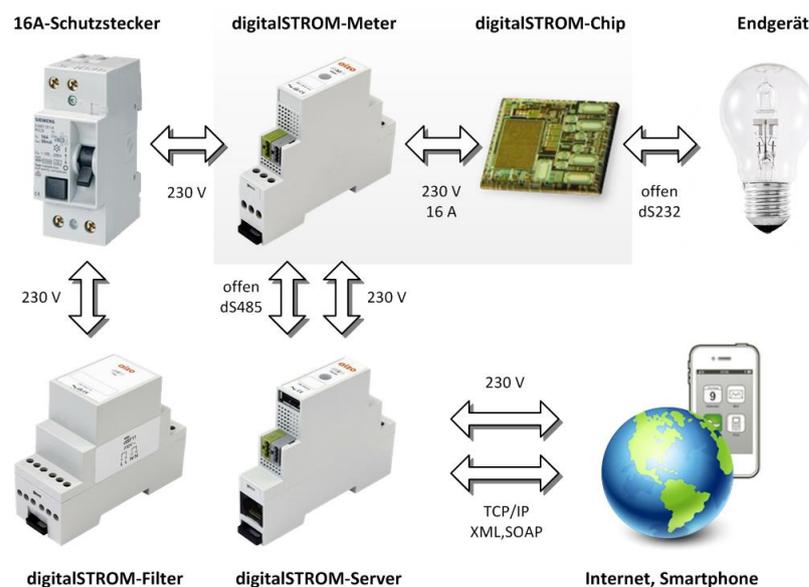


Abbildung 37: Kommunikation im digitalSTROM-Gesamtsystem nach Aizo AG, eigene Darstellung mit Bildmaterial aus Online-Produktkatalog Fa. Aizo AG

⁵⁹ (Aizo AG, Aizo - Technologie - System, 2012)

Die Intelligenz des Systems wird über Klemmen bereitgestellt, die in das jeweilige Elektrogerät (z. B. Leuchte) oder dessen Anschluss (Steckdose) integriert sind. Die darin installierte Steuerungs-Komponente mit integriertem Schaltkreis ist der 4x6 mm große digitalSTROM-Chip. Jeder Chip ist über eine weltweit einmalige, 96-Bit-Identifikationsnummer (dDID) identifiziert und somit individuell ansteuerbar. Die dSID beruht auf dem gleichen Konzept wie eine Mac-Adresse.⁶⁰

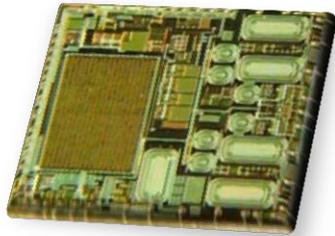


Abbildung 38:
digitalSTROM-Chip (dSID), Quelle: Online-Produktkatalog der Aizo AG (Foto nachbearbeitet)



Abbildung 39:
Lüster-Klemme mit verbautem Chip, Quelle: Online-Produktkatalog der Aizo AG

Der digitalSTROM-Meter registriert nach dem „Plug and Play“-Prinzip automatisch alle im zugehörigen Stromkreis installierten digitalSTROM-Chips. Löst nun ein Benutzer einen Steuerbefehl durch das Drücken eines Schalters aus, so sendet die digitalSTROM-Tasterklemme ein Telegramm mit den Informationen zur Art der Betätigung (Anzahl Klicks etc.) zum digitalSTROM-Meter. Dies geschieht durch Impulsübertragung, also sehr kurzes Ein- und Ausschalten in der Nähe des Nulldurchgangs des Wechselstroms, was andere Geräte nicht wahrnehmen. Der digitalSTROM-Meter sendet daraufhin ein Datentelegramm an alle im Stromkreis registrierten digitalSTROM-Chips, die auf den übertragenen Befehl reagieren und die Spannung für das angeschlossene Gerät konditionieren.⁶¹

Die Konfiguration der Komponenten kann sowohl über die Schalter selbst, als auch über die via dSS angebundene digitalSTROM-Software durchgeführt werden. Die Software steht als Open-Source-Lösung zur Verfügung⁶¹, d.h. sie kann von Geräteherstellern und Softwareentwicklern beliebig erweitert werden.

⁶⁰ (Widmer, digitalSTROM - Das Internet der elektrischen Geräte, 2009)

⁶¹ (Aizo AG, digitalSTROM Installationshandbuch, 2011)

Folgende Abbildung stellt das Gesamtsystem schematisch dar und zeigt von links unten beginnend den digitalSTROM-Filter, den 16A-Schutzstecker, den digitalSTROM-Meter mit angebundenem digitalSTROM-Server sowie die im Stromkreis installierten digitalSTROM-Klemmen.⁶²

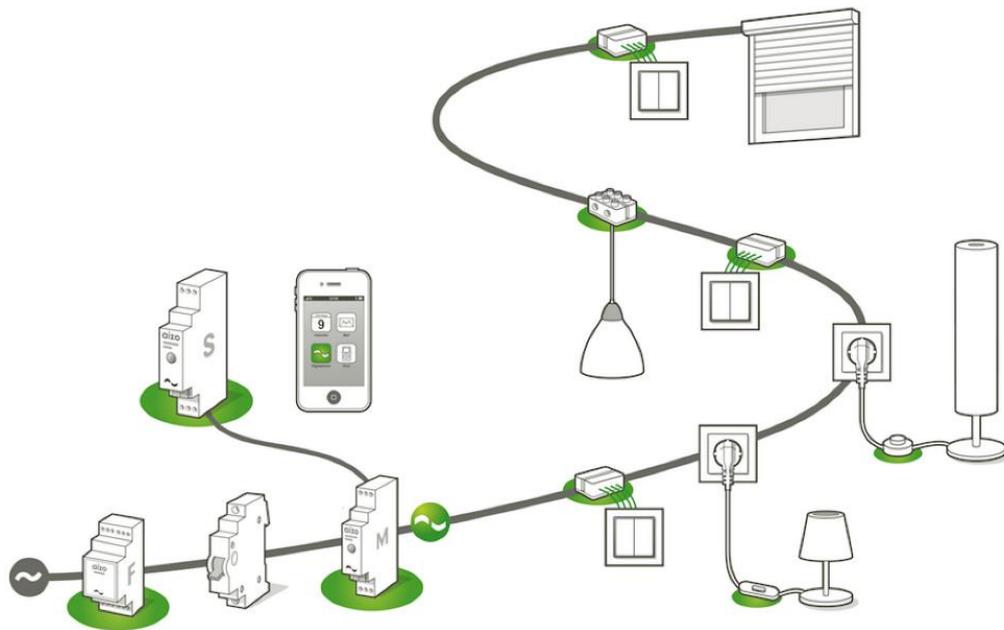


Abbildung 40: Schematische Darstellung des digitalSTROM-Netztes, Quelle: Aizo AG (digitalSTROM - Systemübersicht, 2012)

Die genaue Funktionsweise des Chips ist als geistiges Eigentum der Aizo AG deklariert und daher nicht öffentlich. Die Kommunikation nach außen und zu den einzelnen Geräten basiert hingegen auf offenen Standards. Das Protokoll zwischen dSS und dSM basiert auf dem RS485-Standard, das zwischen dSID und Endgeräten auf dem RS232-Standard.

4.1.6 EnOcean

Der Name EnOcean steht für eine vom gleichnamigen Hersteller entwickelte und patentierte batterielose Funktechnologie. Die EnOcean GmbH ist im Jahr 2001 als Ausgründung der Siemens AG entstanden.

⁶² (Aizo AG, digitalSTROM - Systemübersicht, 2012)

Funkmodule von EnOcean werden heute weltweit von über 100 Produktherstellern von Systemlösungen für Gebäude- und Industrietechnik eingesetzt. Zu den Partnern zählt neben Unternehmen wie Siemens, Sauter, Wago, Beckhoff, Zumtobel, Wieland Electric, Peha, Somfy, Thermokon, Wago, Kieback & Peter oder Eltako Electronics auch die in der Schweiz ansässige Omnio AG.⁶³

Mit EnOcean-Geräten können vorhandene kabelgebundene Bussysteme erweitert werden, da das Protokoll auch mit KNX, LON, BACnet oder TCP/IP kommunizieren kann. Die Funktechnologie basiert auf dem ISO/IEC 14543-3-10, Standard für Funkanwendungen mit einem besonders niedrigen Energieverbrauch.

Doch wie funktioniert ein Funk-Bussystem allgemein? Die Funkübertragung erfolgt zwischen Sendern (Sensoren) und Empfängern (Aktoren). Diese sind z. B. in Unterputzdosen von Schaltern, Steckdosen oder in Leuchten verbaut. Die Systeme können optional durch Signalverstärker oder eine Zentrale ergänzt werden.

„Die Sensoren (Befehlsgeber wie Taster, Fühler, Bewegungsmelder) senden per Funk eine Information an die Aktoren (Befehlsempfänger wie Lichtaktor, Jalousieaktor und Alarmsirene), welche die Informationen auswerten und entsprechend schalten. Eine Zentrale ist nicht zwingend erforderlich. Sie wird nur für Verknüpfungs- und Zeitfunktionen, wie zentrales Ausschalten oder die Anwesenheitssimulation benötigt. Einfache Zentralbefehle können direkt mit den Sendern ausgeführt werden.“⁶⁴



Abbildung 41: Ratio©-Temperatursensor,
Quelle: Online-Produktkatalog Fa. Akktor GmbH



Abbildung 42: Ratio©-Thermostataktor,
Quelle: Online-Produktkatalog Fa. Akktor GmbH

⁶³ (EnOcean Alliance, Hersteller und Produkte, 2013)

⁶⁴ (BauNetz Media GmbH, Baunetzwissen Elektro - Funk-Bussysteme, 2013)

Die Grundidee der EnOcean-Technologie beruht darauf, die für die Funkübertragung notwendige Menge an elektrischer Energie durch den Einsatz von miniaturisierten Energiewandlern aus Quellen wie Umgebungstemperatur, Bewegungen, Licht oder Luftströmungen zu gewinnen. Diese Energiequellen reichen für das Senden von kurzen Funksignalen aus, da dies nur geringe Mengen an Energie benötigt. Konkret kommen Thermowandler, Solarzellen oder Piezoelektrizitätswandler zum Einsatz. Das Gewinnen von Energie aus der Umgebung bezeichnet man als „Energy Harvesting“. Somit können die Sender batterieelos betrieben werden und bleiben weitestgehend wartungsarm.⁶⁵

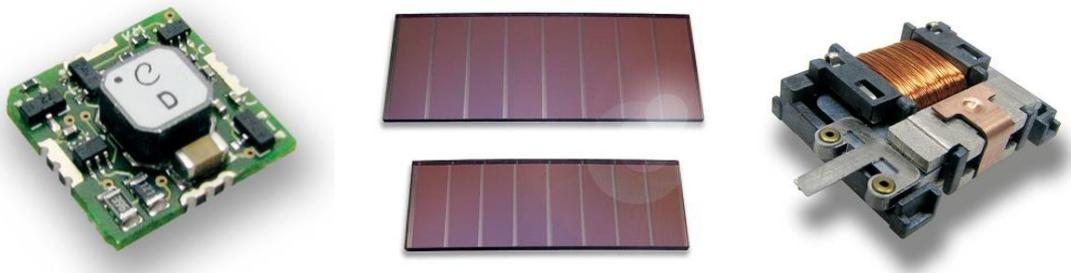


Abbildung 43: Energiewandler mit EnOcean-Technologie, Quelle: EnOcean

Sensoren und Aktoren können über Gateways an andere Systeme, wie z. B. KNX oder LON, angebunden werden. Die Kopplung an einen PC, um sie im Falle von komplexeren Szenarien über eine passende Software konfigurieren und zentral bedienen zu können, ist über ein USB-Gateway möglich.



Abbildung 44: Ratio@-Gateway, Quelle: Online-Produktkatalog Fa. Omnio AG

⁶⁵ (EnOcean, Energiewandler, 2013)

Die Datenübertragung per Funk wird über Telegramme von einer Millisekunde Länge realisiert. Das Signal verwendet die Frequenzbänder 868 MHz (Europa) und 315 MHz (international), die Datenübertragungsrate beträgt 125 Kilobit pro Sekunde. Die Datenpakete werden in zufälligen Intervallen gesendet. Um Sendefehler auszuschließen, wird das Telegramm innerhalb von 30 Millisekunden zweimal wiederholt. Die Reichweite der Funksensoren liegt bei 300 Metern im Freien und bis zu 30 Metern innerhalb des Gebäudes. Jedes Modul verfügt über eine einmalige 32-Bit-Identifikationsnummer, die Überschneidungen mit anderen Funkschaltern ausschließt. Diese ID ist jedoch nicht gleichbedeutend zur IP-Adresse und somit nicht direkt aus dem Internet erreichbar.⁶⁶

Folgende Abbildung zeigt die Funktionsweise der batterielosen Funkübertragung nach der Beschreibung von EnOcean⁶⁷ in eigener Darstellung:

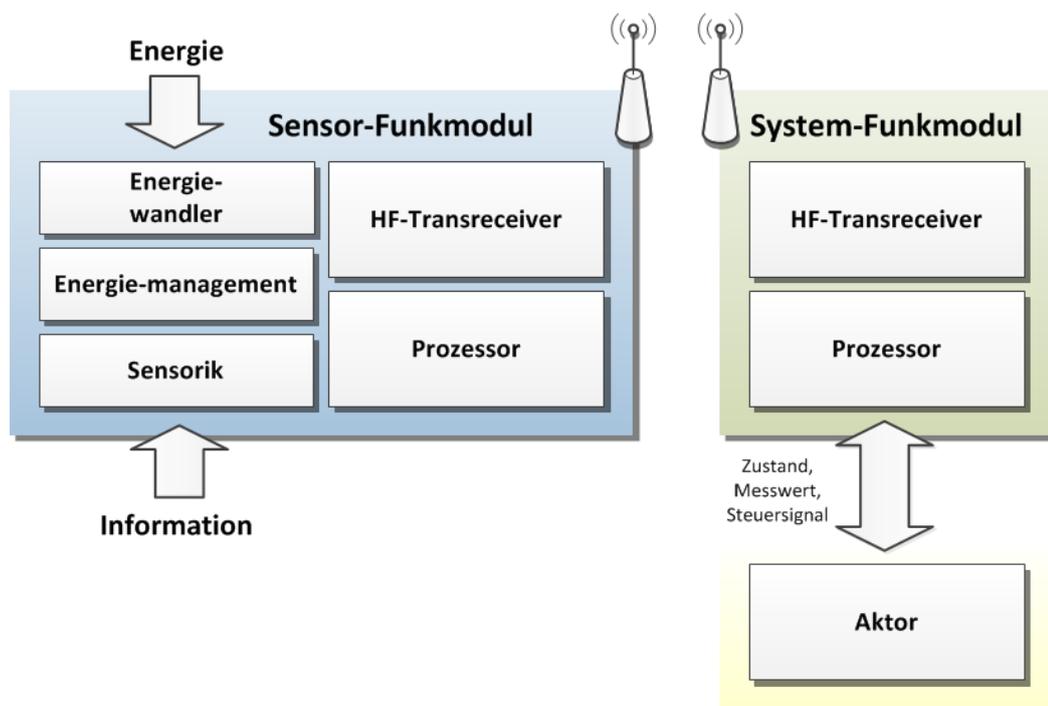


Abbildung 45: Funktionsweise der batterielosen Funkübertragung nach der Beschreibung von EnOcean, eigene Darstellung

Das standardisierte Protokoll deckt die Schichten 1 bis 3 des OSI -Modells ab und umfasst damit den Physical Layer, Data Link Layer und Networking Layer. Das Protokoll ist seit März 2012 in der Norm ISO/IEC 14543-3-10 "Information

⁶⁶ (EnOcean, Unternehmensprofil, 2013)

⁶⁷ (EnOcean, Batterielose Funktechnologie, 2013)

Technology - Home Electronic Systems (HES) - Part 3-10: Wireless Short-Packet (WSP) Protocol optimized for Energy Harvesting - Architecture and Lower Layer Protocols" definiert.⁶⁸

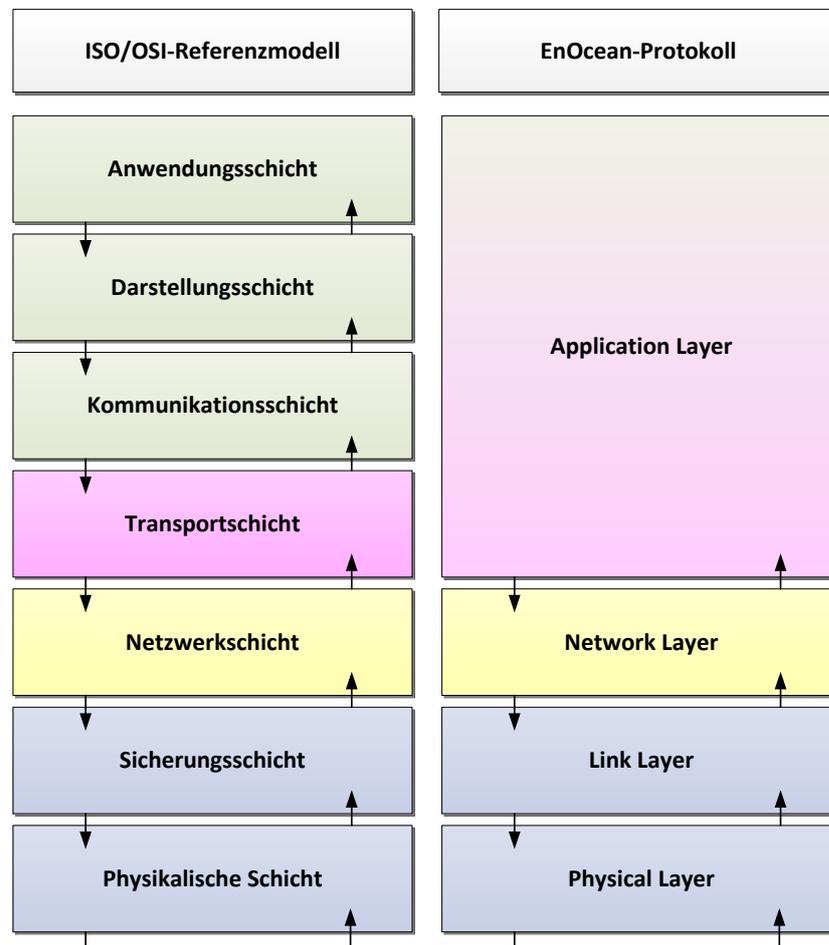


Abbildung 46: Aufbau des EnOcean-Protokolls auf Grundlage des ISO/OSI-Referenzmodells, eigene Darstellung

Nach eigenen Recherchen zufolge können Geräte mit EnOcean-Technologie aber über Fremdprodukte mit dem Internet gekoppelt werden. Dies scheint möglich mit dem BACnet/IP-Controller von Wago, der auch Anschlussmöglichkeiten für EnOcean-Geräte bietet, wie dem Anwendungsbeispiel (Abbildung 47) zu entnehmen ist. Dies konnte im Rahmen dieser Arbeit aber nicht getestet werden.

⁶⁸ (EnOcean, Funkstandard, 2013)

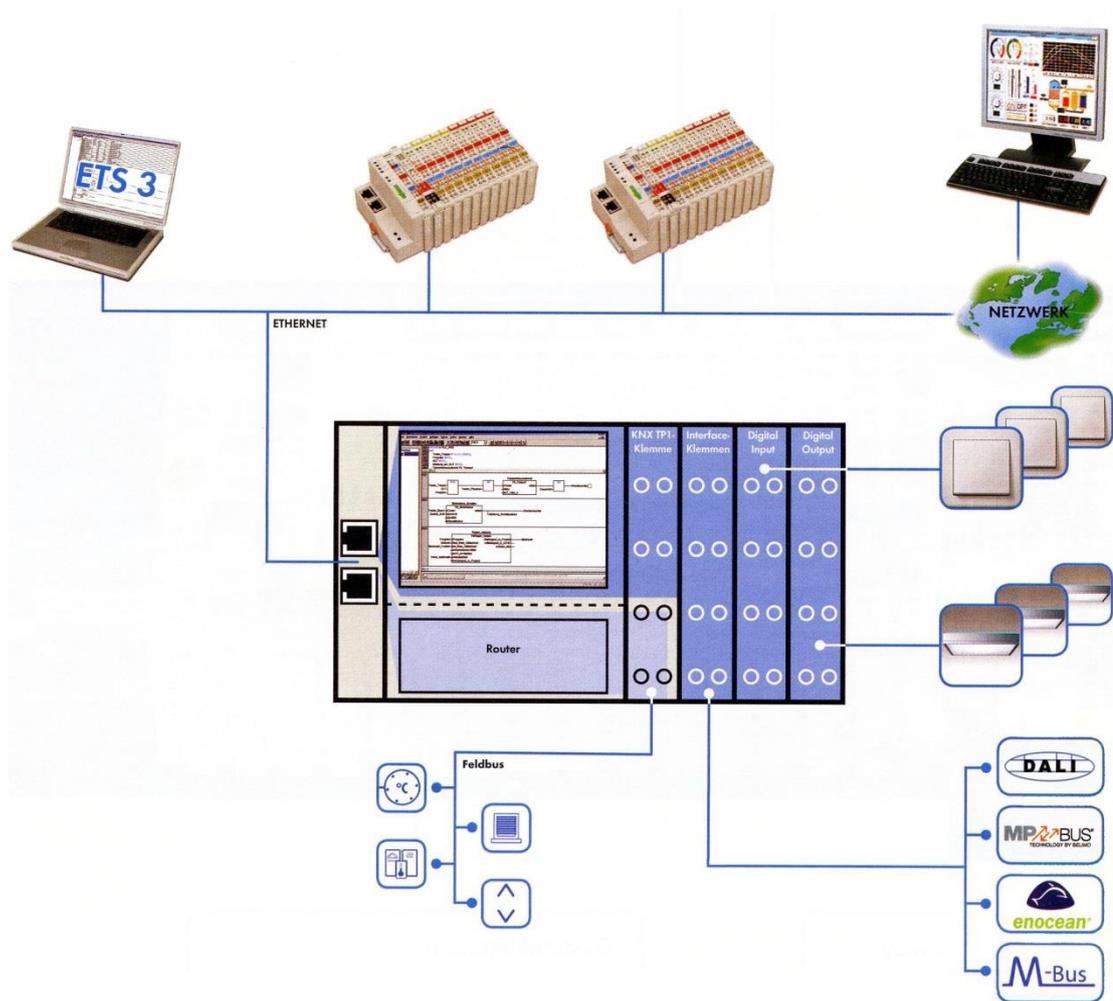


Abbildung 47: Internetanbindung von EnOcean-Geräten via BACnet/IP-Controller, Quelle: Produktkatalog „Automation“ Fa. Wago

Im Rahmen dieser Arbeit stellte die Omnio AG für die Erprobung von Geräten mit Funkmodulen, die auf der EnOcean-Technologie basieren, eine Auswahl an Sensoren und Aktoren sowie ein Gateway aus der Produktfamilie des Funkbussystems Ratio© bereit. Für den Test kamen das USB-Gateway und der Kabelschaltaktor zum Einsatz. Das Gateway wurde via USB mit dem PC verbunden und über einen vom Hersteller bereitgestellten Treiber installiert. Der Aufbau ist in Kapitel 5.1.4 beschrieben. Zur Steuerung der Hardware diente die Demoversion einer Software der BootUp GmbH namens „myHomeControl“. Im weiteren Verlauf dieser Arbeit konnten eigene webbasierte Programmbausteine entwickelt werden. Technisch wurde dies über das Senden des Steuerbefehls über den mit USB simulierten COM-Port realisiert.

4.2 Computernetzwerke

Die Datenübertragung ist der Austausch von Informationen zwischen zwei Geräten. Diese Kommunikation kann über verschiedene Wege und Technologien realisiert werden. Ein Blick auf die Standard-Anschlüsse und Schnittstellen eines Computers genügt, um die gängigsten Möglichkeiten benennen zu können. Neben SATA-, Firewire- und USB-Schnittstellen, die für die lokale Geräteanbindung gedacht sind, finden sich auch LAN-Anschlüsse oder (außerhalb des Gehäuses meist nicht sichtbare) WLAN- und Bluetooth-Adapter für entfernte Geräte. In den folgenden Kapiteln wird beschrieben, über welche Wege und Medien die Daten zwischen entfernten Geräten oder Rechnern übertragen werden können.

Während SATA-Anschlüsse nur für die Kopplung von Speichermedien vorgesehen sind, werden Firewire und USB für den Datenaustausch zwischen PC und lokalen Peripheriegeräten, wie z. B. Drucker, Scanner oder sonstige Geräte zur Ein-/Ausgabe und Steuerung, genutzt.

Unter der Voraussetzung, dass die genannten Geräte netzwerkfähig sind, können diese auch direkt über eine LAN-Verbindung oder drahtlos via WLAN mit dem PC verbunden werden. Sind mehrere Teilnehmer zum Zweck der Datenkommunikation miteinander verbunden, spricht man von einem Netzwerk.

Für den Einsatz von Steuerungsmechanismen im (CA)FM-Bereich kann eine Kombination aus beiden Anbindungsvarianten, lokal via USB und entfernt via LAN, sinnvoll sein. Abhängig von den Anforderungen muss ein geeignetes Konzept entwickelt werden. Vorstellbar wäre eine web- und serverbasierte Lösung, bei dem die Endgeräte lokal angebunden sind. Endgeräte können dabei z. B. Funkgateways sein, wie sie bei Omnio (EnOcean-Technologie, vgl. Kapitel 4.1.6) zum Einsatz kommen. Innovativer wäre der Ansatz, eine web- und serverbasierte Lösung zu entwickeln, bei der die Endgeräte über das Netzwerk mit dem Server kommunizieren, wobei die Implementierung nach dem „Plug and Play“-Prinzip möglichst komfortabel gestaltet werden sollte.

In den folgenden Kapiteln werden die für den Datenaustausch relevanten Schnittstellen und Technologien, die für die Lösung der Aufgabenstellung dieser Arbeit von Bedeutung sind, näher betrachtet.

4.2.1 LAN via Kabel/Ethernet (klassisch)

Die Technologie, welche den Datenaustausch zwischen den Geräten bzw. Computern in einem lokalen Netzwerk (LAN) definiert, nennt man Ethernet. Innerhalb eines gemeinsamen Leitungsnetzes übertragen die Netzwerk-Teilnehmer Nachrichten durch Hochfrequenz. Ethernet spezifiziert sowohl die Software (Protokolle, Signale auf der Bitübertragungsschicht und Paketformate) als auch die Hardware (Kabeltypen, Stecker, Verteiler und Netzwerkkarten) für kabelgebundene Datennetze.

Das Institute of Electrical and Electronics Engineers (IEEE) hat unter der Arbeitsgruppe 802 eine Vielzahl an Standards im Bereich der lokalen Netze (LAN) veröffentlicht. Ethernet ist in IEEE-Norm 802.3 standardisiert.

Die Abkürzung LAN steht für „Local Area Network“. Dabei handelt es sich um ein Computernetzwerk, das sich über kleine, lokale und meist auf Gebäude beschränkte Bereiche erstreckt. Als Beispiele wären ein Heim- oder ein kleines Firmennetzwerk zu nennen, bei dem viele Computer direkt oder indirekt über Hubs und Switches miteinander verbunden sind. Aber nicht nur Computer, sondern auch andere netzwerkfähige Geräte wie z. B. Drucker, Massenspeicher, Kontroll-/Steuereinheiten, Home-Entertainment usw. können in das Netz eingebunden werden.

Ein lokales Netzwerk (LAN) kann über einen Router mit anderen Netzwerken verbunden werden. Diese Netzwerke können ebenfalls LANs sein, aber auch WANs (Wide Area Network) bzw. GANs (Global Area Network). WAN- und GAN-Netzwerke erstrecken sich im Gegensatz zu einem LAN über sehr große geografische Bereiche. Das Internet ist im weitesten Sinne ein GAN, doch diese Bezeichnung wird eher selten verwendet. Man benutzt häufiger den Begriff WAN, auch für das Internet.

Der Router ist dabei die Schnittstelle oder „Brücke“ zwischen diesen Netzwerken und besitzt folglich zwei IP-Adressen für das private LAN (meist statische IP) und das öffentliche WAN (meist dynamische IP durch automatische Vergabe via DHCP). Wichtig ist dabei zu wissen, aus welchen Adressbereichen diese IPs sein können. Jede Netzwerkschnittstelle hat eine global eindeutige IP-Adresse. Das stellt sicher, dass alle Computer in einem Netz unterschiedliche Adressen haben und es zu keinem Adresskonflikt kommen kann. Für private Bereiche sind mehrere IP-Adressblöcke reserviert, die nicht in öffentlichen Netzwerken vorkommen können.

Einer dieser lokalen Adressbereiche beinhaltet den Adressbereich von 192.168.0.0 bis 192.168.255.255. In Kapitel 5.3.1 wird auf die Bedeutung dieses Adressbereichs näher eingegangen.⁶⁹

Als Übertragungsmedium wird meist ein Twisted-Pair-Kabel (Kabel mit verdrehten Adernpaaren) verwendet. Man unterscheidet bei diesen Netzkabeln zwischen verschiedenen Kabel-Kategorien. Für Signalübertragungen mit hohen Datenübertragungsraten werden derzeit überwiegend Kabel der Kategorie 5/5e, auch Cat5e-Kabel genannt, eingesetzt. Kabel einer geringeren Kategorie können weniger hohe Geschwindigkeiten erzielen und sind teilweise auch nicht oder zumindest schlechter abgeschirmt und somit anfälliger für Störungen. Höhere Kategorien sind dagegen leistungsfähiger und besser abgeschirmt. Derzeit sind mit Cat5e-Kabeln Übertragungsraten von 10 Mbit/s, 100 Mbit/s (Fast Ethernet) und 1000 Mbit/s (Gigabit-Ethernet) möglich. Als Stecker werden die gängigen RJ45-Anschlüsse verwendet. Die Verkabelung ist in den Normen DIN EN 50173 sowie der ISO/IEC 11801 standardisiert.⁷⁰



Abbildung 48: LAN-Kabel,
Quelle: Online-Produktkatalog Fa. ADP-Cable



Abbildung 49: RJ45-Stecker,
Quelle: Online-Produktkatalog Fa. Preistaktik

Ethernet definiert die ersten beiden Schichten des ISO/OSI-Referenzmodells (physikalische Schicht und Sicherungsschicht). Zusammen mit den Daten der darüber liegenden Schichten bildet es die Basis des TCP/IP-Protokolls, was in Kapitel 4.2.4 im Detail behandelt wird.

⁶⁹ (Internet Engineering Task Force (IETF) - Network Working Group, RFC1918, 1996)

⁷⁰ (Netzwerkartikel.de, Cat5-Patchkabel, 2013)

4.2.2 LAN via Stromnetz (PowerLAN, Powerline, dLAN)

Das in Kapitel 4.2.1 beschriebene klassische LAN kann auch über andere Leitungswege aufgebaut werden, wie beispielsweise das herkömmliche Stromnetz mit 230 V Spannung bei 50/60 Hz. Die Funktionsweise gleicht aus Sicht der Endgeräte dem herkömmlichen Ethernet, da es auf den gleichen Standards und Protokollen basiert. Lediglich der Übertragungsweg unterscheidet sich vom klassischen LAN - anstelle der Twisted-Pair-Kabel wird eine im Gebäude vorhandene elektrische Leitung zusätzlich zur Übertragung von Daten benutzt. Die Technologie für Datenverbindungen über das Stromnetz ist mittlerweile unter mehreren Bezeichnungen geläufig: PowerLAN, Powerline oder dLAN (Abk. für „Direct Local Area Network“).

Eine solche Technologie ist nicht ohne Standards möglich. Diese werden u. a. festgelegt durch das Institute of Electrical and Electronics Engineers (IEEE), einem Berufsverband von Ingenieuren der Bereiche Elektrotechnik und Informatik. Eine Vielzahl an Gremien des IEEE beschäftigt sich mit Standardisierungen, Normen und Regelwerken für verschiedene Technologiebereiche, eines davon speziell mit der Datenübertragung über das Stromnetz. Dieser sog. Powerline-Standard ist in IEEE Norm 1901 geregelt. Im Detail sind hierin die Frequenzbereiche und Leitungslängen sowie maximalen Geschwindigkeiten festgelegt.⁷¹

Die Organisation HomePlug Alliance, ein Verbund von Firmen zur Entwicklung und Förderung dieser Produktgruppe, hat ebenfalls Regelwerke herausgegeben. Diese Whitepaper beinhalten detaillierte Spezifikationen für verschiedene Standards. Der sog. Homeplug-Standard war in seiner ersten Version für Übertragungsraten von 14 Mbit/s bei einer Frequenz von 2-30 MHz ausgelegt, im zweiten Schritt mit Homeplug-Turbo für 85 Mbit/s. Eine weitere Steigerung auf 200 Mbit/s versprach der Homeplug-AV Standard, der mittlerweile durch Homeplug AV2 abgelöst wurde. Derzeit sind damit 600 MBit/s bei einer Erweiterung des Frequenzbereichs von 2 auf 86 MHz möglich. Die maximale Reichweite beträgt 300 Meter. Seit 2010 wird der neue Standard Homeplug GreenPHY als energieeffiziente und kostengünstige Variante für Smart Grid Anwendungen entwickelt. Der dafür optimierte Datendurchsatz ist auf 10 MBit/s reduziert⁷²

⁷¹ (Institute of Electrical and Electronics Engineers, IEEE 1901-2010, 2010)

⁷² (HomePlug Alliance, Whitepapers, 2013)

Für die praktische Umsetzung sind Adapter notwendig, welche die digitalen Datenströme in hochfrequente Signale zwischen 2-30 MHz (Homeplug-Standard) bzw. 2-86 MHz (Homeplug AV2-Standard) umwandeln.⁷³

Diese Adapter haben eine RJ45-Buchse, die mit dem Endgerät verbunden wird, und einen Netz- bzw. Schuko-Stecker für den Kontakt zur Steckdose.



Abbildung 50: PowerLAN-Adapter, Quelle: Online-Produktkatalog Fa. MSI-Computer

Für ein Netzwerk sind mindestens zwei Adapter notwendig, die miteinander kommunizieren können. Eine besondere Hierarchie gibt es dabei nicht, jeder Adapter ist gleichberechtigt. Man spricht in diesem Fall von einer Peer-to-Peer-Netzwerktopologie.

Bei der Übertragung wird das zur Verfügung stehende Frequenzspektrum zwecks Reduzierung der Störanfälligkeit in Kanäle aufgeteilt. Weitere Maßnahmen gegen Störsignale können Signalfilter und Fehlerkorrekturverfahren sein. Der sendende Adapter moduliert eine Vielzahl an Signalen gleichzeitig auf eine Trägerfrequenz (mit unterschiedlichen Phasen und Amplituden), der empfangende Adapter trennt mittels Bandpass die Signale wieder von der Trägerfrequenz und demoduliert sie. So können Daten über das herkömmliche 230V Stromnetz verteilt werden.

Geräte, die diesem Standard genügen, werden durch die HomePlug Alliance zertifiziert und in einer Liste auf deren Homepage publiziert. In Deutschland sind Hersteller wie Devolo, Allnet, Netgear oder MSI für Powerline-Adapter bekannt.

⁷³ (HomePlug Alliance, Whitepapers, 2013)

4.2.3 LAN via Funk (WLAN)

Die in den vorangegangenen Kapiteln beschriebenen Technologien für die Datenübermittlung waren kabelgebunden. Eine weitere Variante ist die Funktechnologie, besser bekannt unter dem Begriff WLAN (Wireless Local Area Network, auch Wireless LAN oder W-LAN).

Wie bei den PowerLAN-Netzwerken auch besteht bei WLAN hinsichtlich der Funktionsweise der Endgeräte (z. B. Computer oder Router) kein Unterschied zur Netzwerkverbindung über ein Kabel. Datensignale werden von einem Sender in elektromagnetische Wellen im GHz-Bereich moduliert. Bei der Modulation verändert das zu übertragende Datensignal eine sog. Trägerfrequenz im Hochfrequenzbereich. Das Sendesignal nutzt in diesem Bereich eine vom Datensignal abhängige Bandbreite. Die Daten werden vom Empfänger durch einen Demodulator wieder zurückgewonnen.

Grundsätzlich wird bei WLAN-Netzwerken zwischen zwei Betriebsarten, dem Ad-hoc-Modus und dem Infrastrukturmodus, unterschieden. Im Ad-hoc-Modus (gleich einem Peer-to-Peer Netzwerk) werden zwei oder mehrere Clients direkt miteinander verbunden, während die Clients im Infrastrukturmodus über einen Access-Point miteinander kommunizieren. Ein Access-Point ist eine Art zentrale Funkbrücke, die auch mit kabelgebundenen Netzwerken gekoppelt sein kann. Als klassisches Beispiel hierfür wäre ein WLAN-Router zu nennen.⁷⁴



Abbildung 51: Access Point, Quelle: Online-Produktkatalog Fa. TD-Link

⁷⁴ (telekommunikation-online Gutsch & Co. OHG, Was ist WLAN?, 2013)

Drahtlosnetzwerke, auf denen lokale WLAN-Netz basieren, sind nach einer Norm des „Institute of Electrical and Electronics Engineers“ definiert, dem sog. IEEE-802.11-Standard. Da die hardwareseitige Entwicklung stets voran schreitet, müssen auch die Standards angepasst werden, sodass aus der 802.11-Norm eine ganze Normen-Familie geworden ist, die sich durch Buchstabenkürzel im Titel unterscheidet, z. B. IEEE 802.11n. Der Buchstabe am Ende gibt z. B. Aufschluss über das verwendete Frequenzband und die unterstützte Geschwindigkeit. Die derzeit aktuellste Norm ist die IEEE 802.11n mit 600 Mbit/s auf 2,4 bzw. 5 GHz mit einem Frequenzband von 40 MHz. Die folgende Tabelle zeigt den Vergleich einiger Standards der IEEE 802.11-Reihe: ^{75,76}

Norm	Brutto-Datenrate	Netto-Datenrate	Frequenzbereich
IEEE 802.11	2 MBit/s	1,2 MBit/s	2,4 GHz
IEEE 802.11b	11 MBit/s	5 MBit/s	2,4 GHz
IEEE 802.11g	54 MBit/s	25 MBit/s	2,4 GHz
IEEE 802.11a	54 MBit/s	32 MBit/s	5 GHz
IEEE 802.11n	600 Mbit/s	240 MBit/s	2,4 GHz, 5 GHz

Tabelle 2: Vergleich von IEEE 802.11-Standards

Der Vollständigkeit halber seien noch weitere Funktechnologien aufgezählt, die zur Datenübermittlung in Computernetzwerken zum Einsatz kommen können. Die im Folgenden aufgezählten Funktechnologien unterscheiden sich von der gerade beschriebenen WLAN-Technologie, durch die Art der Datenübermittlung. Bis auf RFID können alle Technologien TCP/IP-Protokolle verarbeiten. Drahtlose Ad-Hoc-Gruppenverbindungen lassen sich auch mit Bluetooth, GSM, HSCSD, GPRS, UMTS oder LTE aufbauen, Infrarot hingegen nur mit jeweils 2 Geräten mit Sichtverbindung bis zu 5 m. Hinsichtlich der Reichweite sind Mobilfunknetze nicht mit der WLAN-Technologie vergleichbar, da sie landesweit mehrere Zehntausend Funkzellen umfasst und eine 500-fach höhere Sendeleistung hat. Mit der RFID-Technologie lassen sich keine Netzwerke aufbauen, sie dient lediglich der automatischen Identifizierung, Lokalisierung oder Datenerfassung. ⁷⁶

⁷⁵ (Kleijn, IEEE ratifiziert WLAN-Standard 802.11n, 2009)

⁷⁶ (Hein & Dr. Maciejewski, Wireless LAN, Funknetze in der Praxis, 2003)

Folgende Tabelle zeigt einen Vergleich der verschiedenen Verfahren hinsichtlich Reichweite, Frequenzbereich und Datenrate.⁷⁷

Verfahren	Reichweite	Frequenzbereich	Datenrate
RFID	0,5 -10 m	9 kHz - 125 GHz	100 kBit/s
Infrarot	5 m	340 GHz - 360 GHz	115 kBit/s
Bluetooth	10-100 m	2,40 GHz - 2,48 GHz	1 Mbit/s
GSM	netzabhängig	900 MHz / 1,80 GHz	9,6 kBit/s
HSCSD	netzabhängig	1,80 GHz	57,6 kBit/s
GPRS	netzabhängig	900 MHz	53,6 kBit/s
UMTS	netzabhängig	1,90 - 2,17 GHz	384 kBit/s

Tabelle 3: Vergleich von Funktechnologien

Mit Ausnahme von RFID können prinzipiell alle genannten Funktechnologien für den Aufbau eines Netzwerks zur Steuerung von Anlagen und Geräten in der Gebäudeautomation eingesetzt werden.

4.2.4 Netzwerkprotokoll TCP/IP

Die Kommunikation von Geräten und Anlagen untereinander oder auch mit zentralen Einheiten bzw. Servern funktioniert durch die Übermittlung von Datenpaketen und Protokollen. „Ein Protokoll ist ein Satz von Regeln, nach denen die Kommunikation zwischen zwei Kommunikationspartnern ablaufen muss.“⁷⁸ Diese Protokolle können über verschiedene Medien, wie beispielsweise Leitungsbahnen oder Funkstrecken, übertragen werden.

Eine Kommunikation zwischen zwei entfernten Geräten kann nur erfolgreich sein, wenn diese die gleiche Sprache sprechen. Diese Sprache muss sowohl von der Hardware als auch von den Übertragungseinrichtungen sowie den Schnittstellen gesprochen und verstanden werden. Die Art und Weise des Informationsaustauschs

⁷⁷ (Hein & Dr. Maciejewski, Wireless LAN, Funknetze in der Praxis, 2003)

⁷⁸ (Merz, Hansemann & Hübner, Gebäudeautomation, Kommunikationssysteme mit EIB/KNX, LON und BACnet, 2010)

wird in sog. Netzwerkprotokollen definiert. Für diese Protokolle sind Standards festgelegt, die das Kommunikationsverhalten bzw. den Austausch von Daten durch Formate und Regeln bestimmen.

Die Aufgabe von Protokollen ist, die sichere Übermittlung von Datenpaketen zu gewährleisten. Dazu ist eine Überprüfung am Ende des Sendevorgangs erforderlich. Mit Hilfe einer Prüfsumme kann festgestellt werden, ob ein Datenpaket beschädigt wurde. Ist das Datenpaket erfolgreich übermittelt worden, wird dies durch eine Empfangsbestätigung quittiert. Bleibt die Empfangsbestätigung aus oder ist das Datenpaket defekt, wird das Datenpaket erneut versandt. Dieser Vorgang wird so lange wiederholt, bis der erfolgreiche Empfang quittiert wird.

Die Grundlage für lokale Computernetzwerke und Internetverbindungen ist das sog. TCP/IP-Protokoll, das auf dem Ethernet aufsetzt. Diesem Protokoll liegt das sog. ISO/OSI-Referenzmodell zugrunde.

ISO steht dabei für die „International Organization for Standardization“, OSI ist die Abkürzung für „Open Systems Interconnection“.

„Das ISO/OSI-Referenzmodell ist ein internationaler Standard (ISO IS 7498) und beschreibt, wie man geschichtete Protokolle (Regelsätze zum Ablauf einer Kommunikation) erstellen kann. Weiterhin liefert es eine Beschreibung der Aufgaben, die in den einzelnen Protokollschichten zu implementieren sind.“⁷⁹

Das Modell legt die Regeln für den Ablauf einer Kommunikation fest. Dabei durchläuft jedes Datenpaket alle 7 Schichten des Modells, die jeweils eine bestimmte Aufgabe haben, in der hierarchisch angeordneten Reihenfolge - auf der Seite des Senders von Schicht 7 nach 1 und auf der Empfängerseite von Schicht 1-7. Hinter der Bezeichnung ISO/OSI verbirgt sich damit noch eine weitere Bedeutung: Die gegenläufige Buchstabenfolge steht symbolisch für den Ablauf der Kommunikation.

„Jede Schicht bietet der übergeordneten Schicht ihre Dienste an, z. B. überträgt die Bitübertragungsschicht den von der Sicherheitsschicht gelieferten Bitstrom als Spannungssignal auf dem Übertragungsmedium Kupferleitung.“⁷⁹

⁷⁹ (Merz, Hansemann & Hübner, Gebäudeautomation, Kommunikationssysteme mit EIB/KNX, LON und BACnet, 2010)

Nachstehend ist das TCP/IP-Protokoll auf Grundlage des ISO/OSI-Referenzmodells in eigener Darstellung abgebildet:

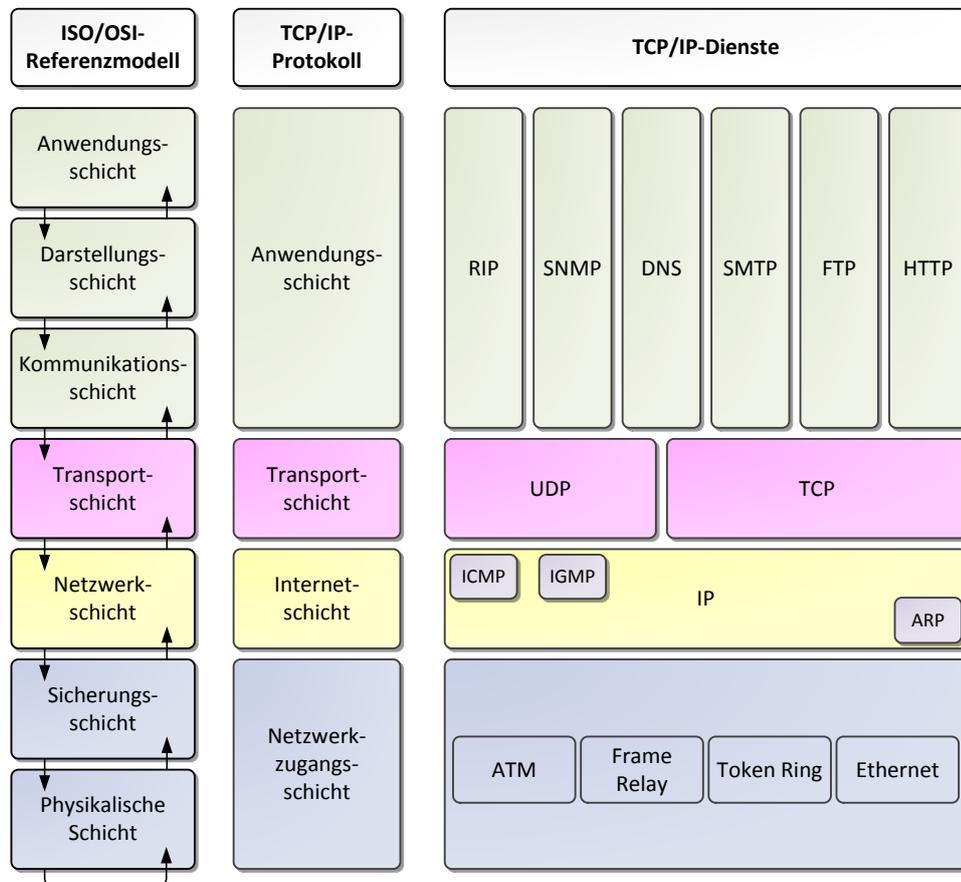


Abbildung 52: TCP/IP-Protokoll auf Grundlage des ISO/OSI-Referenzmodells, eigene Darstellung

- Anwendungsschicht:
Funktionen für Anwendungen sowie die Dateneingabe und -ausgabe
- Darstellungsschicht:
Umwandlung der systemabhängigen Daten in ein unabhängiges Format
- Kommunikationsschicht:
Steuerung der Verbindungen und des Datenaustauschs
- Transportschicht:
Zuordnung der Datenpakete zu einer Anwendung
- Netzwerkschicht:
Routing der Datenpakete zum nächsten Knoten
- Sicherungsschicht:
Segmentierung der Pakete in Frames und Hinzufügen von Prüfsummen
- Physikalische Schicht:
Umwandlung der Bits in ein zum Medium passendes Signal und physikalische Übertragung

Wie läuft nun der Datenaustausch konkret ab und welche Informationen beinhaltet das TCP/IP-Protokoll? Folgende Grafik zeigt den schematischen Aufbau bzw. Ablauf nach Furrer⁸⁰ in eigener Darstellung:

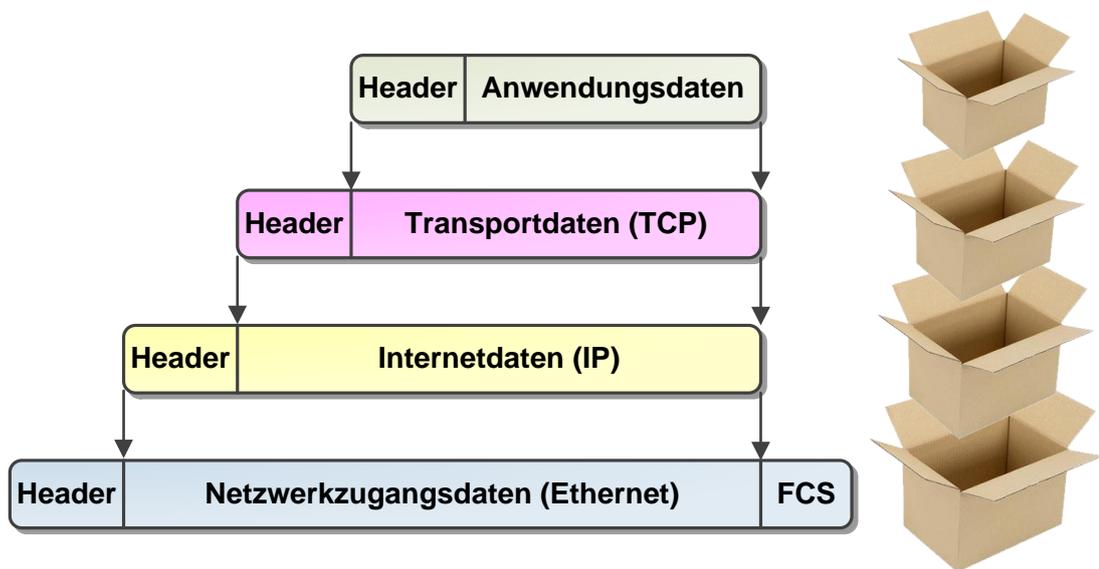


Abbildung 53: Geschichtetes Protokoll für die TCP/IP-Schichten nach Furrer („Industrieautomation mit Ethernet-TCP/IP und Web-Technologie“, 2003), eigene Darstellung

Je nach Anwendung werden Datenanfragen generiert, wie z. B. das Auslesen von Werten einer Datenbank. Die Anwendungsdaten werden gesammelt und mit einem Header versehen. Der Header ist vergleichbar mit einem Paketaufkleber, der den Inhalt des Pakets sowie die zur Adressierung und Übertragung erforderlichen Informationen enthält. Dieses Paket wird nun der Transportschicht übergeben, in das TCP-Protokoll eingebettet und wieder mit einem Header versehen. Das TCP-Protokoll wird auf die gleiche Weise in das IP-Protokoll eingebettet, was letztendlich der Netzwerkzugangsschicht übergeben, durch eine Prüfsumme ergänzt und via Ethernet versandt wird.

Der Vollständigkeit halber sind als weitere Netzwerkprotokolle AppleTalk, NetBEUI und IPX/SPX zu nennen, die aber gegenüber der weiten Verbreitung von IP-basierten Netzwerken zunehmend an Bedeutung verloren haben.⁸¹

⁸⁰ (Furrer, Industrieautomation mit Ethernet-TCP/IP und Web-Technologie, 2003)

⁸¹ (Harnisch, Netzwerktechnik, 2009)

5 Modellentwicklung zur Kopplung von CAFM und GA

Die Zielsetzung dieser Arbeit ist die Entwicklung einer webbasierten Lösung zur Zusammenführung von Feldbussystemen und Computernetzwerken mit Datenbank-anbindung und Steuerungsroutrinen aus einem CAFM-System heraus. Gebäudetechnische Anlagen sollen dabei möglichst komfortabel eingebunden, installiert und konfiguriert werden können.

Wie in den vorangegangenen Kapiteln beschrieben, basieren CAFM-Systeme auf reinen WAN/(W)LAN-Computernetzwerken mit TCP/IP-Protokoll, wohingegen die Gebäudeleit-technik über eigene Busleitungen mit Protokollen wie KNX, LonTalk oder dem herstellerunabhängigen BACnet kommuniziert. Hieraus leiten sich die Möglichkeiten von Schnittstellen zur Steuerung ab. Hauptaufgabe ist die Zusammenführung der unter-schiedlichen Netze hinsichtlich Übertragungsprotokollen und -medien. Bei der gesamten Betrachtung steht aufgrund der Anforderung einer Datenbank-anbindung und CAFM-Integration die Verwendung einer reinen Computernetzwerktechnologie im Vordergrund.

Der Großteil an realisierten Schnittstellenkonzepten dient dem reinen Datenaustausch auf Managementebene. Die Erweiterung des Informationsaustauschs von Befehlen zur Steuerung auf Feldebene führt zu einer Verschmelzung von Automationsebene und Managementebene. Die Steuerzentrale bildet das CAFM-System. Hinsichtlich Netzwerk-fähigkeit und „Intelligenz“ werden neue Anforderungen an die Endgeräte gestellt.

Der Leitgedanke dabei ist, bestehende Techniken und Standards zu nutzen, um daraus neue Lösungen zu schaffen. Der Entwicklungsprozess gliedert sich in folgende Abschnitte:

- Im ersten Schritt werden verschiedene Varianten der Systemintegration anhand hardwareseitiger Schnittstellenmodelle vorgestellt und analysiert.
- Unter der Berücksichtigung maßgebender Anforderungskriterien erfolgt eine Modell-bewertung zwecks Präferenz der zielführenden Variante.
- Nach Festlegung des Aufbaus und der Netzwerkstruktur wird die technische Umsetzung im Detail beschrieben.
- Die Vorgänge werden erfasst und als Prozesskette in einem Workflow-Modell dargestellt.
- Um die hersteller- und produktunabhängige Funktionalität zu gewährleisten, werden abschließend Standards definiert.

Modell, Prozesse und Standards sind Grundlage für die praktische Umsetzung, die im darauf folgenden Kapitel 6 behandelt wird.

5.1 Varianten der Systemintegration durch hardwareabhängige Schnittstellenmodelle

Basierend auf der klassischen Netzwerkstruktur eines CAFM-Systems aus Kapitel 5.1.1 werden in den darauf folgenden Abschnitten 5.1.2-5.1.6 fünf verschiedene Modelle aufgeführt, welche die grundsätzlichen Integrationsvarianten erklären.

5.1.1 Netzwerkstruktur eines CAFM-Systems

Die folgende Variante beschreibt die übliche Netzwerkstruktur eines CAFM-Systems als Basis der in den folgenden Kapiteln vorgestellten Möglichkeiten einer Anbindung zur Gebäudeautomation.

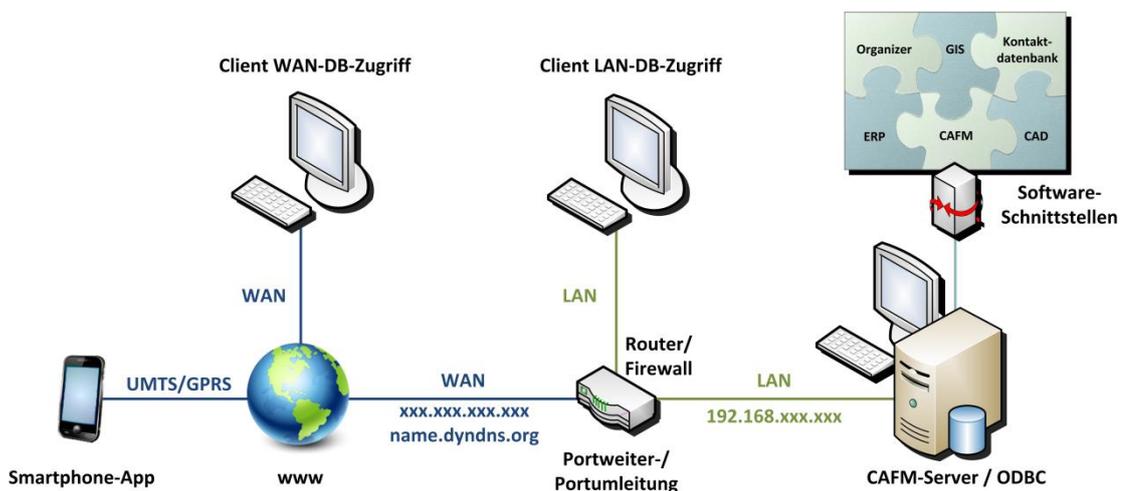


Abbildung 54: Klassische Netzwerkstruktur eines CAFM-Systems, eigene Darstellung

CAFM-Systeme sind überwiegend in Form einer Client/Server-Applikation aufgebaut, nur wenige Hersteller bieten eine vollständig webbasierte Version an. In beiden Fällen beherbergt der Server die Datenbank und stellt die notwendigen Dienste sowie administrative Funktionen, wie z. B. Benutzerverwaltung, Steuerung von Zugriffsrechten, Archivierungsfunktionen, Protokollierung etc., zur Verfügung.

Bei der Client/Server-Struktur wird die Applikation auf den Clients lokal installiert und ausgeführt. Bei der webbasierten Variante dient hingegen der Browser als Benutzeroberfläche. Die Clients können über verschiedene Wege mit dem Server verbunden sein, wie in Abbildung 54 dargestellt. Die Cloud-Lösung wird im weiteren

Verlauf der Modellentwicklung aus Gründen des internen Hostings zur Gewährleistung der eigenen Datenhoheit nicht betrachtet.

Die CAFM-Anwendung bildet logische FM-Prozesse ab. Erfasste Daten, wie beispielsweise Flächen- oder Anlagenattribute, werden in der Datenbank abgelegt bzw. ausgelesen. Meist gibt es individuelle Software-Schnittstellen für den Datenaustausch mit anderen Systemen, wie z. B. GIS, ERP, CAD oder Businessanwendungen, wie Organizer und Kontaktdatenbanken.

Ob die Bedienung über eine lokale Applikation (Client/Server-Struktur) oder über den Browser (webbasierte Lösung) erfolgt, spielt für die hardwareseitige Netz-anbindung und softwareseitigen Datenbankabfragen keine Rolle. In beiden Fällen sind die Teilnehmer mit einem Computernetzwerk verbunden (lokales LAN oder globales WAN) und die Daten werden mittels SQL-Befehlen aus der Datenbank ausgelesen, verändert oder neu angelegt. Der Unterschied zwischen der Client/Server-Struktur und einer webbasierten Lösung liegt in der Programmierung der Benutzeroberfläche, den bereitgestellten Diensten und verwendeten Protokollen. Gleich ist die Datenbankanbindung bzw. Datenbankschnittstelle via ODBC (Open Database Connectivity), welche einen autorisierten Datenbankzugriff erst möglich macht. Über die IP-Adresse des Servers ist die Datenbank aus dem LAN oder WAN erreichbar, unabhängig ob von einer Applikation oder einem Browser.

Konkret sendet der lokale oder ferne Client einen SQL-Befehl, der durch einen Router an die entsprechende IP-Adresse des Servers weitergeleitet wird und mit Hilfe der ODBC-Schnittstelle die Werte der Datenbank auslesen und/oder schreiben kann. Aktuelle Werte werden auf dem gleichen Weg an den Client zurückgegeben, um sie in der entsprechenden Maske der Applikation oder dem Browser anzuzeigen.

Bei diesem Vorgang macht es einen Unterschied, ob sich der Client innerhalb des LAN-Netzwerks (interner Zugriff, siehe Kapitel 5.3.1) oder außerhalb des LAN-Netzwerks (Fernzugriff, siehe ebenfalls Kapitel 5.3.1) befindet. Der Router muss permanent unter einer gleichbleibenden (statischen) Adresse erreichbar sein. Ist die globale IP-Adresse nicht statisch, hilft ein dynamischer Domain-Name-System-Eintrag, wie z. B. DynDNS (siehe Kapitel 5.3.2). Sofern der Server nicht direkt an das WAN angebunden ist, also keine öffentliche IP-Adresse hat, muss der Zugriff über eine Portweiterleitung/Portumleitung des Routers sichergestellt werden (siehe Kapitel 5.3.3).

5.1.2 Unidirektionale Kopplung des Feldbus über lokale Schnittstellenlösung

Die folgende Variante beschreibt eine unidirektionale GA-Anbindung über eine lokale Schnittstellenlösung zwischen CAFM-System und Bussystem mit Leitrechner, wie in der Praxis häufig realisiert.

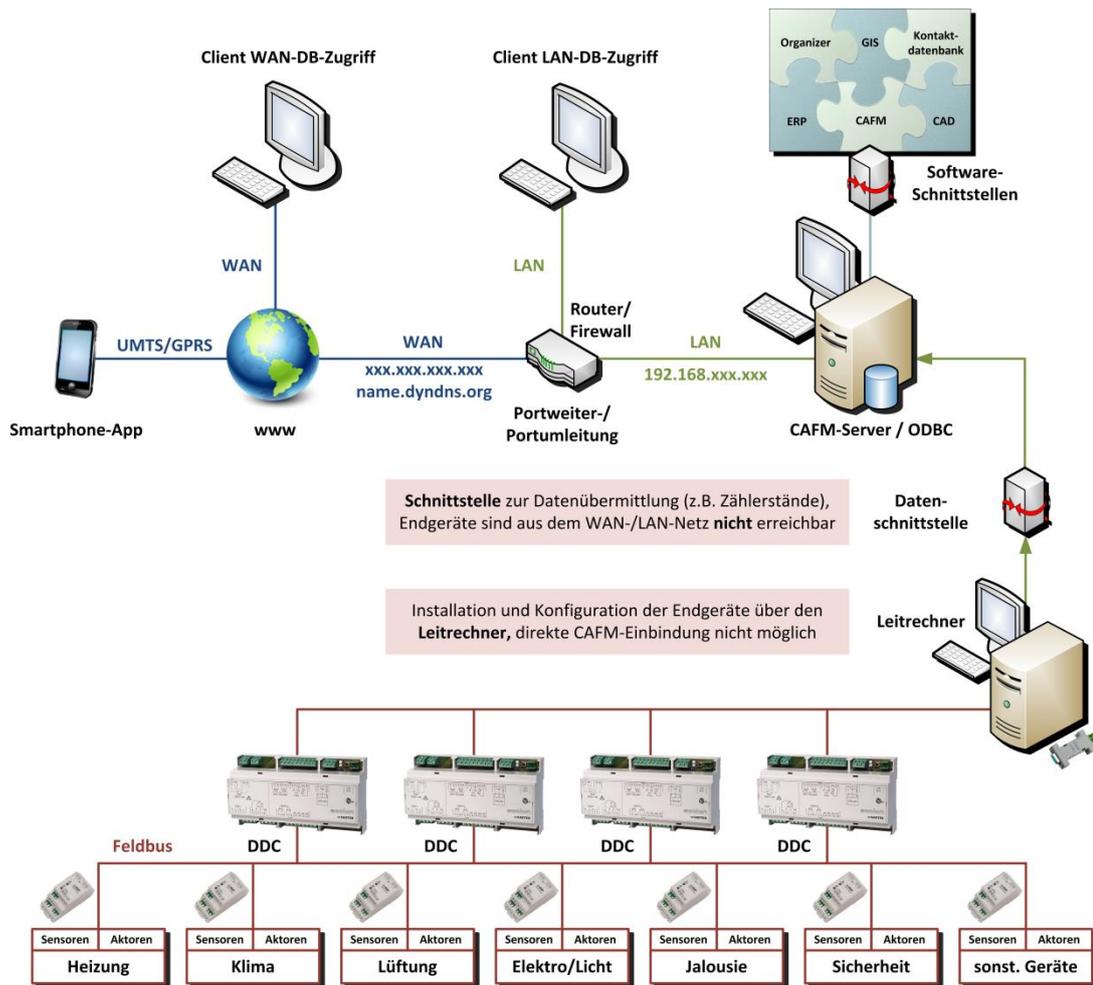


Abbildung 55: Unidirektionale GA-Anbindung über eine lokale Datenschnittstelle zwischen CAFM-System und Bussystem mit Leitrechner, eigene Darstellung

Die gängigste Methode der Kopplung von CAFM-Systemen zur GA ist – ausgehend von dem in Kapitel 5.1.1 beschriebenen Aufbau – eine einfache Software-Schnittstelle auf Managementebene zum Austausch von Daten, wie beispielsweise Zählerstände oder Verbrauchs- und Zustandsdaten. Diese Daten werden in einem vorgegebenen Intervall, meist stündlich oder täglich bzw. nächtlich, mit der Datenbank des CAFM-Systems synchronisiert.

Der Austausch zwischen CAFM-Server und Leitrechner auf der GLT-Seite ist dabei nicht bidirektional, sondern unidirektional von der GA zum CAFM-System. Der CAFM-Server und die über das Internet angebotenen Clients haben somit keinen Zugriff auf den Feldbus. Eine Steuerung der Busteilnehmer ist nur über den Leitrechner möglich.

Auf der GA-Seite gibt es viele verschiedene Technologien, wie in Kapitel 4.1 behandelt. Der hier dargestellte Aufbau basiert auf dem 3-Ebenen-Modell mit Management-, Automations- und Feldebene, wie in den Kapiteln 3.3 und stellenweise auch 4.1 gezeigt. Die Möglichkeiten der Gestaltung einer GA-Landschaft sind mit den vorgestellten Bus-Systemen sehr variabel. Daher sind im Folgenden nur beispielhafte Ausführungen der technischen Anbindung des Feldbusses an die GLT-Zentrale der Managementebene sowie der weiterführenden Datenübermittlung an CAFM-Systeme genannt.

Busgeräte der Feldebene kommunizieren über ein 2- oder 4-adriges Kabel mit der Automationsebene und sind über entsprechende Adapter mit dem Leitrechner der Managementebene verbunden. Diese Adapter werden als RS232/RS485-Converter (siehe Abbildung 56) bezeichnet. Die Leitungskabel werden direkt mit dem Adapter verdrahtet, der an den seriellen COM-Port des Leitrechners angeschlossen ist. Die Software interpretiert dann die eingehenden und ausgehenden Datentelegramme des Ports zur Steuerung der Busgeräte.



Abbildung 56: RS232/RS485-Converter, Quelle: Online-Produktkatalog Fa. Abus

Eine weitere Variante des Anschlusses von Busgeräten an den Leitrechner ist der Weg über einen ISA-Bus-Controller. Diese Steckkarte wird an einem freien ISA-Slot auf dem Mainboard des Leitrechners verbaut, der eine direkte Anbindung des Systembusses an den Leitrechner über eine RJ-11 Buchse ermöglicht. Über einen

Dual-Port-RAM werden die Telegramme zwischen dem Systembus und der übergeordneten Managementebene ausgetauscht. Folgende Abbildung zeigt einen ISA-Bus-Controller der Firma Sauter, die den Anschluss des firmeneigenen Systembus „novaNet“ an einen PC ermöglicht:



Abbildung 57: ISA-Bus-Controller, Quelle: Online-Produktkatalog Fa. Sauter

Für die Anbindung von Bussystemen an Computernetzwerke werden die beiden zentralen Einheiten, der CAFM-Server und der Leitreechner, über das lokale Netzwerk miteinander verbunden. Die Datenschnittstelle der beiden Stationen wird mit Hilfe der Programmierung von Austauschformaten und Übertragungsprotokollen realisiert. Die im Leitreechner gespeicherten Werte werden ausgelesen, ähnlich einer gewöhnlichen Datenbankabfrage über SQL-Befehle. Hierfür muss die GLT-Software die Daten in geeigneter Form bereitstellen und den Zugriff ermöglichen. Die in Relation stehenden Datenfelder müssen aufeinander abgestimmt sein. Seitens der CAFM-Software, die den Austausch auslöst, muss ein entsprechender Programmbaustein für den Datenimport implementiert werden, was vom CAFM-Hersteller umzusetzen ist.

Schnittstellenspezifikationen, die es anderen Programmen ermöglichen, auf eigene Daten zuzugreifen, nennt man auch APIs (Application Programming Interfaces). Da es in der Gebäudeautomation viele hersteller- und produktabhängige Schnittstellen gibt, hat man mit der OPC-Schnittstelle (OLE for Process Control) versucht, den herstellerunabhängigen Datenaustausch zwischen verschiedenen Systemen zu standardisieren.⁸²

⁸² (May, CAFM-Handbuch, 2013)

5.1.3 Unidirektionale Kopplung des Feldbus über webbasierte IP-Controller

Die folgende Variante beschreibt eine unidirektionale GA-Anbindung über eine globale Schnittstellenlösung zwischen CAFM- und webbasiertem Bussystem, wie sie in der Praxis durch neue Technologien zunehmend zum Einsatz kommt.

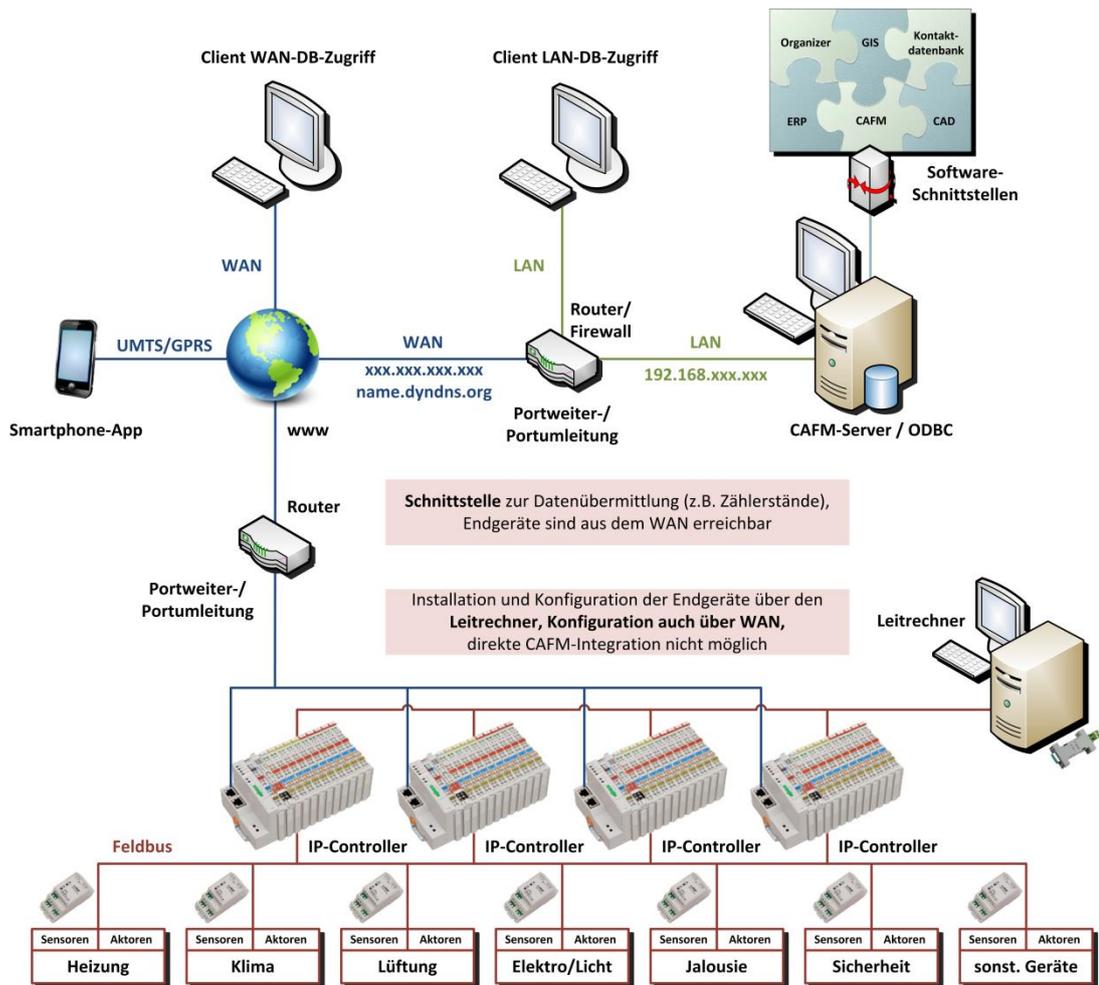


Abbildung 58: Unidirektionale GA-Anbindung über eine webbasierte Datenschnittstelle zwischen CAFM-System und Bussystem mit Leitreechner, eigene Darstellung

Der Aufbau beider Systemabschnitte für sich gleicht hardwareseitig der vorangegangenen Ausführung. Der Unterschied liegt in der Webfähigkeit und -anbindung des Bussystems. Durch netzwerkfähige DDCs, sog. IP-Controller oder IP-Router, ist die GLT an das Internet angebunden, womit Daten und Funktionen der Feldgeräte global bereitgestellt werden können. Das webbasierte CAFM-System kommuniziert nicht mit dem Leitreechner, wie in Abschnitt 5.1.2 dargestellt, sondern bezieht die Daten über die genannten Interfaces. Der Datenaustausch erfolgt also unidirektional über ein globales Netzwerk.

Diese IP-Controller bzw. IP-Router wurden bereits in den Unterkapiteln von Abschnitt 4.1 vorgestellt. Zusammenfassend sind der KNXnet/IP-Controller (Kapitel 4.1.1), der LON IP-Router (Kapitel 4.1.2), der Buskoppler LCN-PK (Kapitel 4.1.3), der BACnet-Controller (4.1.4) und der digitalSTROM-Server (Kapitel 4.1.5) zu nennen. Sie sind nicht zu verwechseln mit Routern eines Computernetzwerkes, auch wenn die Funktion im Grunde die gleiche ist. Die im Bussystem eingesetzten Controller/Router sind eine Schnittstelle zwischen Feldbus (Bus-Adresse) und LAN (lokale IP-Adresse), die im Computernetzwerk verwendeten Router sind eine Schnittstelle zwischen DSL bzw. WAN (globale IP-Adresse) und LAN (lokale IP-Adresse).

Die Teilnehmer der jeweiligen Bussysteme (wie KNX, LON, LCN, digitalSTROM, EnOcean u. a.) kommunizieren über herstellereigene Protokolle oder das herstellerunabhängige Protokoll BACnet. Die Controller ermöglichen mit einem integrierten Web-Server die Bereitstellung einiger Feldbus-Funktionen im Internet und kommunizieren mit Hilfe des TCP/IP-Protokolls. Über spezielle Softwarelösungen lassen sich Busteilnehmer auch über eine grafische Oberfläche ansprechen. Damit ist die Konfiguration und Steuerung über einen Internetbrowser in einem begrenzten Rahmen möglich (bidirektionaler Informationsfluss), die Bedienung aus einem webbasierten CAFM-System heraus aber nicht. Dies könnte zwar mit der Programmierung einer Implementierungsroutine ermöglicht werden, doch die dazu nötigen Geräteinformationen und Steuerbefehle stellt der Controller nicht zur Verfügung. Mit einer geräteabhängigen Schnittstelle ließe es sich technisch dennoch realisieren, doch Einzellösungen sind nicht das Ziel. Daher bleibt es zwischen CAFM und GA in diesem Modell (vorerst) nur bei einer einfachen, unidirektionalen Datenübermittlung.

Wie beim Schnittstellenkonzept des vorangegangenen Kapitels auch, wird die Datenübermittlung mit Hilfe der Programmierung von Austauschformaten und Übertragungsprotokollen realisiert. Für das Auslesen der vom Web-Server des IP-Controllers abrufbaren Werte müssen die Daten in geeigneter Form bereitgestellt und der globale Zugriff durch den Router ermöglicht werden. Die den Vorgang auslösende CAFM-Software greift über die globale IP-Adresse und den freigegebenen Port des GA-seitigen Routers auf die bereitgestellten Daten zu und importiert entsprechende Werte in die eigene Datenbank.

5.1.4 Bidirektionale Kopplung des Feld-/Funkbus über lokalen USB/COM-Port

Die folgende Variante beschreibt eine bidirektionale GA-Anbindung über den seriellen Bus (USB/COM-Port) des CAFM-Servers, wie beispielsweise mit Funkbus-Geräten der Ratio©-Produktreihe (Omnio AG) praktiziert.

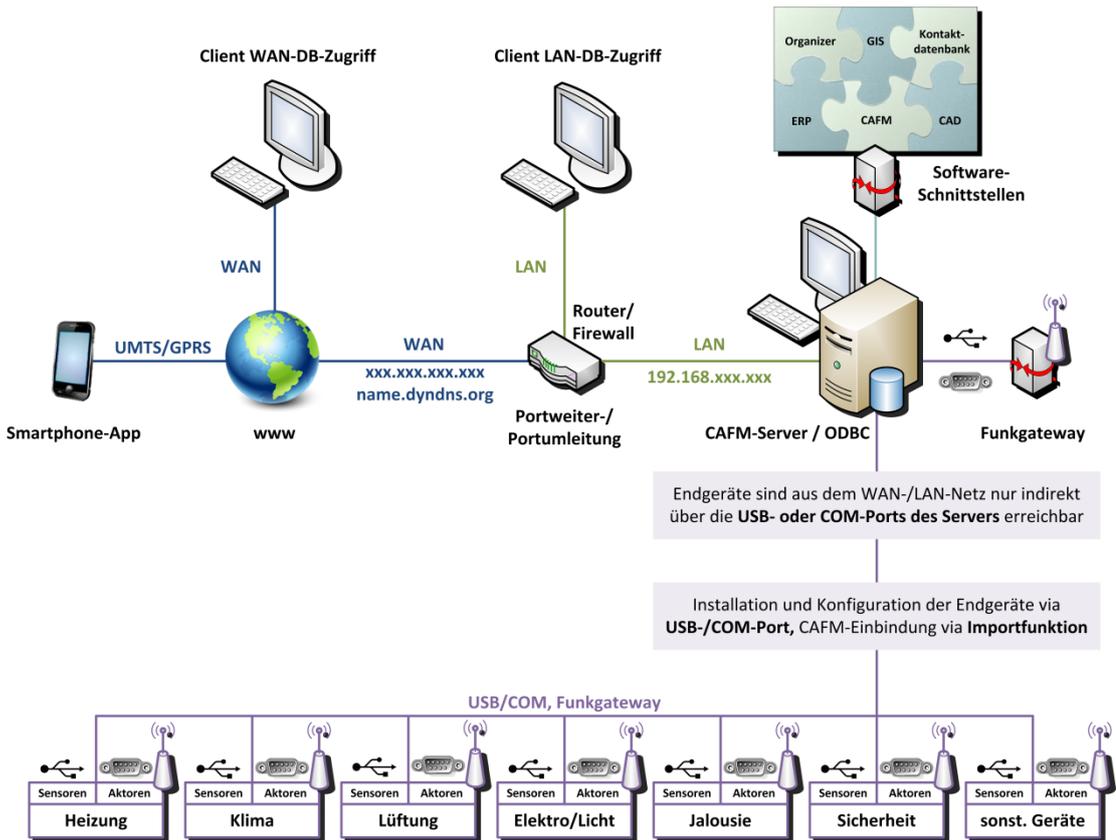


Abbildung 59: Bidirektionale GA-Anbindung über den seriellen Bus (USB/COM-Port) des CAFM-Servers, eigene Darstellung

Dieser Aufbau unterscheidet sich GA-seitig grundlegend von dem eines klassischen Bussystems. In diesem Modell wird die Gebäudeautomation direkt in das Computernetzwerk eingebunden. Sensoren und Aktoren sind nicht mehr über DDCs an einen separaten Leitrechner angeschlossen, sondern direkt mit dem (CAFM-) Server gekoppelt. Im Falle einer reinen Funkbuslösung erfolgt die Integration über ein an den USB-/COM-Port angeschlossenes Gateway. Abbildung 59 zeigt den Aufbau, wie er mit Geräten der EnOcean-Technologie realisiert werden kann. Generell können auch leitungsgebundene Geräte anderer Hersteller direkt über den USB-/COM-Port angeschlossen werden, wie in der Abbildung angedeutet. Die weitere Betrachtung beschränkt sich aber auf die Funkbus-Variante.

Die Kommunikation der Geräte untereinander via Funk ist kein IP-basierter Netzwerkstandard (WLAN), sondern eine eigene Funktechnologie, die spezielle Gateways und Protokolle erfordert (siehe Kapitel 4.1.6). Die Endgeräte sind also nicht netzwerk- bzw. WLAN-fähig, sie basieren auf der EnOcean-Funktechnologie.

Im Folgenden sei der Aufbau anhand eines Beispiels gezeigt, wie es in der Praxis mit Hilfe der bereitgestellten Funkbus-Geräte umgesetzt wurde. Zur Verfügung standen ein USB-Funkgateway, verschiedene Schalt- und Dimm- und Thermostataktoren, Raumtemperaturfühler und Fensterkontakte. Zur Steuerung kam die Software „myHomeControl“ (Version V1.2.8 SP11) der BootUp GmbH zum Einsatz.

Das Gateway wurde direkt an den USB-Port des Testcomputers angeschlossen und mit Hilfe eines vom Hersteller mitgelieferten Treibers installiert. Anschließend mussten die Geräte aufeinander abgestimmt bzw. „eingelernt“ werden. Die Anbindung des Schaltaktors erfolgte durch das Senden eines Signals per Tastendruck während des Suchmodus des Gateways. Durch diese manuelle Konfiguration waren die Geräte miteinander verbunden und konnten mit Hilfe der Software bedient werden. Abbildung 60 zeigt das Anlagenschema in eigener Darstellung mit Bildmaterial der Seiten www.omnio.ch, www.aktor.de und www.hans-wilmsen.de.

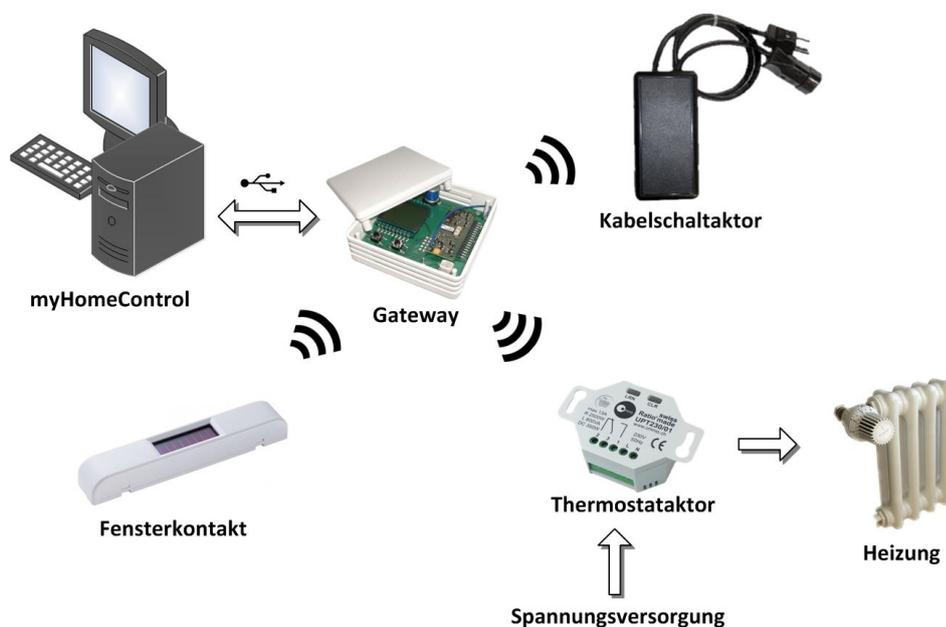


Abbildung 60: Anlagenschema Ratio©-Funkbussystem, eigene Darstellung mit Bildmaterial aus Online-Produktkatalogen Fa. Omnio AG und Akktor GmbH

Aufgrund der Lizenzbestimmungen der Software konnte der Test mit nur einem Aktor durchgeführt werden. Die Wahl fiel auf den Kabelschaltaktor KS230/02, in Kombination mit dem USB-Funkgateway APG03B-USB.

Die Software „myHomeControl“ ist nicht webbasiert, es handelt sich um eine lokal zu installierende Applikation. Als Lösungen zur Fernsteuerung sind Tools wie Remotedesktop, TeamViewer oder RDSPlus vorgesehen.

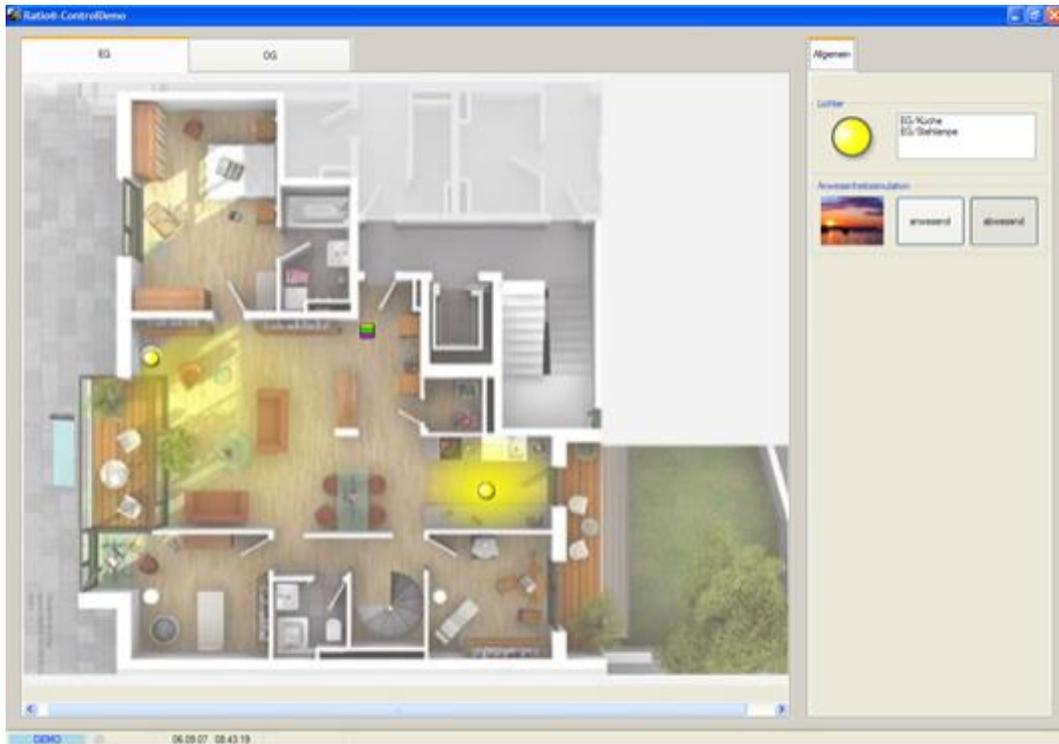


Abbildung 61: Benutzeroberfläche von myHomeControl, Quelle: BootUp GmbH

Aus dieser Situation heraus entstand ein eigenes, webbasiertes Tool, mit dem es möglich ist, dem Funkgateway einen Steuerbefehl zum Ein- bzw. Ausschalten des Kabelschaltaktors zu senden. Die notwendigen Befehle standen nicht zur Verfügung, konnten aber beim manuellen Ein- und Ausschaltvorgang am USB-/COM-Port abgefangen werden.

- Ausschaltbefehl: A5 5A 6B 05 30 00 00 00 FF FF FF 81 10 2E
- Einschaltbefehl: A5 5A 6B 05 10 00 00 00 FF FF FF 81 10 0E

Das Tool wurde auf einem Webserver installiert und war somit über eine URL abrufbar und über den Browser zu bedienen. Der Programmbaustein wird Bestandteil der in Kapitel 6.3 programmierten Anwendung „FMControl“.

5.1.5 Bidirektionale Kopplung netzwerkfähiger Komponenten über lokales LAN

Die folgende Variante beschreibt eine bidirektionale GA-Anbindung über das lokale, interne Netzwerk (LAN) durch netzwerkfähige Endgeräte, wie es in der CAFM-Praxis noch nicht angewandt wird.

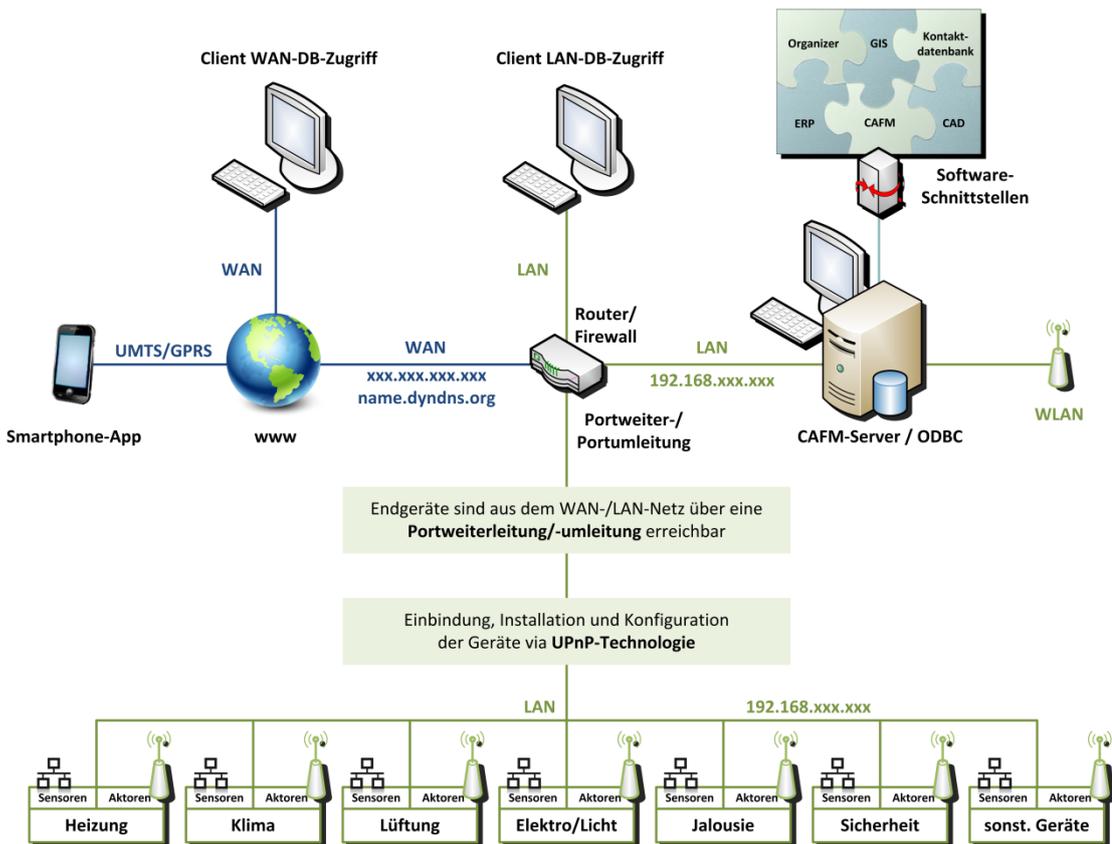


Abbildung 62: Bidirektionale GA-Anbindung über das interne LAN-Netzwerk durch netzwerkfähige Endgeräte, eigene Darstellung

Die Idee einer Verschmelzung von Management und Automationsebene kommt in diesem Konzept zum Tragen. Das Modell basiert auf der Methode, netzwerkfähige Endgeräte über einen (WLAN-)Router direkt in ein lokales Netzwerk einzubinden. Wie der grafischen Darstellung zu entnehmen ist, sieht diese Lösung keine Steuereinheiten (DDCs) oder Gateways mehr vor. Das mag zunächst verwunderlich erscheinen, doch dies ist der wesentliche Ansatz einer neuen Entwicklung. Der Einsatz von DDCs oder Gateways ist nach dieser Methode aber auch nicht mehr nötig, da die Feldgeräte durch eine eigene IP-Adresse identifiziert und auf lokaler

Ebene direkt erreichbar sind. Die Feldgeräte unterscheiden sich von der Erreichbarkeit und der Kommunikation nicht von der des Servers, da sie ein gleichberechtigter Teilnehmer des LAN sind. Auch auf globaler Ebene ist ein webbasierter Zugriff durch eine Portweiterleitung/-umleitung möglich. Die Funktionsweise ist einfach: Anfragen werden über die Portfreigabe gefiltert und an das entsprechende Endgerät weiter- bzw. umgeleitet.

Die Kommunikation erfolgt auf Basis der TCP/IP-Protokolle, siehe Kapitel 4.2.4. Logische Prozesse, Szenarienschaltung und Steuerbefehle, die vom Gerät selbst bereitgestellt werden müssen, werden von zentraler Stelle, dem CAFM-Server, geleitet. Entsprechende Logarithmen müssen implementiert werden. Diese Variante stellt neue Anforderungen an die Endgeräte (Sensoren, Aktoren).

In allen bisher vorgestellten Modellen fand ein wichtiger Aspekt aber noch keine Berücksichtigung: die Forderung einer komfortablen Geräteanbindung und automatisierten Installation - die Idee des „Plug and Play“ auf Netzwerkebene.

Das Verfahren „Plug and Play“ ist von USB-Geräten her bekannt. Nach dem Einstecken des Kabels (Plug) wird das Gerät erkannt und ggf. der fehlende Treiber installiert, sodass es sofort betriebsbereit ist und verwendet werden kann (Play).

Auch auf Netzwerkebene gibt es ein solches Verfahren, wie es in der Unterhaltungselektronik schon vielfach Anwendung findet. Dabei werden netzwerkfähige Massenspeicher mit Musik- und Video-Inhalten für das sog. „Streaming“ mit dem Heimnetzwerk (LAN) verbunden und von einem zentralen Receiver automatisch erkannt. Dieses Verfahren nennt sich „Universal Plug and Play“ (UPnP). Es beruht auf dem gleichen Prinzip, im Gegensatz zur lokalen USB-Variante aber netzwerkbasiert. Damit ist ein Zugriff auf verschiedene Geräte (unabhängig vom Hersteller) über ein lokales IP-basiertes Netzwerk möglich. Grundlage ist eine Reihe von standardisierten Netzwerkprotokollen und Datenformaten, dem sog. UPnP-Standard. Heute gibt es eine Vielzahl an zertifizierten Geräten, die diesem Standard entsprechen. Die automatische Erkennung und Kommunikation von UPnP-fähigen muss aber nicht zwingend kabelgebunden sein, wie der Begriff „Plug“ vermuten lässt. UPnP funktioniert auch über das WLAN.⁸³

⁸³ (IT-Administrator, Universal Plug and Play, 2013)

Das in diesem Kapitel vorgestellte Modell zur bidirektionalen Kopplung netzwerkfähiger Komponenten über das lokale (W)LAN bietet einen entscheidenden Vorteil: Es ist die einzige Variante, mit der sich UPnP realisieren lässt.

Dieser Vorteil ist zugleich ein innovativer und neuartiger Lösungsansatz im Bereich des Facility Managements und der Gebäudeautomation: einfache Installation und direkte Anlagensteuerung durch Implementierung bereitgestellter Steuerbefehle. Hierfür wäre es notwendig, Protokolle und Austauschformate zu definieren. Kern der neuen Standardisierung wäre die Festlegung von Inhalten der bereitzustellenden XML-Files.

5.1.6 Bidirektionale Kopplung internetfähiger Komponenten über globales WAN

Die folgende Variante beschreibt eine bidirektionale GA-Anbindung über das globale, öffentliche Netzwerk (WAN) durch internetfähige Endgeräte, wie sie in der Praxis vereinzelt mit speziellen Produkten realisiert ist.

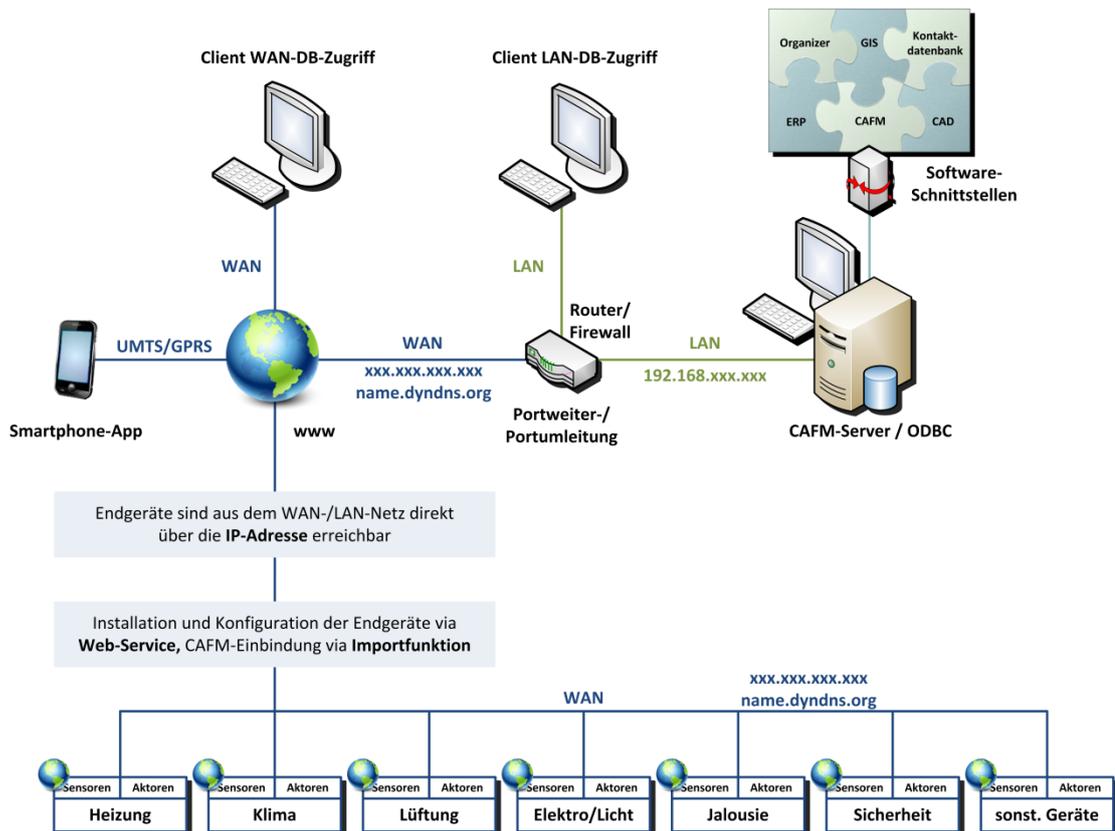


Abbildung 63: Bidirektionale GA-Anbindung über das öffentliche WAN-Netzwerk durch internetfähige Endgeräte, eigene Darstellung

Diese Integrationsvariante zeigt ein Modell, welches Bus- und CAFM-System mit Hilfe internetfähiger Endgeräte zusammenführt. Idee und Aufbau sind sehr ähnlich der in Kapitel 5.1.5 dargestellten Architektur. Hier geht es sogar noch einen Schritt weiter: Die Geräte sind über das globale Netzwerk (WAN) statt dem lokalen Netzwerk (LAN) integriert. Die gewerkeabhängigen Endgeräte (Sensoren/Aktoren) arbeiten dank eigener Prozessoren, integriertem Router und einer mobilen Internetanbindung (Wireless Wide Area Network, kurz WWAN) autark. Dadurch hat man über das Internet direkten Zugriff auf die Anlagen, ohne dass ein zentraler Server benötigt wird, der diese Geräte verwaltet. Über eine geräte- und

herstellerabhängige Software lassen sie sich von zentraler Stelle aus konfigurieren und steuern. Die Kommunikation basiert auch hier auf dem TCP/IP-Protokoll. Der integrierte Web-Server erlaubt den Zugriff aus der Ferne.

Diese Technologie findet nach eigenen Recherchen noch keine Anwendung in den klassischen Gewerken wie Heizungs-, Klima- oder Lüftungstechnik. Bekannteste Anwendung dieser Technologie sind sog. IP-Kameras, wie sie in der Sicherheitstechnik zur Überwachung eingesetzt werden.

Diese Funktionalität scheint ein großer Vorteil dieses Modells zu sein. Doch es gibt einen entscheidenden Nachteil: eine „Komfortanbindung“ via UPnP ist über das WAN nicht möglich.

Konfiguration und aktive Steuerung der Geräte sind zwar über einen Internetbrowser möglich, auch in grafischer Darstellung. Doch die Bedienung über ein an das Netzwerk angeschlossenes CAFM-System ist nicht ohne weiteres umsetzbar. Eigenschaften und Dienste der Endgeräte müssten wie im Modell aus Kapitel 5.1.5 über eine zu programmierende Importfunktion eingebunden und angepasst werden. Doch ohne UPnP werden entsprechende Informationen und Daten nicht automatisiert bereitgestellt.

5.2 Bewertung und Vergleich unter Beachtung maßgebender Anforderungskriterien

Für die Auswahl eines der in Kapitel 5.1 vorgestellten Modelle sind Anforderungskriterien im Sinne von Mindestanforderungen zu treffen und zu bewerten. Eine Entscheidungsmatrix soll bei der Auswertung helfen. Abschließend werden Vor- und Nachteile des gewählten Modells erörtert.

Grundsätzlich soll das Modell folgende Eigenschaften haben: Die Kommunikation über ein reines Computernetzwerk ist obligatorisch. Als webbasierte Zentrale dient der CAFM-Server, um die Summe aller Anlagen verwalten zu können. Gateways können entfallen, wenn netzwerkfähige Sensoren/Aktoren (Endgeräte) vorhanden sind. Eine Komfortanbindung via Plug and Play soll möglich sein – der Schlüssel ist UPnP. Dadurch können Informationen automatisiert für den Datenbankimport und zur CAFM-seitigen Steuerung zur Verfügung gestellt und übermittelt werden. Hierfür wird die Definition von Protokollen bzw. bereitzustellender XML-Dateien erforderlich. Alle Geräte müssen auch weiterhin herkömmlich (manuell) bedient werden können. Weitere Anforderungskriterien sind Netzstrukturoptimierung, Sicherheit, Wirtschaftlichkeit, Historisierung, Hilfsfunktionen für das Anlagen- und Wartungsmanagement sowie einfache Prozessketten.

5.2.1 Anforderungskriterien

Für die Bewertung wurden die 10 wichtigsten Aspekte in Bezug auf die Zielsetzung herangezogen. Zusätzlich sind sie mit einem Faktor versehen, der die Gewichtung des Kriteriums mit einer Punktebewertung widerspiegelt und eine objektive Priorisierung ermöglicht.

- **Reines Computernetzwerk**

Bei der gesamten Betrachtung steht aufgrund der Anforderung einer Datenbank-anbindung bei der Integration von GA und CAFM die Verwendung eines reinen Computernetzwerks im Vordergrund. Bewertet werden Modelle ohne separaten Feldbus. Dieses Auswahlkriterium wird mit 3 Punkten gewichtet.

- **Webbasierte Steuerung**

Wichtigster Faktor ist die Möglichkeit der geräteunabhängigen und gewerke-übergreifenden Steuerung der Feldgeräte. Generell lässt sich eine Webfähigkeit aus technischer Sicht überall realisieren, doch meist nur durch Einzellösungen

und keiner Standards. Hier wird die standardmäßige Webfähigkeit bewertet. Dieses Auswahlkriterium wird mit 3 Punkten gewichtet.

- **Komfortable Geräteankopplung via UPnP**

Die innovative und neuartige Variante der Verwendung des UPnP-Standards im Bereich des Computer Aided Facility Managements und der Gebäudeautomation ist ein weiterer maßgeblicher Faktor der Bewertung. Dieses Auswahlkriterium wird mit 3 Punkten gewichtet.

- **Kabellose Einbindung via WLAN**

Ein weiteres Kriterium ist die kabellose Einbindung der Endgeräte in die CAFM-Umgebung via WLAN. Zur direkten Kopplung an den CAFM-Server können nicht immer und überall Kabel verlegt werden. Insbesondere eine Nachrüstung gestaltet sich meist schwierig. Kabelgebundene Lösungen können aber auch aufgrund lokaler Gegebenheiten von vornherein nicht realisierbar sein. Das macht WLAN erforderlich. Hinsichtlich UPnP ist WLAN neben Bluetooth die einzige anwendbare Funktechnologie. Dieses Auswahlkriterium wird mit 2 Punkten gewichtet.

- **Auflösen der redundanten Datenhaltung**

Da GA- und CAFM-Systeme nicht nur netzwerktechnisch sondern auch datentechnisch getrennte Systeme sind, müssen Anlagen- und Geräteattribute mehrfach erfasst und gepflegt werden. Eine positive Bewertung erfahren die Modelle, welche die doppelte Datenhaltung durch eine Integration auflösen können, sodass die Attribute in nur einer Datenbank redundanzfrei vorliegen. Dieses Auswahlkriterium wird mit 2 Punkten gewichtet

- **Sicherheit und Schutz vor Manipulation durch Hackerangriffe**

Ein nicht zu vernachlässigender Aspekt ist der Schutz vor unautorisiertem Zugriff auf Steuereinheiten. Die direkte Erreichbarkeit von Feldgeräten im WAN birgt trotz passwortgeschützten Logins Gefahren. Portfreigaben/-umleitungen des Routers ermöglichen zwar auch einen direkten Zugriff auf dahinter liegende (im LAN befindliche) Feldgeräte, bieten aber in Kombination mit einer Firewall einen erhöhten Schutz. Geräte des Feldbus sind bei der webfähigen Variante auf die gleiche Weise durch einen IP-Router geschützt. Über Router können auch VPN-Zugänge eingerichtet werden. Die Variante der lokalen Datenschnittstelle bietet eine kleinere Angriffsfläche für Hacker. Dieses Auswahlkriterium wird mit 2 Punkten gewichtet.

- **Wirtschaftlichkeit durch Integration von Management- und Automationsebene**
Dieser Punkt berücksichtigt nicht die Energieeffizienz der Geräte, sondern die Wirtschaftlichkeit des Gesamtsystems in Bezug auf Investitions- und Wartungskosten, die sich anhand der Anzahl an Geräten und Leitungskabel bemessen lassen. Durch die Zusammenführung der beiden Ebenen werden Automationsprozesse zentral verwaltet. Dadurch können separate Steuereinheiten (DDCs) entfallen, was die Infrastruktur schlanker und übersichtlicher macht. Dieser Wegfall wirkt sich dabei ebenso positiv aus wie die Einsparung von Busleitungen durch gemeinsame Nutzung der vorhandenen IT-Infrastruktur bzw. Planung einer solchen. Dieses Auswahlkriterium wird mit 2 Punkten gewichtet
- **Datenbankbasierte Historisierung**
Die Historisierung ist hinsichtlich der Betreiberverantwortung ein sehr wichtiges und wesentliches Funktionsmerkmal. Durch die Datenbankbindung der GLT kann die Aktivität eines Benutzers bzw. der Status eines Objekts in einem Datensatz mit Zeitstempel erfasst und archiviert werden, ähnlich der Funktionsweise einer „Black Box“. Dieses Auswahlkriterium wird mit 1 Punkt gewichtet.
- **Automatisierte Generierung von Wartungsgruppen**
Anhand automatisch übergebener XML-Files mit Attributen zur Gewerke-Kategorisierung der Anlagenkomponente, können ganze Wartungsgruppen automatisiert generiert werden. Dies ist ein großes Plus für das Anlagen- und Wartungsmanagement. Dieses Auswahlkriterium wird mit 1 Punkt gewichtet.
- **Einfache Struktur des Workflows**
Schließlich sei noch ein Aspekt berücksichtigt, der es Programmierern, Installateuren und Anwendern erleichtert, sich in den Abläufen der Kommunikation und Datenübermittlung zu orientieren und Prozesse besser darauf abstimmen zu können. Vorgreifend auf Kapitel 5.4 ist die Komplexität des Datenflusses schon anhand der grafisch dargestellten Modelle erkennbar und bewertbar. Während der Datenfluss im Falle der EnOcean-Funktechnologie vom Sensor über den Server via USB zum Gateway und die Signale von dort per Funk zum Aktor weitergeleitet werden, können im Falle der IP-basierten Variante diese (dann netzwerkfähigen) Komponenten direkt via LAN miteinander kommunizieren und eine Statusmeldung an den Datenbankserver weitergeben. Dieses Auswahlkriterium wird mit 1 Punkt gewichtet

5.2.2 Entscheidungsmatrix

Anhand der gewichteten Anforderungskriterien lässt sich die Bewertung zur Entscheidungsfindung in folgender Matrix abbilden.

	Reines Computernetzwerk +++	Webbasierte Steuerung +++	Komfortable Gerätean Kopplung via UPnP +++	Kabellose Anbindung via WLAN ++	Auflösen der redundanten Datenhaltung ++	Wirtschaftlichkeit durch Integration von Management- und Automationssebene ++	Sicherheit und Schutz vor unautorisiertem Zugriff ++	Datenbankbasierte Historisierung +	Automatisierte Generierung von Wartungsgruppen +	Einfache Struktur des Workflows +	Zeilenweise Summe
Kap. 5.1.2	-	-	-	-	-	-	x	-	-	-	10%
Kap. 5.1.3	-	x	-	-	-	-	x	-	-	-	25%
Kap. 5.1.4	x	-	-	-	x	x	x	-	-	x	50%
Kap. 5.1.5	x	x	x	x	x	x	x	x	x	x	100%
Kap. 5.1.6	x	x	-	-	x	x	-	x	-	x	60%
Gewichtung	15%	15%	15%	10%	10%	10%	10%	5%	5%	5%	100%

Tabelle 4: Entscheidungsmatrix

Die zuvor genannten Anforderungskriterien sind in der Matrix als zutreffend (x) oder nicht zutreffend (-) gekennzeichnet und werden entsprechend ihrer Gewichtung bewertet. In Summe können 20 Punkte erreicht werden. Dabei entspricht ein Bewertungspunkt 5 Prozentpunkten. Die Gewichtung ist in der letzten Zeile in Prozent angegeben (in Summe 100 %). Die letzte Spalte summiert die Prozentpunkte je Variante zeilenweise auf.

Zur Auswahl stehen die Varianten aus den Kapiteln 5.1.2 bis 5.1.6:

- Unidirektionale Kopplung des Feldbus über lokale Schnittstellenlösung
- Unidirektionale Kopplung des Feldbus über webbasierte IP-Controller
- Bidirektionale Kopplung des Feld-/Funkbus über lokalen USB/COM-Port
- Bidirektionale Kopplung netzwerkfähiger Komponenten über lokales LAN
- Bidirektionale Kopplung internetfähiger Komponenten über globales WAN

Die Auswertung zeigt ein eindeutiges Ergebnis. Mit der 100%igen Erfüllung der geforderten Kriterien ist die Variante der bidirektionalen Kopplung netzwerkfähiger Komponenten über das Local Area Network das favorisierte Lösungsmodell. Damit sind der grundlegende Aufbau und die Netzwerkstruktur als Voraussetzung für die in Kapitel 6 folgende technische Umsetzung gewählt bzw. festgesetzt.

5.2.3 Vor- und Nachteile des gewählten Modells

Trotz der 100%igen Erfüllung aller Anforderungskriterien ist das Modell nicht frei von Nachteilen und daher einer kritischen Betrachtung zu unterziehen. Vor- und Nachteile des Lösungsmodells werden gegenübergestellt.

Nachteile

- Hinsichtlich der „Intelligenz“ werden neue Ansprüche an die Endgeräte gestellt. Jeder Sensor und jeder Aktor muss netzwerkfähig sein und über den UPnP-Dienst verfügen. Das im Rahmen der Dissertation entwickelte Konzept kann in der Praxis nur softwareseitig demonstriert werden, da es eine entsprechende Hardware (UPnP-fähige Sensoren/Aktoren in der Gebäudeautomation) nach derzeitigem Kenntnisstand noch nicht gibt. Diese hardwareseitige Entwicklungsanforderung richtet sich an die Hersteller der Endgeräte mit Verweis auf die Nutzung bereits standardisierter Protokolle, wie auch in dieser Arbeit neu definierte Standards bzgl. der bereitzustellenden Daten.
- Im Vergleich zur energieeffizienten EnOcean-Technologie ist der Betrieb in dieser Modellvariante nicht batterieless möglich. Die Komponenten müssen permanent verfügbar sein. Dies kann nur durch eine dauerhafte Netzwerkverbindung und unterbrechungsfreie Stromversorgung sichergestellt werden.

- Der Betrieb über ein an das Internet angebundenes Computernetzwerk birgt auch Risiken bzgl. Hackerangriffen und Manipulation der Gebäudetechnik. Da Sensoren und Aktoren über das WAN erreichbar sind, werden hohe Sicherheitsanforderungen an das System gestellt. Durch Maßnahmen wie VPN-Zugang, Firewall, Portfreigaben-/umleitungen, Rechtevergabe und authentifizierten Login kann die Sicherheit gewährleistet werden (Auswahlkriterium), was aber einen erhöhten Aufwand bei der Installation eines Gerätes verursacht (Nachteil im Vergleich zu einer Feldbus-Lösung).

Vorteile

- + Das gewählte Modell bietet die besten Voraussetzungen einer innovativen CAFM-basierten Steuerung von gebäudetechnischen Steuereinheiten. Die Netzwerkstruktur und das webbasierte CAFM-System erfüllen zusammen mit den netzwerkfähigen Sensoren und Aktoren alle Anforderungen. Die Kommunikation erfolgt ausschließlich über das Computernetzwerk. Diese Eigenschaften bietet zwar auch das Lösungsmodell in Kapitel 5.1.6, doch die gewählte Variante hat einen entscheidenden Vorteil: Die Endgeräte werden nicht über das Internet, sondern über das lokale Netzwerk in die CAFM-Umgebung eingebunden. Damit lässt sich die Integration auch via WLAN mit automatischer Adressvergabe durch DHCP realisieren.
- + Durch die automatisierte Übermittlung von Anlagenattributen in Form von XML-Dateien stehen sämtliche Informationen zur Einordnung des Gewerkes (Kategorisierung z. B. nach VDMA oder AMEV) bereit. Damit lassen sich zur Unterstützung des Anlagen- und Wartungsmanagements automatisiert Wartungsgruppen generieren. Keine Komponente gerät in Vergessenheit, sofern sie einer Wartung unterliegt. Für eine korrekte Wartungsgruppenzuordnung können Fehler durch zusätzliche Abfragen beim Import oder Plausibilitätsprüfungen abgefangen werden.
- + Eine „Komfortanbindung“ via Universal Plug and Play lässt sich nur in lokalen Netzwerken realisieren. Daher ist das gewählte Modell die einzige Variante, die eine komfortable Geräteankopplung via UPnP ermöglicht.

UPnP in der Gebäudeautomation darf man als revolutionär bezeichnen. Es ist die Umsetzung einer Zukunftsvision, in der ein enormes und bislang ungenutztes Potenzial steckt.

5.3 Aufbau des Integrationsmodells und technische Umsetzung

In Abbildung 64 wird der Aufbau des gewählten Modells noch einmal gezeigt, um die Zusammenhänge der technischen Umsetzung in den folgenden Unterkapiteln anschaulich beschreiben zu können.

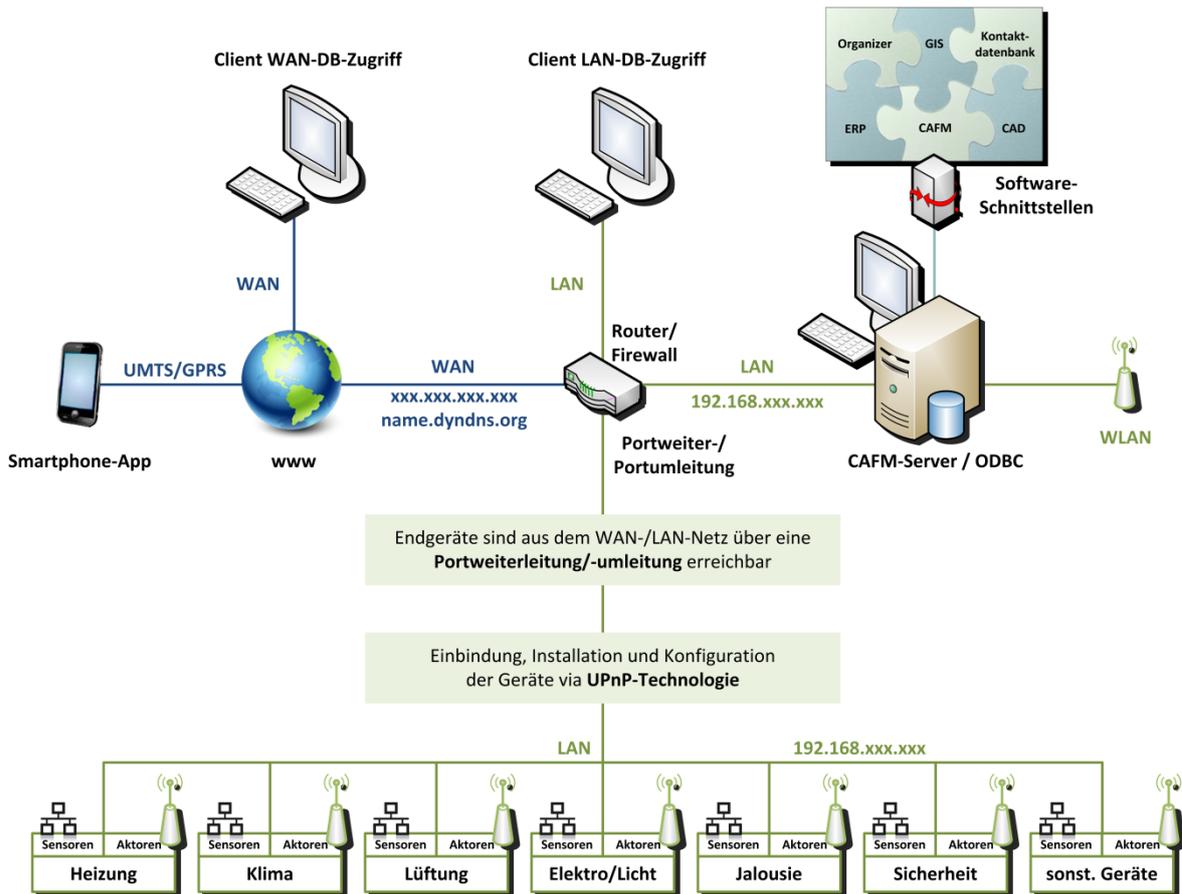


Abbildung 64: Aufbau des gewählten Integrationsmodells, eigene Darstellung

Die Prozesskette einer Kommunikation zwischen Client, Server und Endgeräten setzt nicht nur eine Vielzahl an Definitionen für Protokolle und Datentelegramme voraus, sondern auch die Zugriffs- und Weiterleitungsregeln an den Knotenpunkten des gesamten Netzwerks.

Grundsätzlich gibt es die Varianten, ein CAFM-System von einem lokalen Client über das LAN oder von einem entfernten Client über das WAN zu bedienen (siehe Kapitel 5.3.1). Ausgehend von der Situation, einen Steuerbefehl (z. B. zum Einschalten oder Regeln der Klimatisierung) von einem entfernten Client zu senden, lässt sich die Prozesskette wie folgt beschreiben:

Das CAFM-Programm wird über den Internetbrowser aufgerufen. Die Felder der Ein- und Ausgabemasken werden mit Informationen für die bzw. aus der Datenbank befüllt, was durch sog. SQL-Befehle oder -Abfragen geschieht. Für den Datenbankzugriff sendet der Client diese Kommandos an den CAFM-Server (gleichzeitig Web- und Datenbankserver), dessen Adresse bekannt sein muss. Sie setzt sich aus der IP-Adresse des Routers und einer Portnummer zusammen. Ist die IP-Adresse nicht statisch, sorgt ein Dienst (z. B. DynDNS) für einen gleichbleibenden String zur Erreichbarkeit des Servers (siehe Kapitel 5.3.2). Die erste Station, an der das Telegramm vor einer Schranke steht, ist der Router. Über den angegebenen Port wird die Anfrage weiter- oder umgeleitet, wie in Kapitel 5.3.3 behandelt. Am Server angekommen, werden die übermittelten Werte als Datenbankeintrag, Datenbankabfrage oder Steuerbefehl verarbeitet. Der aktuelle Wert des Feldes wird anschließend dem Client zurückgegeben.

Bei der erstmaligen Verwendung eines Endgerätes meldet sich der neue UPnP-fähige Teilnehmer automatisch im lokalen Netzwerk an und signalisiert seine Existenz. Das Verfahren UPnP wird in Kapitel 5.3.4 erläutert. Der Kontrollpunkt, in diesem Fall der CAFM-Server, erkennt die einzubindende Komponente und hat Zugriff auf die abrufbereiten XML-Dateien, die das Endgerät zur Verfügung stellt. Aufbau und Inhalt sowie Syntax und Semantik dieser Dateien können Kapitel 5.3.5 entnommen werden. Der Kopplungs-, Integrations- und Importvorgang kann beginnen.

5.3.1 Interner Zugriff vs. Fernzugriff

Interner Zugriff

Lokale Clients befinden sich alle im gleichen, lokalen Netzwerk (LAN) und sind über einen oder mehrere Switches (meist in Routern integriert) mit sämtlichen Netzwerkteilnehmern (Server und Endgeräte) gekoppelt. Datenbankzugriffe und über den Server koordinierte Steuerbefehle erfolgen also intern über das LAN-Netzwerk.

Ein für private bzw. lokale Netzwerke reservierter IP-Adressbereich liegt zwischen 192.168.000.000 bis 192.168.255.255. Damit lassen sich 256 Gruppen mit jeweils 256 Geräten bilden. In Summe also 65.536 Einheiten, die direkt über ihre lokale IP-Adresse angesprochen werden können.

Insgesamt gibt es drei spezielle IP-Adressbereiche, die für interne Netzwerke genutzt werden können⁸⁴:

- 010.000.000.000 - 010.255.255.255
- 172.016.000.000 - 172.031.255.255
- 192.168.000.000 - 192.168.255.255

Große Unternehmen benutzen oftmals auch andere Adressbereiche für interne Firmennetzwerke. Die Zuteilung von IP-Netzen im Internet wird von der Internet Assigned Numbers Authority (IANA) geregelt. Auf regionaler Ebene vergeben die Regional Internet Registries (RIR) die ihnen von der IANA zugeteilten Netze an lokale Vergabestellen. Für Deutschland ist das europäische Réseaux IP Européens Network Coordination Centre (RIPE NCC) zuständig.

Fernzugriff

Entfernte Clients oder auch Smartphones und Tablets sind über das World Wide Web mit dem CAFM-Server verbunden. Dabei ist der Router die Schnittstelle zwischen WAN und LAN und „besitzt“ zwei IP-Adressen – die des WANs (vergeben durch den Internet-Provider) und die des privaten LANs (vergeben durch den Netzwerk-Administrator).

Der entfernte Client kann bei einer Datenbankabfrage nur indirekt mit dem Server kommunizieren, da er lediglich die öffentliche WAN-IP-Adresse des Routers und den zum Server gehörenden Port als Empfänger angeben kann. Ist die IP-Adresse nicht statisch, sorgt ein dynamischer DNS-Dienst für einen statische Domain-Namen zur Erreichbarkeit des Servers (siehe Kapitel 5.3.2). Über den angegebenen Port wird die Anfrage weiter- oder umgeleitet, wie in Kapitel 5.3.3 behandelt. Ports sind „Eingangstüren“, die den Datenverkehr zu den Endgeräten filtern und regeln, abhängig vom jeweiligen Dienst.

⁸⁴ (Internet Engineering Task Force (IETF) - Network Working Group, RFC1918, 1996)

5.3.2 Dynamic Domain Name System (DynDNS) bei Fernzugriff

Für den Fall, dass die globale bzw. öffentliche IP-Adresse dynamisch vergeben wird und damit ständig wechselt, wie es bei einer DSL-Einwahl typischerweise geschieht, muss der Server dennoch unter einem konstanten Domainnamen erreichbar sein.

Einen solchen Dienst stellen sog. dynamische DNS-Provider zur Verfügung. Hierbei sendet der Router nach neuer Einwahl seine aktuelle IP-Adresse an den DynDNS-Provider, der den zuvor registrierten Domainnamen dann mit dieser Adresse verknüpft bzw. auf sie verlinkt. So kann der Server über die feste Adresse, z. B. mycafmserver.dyndns.org, erreichbar sein.

Die folgende Abbildung zeigt die Konfiguration, wie sie mit einem handelsüblichen DSL-Router (hier Telekom Speedport W 723V) vorgenommen werden kann:

Abbildung 65: Dynamic DNS, Screenshot aus eigener Speedport-Konfiguration

5.3.3 Portweiterleitung / Portumleitung bei Fernzugriff

Die Aufgabe des Routers ist es, die an einen bestimmten Port seiner IP-Adresse gesendeten Anfragen an die jeweiligen Endgeräte weiterzuleiten. Dazu bedarf es einer Portweiterleitung oder Portumleitung. Ports dienen der Sicherheit und müssen erst freigegeben werden, bevor eine Anfrage durchgestellt werden kann. Der Port ist abhängig vom jeweiligen Dienst (siehe Tabelle 5) und wird mit einem Doppelpunkt nach der IP-Adresse angegeben.

Bei einer reinen Weiterleitung sind der öffentliche Port und der interne Port identisch. Eine HTTP-Anfrage zur Darstellung einer Webseite (z. B. Geräte-URL) wird von Port 80 des Routers an Port 80 eines Endgerätes weitergeleitet. Port 80 ist der Standard für den HTTP-Dienst.

Konkret würde eine Anfrage, die an 123.145167 189:80 (öffentlich) adressiert ist, an 192.168.000.250:80 (intern) weitergeleitet werden.

Abbildung 66 zeigt die Konfiguration, wie sie mit dem DSL-Router „Speedport W 723V“ vorgenommen werden kann:

The screenshot displays the configuration page for a Speedport W 723V router, specifically the 'Netzwerk / NAT & Portregeln / Portregel' section. The interface includes a navigation menu on the left with options like 'Startseite', 'Assistent', 'Schritt für Schritt', 'Konfiguration', 'Sicherheit', 'Netzwerk', 'Telefonie', 'Status', 'Übersicht', 'Details', 'Verwaltung', 'Hilfsmittel', 'Laden & Sichern', and 'Beenden & Logout'. The main content area is titled 'Netzwerk / NAT & Portregeln / Portregel' and contains the following fields and sections:

- Vordefinierte Portregeln:** A dropdown menu labeled 'Auswahl:'.
- Regel-Definition:**
 - Bezeichnung: CAFM-Server
 - Art der Regel: Port Weiterleitung
- Betroffenes Gerät:**
 - Gültig für Gerät: 192.168.000.250
 - Port-Übersicht anzeigen button
- Weitergeleitete Ports / Portbereiche - Öffentlich & Private Client:**
 - TCP-Portbereich(e): 80
 - UDP-Portbereich(e): 80

At the bottom of the configuration area, there are three buttons: '<<<', 'Zurück <<', and 'Speichern <<'. On the right side, there is an 'Informationen' panel with the heading 'Portregel hinzufügen' and a 'Hinweis' section.

Abbildung 66: Portweiterleitung, Screenshot aus eigener Speedport-Konfiguration

Eine Portumleitung hingegen dient im Vergleich zur reinen Weiterleitung einer höheren Sicherheit und wird zur Sicherung eines Kanals für die Übertragung vertraulicher Daten verwendet. Dabei wird eine Anfrage an Port A des Routers auf Port B eines Endgeräts umgeleitet, wobei Port B der Standard-Portzuweisung entsprechen muss. Eine HTTP-Anfrage muss daher die Definition des entsprechenden öffentlichen Ports kennen, um an Port 80 des Endgeräts weitergeleitet werden zu können.

Konkret würde somit eine Anfrage an die öffentliche Adresse 123.145.167.189:4500 an die interne 192.168.000.250:80 weitergeleitet werden.

Mit dem DSL-Router „Speedport W 723V“ lässt sich diese Portumleitung wie folgt konfigurieren:

The screenshot shows the configuration page for a Speedport W 723V router. The main navigation menu on the left includes: Startseite, Assistent, Schritt für Schritt, Konfiguration, Sicherheit, Netzwerk (highlighted), Telefonie, Status, Übersicht, Details, Verwaltung, Hilfsmittel, Laden & Sichern, and Beenden & Logout. The main content area is titled 'Netzwerk / NAT & Portregeln / Portregel'. It features a 'Vordefinierte Portregeln' section with a dropdown menu labeled 'Auswahl:'. Below this is the 'Regel-Definition' section with the following fields: 'Bezeichnung:' (CAFM-Server), 'Art der Regel:' (Port Umleitung), and 'Betroffenes Gerät' (Gültig für Gerät: 192.168.000.250). A 'Port-Übersicht anzeigen' button is located below the device field. The 'Umgeleiteter Port - Öffentlich auf Private Client' section contains two rows: 'TCP-Port: 4500' mapped to 'umgeleitet auf Port: 80' and 'UDP-Port:' mapped to 'umgeleitet auf Port:'. At the bottom of the configuration area are three buttons: '<< <<', 'Zurück <<', and 'Speichern <<'. On the right side, there is an 'Informationen' panel with the heading 'Portregel hinzufügen' and a 'Hinweis' section.

Informationen

Portregel hinzufügen

Hier können Sie beliebige Regeln für die Port Weiterleitung, Umleitung oder Öffnung festlegen. Vordefinierte Regeln werden Ihnen unter 'Auswahl' angeboten.

Hinweis

Bitte beachten Sie, dass mit dem Definieren einer Regel die eingetragenen Ports geöffnet werden und so nicht mehr durch die interne Firewall Ihres Gerätes gegen Zugriffen aus dem Internet geblockt werden.

Abbildung 67: Portumleitung, Screenshot aus eigener Speedport-Konfiguration

Bei dynamischen WAN-IP-Adressen kommt anstelle der 12-stelligen Nummer der über DynDNS gewählte Domainname zum Einsatz, sodass eine entsprechende Anfrage beispielsweise an mycafmserver.dyndns.org:4500 gesendet werden würde.

Diese Tabelle zeigt eine Auswahl von häufig verwendeten Standard-Ports mit den zugehörigen Diensten:

Port	Dienst
21	FTP-Server
22	Secure Shell (SSH)
23	Telnet-Server
25	Mail-Server (SMTP)
53	DNS
69	Trivial FTP
80	Web-Server
113	identd (Auth)
135	MS-RPC
137	NetBIOS Name Service
138	NetBIOS Datagram Service
139	NetBIOS Session Service
443	Web Server (HTTPS)
445	SMB over TCP
515	Druck-Server (LPD)
631	Druck-Server (IPP/Cups)
1214	Kazaa Standard-Port
1433	MS SQL Server
1701	VPN-Server (L2TP)
1723	VPN-Server (PPTP)
1900	Universal PnP
3389	MS Terminal Services
4662	Standard-Port eDonkey
5000	Universal PnP
5800	VNC via HTTP
5900	VNC
6667	IRC Server
6881	Bittorrent Standard-Port
8080	HTTP Proxy
9100	RAW JetDirect

Tabelle 5: Liste ausgewählter Standard-Ports

Hier wird noch einmal der Sicherheitsaspekt einer Portumleitung deutlich: Portscanner können zwar offene Ports identifizieren, wissen aber nicht, welcher Dienst sich dahinter verbirgt. Die Standardports können geschlossen bleiben.

5.3.4 Geräteankopplung via UPnP (Universal Plug and Play)

Bis Mitte der 1990er Jahre war das Installieren von Peripheriegeräten, wie Maus, Drucker, Scanner usw. noch eine recht umständliche und aufwändige Angelegenheit. Beim Einstecken der Verbindungskabel wurden die Geräte an den parallelen und seriellen Bus-Schnittstellen nicht automatisch erkannt. Zudem waren spezielle Treiber nötig, damit sie im Betriebssystem als Hardware erkannt werden und für den Betrieb zur Verfügung stehen konnten. Man strebte eine Lösung an, mit der diese Geräte unmittelbar nach dem Anschließen an einen PC betriebsbereit und funktionsfähig waren. Einstecken und loslegen - „Plug and Play“. Die Lösung nannte sich Universal Serial Bus, kurz USB. Seit 1996 hat sich USB zu einer Standardschnittstelle in der Computertechnik entwickelt.

In der Gebäudeautomation ist USB als Feldbusersatz nur für Systeme mit geringer Ausdehnung oder Funkbus-Gateways geeignet. Das Ziel ist es aber, eine netzwerkfähige und webbasierte Lösung zu finden. Doch gibt es die „Plug and Play“-Technologie auch auf Netzwerkebene und wenn ja, welche Vorteile hätte sie für die Gebäudeautomation?

Der Schlüssel heißt „Universal Plug and Play“, kurz UPnP. Im Grunde gleicht es der lokalen USB-Variante, ist aber netzwerkbasiert. Man unterscheidet zwischen Geräten (Devices) und Kontrollpunkten (Control Points). Im Netz neu hinzugefügte Geräte geben sich durch einen Multicast (Datentelegramm an eine Empfängergruppe) zu erkennen, wodurch sie von den Kontrollpunkten, z. B. ein Windows-Rechner, registriert und lokalisiert werden können. Anschließend kann die Gerätebeschreibung ausgelesen werden.

Eine weitere Erkenntnis bei den Recherchen war, dass UPnP hinsichtlich der Transportmedien nicht nur über das kabelgebundene LAN, sondern auch über das kabellose WLAN realisierbar ist. Das bietet eine hervorragende Basis für gebäudetechnische Automationskonzepte, insbesondere bei einer Nachrüstung. Zudem sind mit UPnP herstellerspezifische Erweiterungen möglich, unabhängig vom Betriebssystem oder der verwendeten Programmiersprache.

Um die UPnP-Technologie für die Gebäudeautomation einsetzen zu können, sind Kenntnisse über die Funktionsweise aus informationstechnischer Sicht sowie über die zum UPnP-Standard gehörenden Datenformate und Netzwerkprotokolle erforderlich.

UPnP-Standards werden durch das UPnP-Forum, das sich aus einer Initiative der Computerindustrie heraus entwickelt hat, definiert. In der „UPnP Device Architecture“ sind die XML-basierte Gerätedarstellung und Protokolle für die Kommunikation zwischen Control Points und Devices spezifiziert.

Zur Erläuterung der Funktionsweise zeigt folgende Abbildung eine Übersicht der aus 6 Schritten bestehenden Prozesskette einer UPnP-Geräteanbindung:

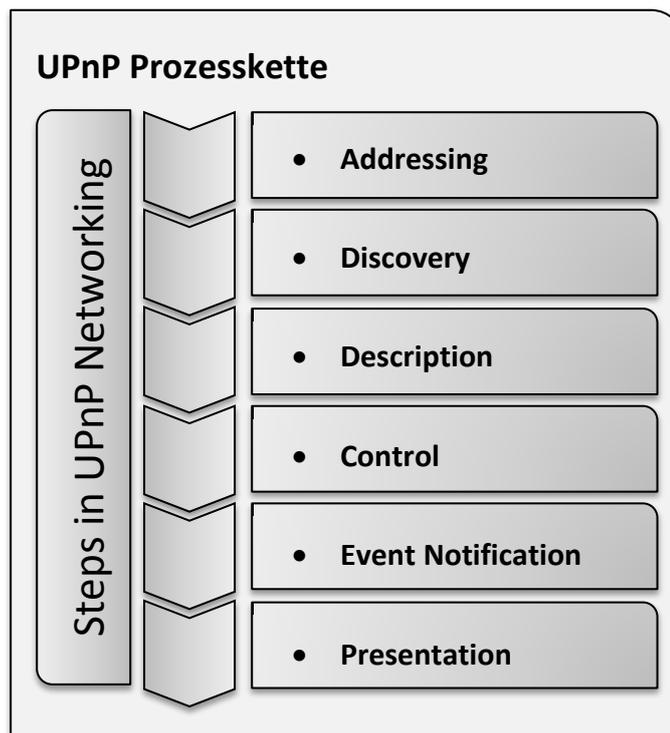


Abbildung 68: UPnP-Prozesskette, eigene Darstellung

Die einzelnen Schritte der Geräteanbindung sind in der „UPnP Device Architecture“⁸⁵ und auf den Seiten der Microsoft Informationsplattform TechNet⁸⁶ nach folgendem Schema beschrieben:

1. Adressierung

Das UPnP-Gerät bezieht zu Beginn eine IP-Adresse zwecks eindeutiger Identifizierung im lokalen Netzwerk. Standardmäßig wird das Gerät versuchen, eine IP-Adresse über DHCP zu erhalten. Die Adressierung kann aber auch durch manuelle Adressvergabe erfolgen.

⁸⁵ (UPnP Forum, UPnP Device Architecture 1.1, 2008)

⁸⁶ (Microsoft TechNet, Universal Plug and Play (UPnP) Client Support, 2013)

2. Lokalisierung

Sobald das Gerät über eine IP-Adresse verfügt, meldet es seine Existenz im Netzwerk auf Basis des Simple Service Discovery Protocol (SSDP). Es sendet eine Nachricht mit Informationen über sich selbst (wie z. B. Name, Typ und URL) an die Multicast-Adresse 239.255.255.250, UDP-Port 1900, was von den Kontrollpunkten registriert wird. Diese Nachricht enthält eine Kopfzeile, ähnlich einer HTTP-Anfrage. Häufig findet man für dieses Protokoll auch die Bezeichnung HTTPU, abgeleitet aus HTTP über UDP.

3. Beschreibung

Jedes Gerät bietet eine Beschreibung seiner Eigenschaften und Dienstleistungen an und stellt diese in einem XML-Dokument zur Verfügung. Die Kontrollpunkte können nun die detaillierte Gerätebeschreibung per HTTP auslesen. Die für den Abruf erforderliche Adresse (URL) wurde bei der Lokalisierung mitgeteilt. Die XML-Datei enthält sämtliche Attribute, wie z. B. Hersteller, Seriennummer, und weitere URLs zur Steuerung, Ereignismeldung und Präsentation. Die Datei wird üblicherweise device.xml benannt. Mit der Angabe von Kommandos und Aktionen sowie Datentypen und Datenbereichen stellt das Gerät neben seinen Eigenschaften auch die Informationen seiner Dienste zur Verfügung. Die Datei wird üblicherweise scpd.xml benannt.

4. Steuerung

Mit den aus der XML-Datei ausgelesenen Kommandos ist der Kontrollpunkt in der Lage, Aktionen auf dem UPnP-Gerät auszulösen und es damit zu steuern. Dazu sendet er eine SOAP-Anforderung an die Steuerungs-URL. SOAP (Simple Object Access Protocol) ist ein Protokoll, das über HTTP läuft und XML verwendet.

5. Ereignismeldungen

Zur Steuerung eines Gerätes, müssen nicht nur die in Layer 3 übermittelten Steuerkommandos bekannt sein, sondern auch die aktuellen Werte der jeweiligen Zustands- bzw. Statusvariablen. Um diese aber nicht permanent abfragen zu müssen, bedient man sich der XML-basierten General Event Notification Architecture (GENA). Damit können die Kontrollpunkte Zustandsgrößen eines Dienstes abonnieren und erhalten im Falle einer Änderung der Statusvariablen eine sog. „Eventmessage“.

6. Präsentation

Mit Hilfe der Präsentationsschicht lässt sich der Zugriff auf Geräte etwas komfortabler und anschaulicher gestalten. Es ist eine Alternative zur Steuerung via SOAP-Mitteilungen und den Ereignismeldungen. Während SOAP die Kommandos an die „Control URL“ sendet, kann der Benutzer auch über die „Presentation URL“ auf das Gerät zugreifen. Hinter dieser URL verbirgt sich eine HTML-basierte Benutzeroberfläche, mit der (meist) die gleichen Funktionen zur Verfügung stehen, wie sie auch durch den standardisierten UPnP-Zugriff über SOAP ausgelöst werden.

Wie in den meisten Fällen der Kommunikationsprotokollstruktur (vgl. LonTalk Kapitel 4.1.2, BACnet Kapitel 4.1.4, EnOcean Kapitel 4.1.6 und TCP/IP Kapitel 4.2.4), orientiert sich auch das UPnP Networking an den Schichten des ISO/OSI-Referenzmodells. Die folgende Abbildung zeigt den Aufbau mit Darstellung der Dienste in der Anwendungsschicht unter Zuordnung der genannten Schritte 2-6 des UPnP Networkings. Schritt 1 (die Adressierung) ist im hier nicht dargestellten Layer 1 (der Netzwerkzugangsschicht) implementiert.

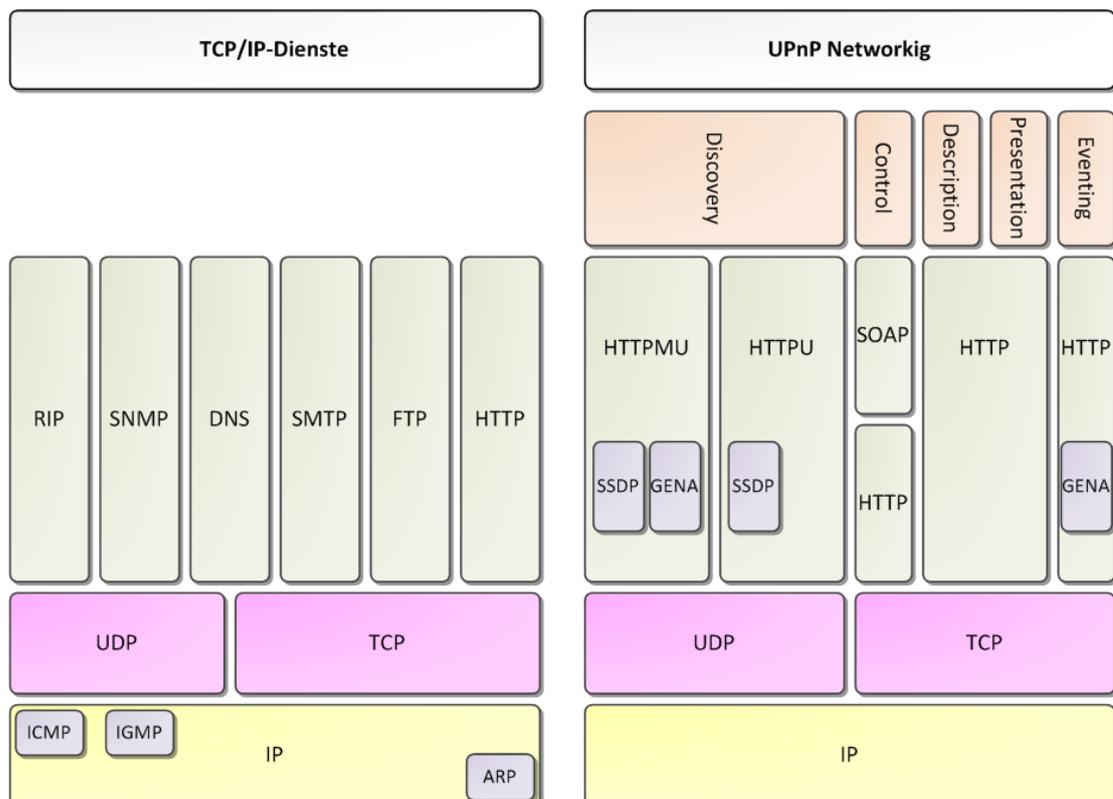


Abbildung 69: Schichten des UPnP-Netzwerkings im Vergleich zu TCP/IP-Diensten, eigene Darstellung

Die zur Lokalisierung, Steuerung und Ereignismeldung notwendigen Protokolle SSDP, SOAP und GENA sind noch einmal im Detail zu betrachten:

- **SSDP (Simple Service Discovery Protocol)**

Dieses Protokoll legt fest, wie Geräte im UPnP-Netzwerk entdeckt werden können. Dafür definiert SSDP Methoden, mit denen die Verfügbarkeit von Geräten im Netzwerk bekannt gegeben wird. Kontrollpunkten wird es so ermöglicht, diese Geräte und deren Services durch das Versenden von SSDP-Anfragen im Netzwerk zu finden. Das Protokoll erlaubt das Suchen nach allen verfügbaren oder auch nach bestimmten einzelnen Geräten. Die Kontrollpunkte überwachen den Multicast-Port 1900 des Netzwerks und reagieren auf passende Anfragen. Sowohl die Anwesenheitsbenachrichtigung wie auch die SSDP-Antwort (Device Response Message) beinhalten einen Verweis auf die Gerätebeschreibung (Device Description).⁸⁷

- **SOAP (Simple Object Access Protocol)**

Dieses Protokoll wird in UPnP-Netzwerken zur Übergabe von Steuerbefehlen, sog. Control Messages an den Geräteservice gebraucht. Es ist sozusagen das „UPnP Control Interface“. Control Messages sind UPnP-Anfragen, welche sog. „Action Requests“ inkl. der dazugehörigen Übergabeparameter enthalten. Antworten oder Fehlermeldungen des Service an den Kontrollpunkt sind ebenfalls SOAP Nachrichten. Sie enthalten den neuen Status des Services, den Rückgabewert sowie den Rückgabeparameter der Aktion. SOAP definiert den Gebrauch von XML und HTTP, um Remote Procedure Calls (RPC) auszuführen. SOAP verwendet die existierende Infrastruktur des Internets und kann daher effektiv mit Firewalls und Proxys arbeiten. Es verwendet die existierenden Einrichtungen des HTTP Connection Managements, wodurch verteilte Kommunikation über das Internet so einfach wie der Zugriff auf Web-Seiten wird.⁸⁷

- **GENA (Generic Event Notification Architecture)**

Dieses Protokoll wird verwendet, um HTTP Mitteilungen über TCP/IP und UDP an Multicast Port 1900 zu senden und zu empfangen. Zudem definiert GENA auch das Konzept der Ereignis-Abonnements. Die „Eventnotifications“ werden mit Hilfe von GENA erzeugt, ebenso die Anwesenheitsankündigungen, die dann mittels SSDP über das Netz gesendet werden.⁸⁷

⁸⁷ (Microsoft Corporation, White Paper - Understanding Universal Plug and Play, 2000)

Durch den Einsatz der UPnP-Technologie in der Gebäudeautomation ergeben sich völlig neue Perspektiven und Chancen im Bereich der CAFM-Entwicklung. Doch an dieser Stelle sei auf eine aktuelle Studie zum Thema UPnP-Sicherheit hingewiesen, nach der erhebliche Schwachstellen in den UPnP-Protokollen entdeckt worden sind. Diese wurden im Februar 2013 in einem Whitepaper von Rapid7, einem Entwickler von Open-Source-Projekten zur Computersicherheit, veröffentlicht. Das Unternehmen warnt vor Hackerangriffen auf UPnP-fähige Geräte. Durch die Sicherheitslücken können UPnP-Dienste zum Absturz gebracht, schädlicher Code eingeschleust und damit Geräte des lokalen Netzwerks manipuliert werden. Es besteht sogar die Gefahr des unbemerkten Öffnens von Ports UPnP-fähiger Router. Insgesamt sind in gleich drei Protokollen Schwachstellen entdeckt worden: dem Simple Service Discovery Protocol (SSDP), dem Simple Object Access Protocol (SOAP) und dem Hypertext Transfer Protocol (http).⁸⁸

Diese grundlegenden Probleme sind durch das UPnP-Forum zu beheben, unabhängig vom Einsatzbereich dieser Technologie. Hinsichtlich der Nutzung und Betreiberverantwortung für die Gebäudeautomation müssen diese Lücken zwingend geschlossen werden, bevor UPnP im Bereich der Gebäudeautomation bzw. CAFM-Integration in Betrieb genommen werden kann. Die Sicherheit muss gewährleistet sein.

Anschließend kann damit begonnen werden, die anwendungsspezifischen Maßgaben im UPnP Networking zu berücksichtigen und bereitzustellende Geräteeigenschaften und -dienste in den entsprechenden XML-Dateien (device.xml und scpd.xml) anzupassen. Der standardmäßigen Anbindung soll - auf Basis der bereitgestellten Dienste - die CAFM-Integration folgen. Die Funktionen der Steuereinheiten sind damit in der Datenbank festgehalten und können mit der Programmierung von logischen Prozessfolgen und Szenarien zu intelligenten CAFM-Steuerungsroutrinen ausgebaut werden.

⁸⁸ (Rapid7, Whitepaper - Security Flaws in Universal Plug and Play, 2013)

5.3.5 XML (Extensible Markup Language)

Die Funktionsweise von UPnP setzt die Bereitstellung von XML-Dateien zur Beschreibung der Geräte und deren Dienste voraus. Diese werden in Step 3 des UPnP Networkings (Description) per HTTP abgerufen, wie Kapitel 5.3.4 zu entnehmen ist. XML-Dateien sind im Grunde einfache Textdateien mit einer speziellen Formatierung.

Der Entwurf eines innovativen Netzwerkmodells zur Einbindung gebäudetechnischer Anlagen in CAFM-Systeme als Kern dieser Arbeit beinhaltet auch die Definition von Standards. Dazu gehört die Bestimmung von Aufbau und Inhalt dieser XML-Dateien für Geräte der Gebäudeautomation.

Mit Hilfe der OpenSource-Software „UPnP Developer Tools“ lässt sich ein virtuelles Gerät der Gebäudeautomation als Musterobjekt erzeugen. Es simuliert das Vorhandensein einer echten Steuereinheit und dient als Vorlage für die Umsetzung der CAFM-Integration. Dazu ist die vorherige Erzeugung der XML-Dateien erforderlich, deren allgemeiner Aufbau in diesem Kapitel verdeutlicht wird.

Doch zunächst seien die Bedeutung, die grundlegende Struktur und die standardmäßige Deklaration einer XML-Datei betrachtet. Der seit Februar 1998 definierte XML-Standard wird vom World Wide Web Consortium (W3C) spezifiziert. XML steht für Extensible Markup Language und bedeutet übersetzt „erweiterbare Auszeichnungssprache“. Mit ihr lassen sich Daten in hierarchisch strukturierter Form darstellen.

Die Struktur einer XML-Datei ist einfach gehalten. Sie besteht aus einem Header (dem Deklarationskopf) und einem sog. Wurzelement, welches mehrere untergeordnete Einzelemente beinhalten kann. Diese Einzelemente können mit einer ID gekennzeichnet werden, um gleichartige Einträge unterscheiden zu können.

Als einführendes Beispiel sei ein einfaches XML-Dokument gewählt, das personenbezogene Informationen bereithält (später werden es gerätebezogene Informationen sein).

Jede XML-Datei beginnt mit einem Header, der aus der Deklaration von Datei, Dokumenttyp (Document Type Definitions) und Formatvorlage (Style Sheets) besteht. Er teilt den Anwendungen mit, wie die Datei verarbeitet werden soll. Dazu wird angegeben, dass es sich um eine XML Datei handelt mit Angabe der

Version. Des Weiteren wird mit dem Schlüsselwort „encoding“ der Zeichensatz, z. B. ISO-8859-1 oder UTF-8, und mit „standalone“ das Vorhandensein optionaler Dokumenttypdefinitionen angegeben.⁸⁹ Für die UPnP-Funktionalität kann auf die Document Type Definitions und Style Sheets verzichtet werden, da formatierte Textdarstellungen von Geräteeigenschaften und –diensten nicht nötig sind. Damit würde die erste Zeile wie folgt aussehen:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

Sollten dennoch Document Type Definitions (DTD) erforderlich sein, können sie über das Schlüsselwort „!DOCTYPE“ im XML-Dokument selbst oder durch Verweis auf eine zusätzliche Datei definiert werden. Bei der internen Deklaration muss der Name der DTD identisch zum Namen des Wurzelements sein.⁸⁹ Die entsprechende Zeile wäre:

```
<!DOCTYPE Name [DTD]> oder <!DOCTYPE Name SYSTEM "Name.dtd">
```

Der Vollständigkeit halber sei noch die Deklaration eines Style Sheets behandelt. Die Textdarstellung (Schriftstil) kann mit Hilfe des Cascading Style Sheets (CSS) formatiert werden, was in einer separaten XSL-Datei gespeichert ist. Auf diese Datei wird mit dem Schlüsselwort href verwiesen unter Angabe des Dateityps.⁸⁹ Die Deklaration würde lauten:

```
<?xml-stylesheet href="/style.xsl" type="text/xsl"?>
```

Nachdem alle Deklarationen getroffen sind und der Header damit abgeschlossen ist, folgt der eigentliche Inhalt, der einem hierarchischen Datenspeicher von Namen und Werten gleich kommt. Die Struktur gliedert sich in die drei Ebenen Wurzelement, Einzelemente und Unterelemente. Eine XML-Datei hat nur ein einziges Wurzelement, für das gewählte Beispiel wäre es der Sammler „Mitarbeiterstammdaten“. Die Einzelemente sind mit einzelnen Personen gleichzusetzen und seien als „Mitarbeiter“ bezeichnet. Eine ID kennzeichnet die verschiedenen Mitarbeiter, um sie bei gleichem Namen unterscheiden zu können. Diese ID sei die Personal-Nummer. In der dritten Ebene folgen die personenbezogenen Einträge, wie z. B. Name, Vorname, Adresse usw., mit entsprechenden Werten.

Die Struktur steht damit fest, es folgt die Umsetzung.

⁸⁹ (Bart, XML-Deklaration, 2013)

Für den fehlerfreien Aufbau eines XML-Dokuments ist die definierte Syntax strikt einzuhalten. Sämtliche Einträge, auch engl. „Tags“ genannt, sind mit einem Namen in Spitzklammern anzugeben. Wichtig dabei ist, auch das Ende eines Tags zu definieren. Dazu gibt man den gleichen Namen mit vorangestelltem Schrägstrich in Spitzklammern an. Unterelemente befinden sich jeweils zwischen Start- und End-Tag.

Im gewählten Beispiel würde sich der Code unter Verzicht von Document Type Definition und Style Sheet folgendermaßen darstellen:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Mitarbeiterstammdaten>
  <Mitarbeiter ID="0001">
    <Vorname>Michael</Vorname>
    <Nachname>Müller</Nachname>
    <Strasse>Parkstraße 5</Strasse>
    <PLZ>12345</PLZ>
    <Ort>Großstadt</Ort>
    <Telefon>01234/56789</Telefon>
    <Geburtstag>01.06.1960</Geburtstag>
    <Geschlecht>männlich</Geschlecht>
    <Eintrittsdatum>01.03.2005</Eintrittsdatum>
    <Position>Leiter Projektentwicklung</Position>
    <Gehaltsstufe>8</Gehaltsstufe>
  </Mitarbeiter>
  <Mitarbeiter ID="0002">
    <Vorname>Andrea</Vorname>
    <Nachname>Maier</Nachname>
    <Strasse>Schlossallee 1</Strasse>
    <PLZ>67890</PLZ>
    <Ort>Reichenstein</Ort>
    <Telefon>09876/54321</Telefon>
    <Geburtstag>15.10.1975</Geburtstag>
    <Geschlecht>weiblich</Geschlecht>
    <Eintrittsdatum>01.07.2008</Eintrittsdatum>
    <Position>Assistenz der Geschäftsführung</Position>
    <Gehaltsstufe>5</Gehaltsstufe>
  </Mitarbeiter>
</Mitarbeiterstammdaten>
```

Die für den UPnP-Betrieb bereitzustellenden XML-Formate zur Geräte- und Servicebeschreibung hat das UPnP-Forum in seiner „Device Architecture 1.1“ definiert.

Die „Device Description“ beinhaltet die Versionsangabe, allgemeine Geräteinformationen, die UUID, Angaben zu Icons und eine Liste der bereitgestellten Services mit Verweis auf die zugehörige „Service Description“.

UPnP-Template für XML Device Description⁹⁰

```
<?xml version="1.0"?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <URLBase>base URL for all relative URLs</URLBase>
  <device>
    <deviceType>urn:schemas-upnp-org:device:deviceType:v</deviceType>
    <friendlyName>short user-friendly title</friendlyName>
    <manufacturer>manufacturer name</manufacturer>
    <manufacturerURL>URL to manufacturer site</manufacturerURL>
    <modelDescription>long user-friendly title</modelDescription>
    <modelName>model name</modelName>
    <modelName>model number</modelName>
    <modelURL>URL to model site</modelURL>
    <serialNumber>manufacturer's serial number</serialNumber>
    <UDN>uuid:UUID</UDN>
    <UPC>Universal Product Code</UPC>
    <iconList>
      <icon>
        <mimetype>image/format</mimetype>
        <width>horizontal pixels</width>
        <height>vertical pixels</height>
        <depth>color depth</depth>
        <url>URL to icon</url>
      </icon>
      XML to declare other icons, if any, go here
    </iconList>
  </device>
</root>
```

⁹⁰ (UPnP Forum, UPnP Device Architecture 1.1, 2008)

```
<serviceList>
  <service>
    <serviceType>urn:schemas-upnp-
      org:service:serviceType:v</serviceType>
    <serviceId>urn:upnp-org:serviceId:serviceID</serviceId>
    <SCPDURL>URL to service description</SCPDURL>
    <controlURL>URL for control</controlURL>
    <eventSubURL>URL for eventing</eventSubURL>
  </service>
  Declarations for other services defined by a UPnP Forum working
  committee (if any) go here
  Declarations for other services added by UPnP vendor (if any) go
  here
</serviceList>
<deviceList>
  Description of embedded devices defined by a UPnP Forum working
  committee (if any) go here
  Description of embedded devices added by UPnP vendor (if any) go
  here
</deviceList>
<presentationURL>URL for presentation</presentationURL>
</device>
</root>
```

Die Service Description deklariert die von dem Dienst bereitgestellten Aktionen (<action>). Jede Aktion beinhaltet Parameter oder Argumente (<argument>) sowie eine Liste von Variablen (<stateVariable>), die den Zustand des Geräts während der Laufzeit beschreiben. Zusammenfassend lässt es sich wie folgt beschreiben:

- Action = Funktionsname
- Argument = Steuerbefehl/Kommando
- Variable = Status/Zustandswert

UPnP-Template für XML Service Description⁹¹

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
```

⁹¹ (UPnP Forum, UPnP Device Architecture 1.1, 2008)

```

<actionList>
  <action>
    <name>ActionName1</name>
    <argumentList>
      <argument>
        <name>ArgumentNameIn1</name>
        <direction>in</direction>
        <retval/>
        <relatedStateVariable>
          StateVariableName1
        </relatedStateVariable>
      </argument>
      <argument>
        <!-- COMMENT Declarations for other in arguments defined by UPnP
        Forum working committee (if any) go here. -->
        <!-- COMMENT If the action acts as a function, returning a
        value, the <retval/> empty element indicates that this argument
        is actually the function return value. Only the first out
        argument may be tagged as the function return value. This type
        of argument is optional. -->
      </argument>
      <argument>
        <name>ArgumentNameOut1</name>
        <direction>out</direction>
        <retval/>
        <relatedStateVariable>
          StateVariableName2
        </relatedStateVariable>
      </argument>
      <!-- /COMMENT -->
      <argument>
        <name>ArgumentNameOut2</name>
        <direction>out</direction>
        <retval/>
        <relatedStateVariable>
          StateVariableName3
        </relatedStateVariable>
      </argument>
      <!-- COMMENT Declarations for other out arguments defined by
      UPnP Forum working committee (if any) go here. -->
    </argumentList>
  </action>

```

```
<!-- COMMENT Declarations for other actions defined by UPnP Forum
working committee (if any) go here. -->
<!-- COMMENT Declarations for other actions defined by UPnP vendor
(if any)go here. -->
</actionList>
<serviceStateTable>
  <stateVariable sendEvents="yes">
    <name>StateVariableName1</name>
    <dataType>string</dataType>
    <defaultValue>Value1</defaultValue>
    <allowedValueList>
      <allowedValue>Value1</allowedValue>
      <allowedValue>Value2</allowedValue>
      <allowedValue>Value3</allowedValue>
    </allowedValueList>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>StateVariableName2</name>
    <dataType>ui4</dataType>
    <defaultValue>default value</defaultValue>
    <allowedValueRange>
      <minimum>MinimumValue</minimum>
      <maximum>MaximumValue</maximum>
      <step>Increment</step>
    </allowedValueRange>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>A_ARG_TYPE_StateVariableName3</name>
    <dataType>string</dataType>
    <defaultValue>default value</defaultValue>
    <allowedValueRange>
      <minimum>MinimumValue</minimum>
      <maximum>MaximumValue</maximum>
      <step>Increment</step>
    </allowedValueRange>
  </stateVariable>
  <!-- COMMENT Declarations for other state variables defined by UPnP
Forum working committee (if any) go here. -->
  <!-- COMMENT Declarations for other state variables defined by UPnP
vendor (if any)go here. -->
</serviceStateTable>
</scpd>
```

Das UPnP-Forum hat bereits eine Vielzahl an Gerätekategorien standardisiert und folgende Templates, sog. Device Control Protocols (DCP), veröffentlicht⁹²:

- Audio/Video
 - MediaServer:1 and MediaRenderer:1
 - MediaServer:2 and MediaRenderer:2
 - MediaServer:3
 - MediaServer:4 and MediaRenderer:3
- Basic
 - Basic Device:1
- Device Management
 - Manageable Device:1
 - Manageable Device:2
- Home Automation
 - Digital Security Camera:1
 - HVAC:1
 - Lighting Controls:1
 - SolarProtectionBlind:1
- Networking
 - Internet Gateway:1
 - Internet Gateway:2
 - WLAN Access Point:1
- Printer
 - Printer Basic:1
 - Printer Enhanced:1
- Remote Access
 - RAClient:1, RAServer:1 and RADiscoveryAgent:1
 - RAServer:2 and RADiscoveryAgent:2
- Remoting
 - Remote UI Client:1 and Remote UI Server:1
- Scanner
 - Scanner:1
- Telephony
 - Telephony:1
 - Telephony:2

⁹² (UPnP Forum, UPnP Device Control Protocols and Device Categories, 2013)

- Add-on Services
 - BasicManagementService:1
 - BasicManagementService:2
 - ContentSync:1
 - ConfigurationManagementService:1
 - ConfigurationManagementService:2
 - DeviceProtection:1
 - Device Security:1 and Security Console:1
 - Low Power:1
 - Quality of Service:1
 - Quality of Service:2
 - Quality of Service:3
 - SoftwareManagementService:1
 - SoftwareManagementService:2

5.4 Workflow-Modelle

Aus den in Kapitel 5.3 aufgezeigten Funktionen zur technischen Umsetzung lassen sich nun die Workflow-Modelle für die erstmalige Einbindung eines UPnP-Geräts sowie dessen clientseitige Steuerung mit folgenden Flussdiagrammen darstellen:

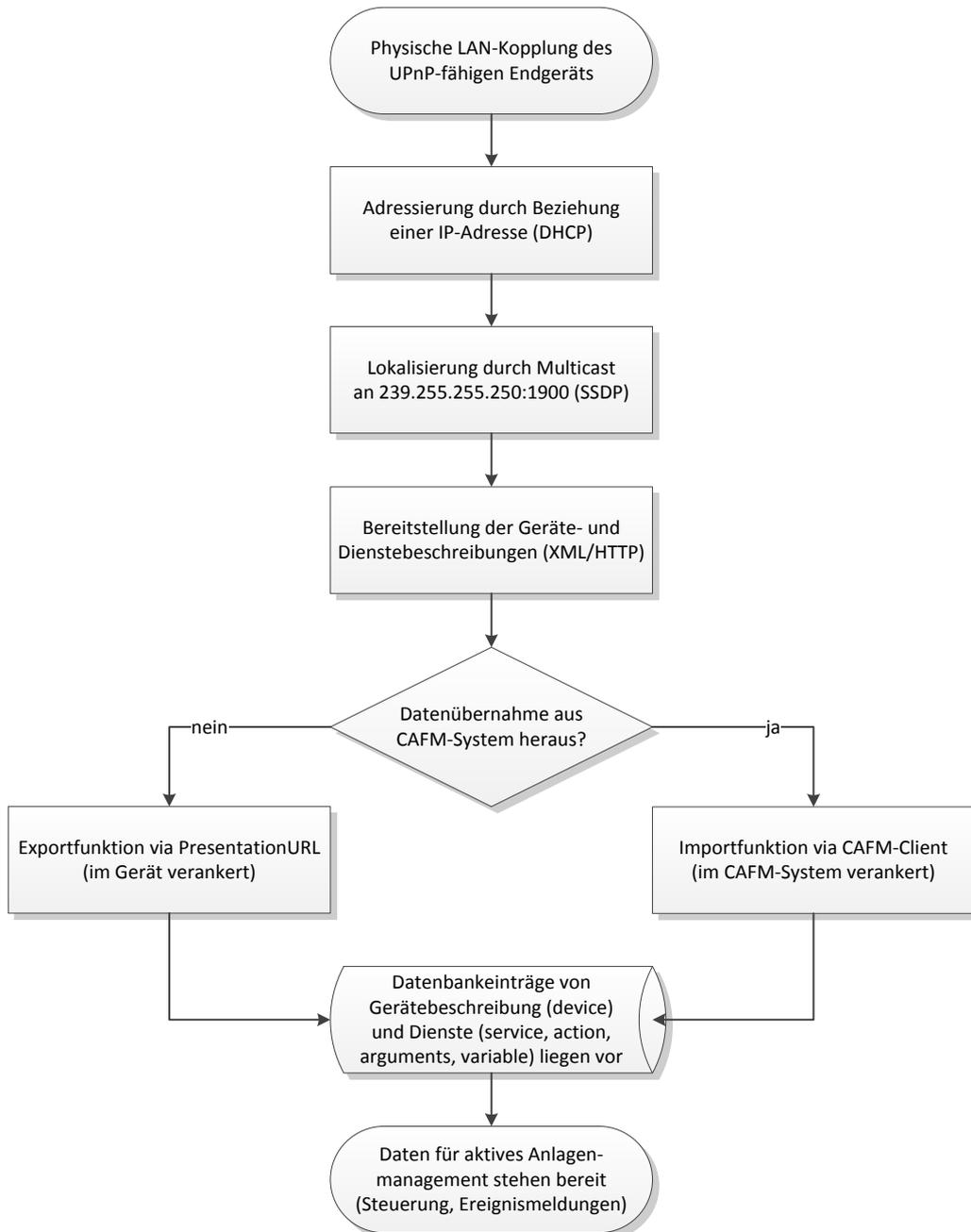


Abbildung 70: Flussdiagramm Geräteanbindung, eigene Darstellung

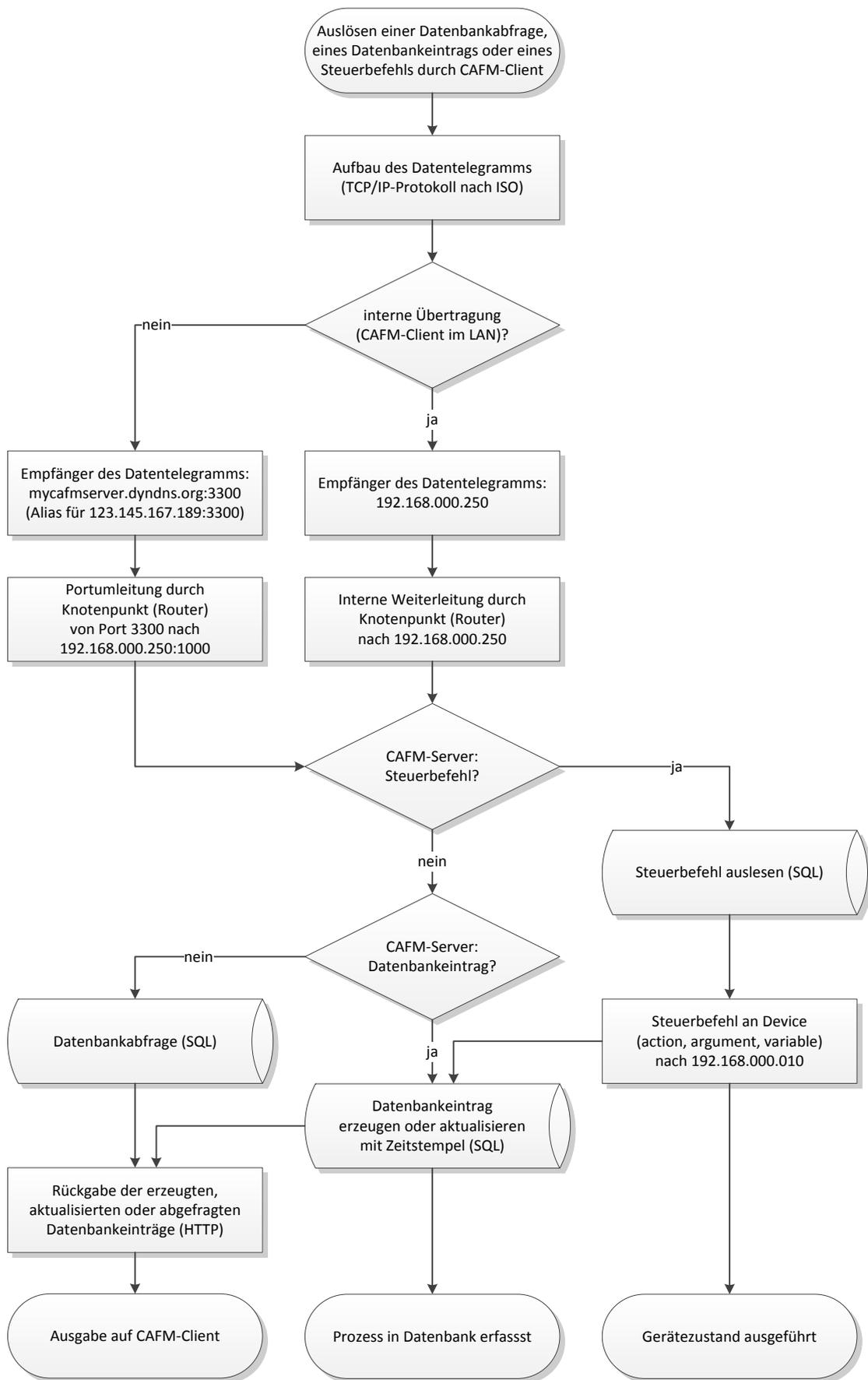


Abbildung 71: Flussdiagramm Datenbankabfrage, Datenbankeintrag und Gerätesteuerung, eigene Darstellung

5.5 Innovative Standards

Um eine reibungslose und herstellerunabhängige Funktionalität gewährleisten zu können, die für sämtliche Mess-, Steuer- und Regeleinheiten der Gebäudeautomation (MSR-Technik, Sensoren und Aktoren) verwendbar und nutzbar ist, bedarf es der Definition von einheitlichen, genormten Bausteinen bzw. Standards.

Das Konzept des erarbeiteten Modells beruht bereits auf einer Reihe von Standards, wie sie in den vorangegangenen Kapiteln behandelt worden sind: die Übertragungsmedien Ethernet und WLAN, das TCP/IP-Protokoll, der UPnP-Standard mit den dafür benötigten XML-Dateien, um nur einige zu nennen. Für die Kombination dieser Standards und die neue, fachspezifische CAFM-Ausrichtung und -Anwendung sind aber spezielle Vorgaben zu treffen, damit Programmierer und Hersteller bei der Entwicklung UPnP-fähiger Geräte die nötigen Dienste, Protokolle und Datenformate implementieren können.

Konkret geht es um den Aufbau und Inhalt der bereitzustellenden XML-Files für die UPnP-Anbindung und die automatisierte Integration in CAFM-Systeme. Es handelt sich um die Dateien `device.xml` und `scpd.xml`. Hinsichtlich der Gerätebeschreibung sind die Definitionen des UPnP-Forums aus der „Device Architecture“ für Geräte der Gebäudeautomation zu übertragen und zu ergänzen. Die Beschreibungen der Dienste hingegen sind entsprechend der neuen Anforderungen anzupassen bzw. neu auszurichten. Dazu bedarf es der Erstellung eines UPnP Service Schemas für einen neuen Gerätetyp. Dieses Schema wäre dem UPnP-Forum zwecks Aufnahme als neuen Standard vorzulegen. Die erweiterten und angepassten XML-Dateien sind in Kapitel 6.2 deklariert.

Um die Dateien `device.xml` und `scpd.xml` aufbauen zu können, müssen zunächst sämtliche relevanten Geräte- und Serviceattribute gesammelt werden, die für die Gebäudeautomation bedeutend sind. Im Anschluss müssen sie entsprechend der Service Description nach Aktionen (Funktion), Argumenten (Steuerbefehl) und Zustand- bzw. Statusvariablen sortiert werden. Diese Kategorisierung ist sowohl für den Aufbau der XML-Datei als auch den Aufbau der Musterdatenbank notwendig. Zudem ist es sinnvoll, zwischen Daten der Anlagenklasse (z. B. Produktserie/Modell-Nr.) und Daten des einzelnen Anlagenobjekts (z. B. Serien-Nr.) zu unterscheiden. Ferner ist die Herkunft der Daten festzulegen. Sie sind entweder vom Hersteller vordefiniert (z. B. Geräteeigenschaften) oder vom Benutzer anzugeben (z. B. Standort, Inventar-Nr. etc.).

Die nachstehenden Tabellen beinhalten sämtliche Attribute einer Steuereinheit, die in den Dateien `device.xml` und `scpd.xml` definiert sein müssen oder vom Anwender vor bzw. nach dem Importvorgang festzulegen sind. Die Sammlung beruht auf den Vorgaben des

UPnP-Standards mit einer umfangreichen Ergänzung von Attributen, die nach eigenen Recherchen für Geräte der technischen Gebäudeausrüstung zweckmäßig erscheinen.

Zusätzlich können auch Bilder und Datenblätter für einen Import auf dem Gerät hinterlegt sein. Als Formate eignen sich .jpg, .gif, .pdf, .txt, .csv, .docx oder .xlsx.

Anlagenklasse: Vordefinierte Eigenschaften

Bezeichnung	Datenbankfeld	Quelle
Gerätetyp	deviceType	UPnP-Standard
Bezeichnung	friendlyName	UPnP-Standard
Hersteller	manufacturer	UPnP-Standard
Hersteller Homepage	manufacturerURL	UPnP-Standard
Modellbeschreibung	modelDescription	UPnP-Standard
Modellbezeichnung	modelName	UPnP-Standard
Modellnummer	modelName	UPnP-Standard
Geräte-Homepage	modelURL	UPnP-Standard
Icon	iconURL	UPnP-Standard
Service ID	serviceId	UPnP-Standard
URL der Servicebeschreibung	SCPDURL	UPnP-Standard
URL der Steuerung	controlURL	UPnP-Standard
URL der Ereignismeldungen	eventSubURL	UPnP-Standard
URL der Gerätedarstellung	presentationURL	UPnP-Standard
Kategorie-Nummer	categoryNumber	UPnP-Erweiterung
Kategorie / Gewerk	categoryName	UPnP-Erweiterung
Kategoriebeschreibung	categoryDescription	UPnP-Erweiterung
Wartungstyp	mntType	UPnP-Erweiterung
Wartungsintervall	mntInterval	UPnP-Erweiterung
Wartungsintervall Einheit	mntIntervalTimeUnit	UPnP-Erweiterung
Wartungstätigkeiten	mntProcedures	UPnP-Erweiterung
Gewicht	deviceWeight	UPnP-Erweiterung
Länge	deviceLength	UPnP-Erweiterung

Tiefe	deviceDepth	UPnP-Erweiterung
Höhe	deviceHeight	UPnP-Erweiterung
Breite	deviceWidth	UPnP-Erweiterung
Volumen	volume	UPnP-Erweiterung
Standfläche	footprint	UPnP-Erweiterung

Tabelle 6: Vordefinierte Eigenschaften der Anlagenklasse

Anlagenobjekt: Vordefinierte Eigenschaften

Bezeichnung	Datenbankfeld	Quelle
Seriennummer	serialNumber	UPnP-Standard
Universal Unique Identifier	UDN	UPnP-Standard
Univeral Product Code	UPC	UPnP-Standard
Baujahr	manufactureDate	UPnP-Erweiterung
Gewährleistung in Monaten	warranty	UPnP-Erweiterung

Tabelle 7: Vordefinierte Eigenschaften des Anlagenobjekts

Anlagenobjekt: Benutzerdefinierte Eigenschaften

Bezeichnung	Datenbankfeld	Quelle
Inventarnummer	identNumber	benutzerdefiniert
Anlagenkennzeichnungsschlüssel	identName	benutzerdefiniert
URL des Geräts	baseURL	benutzerdefiniert
Anschaffungskosten	retailPrice	benutzerdefiniert
Inbetriebnahme (Datum)	startupDate	benutzerdefiniert
Wartungstermin	mntDate	benutzerdefiniert
Wartung Status	mntStatus	benutzerdefiniert

Tabelle 8: Benutzerdefinierte Eigenschaften des Anlagenobjekts

Anlagenklasse: Vordefinierte Services

Bezeichnung	Datenbankfeld	Quelle
Aktion (Dienst/Funktion)	actionName	UPnP-Standard
Argument (Steuerbefehl/Parameter)	argumentName	UPnP-Standard
Argument-/Befehlsrichtung	direction	UPnP-Standard
Argumentvariable	relatedStateVariable	UPnP-Standard

Tabelle 9: Vordefinierte Services der Anlagenklasse

Anlagenobjekt: Vordefinierte Variablen

Bezeichnung	Datenbankfeld	Quelle
Variable Name	stateVariableName	UPnP-Standard
Variable Wert (zur Laufzeit)	stateVariableValue	UPnP-Erweiterung
Variable Typ	dataType	UPnP-Standard
Variable Standardwert	defaultValue	UPnP-Standard
Variable Minimalwert	minimum	UPnP-Standard
Variable Maximalwert	maximum	UPnP-Standard
Variable Stufen/Schrittweite	step	UPnP-Standard
Variable Eventmitteilung	sendEvents	UPnP-Standard

Tabelle 10: Vordefinierte Variablen des Anlagenobjekts

Diese Sammlung von Attributen ist in den CAFM-Datenbanken zu berücksichtigen. Die entsprechenden Felder müssen an geeigneter Stelle vorgehalten werden. Die empfohlene Namensgebung (zweite Spalte) muss nicht zwingend übernommen werden. Bei Abweichungen ist allerdings eine Übersetzungs-/Zuordnungstabelle für den Import notwendig.

Anhand dieser Sammlung wird eine Musterdatenbank entworfen (siehe Kapitel 6.1), mit der die praktische Anwendung demonstriert werden kann. Der Aufbau der Tabellen und Relationen wird sich an der dargestellten Sortierung bzw. Gruppierung orientieren.

Die Werte der Serviceattribute aus Tabelle 9 werden entsprechend ihrer Funktion und unter Berücksichtigung der Vorgaben der UPnP Service Description benannt. Die eigentliche Funktionalität ergibt sich aus den in den Aktionen definierten Argumenten (Parameter/Steuerbefehle) und Variablen (Messwerte und Zustandsdaten). Aus diesem Grund sind in den folgenden Tabellen eigene Empfehlungen von Standard-Funktionen, -Steuerbefehlen und -Variablen aufgeführt, wie sie in den XML-Dateien Anwendung finden sollen. Sie berücksichtigen die Vorgaben des UPnP-Standards zur Namensgebung.

Die Sammlung erhebt keinen Anspruch auf Vollständigkeit, doch sie erfüllt die Anforderungen für die Abbildung grundlegender Funktionen und Prozesse im Bereich des Anlagen- und Wartungsmanagements. Die Attribute sind alphabetisch sortiert.

Funktionen der Anlagenklasse (<action>)

<action>	<name>
Abwärme Zustandswert	GetHeatLoss
Anlage einschalten/ausschalten	SetDevicePower
Anlage Zustandswert	GetDevicePower
Anlaufstrom erhöhen/verringern	SetInitialPower
Anlaufstrom Zustandswert	GetInitialPower
Auflösung (optisch) erhöhen/verringern	SetResolution
Auflösung (optisch) Zustandswert	GetResolution
Beleuchtung einschalten/ausschalten	SetLightningPower
Beleuchtung Zustandswert	GetLightningPower
Betriebsdruck Zustandswert	GetOperatingPressure
Betriebsgeräusch Zustandswert	GetOperatingNoise
Betriebsstunden Zustandswert	GetOperatingHours
Betriebstemperatur Zustandswert	GetOperatingTemperature
Drehzahl erhöhen/verringern	SetOperatingRotationalSpeed
Drehzahl Zustandswert	GetOperatingRotationalSpeed
Druck Kessel Zustandswert	GetBoilerPressure
Druckventil öffnen/schließen	SetPressureValvePosition
Druckventil Zustandswert	GetPressureValvePosition
Fenster öffnen/schließen	SetWindowPosition
Fenster Zustandswert	GetWindowPosition

Förderhöhe erhöhen/verringern	SetOperatingStaticHead
Förderhöhe Zustandswert	GetOperatingStaticHead
Förderleistung erhöhen/verringern	SetPumpDeliveryRate
Förderleistung Zustandswert	GetPumpDeliveryRate
Frequenz erhöhen/verringern	SetOperatingFrequency
Frequenz Zustandswert	GetOperatingFrequency
Geschwindigkeit erhöhen/verringern	SetOperatingSpeed
Geschwindigkeit Zustandswert	GetOperatingSpeed
Heizbrenner einschalten/ausschalten	SetHeatingPower
Heizbrenner Zustandswert	GetHeatingPower
Heizleistung erhöhen/verringern	SetHeatOutput
Heizleistung Zustandswert	GetHeatOutput
Helligkeit erhöhen/verringern	SetBrightness
Helligkeit Zustandswert	GetBrightness
Jalousie öffnen/schließen	SetLouverPosition
Jalousie Zustandswert	GetLouverPosition
Klimatisierung einschalten/ausschalten	SetConditioningPower
Klimatisierung Zustandswert	GetConditioningPower
Kühlleistung erhöhen/verringern	SetCoolingEfficiency
Kühlleistung Zustandswert	GetCoolingEfficiency
Leistungsabgabe Zustandswert	GetPowerOutput
Leistungsaufnahme Zustandswert	GetPowerInput
Leistungsstufe erhöhen/verringern	SetPowerLevel
Leistungsstufe Zustandswert	GetPowerLevel
Luftbefeuchtung erhöhen/verringern	SetHumidification
Luftbefeuchtung Zustandswert	GetHumidification
Luftfeuchte Zustandswert	GetHumidity
Lüftung/Ventilator einschalten/ausschalten	SetVentilationPower
Lüftung/Ventilator Zustandswert	GetVentilationPower
Massenstrom erhöhen/verringern	SetMassFlowRate
Massenstrom Zustandswert	GetMassFlowRate
Spannung erhöhen/verringern	SetVoltage
Spannung Zustandswert	GetVoltage

Temperatur Außenluft Zustandswert	GetAirTemperatureOutside
Temperatur Heizkessel erhöhen/verringern	SetBoilerTemperature
Temperatur Heizkessel Zustandswert	GetBoilerTemperature
Temperatur Luftstrom erhöhen/verringern	SetVentilationTemperature
Temperatur Luftstrom Zustandswert	GetVentilationTemperature
Temperatur Raumluft Zustandswert	GetAirTemperatureInside
Temperatur Rücklauf Wasser Zustandswert	GetWaterTemperatureReturn
Temperatur Vorlauf Wasser Zustandswert	GetWaterTemperatureSupply
Temperatur Wasserspeicher erhöhen/verringern	SetWaterReservoirTemperature
Temperatur Wasserspeicher Zustandswert	GetWaterReservoirTemperature
Tür öffnen/schließen	SetDoorPosition
Tür Zustandswert	GetDoorPosition
Verbrauch Gas Zustandswert	GetGasConsumption
Verbrauch Kraftstoff Zustandswert	GetFuelConsumption
Verbrauch Öl Zustandswert	GetOilConsumption
Verbrauch Strom Zustandswert	GetCurrentConsumption
Verbrauch Wasser Zustandswert	GetWaterConsumption
Volumenstrom erhöhen/verringern	SetVolumeFlowRate
Volumenstrom Zustandswert	GetVolumeFlowRate
Wirkleistung Zustandswert	GetEffectivePower
Wirkungsgrad Zustandswert	GetElectricalEfficiency
Zählerstand Zustandswert	GetCounter

Tabelle 11: Funktionen der Anlagenklasse (<action>)

Aktionen beinhalten mindestens ein Argument. Prinzipiell können sie auch mehrere Argumente beinhalten. Es wäre möglich, der Aktion „Heizung einschalten“ die Argumente „Heizung ein“ und „Fenster zu“ zu hinterlegen. Doch eine solche Logik ist eher im CAFM-System zu implementieren. Diese Steuerbefehle sollten für sich stehen und separat aufrufbar sein. Die Parameter ähneln dem Aktionsnamen. Statt „Get“ wird der Präfix „ret“ (return) und statt „Set“ der Präfix „new“ (neu) verwendet. Mit diesen Befehlen werden die zugehörigen Variablen zurückgegeben oder neu gesetzt. Die Endungen „Target“ und „Status“ stehen als Synonym für die Verwendungsart der zugehörigen Variablen, wie im Folgenden erläutert. Auch hier sind die Vorgaben des UPnP-Standards zur Namensgebung berücksichtigt.

Parameter der Aktionen (<argument>)

<argument>	<name>	in/out
Abwärme Zustandswert	retHeatLossStatus	out
Anlage einschalten/ausschalten	newDevicePowerTarget	in
Anlage Zustandswert	retDevicePowerTarget	out
Anlaufstrom erhöhen/verringern	newInitialPowerTarget	in
Anlaufstrom Zustandswert	retInitialPowerTarget	out
Auflösung (optisch) erhöhen/verringern	newResolutionTarget	in
Auflösung (optisch) Zustandswert	retResolutionTarget	out
Beleuchtung einschalten/ausschalten	newLightningPowerTarget	in
Beleuchtung Zustandswert	retLightningPowerTarget	out
Betriebsdruck Zustandswert	retOperatingPressureStatus	out
Betriebsgeräusch Zustandswert	retOperatingNoiseStatus	out
Betriebsstunden Zustandswert	retOperatingHoursStatus	out
Betriebstemperatur Zustandswert	retOperatingTemperatureStatus	out
Drehzahl erhöhen/verringern	newOperatingRotationalSpeedTarget	in
Drehzahl Zustandswert	retOperatingRotationalSpeedTarget	out
Druck Kessel Zustandswert	retBoilerPressureStatus	out
Druckventil öffnen/schließen	newPressureValvePositionTarget	in
Druckventil Zustandswert	retPressureValvePositionTarget	out
Fenster öffnen/schließen	newWindowPositionTarget	in
Fenster Zustandswert	retWindowPositionTarget	out
Förderhöhe erhöhen/verringern	newOperatingStaticHeadTarget	in
Förderhöhe Zustandswert	retOperatingStaticHeadTarget	out
Förderleistung erhöhen/verringern	newPumpDeliveryRateTarget	in
Förderleistung Zustandswert	retPumpDeliveryRateTarget	out
Frequenz erhöhen/verringern	newOperatingFrequencyTarget	in
Frequenz Zustandswert	retOperatingFrequencyTarget	out
Geschwindigkeit erhöhen/verringern	newOperatingSpeedTarget	in
Geschwindigkeit Zustandswert	retOperatingSpeedTarget	out
Heizbrenner einschalten/ausschalten	newHeatingPowerTarget	in
Heizbrenner Zustandswert	retHeatingPowerTarget	out

Heizleistung erhöhen/verringern	newHeatOutputTarget	in
Heizleistung Zustandswert	retHeatOutputTarget	out
Helligkeit erhöhen/verringern	newBrightnessTarget	in
Helligkeit Zustandswert	retBrightnessTarget	out
Jalousie öffnen/schließen	newLouverPositionTarget	in
Jalousie Zustandswert	retLouverPositionTarget	out
Klimatisierung einschalten/ausschalten	newConditioningPowerTarget	in
Klimatisierung Zustandswert	retConditioningPowerTarget	out
Kühlleistung erhöhen/verringern	newCoolingEfficiencyTarget	in
Kühlleistung Zustandswert	retCoolingEfficiencyTarget	out
Leistungsabgabe Zustandswert	retPowerOutputStatus	out
Leistungsaufnahme Zustandswert	retPowerInputStatus	out
Leistungsstufe erhöhen/verringern	newPowerLevelTarget	in
Leistungsstufe Zustandswert	retPowerLevelTarget	out
Luftbefeuchtung erhöhen/verringern	newHumidificationTarget	in
Luftbefeuchtung Zustandswert	retHumidificationTarget	out
Luftfeuchte Zustandswert	retHumidityStatus	out
Lüftung/Ventilator einschalten/ausschalten	newVentilationPowerTarget	in
Lüftung/Ventilator Zustandswert	retVentilationPowerTarget	out
Massenstrom erhöhen/verringern	newMassFlowRateTarget	in
Massenstrom Zustandswert	retMassFlowRateTarget	out
Spannung erhöhen/verringern	newVoltageTarget	in
Spannung Zustandswert	retVoltageTarget	out
Temperatur Außenluft Zustandswert	retAirTemperatureOutsideStatus	out
Temperatur Heizkessel erhöhen/verringern	newBoilerTemperatureTarget	in
Temperatur Heizkessel Zustandswert	retBoilerTemperatureTarget	out
Temperatur Luftstrom erhöhen/verringern	newVentilationTemperatureTarget	in
Temperatur Luftstrom Zustandswert	retVentilationTemperatureTarget	out
Temperatur Raumluft Zustandswert	retAirTemperatureInsideStatus	out
Temperatur Rücklauf Wasser Zustandswert	retWaterTemperatureReturnStatus	out
Temperatur Vorlauf Wasser Zustandswert	retWaterTemperatureSupplyStatus	out
Temperatur Wasserspeicher erhöhen/verringern	newWaterReservoirTemperatureTarget	in
Temperatur Wasserspeicher Zustandswert	retWaterReservoirTemperatureTarget	out

Tür öffnen/schließen	newDoorPositionTarget	in
Tür Zustandswert	retDoorPositionTarget	out
Verbrauch Gas Zustandswert	retGasConsumptionStatus	out
Verbrauch Kraftstoff Zustandswert	retFuelConsumptionStatus	out
Verbrauch Öl Zustandswert	retOilConsumptionStatus	out
Verbrauch Strom Zustandswert	retCurrentConsumptionStatus	out
Verbrauch Wasser Zustandswert	retWaterConsumptionStatus	out
Volumenstrom erhöhen/verringern	newVolumeFlowRateTarget	in
Volumenstrom Zustandswert	retVolumeFlowRateTarget	out
Wirkleistung Zustandswert	retEffectivePowerStatus	out
Wirkungsgrad Zustandswert	retElectricalEfficiencyStatus	out
Zählerstand Zustandswert	retCounterStatus	out

Tabelle 12: Parameter der Aktionen (<argument>)

Entsprechend des UPnP-Standards müssen Variablen als <stateVariable> in der sog. <serviceStateTable> definiert sein, bevor sie von einem Steuerbefehl aus der <argument>-Anweisung als <relatedStateVariable> aufgerufen bzw. verwendet werden können. <relatedStateVariable> und <stateVariable> müssen dabei den gleichen Namen, den gleichen Datentyp, den gleichen Wertebereich etc. haben.

Grundsätzlich gibt es zwei Arten der Verwendung von Variablen. Zum einen als Mess- und Sensorwerte (Status), zum anderen als Betriebs- und Aktorzustände (Target). Status-Variablen können nur ausgelesen, aber nicht direkt beeinflusst bzw. aktiv gesteuert werden. Target-Variablen können nicht nur ausgelesen, sondern auch vorgegeben und direkt beeinflusst bzw. aktiv gesteuert werden.

Variablen des Anlagenobjekts (<stateVariable>)

<stateVariable>	<name>	usage	unit
Abwärme	HeatLoss	Status	kW
Anlage ein/aus	Power	Target	
Anlaufstrom	InitialPower	Target	A
Außenlufttemperatur	AirTemperatureOutside	Status	°C

Beleuchtung ein/aus	Light	Target	
Betriebsdruck	OperatingPressure	Status	bar
Betriebsgeräusch	OperatingNoise	Status	dB
Betriebsstunden	OperatingHours	Status	h
Betriebstemperatur	OperatingTemperature	Status	°C
Bild-/Videoauflösung (optisch)	Resolution	Target	dpi
Drehzahl	OperatingRotationalSpeed	Target	1/min
Förderhöhe	OperatingStaticHead	Target	m
Förderleistung	PumpDeliveryRate	Target	m ³ /h
Frequenz	OperatingFrequency	Target	Hz
Gasverbrauch	GasConsumption	Status	m ³
Geschwindigkeit	OperatingSpeed	Target	m/s
Heizbrenner ein/aus	HeatingBurner	Target	
Heizkesseltemperatur	BoilerTemperature	Target	°C
Heizleistung	HeatOutput	Target	kW
Helligkeit	Brightness	Target	lumen
Kesseldruck	BoilerPressure	Status	bar
Klimatisierung ein/aus	AirConditioning	Target	
Kraftstoffverbrauch	FuelConsumption	Status	liter
Kühlleistung	CoolingEfficiency	Target	kW
Leistungsabgabe	PowerOutput	Status	kVA
Leistungsaufnahme	PowerInput	Status	kVA
Leistungsstufe	Level	Target	
Luftbefeuchtung	Humidification	Target	
Luftfeuchte	Humidity	Status	%rel
Luftstromtemperatur	VentilationTemperature	Target	°C
Lüftung/Ventilator ein/aus	Ventilation	Target	
Massenstrom	MassFlowRate	Target	kg/h
Ölverbrauch	OilConsumption	Status	liter
Position Druckventil	PressureValve	Target	
Position Fenster	WindowPosition	Target	
Position Jalousie	LouverPosition	Target	
Position Tür	DoorPosition	Target	

Raumlufttemperatur	AirTemperatureInside	Status	°C
Spannung	Voltage	Target	V
Stromverbrauch	CurrentConsumption	Status	kWh
Volumenstrom	VolumeFlowRate	Target	m ³ /h
Wasserspeichertemperatur	WaterReservoirTemperature	Target	°C
Wassertemperatur Rücklauf	WaterTemperatureReturn	Status	°C
Wassertemperatur Vorlauf	WaterTemperatureSupply	Status	°C
Wasserverbrauch	WaterConsumption	Status	m ³
Wirkleistung	EffectivePower	Status	kW
Wirkungsgrad	ElectricalEfficiency	Status	Eta
Zählerstand allgemein	Counter	Status	

Tabelle 13: Variablen des Anlagenobjekts (<stateVariable>)

Die nach UPnP-Vorgaben ausgerichteten und durch die Ergänzung notwendiger und zweckmäßiger Attribute entstandenen XML-Dateien sind das Beispiel einer Steuereinheit für Heizungsanlagen (beschränkt auf die dafür relevanten Attribute) in Kapitel 6.2 für und für ein universelles Gerät der technischen Gebäudeausrüstung (nach den Tabellen 6-13 vollumfängliche Attributliste) im Anhang dieser Arbeit zu finden.

Namensgebung

Für das Schnittstellenkonzept ist ein Name oder Begriff zu deklarieren, der als Synonym für den neuen Standard bzw. das neue Austauschformat steht. Als Begriff für den Datenaustausch und die Steuerung von UPnP-fähigen Geräten und Anlagen im Bereich des Facility Managements eignet sich: **Facility Management and Automation Control Data Exchange**, kurz **FMA.Codex**.

Ein weiterer Standard: Jede CAFM-Software, die den „**FMA.Codex**“ implementiert und die Funktion des UPnP-Geräteimports bereitstellt, kann als „**UPnP ready**“ gekennzeichnet werden.

6 Praktische Umsetzung

Die Entwicklungen im Bereich der Gebäudeautomation tendieren immer mehr in Richtung „Intelligentes Haus“ bzw. Smart Home. Die Vernetzung über das meist vorhandene Computernetzwerk ist naheliegend. Dennoch sind Technologien wie UPnP in der Gebäudeautomation nicht verbreitet. Da es nach derzeitigem Kenntnisstand keine UPnP-fähigen Steuerzentralen in den klassischen Gewerken Heizung, Klima, Lüftung und Sanitär gibt, kann die praktische Umsetzung des Modells derzeit nur bedingt realisiert werden.

Modellentwicklung, Prozessabläufe, Protokolle, XML-Datenformate für den UPnP-Standard, Datenbankentwurf und Programmbausteine sind in dieser Arbeit behandelt und werden auch praktisch umgesetzt. Da es aber an der nötigen Hardware fehlt, lässt sich der reale Betrieb nur simulieren. Der Schwerpunkt liegt in der infrastrukturellen und softwareseitigen Konzeption.

In diesem Kapitel wird auf Basis des entwickelten Integrationsmodells, der ausgewählten Technologien und des hergeleiteten Datenbedarfs ein relationales Datenbankmodell aufgebaut (siehe Kapitel 6.1) und die entsprechenden UPnP Device bzw. Service Descriptions im XML-Format erstellt (siehe Kapitel 6.2). Die Datenbank bildet den Kern einer kleinen webbasierten Applikation mit Import- und Steuerungsfunktion (siehe Kapitel 6.3).

6.1 Aufbau des Datenbankmodells

Sämtliche in Kapitel 5.5 aufgeführten Attribute sind unter Beachtung der empfohlenen Bezeichnung bei der Erstellung des Datenbankmodells zu berücksichtigen. Der Aufbau der Tabellen orientiert sich an der bereits dargestellten Sortierung bzw. Gruppierung. Die Attribute definieren die Spaltennamen einer Tabelle.

Die Datenbank beinhaltet außerdem weitere Tabellen, deren Inhalt zwar nicht über das UPnP-fähige Endgerät bereitgestellt oder durch den Anwender beim Importvorgang angegeben wird, die aber in direktem Zusammenhang mit dem Anlagen- und Wartungsmanagement stehen. Dazu gehören Tabellen mit Angaben zum Standort der Anlage, Tabellen zum Wartungsvertrag und Tabellen zum Hersteller bzw. Vertragspartner.

Dennoch ist dieses Datenbankmodell nicht als vollständige CAFM-Datenbank, sondern als „Auszug“ zu verstehen, der sich auf die grundlegenden Bereiche des Anlagen- und Wartungsmanagements beschränkt. Die Datenbank bildet nur einen Teilbereich ab, der zur Demonstration der Anwendung, insbesondere einer Importfunktion, notwendig ist.

6.1.1 Tabellen

Zu Beginn des Datenbankentwurfs ist zu überlegen, welche Tabellen benötigt werden und welche Attribute sie beinhalten. Diese richten sich nach den zusätzlichen Attributen für die erweiterten UPnP Device und Service Descriptions, wie in Kapitel 5.5 dargestellt. Die Tabellenbezeichnungen sind an deren Inhalt orientiert gewählt. Bei der Konzeption ist strikt darauf zu achten, dass alle Anforderungen an eine relationale Datenbank hinsichtlich Normalformen und redundanzfreier Datenhaltung erfüllt sind.

Der Aufbau basiert auf einer objektorientierten Betrachtungsweise. Es wird zwischen übergeordneter Anlagenklasse als Gruppierung gleichartiger Geräte (modellbezogen) und einzelner Anlagenobjekt als Instanz unterschieden. So sind z. B. alle baugleichen Aktoren (wie Stellmotoren für Lüftungsklappen) zwar als Einzelobjekt mit den geräteabhängigen Attributen (Seriennummer, Inbetriebnahme etc.) erfasst, gehören aber zur selben Anlagenklasse mit gemeinsamen Attributen (Modellnummer, Steuerbefehle etc.).

Tabelle Anlagenklasse (deviceClass)



deviceClass	
ID	
manufacturerID	
modelNumber	
categoryID	
deviceType	
friendlyName	
manufacturerURL	
modelDescription	
modelName	
modelURL	
iconURL	
serviceId	
SCPDURL	
controlURL	
eventSubURL	
presentationURL	
mntContractID	
mntType	
mntInterval	
mntIntervalTimeUnit	
mntProcedures	
deviceWeight	
deviceLength	
deviceDepth	
deviceHeight	
deviceWidth	
volume	
footprint	

Diese Tabelle beinhaltet sämtliche Attribute, die für alle Objekte einer Klasse übergeordnet gelten. Der aus der objektorientierten Sprache abgeleitete Begriff „Klasse“ steht in diesem Fall für die Beschreibung einer Produktserie oder Modellreihe. Neben den modellbezogenen Eigenschaften aus der Standard-Gerätebeschreibung sind in dieser Tabelle auch Daten zur Wartung sowie Maße und Gewicht abgelegt.

Ein zusammengesetzter Primärschlüssel gewährleistet die Eindeutigkeit eines Eintrags. Somit kann die Kombination von Hersteller und Produktserie nur einmal existieren. Sollte es die gleiche Modellnummer auch von einem anderen Hersteller geben, ist dies durch eine neue Kombination möglich. Als Referenz für die Relationen zu untergeordneten Tabellen wird die fortlaufend nummerierte ID herangezogen.

Abbildung 72: Tabelle deviceClass

Tabelle Anlagenobjekt (deviceObject)

deviceObject	
ID	
deviceClassID	
serialNumber	
identNumber	
identName	
baseURL	
UDN	
UPC	
manufactureDate	
warranty	
retailPrice	
startupDate	
mntDate	
mntStatus	
roomID	

In dieser Tabelle sind sämtliche Attribute eines spezifischen Geräts erfasst. Im Unterschied zur Anlagenklasse kennzeichnen diese geräteabhängigen Eigenschaften das Einzelobjekt durch seine Seriennummer, Inventarnummer etc. sowie die Angabe des Standorts. Auch hier gewährleistet ein zusammengesetzter Primärschlüssel von Modellreihe und Seriennummer die Eindeutigkeit eines Eintrags. Die Seriennummer alleine ist nicht ausreichend, da sie auch in anderen Produktreihen (auch anderer Hersteller) vorkommen könnte.

Abbildung 73: Tabelle deviceObject

Tabelle Servicevariablen (serviceStateTable)

serviceStateTable	
ID	
deviceObjectID	
stateVariableName	
stateVariableValue	
dataType	
defaultValue	
minimum	
maximum	
step	
sendEvents	

Zu jedem Anlagenobjekt gehören individuelle Variablen, die in der Service Description definiert werden. Sie beschreiben den Gerätestatus und werden aus den Funktionen (Aktionen) heraus abgerufen bzw. geändert. Anlagenobjekt und Variable bilden den zusammengesetzten Primärschlüssel.

Abbildung 74: Tabelle serviceStateTable

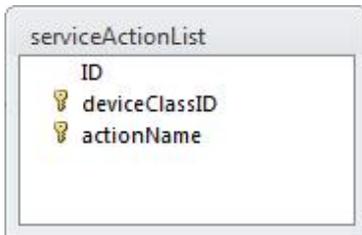
Tabelle Historie (history)

history	
ID	
stateVariableID	
timeStamp	
oldValue	
newValue	

Zur Historisierung der Anlagensteuerung ist die Aufzeichnung einer Änderung von Zustandsvariablen erforderlich. Sie wird in einer separaten Tabelle erfasst und mit einem Zeitstempel versehen.

Abbildung 75: Tabelle history

Tabelle Geräteservices (serviceActionList)



Diese Tabelle beinhaltet die Geräteservices bzw. Aktionen einer Anlagenklasse. In einer Klasse können mehrere Aktionen implementiert sein. Die Kombination von Anlagenklasse und Aktion ist eindeutig.

Abbildung 76: Tabelle serviceActionList

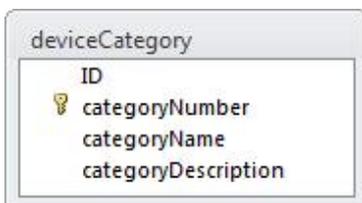
Tabelle Steuerung (serviceArgumentList)



Jede Aktion besitzt ein oder mehrere Argumente bzw. Steuerbefehle. Der Primärschlüssel setzt sich aus der klassenabhängigen Aktion und dem Argument zusammen. Von hier aus werden die Variablen aus der Tabelle serviceStateTable abgerufen.

Abbildung 77: Tabelle serviceArgumentList

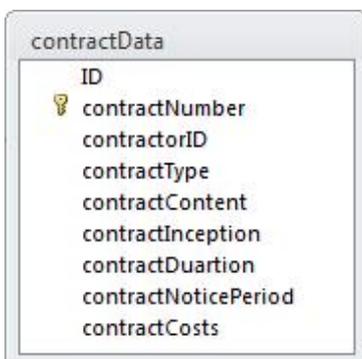
Tabelle Anlagenkategorie (deviceCategory)



Die Kategorisierung der Anlagenklassen sollte nicht im Tabellenblatt der Klasse selbst definiert sein, sondern durch einen Verweis auf eine separate Tabelle, in der sämtliche Gewerke gelistet sind.

Abbildung 78: Tabelle deviceCategory

Tabelle Wartungsvertrag (contractData)



Grundlage des Anlagen- und Wartungsmanagements sind Wartungsverträge, die ebenfalls in der Datenbank abgelegt werden. Im Gegensatz zu den vorigen Tabellen werden diese Attribute nicht automatisiert bereitgestellt, sondern aus dem Prozess des Vertragsmanagements vom CAFM-System vorgehalten.

Abbildung 79: Tabelle contractData

Tabelle Kontakt-/Firmendaten (contactData)

contactData	
ID	
companyName	
companyStreet	
companyZipCode	
companyCity	
companyLand	
companyPhone	
companyURL	

Die Beispieldatenbank ist um eine weitere Tabelle ergänzt, in der die Kontaktdaten von Firmen hinterlegt sind. Dies ist notwendig, um Hersteller und Vertragspartner vollständig abbilden zu können. Die entsprechenden Datenfelder verweisen dann auf den entsprechenden Firmeneintrag.

Abbildung 80: Tabelle contactData

Tabellen zum Standort (locrefBuilding, locrefFloor, locrefRoom)

locrefBuilding	
ID	
buildingName	
buildingDescription	

Der Ortsbezug eines Anlagenobjekts wird nicht erst bei der Kopplung bzw. dem Datenimport erzeugt, sondern ist im Zuge des Flächenmanagements bereits im CAFM-System erfasst worden. Eine Verknüpfung zur Kombination Gebäude/Geschoss/Raum definiert den genauen Standort eines Anlagenobjekts.

Abbildung 81: Tabelle locrefBuilding

locrefFloor	
ID	
buildingID	
floorName	
floorDescription	

Ein Gebäude besteht aus einem oder mehreren Geschossen. Dabei ist die Kombination aus Gebäude und Geschoss eindeutig, wie durch den zusammengesetzten Primärschlüssel festgelegt.

Abbildung 82: Tabelle locrefFloor

locrefRoom	
ID	
floorID	
roomName	
roomDescription	
roomUsage	
roomFlooring	
roomSize	

Schließlich definiert der Raum als kleinste Flächeneinheit den konkreten Standort des Anlagenobjekts. Der Raum bildet zusammen mit dem Geschoss eines Gebäudes die eindeutige Zuordnung.

Abbildung 83: Tabelle locrefRoom

6.1.2 Relationen

Wie zu Beginn des Kapitels gesagt, handelt es sich bei dieser Konzeption um ein relationales Datenbankmodell. Mit der Definition der Tabellen ist der erste Schritt des Entwurfs getan. Sie müssen nun miteinander in Beziehung gebracht werden.

Das zur Verknüpfung herangezogene Datenfeld ist immer die ID des jeweiligen Eintrags. Die Methode der zusammengesetzten Primärschlüssel steht dabei nicht in Widerspruch, sie gewährleisten die Eindeutigkeit der Einträge. Doch als Referenz ist die fortlaufend nummerierte ID einfacher zu handhaben.

Die folgende Abbildung der Relationen ist im Anhang in größerer Darstellung zu finden:

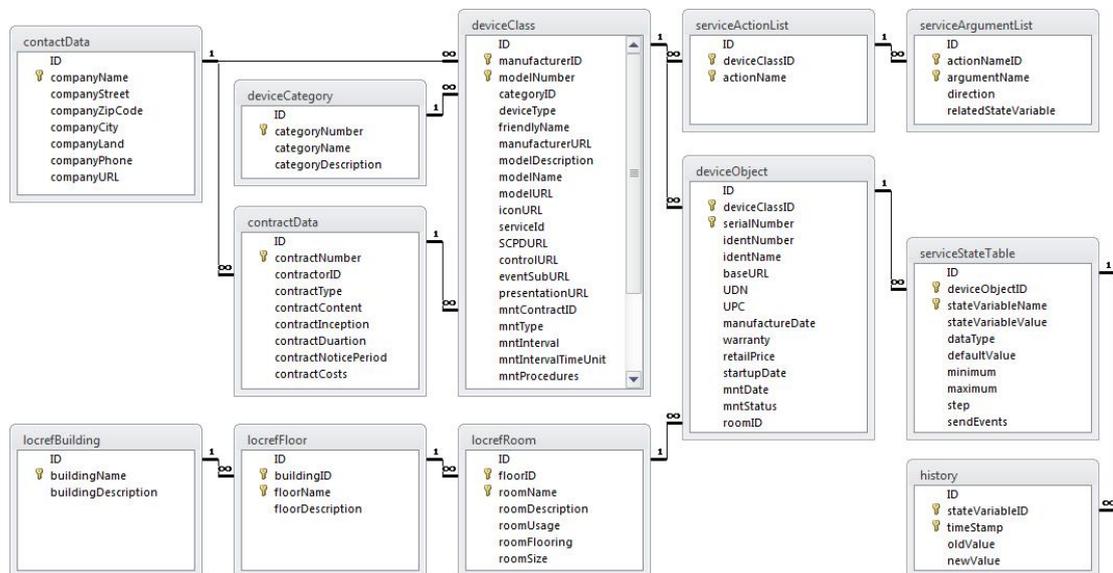


Abbildung 84: Struktur der relationalen Datenbank

Die Tabellenbeziehungen der in der Übersicht abgebildeten Datenbankstruktur sind im Detail erläutert:

- **Beziehung zwischen Kontakt-/Firmendaten und Anlagenklasse**
Eine Firma kann der Hersteller von mehreren Anlageklassen (Modellreihen) sein. Eine Modellreihe wird aber nur von einer Firma hergestellt. Daraus ergibt sich eine 1:n-Beziehung zwischen den Feldern contactData.ID und deviceClass.manufacturerID.

- **Beziehung zwischen Kontakt-/Firmendaten und Wartungsvertrag**
Eine Firma kann Vertragsnehmer mehrere Wartungsverträge sein. Ein Vertrag kann aber nur mit einer Firma abgeschlossen werden. Daraus ergibt sich eine 1:n-Beziehung zwischen den Feldern `contactData.ID` und `contractData.contractorID`.
- **Beziehung zwischen Anlagenkategorie und Anlagenklasse**
Ein Gewerk kann eine Vielzahl von Modellreihen/Anlageklassen beinhalten. Eine Modellreihe kann aber nur einem Gewerk zugeordnet sein. Daraus ergibt sich eine 1:n-Beziehung zwischen den Feldern `deviceCategory.ID` und `deviceClass.categoryID`.
- **Beziehung zwischen Wartungsvertrag und Anlagenklasse**
Ein Wartungsvertrag kann sich auf mehrere Anlagenklassen als Vertragsgegenstand beziehen. Eine Anlagenklasse unterliegt aber nur einem Wartungsvertrag. Daraus ergibt sich eine 1:n-Beziehung zwischen den Feldern `contractData.ID` und `deviceClass.mntContractID`.
- **Beziehung zwischen Anlagenklasse und Anlagenobjekt**
Eine Anlagenklasse ist der übergeordnete Sammler gleicher Komponenten bzw. Objekte. Ein Anlagenobjekt kann aber nur zu einer Klasse gehören. Daraus ergibt sich eine 1:n-Beziehung zwischen den Feldern `deviceClass.ID` und `deviceObject.deviceClassID`.
- **Beziehung zwischen Anlagenklasse und Geräteservices**
Eine Anlagenklasse kann mehrere Aktionen innehaben. Eine Aktion ist aber nur einer einzigen Klasse zugeordnet. Daraus ergibt sich eine 1:n-Beziehung zwischen den Feldern `deviceClass.ID` und `serviceActionList.deviceClassID`.
- **Beziehung zwischen Geräteservices und Steuerung**
Eine Aktion kann sich aus mehreren Steuerbefehlen zusammensetzen. Ein Steuerbefehl kann aber nicht zu verschiedenen Aktionen gehören. Daraus ergibt sich eine 1:n-Beziehung zwischen den Feldern `serviceActionList.ID` und `serviceArgumentList.actionNameID`.

- **Beziehung zwischen Anlagenobjekt und Servicevariablen**
Einem einzelnen Anlagenobjekt sind mehrere Variablen zugeordnet, die den Gerätestatus festhalten. Der Status gilt aber nur für dieses eine Gerät. Daraus ergibt sich eine 1:n-Beziehung zwischen den Feldern `deviceObject.ID` und `serviceStateTable.deviceObjectID`.
- **Beziehung zwischen Servicevariablen und Historie**
Für eine Servicevariable werden fortlaufend (chronologisch) Zustände erfasst. Eine Variable hat damit im Laufe der Zeit eine Vielzahl an Zuständen. Diese Zustände gelten aber nur für eine separate Gerätevariable. Daraus ergibt sich eine 1:n-Beziehung zwischen den Feldern `serviceStateTable.ID` und `history.stateVariableID`.
- **Beziehung zwischen Gebäude und Stockwerk**
Ein Gebäude ist aus einem oder mehreren Stockwerken aufgebaut. Ein Stockwerk gehört aber nur zu einem Gebäude. Daraus ergibt sich eine 1:n-Beziehung zwischen den Feldern `locrefBuilding.ID` und `locrefFloor.buildingID`.
- **Beziehung zwischen Stockwerk und Raum**
Ein Stockwerk besteht aus einem oder mehreren Räumen. Ein Raum ist aber nur einem Stockwerk zugeordnet. Daraus ergibt sich eine 1:n-Beziehung zwischen den Feldern `locrefFloor.ID` und `locrefRoom.floorID`.
- **Beziehung zwischen Raum und Anlagenobjekt**
In einem Raum können mehrere Anlagenobjekte installiert sein. Der Standort eines Gerätes kann aber durch nur einen Raum beschrieben sein. Daraus ergibt sich eine 1:n-Beziehung zwischen den Feldern `locrefRoom.ID` und `deviceObject.roomID`.

Abschließend ist noch einmal kritisch zu prüfen, ob die drei Normalformen im Datenbankentwurf konsequent eingehalten sind.

- Pro Datenfeld ist maximal ein Wert enthalten, d.h. alle Attribute weisen nur einfache Attributwerte auf und sind damit atomar. Dabei werden mehrere Wörter in einem Datenfeld als ein Wert vom Typ String betrachtet. Damit ist die 1. Normalform erfüllt.

- Die Relationen befinden sich in der 1. Normalform und sämtliche Nicht-Schlüssel-Attribute sind vom jeweiligen Primärschlüssel voll funktional abhängig. Der Anlagenkennzeichnungsschlüssel oder das Datum der Inbetriebnahme z. B. sind vom zusammengesetzten Primärschlüssel und damit dem Anlagenobjekt voll funktional abhängig. Damit ist die 2. Normalform erfüllt.
- Die Relationen befinden sich in der 2. Normalform und keines der Nicht-Schlüssel-Attribute folgt aus einem anderen Nicht-Schlüssel-Attribut. Dies könnte man zwar im Falle der Postleitzahl in Frage stellen, da aus diesem Attribut der Ort folgt. Doch global betrachtet kann diese PLZ in anderen Ländern für andere Orte stehen. Ebenso unkritisch ist der Anlagenkennzeichnungsschlüssel (identName) des Anlagenobjekts. Er setzt sich in der Praxis meist aus anderen Attributen zusammen, doch in der 3. Normalform geht es um die Ableitung aus einem einzelnen Nicht-Schlüssel-Attribut. Auch die 3. Normalform ist erfüllt.

6.2 XML-Schema für UPnP

Nach dem Entwurf der Datenbank sind die Dateien device.xml und scpd.xml zu erstellen. Die XML-Files basieren auf den Attributen aus Kapitel 5.5, die für einen neuen Standard empfohlen werden.

Für eine anschauliche Darstellung ist die folgende Ausführung auf das Beispiel einer Steuereinheit für Heizungsanlagen mit den dafür nötigen Attributen und Funktionen beschränkt. Die vollumfänglichen Templates für universelle Geräte der technischen Gebäudeausrüstung sind im Anhang dieser Arbeit zu finden.

6.2.1 device.xml

```
<?xml version="1.0" encoding="utf-8"?>
<root xmlns="urn:schemas-upnp-org:device-1-0">

  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>

  <device>
    <deviceType>urn:schemas-upnp-org:device:FMACodex:1</deviceType>
    <friendlyName>FMA.Codex Heizungssteuereinheit</friendlyName>
    <manufacturer>HotHeat GmbH</manufacturer>
    <manufacturerURL>http://www.hoheat.de</manufacturerURL>
    <modelDescription>Heizungssteuereinheit</modelDescription>

    <modelName>HeatingController</modelName>
    <modelName>HHHC 3000 Plus</modelName>
    <modelURL>http://www.hoheat.de/HC3000Plus</modelURL>
    <serialNumber>121001017</serialNumber>
    <UDN>uuid:0e547a38-a3a6-48df-9c2d-f4b75fe6e8c9</UDN>
    <UPC>UPnP.FMA.Codex.HC.3000</UPC>

    <categoryNumber>VDMA 24186-2</categoryNumber>
    <categoryName>Heizungstechnik</categoryName>
    <categoryDescription>Heizungssteuerung</categoryDescription>

    <mntType>Wartung</mntType>
    <mntInterval>12</mntInterval>
```

```

<mntIntervalTimeUnit>Monate</mntIntervalTimeUnit>
<mntProcedures>Wartung nach VDMA</mntProcedures>

<warranty>60 Monate</warranty>
<manufactureDate>2012-10-01</manufactureDate>

<deviceWidth>0.80</deviceWidth>
<deviceDepth>0.80</deviceDepth>
<deviceHeight>0.80</deviceHeight>
<volume>0.512</volume>
<weight>80</weight>
<footprint>0.64</footprint>

<serviceList>
  <service>
    <serviceType>urn:schemas-upnp-org:service:FMACodexHeating:1
    </serviceType>
    <serviceId>urn:upnp-org:serviceId:FMACodexHeating</serviceId>
    <SCPDURL>FMACodexHeatingService/scpd.xml</SCPDURL>
    <controlURL>FMACodexHeatingService/control</controlURL>
    <eventSubURL />
  </service>
</serviceList>

<presentationURL>/web</presentationURL>

</device>
</root> >

```

6.2.2 scpd.xml

```

<?xml version="1.0" encoding="utf-8"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">

  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>

  <actionList>

```

```
<action>
  <name>GetAirTemperatureInside</name>
  <argumentList>
    <argument>
      <name>retAirTemperatureInsideStatus</name>
      <direction>out</direction>
      <relatedStateVariable>AirTemperatureInside
        </relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>GetAirTemperatureOutside</name>
  <argumentList>
    <argument>
      <name>retAirTemperatureOutsideStatus</name>
      <direction>out</direction>
      <relatedStateVariable>AirTemperatureOutside
        </relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>GetBoilerPressure</name>
  <argumentList>
    <argument>
      <name>retBoilerPressureStatus</name>
      <direction>out</direction>
      <relatedStateVariable>BoilerPressure</relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>GetBoilerTemperature</name>
  <argumentList>
    <argument>
      <name>retBoilerTemperatureTarget</name>
      <direction>out</direction>
      <relatedStateVariable>BoilerTemperature
        </relatedStateVariable>
```

```
        </argument>
    </argumentList>
</action>

<action>
    <name>GetDevicePower</name>
    <argumentList>
        <argument>
            <name>retDevicePowerTarget</name>
            <direction>out</direction>
            <relatedStateVariable>Power</relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>GetDoorPosition</name>
    <argumentList>
        <argument>
            <name>retDoorPositionTarget</name>
            <direction>out</direction>
            <relatedStateVariable>DoorPosition</relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>GetGasConsumption</name>
    <argumentList>
        <argument>
            <name>retGasConsumptionStatus</name>
            <direction>out</direction>
            <relatedStateVariable>GasConsumption</relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>GetHeatingPower</name>
    <argumentList>
        <argument>
            <name>retHeatingPowerTarget</name>
            <direction>out</direction>
```

```
        <relatedStateVariable>HeatingBurner</relatedStateVariable>
    </argument>
</argumentList>
</action>

<action>
    <name>GetHeatLoss</name>
    <argumentList>
        <argument>
            <name>retHeatLossStatus</name>
            <direction>out</direction>
            <relatedStateVariable>HeatLoss</relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>GetHeatOutput</name>
    <argumentList>
        <argument>
            <name>retHeatOutputTarget</name>
            <direction>out</direction>
            <relatedStateVariable>HeatOutput</relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>GetOperatingPressure</name>
    <argumentList>
        <argument>
            <name>retOperatingPressureStatus</name>
            <direction>out</direction>
            <relatedStateVariable>OperatingPressure
                </relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>GetOperatingTemperature</name>
    <argumentList>
        <argument>
```

```
        <name>retOperatingTemperatureStatus</name>
        <direction>out</direction>
        <relatedStateVariable>OperatingTemperature
            </relatedStateVariable>
    </argument>
</argumentList>
</action>

<action>
    <name>GetPressureValvePosition</name>
    <argumentList>
        <argument>
            <name>retPressureValvePositionTarget</name>
            <direction>out</direction>
            <relatedStateVariable>PressureValvePosition
                </relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>GetWaterReservoirTemperature</name>
    <argumentList>
        <argument>
            <name>retWaterReservoirTemperatureTarget</name>
            <direction>out</direction>
            <relatedStateVariable>WaterReservoirTemperature
                </relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>GetWaterTemperatureReturn</name>
    <argumentList>
        <argument>
            <name>retWaterTemperatureReturnStatus</name>
            <direction>out</direction>
            <relatedStateVariable>WaterTemperatureReturn
                </relatedStateVariable>
        </argument>
    </argumentList>
</action>
```

```
<action>
  <name>GetWaterTemperatureSupply</name>
  <argumentList>
    <argument>
      <name>retWaterTemperatureSupplyStatus</name>
      <direction>out</direction>
      <relatedStateVariable>WaterTemperatureSupply
        </relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>GetWindowPosition</name>
  <argumentList>
    <argument>
      <name>retWindowPositionTarget</name>
      <direction>out</direction>
      <relatedStateVariable>WindowPosition</relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>SetBoilerTemperature</name>
  <argumentList>
    <argument>
      <name>newBoilerTemperatureTarget</name>
      <direction>in</direction>
      <relatedStateVariable>BoilerTemperature
        </relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>SetDevicePower</name>
  <argumentList>
    <argument>
      <name>newDevicePowerTarget</name>
      <direction>in</direction>
      <relatedStateVariable>Power</relatedStateVariable>
    </argument>
```

```
    </argumentList>
  </action>

  <action>
    <name>SetDoorPosition</name>
    <argumentList>
      <argument>
        <name>newDoorPositionTarget</name>
        <direction>in</direction>
        <relatedStateVariable>DoorPosition</relatedStateVariable>
      </argument>
    </argumentList>
  </action>

  <action>
    <name>SetHeatingPower</name>
    <argumentList>
      <argument>
        <name>newHeatingPowerTarget</name>
        <direction>in</direction>
        <relatedStateVariable>HeatingBurner</relatedStateVariable>
      </argument>
    </argumentList>
  </action>

  <action>
    <name>SetHeatOutput</name>
    <argumentList>
      <argument>
        <name>newHeatOutputTarget</name>
        <direction>in</direction>
        <relatedStateVariable>HeatOutput</relatedStateVariable>
      </argument>
    </argumentList>
  </action>

  <action>
    <name>SetPressureValvePosition</name>
    <argumentList>
      <argument>
        <name>newPressureValvePositionTarget</name>
        <direction>in</direction>
        <relatedStateVariable>PressureValvePosition
```

```
        </relatedStateVariable>
    </argument>
</argumentList>
</action>

<action>
  <name>SetWaterReservoirTemperature</name>
  <argumentList>
    <argument>
      <name>newWaterReservoirTemperatureTarget</name>
      <direction>in</direction>
      <relatedStateVariable>WaterReservoirTemperature
        </relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>SetWindowPosition</name>
  <argumentList>
    <argument>
      <name>newWindowPositionTarget</name>
      <direction>in</direction>
      <relatedStateVariable>WindowPosition</relatedStateVariable>
    </argument>
  </argumentList>
</action>
</actionList>

<serviceStateTable>
  <stateVariable sendEvents="no">
    <name>AirTemperatureInside</name>
    <dataType>ui4</dataType>
    <defaultValue>0</defaultValue>
  </stateVariable>

  <stateVariable sendEvents="no">
    <name>AirTemperatureOutside</name>
    <dataType>ui4</dataType>
    <defaultValue>0</defaultValue>
  </stateVariable>
</serviceStateTable>
```

```
<stateVariable sendEvents="no">
  <name>BoilerPressure</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>BoilerTemperature</name>
  <dataType>ui4</dataType>
  <defaultValue>75</defaultValue>
  <allowedValueRange>
    <minimum>40</minimum>
    <maximum>80</maximum>
  </allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>DoorPosition</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
  <allowedValueRange>
    <minimum>0</minimum>
    <maximum>10</maximum>
  </allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>GasConsumption</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>HeatingBurner</name>
  <dataType>boolean</dataType>
  <defaultValue>0</defaultValue>
  <allowedValueRange>
    <minimum>0</minimum>
    <maximum>1</maximum>
  </allowedValueRange>
</stateVariable>
```

```
<stateVariable sendEvents="no">
  <name>HeatLoss</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>HeatOutput</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>OperatingPressure</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>OperatingTemperature</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>Power</name>
  <dataType>boolean</dataType>
  <defaultValue>0</defaultValue>
  <allowedValueRange>
    <minimum>0</minimum>
    <maximum>1</maximum>
  </allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>PressureValvePosition</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
  <allowedValueRange>
    <minimum>0</minimum>
    <maximum>10</maximum>
  </allowedValueRange>
</stateVariable>
```

```
<stateVariable sendEvents="no">
  <name>WaterReservoirTemperature</name>
  <dataType>ui4</dataType>
  <defaultValue>60</defaultValue>
  <allowedValueRange>
    <minimum>40</minimum>
    <maximum>70</maximum>
  </allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>WaterTemperatureReturn</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>WaterTemperatureSupply</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>WindowPosition</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
  <allowedValueRange>
    <minimum>0</minimum>
    <maximum>10</maximum>
  </allowedValueRange>
</stateVariable>
</serviceStateTable>

</scpd>
```

6.3 Programmbaustein „FMControl“

Zur Demonstration der Anwendbarkeit des Modells wird ein webbasierter Programmbaustein mit Datenbankbindung unter Verwendung des neuen Standards „FMA.Codex“ entwickelt. Er trägt den Namen „FMControl“ und ist als Modul einer CAFM-Software für den Teilbereich des Anlagenmanagements zu verstehen. „FMControl“ basiert auf der eigens entworfenen Datenbank aus Kapitel 6.1. Die Anwendung beinhaltet eine Import- und Steuerfunktion für UPnP-Geräte der Gebäudeautomation, die zum jetzigen Zeitpunkt nur als virtuelle Devices simuliert werden können.

Für die Programmierung werden Visual Studio 2010 als Entwicklungsumgebung mit der Programmiersprache VB.NET sowie Microsoft Access als Datenbank(management)system verwendet. Die Bereitstellung der erzeugten Website wird durch den Windows-eigenen Internetinformationsdienst(IIS)-Manager 7.0 mit ASP.NET und dem Microsoft .NET-Framework 4 realisiert.

Die Anwendung „FMControl“ besteht aus mehreren Methoden (sog. Funktionen bzw. Prozeduren oder Events). In VB.NET wird wie folgt unterschieden: Funktionen geben einen Wert zurück, Prozeduren arbeiten eine Aufgabenkette ohne Rückgabewert ab. In anderen Programmiersprachen gibt es diese Unterscheidung nicht. Oftmals wird auch bei VB.NET der Begriff „Funktion“ für beide Methoden verwendet, so auch im weiteren Verlauf dieses Kapitels. Events sind durch bestimmte Ereignisse, wie z. B. Mausklicks oder Auswahländerungen, ausgelöste Funktionen. FMControl beinhaltet im Detail folgende Methoden:

Funktionen

- `dbConnection`
Verbindung zur Datenbank aufbauen
- `updateDropDownManufacturer`
Listbox Hersteller aktualisieren
- `updateDropDownModelNumber`
Listbox Modell-Nummer aktualisieren
- `updateDropDownSerialNumber`
Listbox Serien-Nummer aktualisieren
- `updateImportList`
Listbox Import aktualisieren

- `showSelection`
Attribute des ausgewählten Geräts anzeigen
- `clearFields`
Felder zurücksetzen
- `importData`
Ausgewähltes Gerät importieren
- `columnMatch`
Prüfen, ob XML-Eintrag als Feld/Spalte in der Datenbank vorhanden ist
- `buildSQL`
Zusammenbau des SQL-Strings
- `addSQL`
Ergänzung des SQL-Strings um Primärschlüssel und Relationen
- `insertSQL`
Erzeugung des Datenbankeintrags
- `showImport`
Importiertes Gerät auswählen und anzeigen
- `history`
Historisierung von Steuerbefehlen

Events

- `ListBoxManufacturer_SelectedIndexChanged`
Event nach Auswahl eines Herstellers
- `ListBoxModelNumber_SelectedIndexChanged`
Event nach Auswahl einer Modell-Nummer
- `ListBoxImportData_SelectedIndexChanged`
Event nach Auswahl eines zu importierenden Geräts
- `txtExistingDeviceChanged`
Event nach Textänderung in Attributfeldern vorhandener Geräte
- `txtNewDeviceChanged`
Event nach Textänderung in Attributfeldern neu anzulegender Geräte

- `btnSave_Click`
Event nach Klick des Buttons "Speichern"
- `btnNew_Click`
Event nach Klick des Buttons "Neu anlegen"
- `btnPowerOn_Click`
Event nach Klick des Buttons "Einschalten"
- `btnPowerOff_Click`
Event nach Klick des Buttons "Ausschalten"
- `btnMessage_Click`
Event nach Klick des Buttons "Meldung quittieren"
- `btnImport_Click`
Event nach Klick des Buttons "Import"

In den folgenden Unterkapiteln werden die Funktionen und Events des Programms mit auszugsweiser Darstellung des Programmiercodes erläutert.

6.3.1 Aufbau der Importfunktion

Die Importfunktion sieht vor, die zur Verfügung stehenden UPnP-Geräte per Mausklick anzubinden. Durch die Adressierung (via DHCP oder auch manuell) und Lokalisierung (automatisch) wird das Gerät erkannt und angezeigt. Die Beschreibung wird zusammen mit der Geräte-URL importiert, nachdem das zur Verfügung stehende Element aus einem DropDown-Menü selektiert wurde. Der Vorgang wird durch einen Klick auf den Button „Import starten“ ausgelöst.

Der vorgestellte Datenbankentwurf bietet die Grundlage für den Import. Da der Aufbau vorhandener CAFM-Datenbanken sehr unterschiedlich sein kann, muss die Importfunktion flexibel gestaltet sein und in der Praxis auf vorhandene Datenbanken abgestimmt werden. Flexibel heißt in diesem Zusammenhang, dass die Importfunktion auf die Datenbankstruktur ausgerichtet bzw. abgestimmt wird und nicht auf Namen oder Bezeichnungen.

Für die Demonstration wird dem Programmbaustein der Aufbau der Datenbank vorgegeben. Diese Vorgaben werden in einem zweidimensionalen Array (7x5)

festgehalten. 7 Tabellen und je 5 Einträge: Tabellename, Bezeichnung erster eigener Primärschlüssel, Wert erster Primärschlüssel (wird während der Laufzeit zugewiesen), Bezeichnung des ersten verlinkten Primärschlüssels, Bezeichnung des zweiten verlinkten Primärschlüssels.

Nun kommt der XML-Reader/-Parser zum Einsatz. Er muss die Informationen aus den XML-Dateien in 7 Schritten auslesen: 2 Dateien mit 2 bzw. 5 Ebenen. Die Ebenen sind nach den Vorgaben des UPnP-Forums in den Device und Service Templates definiert.

Für die Dateien device.xml und scpd.xml werden folgende Ebenen ausgelesen:

- root/device
- root/device/serviceList/service
- scpd/actionList/action
- scpd/actionList/action/argumentList
- scpd/actionList/action/argumentList/argument
- scpd/serviceStateTable/stateVariable
- scpd/serviceStateTable/stateVariable/allowedValueRange

Diese Angaben geben die Startpunkte des XML-Readers wieder. Die Funktionsweise des Readers lässt sich leicht erklären: Aus den zum Startpunkt gehörenden Einträgen (z. B. Aktionen) wird eine `xmlNodeList` erstellt. Jeder Eintrag stellt eine `xmlNode` dar und dessen jeweilige Untereinträge (z. B. Aktionsnamen) sind die `xmlChildNodes`, die vom Reader ausgelesen werden.

Konkret:

- device ist `xmlNode`, `friendlyName`, `manufacturer` etc. sind `xmlChildNodes`
- service ist `xmlNode`, `SCDPURL`, `controlURL` etc. sind `xmlChildNodes`
- action ist `xmlNode`, `actionNames` sind `xmlChildNode`
- argumentList ist `xmlNode`, `arguments` sind `xmlChildNode`
- argument ist `xmlNode`, `argumentNames` sind `xmlChildNode`
- stateVariable ist `xmlNode`, `stateVariableNames` sind `xmlChildNodes`
- allowedValueRange ist `xmlNode`, `min/max` etc. sind `xmlChildNodes`

Die Abfragen des Readers durchlaufen innerhalb der Funktion `importData()` verschiedene Schleifen je Tabelle und lesen die Werte (Values) der `ChildNodes` aus:

```
'Anzahl der Einträge/Spalten abrufen
Dim CountChildNodes = xmlNode.ChildNodes.Count

'Schleife über Anzahl der Einträge
For j = 0 To CountChildNodes - 1

    'Name und Wert auslesen
    Dim columnName = xmlNode.ChildNodes.Item(j).Name
    Dim valueName = xmlNode.ChildNodes.Item(j).InnerText

    [...]

Next
```

Sofern die Attributfelder in der entsprechenden Tabelle vorkommen, werden die ausgelesenen Werte für den Zusammenbau des SQL-Arguments übergeben. Diese Überprüfung ist in der Funktion `columnMatch()` programmiert:

```
'Prüfen, ob XML-Eintrag als Feld in der Datenbank vorhanden ist
Private Sub columnMatch(ByVal tableName As String, ByVal columnName As
String)

    'DataTable zum Zwischenspeichern der ausgelesenen Spalten
    Dim DataTable As DataTable
    'Variable zur Angabe, ob XML-Eintrag als Feld/Spalte vorhanden ist
    columnPresence = False

    Try
        'Füllen der DataTable
        DataTable = conn.GetOleDbSchemaTable(OleDbSchemaGuid.Columns,
New Object() {Nothing, Nothing, tableName, Nothing})
        'Listet die Spaltennamen aus jedem Eintrag der Schema Table auf

        For i = 0 To DataTable.Rows.Count - 1
            If columnName = DataTable.Rows(i)!COLUMN_NAME.ToString Then
                'Variable ist wahr, wenn XML-Eintrag Feld/Spalte der
                Datenbank identisch
                columnPresence = True
            End If
        Next i
    End Try
```

```

Catch ex As Exception
    'Ausgabe einer Meldung und Aktivierung des Buttons zum
    Quittieren
    lblMessage.Text = "Fehler in columnMatch(): " & ex.Message
    btnMessage.Enabled = True
    ' Variable zurücksetzen
    columnPresence = False
End Try
End Sub

```

Die ausgelesenen Werte werden dabei sukzessive zu einem SQL-String (der den Befehl zum anschließenden Datenbankeintrag vorhält) zusammengebaut, was durch die Funktion `buildSQL()` geschieht:

```

'Zusammenbau des SQL-Strings
Private Sub buildSQL(ByVal tableName As String, ByVal columnName As
String, ByVal valueName As String, ByVal i As Integer)

    'Zusammenbau des SQL-Statements
    If columnPresence Then

        'Prüfe, ob String leer ist oder bereits Einträge vorhanden sind
        If sqlInsertColumn = "" Then
            'SQL-Statement füllen
            sqlInsertColumn = columnName
            sqlInsertValue = "'" & valueName & "'"
        Else
            'SQL-Statement füllen
            sqlInsertColumn = sqlInsertColumn & ", " & columnName
            sqlInsertValue = sqlInsertValue & ", '" & valueName & "'"
        End If

        'Ausgabe einer Meldung und Aktivierung des Buttons zum
        Quittieren
        lblMessage.Text = "SQL-String erzeugt"
        btnMessage.Enabled = True
    End If
End Sub

```

Das Befüllen der Datenbankfelder alleine bildet noch nicht die logischen Zusammenhänge und Beziehungen der Tabellen untereinander ab. Es muss gewährleistet sein, dass entsprechende Einträge (z. B. von Hersteller und Modell) auch zueinander passen. Dazu müssen die Referenzfelder (hier die ID des jeweiligen Eintrags bzw. Datensatzes) abgerufen und zwischengespeichert werden. Sie sind im SQL-String als Referenzfeld zu ergänzen, was durch die Funktion `addSQL()` vollzogen wird.

```
'Ergänzung des SQL-Strings um Primärschlüssel und Relationen
Private Sub addSQL(ByVal tableName As String)

    [...]

    'ID der Relationsfelder auslesen
    Select Case primaryKey

        Case 1
            'Fall 1: die verlinkte Tabelle hat einen einfachen
            Primärschlüssel
            sqlString = "SELECT " & relatedTable & ".ID FROM " &
                relatedTable & " WHERE " & relatedField & "=" & relatedValue
                & ""

        Case 2
            'Fall 2: die verlinkte Tabelle hat einen zusammengesetzten
            Primärschlüssel
            sqlString = "SELECT " & relatedTable & ".ID, " & relatedTable
                & "." & relatedField & ", " & relatedTable2 & "." &
                relatedField2 & " FROM " & relatedTable2 & " INNER JOIN " &
                relatedTable & " ON " & relatedTable2 & ".ID = " &
                relatedTable & "." & linkField2 & " WHERE (((" & relatedTable
                & "." & relatedField & ")=" & relatedValue & "') AND ((" &
                relatedTable2 & "." & relatedField2 & ")=" & relatedValue2 &
                "'))"

        Case 3
            'Fall 3: die verlinkte Tabelle hat einen zusammengesetzten
            Primärschlüssel, der auf eine weitere Tabelle verlinkt
            sqlString = "SELECT " & relatedTable & ".ID, " & relatedTable
                & "." & relatedField & ", " & relatedTable2 & "." &
                relatedField2 & ", " & relatedTable3 & "." & relatedField3 & "
                FROM (" & relatedTable3 & " INNER JOIN " & relatedTable2 & "
```

```

ON " & relatedTable3 & ".ID = " & relatedTable2 & "." &
linkField3 & ") INNER JOIN " & relatedTable & " ON " &
relatedTable2 & ".ID = " & relatedTable & "." & linkField2 & "
WHERE ((((" & relatedTable & "." & relatedField & ")=" &
relatedValue & "') AND ((" & relatedTable2 & "." &
relatedField2 & ")=" & relatedValue2 & "') AND ((" &
relatedTable3 & "." & relatedField3 & ")=" & relatedValue3 &
"''))"

```

```
End Select
```

```
Try
```

```
'Schleife ggf. 2x durchlaufen, wenn Tabelle zwei Linkfelder
besitzt
```

```
For i = 0 To doubleLink
```

```
'Inhalt aus Datenbank auslesen
```

```
cmd.Connection = conn
```

```
cmd.CommandText = sqlString
```

```
dataReader = cmd.ExecuteReader()
```

```
If dataReader.Read() Then
```

```
If Not sqlInsertColumn = "" Then
```

```
sqlInsertColumn = sqlInsertColumn & ", " & linkField
```

```
sqlInsertValue = sqlInsertValue & ", " & dataReader(0)
```

```
End If
```

```
Else
```

```
'Ausgabe einer Meldung und Aktivierung des Buttons zum
Quittieren
```

```
lblMessage.Text = "Fehler bei der Datenbankabfrage."
```

```
btnMessage.Enabled = True
```

```
End If
```

```
'dataReader schließen
```

```
dataReader.Close()
```

```
'Ersetze Felder für zweiten Durchlauf
```

```
If doubleLink Then
```

```
sqlString = "SELECT " & relatedTable2 & ".ID FROM " &
```

```
relatedTable2 & " WHERE " & relatedField2 & "='" &
```

```
relatedValue2 & "'"
```

```
linkField = linkField2
```

```
End If
```

```
Next
```

```
    Catch ex As Exception
        'Ausgabe einer Meldung und Aktivierung des Buttons zum
        Quittieren
        lblMessage.Text = "Fehler in addSQL() " & ex.Message
        btnMessage.Enabled = True
    End Try
[...]
```

Schließlich kann der Eintrag eines Datensatzes über die Funktion `insertSQL()` je Tabelle ausgelöst werden:

```
'Datenbankeintrag erzeugen
Private Sub insertSQL(ByVal tableName As String)

    'Daten in Datenbank schreiben
    Try
        cmd.Connection = conn
        sqlInsert = "INSERT INTO " & tableName & " (" & sqlInsertColumn
        & ") VALUES (" & sqlInsertValue & ")"
        cmd.CommandText = sqlInsert
        cmd.ExecuteNonQuery()
        'Ausgabe einer Meldung und Aktivierung des Buttons zum
        Quittieren
        lblMessage.Text = "Datensatz erfolgreich eingefügt."
        btnMessage.Enabled = True
    Catch ex As Exception
        'Ausgabe einer Meldung und Aktivierung des Buttons zum
        Quittieren
        lblMessage.Text = "Fehler in insertSQL(): " & sqlInsert &
        ex.Message.ToString
        btnMessage.Enabled = True
    End Try
End Sub
```

Sämtliche Gerätedaten und Services sind nun in der Datenbank erfasst und können über die Benutzeroberfläche abgefragt werden. Die Attribute werden unmittelbar nach Abschluss des Imports durch die Funktion `showImport()` in den Feldern der Anlagenmaske ausgegeben.

6.3.2 Aufbau der Steuerfunktion

Die Funktionalitäten der Endgeräte sind durch die Service Templates beschrieben, die nach dem Import in der Datenbank festgehalten sind. Die Aktionen in der Datenbanktabelle „actionList“ definieren die Services. Jede Aktion beinhaltet ein Argument, dessen Wert ein Steuerbefehl ist, um Ereignisse auszulösen und damit den Betriebszustand aktiv regeln zu können.

Durch die Selektion einer Anlage (hier als DropDown-Menü umgesetzt) werden sämtliche Attribute aus der Datenbank ausgelesen und in den Textfeldern des Browserfensters dargestellt. Damit sind die verfügbaren Befehle geladen und können per Mausklick als HTTP-Request gesendet werden.

Als konkretes Beispiel sei die Aktion für das Ein- und Ausschalten einer Anlage erläutert. Die dafür vorgesehene Aktion trägt den Namen „SetDevicePower“. Mit ihr ist ein Steuerbefehl verknüpft, der über die definierten Relationen aus der Argumentliste abgerufen werden kann. Zum Steuerbefehl (argumentName) gehört der einzustellende Wert aus der serviceStateTable. Die entsprechende Variable nennt sich „Power“ und kennt zwei Zustände: Ein (True oder nach UPnP-Vorgabe 1) und Aus (False oder nach UPnP-Vorgabe 0). Um den Befehl senden zu können, muss die Geräteadresse (URL) ergänzt werden. Sie wird aus der Variable baseUrl des Anlagenobjekts und der controlURL der Anlagenklasse gebildet. Der zusammengesetzte Code hat also den Aufbau http:// & baseUrl & controlURL & argumentName & Value. Der Klick auf den Button „Einschalten“ löst die Aktion durch das Event btnPowerOn_Click() aus:

```
'Event nach Klick des Buttons "Einschalten"
Protected Sub btnPowerOn_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles btnPowerOn.Click

    'Prüfe, ob EnOcean-Gateway am COM-Port (nicht für den Standard
    vorgesehen, nur zu Demonstrationszwecken)
    If txtBaseUrl.Text = "COM-Port" Then

        'Steuerbefehl für Gerät am COM-Port zusammenstellen
        Dim command As String = txtMax.Text
        Dim n As Integer = command.Length \ 2
        Dim datagramm(n - 1) As Byte

        For i As Integer = 0 To command.Length - 1 Step 2
            datagramm(i \ 2) = Convert.ToByte(command.Substring(i, 2), 16)
        Next
```

```
Try
    'COM-Port öffnen
    Port.Open()
    'Command senden
    Port.Write(datagramm, 0, 14)
    'COM-Port schließen
    Port.Close()
    history(txtMax.Text)
Catch ex As Exception
    'Ausgabe einer Meldung und Aktivierung des Buttons zum
    Quittieren
    lblMessage.Text = "Steuerbefehl konnte nicht gesendet werden,
    Endgerät nicht erreichbar."
End Try

Else
    'Steuerbefehl für UPnP-Gerät zusammenstellen
    Dim command As String = "http://" & txtBaseUrl.Text & "/" &
    txtControlURL.Text & txtArgument.Text & txtMax.Text

    Try
        'Steuerbefehl senden
        Dim Request As HttpWebRequest = WebRequest.Create(command)
        Dim Response As HttpWebResponse = Request.GetResponse
        history(txtMax.Text)
    Catch ex As Exception
        'Ausgabe einer Meldung und Aktivierung des Buttons zum
        Quittieren
        lblMessage.Text = "Steuerbefehl konnte nicht gesendet werden,
        Endgerät nicht erreichbar."
    End Try

End If

End Sub
```

Die Funktion zum Abschalten nennt sich `btnPowerOff_Click()` und ist analog aufgebaut. Der Unterschied liegt in der Variablen, die beim Abschalten den Wert des vorgegebenen Minimums einnimmt.

Welche Buttons und Steuermöglichkeiten bereitgestellt werden, liegt im Ermessen des CAFM-Entwicklers. Grundsätzlich sollten die Buttons für sämtliche Set-/Get-Anweisungen vorgesehen werden. Der Prototyp beschränkt sich zu Demonstrationszwecken auf die Stromversorgung mit der Aktion „SetDevicePower“. Im Falle UPnP-fähiger Endgeräte wäre der zugehörige Steuerbefehl der Argument-Name „newDevicePowerTarget“ mit dem Variablenwert „1“ (Ein) oder „0“ (Aus).

Jede Aktion wird protokolliert und archiviert. Mit dieser Historisierung lässt sich eindeutig nachvollziehen, welches Gerät zu welchem Zeitpunkt welchen Zustand hatte. Hinsichtlich der Betreiberverantwortung ein sehr bedeutender Mehrwert. Dieses Feature ist in der Funktion `history()` abgebildet:

```
'Historisierung von Steuerbefehlen
Private Sub history(ByVal newValue As String)

    Try
        'Datenbankverbindung öffnen
        dbConnection()
        cmd.Connection = conn

        'stateVariableID und stateVariableValue auslesen
        Dim sqlString As String
        sqlString = "SELECT serviceStateTable.ID,
serviceStateTable.stateVariableValue,
serviceStateTable.stateVariableName, deviceObject.ID FROM
deviceObject INNER JOIN serviceStateTable ON deviceObject.ID =
serviceStateTable.deviceObjectID WHERE
(((serviceStateTable.stateVariableName)='Power') AND
((deviceObject.ID)=" & Me.ListBoxSerialNumber.SelectedValue &
"))"

        cmd.CommandText = sqlString
        dataReader = cmd.ExecuteReader()

        If dataReader.Read() Then
            If IsDBNull(dataReader(0)) Then
                stateVariableID = ""
            Else
                stateVariableID = dataReader(0)
            End If
            If IsDBNull(dataReader(1)) Then
                oldValue = "Gerätezustand nicht gesetzt"
```

```
Else
    oldValue = dataReader(1)
End If
Else
    'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
    lblMessage.Text = "Fehler bei der Datenbankabfrage."
    btnMessage.Enabled = True
End If
dataReader.Close()
'Aktion historisieren
sqlString = "INSERT INTO history (stateVariableID,
dateTimeStamp, oldValue, newValue) VALUES (" & stateVariableID &
", '" & DateAndTime.Now & "', '" & oldValue & "', '" & newValue
& "')"
cmd.CommandText = sqlString
cmd.ExecuteNonQuery()

'Variablenwert aktualisieren
sqlString = "UPDATE serviceStateTable SET stateVariableValue='"
& newValue & "' WHERE ID=" & stateVariableID
txtValue.Text = newValue
cmd.CommandText = sqlString
cmd.ExecuteNonQuery()

'Ausgabe einer Meldung und Aktivierung des Buttons zum
Quittieren
lblMessage.Text = "Steuerbefehl gesendet, Historisierung
erfolgreich."
btnMessage.Enabled = True

'dataReader schließen
dataReader.Close()
conn.Close()

Catch ex As Exception
    'Ausgabe einer Meldung und Aktivierung des Buttons zum
Quittieren
    lblMessage.Text = "Steuerbefehl gesendet, Fehler bei der
Historisierung."
    btnMessage.Enabled = True
End Try

End Sub
```

6.3.3 Vorbereitung der Simulation

Nach Abschluss der Programmierung wird der VB.NET-Code durch die Entwicklungsumgebung Visual Studio kompiliert und lokal publiziert. Die generierten ASP.NET Seiten (im Format .aspx) können im Browser über den „localhost“ aufgerufen werden. Dies eignet sich für die Testphase. Für den Betrieb in der Praxis bedarf es der Bereitstellung als „echte“ webbasierte Anwendung, die über das Internet (WAN) erreichbar ist. Hierfür ist die Installation und Konfiguration des Internetinformationsdienst(IIS)-Managers notwendig. Die Seiten werden im absoluten Verzeichnispfad C:\inetpub\wwwroot\ angelegt, für den entsprechende Zugriffsrechte freizugeben sind. Die vorliegende Anwendung findet sich im Unterverzeichnis „FMControl“ wieder. Die generierte Standardverzeichnisstruktur sieht einen Unterordner namens „App_Data“ für zugehörige Programmdateien vor. Dieser wird als Speicherort für die Datenbankdatei FMControl.mdb genutzt.

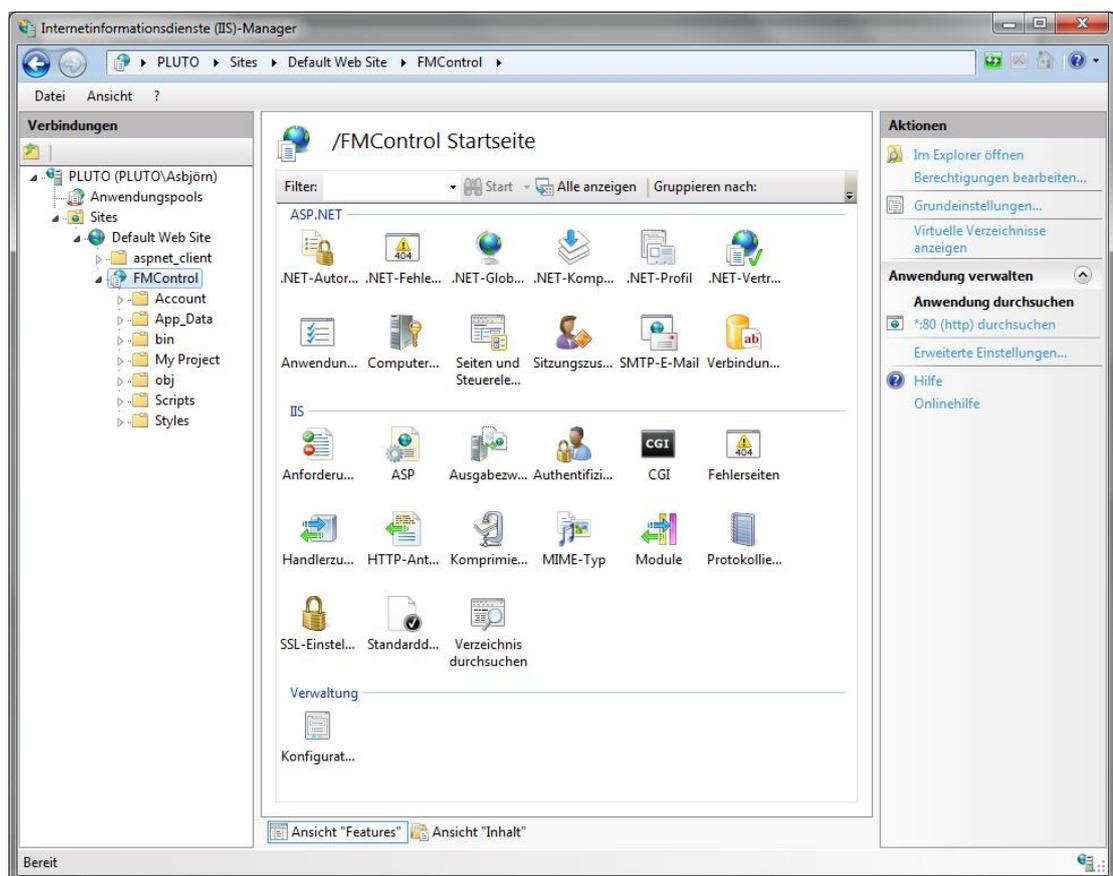


Abbildung 85: Internetinformationsdienst(IIS)-Manager, Screenshot aus eigener Konfiguration

Zum Zwecke der praktischen Erprobung kommen netzwerkfähige Steuerelemente zum Einsatz. Das Verhalten bei einer Netzwerkanbindung kann durch das Tool „UPnP

Developer Tools“ simuliert werden. Windows erkennt das virtuelle Gerät und zeigt es in der Netzwerkumgebung unter der IP-Adresse 127.0.0.1 an. Die Geräte- und Servicebeschreibungen sind analog zu den Ausführungen aus Kapitel 6.2 für den jeweiligen Anlagentyp aufgebaut. Sie liegen auf dem Server zum Import bereit, da die Testgeräte die Dateien nicht vorhalten können.

Die Demonstration erfolgt am Beispiel eines IP- aber nicht UPnP-fähigen Geräts, das die im Standard definierten Befehle nicht versteht. Hier werden der Argument-Name `newDevicePowerTarget` und die Variablenwerte 0 und 1 durch die geräte-spezifischen Commands ersetzt. Es handelt sich um eine IP-Steckdosenleiste der Firma Koukaam, Modell Netio -230B.



Abbildung 86: IP-Steckdosenleiste Netio-230B, Quelle: Online-Produktkatalog Fa. Koukaam

Trotz der nicht dem Modell entsprechenden Technologie, ist die Steuerung der bereitgestellten Ratio©-Geräte der Omnio AG im Programmbaustein implementiert. Es dokumentiert zum einen die Ergebnisse der Testphase aus Kapitel 4.1.6 und zeigt zum anderen die Ausbaufähigkeit zu einer technologieübergreifenden Plattform. Denn auch diese Geräte lassen sich in das Programm- und Datenbankschema einbinden.

6.4 Facility Automation durch Integration von GA und CAFM

Das letzte Unterkapitel der praktischen Umsetzung zeigt das Ergebnis der entstandenen, webbasierten Anwendung mit Datenbankanbindung zur Integration von Gebäudeautomation und CAFM-Systemen.

6.4.1 Benutzeroberfläche

Die Benutzeroberfläche von FMControl wird über den IIS-Manager bereitgestellt und ist über eine eigene URL erreichbar. Die Anwendung wird im Browser wie folgt angezeigt:

The screenshot shows the FMCONTROL web interface. At the top, there's a header with the product name and version information. Below that is a navigation menu with five tabs: 'Fläche', 'Gerät/Anlage', 'Wartung', 'Vertrag', and 'Firma'. The 'Gerät/Anlage' tab is selected. The main content area is divided into several sections:

- Gerät/Anlage:** This section has three columns: 'Auswählen', 'Anlegen', and 'Importieren'. Each column contains input fields for 'Hersteller', 'Modell-Nr.', and 'Serien-Nr.'. The 'Auswählen' column has a dropdown menu and an 'Auswählen' button. The 'Anlegen' column has a 'Neu anlegen' button. The 'Importieren' column has a dropdown menu, an input field for 'Geräteadresse', and an 'Import starten' button.
- Geräte-/Anlagenbeschreibung:** This section contains a grid of input fields for various attributes: 'Gerätename', 'Kategorie', 'Modell-Name', 'UDN', 'UPC', 'Baujahr', 'Gewährleistung', 'Inventarnummer', 'Kennzeichnung (AKS)', 'Inbetriebnahme', 'Anschaffungskosten', 'Standort Gebäude', 'Standort Stockwerk', and 'Standort Raum'. To the right of these fields are buttons for 'Speichern' and 'Abbrechen', along with the text 'Änderung Gerätedaten:'.
- Geräte-/Anlagensteuerung:** This section contains input fields for 'Steuerungsadresse', 'Funktion', and 'Steuerbefehl'. It also has input fields for 'Aktueller Wert (Ein/Aus)', 'Maximawert / Ein', and 'Minimalwert / Aus'. To the right are buttons for 'Einschalten', 'Ausschalten', and 'Meldung quittieren', along with the text 'Änderung Gerätestatus:'.

At the bottom left, there is a 'Meldung:' field with the value 'Bereit'.

Abbildung 87: Webbasierte Benutzeroberfläche von "FMControl", Screenshot der eigenen Entwicklung

Der prinzipielle Aufbau sieht vor, die Prozesse des Flächen-, Anlagen-/Wartungs- und Vertragsmanagements sowie eine Kontaktdatenverwaltung abbilden zu können. Dafür sind die Menübereiche „Fläche“, „Gerät/Anlage“, „Wartung“, „Vertrag“ und „Firma“ angelegt. Der Prototyp beschränkt sich auf den Bereich des Anlagenmanagements mit Importfunktion und Methoden zur Anlagensteuerung.

Die Maske „Gerät/Anlage“ ist in drei Bereiche gegliedert: den Kopfbereich für die Geräteauswahl oder –erfassung, den Informationsbereich mit der Anzeige von Geräteeigenschaften und den Steuerbereich mit Serviceattributen und Aktionsbuttons.

Im Kopfbereich stehen drei Varianten zur Verfügung: die Auswahl eines vorhandenen Geräts, das manuelle Anlegen eines neuen Geräts oder der Import eines via UPnP angebundenes Geräts.

Soll ein vorhandenes Gerät ausgewählt werden, gibt es die Möglichkeit, nach Hersteller und/oder Modell zu filtern. Entsprechend reduziert sich die Auswahl der untergeordneten DropDown-Menüs. Nach Auswahl eines Anlagenobjekts durch die Seriennummer, füllt sich der Informations- und Steuerbereich mit den zugehörigen Attributwerten.

Sofern ein neues Gerät manuell angelegt werden soll, ist die Angabe von Hersteller, Modell-Nr. und Serien-Nr. in den entsprechenden Feldern obligatorisch. Nur die Kombination dieser Angaben ergibt einen eindeutigen Eintrag eines Anlagenobjekts. Sind alle drei Felder ausgefüllt, wird der Button „Neu anlegen“ aktiviert. Ein Klick löst den Datenbankeintrag aus. Anschließend kann die Anlagenbeschreibung durch Ausfüllen der entsprechenden Felder vervollständigt werden.

Sind UPnP-fähige Geräte, die dem FMA.Codex entsprechen, im Netzwerk zu finden, werden sie im DropDown-Menü im Bereich Importieren angezeigt. Nach der Auswahl eines Objekts füllt sich das Feld baseURL automatisch. Dieses Feld wird zusammen mit den Attributen der Geräte- und Servicebeschreibung importiert.

Der Informationsbereich dient der Geräte- und Anlagenbeschreibung. Die angezeigten Felder sind nur eine kleine Auswahl verfügbarer Attribute, die problemlos ergänzt werden kann. Die Felder der linken Seite sind nicht editierbar, da es um Herstellervorgaben handelt. Die Felder der rechten Seite spiegeln die benutzerdefinierten Angaben wider (vgl. Tabelle 8).

Der Steuerbereich beinhaltet die vom Hersteller vorgegebenen Geräte- und Anlagenservices, deren Eigenschaften ebenfalls nicht editiert werden können. Die Demoversion beschränkt sich auf die Ein- und Ausschaltfunktion eines Geräts. Sind die entsprechenden Attributwerte zur Beschreibung des Anlagenzustandes vorhanden, aktivieren sich die Buttons zur Steuerung.

6.4.2 Anlagenimport

Einem Import der Device und Service Descriptions geht die Erkennung der UPnP-Geräte im Netzwerk voraus. Dies lässt sich mit Hilfe der „UPnP Developer Tools“ simulieren. In der Windows Netzwerkumgebung werden sie als neues Gerät dargestellt. Über einen Rechtsklick kann man sich einige Geräteeigenschaften anzeigen lassen.

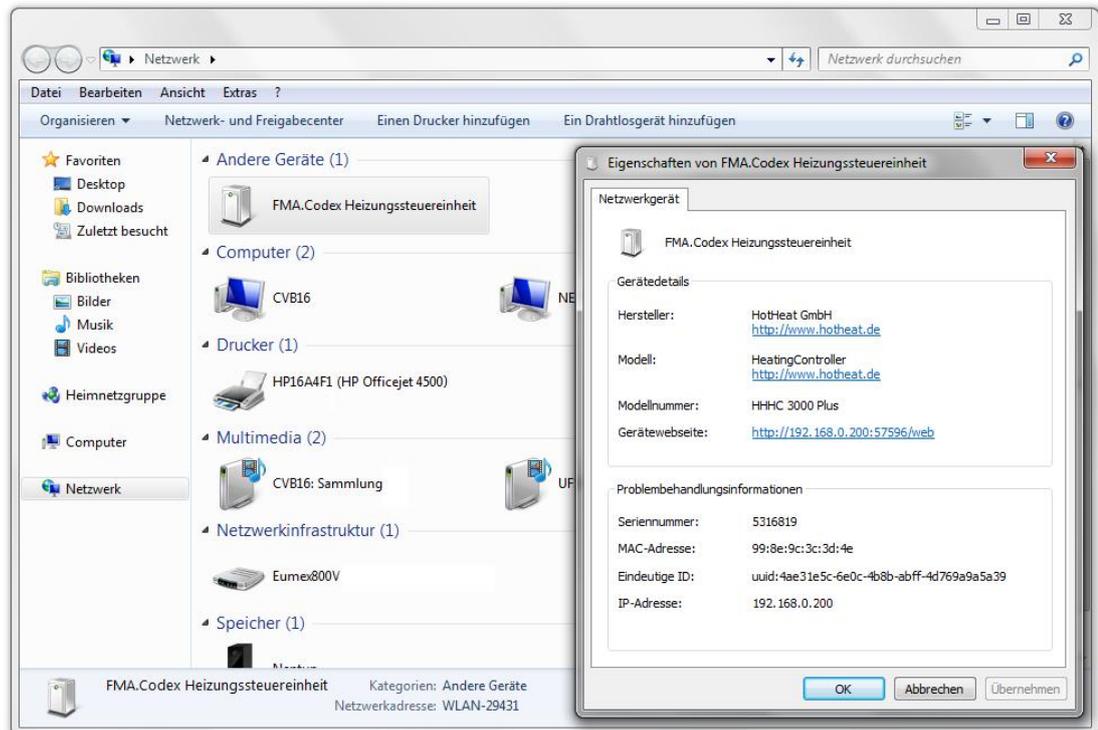


Abbildung 88: Registrierung eines neuen Geräts, Screenshot der eigenen Netzwerkumgebung

Sämtliche UPnP-Geräte, die dem FMA.Codex entsprechen, sind in einem DropDown-Menü der Anwendungsoberfläche erfasst.

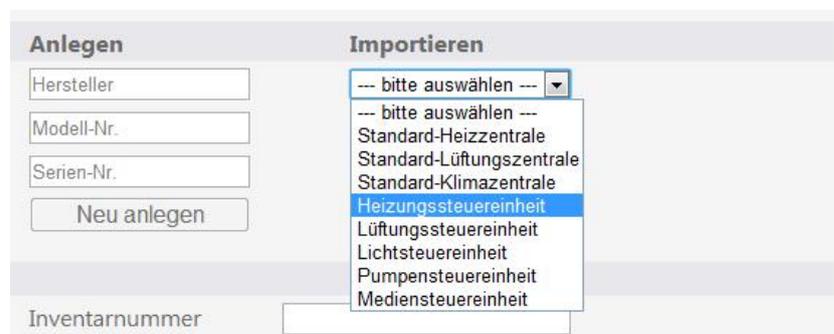


Abbildung 89: Auswahlmenü für Import, Screenshot der eigenen Entwicklung

Der Prototyp beinhaltet eine vordefinierte Auswahl. Eine nicht implementierte, aber empfohlene Routine sollte prüfen, ob ein Gerät bereits importiert wurde, um dieses aus der Liste zu filtern. Durch Selektion eines der zur Verfügung stehenden Elemente wird die aktuelle Geräte-URL ausgelesen und dargestellt. Gleichzeitig wird der Import-Button aktiviert.

Durch Klick auf „Import starten“ wird der Importvorgang ausgelöst und durchläuft die Funktionen dbConnection, btnImport_Click, importData, columnMatch, buildSQL, addSQL, insertSQL, showImport und showSelection. Darin werden die Einträge der XML-Dateien in mehreren Schleifen ausgelesen und in die Datenbank übertragen. Anschließend wird das neu importierte Gerät automatisch ausgewählt und die Informationen dazu werden abgerufen. Das Ergebnis erscheint direkt im Browser:

The screenshot shows the FMCONTROL web interface with the 'Gerät/Anlage' tab selected. The interface is divided into several sections:

- Navigation:** Tabs for 'Fläche', 'Gerät/Anlage', 'Wartung', 'Vertrag', and 'Firma'.
- Gerät/Anlage Section:**
 - Auswählen:** Fields for 'Hersteller' (HotHeat GmbH), 'Modell-Nr.' (HHHC 3000 Plus), and 'Serien-Nr.' (121001017) with an 'Auswählen' button.
 - Anlegen:** Fields for 'Hersteller', 'Modell-Nr.', and 'Serien-Nr.' with a 'Neu anlegen' button.
 - Importieren:** Fields for 'Heizungssteuereinheit' (dropdown) and '192.168.0.153' with an 'Import starten' button.
- Geräte-/Anlagenbeschreibung:**
 - Fields for 'Gerätename' (FMA.Codex Heizungssteuereinheit), 'Kategorie' (Heiztechnik), 'Modell-Name' (HeatingController), 'UDN' (uuid:0e547a38-a3a6-48df-9c2d-f4), 'UPC' (UPnP.FMA.Codex.HC.3000), 'Baujahr' (2012-10-01), and 'Gewährleistung' (60 Monate).
 - Fields for 'Inventarnummer', 'Kennzeichnung (AKS)', 'Inbetriebnahme', 'Anschaffungskosten', 'Standort Gebäude', 'Standort Stockwerk', and 'Standort Raum', all currently set to 'k.A.'.
 - Buttons for 'Speichern' and 'Abbrechen'.
- Geräte-/Anlagensteuerung:**
 - Fields for 'Steuerungsadresse' (FMACodexHeatingService/contro), 'Funktion' (SetDevicePower), and 'Steuerbefehl' (newDevicePowerTarget).
 - Fields for 'Aktueller Wert (Ein/Aus)' (k.A.), 'Maximawert / Ein' (1), and 'Minimalwert / Aus' (0).
 - Buttons for 'Einschalten' and 'Ausschalten'.
- Meldung:** A message box stating 'Datensatz erfolgreich eingefügt.' with a 'Meldung quittieren' button.

Abbildung 90: Anlagenmaske nach Import, Screenshot der eigenen Entwicklung

Die Felder der benutzerdefinierten Attribute sind vorerst leer (Angabe „k.A.“), aber zur Anpassung editierbar. Für jedes Objekt kann eine Inventar-Nr. oder ein Anlagenkennzeichnungsschlüssel vergeben werden. Der Aufbau des AKS z. B. ist vom

Anwender frei definierbar. Oftmals setzt er sich zusammen aus Gewerkekürzel, Raumbezug und fortlaufender Nr. für den Fall, dass mehrere Geräte des gleichen Typs in einem Raum verbaut sind. So würde beispielsweise der Schlüssel „H.G1401101.1“ eine Heizzentrale in Gebäude 14, 1. Stock, Raum 101 kennzeichnen.

Was ist hinsichtlich der Datenbankeinträge passiert? Zur Überprüfung der ordnungsgemäßen Ergänzung der Datensätze durch die SQL-Anweisungen richten wir den Blick auf die Datenbankinhalte in MS Access.

ID	stateVariableName	stateVariableValue	dataType	defaultValue	minimum	maximum
612	WindowPosition	0	ui4	0	0	10
611	WaterTemperatureSupply	0	ui4	0		
610	WaterTemperatureReturn	0	ui4	0		
609	WaterReservoirTemperature	60	ui4	60	40	70
608	PressureValvePosition	0	ui4	0	0	10
607	Power	0	boolean	0	0	1
606	OperatingTemperature	0	ui4	0		
605	OperatingPressure	0	ui4	0		
604	HeatOutput	0	ui4	0		
603	HeatLoss	0	ui4	0		
602	HeatingBurner	0	boolean	0	0	1
601	GasConsumption	0	ui4	0		
600	DoorPosition	0	ui4	0	0	10
599	BoilerTemperature	75	ui4	75	40	80
598	BoilerPressure	0	ui4	0		
597	AirTemperatureOutside	0	ui4	0		
596	AirTemperatureInside	0	ui4	0		

Abbildung 91: Datenbankeinträge, Screenshot der eigenen Datenbank

Der Screenshot zeigt die Tabelle deviceObject in der Datenblattansicht. Je Zeile findet sich ein spezifisches Gerät wieder. Die Spalten geben die Attribute an, deren Werte in den Feldern stehen. Dieses Beispiel zeigt die Auswahl des Objekts mit der ID 114, das durch die Kombination der Primärschlüssel (Serien-Nummer „121001023“ und Anlagenklasse mit der ID 65, hier Modell „HHHC 4000 Plus“) eindeutig gekennzeichnet ist. Die nach unten geöffnete Verzweigung (orange

gekennzeichneter Bereich) verweist auf die dem Gerät zugeordneten Service-Variablen. Hier wird die Leistung der Importfunktion hinsichtlich übertragener Relationen deutlich. Die in der Ansicht zu sehenden Variablennamen sind nicht einmalig, sie wiederholen sich je Gerät. Hier ist es die Kombination aus den Primärschlüsseln Name und ID des Anlagenobjekts (in diesem Beispiel 114), was die Eindeutigkeit des Eintrags gewährleistet.

6.4.3 Anlagensteuerung

Abschließend wird der technisch innovative und dennoch durch wenige Mausklicks sehr einfach zu bedienende Steuerungsprozess vorgestellt.

Die vom Hersteller vorgegebenen Eigenschaften zur Bedienung eines Geräts werden im unteren Bereich des Fensters in nicht editierbaren Feldern angezeigt. Die Steuerungsadresse gibt die URL wieder, an die der Steuerbefehl einer Funktion gesendet wird. Der Prototyp ist hinsichtlich des eingesetzten Schaltaktors auf die Funktion „SetPowerDevice“ mit dem Steuerbefehl „newDevicePowerTarget“ limitiert. Sofern in Zukunft UPnP-fähige Geräte zur Gebäudeautomation am Markt erhältlich sind, ist das Feld „Funktion“ als DropDown-Menü zur Auswahl der gerätespezifischen Funktionen bereitzustellen. Die Felder der rechten Seite beinhalten den aktuellen Anlagenstatus, sowie den Minimal- und Maximalwert. Sind funktionsabhängig Zwischenwerte erlaubt, muss ein weiteres Feld für die Eingabe des Soll-Wertes bereitgestellt werden. Ein Klick auf den Button Einschalten oder Ausschalten löst die Aktion aus und protokolliert den Vorgang.

Geräte-/Anlagensteuerung			
Steuerungsadresse	FMACodexHeatingService/control	Aktueller Wert (Ein/Aus)	k.A.
Funktion	SetDevicePower	Maximawert / Ein	1
Steuerbefehl	newDevicePowerTarget	Minimalwert / Aus	0
			Änderung Gerätestatus:
			<input type="button" value="Einschalten"/>
			<input type="button" value="Ausschalten"/>
Meldung:	Datensatz erfolgreich eingefügt.		<input type="button" value="Meldung quittieren"/>

Abbildung 92: Maskenbereich Geräte-/Anlagensteuerung, Screenshot der eigenen Entwicklung

Die Protokollierung dient der transparenten Dokumentation von Änderungen des Gerätezustandes. Erfasst werden der alte Zustandswert (oldValue) und neue Zustandswert (newValue) mit Zeitstempel (dateTimeStamp) und Verweis zur entsprechenden Variable (stateVariableID).

Hinweis zur folgenden Grafik: Da das zur Simulation verwendete, nicht UPnP-fähige Gerät die Standardwerte 0 und 1 für die Zustandsvariable nicht interpretieren kann, mussten die geräteabhängigen Werte verwendet werden.

ID	stateVariableID	dateTimeStamp	oldValue	newValue
30	624	19.03.2013 14:07:27	?l=p:control:go&p=uu0u&p=l&q=q	?l=p:control:go&p=uu1u&p=l&q=q
29	624	19.03.2013 14:07:23	?l=p:control:go&p=uu1u&p=l&q=q	?l=p:control:go&p=uu0u&p=l&q=q
28	624	19.03.2013 14:07:17	?l=p:control:go&p=uu0u&p=l&q=q	?l=p:control:go&p=uu1u&p=l&q=q
27	1985	19.03.2013 14:03:51	A55A6B0510000000FFFFFFFF81302E	A55A6B0530000000FFFFFFFF81304E
26	1985	19.03.2013 14:03:47	A55A6B0530000000FFFFFFFF81304E	A55A6B0510000000FFFFFFFF81302E
25	624	19.03.2013 13:34:42	?l=p:control:go&p=uu1u&p=l&q=q	?l=p:control:go&p=uu0u&p=l&q=q
24	624	19.03.2013 13:34:37	?l=p:control:go&p=uu0u&p=l&q=q	?l=p:control:go&p=uu1u&p=l&q=q
23	1985	19.03.2013 11:45:01	A55A6B0510000000FFFFFFFF81302E	A55A6B0530000000FFFFFFFF81304E
21	624	19.03.2013 03:53:36	?l=p:control:go&p=uu1u&p=l&q=q	?l=p:control:go&p=uu0u&p=l&q=q
20	624	19.03.2013 03:53:30	Gerätezustand nicht gesetzt	?l=p:control:go&p=uu1u&p=l&q=q
19	505	19.03.2013 03:50:53	?l=p:control:go&p=1uuu&p=l&q=q	?l=p:control:go&p=0uuu&p=l&q=q
18	505	19.03.2013 03:50:35	Gerätezustand nicht gesetzt	?l=p:control:go&p=1uuu&p=l&q=q
17	1985	19.03.2013 01:55:58	A55A6B0530000000FFFFFFFF81304E	A55A6B0510000000FFFFFFFF81302E
16	1985	19.03.2013 01:55:54	A55A6B0510000000FFFFFFFF81302E	A55A6B0530000000FFFFFFFF81304E
15	1985	19.03.2013 01:55:29	Gerätezustand nicht gesetzt	A55A6B0510000000FFFFFFFF81302E
* (Neu)				

Abbildung 93: Datensätze zur Historisierung, Screenshot der eigenen Datenbank

Jeder Steuerungsvorgang wird erfasst und als Datenbankeintrag gespeichert. So können die Vorgänge bzw. Anlagenzustände beliebig gefiltert und ausgewertet werden, um den Status einer Anlage zu einem bestimmten Zeitpunkt festzustellen. Dies kann im Rahmen der Betreiberverantwortung oder bei Schadensfällen von höchster Bedeutung sein.

7 Fazit und Ausblick

Im letzten Kapitel dieser Arbeit wird das Ergebnis unter Betrachtung der Zielsetzung bewertet und der Nutzen aus FM-Sicht dargestellt. Die Dissertation schließt mit Handlungsempfehlungen für Weiterentwicklungen und einem Ausblick für den Einsatz in der Praxis.

7.1 Fazit dieser Arbeit

Das Ziel dieser Arbeit war, einen Standard zur Kopplung von CAFM-Systemen und Gebäudeautomationstechnologien auf Management- und Automationsebene unter Verwendung eines reinen Computernetzwerks zu entwickeln. Die Anforderung an das webbasierte Lösungsmodell mit Datenbankanbindung war die komfortable Anbindung gebäudetechnischer Anlagen nach dem „Plug and Play“ Prinzip mit CAFM-seitiger Steuerungsmöglichkeit. Die Funktionalität des Integrationsmodells war durch die praktische Umsetzung in Form einer Weboberfläche mit Import- und historisierter Steuerungsfunktion zu demonstrieren.

Das entwickelte Lösungskonzept, der entworfene Standard, das Datenbankmodell und der Programmbaustein erfüllen alle genannten Anforderungskriterien. Das Ziel wurde erreicht.

Die methodische und analytische Vorgehensweise bei der Modellentwicklung, die Bewertung anhand maßgebender Anforderungskriterien mit Hilfe einer Entscheidungsmatrix sowie der Aufbau des Datenbankmodells gehören zu den wissenschaftlichen Elementen der Arbeit. Der für eine Standardisierung notwendige Entwurf der XML Service und Device Descriptions sowie die Anwendungs- und Schnittstellenprogrammierung mit VB.NET unter der Entwicklungsumgebung Visual Studio sind als Ingenieurleistungen zu betrachten.

Eine besondere Herausforderung war die Interdisziplinarität der Aufgabenstellung mit fachspezifischer CAFM-Ausrichtung. Die Lösungsstrategie erforderte die Betrachtung von Methoden und Standards der Bereiche E-Technik, Informatik und Facility Management. Nach dem konzeptionellen Entwurf des Integrationsmodells bestand die Aufgabe darin, vorhandene Technologien und Standards zu nutzen, um daraus neue Lösungen zu schaffen. Der Schlüssel und zugleich neuartige Ansatz ist die innovative Kombination aus Gebäude-, CAFM- und UPnP-Technologie.

Der größte Nutzen aus FM-Perspektive ist die Funktionserweiterung von CAFM-Systemen durch die Import- und Steuerungsmechanismen für gebäudetechnische Einrichtungen. Der Ausbau zu einem leistungsstarken Steuerungstool ist ein bedeutender Mehrwert für ein aktives Anlagenmanagement. So kann z. B. im Rahmen des Energiecontrollings direkt Einfluss auf Prozesse genommen werden.

Nach Angaben der CAFM-Trendstudie 2013⁹³ planen ca. 25 % der Unternehmen eine Erweiterung der CAFM-Systeme auf weitere Bereiche, wie z. B. das Energiecontrolling. Mehr als 30 % haben bereits Datenschnittstellen zur Gebäudeautomation. Noch ist das Bewusstsein für den Einsatz einer vollständigen Systemintegration aber nicht geschärft. Das Prinzip von Angebot und Nachfrage kommt hier zum Tragen. Ohne Standard wird die Entwicklung nicht voranschreiten. Der erste Schritt ist getan. Wird der vorgestellte FMA.Codex eingeführt, haben Programmierer und Gerätehersteller eine Grundlage für die Entwicklung UPnP-fähiger Bauteile bzw. CAFM-Systeme, um die nötigen Dienste, Protokolle und Datenformate implementieren zu können. Ist das Angebot erst einmal da, wird die Nachfrage steigen.

7.2 Ausblick

Das Ergebnis dieser Arbeit bietet eine solide Grundlage für die Neuausrichtung der Entwicklung von gebäudetechnischen Geräten und Anlagen. Doch welche Schritte sind für die Umsetzung zum effektiven Einsatz in der Praxis notwendig?

Es ist beabsichtigt, die entwickelten Device und Service Descriptions dem UPnP-Forum zwecks Aufnahme in das UPnP Service Schema für einen neuen Bereich „Facility Automation“ vorzulegen. Gegebenenfalls sind die Templates für einzelne Gewerke zu trennen, da hier eine universelle, gewerkeübergreifende Vorlage erarbeitet wurde. Ein weiteres Ziel ist es, dem CAFM-Arbeitskreis der GEFMA den FMA.Codex vorzustellen und in der Branche zu etablieren.

Hinsichtlich des Programmbausteins „FMControl“ gibt es einige Empfehlungen der Weiterentwicklung. Die Auswahl der importierbaren UPnP-Geräte im DropDown-Menü ist im Prototyp zur Simulation fest vorgegeben. Diese sollte einer automatisierten Erkennungsroutine gemäß UPnP-Standard folgen. Des Weiteren bietet der Prototyp nur

⁹³ (Hohmann, Fachvortrag CAFM-Trendstudie, 2013)

die Funktion „SetDevicePower“ zum Ein- und Ausschalten des Mustergeräts an. Hier sollte es eine Auswahlmöglichkeit sämtlicher, durch die neuen UPnP-Geräte zur Verfügung gestellter Dienste geben, mit entsprechenden Buttons zur Steuerung.

Die Vision ist die CAFM-seitige Steuerung sämtlicher elektrischen Verbraucher (wie z. B. Lichtquellen, Jalousien, Pumpen, Ventilatoren oder auch Multimedia-Geräte), die Regelung des Raumklimas (HKL mit Klimaplan, Kalender-/Timerfunktion etc.) sowie die Kontrolle und Überwachung (unter Einsatz der Sicherheitstechnik mit Kameras, Meldern etc.).

Die in Kapitel 6.1 entwickelte Datenbank ist hinsichtlich der vorgehaltenen Attributfelder und Namensgebung auf die Vorgaben der Service und Device Descriptions abgestimmt. Zur Integration in vorhandene CAFM-Systeme müssen deren Datenbankfelder ergänzt werden und ggf. Übersetzungsroutinen für die Datenbankimporte generiert werden.

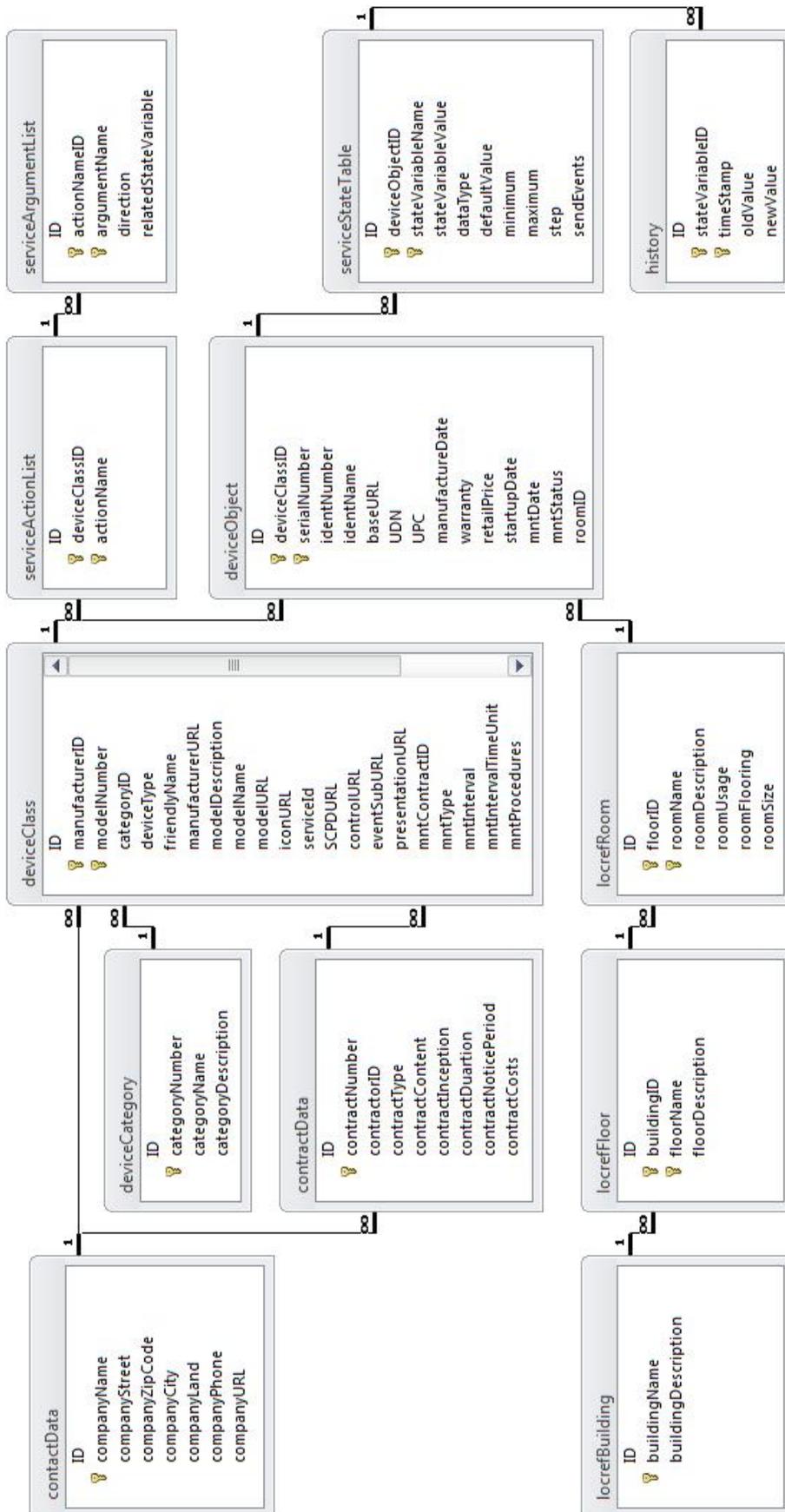
In Bezug auf die Kategorisierung ist es sehr zu empfehlen, verbindliche Vorgaben für die Benennung der Kategorien zu treffen. Dabei sollte man sich an den Katalogen des Verbandes Deutscher Maschinen- und Anlagenbau e.V. (VDMA), des Arbeitskreises Maschinen- und Elektrotechnik staatlicher und kommunaler Verwaltungen (AMEV) oder der DIN 276 (Kostengruppe 400) orientieren. Davon sollten die Bezeichnungen für das Feld „categoryName“ im UPnP-Gerätestandard nicht abweichen.

Schließlich sind die Gerätehersteller aufgerufen, UPnP-fähige Komponenten zu entwickeln und auf den Markt zu bringen. Diese Arbeit zeigt durch die Realisierung des Prototyps „FMControl“, dass die softwareseitige und datenbankgestützte Integration und Steuerung möglich ist.

Langfristig sollte die Weiterentwicklung in Richtung Komfort, Sicherheit und Energieeinsparung vorangetrieben werden, in erster Linie hardwareseitig. Das Ziel muss sein, den FMA.Codex nicht nur als technische Innovation, sondern auch als wirtschaftlich effektive, effiziente und nachhaltige Technologie zu vermarkten. Gütesiegel und Zertifikate, wie z. B. DGNB, LEED oder BREEAM, runden das Gesamtpaket künftig ab.

I. Anhang

I.a Struktur der relationalen Datenbank, Tabellenbeziehungen



I.b FMA.Codex Device Description device.xml

```
<?xml version="1.0" encoding="utf-8"?>
<root xmlns="urn:schemas-upnp-org:device-1-0">

  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>

  <device>
    <deviceType>urn:schemas-upnp-org:device:FMACodex:1</deviceType>
    <friendlyName>FMA.Codex Steuereinheit</friendlyName>
    <manufacturer>Hersteller</manufacturer>
    <manufacturerURL>URL des Herstellers</manufacturerURL>
    <modelDescription>FMA.Codex Modellbeschreibung</modelDescription>
    <modelName>FMA.Codex Modellname</modelName>
    <modelName>FMA.Codex Modellname</modelName>
    <modelNumber>FMA.Codex Modellnummer</modelNumber>
    <modelURL>URL des Modells</modelURL>
    <serialNumber>Seriennummer</serialNumber>
    <UDN>uuid</UDN>
    <UPC>FMA.Codex Produktcode</UPC>

    <categoryNumber>Kategorienummer</categoryNumber>
    <categoryName>Kategoriebezeichnung</categoryName>
    <categoryDescription>Kategoriebeschreibung</categoryDescription>

    <mntType>Wartungstyp</mntType>
    <mntInterval>x</mntInterval>
    <mntIntervalTimeUnit>Monate</mntIntervalTimeUnit>
    <mntProcedures>Wartungstätigkeiten</mntProcedures>

    <warranty>x Monate</warranty>
    <manufactureDate>yyyy-mm-dd</manufactureDate>

    <deviceLength>Länge</deviceLength>
    <deviceWidth>Breite</deviceWidth>
    <deviceDepth>Tiefe</deviceDepth>
    <deviceHeight>Höhe</deviceHeight>
    <volume>Volumen</volume>
    <weight>Gewicht</weight>
    <footprint>Standfläche</footprint>
```

```
<iconList>
  <icon>
    <mimetype>image/png</mimetype>
    <width>48</width>
    <height>48</height>
    <depth>32</depth>
    <url>/icon.png</url>
  </icon>
</iconList>

<serviceList>
  <service>
    <serviceType>urn:schemas-upnp-org:service:FMACodex:1
    </serviceType>
    <serviceId>urn:upnp-org:serviceId:FMACodex.1</serviceId>
    <SCPDURL>FMACodexService/scpd.xml</SCPDURL>
    <controlURL>FMACodexService/control</controlURL>
    <eventSubURL />
  </service>
</serviceList>

<presentationURL>/web</presentationURL>

</device>
</root>
```

I.c FMA.Codex Service Description scpd.xml

```
<?xml version="1.0" encoding="utf-8"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">

  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>

  <actionList>

    <action>
      <name>GetAirTemperatureInside</name>
      <argumentList>
        <argument>
          <name>retAirTemperatureInsideStatus</name>
          <direction>out</direction>
          <relatedStateVariable>AirTemperatureInside
            </relatedStateVariable>
        </argument>
      </argumentList>
    </action>

    <action>
      <name>GetAirTemperatureOutside</name>
      <argumentList>
        <argument>
          <name>retAirTemperatureOutsideStatus</name>
          <direction>out</direction>
          <relatedStateVariable>AirTemperatureOutside
            </relatedStateVariable>
        </argument>
      </argumentList>
    </action>

    <action>
      <name>GetBoilerPressure</name>
      <argumentList>
        <argument>
          <name>retBoilerPressureStatus</name>
          <direction>out</direction>
          <relatedStateVariable>BoilerPressure</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
  </actionList>
</scpd>
```

```
    </argumentList>
  </action>

  <action>
    <name>GetBoilerTemperature</name>
    <argumentList>
      <argument>
        <name>retBoilerTemperatureTarget</name>
        <direction>out</direction>
        <relatedStateVariable>BoilerTemperature
        </relatedStateVariable>
      </argument>
    </argumentList>
  </action>

  <action>
    <name>GetBrightness</name>
    <argumentList>
      <argument>
        <name>retBrightnessTarget</name>
        <direction>out</direction>
        <relatedStateVariable>Brightness</relatedStateVariable>
      </argument>
    </argumentList>
  </action>

  <action>
    <name>GetConditioningPower</name>
    <argumentList>
      <argument>
        <name>retConditioningPowerTarget</name>
        <direction>out</direction>
        <relatedStateVariable>AirConditioning
        </relatedStateVariable>
      </argument>
    </argumentList>
  </action>

  <action>
    <name>GetCoolingEfficiency</name>
    <argumentList>
      <argument>
        <name>retCoolingEfficiencyTarget</name>
```

```
        <direction>out</direction>
        <relatedStateVariable>CoolingEfficiency
        </relatedStateVariable>
    </argument>
</argumentList>
</action>

<action>
    <name>GetCounter</name>
    <argumentList>
        <argument>
            <name>retCounterStatus</name>
            <direction>out</direction>
            <relatedStateVariable>Counter</relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>GetCurrentConsumption</name>
    <argumentList>
        <argument>
            <name>retCurrentConsumptionStatus</name>
            <direction>out</direction>
            <relatedStateVariable>CurrentConsumption
            </relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>GetDevicePower</name>
    <argumentList>
        <argument>
            <name>retDevicePowerTarget</name>
            <direction>out</direction>
            <relatedStateVariable>Power</relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>GetDoorPosition</name>
```

```
<argumentList>
  <argument>
    <name>retDoorPositionTarget</name>
    <direction>out</direction>
    <relatedStateVariable>DoorPosition</relatedStateVariable>
  </argument>
</argumentList>
</action>

<action>
  <name>GetEffectivePower</name>
  <argumentList>
    <argument>
      <name>retEffectivePowerStatus</name>
      <direction>out</direction>
      <relatedStateVariable>EffectivePower</relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>GetElectricalEfficiency</name>
  <argumentList>
    <argument>
      <name>retElectricalEfficiencyStatus</name>
      <direction>out</direction>
      <relatedStateVariable>ElectricalEfficiency
    </relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>GetFuelConsumption</name>
  <argumentList>
    <argument>
      <name>retFuelConsumptionStatus</name>
      <direction>out</direction>
      <relatedStateVariable>FuelConsumption
    </relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

```
<action>
  <name>GetGasConsumption</name>
  <argumentList>
    <argument>
      <name>retGasConsumptionStatus</name>
      <direction>out</direction>
      <relatedStateVariable>GasConsumption</relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

```
<action>
  <name>GetHeatingPower</name>
  <argumentList>
    <argument>
      <name>retHeatingPowerTarget</name>
      <direction>out</direction>
      <relatedStateVariable>HeatingBurner</relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

```
<action>
  <name>GetHeatLoss</name>
  <argumentList>
    <argument>
      <name>retHeatLossStatus</name>
      <direction>out</direction>
      <relatedStateVariable>HeatLoss</relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

```
<action>
  <name>GetHeatOutput</name>
  <argumentList>
    <argument>
      <name>retHeatOutputTarget</name>
      <direction>out</direction>
      <relatedStateVariable>HeatOutput</relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

```
<action>
  <name>GetHumidification</name>
  <argumentList>
    <argument>
      <name>retHumidificationTarget</name>
      <direction>out</direction>
      <relatedStateVariable>Humidification</relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

```
<action>
  <name>GetHumidity</name>
  <argumentList>
    <argument>
      <name>retHumidityStatus</name>
      <direction>out</direction>
      <relatedStateVariable>Humidity</relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

```
<action>
  <name>GetInitialPower</name>
  <argumentList>
    <argument>
      <name>retInitialPowerTarget</name>
      <direction>out</direction>
      <relatedStateVariable>InitialPower</relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

```
<action>
  <name>GetLightningPower</name>
  <argumentList>
    <argument>
      <name>retLightningPowerTarget</name>
      <direction>out</direction>
      <relatedStateVariable>Light</relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

```
<action>
  <name>GetLouverPosition</name>
  <argumentList>
    <argument>
      <name>retLouverPositionTarget</name>
      <direction>out</direction>
      <relatedStateVariable>LouverPosition</relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

```
<action>
  <name>GetMassFlowRate</name>
  <argumentList>
    <argument>
      <name>retMassFlowRateTarget</name>
      <direction>out</direction>
      <relatedStateVariable>MassFlowRate</relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

```
<action>
  <name>GetOilConsumption</name>
  <argumentList>
    <argument>
      <name>retOilConsumptionStatus</name>
      <direction>out</direction>
      <relatedStateVariable>OilConsumption</relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

```
<action>
  <name>GetOperatingFrequency</name>
  <argumentList>
    <argument>
      <name>retOperatingFrequencyTarget</name>
      <direction>out</direction>
      <relatedStateVariable>OperatingFrequency
        </relatedStateVariable>
    </argument>
  </argumentList>
```

```
</action>
```

```
<action>
```

```
  <name>GetOperatingHours</name>
```

```
  <argumentList>
```

```
    <argument>
```

```
      <name>retOperatingHoursStatus</name>
```

```
      <direction>out</direction>
```

```
      <relatedStateVariable>OperatingHours</relatedStateVariable>
```

```
    </argument>
```

```
  </argumentList>
```

```
</action>
```

```
<action>
```

```
  <name>GetOperatingNoise</name>
```

```
  <argumentList>
```

```
    <argument>
```

```
      <name>retOperatingNoiseStatus</name>
```

```
      <direction>out</direction>
```

```
      <relatedStateVariable>OperatingNoise</relatedStateVariable>
```

```
    </argument>
```

```
  </argumentList>
```

```
</action>
```

```
<action>
```

```
  <name>GetOperatingPressure</name>
```

```
  <argumentList>
```

```
    <argument>
```

```
      <name>retOperatingPressureStatus</name>
```

```
      <direction>out</direction>
```

```
      <relatedStateVariable>OperatingPressure
```

```
      </relatedStateVariable>
```

```
    </argument>
```

```
  </argumentList>
```

```
</action>
```

```
<action>
```

```
  <name>GetOperatingRotationalSpeed</name>
```

```
  <argumentList>
```

```
    <argument>
```

```
      <name>retOperatingRotationalSpeedTarget</name>
```

```
      <direction>out</direction>
```

```
      <relatedStateVariable>OperatingRotationalSpeed
```

```
        </relatedStateVariable>
    </argument>
</argumentList>
</action>

<action>
  <name>GetOperatingSpeed</name>
  <argumentList>
    <argument>
      <name>retOperatingSpeedTarget</name>
      <direction>out</direction>
      <relatedStateVariable>OperatingSpeed</relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>GetOperatingTemperature</name>
  <argumentList>
    <argument>
      <name>retOperatingTemperatureStatus</name>
      <direction>out</direction>
      <relatedStateVariable>OperatingTemperature
        </relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>GetOperationStaticHead</name>
  <argumentList>
    <argument>
      <name>retOperationStaticHeadTarget</name>
      <direction>out</direction>
      <relatedStateVariable>OperatingStaticHead
        </relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>GetPowerInput</name>
  <argumentList>
    <argument>
```

```
        <name>retPowerInputStatus</name>
        <direction>out</direction>
        <relatedStateVariable>PowerInput</relatedStateVariable>
    </argument>
</argumentList>
</action>

<action>
    <name>GetPowerLevel</name>
    <argumentList>
        <argument>
            <name>retPowerLevelTarget</name>
            <direction>out</direction>
            <relatedStateVariable>Level</relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>GetPowerOutput</name>
    <argumentList>
        <argument>
            <name>retPowerOutputStatus</name>
            <direction>out</direction>
            <relatedStateVariable>PowerOutput</relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>GetPressureValvePosition</name>
    <argumentList>
        <argument>
            <name>retPressureValvePositionTarget</name>
            <direction>out</direction>
            <relatedStateVariable>PressureValvePosition
        </relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>GetPumpDeliveryRate</name>
```

```
<argumentList>
  <argument>
    <name>retPumpDeliveryRateTarget</name>
    <direction>out</direction>
    <relatedStateVariable>PumpDeliveryRate
      </relatedStateVariable>
  </argument>
</argumentList>
</action>

<action>
  <name>GetResolution</name>
  <argumentList>
    <argument>
      <name>retResolutionTarget</name>
      <direction>out</direction>
      <relatedStateVariable>Resolution</relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>GetVentilationPower</name>
  <argumentList>
    <argument>
      <name>retVentilationPowerTarget</name>
      <direction>out</direction>
      <relatedStateVariable>Ventilation</relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>GetVentilationTemperature</name>
  <argumentList>
    <argument>
      <name>retVentilationTemperatureTarget</name>
      <direction>out</direction>
      <relatedStateVariable>VentilationTemperature
        </relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

```
<action>
  <name>GetVoltage</name>
  <argumentList>
    <argument>
      <name>retVoltageTarget</name>
      <direction>out</direction>
      <relatedStateVariable>Voltage</relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

```
<action>
  <name>GetWaterConsumption</name>
  <argumentList>
    <argument>
      <name>retWaterConsumptionStatus</name>
      <direction>out</direction>
      <relatedStateVariable>WaterConsumption
        </relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

```
<action>
  <name>GetWaterReservoirTemperature</name>
  <argumentList>
    <argument>
      <name>retWaterReservoirTemperatureTarget</name>
      <direction>out</direction>
      <relatedStateVariable>WaterReservoirTemperature
        </relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

```
<action>
  <name>GetWaterTemperatureReturn</name>
  <argumentList>
    <argument>
      <name>retWaterTemperatureReturnStatus</name>
      <direction>out</direction>
      <relatedStateVariable>WaterTemperatureReturn
        </relatedStateVariable>
    </argument>
  </argumentList>
```

```
        </argument>
    </argumentList>
</action>

<action>
    <name>GetWaterTemperatureSupply</name>
    <argumentList>
        <argument>
            <name>retWaterTemperatureSupplyStatus</name>
            <direction>out</direction>
            <relatedStateVariable>WaterTemperatureSupply
                </relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>GetWindowPosition</name>
    <argumentList>
        <argument>
            <name>retWindowPositionTarget</name>
            <direction>out</direction>
            <relatedStateVariable>WindowPosition</relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>SetBoilerTemperature</name>
    <argumentList>
        <argument>
            <name>newBoilerTemperatureTarget</name>
            <direction>in</direction>
            <relatedStateVariable>BoilerTemperature
                </relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>SetBrightness</name>
    <argumentList>
        <argument>
```

```
        <name>newBrightnessTarget</name>
        <direction>in</direction>
        <relatedStateVariable>Brightness</relatedStateVariable>
    </argument>
</argumentList>
</action>
```

```
<action>
    <name>SetConditioningPower</name>
    <argumentList>
        <argument>
            <name>newConditioningPowerTarget</name>
            <direction>in</direction>
            <relatedStateVariable>AirConditioning
                </relatedStateVariable>
        </argument>
    </argumentList>
</action>
```

```
<action>
    <name>SetCoolingEfficiency</name>
    <argumentList>
        <argument>
            <name>newCoolingEfficiencyTarget</name>
            <direction>in</direction>
            <relatedStateVariable>CoolingEfficiency
                </relatedStateVariable>
        </argument>
    </argumentList>
</action>
```

```
<action>
    <name>SetDevicePower</name>
    <argumentList>
        <argument>
            <name>newDevicePowerTarget</name>
            <direction>in</direction>
            <relatedStateVariable>Power</relatedStateVariable>
        </argument>
    </argumentList>
</action>
```

```
<action>
  <name>SetDoorPosition</name>
  <argumentList>
    <argument>
      <name>newDoorPositionTarget</name>
      <direction>in</direction>
      <relatedStateVariable>DoorPosition</relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>SetHeatingPower</name>
  <argumentList>
    <argument>
      <name>newHeatingPowerTarget</name>
      <direction>in</direction>
      <relatedStateVariable>HeatingBurner</relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>SetHeatOutput</name>
  <argumentList>
    <argument>
      <name>newHeatOutputTarget</name>
      <direction>in</direction>
      <relatedStateVariable>HeatOutput</relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>SetHumidification</name>
  <argumentList>
    <argument>
      <name>newHumidificationTarget</name>
      <direction>in</direction>
      <relatedStateVariable>Humidification</relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

```
<action>
  <name>SetInitialPower</name>
  <argumentList>
    <argument>
      <name>newInitialPowerTarget</name>
      <direction>in</direction>
      <relatedStateVariable>InitialPower</relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>SetLightningPower</name>
  <argumentList>
    <argument>
      <name>newLightningPowerTarget</name>
      <direction>in</direction>
      <relatedStateVariable>Light</relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>SetLouverPosition</name>
  <argumentList>
    <argument>
      <name>newLouverPositionTarget</name>
      <direction>in</direction>
      <relatedStateVariable>LouverPosition</relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>SetMassFlowRate</name>
  <argumentList>
    <argument>
      <name>newMassFlowRateTarget</name>
      <direction>in</direction>
      <relatedStateVariable>MassFlowRate</relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

```
<action>
  <name>SetOperatingFrequency</name>
  <argumentList>
    <argument>
      <name>newOperatingFrequencyTarget</name>
      <direction>in</direction>
      <relatedStateVariable>OperatingFrequency
    </relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>SetOperatingRotationalSpeed</name>
  <argumentList>
    <argument>
      <name>newOperatingRotationalSpeedTarget</name>
      <direction>in</direction>
      <relatedStateVariable>OperatingRotationalSpeed
    </relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>SetOperatingSpeed</name>
  <argumentList>
    <argument>
      <name>newOperatingSpeedTarget</name>
      <direction>in</direction>
      <relatedStateVariable>OperatingSpeed</relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>SetOperationStaticHead</name>
  <argumentList>
    <argument>
      <name>newOperationStaticHeadTarget</name>
      <direction>in</direction>
      <relatedStateVariable>OperatingStaticHead
    </relatedStateVariable>
```

```
        </argument>
    </argumentList>
</action>

<action>
    <name>SetPowerLevel</name>
    <argumentList>
        <argument>
            <name>newPowerLevelTarget</name>
            <direction>in</direction>
            <relatedStateVariable>Level</relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>SetPressureValvePosition</name>
    <argumentList>
        <argument>
            <name>newPressureValvePositionTarget</name>
            <direction>in</direction>
            <relatedStateVariable>PressureValvePosition
            </relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>SetPumpDeliveryRate</name>
    <argumentList>
        <argument>
            <name>newPumpDeliveryRateTarget</name>
            <direction>in</direction>
            <relatedStateVariable>PumpDeliveryRate
            </relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>SetResolution</name>
    <argumentList>
        <argument>
```

```
        <name>newResolutionTarget</name>
        <direction>in</direction>
        <relatedStateVariable>Resolution</relatedStateVariable>
    </argument>
</argumentList>
</action>

<action>
    <name>SetVentilationPower</name>
    <argumentList>
        <argument>
            <name>newVentilationPowerTarget</name>
            <direction>in</direction>
            <relatedStateVariable>Ventilation</relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>SetVentilationTemperature</name>
    <argumentList>
        <argument>
            <name>newVentilationTemperatureTarget</name>
            <direction>in</direction>
            <relatedStateVariable>VentilationTemperature
                </relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>SetVoltage</name>
    <argumentList>
        <argument>
            <name>newVoltageTarget</name>
            <direction>in</direction>
            <relatedStateVariable>Voltage</relatedStateVariable>
        </argument>
    </argumentList>
</action>

<action>
    <name>SetVolumeFlowRate</name>
```

```

    <argumentList>
      <argument>
        <name>retVolumeFlowRateTarget</name>
        <direction>out</direction>
        <relatedStateVariable>VolumeFlowRate</relatedStateVariable>
      </argument>
    </argumentList>
  </action>

  <action>
    <name>SetWaterReservoirTemperature</name>
    <argumentList>
      <argument>
        <name>newWaterReservoirTemperatureTarget</name>
        <direction>in</direction>
        <relatedStateVariable>WaterReservoirTemperature
        </relatedStateVariable>
      </argument>
    </argumentList>
  </action>

  <action>
    <name>SetWindowPosition</name>
    <argumentList>
      <argument>
        <name>newWindowPositionTarget</name>
        <direction>in</direction>
        <relatedStateVariable>WindowPosition</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
</actionList>

<serviceStateTable>
  <stateVariable sendEvents="no">
    <name>AirConditioning</name>
    <dataType>boolean</dataType>
    <defaultValue>0</defaultValue>
    <allowedValueRange>
      <minimum>0</minimum>
      <maximum>1</maximum>
    </allowedValueRange>
  </stateVariable>

```

```
<stateVariable sendEvents="no">
  <name>AirTemperatureInside</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>AirTemperatureOutside</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>BoilerPressure</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>BoilerTemperature</name>
  <dataType>ui4</dataType>
  <defaultValue>75</defaultValue>
  <allowedValueRange>
    <minimum>40</minimum>
    <maximum>80</maximum>
  </allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>Brightness</name>
  <dataType>ui4</dataType>
  <defaultValue>600</defaultValue>
  <allowedValueRange>
    <minimum>1</minimum>
    <maximum>1000</maximum>
  </allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>CoolingEfficiency</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>
```

```
<stateVariable sendEvents="no">
  <name>Counter</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>CurrentConsumption</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>DoorPosition</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
  <allowedValueRange>
    <minimum>0</minimum>
    <maximum>10</maximum>
  </allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>EffectivePower</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>ElectricalEfficiency</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>FuelConsumption</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>GasConsumption</name>
  <dataType>ui4</dataType>
```

```
<defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>HeatingBurner</name>
  <dataType>boolean</dataType>
  <defaultValue>0</defaultValue>
  <allowedValueRange>
    <minimum>0</minimum>
    <maximum>1</maximum>
  </allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>HeatLoss</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>HeatOutput</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>Humidification</name>
  <dataType>ui4</dataType>
  <defaultValue>40</defaultValue>
  <allowedValueRange>
    <minimum>0</minimum>
    <maximum>100</maximum>
  </allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>Humidity</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
  <allowedValueRange>
    <minimum>0</minimum>
    <maximum>100</maximum>
  </allowedValueRange>
```

```
</stateVariable>

<stateVariable sendEvents="no">
  <name>InitialPower</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>Level</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
  <allowedValueRange>
    <minimum>1</minimum>
    <maximum>10</maximum>
  </allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>Light</name>
  <dataType>boolean</dataType>
  <defaultValue>0</defaultValue>
  <allowedValueRange>
    <minimum>0</minimum>
    <maximum>1</maximum>
  </allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>LouverPosition</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
  <allowedValueRange>
    <minimum>0</minimum>
    <maximum>10</maximum>
  </allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>MassFlowRate</name>
  <dataType>ui4</dataType>
  <defaultValue>40</defaultValue>
  <allowedValueRange>
```

```
<minimum>0</minimum>
<maximum>60</maximum>
</allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>OilConsumption</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>OperatingFrequency</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>OperatingHours</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>OperatingNoise</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>OperatingPressure</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>OperatingRotationalSpeed</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>OperatingSpeed</name>
```

```
<dataType>ui4</dataType>
<defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>OperatingStaticHead</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>OperatingTemperature</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>Power</name>
  <dataType>boolean</dataType>
  <defaultValue>0</defaultValue>
  <allowedValueRange>
    <minimum>0</minimum>
    <maximum>1</maximum>
  </allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>PowerInput</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>PowerOutput</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>PressureValvePosition</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
  <allowedValueRange>
```

```
<minimum>0</minimum>
<maximum>10</maximum>
</allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>PumpDeliveryRate</name>
  <dataType>ui4</dataType>
  <defaultValue>40</defaultValue>
  <allowedValueRange>
    <minimum>0</minimum>
    <maximum>60</maximum>
  </allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>Resolution</name>
  <dataType>ui4</dataType>
  <defaultValue>960000</defaultValue>
  <allowedValueRange>
    <minimum>240000</minimum>
    <maximum>6291456</maximum>
  </allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>Ventilation</name>
  <dataType>boolean</dataType>
  <defaultValue>0</defaultValue>
  <allowedValueRange>
    <minimum>0</minimum>
    <maximum>1</maximum>
  </allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>VentilationTemperature</name>
  <dataType>ui4</dataType>
  <defaultValue>20</defaultValue>
  <allowedValueRange>
    <minimum>10</minimum>
    <maximum>30</maximum>
  </allowedValueRange>
```

```
</stateVariable>

<stateVariable sendEvents="no">
  <name>Voltage</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>VolumeFlowRate</name>
  <dataType>ui4</dataType>
  <defaultValue>40</defaultValue>
  <allowedValueRange>
    <minimum>0</minimum>
    <maximum>60</maximum>
  </allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>WaterConsumption</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>WaterReservoirTemperature</name>
  <dataType>ui4</dataType>
  <defaultValue>60</defaultValue>
  <allowedValueRange>
    <minimum>40</minimum>
    <maximum>70</maximum>
  </allowedValueRange>
</stateVariable>

<stateVariable sendEvents="no">
  <name>WaterTemperatureReturn</name>
  <dataType>ui4</dataType>
  <defaultValue>0</defaultValue>
</stateVariable>

<stateVariable sendEvents="no">
  <name>WaterTemperatureSupply</name>
  <dataType>ui4</dataType>
```

```
        <defaultValue>0</defaultValue>
    </stateVariable>

    <stateVariable sendEvents="no">
        <name>WindowPosition</name>
        <dataType>ui4</dataType>
        <defaultValue>0</defaultValue>
        <allowedValueRange>
            <minimum>0</minimum>
            <maximum>10</maximum>
        </allowedValueRange>
    </stateVariable>

</serviceStateTable>

</scpd>
```

I.d Quellcode FMControl

```
Imports System.IO
Imports System.Xml
Imports System.Data.OleDb
Imports System.Data.SqlClient
Imports System.IO.Ports
Imports System.Web.Services.Configuration
Imports System.Net

Public Class _Default
    Inherits System.Web.UI.Page

    'Erzeugen eines Objektes der Klasse OleDbConnection
    Private conn As New OleDbConnection()

    'Erzeugen zweier Objekte der Klasse OleDbCommand
    Private cmd As New OleDbCommand()
    Private cmd2 As New OleDbCommand()

    'Erzeugen zweier DataReader- und DataCounter-Objekte
    Private dataReader As OleDbDataReader
    Private dataReader2 As OleDbDataReader

    'Erzeugen globaler Variablen
    Private columnPresence As Boolean = False
    Private preSelect As Boolean = False
    Private selectedManufacturer As String = ""
    Private selectedModelNumber As String = ""
    Private selectedSerialNumber As String = ""

    'Erzeugen eines XML-Dokuments
    Private xmlDocument As New XmlDocument()

    'Definition von SQL-Variablen
    Private sqlSelect As String
    Private sqlInsert As String
    Private sqlInsertColumn As String = ""
    Private sqlInsertValue As String = ""

    'Definition der XML-Datei (Vorgabe für Demo)
    Private xmlFilePath As String = "C:\inetpub\wwwroot\FMControl\App_Data"
    Private xmlDeviceFile As String = "device.xml"
    Private xmlServiceFile As String = ""
```

```
Private stateVariableID As String
Private oldValue As String

'Definition des Dateipfades der Datenbankdatei
Private mdbFilePath As String =
"C:\inetpub\wwwroot\FMControl\App_Data\FMControl.mdb"

'Definition von Tabellennamen und Relationsfeldern aus der CAFM-Datenbank
Public tableNameList(,) As String = New String(6, 4) {"contactData",
"companyName", "", "", ""}, {"deviceCategory", "categoryNumber", "", "",
""}, {"deviceClass", "modelName", "", "manufacturerID", "categoryID"},
{"deviceObject", "serialNumber", "", "deviceClassID", ""},
{"serviceActionList", "actionName", "", "deviceClassID", ""},
{"serviceArgumentList", "argumentName", "", "actionNameID", ""},
{"serviceStateTable", "stateVariableName", "", "deviceObjectID", ""}}

'Definition des Dateipfades des Connection-Strings
Private connStr As String = ("Provider=Microsoft.Jet.OLEDB.4.0; " & "Data
Source=" & mdbFilePath)

'Definition des COM-Ports (nicht für den Standard vorgesehen, nur zu
Demonstrationszwecken)
Private WithEvents Port As SerialPort = New SerialPort("COM14", 9600,
Parity.None, 8, StopBits.One)

'Seite laden
Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load

End Sub

'Form laden
Private Sub Form1_Load(ByVal sender As Object, ByVal e As EventArgs)
Handles MyBase.Load

    If Not IsPostBack Then

        'Datenbankverbindung öffnen
        dbConnection()
        cmd.Connection = conn

        'Auswahlbox Hersteller füllen
        updateDropDownManufacturer()
```

```

'Auswahlbox Import füllen
updateImportList()

'Datenbankverbindung schließen
conn.Close()

'Meldung zurücksetzen und Buttons zum Quittieren zurücksetzen
lblMessage.Text = "Bereit"
btnMessage.Enabled = False

End If

End Sub

'***** Events *****

'Event nach Auswahl eines Herstellers
Protected Sub ListBoxManufacturer_SelectedIndexChanged(ByVal sender As
Object, ByVal e As EventArgs) Handles
ListBoxManufacturer.SelectedIndexChanged

'Ausgewählten Eintrag abfragen (Text und Wert)
selectedManufacturer = Me.ListBoxManufacturer.SelectedItem.Text
Dim selectedManufacturerID = Me.ListBoxManufacturer.SelectedValue

'Datenbankverbindung öffnen
dbConnection()
cmd.Connection = conn

'Verfügbarkeit von Feldern und Buttons definieren
btnSelect.Enabled = False
btnSave.Enabled = False
btnExit.Enabled = False
ListBoxSerialNumber.Enabled = False

'Listen und Felder zuvor ausgewählter Geräte leeren
ListBoxSerialNumber.Items.Clear()
ListBoxModelNumber.Items.Clear()
clearFields()

'Ausgewählten Eintrag prüfen
If selectedManufacturer = "--- bitte auswählen ---" Then
    ListBoxModelNumber.Enabled = False

```

```
If preSelect = False Then
    'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
    lblMessage.Text = "Kein Hersteller gewählt"
    btnMessage.Enabled = True
End If
Else
    ListBoxModelNumber.Enabled = True
    'Listbox Modell-Nr. filtern und anzeigen
    updateDropDownModelNumber(selectedManufacturer,
    selectedManufacturerID)
    If preSelect = False Then
        'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
        lblMessage.Text = " Hersteller gewählt"
        btnMessage.Enabled = True
    End If
End If

'Datenbankverbindung schließen
conn.Close()

End Sub

'Event nach Auswahl einer Modell-Nummer
Protected Sub ListBoxModelNumber_SelectedIndexChanged(ByVal sender As
Object, ByVal e As EventArgs) Handles
ListBoxModelNumber.SelectedIndexChanged

    'Ausgewählten Eintrag abfragen (Text und Wert)
    selectedModelNumber = Me.ListBoxModelNumber.SelectedItem.Text
    Dim selectedModelNumberID = Me.ListBoxModelNumber.SelectedValue

    'Datenbankverbindung öffnen
    dbConnection()
    cmd.Connection = conn

    'Verfügbarkeit von Feldern und Buttons definieren
    btnSelect.Enabled = False
    btnSave.Enabled = False
    btnExit.Enabled = False

    'Listen und Felder zuvor ausgewählter Geräte leeren
    ListBoxSerialNumber.Items.Clear()
    clearFields()
```

```

'Ausgewählten Eintrag prüfen
If selectedModelNumber = "--- bitte auswählen ---" Then
    ListBoxSerialNumber.Enabled = False
    If preSelect = False Then
        'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
        lblMessage.Text = "Keine Modell-Nummer gewählt"
        btnMessage.Enabled = True
    End If
Else
    ListBoxSerialNumber.Enabled = True
    'Listbox Serien-Nr. filtern und anzeigen
    updateDropDownSerialNumber(selectedModelNumber,
    selectedModelNumberID)
    If preSelect = False Then
        'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
        lblMessage.Text = "Modell-Nummer gewählt"
        btnMessage.Enabled = True
    End If
End If

'Datenbankverbindung schließen
conn.Close()

End Sub

'Event nach Auswahl eines zu importierenden Geräts
Protected Sub ListBoxImportData_SelectedIndexChanged(ByVal sender As
Object, ByVal e As EventArgs) Handles
ListBoxImportData.SelectedIndexChanged

    'Ausgewählten Eintrag abfragen (Text und Wert)
    Dim selectedImportDevice = Me.ListBoxImportData.SelectedItem.Text
    Dim selectedImportDeviceID = Me.ListBoxImportData.SelectedValue

    'Ausgewählten Eintrag prüfen
    If selectedImportDevice = "--- bitte auswählen ---" Then
        txtBaseURLImport.Text = "IP-Adresse"
        'Button Import deaktivieren
        btnImport.Enabled = False
        'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
        lblMessage.Text = "Kein Gerät für den Import gewählt"
        btnMessage.Enabled = True
    Else

```

```
'Feld baseURL füllen
txtBaseURLImport.Text = selectedImportDeviceID
'Button Import aktivieren
btnImport.Enabled = True
'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
lblMessage.Text = "Gerät für den Import gewählt"
btnMessage.Enabled = True
End If

End Sub

'Event nach Textänderung in Attributfeldern vorhandener Geräte
Protected Sub txtExistingDeviceChanged(ByVal sender As Object, ByVal e As
EventArgs) Handles txtIdentName.TextChanged, txtStartupDate.TextChanged,
txtRetailPrice.TextChanged, txtBuilding.TextChanged,
txtFloor.TextChanged, txtRoom.TextChanged, txtIdentNumber.TextChanged

'Buttons "Speichern" und "Abbrechen" aktivieren
btnSave.Enabled = True
btnExit.Enabled = True

End Sub

'Event nach Textänderung in Attributfeldern neu anzulegender Geräte
Protected Sub txtNewDeviceChanged(ByVal sender As Object, ByVal e As
EventArgs) Handles txtNewManufacturer.TextChanged,
txtNewModelNumber.TextChanged, txtNewSerialNumber.TextChanged

'Prüfen, ob alle obligatorischen Felder ausgefüllt sind
If Not ((txtNewManufacturer.Text = "" Or txtNewManufacturer.Text =
"Hersteller") Or (txtNewModelNumber.Text = "" Or txtNewModelNumber.Text
= "Modell-Nr.)) Or (txtNewSerialNumber.Text = "" Or
txtNewSerialNumber.Text = "Serien-Nr.)) Then
'Buttons "Neu anlegen" aktivieren
btnNew.Enabled = True
Else
'Buttons "Neu anlegen" deaktivieren
btnNew.Enabled = False
End If

End Sub
```

```

'Event nach Klick des Buttons "Speichern"
Protected Sub btnSave_Click(ByVal sender As Object, ByVal e As EventArgs)
Handles btnSave.Click

    'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
    lblMessage.Text = "Diese Funktion steht in der Demoversion nicht zur
    Verfügung."
    btnMessage.Enabled = True

End Sub

'Event nach Klick des Buttons "Neu anlegen"
Protected Sub btnNew_Click(ByVal sender As Object, ByVal e As EventArgs)
Handles btnNew.Click

    'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
    lblMessage.Text = "Diese Funktion steht in der Demoversion nicht zur
    Verfügung."
    btnMessage.Enabled = True

    'Textfelder rücksetzen, Button "Neu anlegen" deaktivieren
    txtNewManufacturer.Text = "Hersteller"
    txtNewModelNumber.Text = "Modell-Nr."
    txtNewSerialNumber.Text = "Serien-Nr."
    btnNew.Enabled = False

End Sub

'Event nach Klick des Buttons "Einschalten"
Protected Sub btnPowerOn_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles btnPowerOn.Click

    'Prüfe, ob EnOcean-Gateway am COM-Port (nicht für den Standard
    vorgesehen, nur zu Demonstrationszwecken)
    If txtBaseURL.Text = "COM-Port" Then

        'Steuerbefehl für Gerät am COM-Port zusammenstellen
        Dim command As String = txtMax.Text
        Dim n As Integer = command.Length \ 2
        Dim datagramm(n - 1) As Byte
        For i As Integer = 0 To command.Length - 1 Step 2
            datagramm(i \ 2) = Convert.ToByte(command.Substring(i, 2), 16)
        Next
    End If

```

```
Try
    'COM-Port öffnen
    Port.Open()
    'Command senden
    Port.Write(datagramm, 0, 14)
    'COM-Port schließen
    Port.Close()
    history(txtMax.Text)
Catch ex As Exception
    'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
    lblMessage.Text = "Steuerbefehl konnte nicht gesendet werden,
    Endgerät nicht erreichbar."
End Try

Else

    'Steuerbefehl für UPnP-Gerät zusammenstellen
    Dim command As String = "http://" & txtBaseURL.Text & "/" &
    txtControlURL.Text & txtArgument.Text & txtMax.Text
    Try
        'Steuerbefehl senden
        Dim Request As HttpWebRequest = WebRequest.Create(command)
        Dim Response As HttpWebResponse = Request.GetResponse
        history(txtMax.Text)
    Catch ex As Exception
        'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
        lblMessage.Text = "Steuerbefehl konnte nicht gesendet werden,
        Endgerät nicht erreichbar."
    End Try

End If

End Sub

'Event nach Klick des Buttons "Ausschalten"
Protected Sub btnPowerOff_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles btnPowerOff.Click

    'Prüfe, ob EnOcean-Gateway am COM-Port (nicht für den Standard
vorgesehen, nur zu Demonstrationszwecken)
    If txtBaseURL.Text = "COM-Port" Then
```

```

'Steuerbefehl für Gerät am COM-Port zusammenstellen
Dim command As String = txtMin.Text
Dim n As Integer = command.Length \ 2
Dim datagramm(n - 1) As Byte
For i As Integer = 0 To command.Length - 1 Step 2
    datagramm(i \ 2) = Convert.ToByte(command.Substring(i, 2), 16)
Next

Try
    'COM-Port öffnen
    Port.Open()
    'Command senden
    Port.Write(datagramm, 0, 14)
    'COM-Port schließen
    Port.Close()
    history(txtMin.Text)
Catch ex As Exception
    'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
    lblMessage.Text = "Steuerbefehl konnte nicht gesendet werden,
    Endgerät nicht erreichbar."
End Try

Else

'Steuerbefehl für UPnP-Gerät zusammenstellen
Dim command As String = "http://" & txtBaseUrl.Text & "/" &
txtControlURL.Text & txtArgument.Text & txtMin.Text
Try
    'Steuerbefehl senden
    Dim Request As HttpWebRequest = WebRequest.Create(command)
    Dim Response As HttpWebResponse = Request.GetResponse
    history(txtMin.Text)
Catch ex As Exception
    'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
    lblMessage.Text = "Steuerbefehl konnte nicht gesendet werden,
    Endgerät nicht erreichbar."
    btnMessage.Enabled = True
End Try

End If

End Sub

```

```
'Event nach Klick des Buttons "Meldung quittieren"
Protected Sub btnMessage_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles btnMessage.Click

    'Meldung zurücksetzen und Buttons zum Quittieren zurücksetzen
    lblMessage.Text = "Bereit"
    btnMessage.Enabled = False

End Sub

'Event nach Klick des Buttons "Import"
Protected Sub btnImport_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles btnImport.Click

    'Verbindung zur Datenbank aufbauen
    dbConnection()

    'Ausgewählten Eintrag abfragen
    Dim Value As String = Me.ListBoxImportData.SelectedValue.ToString()

    'XML-Dateien auslesen und Werte in Datenbank schreiben
    importData(Value)

    'Verbindung zur Datenbank schließen
    conn.Close()

    'Importiertes Gerät auswählen und anzeigen
    showImport()

End Sub

'***** Funktionen *****

'Verbindung zur Datenbank aufbauen
Private Sub dbConnection()

    Try

        'Datenbankverbindung öffnen
        conn.ConnectionString = connStr
        conn.Open()

        If preSelect = False Then
```

```

        'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
        lblMessage.Text = "Datenbankverbindung geöffnet"
        btnMessage.Enabled = True
    End If

Catch ex As Exception

    'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
    lblMessage.Text = "Feher bei Datenbankverbindung: " & ex.ToString
    btnMessage.Enabled = True

End Try

End Sub

'Listbox Hersteller aktualisieren
Private Sub updateDropDownManufacturer()

    Try
        'Inhalt aus Datenbank auslesen
        cmd.CommandText = "SELECT contactData.ID, contactData.companyName
        FROM contactData ORDER BY contactData.companyName"
        Dim manufacturerList As DataTable = New DataTable()
        dataReader = cmd.ExecuteReader()
        If dataReader.HasRows Then
            manufacturerList.Load(dataReader)
        End If
        'Listbox füllen
        ListBoxManufacturer.DataSource = manufacturerList
        ListBoxManufacturer.DataValueField = "ID"
        ListBoxManufacturer.DataTextField = "companyName"
        ListBoxManufacturer.DataBind()
        ListBoxManufacturer.Items.Insert(0, "--- bitte auswählen ---")
        'Eintrag selektieren, falls Anzeige nach Importvorgang
        If preSelect = True Then
            Dim selectedItem() As DataRow =
            manufacturerList.Select("companyName='" & selectedManufacturer &
            "'")
            ListBoxManufacturer.SelectedValue = selectedItem(0).Item("ID")
        End If
        If preSelect = False Then
            'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
            lblMessage.Text = "Liste der Hersteller aktualisiert"
        End If
    End Try
End Sub

```

```
        btnMessage.Enabled = True
    End If

    Catch ex As Exception
        If preSelect = False Then
            'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
            lblMessage.Text = "Liste der Hersteller konnte nicht aktualisiert
            werden"
            btnMessage.Enabled = True
        End If
    End Try

End Sub

'Listbox Modell-Nummer aktualisieren
Private Sub updateDropDownModelNumber(ByVal text As String, ByVal value
As String)

    Try
        'Inhalt aus Datenbank auslesen
        cmd.CommandText = "SELECT deviceClass.ID, deviceClass.modelNumber,
        deviceClass.manufacturerID FROM deviceClass ORDER BY deviceClass.ID"
        Dim modelList As New DataTable
        dataReader = cmd.ExecuteReader()
        If dataReader.HasRows Then
            modelList.Load(dataReader)
        End If
        'Listbox füllen und nach Auswahl des Herstellers filtern
        ListBoxModelNumber.DataSource = New DataView(modelList,
        "manufacturerID=" & value, "ID", DataViewRowState.CurrentRows)
        ListBoxModelNumber.DataValueField = "ID"
        ListBoxModelNumber.DataTextField = "modelNumber"
        ListBoxModelNumber.DataBind()
        ListBoxModelNumber.Items.Insert(0, "--- bitte auswählen ---")
        'Eintrag selektieren, falls Anzeige nach Importvorgang
        If preSelect = True Then
            Dim selectedItem() As DataRow = modelList.Select("manufacturerID="
            & value & " And modelNumber='" & selectedModelNumber & "'")
            ListBoxModelNumber.SelectedValue = selectedItem(0).Item("ID")
            'ListBoxModelNumber.SelectedValue = selectedModelNumber
        End If
        If preSelect = False Then
```

```

        'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
        lblMessage.Text = "Liste der Modell-Nummern aktualisiert"
        btnMessage.Enabled = True
    End If

Catch ex As Exception
    If preSelect = False Then
        'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
        lblMessage.Text = "Liste der Modell-Nummern konnte nicht
        aktualisiert werden"
        btnMessage.Enabled = True
    End If
End Try

End Sub

'Listbox Serien-Nummer aktualisieren
Private Sub updateDropDownSerialNumber(ByVal text As String, ByVal value
As String)

Try
    'Inhalt aus Datenbank auslesen
    cmd.CommandText = "SELECT deviceObject.ID, deviceObject.serialNumber,
    deviceObject.deviceClassID FROM deviceObject"
    Dim serialList As New DataTable
    dataReader = cmd.ExecuteReader()
    If dataReader.HasRows Then
        serialList.Load(dataReader)
    End If
    'Listbox füllen und nach Auswahl der Modell-Nummer filtern
    ListBoxSerialNumber.DataSource = New DataView(serialList,
    "deviceClassID=" & value, "ID", DataViewRowState.CurrentRows)
    ListBoxSerialNumber.DataValueField = "ID"
    ListBoxSerialNumber.DataTextField = "serialNumber"
    ListBoxSerialNumber.DataBind()
    ListBoxSerialNumber.Items.Insert(0, "--- bitte auswählen ---")
    'Eintrag selektieren, falls Anzeige nach Importvorgang
    If preSelect = True Then
        Dim selectedItem() As DataRow = serialList.Select("deviceClassID="
        & value & " And serialNumber='" & selectedSerialNumber & "'")
        ListBoxSerialNumber.SelectedValue = selectedItem(0).Item("ID")
    End If
    If preSelect = False Then

```

```
'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
lblMessage.Text = "Liste der Serien-Nummern aktualisiert"
btnMessage.Enabled = True
End If

Catch ex As Exception
If preSelect = False Then
'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
lblMessage.Text = "Liste der Serien-Nummern konnte nicht
aktualisiert werden"
btnMessage.Enabled = True
End If
End Try

End Sub

'ListBox Import aktualisieren
Private Sub updateImportList()

'ListBox leeren
ListBoxImportData.Items.Clear()

'Neue DataTable mit Werten für Demo-Import erzeugen
Dim importList As DataTable = New DataTable()
importList.Columns.Add.ColumnName = "IP"
importList.Columns.Add.ColumnName = "Name"
importList.Rows.Add("192.168.0.150", "Standard-Heizzentrale")
importList.Rows.Add("192.168.0.151", "Standard-Lüftungszentrale")
importList.Rows.Add("192.168.0.152", "Standard-Klimazentrale")
importList.Rows.Add("192.168.0.153", "Heizungssteuereinheit")
importList.Rows.Add("192.168.0.154", "Lüftungssteuereinheit")
importList.Rows.Add("192.168.0.155", "Lichtsteuereinheit")
importList.Rows.Add("192.168.0.156", "Pumpensteuereinheit")
importList.Rows.Add("192.168.0.157", "Mediensteuereinheit")

'ListBox füllen
ListBoxImportData.DataSource = importList
ListBoxImportData.DataValueField = "IP"
ListBoxImportData.DataTextField = "Name"
ListBoxImportData.DataBind()
ListBoxImportData.Items.Insert(0, "--- bitte auswählen ---")
```

```
'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
lblMessage.Text = "Import-Liste aktualisiert"
btnMessage.Enabled = True
```

```
End Sub
```

```
'Attribute des ausgewählten Geräts anzeigen
Private Sub showSelection() Handles
ListBoxSerialNumber.SelectedIndexChanged, btnSelect.Click, btnExit.Click
```

```
'Verfügbarkeit von Feldern und Buttons definieren
btnSelect.Enabled = True
btnSave.Enabled = False
btnExit.Enabled = False
```

```
'Ausgewählten Eintrag abfragen (Text und Wert)
selectedSerialNumber = Me.ListBoxSerialNumber.SelectedItem.Text
Dim selectedSerialNumberID As String =
Me.ListBoxSerialNumber.SelectedValue
```

```
'Kriterien für SQL-Abfrage definieren
Dim crit_1 As String = selectedSerialNumberID
Dim crit_2 As String = "SetDevicePower"
Dim crit_3 As String = "Power"
Dim crit_4 As String = ""
```

```
'Ausgewählten Eintrag prüfen
If selectedSerialNumber = "--- bitte auswählen ---" Then
'Felder leeren und Button zurücksetzen
clearFields()
btnSelect.Enabled = False
If preSelect = False Then
'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
lblMessage.Text = "Keine Serien-Nummer gewählt"
btnMessage.Enabled = True
End If
Else
```

```
'Datenbankverbindung öffnen
dbConnection()
cmd.Connection = conn
cmd2.Connection = conn
```

Try

```
'Inhalt aus Datenbank auslesen
cmd.CommandText = "SELECT contactData.*, deviceCategory.*,
deviceClass.*, deviceObject.*, deviceObject.ID,
serviceActionList.*, serviceActionList.actionName,
serviceArgumentList.*, serviceStateTable.*,
serviceStateTable.stateVariableName FROM (((deviceCategory INNER
JOIN (contactData INNER JOIN deviceClass ON contactData.ID =
deviceClass.manufacturerID) ON deviceCategory.ID =
deviceClass.categoryID) INNER JOIN deviceObject ON deviceClass.ID =
deviceObject.deviceClassID) INNER JOIN serviceActionList ON
deviceClass.ID = serviceActionList.deviceClassID) INNER JOIN
serviceArgumentList ON serviceActionList.ID =
serviceArgumentList.actionNameID) INNER JOIN serviceStateTable ON
deviceObject.ID = serviceStateTable.deviceObjectID WHERE
(((deviceObject.ID)=" & crit_1 & ") AND
((serviceActionList.actionName)=' & crit_2 & "') AND
((serviceStateTable.stateVariableName)=' & crit_3 & "'))"
dataReader = cmd.ExecuteReader()
```

'Textfelder füllen

```
If dataReader.Read() Then
    If IsDBNull(dataReader.Item("friendlyName")) Then
        txtFriendlyName.Text = "k.A."
    Else
        txtFriendlyName.Text = dataReader.Item("friendlyName")
    End If

    If IsDBNull(dataReader.Item("categoryName")) Then
        txtCategory.Text = "k.A."
    Else
        txtCategory.Text = dataReader.Item("categoryName")
    End If

    If IsDBNull(dataReader.Item("modelName")) Then
        txtModelName.Text = "k.A."
    Else
        txtModelName.Text = dataReader.Item("modelName")
    End If

    If IsDBNull(dataReader.Item("UDN")) Then
        txtUDN.Text = "k.A."
    End If
End If
```

```
Else
    txtUDN.Text = dataReader.Item("UDN")
End If

If IsDBNull(dataReader.Item("UPC")) Then
    txtUPC.Text = "k.A."
Else
    txtUPC.Text = dataReader.Item("UPC")
End If

If IsDBNull(dataReader.Item("manufactureDate")) Then
    txtManufactureDate.Text = "k.A."
Else
    txtManufactureDate.Text = dataReader.Item("manufactureDate")
End If

If IsDBNull(dataReader.Item("warranty")) Then
    txtWarranty.Text = "k.A."
Else
    txtWarranty.Text = dataReader.Item("warranty")
End If

If IsDBNull(dataReader.Item("identNumber")) Then
    txtIdentNumber.Text = "k.A."
Else
    txtIdentNumber.Text = dataReader.Item("identNumber")
End If

If IsDBNull(dataReader.Item("identName")) Then
    txtIdentName.Text = "k.A."
Else
    txtIdentName.Text = dataReader.Item("identName")
End If

If IsDBNull(dataReader.Item("startupDate")) Then
    txtStartupDate.Text = "k.A."
Else
    txtStartupDate.Text = dataReader.Item("startupDate")
End If

If IsDBNull(dataReader.Item("retailPrice")) Then
    txtRetailPrice.Text = "k.A."
Else
```

```
txtRetailPrice.Text = dataReader.Item("retailPrice")
End If

'Prüfen, ob mit Standort verknüpft
If IsDBNull(dataReader.Item("roomID")) Then
    txtRoom.Text = "k.A."
    txtFloor.Text = "k.A."
    txtBuilding.Text = "k.A."
Else
    'Kriterium für SQL-Abfrage definieren
    crit_4 = dataReader.Item("roomID")
    'Inhalt aus Datenbank auslesen
    cmd2.CommandText = "SELECT locrefRoom.*, locrefFloor.*,
    locrefBuilding.*, locrefRoom.ID FROM locrefBuilding INNER JOIN
    (locrefFloor INNER JOIN locrefRoom ON locrefFloor.ID =
    locrefRoom.floorID) ON locrefBuilding.ID =
    locrefFloor.buildingID WHERE (((locrefRoom.ID)=" & crit_4 &
    "))"
    dataReader2 = cmd2.ExecuteReader()
    Try
        'Textfelder zum Standort füllen
        If dataReader2.Read() Then
            If IsDBNull(dataReader2.Item("buildingName")) Then
                txtBuilding.Text = "k.A."
            Else
                txtBuilding.Text = dataReader2.Item("buildingName")
            End If

            If IsDBNull(dataReader2.Item("floorName")) Then
                txtFloor.Text = "k.A."
            Else
                txtFloor.Text = dataReader2.Item("floorName")
            End If

            If IsDBNull(dataReader2.Item("roomName")) Then
                txtRoom.Text = "k.A."
            Else
                txtRoom.Text = dataReader2.Item("roomName")
            End If
        Else
            'Ausgabe einer Meldung und Aktivierung des Buttons zum
            Quittieren
            lblMessage.Text = "Fehler bei der Datenbankabfrage."
```

```
        btnMessage.Enabled = True
    End If
    'dataReader2 schließen
    dataReader2.Close()

    Catch ex As Exception
        'Ausgabe einer Meldung und Aktivierung des Buttons zum
        Quittieren
        lblMessage.Text = ex.ToString
        btnMessage.Enabled = True
    End Try
End If

If IsDBNull(dataReader.Item("controlURL")) Then
    txtControlURL.Text = "k.A."
Else
    txtControlURL.Text = dataReader.Item("controlURL")
End If

txtAction.Text = crit_2

If IsDBNull(dataReader.Item("argumentName")) Then
    txtArgument.Text = "k.A."
Else
    txtArgument.Text = dataReader.Item("argumentName")
End If

If IsDBNull(dataReader.Item("baseURL")) Then
    txtBaseURL.Text = "k.A."
Else
    txtBaseURL.Text = dataReader.Item("baseURL")
End If

If IsDBNull(dataReader.Item("stateVariableValue")) Then
    txtValue.Text = "k.A."
Else
    txtValue.Text = dataReader.Item("stateVariableValue")
End If

If IsDBNull(dataReader.Item("maximum")) Then
    txtMax.Text = "k.A."
    btnPowerOn.Enabled = False
```

```
Else
    txtMax.Text = dataReader.Item("maximum")
    btnPowerOn.Enabled = True
End If

If IsDBNull(dataReader.Item("minimum")) Then
    txtMin.Text = "k.A."
    btnPowerOff.Enabled = False
Else
    txtMin.Text = dataReader.Item("minimum")
    btnPowerOff.Enabled = True
End If

If preSelect = False Then
    'Ausgabe einer Meldung und Aktivierung des Buttons zum
    Quittieren
    lblMessage.Text = "Datensatz aktualisiert"
    btnMessage.Enabled = True
End If

Else
    If preSelect = False Then
        'Ausgabe einer Meldung und Aktivierung des Buttons zum
        Quittieren
        lblMessage.Text = "Fehler bei der Datenbankabfrage."
        btnMessage.Enabled = True
    End If
End If

'dataReader schließen
dataReader.Close()

Catch ex As Exception
    If preSelect = False Then
        'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
        lblMessage.Text = ex.ToString
        btnMessage.Enabled = True
    End If
End Try
End If

End Sub
```

```
'Felder zurücksetzen
Private Sub clearFields()

    'Attributfelder leeren
    txtFriendlyName.Text = ""
    txtCategory.Text = ""
    txtModelName.Text = ""
    txtUDN.Text = ""
    txtUPC.Text = ""
    txtManufactureDate.Text = ""
    txtWarranty.Text = ""
    txtIdentNumber.Text = ""
    txtIdentName.Text = ""
    txtStartupDate.Text = ""
    txtRetailPrice.Text = ""
    txtBuilding.Text = ""
    txtFloor.Text = ""
    txtRoom.Text = ""
    txtControlURL.Text = ""
    txtAction.Text = ""
    txtArgument.Text = ""
    txtBaseURL.Text = ""
    txtValue.Text = ""
    txtMax.Text = ""
    txtMin.Text = ""

    'Buttons zum Ein-/Ausschalten zurücksetzen
    btnPowerOn.Enabled = False
    btnPowerOff.Enabled = False

End Sub

'Ausgewähltes Gerät importieren
Private Sub importData(ByVal location As String)

    'Variablen deklarieren
    Dim xmlDocument As XmlDocument
    Dim xmlns As String
    Dim nsmgr As XmlNamespaceManager
    Dim xmlNodeList As XmlNodeList
    Dim xmlNodeListElements As Integer = 0
    Dim xmlNode As XmlNode
    Dim xmlArgumentListNodeList As XmlNodeList
```

```
Dim xmlArgumentListNodeListElements As Integer = 0
Dim xmlArgumentListNode As XmlNode
Dim xmlArgumentNodeList As XmlNodeList
Dim xmlArgumentNodeListElements As Integer = 0
Dim xmlArgumentNode As XmlNode
Dim xmlValueRangeNodeList As XmlNodeList
Dim xmlValueRangeNodeListElements As Integer = 0
Dim xmlValueRangeNode As XmlNode
Dim tableName As String = ""
Dim argumentCounter As Integer = 0
Dim valueCounter As Integer = 0
Dim xmlPathDeamon As String = ""

Try
    'Schleife 2x durchlaufen, für Device Description und Service
    Description
    For n = 0 To 1

        'Erzeugen des XML-Objekts
        xmlDocument = New XmlDocument()

        'Laden der XML-Datei
        Try
            Select Case n
                Case 0
                    'Laden der Device Description
                    xmlDocument.Load(xmlFilePath & "\" & location & "\" &
                        xmlDeviceFile)
                Case 1
                    'Laden der Service Description
                    xmlDocument.Load(xmlFilePath & "\" & location & "\" &
                        xmlServiceFile)
            End Select
        Catch ex As Exception
            'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
            xmlPathDeamon = "Gerätebeschreibung nicht gefunden."
            btnMessage.Enabled = True
        End Try

        'xmlns im root-Element berücksichtigen
        xmlns = xmlDocument.DocumentElement.Attributes("xmlns").Value
        nsmgr = New XmlNamespaceManager(xmlDocument.NameTable)
        nsmgr.AddNamespace("MsBuild", xmlns)
    
```

```

'Schleife über alle Datenbanktabellen
For tableIndex = 0 To tableNameList.GetLength(0) - 1

    tableName = tableNameList(tableIndex, 0)

'Schleife 2x durchlaufen, für verschiedene Ebenen der Nodes
For p = 0 To 1
    If n = 0 And p = 0 Then
        xmlNodeList =
            xmlDocument.SelectNodes("/MsBuild:root/MsBuild:device",
            nsmgr)
    ElseIf n = 0 And p = 1 Then
        xmlNodeList =
            xmlDocument.SelectNodes("/MsBuild:root/MsBuild:device/
            MsBuild:serviceList/MsBuild:service", nsmgr)
    ElseIf n = 1 And p = 0 Then
        xmlNodeList =
            xmlDocument.SelectNodes("/MsBuild:scpd/MsBuild:actionList/
            MsBuild:action", nsmgr)
        xmlArgumentListNodeList =
            xmlDocument.SelectNodes("/MsBuild:scpd/MsBuild:actionList/
            MsBuild:action/MsBuild:argumentList", nsmgr)
        xmlArgumentNodeList =
            xmlDocument.SelectNodes("/MsBuild:scpd/MsBuild:actionList/
            MsBuild:action/MsBuild:argumentList/MsBuild:argument", nsmgr)
    ElseIf n = 1 And p = 1 Then
        xmlNodeList =
            xmlDocument.SelectNodes("/MsBuild:scpd/
            MsBuild:serviceStateTable/MsBuild:stateVariable", nsmgr)
        xmlValueRangeNodeList =
            xmlDocument.SelectNodes("/MsBuild:scpd/
            MsBuild:serviceStateTable/MsBuild:stateVariable/
            MsBuild:allowedValueRange", nsmgr)
    Else
        Exit For
    End If

'Schleife über die Einträge/Spalten
For z = 0 To xmlNodeList.Count - 1
    xmlNode = xmlNodeList(z)

    'Auslesen der Attribute sendEvents, falls Tabelle =
    serviceStateTable

```

```
If n = 1 And p = 1 And tableName = "serviceStateTable" Then
    Try
        If xmlNode.Attributes.GetNamedItem("sendEvents").Value <>
            "" Then
                columnMatch(tableName, "sendEvents")
                buildSQL(tableName, "sendEvents",
                    xmlNode.Attributes.GetNamedItem("sendEvents").Value, 0)
            End If
        Catch ex As Exception
            'Ausgabe einer Meldung und Aktivierung des Buttons zum
            Quittieren
            lblMessage.Text = "Fehler in
            xmlNode.Attributes.GetNamedItem('sendEvents'): " &
            ex.Message
            btnMessage.Enabled = True
        End Try
    End If

    'Anzahl der Einträge/Spalten abrufen
    Dim CountChildNodes = xmlNode.ChildNodes.Count

    'Schleife über Anzahl der Einträge
    For j = 0 To CountChildNodes - 1

        'Name und Wert auslesen
        Dim columnName = xmlNode.ChildNodes.Item(j).Name
        Dim valueName = xmlNode.ChildNodes.Item(j).InnerText

        'Zwischenspeicher während Import füllen
        If n = 0 And columnName = "SCPDURL" Then xmlServiceFile =
            valueName
        If n = 0 And columnName = "manufacturer" Then
            selectedManufacturer = valueName
        If n = 0 And columnName = "modelName" Then
            selectedModelNumber = valueName
        If n = 0 And columnName = "serialNumber" Then
            selectedSerialNumber = valueName

        'Übersetzung von Attributen bei abweichender Bezeichnung
        If n = 0 And columnName = "manufacturer" Then columnName =
            "companyName"
        If n = 1 And p = 0 And columnName = "name" Then columnName =
            "actionName"
```

```

If n = 1 And p = 1 And columnName = "name" Then columnName
= "stateVariableName"

'Relationsfelder speichern
For k = 0 To tableNameList.GetLength(0) - 1
  If tableName = tableNameList(k, 0) And columnName =
  tableNameList(k, 1) Then tableNameList(k, 2) =
  (valueName)
Next

'Überprüfung, ob eine der vorgegebenen Spalten mit dem
Eintrag der XML-Datei übereinstimmt
columnMatch(tableName, columnName)

'Zusammenbau des SQL-Statements
buildSQL(tableName, columnName, valueName, j)

If n = 1 And p = 0 Then
  'Verknüpfungsfelder mit Primärschlüssel ergänzen
  addSQL(tableName)

  'Daten in Datenbank schreiben
  If sqlInsertColumn = "" Then
  Else
    insertSQL(tableName)
  End If

  'Strings für den nächsten Durchlauf zurücksetzen
  sqlInsertColumn = ""
  sqlInsertValue = ""
End If

'Bei der actionList muss je Eintrag auch die argumentList
ausgelesen und importiert werden
If n = 1 And p = 0 And tableName = "serviceActionList" And
columnName = "actionName" Then

  'Anzahl der ChildNodes (=Argumente) der jeweiligen
  ArgumentList
  xmlArgumentListNode = xmlArgumentListNodeList(j)

  'Schleife über die Argumente je ArgumentList/Aktion
  For y = (0 + argumentCounter) To

```

```
(xmlArgumentListNode.ChildNodes.Count + argumentCounter -
1)

'Anzahl der Einträge/Spalten des Arguments
xmlArgumentNode = xmlArgumentNodeList(y)
Dim CountArgumentChildNodes =
xmlArgumentNode.ChildNodes.Count

'Schleife über Anzahl der Einträge (name, direction,
relatedStateVariable)
For l = 0 To CountArgumentChildNodes - 1

    'Name und Wert auslesen
    Dim columnArgumentName =
xmlArgumentNode.ChildNodes.Item(l).Name
    Dim valueArgumentName =
xmlArgumentNode.ChildNodes.Item(l).InnerText

    'Übersetzung von Attributen bei abweichender
Bezeichnung
    If columnArgumentName = "name" Then
columnArgumentName = "argumentName"

    'Relationsfelder speichern
    For m = 0 To tableNameList.GetLength(0) - 1
        If "serviceArgumentList" = tableNameList(m, 0) And
columnArgumentName = tableNameList(m, 1) Then
            tableNameList(m, 2) = (valueArgumentName)
        Next

    'Überprüfung, ob eine der Spalten mit dem Eintrag der
XML-Datei übereinstimmt
    columnMatch("serviceArgumentList",
columnArgumentName)

    'Zusammenbau des SQL-Statements
    buildSQL("serviceArgumentList", columnArgumentName,
valueArgumentName, 1)

Next

'Verknüpfungsfelder mit Primärschlüssel ergänzen
addSQL("serviceArgumentList")
```

```

'Daten in Datenbank schreiben
If sqlInsertColumn = "" Then
Else
    insertSQL("serviceArgumentList")
End If

'Strings für den nächsten Durchlauf zurücksetzen
sqlInsertColumn = ""
sqlInsertValue = ""

Next

'Counter hochzählen
argumentCounter = argumentCounter +
xmlArgumentListNode.ChildNodes.Count
End If

'Bei der serviceStateVariable muss je Eintrag auch die
allowedValueRange ausgelesen und importiert werden
If n = 1 And p = 1 And tableName = "serviceStateTable" And
columnName.Contains("allowedValueRange") Then

    For u = (valueCounter) To (valueCounter)

        xmlValueRangeNode = xmlValueRangeNodeList(u)

        'Schleife über Anzahl der Einträge
        For v = 0 To xmlValueRangeNode.ChildNodes.Count - 1

            'Name und Wert auslesen
            Dim columnValueRangeName =
                xmlValueRangeNode.ChildNodes.Item(v).Name
            Dim valueValueRangeName =
                xmlValueRangeNode.ChildNodes.Item(v).InnerText

            'Überprüfung, ob eine der vorgegebenen Spalten mit
            dem Eintrag der XML-Datei übereinstimmt
            columnMatch(tableName, columnValueRangeName)

            'Zusammenbau des SQL-Statements
            buildSQL(tableName, columnValueRangeName,
                valueValueRangeName, v)
        Next
    Next
Next

```

```
        'Counter hochzählen
        valueCounter = valueCounter + 1
    End If

Next

If n = 1 And p = 1 Then
    'Verknüpfungsfelder mit Primärschlüssel ergänzen
    addSQL(tableName)

    'Daten in Datenbank schreiben
    If sqlInsertColumn = "" Then
    Else
        insertSQL(tableName)
    End If

    'Strings für den nächsten Durchlauf zurücksetzen
    sqlInsertColumn = ""
    sqlInsertValue = ""
End If

Next

Next
If n = 0 Then
    'Verknüpfungsfelder mit Primärschlüssel ergänzen
    addSQL(tableName)

    'Daten in Datenbank schreiben
    If sqlInsertColumn = "" Then
    Else
        insertSQL(tableName)
    End If

    'Strings für den nächsten Durchlauf zurücksetzen
    sqlInsertColumn = ""
    sqlInsertValue = ""
End If

Next
Next
```

```

Catch ex As Exception
    'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
    lblMessage.Text = xmlPathDeamon & "Fehler in importData(): " &
    ex.Message
    xmlPathDeamon = ""
End Try

End Sub

'Prüfen, ob XML-Eintrag als Feld/Spalte in der Datenbank vorhanden ist
Private Sub columnMatch(ByVal tableName As String, ByVal columnName As
String)

    'DataTable zum Zwischenspeichern der ausgelesenen Spalten
    Dim DataTable As DataTable
    'Variable zur Angabe, ob XML-Eintrag als Feld/Spalte vorhanden ist
    columnPresence = False

    Try
        'Füllen der DataTable
        DataTable = conn.GetOleDbSchemaTable(OleDbSchemaGuid.Columns, New
Object() {Nothing, Nothing, tableName, Nothing})
        'Listet die Spaltennamen aus jedem Eintrag der Schema Table auf
        For i = 0 To DataTable.Rows.Count - 1
            If columnName = DataTable.Rows(i)!COLUMN_NAME.ToString Then
                'Variable ist wahr, wenn XML-Eintrag Feld/Spalte der Datenbank
                identisch
                columnPresence = True
            End If
        Next i

    Catch ex As Exception
        'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
        lblMessage.Text = "Fehler in columnMatch(): " & ex.Message
        btnMessage.Enabled = True
        ' Variable zurücksetzen
        columnPresence = False
    End Try

End Sub

```

```
'Zusammenbau des SQL-Strings
Private Sub buildSQL(ByVal tableName As String, ByVal columnName As
String, ByVal valueName As String, ByVal i As Integer)

    'Zusammenbau des SQL-Statements
    If columnPresence Then
        'Prüfe, ob String leer ist oder bereits Einträge vorhanden sind
        If sqlInsertColumn = "" Then
            'SQL-Statement füllen
            sqlInsertColumn = columnName
            sqlInsertValue = "" & valueName & ""
        Else
            'SQL-Statement füllen
            sqlInsertColumn = sqlInsertColumn & ", " & columnName
            sqlInsertValue = sqlInsertValue & ", " & valueName & ""
        End If
        'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
        lblMessage.Text = "SQL-String erzeugt"
        btnMessage.Enabled = True
    End If

End Sub

'Ergänzung des SQL-Strings um Primärschlüssel und Relationen
Private Sub addSQL(ByVal tableName As String)
    'Variablen deklarieren
    Dim relatedTable As String = ""
    Dim relatedField As String = ""
    Dim relatedValue As String = ""
    Dim relatedTable2 As String = ""
    Dim relatedField2 As String = ""
    Dim relatedValue2 As String = ""
    Dim relatedTable3 As String = ""
    Dim relatedField3 As String = ""
    Dim relatedValue3 As String = ""
    Dim linkField As String = ""
    Dim linkField2 As String = ""
    Dim linkField3 As String = ""
    Dim sqlString As String = ""
    Dim sqlString2 As String = ""
    Dim exe As Boolean = False
    Dim primaryKey As Integer = 0
    Dim doubleLink As Integer = 0
```

```

'Primärschlüssel und Relationsfelder für Tabelle "deviceClass"
definieren
If tableName = tableNameList(2, 0) Then
    exe = True
    primaryKey = 1
    doubleLink = 1
    relatedTable = tableNameList(0, 0)
    relatedField = tableNameList(0, 1)
    relatedValue = tableNameList(0, 2)
    relatedTable2 = tableNameList(1, 0)
    relatedField2 = tableNameList(1, 1)
    relatedValue2 = tableNameList(1, 2)
    linkField = tableNameList(2, 3)
    linkField2 = tableNameList(2, 4)
End If

'Primärschlüssel und Relationsfelder für Tabelle "deviceObject"
definieren
If tableName = tableNameList(3, 0) Then
    exe = True
    primaryKey = 2
    relatedTable = tableNameList(2, 0)
    relatedField = tableNameList(2, 1)
    relatedValue = tableNameList(2, 2)
    relatedTable2 = tableNameList(0, 0)
    relatedField2 = tableNameList(0, 1)
    relatedValue2 = tableNameList(0, 2)
    linkField = tableNameList(3, 3)
    linkField2 = tableNameList(2, 3)
End If

'Primärschlüssel und Relationsfelder für Tabelle "serviceActionList"
definieren
If tableName = tableNameList(4, 0) Then
    exe = True
    primaryKey = 2
    relatedTable = tableNameList(2, 0)
    relatedField = tableNameList(2, 1)
    relatedValue = tableNameList(2, 2)
    relatedTable2 = tableNameList(0, 0)
    relatedField2 = tableNameList(0, 1)
    relatedValue2 = tableNameList(0, 2)
    linkField = tableNameList(4, 3)
    linkField2 = tableNameList(2, 3)
End If

```

```
'Primärschlüssel und Relationsfelder für Tabelle "serviceArgumentList"
definieren
```

```
If tableName = tableNameList(5, 0) Then
```

```
    exe = True
```

```
    primaryKey = 3
```

```
    relatedTable = tableNameList(4, 0)
```

```
    relatedField = tableNameList(4, 1)
```

```
    relatedValue = tableNameList(4, 2)
```

```
    relatedTable2 = tableNameList(2, 0)
```

```
    relatedField2 = tableNameList(2, 1)
```

```
    relatedValue2 = tableNameList(2, 2)
```

```
    relatedTable3 = tableNameList(0, 0)
```

```
    relatedField3 = tableNameList(0, 1)
```

```
    relatedValue3 = tableNameList(0, 2)
```

```
    linkField = tableNameList(5, 3)
```

```
    linkField2 = tableNameList(4, 3)
```

```
    linkField3 = tableNameList(2, 3)
```

```
End If
```

```
'Primärschlüssel und Relationsfelder für Tabelle "serviceStateTable"
definieren
```

```
If tableName = tableNameList(6, 0) Then
```

```
    exe = True
```

```
    primaryKey = 3
```

```
    relatedTable = tableNameList(3, 0)
```

```
    relatedField = tableNameList(3, 1)
```

```
    relatedValue = tableNameList(3, 2)
```

```
    relatedTable2 = tableNameList(2, 0)
```

```
    relatedField2 = tableNameList(2, 1)
```

```
    relatedValue2 = tableNameList(2, 2)
```

```
    relatedTable3 = tableNameList(0, 0)
```

```
    relatedField3 = tableNameList(0, 1)
```

```
    relatedValue3 = tableNameList(0, 2)
```

```
    linkField = tableNameList(6, 3)
```

```
    linkField2 = tableNameList(3, 3)
```

```
    linkField3 = tableNameList(2, 3)
```

```
End If
```

```
'Ausführen, wenn einer der Fälle eingetreten und exe=true ist
```

```
If exe = True Then
```

```
    'ID der Relationsfelder auslesen
```

```
    Select Case primaryKey
```

Case 1

'Fall 1: die verlinkte Tabelle hat einen einfachen
Primärschlüssel

```
sqlString = "SELECT " & relatedTable & ".ID FROM " & relatedTable
& " WHERE " & relatedField & "=" & relatedValue & ""
```

Case 2

'Fall 2: die verlinkte Tabelle hat einen zusammengesetzten
Primärschlüssel

```
sqlString = "SELECT " & relatedTable & ".ID, " & relatedTable &
"." & relatedField & ", " & relatedTable2 & "." & relatedField2 &
" FROM " & relatedTable2 & " INNER JOIN " & relatedTable & " ON "
& relatedTable2 & ".ID = " & relatedTable & "." & linkField2 & "
WHERE ((((" & relatedTable & "." & relatedField & ")=" &
relatedValue & "') AND ((" & relatedTable2 & "." & relatedField2
& ")=" & relatedValue2 & "'))"
```

Case 3

'Fall 3: die verlinkte Tabelle hat einen zusammengesetzten
Primärschlüssel, der auf eine weitere Tabelle verlinkt

```
sqlString = "SELECT " & relatedTable & ".ID, " & relatedTable &
"." & relatedField & ", " & relatedTable2 & "." & relatedField2 &
", " & relatedTable3 & "." & relatedField3 & " FROM (" &
relatedTable3 & " INNER JOIN " & relatedTable2 & " ON " &
relatedTable3 & ".ID = " & relatedTable2 & "." & linkField3 & ")
INNER JOIN " & relatedTable & " ON " & relatedTable2 & ".ID = " &
relatedTable & "." & linkField2 & " WHERE ((((" & relatedTable &
"." & relatedField & ")=" & relatedValue & "') AND ((" &
relatedTable2 & "." & relatedField2 & ")=" & relatedValue2 & "')
AND ((" & relatedTable3 & "." & relatedField3 & ")=" &
relatedValue3 & "'))"
```

End Select

Try

'Schleife ggf. 2x durchlaufen, wenn Tabelle zwei Linkfelder besitzt

For i = 0 To doubleLink

'Inhalt aus Datenbank auslesen

cmd.Connection = conn

cmd.CommandText = sqlString

dataReader = cmd.ExecuteReader()

If dataReader.Read() Then

If Not sqlInsertColumn = "" Then

sqlInsertColumn = sqlInsertColumn & ", " & linkField

```
        sqlInsertValue = sqlInsertValue & ", " & dataReader(0)
    End If
Else
    'Ausgabe einer Meldung und Aktivierung des Buttons zum
    Quittieren
    lblMessage.Text = "Fehler bei der Datenbankabfrage."
    btnMessage.Enabled = True
End If

'dataReader schließen
dataReader.Close()

'Ersetze Felder für zweiten Durchlauf
If doubleLink Then
    sqlString = "SELECT " & relatedTable2 & ".ID FROM " &
    relatedTable2 & " WHERE " & relatedField2 & "='" &
    relatedValue2 & "'"
    linkField = linkField2
End If
Next

Catch ex As Exception
    'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
    lblMessage.Text = "Fehler in addSQL() " & ex.Message
    btnMessage.Enabled = True
End Try
End If

End Sub

'Datenbankeintrag erzeugen
Private Sub insertSQL(ByVal tableName As String)

'Daten in Datenbank schreiben
Try
    cmd.Connection = conn
    sqlInsert = "INSERT INTO " & tableName & " (" & sqlInsertColumn & ")
    VALUES (" & sqlInsertValue & ")"
    cmd.CommandText = sqlInsert
    cmd.ExecuteNonQuery()
    'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
    lblMessage.Text = "Datensatz erfolgreich eingefügt."
    btnMessage.Enabled = True
End Try
```

```

Catch ex As Exception
    'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
    lblMessage.Text = "Fehler in insertSQL(): " & sqlInsert &
    ex.Message.ToString
    btnMessage.Enabled = True
End Try

End Sub

'Importiertes Gerät auswählen und anzeigen
Private Sub showImport()

    'Felder leeren, Buttons zurücksetzen
    clearFields()
    btnSelect.Enabled = False
    btnSave.Enabled = False
    btnExit.Enabled = False
    ListBoxSerialNumber.Items.Clear()
    ListBoxSerialNumber.Enabled = False
    ListBoxModelNumber.Items.Clear()
    ListBoxModelNumber.Enabled = False
    ListBoxManufacturer.Items.Clear()
    ListBoxManufacturer.Enabled = True
    preSelect = True

    'Update ListBoxManufacturer
    dbConnection()
    cmd.Connection = conn
    updateDropDownManufacturer()
    conn.Close()

    'Update ListBoxModelNumber
    ListBoxManufacturer_SelectedIndexChanged(Nothing, EventArgs.Empty)

    'Update ListBoxSerialNumber
    ListBoxModelNumber_SelectedIndexChanged(Nothing, EventArgs.Empty)

    'Update showSelection
    showSelection()

    'Variable für automatische Selektion nach Import zurücksetzen
    preSelect = False
End Sub

```

```
'Historisierung von Steuerbefehlen
Private Sub history(ByVal newValue As String)

    Try
        'Datenbankverbindung öffnen
        dbConnection()
        cmd.Connection = conn

        'stateVariableID und stateVariableValue auslesen
        Dim sqlString As String
        sqlString = "SELECT serviceStateTable.ID,
serviceStateTable.stateVariableValue,
serviceStateTable.stateVariableName, deviceObject.ID FROM
deviceObject INNER JOIN serviceStateTable ON deviceObject.ID =
serviceStateTable.deviceObjectID WHERE
    ((serviceStateTable.stateVariableName)='Power') AND
    ((deviceObject.ID)=" & Me.ListBoxSerialNumber.SelectedValue & ")"

        cmd.CommandText = sqlString
        dataReader = cmd.ExecuteReader()

        If dataReader.Read() Then
            If IsDBNull(dataReader(0)) Then
                stateVariableID = ""
            Else
                stateVariableID = dataReader(0)
            End If

            If IsDBNull(dataReader(1)) Then
                oldValue = "Gerätezustand nicht gesetzt"
            Else
                oldValue = dataReader(1)
            End If

        Else
            'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
            lblMessage.Text = "Fehler bei der Datenbankabfrage."
            btnMessage.Enabled = True
        End If
        dataReader.Close()
    End Try
End Sub
```

```

'Aktion historisieren
sqlString = "INSERT INTO history (stateVariableID, dateTimeStamp,
oldValue, newValue) VALUES (" & stateVariableID & ", '" &
DateAndTime.Now & "', '" & oldValue & "', '" & newValue & "')"
cmd.CommandText = sqlString
cmd.ExecuteNonQuery()

'Variablenwert aktualisieren
sqlString = "UPDATE serviceStateTable SET stateVariableValue='" &
newValue & "' WHERE ID=" & stateVariableID
txtValue.Text = newValue
cmd.CommandText = sqlString
cmd.ExecuteNonQuery()

'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
lblMessage.Text = "Steuerbefehl gesendet, Historisierung
erfolgreich."
btnMessage.Enabled = True

'dataReader schließen
dataReader.Close()
conn.Close()

Catch ex As Exception
'Ausgabe einer Meldung und Aktivierung des Buttons zum Quittieren
lblMessage.Text = "Steuerbefehl gesendet, Fehler bei der
Historisierung."
btnMessage.Enabled = True
End Try

End Sub

End Class

```


II. Literaturverzeichnis

Aizo AG. (November 2011). *digitalSTROM Installationshandbuch*. Schlieren, Schweiz.

Aizo AG. (2012). *Aizo - Technologie - System*. Abgerufen am 10. Januar 2013 von www.aizo.com: http://www.aizo.com/de/technologie/tech_system.php

Aizo AG. (2012). *digitalSTROM - Systemübersicht*. Abgerufen am 10. Januar 2013 von www.digitalstrom.com: <http://www.digitalstrom.com/System/UEbersicht/>

Arbeitskreis Maschinen- und Elektrotechnik staatlicher und kommunaler Verwaltungen (AMEV). (2005). *Hinweise für Planung, Ausführung und Betrieb der Gebäudeautomation in öffentlichen Gebäuden*. Berlin.

ASHRAE SSPC 135. (2013). *Overview*. Abgerufen am 06. Januar 2013 von www.bacnet.org: <http://www.bacnet.org/Overview/index.html>

Bart, J. (2013). *XML-Deklaration*. Abgerufen am 28. Dezember 2012 von www.selfxml.de: <http://www.selfxml.de/index.php/deklaration>

BauNetz Media GmbH. (2013). *Baunetzwissen Elektro - BACnet-Bussystem*. Abgerufen am 12. Januar 2013 von www.baunetzwissen.de: http://www.baunetzwissen.de/standardartikel/Elektro_BACnet-Building-Automation-and-Control-network_153116.html

BauNetz Media GmbH. (2013). *Baunetzwissen Elektro - Ebenen der Gebäudeautomation*. Abgerufen am 12. Januar 2013 von www.baunetzwissen.de: http://www.baunetzwissen.de/standardartikel/Elektro-Ebenen-der-Gebaeudeautomation_1644069.html

BauNetz Media GmbH. (2013). *Baunetzwissen Elektro - Funk-Bussysteme*. Abgerufen am 12. Januar 2013 von www.baunetzwissen.de: http://www.baunetzwissen.de/standardartikel/Elektro_Funk-Bussysteme_153114.html

BauNetz Media GmbH. (2013). *Baunetzwissen Elektro - Grundprinzip und Elemente von Bussystemen*. Abgerufen am 12. Januar 2013 von www.baunetzwissen.de: http://www.baunetzwissen.de/standardartikel/Elektro_Bussysteme-ein-Grundprinzip-mit-vielen-Vorteilen_153102.html

Behaneck, M. (2012). *Cloud Computing - In der Wolke rechnen*. Deutsches Ingenieurblatt.

BITKOM (Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V.). (2009). *BITKOM-Leitfaden: Cloud Computing - Evolution in der Technik, Revolution im Business*. Berlin.

- DATACOM Buchverlag GmbH.** (2013). *ITWissen - GA (Gebäudeautomation)*. Abgerufen am 20. 01 2013 von www.itwissen.info: <http://www.itwissen.info/definition/lexikon/Gebaeudeautomation-building-automation-GA.html>
- DATACOM Buchverlag GmbH.** (2013). *ITWissen - Middleware*. Abgerufen am 11. 03 2013 von www.itwissen.info: <http://www.itwissen.info/definition/lexikon/Middleware-middleware.html>
- Deutsches Institut für Normung e.V.** (August 2000). DIN 32736. *Gebäudemanagement*. Berlin.
- Deutsches Institut für Normung e.V.** (Oktober 2004). DIN EN ISO 16484-2. *Systeme der Gebäudeautomation (GA) - Teil 2: Hardware*. Berlin.
- Deutsches Institut für Normung e.V.** (Januar 2007). DIN EN 15221-1. *Facility Management – Teil 1: Begriffe*. Berlin.
- Deutsches Institut für Normung e.V.** (September 2011). DIN EN 50173. *Informationstechnik - Anwendungsneutrale Kommunikationskabelanlage*. Berlin.
- Deutsches Institut für Normung e.V.** (Februar 2012). DIN EN 13321-2. *Offene Datenkommunikation für die Gebäudeautomation und Gebäudemanagement – Elektrische Systemtechnik für Heim und Gebäude – Teil 2: KNXnet/IP Kommunikation*. Berlin.
- Deutsches Institut für Normung e.V.** (2013). *Dossier: Energieeffizienz*. Abgerufen am 02. Januar 2013 von <http://www.din.de/cmd?level=tpl-unterrubrik&menuid=47392&cmsareaid=47392&cmsrubid=47536&menurubricid=47536&cmssubrubid=132106&menusubrubid=132106&languageid=de>
- Dietrich, D., Loy, D. & Schweinzer, H.-J.** (1998). *LON-Technologie*. Wien: Hüthig Verlag Heidelberg.
- EnOcean.** (2013). *Batterielose Funktechnologie*. Abgerufen am 24. Januar 2013 von www.enocean.com: <http://www.enocean.com/de/batterielose-funktechnologie/>
- EnOcean.** (2013). *Energiewandler*. Abgerufen am 24. Januar 2013 von www.enocean.com: www.enocean.com/de/energiewandler
- EnOcean.** (2013). *Funkstandard*. Abgerufen am 24. Januar 2013 von www.enocean.com: <http://www.enocean.com/de/enOcean-funkstandard/>
- EnOcean.** (2013). *Unternehmensprofil*. Abgerufen am 24. Januar 2013 von www.enocean.com: <http://www.enocean.com/de/unternehmensprofil/>
- EnOcean Alliance.** (2013). *Hersteller und Produkte*. Abgerufen am 24. Januar 2013 von www.enocean-alliance.org: <http://www.enocean-alliance.org/de/produkte/>

- Furrer, F.** (2003). *Industrieautomation mit Ethernet-TCP/IP und Web-Technologie*. Heidelberg: Hüthig GmbH & Co. KG.
- Gartner.** (Mai 2012). *Magic Quadrant for Integrated Workplace Management Systems*.
- German Facility Management Association.** (1996). GEFMA-Richtlinie 100. *Facility Management - Begriff, Struktur, Inhalte*. Bonn.
- German Facility Management Association.** (2004). GEFMA-Richtlinie 100-1. *Facility Management - Grundlagen*. Bonn.
- German Facility Management Association.** (2007). GEFMA-Richtlinie 410. *Schnittstellen zur IT-Integration von CAFM-Software*. Bonn.
- German Facility Management Association.** (2009). GEFMA-Richtlinie 450 (ersetzt durch VDI/GEFMA-Richtlinie 3814 Blatt 3.1). *Gebäudeautomation im FM*. Bonn.
- German Facility Management Association.** (2012). GEFMA-Richtlinie 940. *Marktübersicht CAFM Software*. Bonn.
- German Facility Management Association.** (2013). *GEFMA - Computer Aided Facility Management*. Abgerufen am 12. Januar 2013 von www.gefma.de: www.gefma.de/cafm.html
- German Facility Management Association.** (2013). GEFMA-Richtlinie 400. *CAFM - Begriffsbestimmungen, Leistungsmerkmale*. Bonn.
- Harnisch, C.** (2009). *Netzwerktechnik*. Heidelberg: Verlagsgruppe Hüthig Jehle Rehm GmbH.
- Hein, M. & Dr. Maciejewski, B.** (2003). *Wireless LAN, Funknetze in der Praxis*. Poing: Franzis' Verlag GmbH.
- Hermos.** (2013). *Hermos FIS#Facility*. Abgerufen am 11. März 2013 von Facility und Building Automation: http://www.hermos.com/index.php?option=com_content&view=article&id=80&Itemid=75&lang=de
- Hohmann, J.** (27. Februar 2013). Fachvortrag CAFM-Trendstudie. *Erfahrungen & Entwicklungen aus Anwendersicht*. Frankfurt.
- HomePlug Alliance.** (2013). *Whitepapers*. Abgerufen am 10. Februar 2013 von www.homeplug.org: www.homeplug.org/tech/whitepapers/
- Institute of Electrical and Electronics Engineers.** (2010). IEEE 1901-2010. *Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications*. Institute of Electrical and Electronics Engineers.

- International Facility Management Association.** (2013). *IFMA - What is FM?* Abgerufen am 24. Januar 2013 von www.ifma.org: www.ifma.org/know-base/browse/what-is-fm-
- Internet Engineering Task Force (IETF) - Network Working Group.** (Februar 1996). RFC1918. *Address Allocation for Private Internets.*
- Invensys Systems GmbH.** (2013). *Wonderware in der Gebäudeleittechnik - Industrial Operations.* Abgerufen am 11. März 2013 von global.wonderware.com: http://global.wonderware.com/DE/Pages/WonderwareGLT_Industrial.aspx
- Issendorff KG.** (Dezember 2009). *LCN - Gebäudeleittechnik (Systembeschreibung).* Rethen.
- IT-Administrator.** (2013). *Universal Plug and Play.* Abgerufen am 24. Februar 2013 von www.it-administrator.de: http://www.it-administrator.de/lexikon/universal_plug_and_play.html
- Kleijn, A.** (12. September 2009). *IEEE ratifiziert WLAN-Standard 802.11n.* Abgerufen am 24. Februar 2013 von www.heise.de: <http://www.heise.de/newsticker/meldung/IEEE-ratifiziert-WLAN-Standard-802-11n-755625.html>
- KNX Association.** (2013). *KNX Association - Introduction.* Abgerufen am 14. Januar 2013 von www.knx.org: <http://www.knx.org/de/knx-association/introduction/>
- KNX Association.** (2013). *KNX Standard - Einführung.* Abgerufen am 14. Januar 2013 von www.knx.org: <http://www.knx.org/de/knx-standard/einfuehrung/>
- KNX Association.** (2013). *KNX Standard - Kommunikationsmedien.* Abgerufen am 14. Januar 2013 von www.knx.org: <http://www.knx.org/de/knx-standard/communication-media/>
- LonMark International.** (Dezember 2008). *LON Archives ISO/IEC Standardization.* Abgerufen am 12. Januar 2013 von www.lonmark.org: http://www.lonmark.org/news_events/press/2008/1208_iso_standard
- May, M.** (2013). *CAFM-Handbuch.* Berlin: Springer Vieweg.
- Merz, H., Hansemann, T. & Hübner, C.** (2010). *Gebäudeautomation, Kommunikationssysteme mit EIB/KNX, LON und BACnet.* München: Carl Hanser Verlag.
- Microsoft Corporation.** (Juni 2000). *White Paper - Understanding Universal Plug and Play.* Redmond: http://www.upnp.org/download/UPNP_understandingUPNP.doc.
- Microsoft TechNet.** (2013). *Universal Plug and Play (UPnP) Client Support.* Abgerufen am 16. Februar 2013 von technet.microsoft.com: <http://technet.microsoft.com/de-de/library/bb727027.aspx>

- Nävy, J.** (2006). *Facility Management*. Solingen: Springer.
- Netzwerkartikel.de.** (2013). *Cat5-Patchkabel*. Abgerufen am 24. 02 2013 von www.cat5.de - Patchkabel: www.cat5.de/patchkabel/patchkabel-typen.htm
- Rapid7.** (29. Januar 2013). *Whitepaper - Security Flaws in Universal Plug and Play*.
- Siemens Building Technologies.** (17. Februar 2003). *Gebäudeautomation – Begriffe, Definitionen und Abkürzungen*. Siemens Building Technologies.
- Siemens Building Technologies.** (2013). *Optimierung der Gebäudeenergieeffizienz*. Abgerufen am 04. Januar 2013 von www.buildingtechnologies.siemens.com: <http://www.buildingtechnologies.siemens.com/bt/global/de/gebaeudeautomation-hlk/gebaeudeautomationssysteme/applikationen-werkzeuge/Seiten/gebaeudeeffizienzanalyse.aspx>
- Silies, H.** (26. Januar 2012). *Cloud Computing: Die Zukunft liegt in den Wolken*. (c. e. GmbH, Hrsg.) Abgerufen am 31. August 2012 von www.connectiv.de: http://www.connectiv.de/de/aktuelles/news/cloud_computing_die_zukunft_liegt_in_den_wolken.html
- sMOTIVE.** (2013). *sMOTIVE Web-Portal*. Abgerufen am 11. März 2013 von www.smotive.de: <http://www.smotive.de/facility-management-dienstleister-edition.html>
- telekommunikation-online Gutsch & Co. OHG.** (2013). *Was ist WLAN?* Abgerufen am 24. Februar 2013 von www.was-ist-wlan.de: www.was-ist-wlan.de
- Theis, T.** (2008). *Einstieg in Visual Basic 2008*. (G. Computing, Hrsg.)
- Thomzik, M.** (2010). *FM-Branchenreport, IAI e.V.* Bochum.
- Tierling, E.** (14. Dezember 2004). *Tom's Networking Guide, UPnP-Grundlagen*. Abgerufen am 25. Februar 2013 von www.tomsnetworking.de: http://www.tomsnetworking.de/content/reports/j2004a/background_upnp/page2.html
- UPnP Forum.** (15. Oktober 2008). *UPnP Device Architecture 1.1*.
- UPnP Forum.** (2013). *UPnP Device Control Protocols and Device Categories*. Abgerufen am 26. Februar 2013 von www.upnp.org: <http://upnp.org/sdcps-and-certification/standards/sdcps/>
- Verband Deutscher Ingenieure e.V.** (Mai 2011). *VDI Richtlinie 3813-2. Gebäudeautomation (GA) - Raumautomationsfunktionen (RA-Funktionen)*. Düsseldorf.
- Verband Deutscher Ingenieure e.V.** (September 2012). *VDI/GEFMA-Richtlinie 3814, Blatt 3.1. Gebäudeautomation (GA), Hinweise für das Gebäudemanagement, Planung, Betrieb und Instandhaltung, Schnittstelle zum Facility Management*. Düsseldorf.

Widmer, L. (2009). digitalSTROM - Das Internet der elektrischen Geräte. *Strompraxis-Sonderheft*.

Zentralverband Elektrotechnik- und Elektronikindustrie e.V. (2013). *Initiative Intelligentes Wohnen*. Abgerufen am 11. März 2013 von www.intelligenteswohnen.com:
http://www.intelligenteswohnen.com/iw_de/presse/bilder.php?navanchor=2110051

Zentralverband Elektrotechnik- und Elektronikindustrie e.V., Zentralverband der Deutschen Elektro- und Informationstechnischen Handwerke. (2006). *Handbuch Haus- und Gebäudesystemtechnik, Grundlagen*. Frankfurt.

III. Abbildungsverzeichnis

Abbildung 1:	IST-Zustand, eigene Darstellung mit Bildmaterial der Initiative Intelligentes Wohnen, www.intelligenteswohnen.com	19
Abbildung 2:	SOLL-Zustand, eigene Darstellung mit Bildmaterial der Initiative Intelligentes Wohnen, www.intelligenteswohnen.com	21
Abbildung 3:	Phasenmodell, eigene Darstellung.....	23
Abbildung 4:	Gliederung nach GEFMA-Richtlinie 100, eigene Darstellung.....	27
Abbildung 5:	Abgrenzung Facility-/Gebäudemanagement, Quelle: GEFMA-Richtlinie 100-1, 2004.....	29
Abbildung 6:	Beispiele von üblichen CAFM-Schnittstellen, eigene Darstellung.....	30
Abbildung 7:	Realisierte Schnittstellen von CAFM-Systemen, Quelle: Hohmann, Fachvortrag CAFM-Trendstudie 2013.....	33
Abbildung 8:	Ausweitung der Einsatzbereiche von CAFM, Quelle: Hohmann, Fachvortrag CAFM-Trendstudie 2013.....	33
Abbildung 9:	Client/Server-Modell, eigene Darstellung.....	35
Abbildung 10:	Webbasierte Lösung, eigene Darstellung.....	36
Abbildung 11:	Cloud Computing, eigene Darstellung.....	37
Abbildung 12:	Energieeffizienz durch Gebäudeautomation, Quelle: Siemens Building Technologies	40
Abbildung 13:	Funktionsebenen der Gebäudeautomation, eigene Darstellung mit Bildmaterial aus Online-Produktkatalog Fa. Sauter AG.....	42
Abbildung 14:	Integration technischer und kaufmännischer Prozesse ohne und mit Middleware, eigene Darstellung.....	43
Abbildung 15:	Softwarekomponenten eines CAFM-Systems, Quelle: GEFMA-Richtlinie 410.....	48

Abbildung 16: Buskabel, Quelle: Online-Produktkatalog Fa. VOKA.....	54
Abbildung 17: KNX-Kabel, Quelle: Online-Produktkatalog Fa. VOKA.....	56
Abbildung 18: Topologie eines KNX-Netzwerks nach DIN EN 13321-2, eigene Darstellung.....	57
Abbildung 19: Aufbau der Adresse eines KNX-Teilnehmers, eigene Darstellung.....	58
Abbildung 20: Aufbau eines KNX-Standarddatentelegramms, eigene Darstellung.....	59
Abbildung 21: Topologie eines KNXnet/IP-Netzwerks nach DIN EN 13321-2, eigene Darstellung.....	59
Abbildung 22: KNXnet/IP-Controller aus dem Wago I/O-System, Quelle: Online-Produktkatalog Fa. Wago.....	60
Abbildung 23: Aufbau eines Neuron-Chips nach Dietrich, Loy & Schweinzer („LON-Technologie“, 1998), eigene Darstellung.....	62
Abbildung 24: Raumcontroller mit Klappenstellantrieb, Quelle: Online-Produktkatalog Fa. Schneider Electric Buildings	62
Abbildung 25: LON- Kabel, Quelle: Online-Produktkatalog Fa. VOKA.....	62
Abbildung 26: Gemischte Topologie eines LON-Netzwerks, eigene Darstellung.....	63
Abbildung 27: Aufbau des LonTalk-Protokolls auf Grundlage des ISO/OSI-Referenzmodells nach Dietrich, Loy & Schweinzer („LON-Technologie“, 1998), eigene Darstellung	64
Abbildung 28: LON-Topologie beim Einsatz von IP-Routern, eigene Darstellung.....	65
Abbildung 29: LON IP-Router, Quelle: Online-Produktkatalog Fa. Loytec	66
Abbildung 30: Busmodul LCN-UPP für Unterputzdose, Quelle: Online-Produktkatalog Fa. Issendorff KG	67

Abbildung 31: NYM-Kabel, Quelle: OnlineProduktkatalog Fa. VOKA	68
Abbildung 32: Topologie eines LCN-Bus, eigene Darstellung.....	68
Abbildung 33: Aufbau eines LCN-Datentelegramms, eigene Darstellung.....	69
Abbildung 34: Buskoppler LCN-PK (RS232-Schnittstelle), Quelle: Online-Produktkatalog Fa. Issendorff KG	69
Abbildung 35: Aufbau des BACnet-Protokolls auf Grundlage des ISO/OSI-Referenzmodells, eigene Darstellung.....	71
Abbildung 36: BACnet/IP-Controller (KNX/LON-Gateway), Quelle: Online-Produktkatalog Fa. Wago.....	72
Abbildung 37: Kommunikation im digitalSTROM-Gesamtsystem nach Aizo AG, eigene Darstellung mit Bildmaterial aus Online-Produktkatalog Fa. Aizo AG	74
Abbildung 38: digitalSTROM-Chip (dSID), Quelle: Online-Produktkatalog der Aizo AG (Foto nachbearbeitet)	75
Abbildung 39: Lüster-Klemme mit verbautem Chip, Quelle: Online-Produktkatalog der Aizo AG.....	75
Abbildung 40: Schematische Darstellung des digitalSTROM-Netzes, Quelle: Aizo AG (digitalSTROM - Systemübersicht, 2012)	76
Abbildung 41: Ratio©-Temperatursensor, Quelle: Online-Produktkatalog Fa. Akktor GmbH	77
Abbildung 42: Ratio©-Thermostataktor, Quelle: Online-Produktkatalog Fa. Akktor GmbH	77
Abbildung 43: Energiewandler mit EnOcean-Technologie, Quelle: EnOcean.....	78
Abbildung 44: Ratio©-Gateway, Quelle: Online-Produktkatalog Fa. Omnio AG	78
Abbildung 45: Funktionsweise der batterielosen Funkübertragung nach der Beschreibung von EnOcean, eigene Darstellung	79
Abbildung 46: Aufbau des EnOcean-Protokolls auf Grundlage des ISO/OSI-Referenzmodells, eigene Darstellung.....	80

Abbildung 47: Internetanbindung von EnOcean-Geräten via BACnet/IP-Controller, Quelle: Produktkatalog „Automation“ Fa. Wago	81
Abbildung 48: LAN-Kabel, Quelle: Online-Produktkatalog Fa. ADP-Cable	84
Abbildung 49: RJ45-Stecker, Quelle: Online-Produktkatalog Fa. Preistaktik	84
Abbildung 50: PowerLAN-Adapter, Quelle: Online-Produktkatalog Fa. MSI-Computer	86
Abbildung 51: Access Point, Quelle: Online-Produktkatalog Fa. TD-Link	87
Abbildung 52: TCP/IP-Protokoll auf Grundlage des ISO/OSI-Referenzmodells, eigene Darstellung.....	91
Abbildung 53: Geschachteltes Protokoll für die TCP/IP-Schichten nach Furrer („Industrieautomation mit Ethernet-TCP/IP und Web-Technologie“, 2003), eigene Darstellung.....	92
Abbildung 54: Klassische Netzwerkstruktur eines CAFM-Systems, eigene Darstellung.....	94
Abbildung 55: Unidirektionale GA-Anbindung über eine lokale Datenschnittstelle zwischen CAFM-System und Bussystem mit Leitreehner, eigene Darstellung.....	96
Abbildung 56: RS232/RS485-Converter, Quelle: Online-Produktkatalog Fa. Abus	97
Abbildung 57: ISA-Bus-Controller, Quelle: Online-Produktkatalog Fa. Sauter	98
Abbildung 58: Unidirektionale GA-Anbindung über eine webbasierte Datenschnittstelle zwischen CAFM-System und Bussystem mit Leitreehner, eigene Darstellung.....	99
Abbildung 59: Bidirektionale GA-Anbindung über den seriellen Bus (USB/COM-Port) des CAFM-Servers, eigene Darstellung	101
Abbildung 60: Anlagenschema Ratio©-Funkbussystem, eigene Darstellung mit Bild- material aus Online-Produktkatalogen Fa. Omnio AG und Akktor GmbH...	102
Abbildung 61: Benutzeroberfläche von myHomeControl, Quelle: BootUp GmbH.....	103

Abbildung 62: Bidirektionale GA-Anbindung über das interne LAN-Netzwerk durch netzwerkfähige Endgeräte, eigene Darstellung.....	104
Abbildung 63: Bidirektionale GA-Anbindung über das öffentliche WAN-Netzwerk durch internetfähige Endgeräte, eigene Darstellung.....	107
Abbildung 64: Aufbau des gewählten Integrationsmodells, eigene Darstellung.....	115
Abbildung 65: Dynamic DNS, Screenshot aus eigener Speedport-Konfiguration.....	118
Abbildung 66: Portweiterleitung, Screenshot aus eigener Speedport-Konfiguration.....	119
Abbildung 67: Portumleitung, Screenshot aus eigener Speedport-Konfiguration.....	120
Abbildung 68: UPnP-Prozesskette, eigene Darstellung.....	123
Abbildung 69: Schichten des UPnP-Workings im Vergleich zu TCP/IP-Diensten, eigene Darstellung.....	125
Abbildung 70: Flussdiagramm Geräteanbindung, eigene Darstellung.....	137
Abbildung 71: Flussdiagramm Datenbankabfrage, Datenbankeintrag und Gerätesteuerung, eigene Darstellung.....	138
Abbildung 72: Tabelle deviceClass.....	152
Abbildung 73: Tabelle deviceObject.....	153
Abbildung 74: Tabelle serviceStateTable.....	153
Abbildung 75: Tabelle history.....	153
Abbildung 76: Tabelle serviceActionList.....	154
Abbildung 77: Tabelle serviceArgumentList.....	154

Abbildung 78: Tabelle deviceCategory	154
Abbildung 79: Tabelle contractData	154
Abbildung 80: Tabelle contactData	155
Abbildung 81: Tabelle locrefBuilding	155
Abbildung 82: Tabelle locrefFloor	155
Abbildung 83: Tabelle locrefRoom	155
Abbildung 84: Struktur der relationalen Datenbank	156
Abbildung 85: Internetinformationsdienst(IIS)-Manager, Screenshot aus eigener Konfiguration	185
Abbildung 86: IP-Steckdosenleiste Netio-230B, Quelle: Online-Produktkatalog Fa. Koukaam	186
Abbildung 87: Webbasierte Benutzeroberfläche von "FMControl", Screenshot der eigenen Entwicklung	187
Abbildung 88: Registrierung eines neuen Geräts, Screenshot der eigenen Netzwerkumgebung	189
Abbildung 89: Auswahlmenü für Import, Screenshot der eigenen Entwicklung	189
Abbildung 90: Anlagenmaske nach Import, Screenshot der eigenen Entwicklung	190
Abbildung 91: Datenbankeinträge, Screenshot der eigenen Datenbank	191
Abbildung 92: Maskenbereich Geräte-/Anlagensteuerung, Screenshot der eigenen Entwicklung	192
Abbildung 93: Datensätze zur Historisierung, Screenshot der eigenen Datenbank	193

IV. Tabellenverzeichnis

Tabelle 1:	GEFMA-zertifizierte CAFM-Unternehmen und -Software, Quelle: GEFMA-Richtlinie 940.....	34
Tabelle 2:	Vergleich von IEEE 802.11-Standards	88
Tabelle 3:	Vergleich von Funktechnologien	89
Tabelle 4:	Entscheidungsmatrix.....	112
Tabelle 5:	Liste ausgewählter Standard-Ports	121
Tabelle 6:	Vordefinierte Eigenschaften der Anlagenklasse	141
Tabelle 7:	Vordefinierte Eigenschaften des Anlagenobjekts.....	141
Tabelle 8:	Benutzerdefinierte Eigenschaften des Anlagenobjekts.....	141
Tabelle 9:	Vordefinierte Services der Anlagenklasse	142
Tabelle 10:	Vordefinierte Variablen des Anlagenobjekts	142
Tabelle 11:	Funktionen der Anlagenklasse (<action>)	145
Tabelle 12:	Parameter der Aktionen (<argument>)	148
Tabelle 13:	Variablen des Anlagenobjekts (<stateVariable>).....	150

V. Abkürzungsverzeichnis

AMEV	Arbeitskreis Maschinen- und Elektrotechnik staatlicher und kommunaler Verwaltungen
BACnet	Building Automation and Control Networks
BRE	Building Research Establishment
BREEAM	BRE Environmental Assessment Method
CAD	Computer Aided Design
CAFM	Computer Aided Facility Management
CMMS	Computerized Maintenance Management System
DB	Datenbank
DBMS	Datenbankmanagementsystem
DGNB	Deutsche Gesellschaft für Nachhaltiges Bauen e.V.
DHCP	Dynamic Host Configuration Protocol
dLAN	direct LAN
DynDNS	dynamischer Domain-Name-System-Eintrag
ERP	Enterprise-Resource-Planning
FMA.Codex	Facility Management Automation Control Data Exchange
GA	Gebäudeautomation
GENA	Generic Event Notification Architecture
GIS	Geoinformationssystem
GLT	Gebäudeleittechnik
HKL	Heizung, Klima, Lüftung
HTTP	Hypertext Transfer Protocol
HVAC	Heating, Ventilation and Air Conditioning
IEEE	Institute of Electrical and Electronics Engineers
IIS	Internetinformationsdienst
ISO	International Organization for Standardization
IWMS	Integrated Workplace Management System
KNX	Installationsbus der Konnex-Association
LAN	Local Area Network
LCN	Local Control Network
LEED	Leadership in Energy and Environmental Design
LON	Local Operating Network
ODBC	Open Database Connectivity
OSI	Open Systems Interconnection

PnP	Plug and Play
SOAP	Simple Object Access Protocol
SSDP	Simple Service Discovery Protocol
TCP/IP	Transmission Control Protocol / Internet Protocol
TGA	Technische Gebäudeausrüstung
UDP	User Datagram Protocol
UPnP	Universal Plug and Play
URL	Uniform Resource Locator
USB	Universal Serial Bus
VB.NET	Visual Basic mit .NET-Framework
VDMA	Verband Deutscher Maschinen- und Anlagenbau e.V.
VPN	Virtual Private Network
WAN	Wide Area Network
WLAN	Wireless Local Area Network
XML	Extensible Markup Language

VI. Lebenslauf

BERUF

- seit 04/2013
Leiter Operatives Management (COO) und
Leiter Vertrieb & Marketing (CSO/CMO)
BFM Building + Facility Management GmbH
Eschborn
- 05/2008 – 03/2013
Wissenschaftlicher Mitarbeiter, Fachstudienberater
Lehrstuhl Bauinformatik/Facility Management
Technische Universität Kaiserslautern
- 02/2006 – 03/2013
Freiberufliche Tätigkeit als
selbständiger CAFM-Berater
- 01/2006 – 04/2008
Promotionsstipendiat und
wissenschaftliche Hilfskraft mit Abschluss
Lehrstuhl Bauinformatik/Facility Management
Technische Universität Kaiserslautern

AUSBILDUNG

- 10/1999 – 12/2005
Studium
Technische Universität Kaiserslautern
Fachrichtung Bauingenieurwesen
Abschluss: Diplom-Ingenieur (TU)
- 07/1998 – 09/1998
Ausbildung im Rettungsdienst
Arbeiter-Samariter-Bund
Mannheim
Abschluss: Rettungssanitäter
- 08/1989 – 06/1998
Hohenstaufen-Gymnasium
Kaiserslautern
Abschluss: Allgemeine Hochschulreife
- 08/1985 – 07/1989
Grundschule Hohenecken
Kaiserslautern

ZIVILDIENTST

- 07/1998 – 08/1999
Zivildienst
Arbeiter-Samariter-Bund
Kaiserslautern