

Set-Top-Box-basierte Ambient Assisted Living Plattform

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Medieninformatik

eingereicht von

Miroslav Sili

Matrikelnummer 0525015

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung
Betreuer: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Wolfgang Zagler

Wien, 24.01.2012

(Unterschrift Verfasser)

(Unterschrift Betreuer)

Erklärung zur Verfassung der Arbeit

Miroslav Sili

Dr. Walter Gebhart Gasse 13

A-2540 Bad Vöslau

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Bad Vöslau, am 24.01.2012

(Miroslav Sili, BSc)

Anmerkung zur geschlechtsneutralen Formulierung:

Zugunsten besserer Lesbarkeit wurde in der folgenden Arbeit auf die geschlechterspezifische Schreibweise wie z.B. Benutzer/innen verzichtet. Die folgenden Formulierungen sind selbstverständlich an beide Geschlechter gleichermaßen gerichtet.

Kurzfassung

Diese Diplomarbeit beschäftigt sich mit dem Themenbereich Ambient Assisted Living (AAL) und verfolgt das Ziel eine alternative und benutzerfreundliche Interaktionsmöglichkeit zwischen dem Bewohner einer AAL-Wohnumgebung und dem dahinter befindlichen System zu schaffen. Die Benutzerinteraktion mit solch einem System erfolgt bisher meistens über einen PC oder einen Touchscreens-PC, der an einer zentralen Stelle im Haushalt aufgestellt wird. Diese Diplomarbeit erarbeitet dagegen eine Set-Top-Box-basierte Interaktionsmöglichkeit, die nicht an solch ein Zusatzgerät angewiesen ist, sondern direkt auf dem eigenen HDTV-fähigen TV-Gerät erfolgen kann. Sie ermöglicht sowohl die Steuerung einer Smart-Home-Umgebung, als auch die Alarmierung des Benutzers bei außergewöhnlichen, kritischen Situationen. Ein Bestreben dieser Diplomarbeit liegt in der Erstellung einer generischen Lösung, die bei möglichst vielen AAL-Systemen eingesetzt werden kann. Für diesen Zweck wurden die typischen Charakteristika eines AAL-Systems anhand einiger ausgewählter, im Einsatz befindlicher Plattformen herausgearbeitet. Sie bilden die Grundlagen für das Design der Software die in dieser Diplomarbeit implementiert wurde. Eine genauere Untersuchung von aktuellen Verfahren und Standards auf dem Gebiet digitale Fernsehsignalübertragung und interaktives Fernsehen ermöglicht die Wahl einer Set-Top-Box, die einerseits zukunftsorientiert ist und andererseits europaweit eingesetzt werden kann. Sowohl der Softwaredesignprozess als auch die konkrete Implementierung auf der ausgewählten Set-Top-Box wurden im Detail ausgearbeitet und beschrieben.

Abstract

This thesis deals with the topic of Ambient Assisted Living and aims to provide an alternative and user friendly interaction opportunity between an inhabitant of the Ambient Assisted Living home environment and a system which is located behind this home environment. The usual interaction in such a system is based on a PC or a touchscreen-PC. This thesis provides a Set-Top-Box based user interaction, which is not depended on such additional devices. It can use the inhabitant's high definition TV instead. On the one hand the interaction opportunity should be able to control the home environment and on the other hand it should be able to warn the inhabitant about an unusual and critical situation. This thesis aims to develop a generic solution, which may be used in various AAL systems. The main characteristics of an AAL system are evaluated on the basis of some selected and platforms that are still in use. They represent the groundwork for the software development step. A detail exploration of currently used proceedings and standards in the field of digital video broadcasting and interactive television enables the right selection of a Set-Top-Box that may be also used in the future and at any European location. The software design steps as well as the implementation steps are discussed in detail.

Inhaltsverzeichnis

1. Einleitung	8
2. Bestehende AAL - Systeme und die Softwareanforderungen	11
2.1. Bestehende AAL-Systeme	11
2.1.1. Nicht TV-basierte Systeme	11
2.1.1.1. HOMER Plattform	11
2.1.1.2. eHome Plattform	12
2.1.2. TV-basierte AAL-Systeme	13
2.1.2.1. HOMEBUTLER	13
2.1.2.2. HERA-project	14
2.1.3. Vergleich der Systeme	15
2.2. Softwareanforderungen	16
2.2.1. Funktionale Anforderungen	16
2.2.1.1. Integration in bestehende AAL-Backend-Systeme	16
2.2.1.2. Wiedergabe von Warn- und Ereignismeldungen, Home-Monitoring	16
2.2.1.3. Audiovisuelle Ausgabe	16
2.2.1.4. Abfrage von Sensoren und Steuerung von Geräten, Home-Control	17
2.2.1.5. Bedienbarkeit mit Hilfe einer Fernbedienung	17
2.2.2. Nichtfunktionale Anforderungen	17
2.2.2.1. Einfache Adaptierung	18
2.2.2.2. One-Box-System	18
2.2.2.3. Open-Source-System	18
2.2.2.4. Parametrisierung	18
3. Anforderungen an die Hardware	19
3.1. Empfangsmöglichkeiten für TV - Signale	19
3.1.1. Terrestrisch-, kabel- und satellitenbasierte TV - Signalübertragung	19
3.1.2. Fernsehen über das Internet	20
3.1.3. Mobiles Fernsehen	21
3.2. Interaktivität	22
3.2.1. Charakteristika des interaktiven Fernsehens	22
3.2.2. AAL-Funktionalität als interaktiver Dienst	24
3.2.3. Standards am Sektor interaktive Dienste	25
3.2.3.1. Multimedia and Hypermedia Experts Group (MHEG) - Standard	25
3.2.3.2. Multimedia Home Platform (MHP) - Standard	26
3.2.3.3. Hybrid Broadcast Broadband TV (HbbTV) - Standard	26
3.2.4. Die Einsatzmöglichkeiten der vorgestellten Standards für die AAL - Plattform	27
3.3. Der Personal Computer (PC) als TV – Gerät	28
3.4. Smart-TV / Hybrid-TV / Connected-TV	30
3.4.1. AAL-Frontend in der Smart-TV-Umgebung	31
3.4.2. Vergleich verschiedener Smart-TV-Gerätehersteller	32
3.4.3. Google TV	34
3.5. Set-Top-Boxen	35
3.5.1. Dreambox & Enigma	35
3.5.2. ReelBox & VDR	36
3.6. Entscheidungskriterien für die ReelBox & VDR-Software	37
4. Softwaredesign	38
4.1. AAL - Funktionalität als VDR-Plugin	38
4.2. AAL-Plattform als ein verteiltes System	38
4.2.1. Auswahlkriterien für einen HTTP-Webserver	40
4.2.2. Der Mongoose Webserver	41
4.3. Umgang mit verschiedenartigen Meldungen	41
4.3.1. Klassifizierung von Meldungen	41
4.3.1.1. Alarmmeldung	41

4.3.1.2.	Warnmeldung	42
4.3.1.3.	Informationsmeldung	42
4.3.1.4.	Stille Meldung	42
4.3.1.5.	Die Home-Control - Aktivität	42
4.3.2.	Regelwerk für die Aufzeichnung und die Anzeige von Meldungen	42
4.3.2.1.	Priorität	43
4.3.2.2.	Speicherart	43
4.3.2.3.	Anzeigeart	44
4.3.2.4.	Anzeigedauer	44
4.4.	Steuerung	44
4.4.1.	Ziffernblock als Auswahl Tasten	45
4.4.2.	Softkey-Tasten als Kontrolltasten	45
4.5.	Visualisierung	45
4.5.1.	Strukturelle Aspekte	46
4.5.1.1.	Übersichtsbereich	47
4.5.1.2.	Inhaltsbereich	47
4.5.1.3.	Steuerbereich	49
4.5.2.	Technische Aspekte	50
4.6.	Akustische Ausgabe	51
4.6.1.	Wiedergabe von Audiodateien	51
4.6.2.	Text-To-Speech-Ausgabe	51
4.6.2.1.	Kommunikationsstrecke AAL-Plugin - TTS-Engine	52
4.6.2.2.	Auswahlkriterien für eine TTS-Engine	53
4.6.2.3.	Die Wahl von zwei TTS-Systemen	54
4.7.	Abbildung des Settings	55
4.7.1.	AALItem	55
4.7.2.	Lokale Abbildung vs. Just-in-time-Anforderung über das Netzwerk	56
4.7.2.1.	Just-in-time - Anforderung	57
4.7.2.2.	Lokale Abbildung	57
4.7.3.	Der Einsatz von Scalable Vector Graphics für die Abbildung des Settings	58
4.7.3.1.	Die Vorteile des SVG-Formats für die Abbildung des Settings	59
4.7.3.2.	Definition eines AALItems innerhalb der SVG-Datei	59
4.7.4.	Die SVG - Verarbeitungskette	61
4.8.	Sicherheit	62
4.9.	Usability und Anwenderfreundlichkeit	62
5.	Implementierung	64
5.1.	Der Model-Viewer-Controller - Ansatz	64
5.2.	Die Kommunikation mit dem AAL-Backend	65
5.2.1.	Frontend zu Backend Kommunikation	66
5.2.2.	Backend zu Frontend Kommunikation	67
5.3.	Klassendiagramm	68
5.4.	Klassenbeschreibung	69
5.5.	Verwendete Bibliotheken und Applikationen	76
5.5.1.	Apache log4cxx	76
5.5.2.	GraphicsMagick	76
5.5.3.	Apache Xerces XML-Parser	76
5.5.4.	Inkscape Applikation	77
5.5.5.	Mongoose Webserver	77
5.5.6.	eSpeak und MARY	77
5.5.7.	Libcurl	77
5.6.	Quellcode und die Installation auf der ReelBox	78
5.6.1.	Quellcode	78
5.6.2.	Installation auf der ReelBox	78
5.7.	vdr4aal-Plugin im Einsatz	79
6.	Zusammenfassung	82

Literaturverzeichnis	84
Abbildungsverzeichnis	89
Tabellenverzeichnis	90
Sourcecode Listings	91
Abkürzungs- und Akronymverzeichnis	92

1. Einleitung

Die Altersstruktur der Bevölkerung ist einem Wandel unterzogen. Während die Zahl an erwerbstätigen Personen im Alter zwischen 15 und 60 rückläufig ist, nimmt zugleich die Anzahl der über 60-jährigen Menschen stetig zu. Im Jahr 2010 machte die Altersgruppe 60+ etwa 21% der Österreichischen Bevölkerung aus. Laut langfristigen Prognosen wird diese Altersgruppe im Jahr 2018 rund 25%, im Jahr 2027 etwa 30% und schließlich im Jahr 2050 einen Anteil von rund 34% der österreichischen Bevölkerung ausmachen (Statistik Austria, 2011). Länderübergreifende Prognosen für die Gesamtweltbevölkerung kommen zwar nicht zu solchen drastischen, aber dennoch zu ähnlichen Werten. Sie prognostizieren, dass die Altersgruppe 60+ im Jahr 2050 einen Anteil von 22% der Gesamtweltbevölkerung ausmachen wird (United Nations, 2010).

Dieser Wandel hat weitreichende Folgen für die Gesellschaft und erfordert soziale sowie wirtschaftliche Anpassungen. Bedingt durch den Alterungsprozess der Gesellschaft steigen die Kosten für das Sozial- und Gesundheitswesen. Diese Kosten können jedoch von der zugleich schrumpfenden Erwerbsbevölkerung nicht kompensiert werden. Ambient Assisted Living (AAL)-Technologien helfen dabei, sowohl die Kosten für das Sozial- und Gesundheitswesen zu senken, als auch die Bedürfnisse von älteren Menschen zu bewahren. Eine AAL-Umgebung stellt den Menschen und seine Bedürfnisse nach Unabhängigkeit und Gesundheit in den Mittelpunkt. Die Technologien unterstützen dabei den Menschen und ermöglichen ein eigenständiges, sicheres und komfortables Leben in den eigenen Wohnräumen bis ins hohe Alter.

Andrushevich (Andrushevich, Kistler, Bieri, & Klapproth, 2009) kategorisiert folgende Anwendungsfelder im AAL-Bereich:

- **Informationsassistentz**

Ein intelligentes Haus bzw. Wohnung soll Informationen bereitstellen, die von einer Person aktuell benötigt werden. Solch eine Information könnte z.B. eine Medikationsanweisung oder der Schlüsselfinder sein. Diese Informationsbereitstellung kann auch kontextabhängig sein und z.B. in Form von einer Universalfernbedienung erfolgen, die nur die Geräte des aktuell anwesenden Raumes steuert.

- **Intelligentes Umgebungsverhalten**

Das AAL-System soll durch das Verhalten des Benutzers dazulernen und zum gegebenen Zeitpunkt die erforderliche Aktion anbieten. So können Handlungsabläufe au-

tomatisiert werden und z.B. beim morgigen Aufstehen das Licht eingeschaltet, die Rollläden hochgezogen und das Badezimmer beleuchtet werden.

- **Vorhersagen von Notfallsituationen**

Basierend auf physischen Veränderungen im Verhaltensmuster des Nutzers sollen Vorhersagen für eventuell eintreffende Notfallsituationen gemacht und Präventivmaßnahmen eingeleitet werden. Ein Szenario dafür ist ein vergessener, eingeschalteter Elektroherd. Der Benutzer wird auf diese kritische Situation durch die Ausgabe einer Warnmeldung aufmerksam gemacht und bei einer fehlenden Reaktion wird der Elektroherd selbstständig ausgeschaltet.

- **Erkennung von Notfallsituationen**

Das System soll Notsituationen erkennen und dementsprechend handeln. So soll z.B. erkannt werden, dass der Bewohner am Morgen nicht wie gewohnt aufgestanden ist. Als Konsequenz könnte z.B. eine automatische Alarmierung von nahestehenden Personen und Verwandten erfolgen. Ein zweites Szenario, in dem das System sofort agieren muss, ist die Sturzdetektion. Auch hier muss bei fehlender Reaktion seitens des Benutzers eine Alarmierung erfolgen.

- **Sicherheit & Privatsphäre**

Bedingt durch mögliche physische und/oder mentale Einschränkungen benötigen ältere Mitmenschen unter Umständen besondere Sicherheitsmaßnahmen. Ein Szenario dafür ist die Ausgabe einer Warnmeldung bei einer unverschlossenen Eingangstür. Bei fehlender Reaktion kann die Tür schließlich automatisch versperrt werden.

Wie die Anwendungsfelder verdeutlichen, muss ein AAL-System eng mit dem Bewohner zusammenarbeiten. Die Benutzerinteraktion zwischen dem Bewohner und dem AAL-System erfolgt bisher jedoch meist über zusätzliche Personal Computer (PC) und/oder Touchscreen-PCs. Solche zusätzlichen und oft auch nicht vertrauten Geräte stellen für die Zielgruppe ein nicht unerhebliches Hindernis dar. Sie werden einerseits häufig als ein störendes Element des eigenen Wohnungsinventars betrachtet und andererseits ist die Interaktionsakzeptanz oft sehr niedrig, da die Personen das Gefühl haben etwas falsch machen zu können. In dieser Arbeit soll eine alternative und benutzerfreundliche TV- bzw. Set-Top-Box-basierte Interaktionsmöglichkeit für bestehende AAL-Systeme konzeptioniert und ausgearbeitet werden. Die Interaktion mit dem Benutzer soll nicht wie bisher nur mit Hilfe von zusätzlichen PCs und/oder Touchscreen PCs, sondern auch über handelsübliche, HDTV-fähige TV-Geräte erfolgen können.

Diese Diplomarbeit gliedert sich in vier Teile. Der erste Teil untersucht bestehende AAL-Systeme und erarbeitet so ein Set von Anforderungen, welche die zu implementierende Software erfüllen muss. Im Kapitel Hardwareanforderungen werden zuerst die gängigen

Standards für interaktives TV und TV-Signalübertragung behandelt und anschließend einige Geräte vorgestellt, die eine konkrete Softwareimplementierung erlauben würden. Hier wird auch eine Hardware ausgewählt, auf dem das System schließlich installiert und betrieben wird. Der Bereich Softwaredesign zielt darauf ab, eine Software zu gestalten, welche den Softwareanforderungen des ersten Kapitels gerecht wird und auf der Hardwareplattform eingesetzt werden kann, die im zweiten Kapitel ausgewählt wurde. Das letzte Kapitel behandelt schließlich die konkrete Implementierung bzw. die Installation der Software und stellt somit das Resultat bzw. Ergebnis dieser Diplomarbeit dar.

2. Bestehende AAL - Systeme und die Softwareanforderungen

Wie schon vorab in der Einleitung beschrieben, beschäftigt sich diese Diplomarbeit mit der Konzeption und Implementierung einer TV-basierten Interaktionsmöglichkeit für bereits bestehende AAL-Systeme. Dieses Vorhaben stellt eine nicht geringe Herausforderung an die Softwaremodellierung dar, weil die Implementierung bei möglichst vielen und verschiedenartigen AAL-Systemen einsetzbar sein soll. Somit ergibt sich die folgende Fragestellung: „Wie sieht ein bestehendes AAL-System aus bzw. wie ist es aufgebaut?“ Diese Frage soll anhand von vier AAL-Systemen näher untersucht werden und so eine Basis für die folgenden Softwareanforderungen liefern.

2.1. Bestehende AAL-Systeme

Eine vollständige Auflistung aller möglichen und gängigen AAL-Systeme würde einerseits den Rahmen dieser Diplomarbeit sprengen und sich andererseits wegen dem dynamischen Umfeld des Themenbereiches als ein kaum mögliches Unterfangen erweisen. Aus diesem Grund beschränkt sich die Untersuchung auf zwei TV-basierte und zwei nicht TV-basierte AAL-Systeme. Das Ziel der Untersuchung ist nicht die Gegenüberstellung von einzelnen Stärken und Schwächen hinsichtlich der AAL-Anwendungsfelder, sondern die Findung von Gemeinsamkeiten und das Aufzeigen von Unterschieden bezüglich der Gestaltung einer Software, die in möglichst verschiedenen Systemen eingesetzt werden kann.

2.1.1. Nicht TV-basierte Systeme

Die folgenden zwei AAL-Systeme bieten eine Interaktionsmöglichkeit an, die nicht auf einem TV-Gerät basiert. Genau diese Art von Systemen kommen für den Betrieb der Software, die im Rahmen dieser Diplomarbeit erarbeitet werden soll, in Frage. Das TV-Gerät soll bei diesen Systemen ein weiterer Interaktionspunkt sein und neben anderen Geräten wie z.B. Touchscreens, Tablet-PCs und Smartphones koexistieren.

2.1.1.1. HOMER Plattform

Die HOMER Plattform (AIT - Austrian Institute of Technology GmbH, 2011) basiert auf einem OSGi¹ Framework, das eine Event- und Situationserfassung innerhalb einer Smart-Home²-Umgebung ermöglicht und dessen aktive Steuerung erlaubt. Die ankommenden Sensorsignale werden in der Plattform ausgewertet und auf Abweichungen von bekannten Verhaltensmustern des Bewohners untersucht. In vordefinierten Ausnahmesituationen kann zuerst der Bewohner selbst und bei fehlender Reaktion auch eine Kontaktperson

¹ OSGi: Open Service Gateway initiative. Vergl. <http://www.osgi.org/>

² Smart-Home: Andere Bezeichnung für intelligentes, vernetztes Heim. Vergl. <http://www.itwissen.info/definition/lexikon/E-Home-eHome-electronic-home.html>

oder eine Notrufzentrale verständigt werden. Die Plattform ermöglicht die Anbindung von drahtlosen und drahtgebunden Sensoren/Aktuatoren, sowie Geräten, die den ISO/IEEE 11073 Standard³ verwenden. Die Ausgabe der Benutzeroberfläche erfolgt auf einem Touchscreen-PC und/oder einem Tablet-PC (siehe **Abbildung 1** links). Die Konfiguration des Systems kann direkt an der graphischen Benutzeroberfläche der Plattform durchgeführt werden (siehe **Abbildung 1** rechts). Aktuell wird das HOMER-System im Passivhaus 159 der Firma ELK Fertighaus AG in der Blauen Lagune in der Marktgemeinde Vösendorf eingesetzt (NovaHome-Konsortium, 2011), sowie in einer Demonstrationswohnung namens „2-Raumwohnung“ am Standort Wiener Neustadt, des Unternehmens Austrian Institute of Technology GmbH.

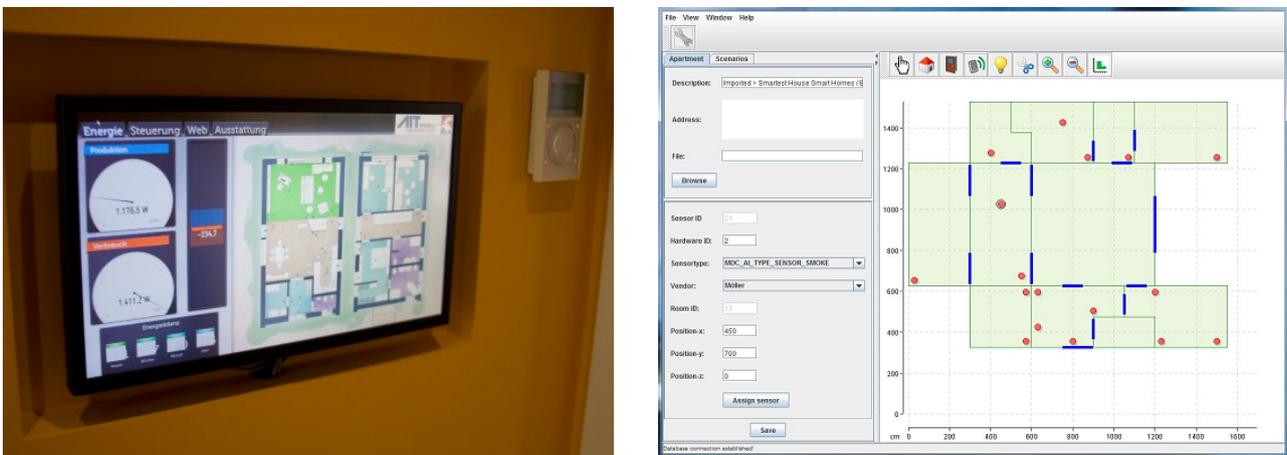


Abbildung 1: Links: HOMER Benutzeroberfläche die aktuell den Energieverbrauch des Hauses darstellt. Quelle: <http://www.novahome.at/musterhaus/galerie>. Rechts: Konfigurationsbereich der HOMER Plattform Quelle: <http://homer.aaloa.org/redmine/projects/homer-core/wiki>

2.1.1.2. eHome Plattform

Das eHome - Projekt (Technische Universität Wien, Institut 'integriert studieren', 2011) beschäftigt sich ebenfalls mit dem Monitoring von Aktivitäten und der Erkennung von gefährlichen Ausnahmesituationen, die nicht dem normalen Verhaltensmuster des Bewohners entsprechen. Das System baut bei kritischen Situationen, in denen keine Rückmeldung seitens des Nutzers erfasst werden kann, automatisch eine telefonische Verbindung zu einer Kontaktperson oder zu einer Notrufzentrale auf. Auch die Vorhersage von Notfallsituationen wie z.B. das Erkennen eines vergessenen, eingeschalteten Elektroherdes wird von der Plattform ermöglicht. Für die Datenerfassung werden sowohl kommerzielle Sensoren, als auch eigenentwickelte ZigBee-basierte Sensorboxen verwendet. Die Kommunikation mit der Zentraleinheit wird drahtlos durchgeführt. Die eigenentwickelte Sensorbox ermöglicht die Erfassung von Vibrationen und kann für die Sturzerkennung verwendet wer-

³ ISO/IEEE 11073-Standard: Siehe http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=36347

den. Die Benutzerinteraktion erfolgt mit Hilfe eines Touchscreen-PCs, welcher auch Video-Telefonie erlaubt (siehe **Abbildung 2** links). Das System kann mit Hilfe eines Webbrowsers konfiguriert werden (siehe **Abbildung 2** rechts). Für das Projekt wurden 5 eHome-Systeme erstellt und insgesamt in 11 privaten Wohnungen über einem Zeitraum von insgesamt 553 Tagen erprobt. Aktuell kann die eHome Installation in der AAL-Demo-Wohnung im Seniorenzentrum der Stadtgemeinde Schwechat besichtigt werden (CEIT Central European Institute of Technology gemeinnützige GmbH, 2009) (Seniorenzentrum Schwechat, 2011).

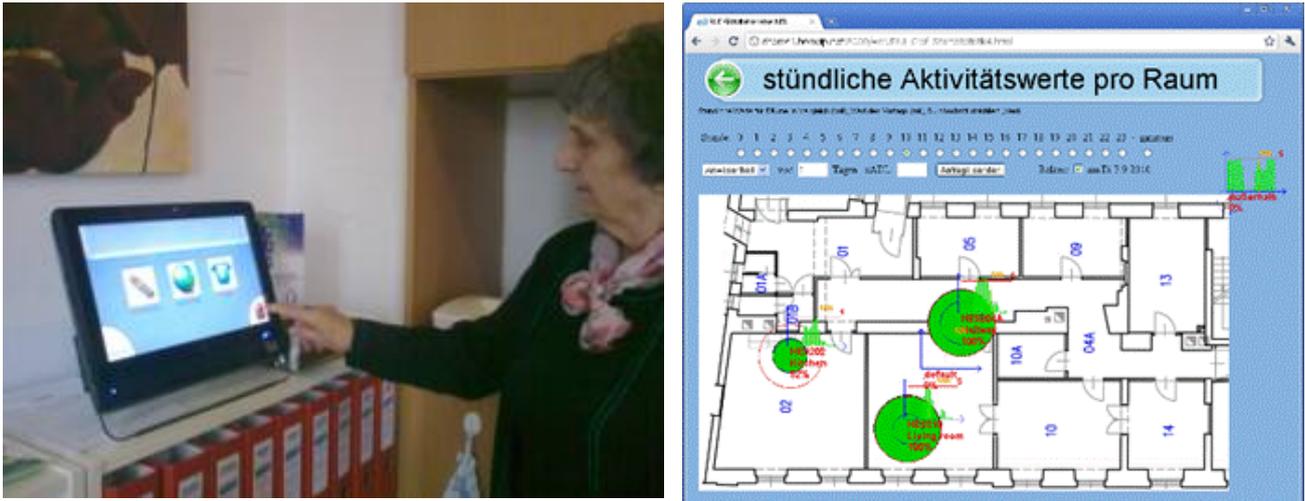


Abbildung 2: Links: Benutzerinteraktion mit der eHome-Plattform. Rechts: Konfigurationsbereich der eHome-Plattform. Quelle: <http://www.is.tuwien.ac.at/ehome/e-HomeErgebnisse.html>

2.1.2. TV-basierte AAL-Systeme

Anders als die zuvor vorgestellten Plattformen HOMER und eHome, nimmt das TV-Gerät bei den folgenden zwei Systemen eine zentrale Komponente des Systems ein. Dieser all-in-one-Ansatz hat den Vorteil, dass keine Zusatzgeräte wie z.B. ein separater PC für die Erfassung bzw. Auswertung von Sensorsignalen und die Steuerung von Aktuatoren benötigt sind. Der Ansatz hat jedoch zugleich den großen Nachteil, dass die AAL-Funktionalität sehr stark an ein TV-Gerät gebunden ist und das System auch nur an dieser Stelle vom Benutzer angesprochen bzw. bedient werden kann.

2.1.2.1. HOMEBUTLER

Das von dem Unternehmen BEKO Engineering & Informatik AG entwickelte und kommerziell vertriebene Produkt HOMEBUTLER (BEKO Engineering & Informatik AG, 2011) ist ein Set-Top-Box-basiertes System, das sowohl in privaten Haushalten als auch in betreuten Wohnanlagen eingesetzt werden kann. Die Set-Top-Box stellt dabei ein all-in-one-System dar, das neben der Wiedergabe von Fernsehinhalten weitere Dienste wie z.B. Notrufgerät, Telefon, Internet und die Hausautomatisierung ermöglicht. In der **Abbildung 3**

rechts werden die einzelnen Dienste, die von dem HOMEBUTLER angeboten werden, aufgelistet. Das System bietet durch die angebundenen Sensoren/Aktuatoren auch Aufsichts- und Steuerfunktionalitäten wie z.B. Wasserstopp, Brandmeldung oder Herdplattenkontrolle an. Aktuell kann der HOMEBUTLER in einer Musterwohnung namens „FORTUNA aktiv“ am Klesplatz 6, 1120-Wien besichtigt werden.



Abbildung 3: Links: HOMEBUTLER Benutzeroberfläche. Rechts: Übersicht der Dienste die von der HOMEBUTLER Plattform angeboten werden. Quelle: <http://www.beko.at/index.php?id=1048>

2.1.2.2. HERA-project

Das HERA-Projekt (Spanoudakis et al., 2010) verwendet ebenfalls eine Set-Top-Box für die Benutzerinteraktion. Anders als bei den drei vorgestellten Systemen HOMER, eHome und HOMEBUTLER findet bei dieser Plattform kein direktes Monitoring des Bewohners über Sensoren/Aktuatoren einer Smart-Home-Umgebung statt. Das Primärziel liegt in der Aufzeichnung von medizinischen Daten wie z.B. die eines Blutdruckmessgerätes oder einer Waage und dessen Übermittlung an einen Arzt. Das System zielt darauf ab, die Kommunikation zwischen dem Arzt und dem Patienten wesentlich zu erleichtern und zu intensivieren. So soll das Versenden von textbasierten Nachrichten nicht nur in eine Richtung vom Benutzer an den Arzt, sondern auch in die umgekehrte Richtung möglich sein. Neben einer Erinnerungsfunktion werden auch Spiele am System angeboten, die ein mentales Training ermöglichen. Wie in der **Abbildung 4** zu sehen ist, verwendet die HERA-Plattform das Medium Internet, um die erfassten Daten an einen extern stehenden Applikationsserver zu übertragen. Das Internet ist zugleich auch das einzige Übertragungsmedium für die TV-Signale und beschränkt somit den Einsatz auf Regionen bzw. Ballungsräumen, in denen IPTV-Signale (siehe Abschnitt 3.1.2) empfangen werden können.

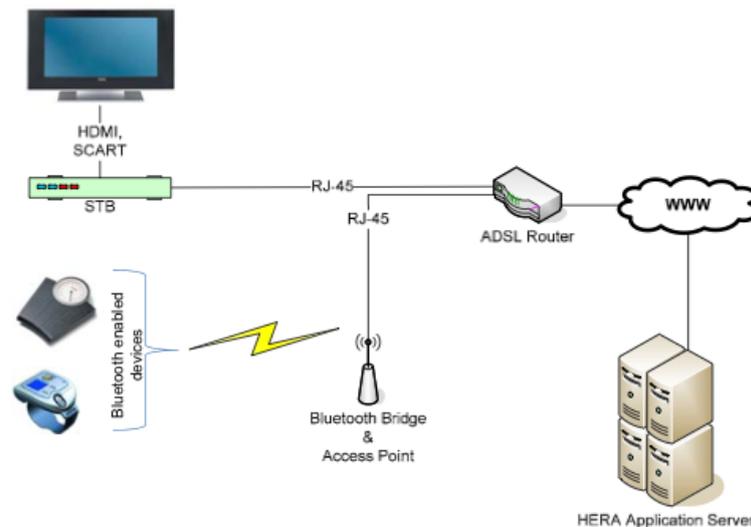


Abbildung 4: Aufbau der HERA - Plattform. Quelle: http://www.math-info.univ-paris5.fr/~nikos/HERA_silo_wp5_d522_final.pdf

2.1.3. Vergleich der Systeme

Die zwei nicht TV-basierten Systeme verwenden für die Datenerfassung und Datenauswertung ein eigenes Backend, das von der eigentlichen Benutzerinteraktion getrennt ist. Die TV-basierten Systeme weisen diese Unterteilung in Backend und Frontend nicht auf. Dieser all-in-one-Ansatz mag auf den ersten Blick praktisch erscheinen, da nur ein einziges Gerät benötigt wird, ist aber bezüglich der Bedienbarkeit unvorteilhaft. Der Nutzer muss bei solch einem System immer den Raum, in dem sich das Gerät befindet, aufsuchen, um eine Aktion am AAL-System durchführen zu können. Dies entspricht nicht einem benutzerfreundlichen Verhalten und sollte eher wie folgt aussehen: wenn sich der Benutzer vor dem TV-Gerät befindet und dort eine TV-Sendung konsumiert, dann soll eine Ereignisanzeige auch am TV-Gerät ausgegeben werden. Wenn der Nutzer nicht in dem TV-Zimmer anwesend ist, dann soll am TV-Gerät auch keine Ereignisanzeige erfolgen, sondern andere Maßnahmen eingeleitet werden. Um dieses Verhalten zu ermöglichen sollte die Software, die im Rahmen dieser Diplomarbeit entwickelt wird, möglichst unabhängig von dem eingesetzten Backend funktionieren.

Eine Gemeinsamkeit, die alle vier vorgestellten Systeme teilen, ist die Möglichkeit der Anzeige und der Bestätigung von Warn- und Ereignismeldungen. Die Software sollte auch diese Funktionalität zur Verfügung stellen. Drei der vier vorgestellten Systeme unterstützen die aktive Steuerung von Geräten, die sich in der Smart-Home-Umgebung befinden. Auch dies stellt eine Anforderung dar, welche die zu entwickelnde Software ermöglichen muss.

2.2. Softwareanforderungen

Aus dem direkten Vergleich der vier vorgestellten Systeme wurden folgende funktionale und nichtfunktionale Anforderungen an die Software erarbeitet.

2.2.1. Funktionale Anforderungen

Funktionale Anforderungen beschreiben das Verhalten, das ein zu erstellendes Softwaresystem unbedingt erfüllen muss (Versteegen, 2002, S. 19). Das zu erwartende Verhalten ist sehr stark von der Anwendergruppe abhängig. So können die Erwartungen aus der Sicht des Benutzers, des Entwicklers oder des Kunden unterschiedlich ausfallen. Ausgenommen der ersten funktionalen Anforderung beziehen sich die restlichen hier aufgelisteten Anforderungen auf die Sicht des Benutzers.

2.2.1.1. Integration in bestehende AAL-Backend-Systeme

Aus der Sicht des Entwicklers bzw. des Handwerkers, der die Inbetriebnahme vornimmt, muss sich die zu entwickelnde Software möglichst leicht in ein bestehendes AAL-System integrieren lassen. Das erfordert ein Frontend, das vom eingesetzten Backend weitgehend unabhängig betrieben werden kann. Dies ermöglicht nicht nur den Einsatz von verschiedenen AAL-Systemen, sondern auch den Einsatz von mehreren Frontends, die abhängig von der Position des Nutzers gezielt ein- bzw. ausgeschaltet werden können.

2.2.1.2. Wiedergabe von Warn- und Ereignismeldungen, Home-Monitoring

Die Software soll in der Lage sein, das aktuelle TV-Programm zu unterbrechen bzw. den Inhalt zu überblenden, um so auf eine kritische Situation hinzuweisen. Diese Funktionalität wird in dieser Ausarbeitung als Home-Monitoring-Funktion bezeichnet. Obwohl die Monitoring Aktivitäten nicht primär von dem TV-Gerät, sondern von den im Haushalt verbauten Sensoren und dem Backend durchgeführt werden, erfolgt die Ausgabe der daraus resultierenden Meldungen am TV-Gerät. So stellt das TV-Gerät die Spitze der Verarbeitungskette einer Home-Monitoring-Funktion dar. Des Weiteren trägt die Anzeige und die darauf folgende Reaktion des Nutzers ebenfalls zum Monitoring bei, da z.B. bei einem Fehlen von Rückmeldungen kritische Situationen erkannt und weitere Maßnahmen eingeleitet werden können.

2.2.1.3. Audiovisuelle Ausgabe

Kritische Situationen sollen nicht nur visuell, sondern auch akustisch ausgegeben werden. Das System soll neben der Ausgabe eines kurzen akustischen Warnsignals auch die Mög-

lichkeit anbieten, dass Alarmmeldungen automatisch vorgelesen werden. Dabei ist darauf zu achten, dass die Ausgabe auch auf Deutsch möglich ist.

2.2.1.4. Abfrage von Sensoren und Steuerung von Geräten, Home-Control

Eine weitere funktionale Anforderung ist die aktive Steuerung von Geräten und Umgebungsgegenständen. Diese Aktivität wird in Folge als Home-Control-Funktion bezeichnet. Der Nutzer soll in der Lage sein, einen im Haushalt befindlichen Sensor, ein Gerät oder einen steuerbaren Gegenstand auszuwählen und dessen Status zu betrachten bzw. dessen Wert steuern zu können. So soll das Ein-/Ausschalten des Lichtes in einem Nebenzimmer, die Steuerung der Jalousie oder die Nachjustierung der Raumtemperatur über das TV-Gerät möglich sein.

2.2.1.5. Bedienbarkeit mit Hilfe einer Fernbedienung

Das System muss über die Fernbedienung der Set-Top-Box oder des TV-Gerätes steuerbar sein. Die Verwendung einer Tastatur oder einer Computermaus soll nicht erforderlich sein. Dies gilt sowohl für die Bestätigung von Alarmmeldungen in der Home-Monitoring-Funktion als auch für die Steuerung von Geräten in der Home-Control-Funktion.

2.2.2. Nichtfunktionale Anforderungen

Nichtfunktionale Anforderungen sind - wie schon der Name vermuten lässt - Anforderungen, die nicht mit der Funktion des Systems zusammenhängen, sondern die Qualitätskriterien des Systems betreffen (Versteegen, 2002, S. 19). Versteegen listet folgende Qualitätsattribute als nichtfunktionale Anforderungen eines Systems auf:

- Anwenderfreundlichkeit
- Wartbarkeit
- Performance
- Zuverlässigkeit

Diese Einteilung kann jedoch nicht bei jedem System angewendet werden. So kann z.B. die Zuverlässigkeit bei sicherheitsrelevanten und lebenserhaltenden Systemen nicht als eine nichtfunktionale Anforderung definiert werden. Ebenso kann die Anwenderfreundlichkeit bei der Primärzielgruppe des AAL-Systems, den 65+ Jährigen, nicht nur als nichtfunktionale Anforderung definiert werden. Sie muss kontinuierlich mitberücksichtigt werden und stellt dadurch einen essentiellen Bestandteil von jeder funktionalen, aber auch von jeder nichtfunktionalen Anforderung dar. Für die zu entwickelnde Software wurden deswegen

nichtfunktionale Anforderungen vor allem aus den Bereichen Wartbarkeit und Performance definiert.

2.2.2.1. Einfache Adaptierung

Die Software soll möglichst einfach und rasch an eine neue Wohnung, ein neues Haus oder eine neue Situation im gleichen Haus angepasst werden können. Der Umgang mit mehr als einer Wohnung bzw. einem Haus muss unterstützt werden. Auf diese Weise können zusätzliche Räumlichkeiten wie z.B. eine Garage, ein Kellerabteil oder auch eine Zweitwohnung angezeigt und aktiv gesteuert werden.

2.2.2.2. One-Box-System

Neben der Unabhängigkeit von dem eingesetzten Backend soll die zu entwickelnde Software auch unabhängig von anderen Systemen sein. Ein außenstehender Server, wie es bei dem HERA-Projekt angewendet wurde, soll vermieden werden. Die oben beschriebenen funktionalen Anforderungen sollen, wenn möglich, von einer einzigen Set-Top-Box oder einem einzigem TV-Gerät erfüllt werden.

2.2.2.3. Open-Source-System

Die zu entwickelnde Software soll unter die GNU General Public License (GPL) gestellt und so für alle Interessenten zugänglich gemacht werden. Somit dürfen keine weiteren Applikationen oder Softwarepakete verwendet werden, die eine Veröffentlichung unter der GPL nicht zulassen würden.

2.2.2.4. Parametrisierung

Parameter wie z.B. die verwendete Schriftgröße, Schriftfarbe, Hintergrundfarbe, Lautstärke usw. sollen an die Bedürfnisse des Benutzers angepasst werden können. Auf diese Art soll die Lesbarkeit von visuellen und die Verständlichkeit von auditiven Meldungen gesteigert werden. Eine Modifizierung muss auch während des Betriebs des Systems und ohne eine erneuerte Übersetzung (Kompilierung) der Software möglich sein. Die Parametrisierung ist ein wesentlicher Teil, der die Usability und die Anwenderfreundlichkeit der Software steigert. Weitere Aspekte werden im Abschnitt 4.9 behandelt.

3. Anforderungen an die Hardware

Das zu entwickelnde, TV-basierte AAL-System soll sowohl einen geringen Umstellungsaufwand als auch geringe Anschaffungs- und Installationskosten für den Benutzer verursachen. Um diese Ziele zu erreichen, ist die richtige Wahl der Hardware, auf der die Software installiert und betrieben werden soll, von wesentlicher Bedeutung. Der folgende Abschnitt behandelt zunächst aktuelle Techniken und Standards am TV-Sektor und versucht so die Mindestanforderungen an die Hardware zu erarbeiten. Danach werden verschiedene am Markt erhältliche Geräte auf ihre Einsatzmöglichkeiten für ein AAL-Frontend untersucht und schließlich eine Hardwareplattform für die Implementierung ausgewählt.

3.1. Empfangsmöglichkeiten für TV - Signale

Eine wichtige Anforderung an die Hardware stellen die Empfangstechnologien für TV - Signale dar. Durch die Abdeckung möglichst verschiedener Empfangsarten kann das System bereits vorhandene Empfangsanlagen ausnutzen. Dies verringert nicht nur die Installationskosten und die Inbetriebnahme-Dauer, sondern ermöglicht zudem noch den Einsatz an Orten mit eingeschränkten Empfangsmöglichkeiten für TV-Signale. Orte mit eingeschränkten Möglichkeiten sind z.B. Wohnsiedlungen, in denen kein Kabelsignal zur Verfügung steht. In solch einem Fall könnte ein auf das Kabelsignal-basierendes System nicht verwendet werden. Aus diesem Grund ist es essentiell auf Hardware zu setzen, die möglichst breitgefächerte Empfangsmöglichkeiten aufweist. Hier stellten sich jedoch die Fragen, welche Empfangsarten es überhaupt gibt und welche in Europa am häufigsten verwendet werden? Der nachfolgende Abschnitt diskutiert die für den europäischen Raum ausschlaggebenden Empfangsarten.

3.1.1. Terrestrisch-, kabel- und satellitenbasierte TV - Signalübertragung

TV-Signale werden heutzutage, abgesehen von den kaum noch verwendeten analogen Übertragungstechniken, primär in digitaler Form übertragen. Die Umstellung der Übertragungstechnik wurde in Österreich vor allem im Jahr 2006 spürbar, als das digitale Antennenfernsehen startete und das analoge dafür sukzessiv eingestellt wurde (Digitales Fernsehen Förder GmbH, 2011a). Diese Entwicklung kann aktuell auch an der satellitenbasierten Empfangstechnologie beobachtet werden. Einige deutschsprachige TV-Kanäle können noch analog via ASTRA⁴ empfangen werden. Der Termin für die Abschaltung ist jedoch schon festgelegt und wird am 30. April 2012 erfolgen (SES Astra, 2011).

⁴ ASTRA-Satellitensystem: Vergl. <http://www.astra.de/2117/de>

Die in Europa verwendeten Standards für digitale Videoübertragung, die sogenannten Digital Video Broadcasting (DVB)-Standards, werden von einem industriellen Konsortium entwickelt und von der Organisation "European Telecommunications Standards Institute" (ETSI) veröffentlicht (DVB Project, 2010). Zu der Reihe der DVB-Standards zählen die weiterverbreiteten Standards DVB-T (EN 300 744), DVB-C (EN 300 429) und DVB-S (EN 300 421). Die Endbuchstaben kennzeichnen das verwendete Übertragungsmedium Terrestrik, Breitbandkabel und Satellit.

Die von der ETSI herausgegebenen DVB-Standards sind weltweit gesehen die meistverwendeten, jedoch nicht die einzigen Standards für digitale Rundfunkübertragung. Dies verdeutlicht der vom DVB-Konsortium in Auftrag gegebene und von der Firma Screen Digest erstellte Report "The Global Transmission Market" (Screen Digest, 2010). Im Gegensatz zum in Europa, Afrika, Australien und in großen Teilen Asiens verwendeten DVB-T-Standard, wird in Nordamerika der von dem Advanced Television System Committee (ATSC) festgelegte ATSC-Standard verwendet. In Südamerika wird der etwas abgewandelte und ursprünglich aus Japan stammende Integrated Service Digital Broadcasting (ISDB-T)-Standard eingesetzt. China, Hongkong und Macau verwenden den Digital Terrestrial Multimedia Broadcast (DTMB)-Standard.

Da diese Diplomarbeit in Österreich erstellt wird, sind die in hier verwendeten DVB-Standards ein wesentliches Auswahlkriterium. Die verwendete Hardware muss in der Lage sein, die Standards DVB-T, DVB-S und DVB-C entweder direkt, oder mit Zuhilfenahme von z.B. Tuner-Steckkarten zu beherrschen.

3.1.2. Fernsehen über das Internet

Die rasante Entwicklung und Verfügbarkeit von Breitband-Internetzugangängen forderte auch die Einführung einer Empfangstechnologie, die auf dem Internet Protokoll (IP) basiert. Unter IP-basierten Übertragungstechnologien versteht man allgemein die digitale Übertragung von Audio- und Videosignalen über eine Internetverbindung. Dabei muss eine klare Unterscheidung zwischen den oft als Synonym verwendeten Begriffen Internet Protocol Television (IPTV) und WebTV (Internet-TV) gemacht werden.

Beim IPTV werden die TV-Inhalte mit garantierter Qualität und garantierten Bandbreiten übertragen. Dazu ist ein lokales Telekommunikationsunternehmen notwendig, welches neben den TV-Inhalten auch den Breitbandanschluss zur Verfügung stellt und so den IPTV-Nutzer in ein geschlossenes und kontrolliertes IP-Netzwerk einbindet. Die notwendigen Bandbreiten werden unter anderem dadurch erzielt, dass neben den üblichen Requ-

est-Response-Verfahren auch Multicast-Übertragungsart angewendet wird (Riegamer, 2008, S. 10-13).

WebTV ist hingegen meistens global verfügbar, garantiert jedoch nicht die Qualität und die Bandbreiten wie sie beim IPTV gemacht werden. Zudem ist das WebTV-Angebot unübersichtlich und meistens an sehr individuelle Zielgruppen gerichtet. Nahezu jedes online Portal bietet einen eigenen WebTV-Dienst an. Eine Programmübersicht, wie sie beim klassischen Fernsehen üblich ist, kann beim WebTV wegen der fehlender Transparenz nicht realisiert werden (Maier, 2008, S. 20-21) (Zier, 2007).

Ähnlich wie bei der terrestrischen TV-Signalübertragung (siehe Abschnitt 3.1.1) wurden fast gleichzeitig auch mehrere Standards für die IPTV-Technologie entwickelt. Im Artikel „An Overview of IPTV Standards Development“ (Julien Maisonneuve, 2009) werden die fünf signifikantesten IPTV-Standards gegenübergestellt und miteinander verglichen. In Österreich bietet z.B. das Unternehmen A1 Telekom Austria AG seit 2006 ein Produkt namens A1 TV (vormals aonTV) an, das den DVB-IPTV-Standard (TR 102 033) verwendet (Fischer, 2009, S. 565). Ähnlich wie in Österreich sieht die Situation in Deutschland aus. Hier versorgen die Anbieter Alice, Acor und T-Home ihre Kunden mit IPTV (Beutel, 2009, S. 14).

Die Zahl der IPTV-Abonnenten wächst kontinuierlich und wird im Jahr 2014 weltweit voraussichtlich 109 Millionen betragen (RNCOS Market Research, 2011). Diesen Sachverhalt spiegeln auch die Anzahl der Zuseher des Österreichischen IPTV-Anbieters A1 wider. Laut Pressemitteilung des Unternehmens nutzten im September 2011 bereits 175.000 Kunden und damit rund 400.000 Zuseher das A1 TV-Angebot (A1 Telekom Austria AG, 2011). Die IPTV-Technologie stellt angesichts der relativ großen Zahl an Zusehern ein notwendiges Auswahlkriterium für die Hardware dar. Das eingesetzte TV-Gerät oder die eingesetzte Set-Top-Box müssen die Möglichkeit besitzen IPTV-Signale verarbeiten zu können.

3.1.3. Mobiles Fernsehen

Obwohl das mobile Fernsehen nicht in das Konzept für ein TV-basiertes und stationäres AAL-System passt, soll es hier dennoch kurz erwähnt werden und so die Liste der Fernsehempfangsmöglichkeiten für Europa vervollständigen. Für die Übertragung von TV-Signalen an Mobilgeräte kommen zwei Technologien in Frage. Als Erstes die Datenübertragung via mobiles Internet und als Zweites die Datenübertragung durch das digitale Antennenfunknetz. Für die zweite Variante wurden - ähnlich wie es bei den anderen Techno-

logien der Fall war - fast zeitgleich mehrere Standards entwickelt. Der für Europa signifikante Standard ist der von der ETSI veröffentlichte DVB-H-Standard (TR 102 033).

Entgegen der von Teiwes gemachten Prognose (Sebastian, 2009, S. 3), dass sich für die mobile Datenübertragung der DVB-H-Standard durchsetzen wird, werden TV-Signale heutzutage primär über das mobile Internet verbreitet. So wurde der österreichische DVB-H-Dienst nach der Einführung am 06.06.2008 (Österreichischer Rundfunk, 2008) wegen geringer Nachfrage am 31.12.2010 wieder eingestellt (Heise Zeitschriften Verlag, 2010). Wie schon vorab erwähnt, geht das mobile Fernsehen am Konzept eines TV-basierten, stationären AAL-Systems vorbei und stellt deshalb kein Auswahlkriterium für die zu verwendende Hardware dar.

3.2. Interaktivität

Das Aufgabengebiet von TV-Geräten ist seit der Einführung der digitalen Übertragungstechnik einem gewissen Wandel unterzogen. Das Fernsehgerät kann durch die neuen Funktionalitäten, die es in den letzten Jahren dazugewonnen hat, nicht mehr nur als ein klassisches Ausgabegerät für lineare TV-Programme (siehe Abschnitt 3.2.1), sondern als ein interaktives Werkzeug angesehen werden. Dem Benutzer wird die Möglichkeit geboten, in das laufende Programm einzugreifen, den Inhalt selbst zu bestimmen und weiterführende Informationen abzurufen (Beutel, 2009, S. 18). Mahr (Mahr, S. 91) beschreibt die Herausforderung des interaktiven Fernsehens wie folgt: „Die große Herausforderung besteht darin, auf intelligente Art die Angebote des Internets mit bewegten Bildern zu verknüpfen und umgekehrt das TV-Programm um interaktive Services zu erweitern.“ Unter dem Aspekt einer zukunftsorientierten AAL-Plattform müssen die interaktiven Fähigkeiten eines TV-Gerätes mitberücksichtigt werden. In den folgenden Abschnitten wird auf die Charakteristika, die technische Voraussetzung und die Standards für das interaktive Fernsehen bzw. interaktive Dienste eingegangen.

3.2.1. Charakteristika des interaktiven Fernsehens

Das klassische Fernsehen bzw. der Empfang und die Wiedergabe von Fernsehsignalen sind durch zwei wesentliche Charakteristiken gekennzeichnet. Einerseits durch die Wiedergabe von linearen TV-Inhalten und andererseits durch die lokale Interaktion. Interaktives Fernsehen hat indessen Eigenschaften, die weit darüber hinausgehen.

Unter linearem Fernsehen versteht man das zeitgleiche Ausstrahlen von Fernsehprogrammen (Scolik, 2007). Diese Multicast-ähnliche Übertragungsart ermöglicht zwar den simultanen Empfang von mehreren Fernsehprogrammen bei einer Vielzahl von Emp-

fangsgeräten, kann jedoch den zeitlichen Programmablauf nicht beeinflussen. Nichtlineares Fernsehen bietet hingegen diese Möglichkeit an. Der Benutzer bestimmt dabei sowohl den gewünschten Zeitpunkt als auch den gewünschten TV-Inhalt. Der Begriff TV-Inhalt wird dabei als ein Synonym für verschiedenste digitale Medien wie z.B. Text, Audio, Video oder Bilder verwendet. Bei Videodateien spricht man in diesem Zusammenhang auch von Video on Demand (VoD) bzw. Video auf Abruf. Die Koexistenz von linearen und nichtlinearen TV-Inhalten und deren Übertragungskanäle sind in Abbildung 5 illustriert.

Die zweite Charakteristik, die lokale Interaktion, beschreibt die Kommunikationsart zwischen dem Benutzer und dem Gerät. Nach Lugmayer (Lugmayer, Niiranen, & Kalli, 2004) findet bei der lokalen Interaktion nur ein Datenaustausch zwischen dem Benutzer und dem TV-Gerät selbst statt. In den Interaktionsprozess ist kein weiteres Gerät eingebunden. Solche Interaktionen sind z.B. die Anforderung der elektronischen Programmzeitschrift (EGP), das Umschalten des Fernsehsenders oder das Einstellen der Lautstärke. Diese Befehle kann das TV-Gerät bzw. Set-Top-Box selbst lokal abwickeln, ohne zusätzliche Daten mit einer Drittkomponente austauschen zu müssen. Im Gegensatz zum klassischen Fernsehen kommt das interaktive Fernsehen mit einer lokalen Interaktion nicht aus. Um Angebote wie z.B. VoD oder online Shopping realisieren zu können, muss ein zusätzlicher Kommunikationskanal, der sogenannte Rückkanal, zwischen dem Benutzer und der Sendeeinheit hergestellt werden. Der Rückkanal kann über eine Modemleitung, einen Breitband-Internetanschluss oder auch über systemspezifische Standards wie DVB-RCC, DVB-RCS und DVB-RCT (Beutel, 2009, S. 18) realisiert werden. Die systemspezifischen DVB-Rückkanäle sind jedoch kaum in Verwendung und es wird meist auf den Breitband-Internetanschluss als Übertragungsmedium zurückgegriffen. Abbildung 5 verdeutlicht ebenfalls die Rolle des Rückkanals bei der Anforderung von benutzerspezifischen Inhalten und die Bereitstellung der Daten über den nichtlinearen Übertragungskanal.

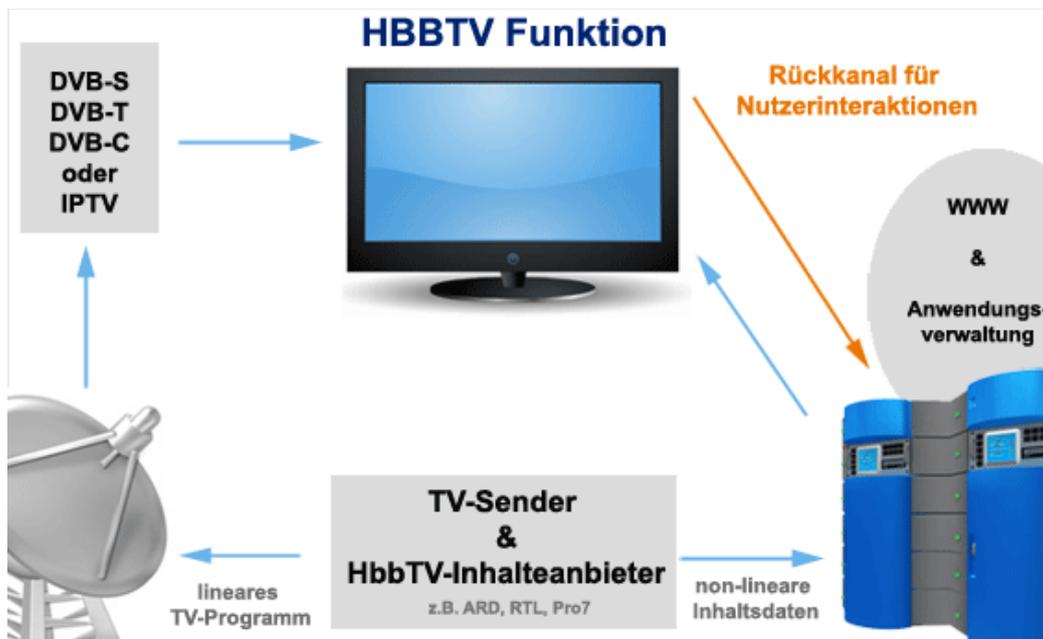


Abbildung 5 Interaktives Fernsehen anhand des HbbTV Standards. Verdeutlicht werden die verschiedenen Übertragungsmedien für lineare und nichtlineare TV-Inhalte (Quelle: HbbTV Infoportal)

3.2.2. AAL-Funktionalität als interaktiver Dienst

Die AAL-Funktionalitäten, das Eingreifen in das laufende TV-Programm in Alarmsituationen oder das Steuern von Aktuatoren mit Hilfe der TV-Fernbedienung können zusammengefasst als ein interaktiver Dienst angesehen werden. Die Kommunikation erfolgt wie bei den übrigen Diensten über den Rückkanal, wobei die Daten nicht über das Internet, sondern nur über das lokale Netzwerk übertragen werden müssen.

Die Fragestellung, die sich hier auftut, ist: „Wie können die AAL-Funktionalitäten als ein interaktiver Dienst definiert, übertragen und am TV-Gerät interpretiert werden?“ Ein AAL-Dienst ist im Gegensatz zu einem VoD-Dienst oder Plattformdienste wie Flickr⁵ oder YouTube⁶ nicht massenmarktauglich. Solch ein Dienst wird weder von den Rundfunkveranstaltern noch von den Geräteherstellern angeboten und ist nur für die Benutzergruppe, die ein AAL-System betreibt von Bedeutung. Aus diesem Grund müssen Anknüpfungspunkte gefunden werden, sodass neben den Geräteherstellern und den Rundfunkveranstaltern auch Drittanbieter ihre interaktiven Dienste an TV-Geräten anbieten können. Realisiert werden solche Anknüpfungspunkte einerseits durch Standards und andererseits durch gerätespezifische Schnittstellen.

⁵ Flickr: Eine Plattform, die den Austausch von Bildern ermöglicht. Siehe <http://www.flickr.com>

⁶ YouTube: Eine Plattform, die den Austausch von Bildern ermöglicht. Siehe <http://www.youtube.com>

3.2.3. Standards am Sektor interaktive Dienste

Um proprietäre Lösungen einzelner Rundfunkveranstalter zu vermeiden, wurden neben den Standards für die digitale Datenübertragung auch Standards für interaktive Dienste entwickelt. Standardisierte Daten und Schnittstellen bieten Vorteile für die Rundfunkanstalten aber auch für die Endgerätehersteller. Die Endgerätehersteller haben die Möglichkeit interoperable⁷ Endgeräte herzustellen und die Rundfunkanstalten können auf diese Endgeräte setzen und somit davon ausgehen, dass ihre Inhalte beim Endkunden von den Geräten interpretiert werden können. Leider wurden - ähnlich wie bei den digitalen Übertragungstechniken - auch hier nahezu parallel mehrere Standards entwickelt. Es geht sogar soweit, dass verschiedene europäische Länder verschiedene Standards für interaktive Dienste einsetzen. Die in Europa verwendeten Standards sind die MHEG-, MHP- und der HbbTV-Standard.

3.2.3.1. Multimedia and Hypermedia Experts Group (MHEG) - Standard

Der MHEG 1-Standard wurde von der International Standardization Organization (ISO) erstmals im Jahr 1995 unter der Namen ISO/IEC 13522-1 publiziert. Das Subkomitee, welches den MHEG-Standard entwickelt hat, war auch für die Entwicklungen von JPEG (Joint Photographic Experts Group) und MPEG (Moving Picture Experts Group) zuständig. Der Standard beschreibt sowohl das Format als auch die Kodierung und die Repräsentation von multimedialen und hypermedialen Objekten (Colaitis & Jourdan, 1995). Hester (Hester Bornman, 1993) definiert Hypertext, Multimedia und Hypermedia wie folgt: „Hypertext is the grouping of relevant information in the form of nodes. These nodes are then connected together through links. In the case of hypertext the nodes contain text or graphics. Multimedia is the combining of different media types for example sound, animation, text, graphics and video for the presentation of information by making use of computers. Hypermedia can be viewed as an extension of hypertext and multimedia.“

Obwohl der MHEG 1 oft allgemein als der „MHEG-Standard“ bezeichnet wird, ist er nur der erste Teil einer Serie von Standards. Für das interaktive Fernsehen, wie es in dieser Diplomarbeit benötigt wird, ist vor allem der MHEG 5-Standard von Bedeutung. MHEG 5 wurde 1997 veröffentlicht und definiert die Implementierung der MHEG-Funktionseinheit (Engine) auf Plattformen wie z.B. Set-Top-Boxen. Gegenwärtig wird MHEG 5 von mehreren Rundfunkveranstaltern in Großbritannien auf der DVB-T Plattform eingesetzt (Reimers, 2008, S. 405)(Walter, 2009, S. 551).

⁷ Interoperabilität: Fähigkeit unterschiedlicher Systeme, möglichst nahtlos zusammenzuarbeiten (Quelle: Duden online).

3.2.3.2. Multimedia Home Platform (MHP) - Standard

Ein weiterer von der ESTI erstmals in Juli 2000 veröffentlichter Standard für interaktive Dienste ist der MHP-Standard. MHP ist genau genommen ein kollektiver Name für ein Set von Middleware⁸-Spezifikationen, die von dem DVB-Projekt entwickelt wurden. Der auf der Programmiersprache Java und der Metasprache HTML⁹ basierte Standard wurde so konzeptioniert, dass alle DVB-Transmissionstechnologien unterstützt werden (DVB Project, 2008). MHP wurde erstmals in Deutschland auf der IFA¹⁰ 1999 präsentiert und in Finnland im Jahr 2002 auf der DVB-T Plattform eingesetzt. Seit Juni 2009 ist der MHP Bestandteil des plattformunabhängigen Globally Executable MHP (DVB-GEM)-Standards, der von der ESTI unter dem Namen TS 102 728 V1.2.1 publiziert wurde (DVB Project, 2011). MHP wurde 2006 in Österreich von den Rundfunksendern ORF und ATV eingeführt. Der Dienst konnte sich trotz den damals eingeführten Fördermaßnahmen für MHP-fähige Endgeräte (RTR - GmbH, 2006) beim Endkunden nicht durchsetzen und wurde am 30 Juni 2011 wieder eingestellt. Das österreichische DVB-T-Portal begründet die Einstellung des MHP-Dienstes wie folgt: „Der technische Standard MHP des ORF OK MultiText, den der ORF im Jahr 2006 eingeführt hat, ist aufgrund der technologischen Entwicklungen heutzutage überholt.“ (Digitales Fernsehen Förder GmbH, 2011). Die durch die Einstellung des Dienstes frei gewordenen Bandbreiten werden seither für die Verbesserung der Bildqualität von ORF2, genauer gesagt für die Regionalprogramme, genutzt.

3.2.3.3. Hybrid Broadcast Broadband TV (HbbTV) - Standard

Der relativ neue Standard für interaktive TV-Dienste ist der HbbTV-Standard. Die Version 1.1.1 wurde von der ESTI im Juni 2010 veröffentlicht. Der Standard fand auch eine rasche Umsetzung in Frankreich und Deutschland. So startete z.B. die ProSieben-Sat1 Media AG ihren HbbTV-Dienst am 01 September 2010 und folgte damit den öffentlich-rechtlichen Sendern wie ARD, ZDF und Arte, die den HbbTV-Dienst einige Monate zuvor in Betrieb nahmen (SevenOne Media GmbH, 2011)(HbbTV Infoportal, 2011). In Österreich befindet sich HbbTV noch im Probetrieb und soll in naher Zukunft den mittlerweile eingestellten MHP-Standard ablösen (Österreichischer Wirtschaftsverlag GmbH, 2011). Da der Standard relativ neuartig ist, existieren derzeit nur wenige HbbTV-fähige Empfangsgeräte am Markt.

⁸ Middleware: Software für den Datenaustausch zwischen Anwendungsprogrammen, die unter verschiedenen Betriebssystemen oder in heterogenen Netzen arbeiten (Quelle: Duden online)

⁹ HTML: Hypertext Markup Language. Vergl. <http://www.w3.org/TR/1999/REC-html401-19991224/>

¹⁰ IFA: Messe für Consumer Electronics und Home Appliances in Berlin, Deutschland.

3.2.4. Die Einsatzmöglichkeiten der vorgestellten Standards für die AAL - Plattform

Aus der aktuellen Sicht ist es unvorteilhaft, auf den MHEG-Standard zu setzen. Dies gilt vor allem für den deutschsprachigen Raum, da der Standard von den diesen Rundfunkveranstaltern nicht verwendet wird und dadurch auch wenige Endgeräte am Markt verfügbar sind.

Ähnlich sieht die Situation beim MHP-Standard aus. Eine MHP-basierte Softwarelösung ist ebenfalls nicht erstrebenswert, da der Dienst in Österreich eingestellt wurde und die Zahl an verfügbaren MHP-Endgeräten ebenfalls abnehmend ist.

Der HbbTV-Standard könnte sich als außerordentlich progressiv erweisen. Der relativ neuartige Standard hat jedoch den Nachteil, dass aktuell zu wenige HbbTV-fähige Endgeräte am Markt verfügbar sind. In diesem Zusammenhang stellt sich zusätzlich die Frage, ob der HbbTV-Standard ein aktueller kurzfristiger Hype ist, oder ob sich der HbbTV-Standard behaupten wird. Grinschgl (RTR - GmbH, 2011) stellt richtig fest, dass gegenwärtig eine Verschmelzung des Fernsehens mit dem Internet stattfindet. Für die Rundfunkveranstalter ist es daher bedeutsam, auf die aktuellen Entwicklungen zu reagieren und ihre Angebote dementsprechend zu erweitern. Die deutschsprachigen Rundfunkanstalten reagieren relativ rasch und setzen den HbbTV-Standard schon teilweise ein. Anthony (Anthony, 2009) verdeutlicht in seiner Ausarbeitung jedoch, dass HbbTV nicht als eine neue Technologie, sondern eher als ein Konzept angesehen werden sollte. Die Funktionalitäten, die nun mit dem neuen HbbTV-Standard realisierbar sind, werden z.B. in Italien durch den MHP-Standard und einem IP-erweiterten Broadcasting schon seit 2004 angeboten. Ob sich der HbbTV-Standard im Gegensatz zum MHP-Standard etablieren wird, kann allerdings erst in einigen Jahren beantwortet werden.

3.3. Der Personal Computer (PC) als TV – Gerät

Auf der Suche nach einer geeigneten Hardware für eine TV-basierte AAL-Plattform darf der PC nicht außer Acht gelassen werden. Eine PC-basierte Lösung würde die Implementierung des AAL-Dienstes enorm erleichtern und zudem auch eine Implementierung für nahezu alle Empfangstechnologien erlauben. Mit Hilfe einer TV-Karte bzw. eines USB-Zusatzgerätes lässt sich ein PC innerhalb kürzester Zeit in ein TV-Empfangsgerät umfunktionieren. Solche Empfangseinheiten sind sowohl für die DVB-T, DVB-C als auch für die DVB-S-Plattform erhältlich (Siehe Abschnitt 3.1). Die dazu benötigte Software ist im Lieferumfang der Hardware inkludiert und bietet neben der Wiedergabe von linearen TV-Sendungen auch meistens die Funktionalität eines Digital Video Recorders an. Die TV-Inhalte können dabei auf die Festplatte des Computers aufgenommen und zu einem späteren Zeitpunkt wiedergegeben werden.

Hinsichtlich Interaktivität bietet eine PC-basierte Lösung gewisse Vorteile. Die AAL-Funktionalität lässt sich in einer PC Umgebung z.B. als eine eigenständige Applikation implementieren. In Alarmsituationen oder auf Befehl des Benutzers würde eine solche Applikation den Primärfokus bekommen und somit die Interaktion mit dem Benutzer ermöglichen.

Obwohl der PC unterschiedliche Empfangstechnologien abdecken kann und auch ein AAL-Dienst relativ rasch realisierbar ist, machen ihn seine typischen PC-Eigenschaften nahezu unbrauchbar für den Ersatz als ein TV-Gerät. Probst (Probst, 2011, S. 8,9) stellt fest, dass ein PC mit einer TV-Tuner-Karte noch lange nicht als ein Fernsehgerät angesehen werden kann. Er verdeutlicht die Problematik anhand folgender Punkte:

- PC in einem TV-Gerät ist nicht massenmarkttauglich
 - TV-Geräte haben limitierte Ressourcen an RAM, Festplatte und Prozessor
 - Firmware Updates sind selten, Feature Updates noch seltener
 - Schlechte Administrierbarkeit durch den Nutzer
- Unterschiedliche Nutzungskonzepte
 - Andere Betrachtungsbedingungen
 - Andere Interaktionsbedingungen (Fernbedienung statt Tastatur)
 - TV Nutzer sind passiver

Auch Gerdes (Gerdes, 2003, S. 56) zeigt in ihrer Gegenüberstellung der Aufgabengebiete eines PCs und eines TV-Gerätes markante Unterschiede. Viele dieser Unterschiede können jedoch aus heutiger Sicht als überholt angesehen werden. Gerdes führt unter ande-

rem auf, dass der PC eher für Information und Kommunikation, das TV-Geräte hingegen für Unterhaltung genutzt wird. Abgesehen vom Computerspielsektor, tragen insbesondere einige Videoportale wie z.B. YouTube, Clipfish¹¹ oder MyVideo¹² wesentlich dazu bei, dass der PC für Unterhaltungszwecke genutzt wird. Auch die durchschnittliche private Nutzung des PCs hat in den letzten Jahren wesentlich zugenommen. Das zeigt auch eine aktuelle in Deutschland durchgeführte Studie, nach der 79% der deutschen Bundesbürger den PC täglich benutzen. Die Studie zeigt jedoch auch, dass die Nutzung sehr stark vom Alter des Nutzers abhängig ist. Während der PC in der Altersgruppe der 18-29 Jährigen zu 98% genutzt wird, kommt es in der Altersgruppe der 50 – 64 Jährigen nur zu einer Nutzung von 79%. Die Primärzielgruppe des AAL-Systems, die 65+ Jährigen, nutzen den PC nur noch zu 41% (Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V., 2011).

Die relativ geringe Nutzungsgrad in der Altersgruppe 65+ und die weiter oben beschriebenen unterschiedlichen Nutzungskonzepte machen den gewöhnlichen PC zu keiner echten Alternative zu einem TV-Gerät. Anders sieht die Situation bei Smart-TV Geräten aus, die sowohl die Einfachheit eines Fernsehgerätes als auch die Komplexität eines PCs zu vereinen versuchen.

¹¹ Clipfish: Videoportal. Siehe <http://www.clipfish.de>

¹² MyVideo: Videoportal. Siehe <http://www.myvideo.de>

3.4. Smart-TV / Hybrid-TV / Connected-TV

Die relativ neuartigen Entwicklungen am TV-Segment sind die sogenannten Smart-TV- / Hybrid-TV- bzw. Connected-TV-Geräte. All diese Begriffe kennzeichnen Geräte, die sowohl den Empfang und die Wiedergabe von linearen TV-Inhalten, als auch die Nutzung von interaktiven Diensten ermöglichen (siehe Abschnitt 3.2). Geräte neuer Bauart beinhalten des Weiteren auch diverse DVB-Empfangsmodule und ermöglichen dadurch den Empfang von linearem als auch nichtlinearem TV-Programm. Eine zusätzliche externe Set-Top-Box wird somit nicht mehr benötigt. Die Begriffe Hybrid-TV bzw. Connected-TV deuten schon an, dass hier nicht nur eine, sondern auch eine zweite Technologie - die Internettechnologie - zum Einsatz kommt. Laut Produktwerbungen verschiedenster Gerätehersteller steht der Begriff Smart-TV über den beiden Begriffen Hybrid-TV und Connected-TV. Die Vernetzung mit dem Internet für den Betrieb von interaktiven Diensten ist demnach nur eine, von verschiedenen Funktionalitäten die ein Smart-TV-Gerät ermöglichen kann. Weitere Funktionen sind z.B. die Aufnahme auf ein Speichermedium, die Steuerung durch ein Smartphone oder auch die drahtlose Kommunikation mit anderen im Haushalt befindlichen Geräten. Bei Philips heißen diese Funktionen „Program“, „Control“ und „SimplyShare“ (Philips Deutschland GmbH, 2011). Um die Lesbarkeit zu fördern wird in den folgenden Abschnitten auf die Namensvielfalt für diese Art von Geräten verzichtet und stattdessen nur noch der Begriff Smart-TV verwendet.

Anders als bei den PCs stellen die Smart-TV-Geräte keine unüberwindbaren Bedienungshürden für den Benutzer dar, weil sie die gewohnte Form und Steuermöglichkeiten eines Fernsehgerätes bzw. einer Set-Top-Box haben. Abgesehen von der Erweiterbarkeit und der Prozessorgeschwindigkeit ist die eingesetzte Hardware eines Smart-TV-Gerätes mit der eines PCs vergleichbar. Auch am Softwaresektor können gewisse Parallelen zu einem PC gezogen werden. Smart-TV-Geräte setzen sehr oft komplexere Betriebssysteme wie z.B. Ubuntu¹³ oder Android¹⁴ ein. Der große Unterschied zu einem PC besteht jedoch in der Transparenz des Systems. Dem Benutzer wird ein spezieller, limitierter Zugang zum eingesetzten Betriebssystem und dessen Möglichkeiten geboten. Trotz vieler Zusatzdienste bleibt die Primärfunktion, die Wiedergabe von TV-Sendungen, erhalten und wird beim Starten des Gerätes automatisch in den Vordergrund gestellt. Interaktive Dienste und auch eigenständige Applikationen, die sogenannten TV-Apps, können durch ein Menü oder durch einzelne Icons erreicht werden. Die Navigation erfolgt nicht durch eine Maus und eine Tastatur, sondern über die TV-Fernbedienung. Alle diese Eigenschaften versuchen

¹³ Ubuntu: Eine auf Debian-basiertes Linux-Betriebssystem. Siehe <http://www.ubuntu.com/>

¹⁴ Android: Von Google Inc. entwickeltes Betriebssystem für Smartphones und Tablet-PCs mit einem Linux-Kernel. Siehe <http://www.android.com/>

die Komplexität des Systems zu verbergen und somit die passive Haltung des Benutzers zu überwinden.

Das Fernsehgerät kann bei der Verarbeitung von digitalen Signalen und Daten als Endgerät zum Benutzer angesehen werden. Anders als bei den Set-Top-Boxen, werden die Daten hier nicht nur dekodiert und weitergeleitet, sondern in einer audiovisuellen, für den Benutzer verständlichen, Weise ausgegeben. Für den Einsatz als ein AAL-Frontend eignet sich diese Positionierung besonders gut, da der TV-Inhalt in Alarmsituationen direkt hier am Endgerät angehalten oder überblendet werden kann. Es müssen keine weiteren Geräte mitberücksichtigt werden. Es ist demnach unerheblich, ob der Benutzer TV-Inhalte über Satellit, eine DVD oder über das Internet empfängt. Das Fernsehgerät ist die letzte Instanz bei der Verarbeitung dieser Daten und eine hier angesiedelte Alarmierung würde die Aufmerksamkeit des Benutzers auf jeden Fall auf sich ziehen.

3.4.1. AAL-Frontend in der Smart-TV-Umgebung

Ob sich ein AAL-Frontend in ein Smart-TV integrieren lässt oder nicht, hängt sehr stark vom jeweiligen Smart-TV-Gerät und auch vom jeweiligen Gerätehersteller ab, da hier unterschiedliche Strategien angewendet werden. Während einige Hersteller wie Samsung oder LG die Vielfalt von TV-Apps an ihren Geräten fördern wollen und deswegen auch spezielle Software-Development-Kits (SDK's) zu Verfügung stellen, halten sich andere Hersteller wie z.B. LOEWE diesbezüglich etwas zurück. Ihre Geräte bieten zwar einige ausgewählte Dienste wie z.B. YouTube, Facebook und Flickr an, aber die eigenständige Entwicklung scheint nicht gewollt zu sein, da keine SDK's oder Entwickleranleitungen kundgemacht werden.

Interaktive Inhalte lassen sich prinzipiell auf folgende zwei Arten in ein Smart-TV-Gerät einbinden:

- Als Webdienst, der im TV-Gerät nur verlinkt und in einem eigenen TV-Browser dargestellt wird.
- Als eigenständige TV-Applikation (TV-App), die am Gerät installiert wird.

Je nach Gerätehersteller stehen beide Möglichkeiten, eine Möglichkeit oder, wie vorab erwähnt, auch gar keine Möglichkeit zur Verfügung, um eigene interaktive Inhalte einzubinden.

Obwohl die TV-Apps bei den meisten Geräteherstellern ebenfalls aus Komponenten wie HTML, JavaScript (JS), CSS¹⁵ und Flash-Dateien bestehen, existiert dennoch ein Unterschied zwischen den beiden Kategorien. Geräte, die TV-Apps unterstützen, ermöglichen meistens auch den Zugriff auf das darunterliegende System. Die TV-Apps können z.B. die Lautstärke des Gerätes kontrollieren, auf die Flash-Speicherkarte schreiben oder auch das Fernsehbild anhalten bzw. überdecken. Geräte, die nur Webdienste innerhalb des TV-Webbrowser anzeigen können, ermöglichen den Zugriff auf das darunterliegende System für gewöhnlich nicht.

Für die Home-Control-Funktion reicht diese browserbasierte Interaktions- und Darstellungsart vollkommen aus. Hier leitet der Benutzer die Interaktion ein und dies kann in dem TV-Browser erfolgen. Die Situation sieht bei der Home-Monitoring-Funktion anders aus. Hier muss eine Alarmierung unabhängig von einer Benutzereingabe und außerhalb des TV-Browsers während der Wiedergabe von TV-Sendungen erfolgen. Zusätzlich muss auf das darunterliegende System wie z.B. auf das Soundsystem zugegriffen werden. Um beide Funktionen zu ermöglichen, bietet sich somit nur die Variante einer TV-Applikation an.

3.4.2. Vergleich verschiedener Smart-TV-Gerätehersteller

Wie schon vorab erwähnt, sind die Möglichkeiten einer Erweiterung hinsichtlich eines AAL-Frontends sehr vom jeweiligen Gerätehersteller abhängig. Die nachfolgende Tabelle vergleicht einige namhafte Smart-TV-Gerätehersteller bezüglich ihrer Erweiterungsmöglichkeiten für die hier zu implementierende Software.

An dieser Stelle sei angemerkt, dass es zum Zeitpunkt der Marktrecherche für diese Diplomarbeit weitaus weniger Möglichkeiten gab, in ein TV-System einzugreifen, um so eine eigene AAL-Applikation darauf zu installieren. Viele der folgenden Geräte wurden im deutschsprachigen Raum erstmals auf der IFA 2010 bzw. 2011 vorgestellt. Aus heutiger Sicht würden sich für die Umsetzung der AAL - Plattform mehre Alternativen und auch eventuell andere Hardwarekomponenten anbieten.

¹⁵ CSS: Cascading Style Sheets. Vergl. <http://www.w3.org/Style/CSS/>

Hersteller	Bezeichnung	Webdienst / TV-App	Kommentar	Entwicklerwebseite
Samsung	Smart TV	TV-Apps	Eigenes SDK. TV-App besteht aus HTML, JS, CSS, Bilder und Flash-Dateien.	http://www.samsungforum.com/
Philips	Net TV	Webdienst	Informationen bezüglich der Entwicklung können erst nach der Registrierung als Developer erhalten werden. Verwendet den CE-HTML-Standard ¹⁶	http://www.supportforum.philips.com/en/forumdisplay.php?5-Net-TV-Forum
Panasonic	Viera-Connect	TV-Apps	Kein Download am Gerät, sondern Online-Apps, die aber dank JavaScript den Zugriff auf das darunterliegende System erlauben.	http://developer.vieraconnect.com
LG	NetCast	TV-Apps	Bietet auch ein SDK an. TV-Apps bestehen aus den gleichen Komponenten wie die bei Samsung.	http://developer.lgappstv.com
LÖWE	Media-Net	Webdienst	Es konnten keine Informationen bezüglich Erweiterungen gefunden werden.	?
Sony	Bravia Internet	Webdienst	Erweiterung nur mit Hilfe von Yahoo-Widgets.	http://yahoo.sonydeveloper.com/
Sony	Google TV	TV-Apps	TV-Gerät das mit dem Betriebssystem Android betrieben wird. Siehe 0	http://www.google.com/tv/

Tabelle 1: Übersicht verschiedener Smart-TV-Gerätehersteller hinsichtlich Erweiterungsmöglichkeiten für eigene TV-Apps.

Die Tabelle 1 verdeutlicht, dass sich aus der heutigen Sicht eine Implementierung am ehesten an Smart-TV-Geräten des Herstellers Samsung oder LG realisieren lassen würde.

¹⁶ CE-HTML: Abkürzung für Consumer Electronic - HTML. CE-HTML ist ein Teil des CEA-2014-Standards. Siehe http://www.ce.org/Standards/browseByCommittee_2757.asp

3.4.3. Google TV

Die Suche nach einer geeigneten Hardware für das AAL-Frontend liefert relativ rasch die Google TV Plattform. Google TV nimmt in der TV-basierten und Set-Top-Box-basierten Klassifizierung eine Sonderstelle ein, da es sowohl als Smart-TV-Fernsehgerät als auch als Set-Top-Box angeboten wird. (Vergl. Google Inc., 2011)

Das besondere Merkmal von Google TV ist die eingesetzte Open-Source-Software namens Android, die auch bei einigen modernen Smartphones und Tablet-PCs eingesetzt wird. Im Gegensatz zu den proprietären Betriebssystemen anderer Smart-TV-Gerätehersteller, können hier nahezu uneingeschränkte Softwareerweiterungen durchgeführt werden. Das Betriebssystem selbst, aber auch jede Kommunikationsschnittstelle, sind offengelegt und dokumentiert. Für den Betrieb eines AAL-Frontends würde sich die Android-Plattform besonders gut eignen, da eine Implementierung sowohl auf dem TV-Gerät als auch auf der STB lauffähig wäre. Je nach Setting kann der Benutzer wählen, ob ein TV-Gerät oder die STB eingesetzt werden soll.

Der große Nachteil von Google TV ist, dass es in Europa noch nicht verfügbar ist. Auch die aktuellen Pressemeldungen, die besagen, dass der Hersteller Logitech aus dem Google TV-Geschäft aussteigen will, zeigen, dass es noch Zeit in Anspruch nehmen wird, bis sich die Plattform etabliert hat (Die Presse, 2011). Aus den soeben genannten Gründen eignet sich die Google TV-Plattform für die Implementierung eines Prototyps nicht und es muss eine andere Hardware eingesetzt werden.

3.5. Set-Top-Boxen

Eine Set-Top-Box ist ein Gerät, das einen Fernseher mit einer externen Signalquelle verbindet. Die Signalquelle wird von der STB so umgewandelt, dass sie am Fernsehgerät wiedergegeben werden kann (Lambert M Surhone, 2010). Die Begriffserklärung spezifiziert dabei die externe Signalquelle nicht genauer. Diese kann eine DVB-Signalquelle, eine IPTV-Signalquelle oder auch ein optisches Medium wie z.B. eine DVD sein. Letzteres wird jedoch als DVD-Player bezeichnet. Bei einem STB Setting fungiert das Fernsehgerät als ein reines Ausgabegerät, das neben der Bildhelligkeit oder der Lautstärke für nahezu keine weiteren Funktionalitäten zuständig ist. Alle anderen Aufgaben werden direkt von der STB verarbeitet. Benutzerbefehle wie z.B. das Wechseln des TV-Kanals oder der Aufruf von Teletext werden ebenfalls direkt an die STB gerichtet und auch dort verarbeitet.

Für den Betrieb einer AAL-Plattform hat ein STB-basiertes Setting einen wesentlichen Nachteil. Die Implementierung kann nicht sicherstellen, dass eine Alarmmeldung vom Anwender wahrgenommen wird. Dies gilt vor allem dann, wenn sich mehrere Set-Top-Boxen, wie z.B. ein DVD-Player und ein DVB-T-Empfänger, im Einsatz befinden. Während das Fernsehgerät für die Wiedergabe eines DVD-Inhaltes benutzt wird, bleiben die am DVB-T-Empfänger auftretenden Alarme unbemerkt.

In den meisten Fällen wird eine STB für den Empfang von linearen TV-Inhalten verwendet. Als Beispiel dafür kann ein preiswerter Sat- oder DVB-T-Receiver genannt werden. Diese STBs haben entweder keinen LAN-Anschluss, oder wenn doch, dann wird dieser meistens nur für ein Softwareupdate genutzt. Geräte wie z.B. Boxee¹⁷ und Apple TV¹⁸ besitzen dagegen einen LAN-Anschluss und können nichtlineare TV-Inhalte empfangen, haben jedoch keine Möglichkeiten DVB-Signale zu empfangen. Um möglichst viele Haushalte abdecken zu können und dabei auch noch den Empfang von nichtlinearen TV-Inhalten zu ermöglichen, wird für den Betrieb eines AAL-Frontends eine Kombination aus den beiden genannten Gerätearten benötigt. Aus diesem Grund werden hier zwei Set-Top-Boxen in Betracht gezogen, die sowohl diverse DVB-Empfangsmodule als auch einen LAN-Anschluss besitzen und zudem Softwareerweiterungen zulassen.

3.5.1. Dreambox & Enigma

Die Dreambox-Geräte des Herstellers Dream Multimedia GmbH erfüllen die Anforderungen, die für den Betrieb eines AAL-Frontends benötigt werden. Nichtlineare TV-Inhalte können über den integrierten LAN-Anschluss empfangen werden. Mit Hilfe von DVB-

¹⁷ Boxee: Siehe <http://www.boxee.tv>

¹⁸ Apple TV: Siehe <http://www.apple.com/de/appletv/>

Tuner-Steckkarten können die Geräte DVB-S-, DVB-T- und auch DVB-C-Signale empfangen (Dream Multimedia GmbH, 2011). Somit werden alle im Abschnitt 3.1 aufgelisteten für Europa üblichen Empfangsmöglichkeiten abgedeckt.

Die Dreambox verwendet ein Linux-basiertes Betriebssystem namens Opendreambox¹⁹, welches speziell auf die Hardware der STBs zugeschnitten ist. Auf diesem Betriebssystem wird das Programm Enigma bzw. Enigma2 ausgeführt, das schließlich die Benutzeroberfläche des Systems darstellt. Die Abbildung 6 zeigt zwei Beispiele für die Benutzeroberfläche von Enigma2. Der wesentliche Vorteil von Enigma ist die Erweiterbarkeit des Programms durch Softwarekomponenten, den sogenannten Plugins, die von Dreambox-Community entwickelt werden (Dream Multimedia GmbH, 2011 a).



Abbildung 6: Benutzeroberfläche des Programms Enigma2. Abbildung links: Die Wahl des TV-Senders, Abbildung rechts: Darstellung von Informationen zum aktuellen TV-Programm. Quelle: <http://www.saunalahti.fi/~znark/dreambox/dm7025/kuvagalleria/enigma2/>

3.5.2. ReelBox & VDR

Ein zweites System, das die Anforderungen für den Betrieb eines AAL-Frontends abdeckt ist die STB ReelBox Avantgarde II, die von Reel Multimedia Vertriebs GmbH vertrieben wird. Mit der ReelBox können ebenfalls TV-Signale über DVB-T, DVB-C und DVB-S empfangen werden (Reel Multimedia Vertriebs GmbH, 2011). Der integrierte LAN-Anschluss ermöglicht auch bei der ReelBox den Empfang und die Nutzung von nichtlinearen TV-Inhalten.

Die ReelBox verwendet das in der Linux Domäne bekannte Betriebssystem Ubuntu. Auf dem Betriebssystem wird - ähnlich wie bei der Dreambox - ein Programm ausgeführt, das die Benutzerschnittstelle für das System darstellt. Das eingesetzte Programm heißt bei der

¹⁹ Opendreambox: Siehe <http://opendreambox.org>

ReelBox Video Disk Recorder (VDR). Es wird von Klaus Schmidinger (Schmidinger, 2009) und einer Entwicklergemeinde erstellt und ständig weiterentwickelt (VDR Wiki, 2011). Das ursprünglich aus einem Hobbyprojekt entstandene VDR-Programm wird heute unter der GPL-Lizenz kostenlos zum Download bereitgestellt und kann ebenfalls über den Paketmanager auf einem Ubuntu Betriebssystem installiert werden. Analog zu der Dreambox, kann das VDR-Programm ebenfalls mit Hilfe von Plugins erweitert werden. Die VDR-Software wird für den Einsatz auf einer ReelBox von der Reel Multimedia Vertriebs GmbH speziell erweitert (Reel Multimedia Vertriebs GmbH, 2011).



Abbildung 7: Benutzeroberfläche des Programms VDR auf der ReelBox. Abbildung links: Die Wahl des TV-Senders, Abbildung rechts: Elektronischer Programmführer auf der ReelBox. Quelle: <http://wiki.reel-multimedia.com>

3.6. Entscheidungskriterien für die ReelBox & VDR-Software

Für die Implementierung eines Prototyps wurde die ReelBox ausgewählt, da die in diesem Kapitel erwähnten Hardwarekriterien vollkommen abgedeckt werden. Es werden sowohl alle 3 DVB-Empfangstechnologien unterstützt als auch die Möglichkeit geboten IPTV zu empfangen und zu verarbeiten. Auch im Bereich interaktives Fernsehen kann die ReelBox durch ihren LAN-Anschluss und ihre modulare, erweiterbare und Plugin-basierte VDR-Software mithalten. Aufkommende Standards, wie z.B. der HbbTV-Standard, lassen sich in das System mit Hilfe von Softwareupdates integrieren. Durch den eingebauten DVD-Player ermöglicht die ReelBox eine One-Box-Strategie, bei der nur ein Gerät neben dem Fernsehgerät benötigt wird. Im Gegensatz zu der Dreambox ermöglicht das auf der Reel-Box eingesetzte Betriebssystem Ubuntu den leichteren Einsatz von Drittsoftware und Paketen wie z.B. Bildverarbeitungspaketen oder XML-Parsern. Aus den soeben genannten Gründen wurde die ReelBox für die Installation und den Betrieb der Software ausgewählt.

4. Softwaredesign

Im Kapitel 2 wurden verschiedene AAL-Systeme vorgestellt und die grundlegenden Anforderungen an eine TV-basierte AAL-Plattform gestellt. Kapitel 3 beschäftigte sich mit der TV-Signalübertragung, den verschiedenen Empfangsgeräten und den Standards auf dem Gebiet interaktives Fernsehen. Dies führte zur Findung einer passenden Set-Top-Box, die für den Betrieb der Software verwendet wird. Das Ziel des Softwaredesigns besteht nun darin, die Software so zu konzeptionieren, dass sie den geforderten Anforderungen entspricht, für viele verschiedene AAL-Backendsysteme einsetzbar bleibt und auf der ausgewählten VDR-Hardwareplattform installiert und betrieben werden kann.

4.1. AAL - Funktionalität als VDR-Plugin

Das VDR-Programm ist ein modulares Programm, dessen Funktionsumfang durch sogenannte VDR-Plugins erweitert werden kann. Ein Plugin ist dabei eine vorkompilierte Linux - Programm-Bibliothek. Sie wird beim Starten des Systems durch die Angabe des Namens referenziert und so in das VDR-Programm einbezogen. Mittlerweile existieren zahlreiche Plugins die von einer Community erstellt werden. Sie sind in erster Linie in dem VDR-Wiki (VDR Wiki, 2011) anzutreffen. In einer VDR-Umgebung sind die Plugins mit den technischen Features von gewöhnlichen DVB-Empfangsgeräten vergleichbar. So wird sowohl die Funktion des elektronischen Programmführers (EPGs) als auch die des DVD-Players von einem eigenen Plugin bereitgestellt. Weitere Plugins ermöglichen z.B. Internetradio, den Empfang von Internetfernsehen (IPTV) oder auch die Bild-im-Bild-Funktion, die während der Wiedergabe eines TV-Programms eine Vorschau eines Zweitsenders in einem kleinen Fenster ermöglicht.

Plugins haben den enormen Vorteil, dass sie in eine Applikation eingefügt und auch wieder entfernt werden können. Dies ermöglicht die Gestaltung einer individuellen und auf die Bedürfnisse des Benutzers angepassten Applikation. Der modulare Aufbau erleichtert zudem die Aktualisierung, da sowohl die Applikation selbst als auch die einzelnen Plugins unabhängig voneinander aktualisiert werden können. Hinsichtlich einer AAL-Plattform eignet sich der modulare Aufbau ebenfalls gut, da die notwendigen AAL-Merkmale gebündelt als ein einziges Plugin realisiert werden können und dadurch von den Basisfunktionen des VDR-Programms getrennt bleiben.

4.2. AAL-Plattform als ein verteiltes System

Schill (Schill & Springer, 2007) definiert ein verteiltes System wie folgt: „Ein verteiltes System setzt sich aus mehreren Einzelkomponenten auf unterschiedlichen Rechnern zusammen, die in der Regel nicht über gemeinsamen Speicher verfügen und somit mittels Nach-

richtenaustausch kommunizieren, um in Kooperation eine gemeinsame Zielsetzung - etwa die Realisierung eines Geschäftsablaufs - zu erreichen.“ Ausgehend von dieser Definition ist die hier zu entwickelnde AAL-Plattform ebenfalls ein verteiltes System. Sie besteht aus einem Frontend und einem Backend, die auf unterschiedlichen Rechnern betrieben werden. Somit muss hier eine Kommunikationsstrecke für den Austausch von Nachrichten bzw. Daten zwischen den einzelnen Geräten gestaltet werden.

Aus der technisch-physikalischen Sicht bietet sich für den Datenaustausch eine netzwerk-basierte Kommunikationsstrecke an, da sie im Vergleich zu anderen Technologien, wie z.B. Bluetooth oder einer USB-Kabelverbindung, eine einfache und rasch realisierbare Lösung darstellt. Die ReelBox besitzt serienmäßig einen LAN-Anschluss und es besteht zudem die Möglichkeit, einen USB-WLAN-Stick mit dem Gerät zu betreiben. Damit kann auf das lokale Netzwerk zugegriffen und Daten mit Hilfe von TCP/IP²⁰ mit dem AAL-Backend ausgetauscht werden.

Da die Hardware für die Kommunikationsstrecke festgelegt wurde, muss nur noch aus der Sicht der Software bestimmt werden, wie der Nachrichtenaustausch vonstattengehen soll. Um eine lose Kopplung der Systeme zu ermöglichen, muss der Nachrichtenaustausch unabhängig von der Programmiersprache und des Betriebssystems erfolgen. Des Weiteren soll der Kommunikationsmechanismus möglichst einfach gestaltet werden, sodass am AAL-Backend nicht allzu viel Implementierungsaufwand erforderlich ist. Ein einfaches Client/Server-Modell (Vergl. Schill & Springer, 2007, S. 14ff), welches das Übertragungsprotokoll HTTP verwendet, deckt die angeführten Anforderungen ab und wird deshalb für den Nachrichtenaustausch verwendet.

Aus der Sicht der Benutzeroberfläche könnte man meinen, dass die Client/Server-Aufteilung mit AAL-Frontend als Client und AAL-Backend als Server eindeutig wäre. Dies ist jedoch nur bedingt richtig. Wie die Abbildung 4.2 zeigt, ist die Aufteilung nicht eindeutig, sondern von der verwendeten Funktion abhängig. In der Home-Control-Funktion fordert der Client (AAL-Frontend) eine Aktion (Licht einschalten) vom Server (AAL-Backend) an. In der Home-Monitoring-Funktion muss ein Alarm angezeigt und bestätigt werden. Hier ist das Backend die treibende Kraft (Client), die am Frontend (Server) das Anzeigen des Alarms bezwecken will. Das Frontend ist hier Dienstbringer, also der Server und das Backend ist der Dienstnutzer, also der Client.

²⁰ TCP/IP: Akronym für Transmission Control Protocol / Internet Protocol

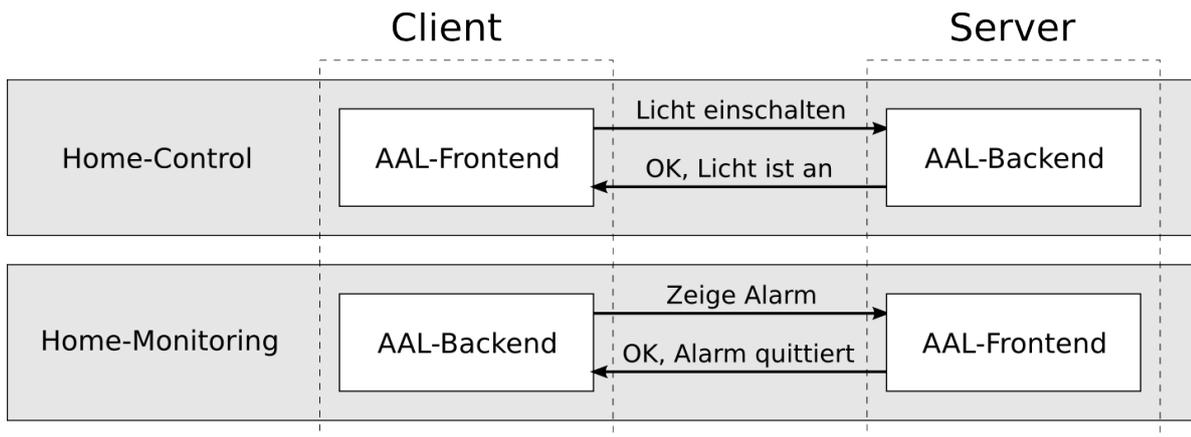


Abbildung 8: Client/Server-Modell der AAL-Plattform. In der Home-Control - Funktion fungiert das AAL-Frontend als Client und in der Home-Monitoring - Funktion als Server. Analoges gilt für das AAL-Backend.

Wie das soeben geschilderte Szenario verdeutlicht, benötigt das AAL-Frontend sowohl einen HTTP-Webserver, als auch einen HTTP-Client.

4.2.1. Auswahlkriterien für einen HTTP-Webserver

Bei der Suche nach einem passenden HTTP-Webserver wurden folgende Auswahlkriterien verwendet:

- **Lizenzbedingungen**

Das erste Auswahlkriterium sind die Lizenzbedingungen, unter den der Webserver vertrieben bzw. gestellt wird. Da der Prototyp unter der GPL-Lizenz entwickelt wird, dürfen keine Module oder Bibliotheken in die Software inkludiert werden, die restriktiveren Lizenzbedingungen unterzogen sind.

- **Sicherheit**

Die Sicherheit stellt ein wesentliches Auswahlkriterium dar, da das System unbefugten Personen den Zugang verwehren sollte. Mit heutzutage üblichen Sicherheitsmechanismen wie z.B. Datenverschlüsselung oder einer Authentifizierung sollte der Webserver ebenfalls umgehen können.

- **Einfache Integrierung**

Ein weiteres Auswahlkriterium stellt die einfache Integration in das VDR-System. Der Server soll entweder eine leicht zu integrierende Programmbibliothek, oder ein in der Programmiersprache C geschriebenes Programm sein.

- **Schlankes Design**

Das letzte Auswahlkriterium ist ein schlankes Design, da der Webserver innerhalb eines VDR-Plugins betrieben werden soll. Ein großer Funktionsumfang ist für diesen Einsatz nicht erforderlich.

4.2.2. Der Mongoose Webserver

Ausgehend von den soeben definierten Auswahlkriterien wurde der Mongoose Webserver ausgewählt. Der von Sergey Lyubka (Lyubka, 2011) entwickelte und unter die MIT-Lizenz²¹ gestellte Webserver besteht aus einem einzigen C Sourcecode-File. Für die Integration in ein VDR-Plugin eignet sich diese Single-File-Lösung hervorragend, da sonst keine externen Bibliotheken benötigt werden.

4.3. Umgang mit verschiedenartigen Meldungen

Neben der Anzeige von Sensoren und der Steuerung von Aktuatoren, besteht die zweite Primärfunktion des AAL-Frontends in der audiovisuellen Ausgabe von Meldungen am TV-Bildschirm. Da in einem AAL-System potenziell viele Ereignisse auftreten können, die zudem verschiedenartige Meldungen generieren, muss eine priorisierte Meldungsklassifikation durchgeführt werden. So soll verhindert werden, dass der Nutzer von jeder einzelnen Meldung in Kenntnis gesetzt wird, sondern nur von jenen, die eine gewisse Priorität besitzen.

4.3.1. Klassifizierung von Meldungen

Für das AAL-Frontend wurden die Klassen Alarmmeldung, Warnmeldung, Informationsmeldung, Home-Control-Aktivität und die stille Meldung festgelegt. Abgesehen von der Home-Control-Aktivität ist die Klasseneinteilung besonders für das AAL-Backend von Bedeutung. Das Backend hat die Aufgabe eine auszugebende Meldung zuerst in eine Klasse einzuteilen und dann an das Frontend zu senden. Damit kann das Frontend entscheiden, ob die ankommende Meldung sofort ausgegeben werden muss, oder auch später ausgegeben werden kann.

4.3.1.1. Alarmmeldung

Eine Alarmmeldung muss vom Backend ausgesendet werden, wenn ein akuter Handlungsbedarf besteht. Ein Beispiel für eine Alarmmeldung könnte die Nachricht „Der Rauchmelder in der Küche meldet Alarm!“ sein. Da es sich hier um ein Feuer handeln könnte, besteht ein akuter Handlungsbedarf.

²¹ MIT-Lizenz: Siehe <http://www.opensource.org/licenses/mit-license.php>

4.3.1.2. Warnmeldung

Das Backend kann in Ausnahmesituationen, die keine akute Handlung seitens des Benutzers erfordern, eine Warnmeldung ausgeben. Ein Beispiel für eine Warnmeldung könnte die Nachricht „Die Eingangstür ist nicht verschlossen!“ sein.

4.3.1.3. Informationsmeldung

Eine Informationsmeldung sollte ausgegeben werden, wenn gar kein Handlungsbedarf besteht, aber der Benutzer trotzdem auf eine Situation hingewiesen werden soll. Ein Beispiel für eine Informationsmeldung könnte die Nachricht „Die Raumtemperatur von 22°C wurde erreicht“ sein.

4.3.1.4. Stille Meldung

Eine stille Meldung sollte vom Backend ausgesendet werden, wenn der Status oder der Wert eines Sensors/Aktuators gerändert werden soll, der Benutzer von dieser Änderungsmaßnahme jedoch nicht in Kenntnis gesetzt werden muss. Ein Beispiel für eine Informationsmeldung könnte die Nachricht „Die Raumtemperatur ist um 1°C gestiegen“ sein.

4.3.1.5. Die Home-Control - Aktivität

Die Home-Control-Aktivität nimmt bei dieser Klassifizierung von Meldungen eine Sonderrolle ein. Sie wird nicht - wie die anderen vier Meldungen - vom Backend an das Frontend gesendet, sondern wird im Frontend generiert und für die Anzeige des Home-Control-Verlaufes verwendet. Die Anzeige einer Home-Control-Aktivität, wie z.B. das Betrachten eines Sensorwertes, wie sie in der Abbildung 10 c zu sehen ist, muss nach dem Auftreten und der Bestätigung eines Alarms wieder an der selben Stelle fortgeführt werden, an der sie sich vor der Alarmsituation befand. Aus diesem Grund muss eine Home-Control-Aktivität im Regelwerk des Systems mitberücksichtigt werden.

4.3.2. Regelwerk für die Aufzeichnung und die Anzeige von Meldungen

Das Verhalten des Systems muss so gestaltet werden, dass eintreffende Meldungen nicht verloren gehen, sondern im Hintergrund gesichert und zu einem gegebenen Zeitpunkt wiedergegeben werden können. Aus diesem Grund benötigt das System ein Regelwerk, das eine interne Verarbeitungskette bildet. Sie regelt sowohl die Anzeigereihenfolge als auch die Speicherreihenfolge von Meldungen. Auf der Seite der Anzeige wird bestimmt, was bereits angezeigt wurde, was aktuell angezeigt wird und was noch angezeigt werden muss. Auf der Seite der Speicherung muss eine Entscheidung getroffen werden, an wel-

cher Stelle eine neue Meldung in die Verarbeitungskette eingereicht werden kann. Bei jeder eintreffenden Meldung wird die Verarbeitungskette validiert und gegebenenfalls neu gereiht. So wird verhindert, dass z.B. eine Informationsmeldung angezeigt wird, obwohl eine neue Alarmmeldung eingetroffen ist. Das Regelwerk für die Verarbeitungskette besteht aus vier Parametern und kann der Tabelle 2 entnommen werden.

	Priorität	Speicherart	Anzeigeart	Anzeigedauer
Alarmmeldung	1	An die Alarm-Liste anhängen	FIFO ²²	Bis die manuelle Quittierung erfolgt
Warnmeldung	2	An die Warn-Liste anhängen	LIFO ²³	7 Sekunden
Informationsmeldung	3	Überschreibt die vorherige Informationsmeldung	Nur die zuletzt auftretende Meldung	4 Sekunden
Home-Control Aktivität	4	An die Show-Liste anhängen	LIFO	Bis andere Aktivität gewählt wird
Stille Meldung	-	Keine Speicherung Sofortige Ausführung	Keine Anzeige	Keine Anzeigedauer

Tabelle 2: Regelwerk für die festgelegten Meldungskategorien

4.3.2.1. Priorität

Der erste Parameter des Regelwerks ist die Priorität. Alle in der Verarbeitungskette vorkommenden bzw. zwischengespeicherten Meldungen werden in der Reihenfolge ihrer Priorität wiedergegeben. So werden zuerst alle Alarmmeldungen, gefolgt von allen Warnmeldungen und die letzte Informationsmeldung, die das System erreicht hat, angezeigt. Das Schlusslicht in der Verarbeitungskette bilden die Home-Control-Aktivitäten mit der niedrigsten Priorität 4. Stille Meldungen haben hingegen keine Priorität. Sie beeinflussen keine anderen Meldungen, weil sie unsichtbar im Hintergrund verarbeitet und nicht angezeigt werden müssen.

4.3.2.2. Speicherart

Die Speicherart bestimmt, an welche Stelle eine eintreffende Meldung zwischengespeichert werden soll. Die Meldungsarten der Priorität 1,2 und 4 haben intern eine eigene History-Liste, in denen die eintreffenden Meldungen nacheinander gespeichert werden.

²² FIFO: Akronym für „First In First Out“. Kennzeichnet die Abarbeitung von Elementen in einer Liste. Das erste Element, welches in die Liste eingefügt wurde, wird als erstes Element aus der Liste entfernt.

²³ LIFO: Akronym für „Last In First Out“. Das letzte Element, welches in die Liste eingefügt wurde, wird als erstes aus der Liste entfernt.

Die Informationsmeldung besitzt keine History-Liste, da diese Meldungsart keinen hohen Stellenwert hat und der Benutzer nicht von jeder neu auftretender Informationsmeldung in Kenntnis gesetzt werden muss. Eine neu eintreffende Informationsmeldung überschreibt somit eine bereits vorhandene Informationsmeldung. Dem Benutzer wird nur die neueste Meldung präsentiert. Auch stille Meldungen werden nicht zwischengespeichert, sondern sofort im Hintergrund ausgeführt.

4.3.2.3. Anzeigart

Die Anzeigart bestimmt in welcher Reihenfolge Meldungen der gleichen Priorität ausgegeben werden. Alarme werden nach dem FIFO - Prinzip gereiht. Die erste eintreffende Alarmmeldung wird auch als erstes ausgegeben. Die Meldungsarten der Priorität 2 und 4 verwenden das LIFO - Prinzip. Hier wird die zuletzt eintreffende Meldung als erstes ausgegeben. Diese Vorgehensweise ist insbesondere für die Home-Control-Funktion von essentieller Bedeutung, da nach der Anzeige einer höher priorisierten Meldung der Benutzer wieder zu der Anzeige zurückkehren muss, die vor der Meldung dargestellt wurde.

4.3.2.4. Anzeigedauer

Die Anzeigedauer wurde eingeführt, um zu verhindern, dass der Nutzer jede einzelne auftretende Meldung manuell bestätigen muss. Nur die Alarmmeldungen müssen - bedingt durch ihren akuten Handlungsbedarf - manuell quittiert werden. Warnmeldungen und Informationsmeldungen sind für 14 bzw. 7 Sekunden sichtbar, bevor sie automatisch ohne Zutun des Nutzers wieder vom TV-Bildschirm entfernt werden. Die Home-Control-Aktivität unterliegt auch keiner Zeitvorgabe, da diese Anzeige aktiv angefordert wurde und so lange aktiv bleibt, bis der Nutzer eine andere Home-Control-Aktivität wählt.

4.4. Steuerung

Die definierten Anforderungen verlangen die Steuerung der Benutzeroberfläche mit Hilfe der Gerätefernbedienung. Die Fernbedienung der ReelBox unterscheidet sich hinsichtlich Anzahl und Anordnung der Tasten kaum von anderen Fernbedienungen. Die ausschließliche Nutzung von Sondertasten für den Betrieb des AAL-Frontends kann nicht in Betracht gezogen werden, da die wenigen Sondertasten mit Symbolen versehen sind, die für eine AAL-Funktion nicht passen. Des Weiteren sind die Sondertasten zu weit voneinander entfernt und der Benutzer müsste regelmäßig den Blick auf die Fernbedienung richten, um die richtigen Tasten zu finden.

Sondertasten haben je nach Fernbedienungsmodell andere Funktionen, Symbole und auch Positionen. Die Ziffertasten und die Softkey-Tasten sind bei fast jeder Fernbedienung

im gleichen Bereich angesiedelt. Aus diesem Grund wurden diese Tastengruppen für die Steuerung des AAL-Frontends gewählt.

4.4.1. Ziffernblock als Auswahl Tasten

Der Ziffernblock mit den Nummern 0-9 bietet die Möglichkeit, 10 unterschiedliche Elemente auszuwählen und ist zudem bei den meisten Fernbedienungen im oberen Bereich angesiedelt. Der Benutzer kann den Ziffernblock somit relativ schnell lokalisieren. Diese Tasten können in der Home-Control-Funktion dazu verwendet werden, um einzelne Elemente auszuwählen. Dafür wird zuerst jedes auswählbare Element mit einer Zahl gekennzeichnet. Die Eingabe dieser Zahl auf dem Ziffernblock der Fernbedienung führt dann schließlich zur Auswahl des Elementes. Wenn das AAL-Frontend z.B. einen Raum mit 2 Sensoren und 3 Aktuatoren visualisiert, dann können die Sensoren durch die Ziffern 1-2 und die Aktuatoren durch die Ziffern 3-5 erreicht werden. Diese sogenannten Hotkeys bzw. Tastaturkurzbefehle sind in der Abbildung 10 verdeutlicht. Die Implementierung soll mit dem Zahlenbereich von 0 bis 9 auskommen. Diese Limitierung erlaubt nur 10 auswählbare Elemente pro Anzeige. Um bei einer Folgeanzeige ebenfalls den Zahlenbereich 0-9 verwenden zu können, muss die Zuweisung einer Zahl zu einem Element dynamisch erfolgen. Da eine dynamische Vergabe von Tastenkürzel manche Benutzer irritieren könnte, soll die Nutzung von fixen Hotkeys ebenfalls ermöglicht werden. Für bestimmte Elemente wird dann ein fest vorgegebener und immer gleichbleibender Hotkey verwendet.

4.4.2. Softkey-Tasten als Kontrolltasten

Neben dem Ziffernblock, der eine Auswahl ermöglicht, können die vier Softkey-Tasten Rot, Grün, Gelb und Blau dazu verwendet werden, um die getroffene Auswahl zu steuern. Die Softkey-Tasten sind bei den meisten TV- bzw. STB-Fernbedienungen in der gleichen, oben angeführten, Reihenfolge angeordnet. Somit können auch farbenblinde Personen anhand der Tastenposition unterscheiden, welche angezeigte Funktion auf welcher Taste zu finden ist. Da verschiedenartige Elemente z.B. Sensor, Aktuator, Wohnungsraum usw. ausgewählt werden können, ist die Belegung der Kontrolltasten vom jeweiligen Element abhängig. Die unterschiedlichen Funktionen und Sichtbarkeiten der Kontrolltasten werden im Abschnitt 4.5.1.3 detailliert behandelt.

4.5. Visualisierung

Laposa (Laposa, 2005, S. 9) stellt fest, dass das Fernsehgerät primär ein visuelles Medium ist. Aus diesem Grund ist die Visualisierung auch ein ausschlaggebender Bestandteil des AAL-Frontends. Die Herausforderungen, die sich hier ergeben, sind sowohl struktureller als auch technischer Natur. Der technische Bereich beschäftigt sich mit der Fragestel-

lung, wie AAL-Inhalte in einem VDR-System ausgegeben werden können. Auf der strukturellen Seite werden Themen der Visualisierung behandelt wie z.B. Positionierung, Anordnung und Priorität von verschiedenen AAL-Inhalten.

4.5.1. Strukturelle Aspekte

Das AAL-Frontend soll - wie bereits in den Softwareanforderungen geschildert - zwei Anwendungsfälle abdecken. Die Home-Monitoring-Funktion und die Home-Control-Funktion. Bei der Visualisierung von diesen Anwendungsfällen müssen die besonderen Eigenheiten wie z.B. der limitierte Anzeigebereich mitberücksichtigt werden. Bei der Home-Monitoring-Funktion können Features wie z.B. Scrollbalken nicht verwendet werden. Der limitierte Anzeigebereich muss so aufgeteilt und ausgenutzt werden, dass eine Alarmmeldung auf dem TV-Gerät jederzeit und ohne jegliches Zutun des Benutzers sichtbar wird und auch bleibt. Die Home-Control Funktionalität benötigt hingegen eine möglichst große Granularität bei der Darstellung der im Haushalt verbauten Sensoren und Aktuatoren. Hier würden sich horizontale und vertikale Scrollbalken eventuell bezahlt machen. Diese zwei Aspekte verdeutlichen die Problematik. Es werden einerseits fixe Bereiche z.B. für die Alarmierungen benötigt, aber andererseits nehmen gerade diese fixen Bereiche einen nicht unerheblichen Teil des ohnehin limitieren Anzeigebereiches ein. Um beide Anforderungen gerecht zu werden kann in den fixen Bereichen eine hierarchische Darstellung mit mehreren level-of-details angewendet werden (Vergl. Wang & Shen, 2005). Durch diese Methode wird die geforderte Granularität erreicht und zudem zusätzlicher Platz für die Darstellung von Alarmmeldungen und der „globalen“ Navigation (Siehe Abschnitt 4.5.1.3) geschaffen. Abbildung 9 verdeutlicht die drei Bereiche Übersicht-, Inhalt- und Steuerbereich.

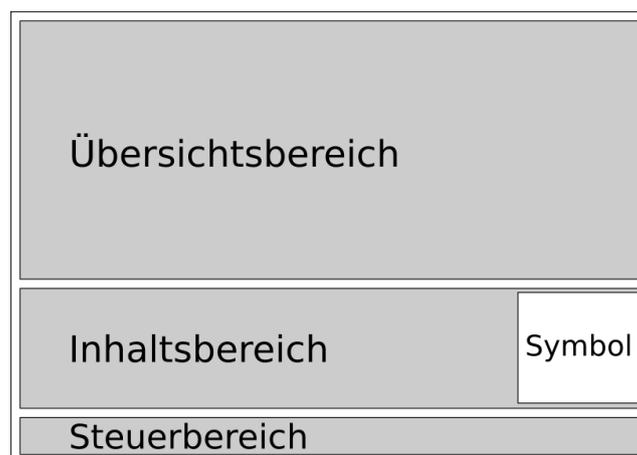


Abbildung 9: Die drei Visualisierungsbereiche des AAL - Frontends

4.5.1.1. Übersichtsbereich

Der Übersichtsbereich visualisiert das Umfeld der Home-Control-Funktion. Hier werden die verfügbaren Sensoren und Aktuatoren dargestellt und dem Benutzer zur Auswahl angeboten. Der Bereich ist nur in der Home-Control-Funktion sichtbar, also in der Situation, in der ein Sensor oder einen Aktuator gezielt ausgewählt und dessen Wert betrachtet oder gesteuert wird.

Um die Fülle an Sensoren und Aktuatoren in dem begrenzten Anzeigefeld darstellen zu können, wird hier die soeben erwähnte hierarchische Darstellung mit mehreren levels-of-details angewendet. Der Benutzer erhält vorerst mehrere Kategorien wie z.B. „Licht“, „Heizung“ oder „Jalousien“ angezeigt. Nach der Wahl einer Kategorie werden im zweiten Schritt die Unterkategorien wie z.B. „Wohnzimmerlicht“, „Garage“ oder „Außenbeleuchtung“ visualisiert. Natürlich kann die Kategorisierung auch anders ausfallen und z.B. die Struktur „Wohnung-Raum-Sensor/Aktuator“ abbilden. Hier wird zuerst die Wohnung mit den einzelnen Räumen dargestellt. Nach der Wahl eines Raumes wird dieser vergrößert angezeigt und die darin befindlichen Sensoren und Aktuatoren als Symbole visualisiert. Die Wahl eines Sensors oder eines Aktuators leitet schließlich die Detailanzeige im Inhaltsbereich ein. Die zwei unterschiedlichen Startmöglichkeiten für den Übersichtsbereich sind in der Abbildung 10 (a) und Abbildung 10 (b) dargestellt. Ein Beispiel für die hierarchische Darstellung kann ebenfalls der Abbildung 10 (b) bis (d) entnommen werden.

4.5.1.2. Inhaltsbereich

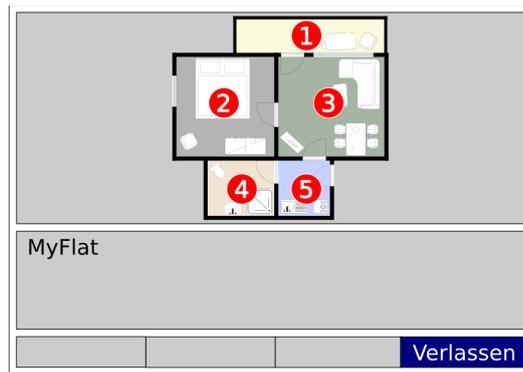
Der Inhaltsbereich ist sowohl für die Darstellung von Alarmmeldungen, als auch für die Darstellung von Detailinformationen eines Sensors bzw. eines Aktuators zuständig. Dieser Bereich ist - neben dem Steuerbereich - bei jeder AAL-Aktivität die am TV-Gerät ausgegeben wird, sichtbar und unterscheidet sich wesentlich vom Übersichtsbereich. Im Sinne der Fokus & Kontext - Visualisierung (Vergl. Wang & Shen, 2005, S. 6) ist der Übersichtsbereich für den Kontext zuständig. Der Inhaltsbereich, der die wesentlichen Vorgänge darstellt, soll hingegen den Fokus des Benutzers bewahren. Aus diesem Grund wird dieser Bereich sowohl für die Home-Control-, als auch für die Home-Monitoring-Funktion verwendet. Der fixe Bereich bietet auch bei der priorisierten Darstellung von Ereignissen einen gewissen Vorteil. Die Home-Monitoring Funktion - also die Ausgabe von Alarmmeldungen - hat eine höhere Priorität, als die Home-Control-Funktion. Tritt solch eine Alarmierung ein, dann soll der aktuelle Inhalt des Inhaltsbereiches mit dem, der Alarmmeldung überschrieben werden und so die Aufmerksamkeit des Benutzers auf das höherpriorisierte Ereignis lenken. Nach der Quittierung der Alarmmeldung kann sich der Inhaltsbereich wieder der vorherigen Anzeige widmen.

Der Informationsbereich ist in zwei Abschnitte unterteilt: den linken Abschnitt, der die Vorgänge textuell oder tabellarisch darstellt und den rechten Abschnitt, der die Vorgänge durch ein repräsentatives Symbol untermalt (siehe Abbildung 10 (d) und (e)). Die symbolische Darstellung bietet den Vorteil, dass der präsentierte Inhalt rascher erfasst werden kann. Diese Symbol- bzw. Icon-basierte Darstellungsart hat sich mittlerweile bewährt und wird auch bei Smartphones oder Tablet-PCs verwendet.

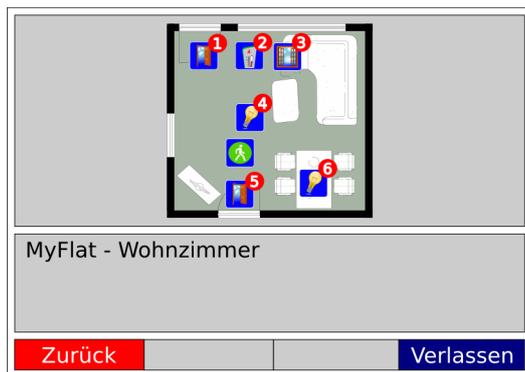
Da der Übersichtsbereich eine hierarchische Darstellung anwendet, in dem ein Bild von einem Folgebild überschrieben wird, benötigt der Nutzer Informationen darüber, in welchem Abschnitt der Applikation er sich momentan befindet. Diese Information kann über eine dynamische Namensgebung bzw. Breadcrumbs entnommen werden. Für die im oberen Abschnitt erwähnten Struktur „Wohnung-Raum-Sensor/Aktuator“ würde solch ein Namenspfad z.B. wie folgt aussehen: MyFlat-Wohnzimmer-Licht. Aus dieser Komposition ist es für den Benutzer ersichtlich in welcher Ebene er sich aktuell befindet und welcher Aktuator grade gesteuert wird. Weil die Steuerung eines Aktuators im Inhaltsbereich erfolgt, bietet sich der Bereich auch für die Darstellung des dynamischen Namens an. Der sukzessive Aufbau des Namens ist in der Abbildung 10 (b) bis (d) dargestellt.



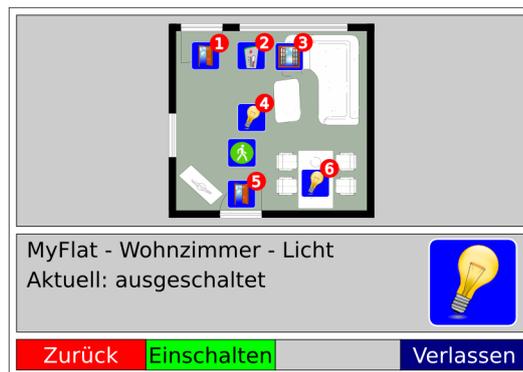
(a)



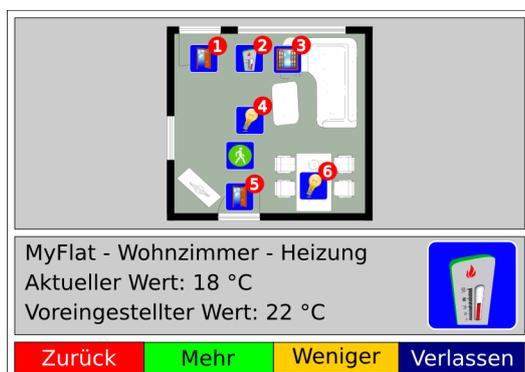
(b)



(c)



(d)



(e)

Abbildung 10: Zwei Möglichkeiten für verschieden Startanzeigen des Übersichtsbereiches (a) und (b). Dynamische Namensgebung und die Auswahl eines Lichttaktors (b), (c) und (d). Dynamische Namensgebung und die Auswahl der Heizungssteuerung (b), (c) und (e).

4.5.1.3. Steuerbereich

Der Steuerbereich ist der unterste Teil der AAL-Anzeige und rendert die vier Softkey-Tasten der Fernbedienung Rot, Grün, Gelb und Blau. Die Funktionsbelegung und die Sichtbarkeit der Tasten sind von der aktuellen Anzeige des Inhaltsbereiches abhängig. In der Tabelle 3 sind die verschiedenen Funktionen und die Sichtbarkeiten aufgeschlüsselt.

Taste	Bedeutung	Sichtbar	Kommentar
Rot	Zurück	Bei Folgeanzeigen	Kehrt zur vorherigen Anzeige zurück.
Grün	Ein- oder Ausschalten	Bei Binäraktuatoren	Schaltet ein Aktuator ein bzw. wieder aus falls sich der Aktuator im eingeschalteten Zustand befand.
	Wert verringern	Bei n-bit Aktuatoren ²⁴ Aktueller Wert > Min	Sichtbar z.B. bei einem Dimmer, dessen Wert noch nicht 0 erreicht hat.
Gelb	Wert erhöhen	Bei n-bit Aktuatoren. Aktueller Wert < Max	Sichtbar z.B. bei einem Dimmer, dessen Wert noch nicht das Maximum erreicht hat.
Blau	Verlassen	Immer	Schließt das AAL-Plugin und kehrt zum Fernsehprogramm im Vollbildmodus zurück.

Tabelle 3: Funktionsbelegungen und Sichtbarkeit der vier Softkey-Tasten

4.5.2. Technische Aspekte

Auf der technischen Seite stellt sich die Frage, wie Texte und Grafik auf dem VDR-System gerendert und dargestellt werden können. Zudem muss eine Überlagerung des normalen Fernsehbildes mit den Aktivitäten, die von der AAL-Umgebung gemeldet werden, erfolgen.

Ausgehend von den Anzeigemöglichkeiten die eine unveränderte Software einer ReelBox visualisiert, muss es Möglichkeiten geben, die eine Überblendung des aktuellen Fernsehprogrammes erlauben. Der Lautstärkebalken, der über das normale Fernsehprogramm gelegt wird, ist ein Anzeichen dafür, dass das System eine Erweiterung hinsichtlich dieser Funktionalität erlaubt. Das VDR-System ist - wie bereits erwähnt - eine Software, die unter GPL steht. Der Sourcecode der Implementierung ist zugänglich und kann an die eigenen Bedürfnisse angepasst werden. Ist diese Überblendungsfunktion jedoch auch mit Hilfe eines Plugins realisierbar, sodass kein Eingriff in das Kernsystem nötig wird? Das VDR-API stellt genau für diese Zwecke die Schnittstelle „On Screen Display“ (OSD) zu Verfügung, die in einer Plugin-Umgebung verwendet werden kann.

Für die Visualisierung des Inhaltsbereiches - wie er im Abschnitt 4.5.1.2 beschrieben wird - ist ein Umgang mit Text und Bilddaten von Nöten. Mit den Schnittstellen die von den OSD-Klassen zu Verfügung gestellt werden, können nur primitive Objekte und Text gerendert werden. Für die Darstellung von Text reichen diese Schnittstellen aus. Für die Darstellung von Symbolen jedoch leider nicht. Repräsentative Symbole bestehen aus mehr Elementen

²⁴ n-bit Aktuator. Ein Aktuator der n-bit Zustände kennt. Im Unterschied zu einem Binäraktuator, der nur 2 Zustände (ein/aus) kennt, kann ein n-bit Aktuator 2^n Zustände annehmen. Beispiel: Lichtdimmer.

als nur aus Linien, Rechtecken und Ellipsen. Hier müssen echte Rastergrafiken wiedergegeben werden. Für diese Zwecke stellen die OSD-Klassen Schnittstellen bereit, die mit ARGB - Pixeln²⁵ umgehen können. Für die Implementierungsphase ist damit sowohl die Frage der Möglichkeit einer Überblendung als auch die Möglichkeit der Visualisierung von komplexeren Bildern und Symbolen geklärt.

4.6. Akustische Ausgabe

Laut den Softwareanforderungen soll die akustische Ausgabe auf zwei unterschiedliche Arten möglich sein. Einerseits soll in Alarmsituationen ein kurzes, akustisches Warnsignal ausgegeben und andererseits muss die Möglichkeit geschaffen werden, dass Alarmmeldungen vom System automatisch vorgelesen werden. Letzteres wird auch Text-To-Speech (TTS), also die Umwandlung von schriftlich vorhandenem Text in eine sprachliche Ausgabe, genannt. Es bedarf unterschiedlicher Vorgehensweisen bzw. Technologien, um diesen zwei unterschiedlichen Anforderungen gerecht zu werden.

4.6.1. Wiedergabe von Audiodateien

Der Umgang mit Audiodateien wird sowohl für die Warnsignale, als auch für das Vorlesen von Alarm- und Warnmeldungen benötigt. Für die Wiedergabe wird ein Audioplayer verwendet, der zumindest mit dem RIFF WAVE-Dateiformat²⁶ umgehen kann. Der Player muss entweder direkt im VDR-Programm integriert sein oder mit dem VDR-Programm eng zusammenarbeiten um den folgenden Ablauf zu ermöglichen.

- Die Lautstärke des aktuell laufenden Fernsehprogramms minimieren oder gegebenenfalls völlig ausschalten.
- Einstellung der vordefinierten Lautstärke für die Audiowiedergabe.
- Wiedergabe einer übergebenen WAV-Datei.
- Wiederherstellung der ursprünglich eingestellten Lautstärke des aktuell laufenden Fernsehprogramms.

4.6.2. Text-To-Speech-Ausgabe

Das Vorlesen von Alarm- und Warnmeldungen erfordert die Einbindung von Drittkomponenten, da solch eine Funktion weder vom VDR-Programm selbst, noch von einem VDR-Plugin ermöglicht wird. Auch eine eigenständige TTS-Implementierung kann nicht in Betracht gezogen werden, da diese ein eigenes und zudem recht umfangreiches For-

²⁵ ARGB - Pixel: Pixel das neben den Farbkanälen Rot, Grün und Blau noch zusätzlich einen Alpha-Kanal besitzen und somit Transparenz erlauben.

²⁶ RIFF WAVE: <https://ccrma.stanford.edu/courses/422/projects/WaveFormat/>

schungsumfeld ist, welches über den Rahmen dieser Diplomarbeit hinausgehen würde. Die Aufgabenstellung beschränkt sich somit auf das Auffinden einer geeigneten TTS-Engine und auf das Modellieren der Kommunikationsstrecke zwischen dem AAL-Plugin und der TTS-Engine.

4.6.2.1. Kommunikationsstrecke AAL-Plugin - TTS-Engine

Die Kommunikationsstrecke bzw. die nötigen Abläufe, die durchgeführt werden müssen, bevor aus einer AAL-Alarmmeldung eine synthetisierte Audioausgabe generiert wird, sind in der Abbildung 11 dargestellt. Die Abbildung zeigt zugleich zwei verschiedene Verarbeitungswege. Die direkte TTS-Wiedergabe mit dem Pfad 1-2-3 und die Rückgabe der Kontrolle an das AAL-Plugin mit dem Pfad 1-2-4-5-6.

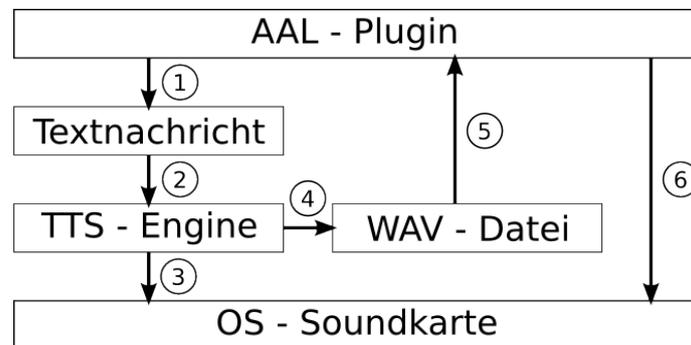


Abbildung 11: Ablauf der TTS-Audioausgabe für Alarme und Warnmeldungen. Die Schritte 1-3 stellen eine direkte Ausgabe, die von der TTS-Engine gesteuert wird, dar. Die Schritte 1,2,4,5 und 6 stellen eine indirekte Ausgabe, die durch das AAL-Plugin gesteuert wird, dar.

Direkte Ausgabe

Bei der direkten Ausgabe wird die Textnachricht von TTS-Engine selbst ausgegeben bzw. an das Betriebssystem gesendet und dort von der entsprechenden Hardware ausgegeben. Hier hat man nahezu keine Möglichkeiten in den Ablaufprozess einzugreifen. Die Ausgabe erfolgt für gewöhnlich unmittelbar nach der Verarbeitung der Textnachricht. Einstellungen wie die Regelung der Lautstärke des aktuellen Fernsehprogrammes oder der Alarmmeldung können nicht vorgenommen werden. Natürlich sind die Steuermöglichkeiten von der eingesetzten TTS-Engine abhängig, der Einsatz einer direkten und dadurch TTS-Engine-abhängigen Audioausgabe erweist sich jedoch als wenig praktikabel.

Rückgabe der Kontrolle an das AAL-Plugin

Der zweite Ansatz, die Rückgabe der Kontrolle an das AAL-Plugin, ermöglicht dagegen die Regelung der Lautstärke und ist zudem unabhängig von der eingesetzten TTS-Engine. Dieser flexible Ansatz ermöglicht sogar die Verwendung von verschiedenen TTS-Systemen. Die Wiedergabe der WAV-Dateien und die Lautstärkeregelung muss nicht se-

parat implementiert werden, sondern kann von dem im Abschnitt 4.6.1 beschriebenen Player vorgenommen werden.

Zusammenfassend zeigt sich, dass die Methode der Kontrollrückgabe an das AAL-Plugin einige Vorteile gegenüber der direkten Methode aufweist. Aus diesem Grund wird die Implementierung diesen Ansatz verfolgen.

4.6.2.2. Auswahlkriterien für eine TTS-Engine

Bei der Suche nach einer passenden TTS-Engine werden Auswahlkriterien benötigt, um die Fülle an verfügbaren TTS-Systemen sukzessive eingrenzen zu können. Für die Verwendung in Kombination mit einem AAL-Plugin wurden folgende, der Reihenfolge ihrer Priorität aufgelisteten, Auswahlkriterien verwendet:

- **Verwendete Sprache**

Die verwendete Sprache bzw. die Möglichkeiten eine bestimmte Sprache auszuwählen, stellt ein primäres Auswahlkriterium dar. Da der Prototyp vorwiegend im deutschsprachigen Raum eingesetzt werden soll, muss die TTS-Engine eine Ausgabe in deutscher Sprache unterstützen.

- **Lizenzbedingungen**

Wie schon vorab beim HTTP-Webserver, gilt auch für die TTS-Engine, dass keine Module oder Bibliotheken verwendet werden können, die eine restriktivere Lizenzbedingung besitzen als die der GPL.

- **Aussprache und Verständlichkeit**

Ein weiteres Auswahlkriterium ist die Aussprache und die Verständlichkeit des Systems. Viele Systeme unterstützen zwar die deutsche Sprache, klingen aber sehr „englisch“ weil sie ursprünglich für die englische Sprache entwickelt wurden. Gerade für die Zielgruppe 65+ ist eine klare und verständliche Aussprache besonders von Bedeutung. Sie könnte die Akzeptanz des Systems steigern. Aus diesem Grund werden Systeme präferiert, die eine klare deutsche Aussprache unterstützen.

- **Betriebssystemanforderungen und Kommunikationsschnittstellen**

Das letzte Auswahlkriterium stellten die Betriebssystemanforderungen und die Implementierungsmöglichkeiten dar. Da die Software auf der ReelBox eingesetzt wird, muss die verwendete TTS-Engine entweder auf einem Ubuntu-Betriebssystem ausführbar sein, oder zumindest Kommunikationsschnittstellen zur Verfügung stellen, die von einem VDR-Plugin aus erreicht werden können.

4.6.2.3. Die Wahl von zwei TTS-Systemen

Die Menge an verfügbaren TTS-Systemen kann nun mit Hilfe der Auswahlkriterien sukzessive eingeengt werden, bis eine passende TTS-Engige gefunden wird. Eine enorme Hilfestellung beim Vergleich von verschiedenen Systemen bietet die aktuelle und sehr umfangreiche Projektseite von Dr. Felix Burkhardt (Burkhardt, 2011). Burkhardt listet sowohl kommerzielle als auch universitäre und frei zugängliche TTS-Systeme auf. Zudem stellt er Testsamples zu Verfügung, die einen direkten Vergleich ermöglichen. Die Vergleichsuntersuchung lieferte ausgehend von den Auswahlkriterien zwei Systeme, die für den Prototyp eingesetzt werden können: die Open-Source-Software eSpeak (Jonathan Duddington, 2011) und die Plattform namens MARY (Deutsches Forschungszentrum für Künstliche Intelligenz, 2010).

eSpeak

Der große Vorteil von eSpeak liegt darin, dass die Software auch als ein vorkompiliertes Linux Paket erhältlich ist. Da die Software auf einer Linux Umgebung betrieben wird, kann der Paketmanager²⁷ des Betriebssystems dazu verwendet werden, um die eSpeak Software zu installieren. Nach der Installation kann eSpeak über Kommandozeilenbefehle angesprochen werden. Für das AAL-Plugin stellt die Kommunikation über die Kommandozeile kein Problem dar, da das Plugin durch die C++ API Zugriff auf das Betriebssystem hat.

MARY

Das vom Deutschen Forschungszentrum für Künstliche Intelligenz GmbH entwickelte System MARY ist eine Java Anwendung, die als ein Server betrieben wird. Die Clients könnten über HTTP, TCP/IP-Sockets oder auch Kommandozeilenbefehle mit dem Server kommunizieren. Im Vergleich zur eSpeak-Software ist das MARY-System ein wesentlich komplexeres und schwieriger zu installierendes System. Die deutschsprachigen Datenbanken erfordern rund 1.2 GB an Festplattenkapazität. Des Weiteren ist ein leistungsstarker Rechner erforderlich, damit die Antwortzeiten akzeptable Werte erreichen. Der wesentliche Vorteil liegt bei MARY jedoch in der deutlich verständlicheren Ausgabe.

²⁷ Paketmanager: Eine Software die Softwarepakete und Applikationen in einem Betriebssystem verwaltet.

4.7. Abbildung des Settings

In der Home-Monitoring-Funktion benötigt das AAL-Frontend keine besondere Kenntnis über das Umfeld, in dem es eingesetzt wird. Es muss nicht wissen, wie viele Sensoren und Aktuatoren das Setting insgesamt hat, oder in welchem Raum sich diese Signalgeber und Signalnehmer befinden. Für die Wiedergabe von Alarmmeldungen ist es ausreichend, wenn nur das AAL-Backend das Setting kennt. Die Sensoren melden ihren Status an das Backend und dieses formuliert in Ausnahmesituationen eine Meldung, die an das AAL-Frontend gesendet wird. Das Frontend muss diese Meldung nur textuell bzw. akustisch ausgeben. Die Meldung muss am Frontend nicht interpretiert werden. Ein Beispiel für diese Situation könnte die schon zitierte Meldung „Der Rauchmelder in der Küche meldet Alarm!“ sein. Da das Setting dem Benutzer bekannt ist, ist es für ihn auch ersichtlich, dass der Alarm aus der Küche kommt und dass sich die Küche z.B. im unteren Stockwerk des Hauses neben dem Wohnzimmer befindet. Für die bloße Wiedergabe der Alarmmeldung ist die Lage der Küche für das Frontend nicht von wesentlicher Relevanz.

Für die Home-Control-Funktion sieht die Situation anders aus. Hier ist das Setting sehr wichtig, da in diesem Betrieb einzelne Sensoren und Aktuatoren gezielt ausgewählt und gesteuert werden. Dafür benötigt das Frontend genaue Informationen über die Lage und die Raumzuordnung von jedem einzelnen Sensor und Aktuator. Auch die Geometrie und die Größe der Räume, sowie die Lage innerhalb der Wohnung müssen dem AAL-Frontend bekannt sein, damit eine Navigation und die im Abschnitt 4.5.1 beschriebene level-of-details-Visualisierung ermöglicht werden können.

Die soeben beschriebenen Informationen sind für gewöhnlich in dem jeweiligen AAL-Backend hinterlegt. Auf welche Art und Weise sie jedoch hinterlegt sind hängt von dem eingesetzten Backend ab. Um trotz dieser Unbestimmtheit ein generisches AAL-Frontend zu ermöglichen, wird ein minimales Set von Parametern definiert, sodass ein Objekt innerhalb des Settings beschrieben werden kann. Eine Liste solcher Objekte ermöglicht dann schließlich die Abbildung des Settings am Frontend. Unter einem Objekt ist nicht nur ein Aktuator oder ein Sensor zu verstehen, sondern jedes Objekt, das angezeigt oder ausgewählt werden kann. Das kann z.B. ein Raum innerhalb der Wohnung oder auch die Garage eines Hauses sein. Um die Leserlichkeit zu fördern, wird ein Objekt innerhalb des Settings nachfolgend als AALItem bezeichnet.

4.7.1. AALItem

Die Darstellung des Settings muss am Frontend nicht allzu sehr detailliert sein. Wie in Abbildung 10 verdeutlicht, reicht die ungefähre Form-, Position- und Größenangabe der einzel-

nen AALItems aus. Aus diesem Grund werden folgende Parameter für die Beschreibung eines AALItems verwendet:

- **Identifikation**

Identifikation eines Objektes. Kann ein Name, eine Identifikationsnummer oder auch die Kombination aus beiden sein.

- **Typus**

Beschreibt die Art des Objektes. Kann z.B. ein Sensor, ein Aktuator oder auch ein übergeordnetes Objekt wie z.B. ein Raum oder eine Wohnung sein.

- **Elternobjekt**

Ist eine Referenz auf das Elternobjekt. Ermöglicht die Abbildung einer Hierarchie wie z.B. Wohnung-Raum-Sensor.

- **Position und Größe**

Beschreibt die Position und die Größe des Objektes. Besonders für Aktuatoren und Sensoren ist die Position innerhalb des Raumes relevant. Die Größe von Objekten ist für übergeordnete Objekte, wie z.B. für den Raum, von Bedeutung.

- **Geometrie**

Beschreibt die Form eines Objektes. Die Geometrie spielt nur für übergeordnete Objekte eine Rolle und wird verwendet, um z.B. die Form eines Raumes zu beschreiben.

Die hier beschriebenen Parameter sind nicht als Klassenattribute zu verstehen, sondern beschreiben nur grob die Eigenschaften von AALItems, die für die Visualisierung eines Settings benötigt werden. Aus diesem Grund werden hier auch keine Datentypen aufgelistet.

4.7.2. Lokale Abbildung vs. Just-in-time-Anforderung über das Netzwerk

Nachdem die Parameter festgelegt wurden, können im nächsten Schritt Überlegungen angestellt werden, wie die AALItems auch dem Frontend zugänglich gemacht werden können. Dafür bieten sich folgende Möglichkeiten an:

- Just-in-time-Anforderung über das Netzwerk.
- Lokale Abbildung mit Hilfe einer Datenbank oder einer XML-Datei.

Nachfolgend werden die einzelnen Vor- und Nachteile der zwei aufgeführten Möglichkeiten diskutiert.

4.7.2.1. Just-in-time - Anforderung

Bei der Just-in-time-Anforderung wird das Setting nicht lokal abgebildet, sondern im Bedarfsfall beim Backend über das Netzwerk angefordert. Wenn z.B. wie in Abbildung 10 (c) ein Wohnzimmer dargestellt wird, dann fordert das Frontend gezielt die einzelnen AALItems für das Wohnzimmer beim AAL-Backend an. Diese Vorgehensweise hat den Vorteil, dass das AAL-Frontend keine allzu große Logikimplementierung benötigt. Es muss nur alle Subobjekte eines Objektes beim Backend anfordern und sie entsprechend der AALItem-Parameter am Bildschirm positionieren. Ein weiterer Vorteil ergibt sich bei der Änderung des Settings. Eine Änderung wird nur noch an einer Stelle, dem AAL-Backend, nachgezogen. Da die Daten fortlaufend vom Backend angefordert werden, wird eine Änderungen des Settings auch automatisch an das Frontend übertragen.

Der große Nachteil dieser Vorgehensweise liegt in der gegenseitigen Anpassung der Systeme. Das Frontend muss das jeweils eingesetzte Backend kennen, um so die Daten anfordern zu können. Analog ist es auch für das Backend notwendig eine Schnittstelle für das Frontend zu Verfügung zu stellen, die eine Liste von AALItems liefern kann. Hier ist zudem fraglich, ob das jeweilige Backend alle nötigen Parameter eines AALItems kennt und auch liefern kann. Der etwas weniger bedeutsame, jedoch erwähnenswerte Nachteil dieser Methode ist der starke Kommunikationsbedarf mit dem AAL-Backend. Das Backend muss permanent erreichbar sein und Anfragen unverzüglich beantworten können, damit am Frontend keine allzu langen Verzögerungen entstehen.

4.7.2.2. Lokale Abbildung

Die lokale Abbildung des Settings erfordert im Gegensatz zur just-in-time-Anforderung keine permanente Erreichbarkeit des Backends. Hier wird das Setting dem Frontend gewissermaßen angelernt und kann danach für die Visualisierung, die Navigation oder auch für die dynamische Namensgebung (siehe Abschnitt 4.5.1.2) verwendet werden. Die lokale Abbildung muss hier direkt am Frontend persistiert werden. Dafür würde sich eine lokale Datenbank, aber auch eine einfachere Struktur, wie z.B. eine XML-Datei, anbieten. In den meisten Fällen ist das Setting recht überschaubar und beinhaltet kaum mehr als 100 AALItems. Für die geringe Datenmenge stellt die Verwendung einer lokalen Datenbank einen viel zu großen Aufwand dar. Deswegen wird für die Implementierung eine XML-Datei verwendet. Das folgende Listing zeigt anhand eines Beispiels ein minimales Setting. Um eine leichtere Lesbarkeit zu erreichen, wurden die Positions- und Größenangaben weggelassen.

```
1. <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2. <AALItem id="0" name="system" type="system" template="system" posAndSize ="...">
3.   <AALItem id="5000" name="Wohnung" type="flat" template="mainFlat" posAndSize ="...">
4.     <AALItem id="300" name="Wohnzimmer" type="room" template="livingRoom" posAndSize ="...">
5.       <AALItem id="301" name="Tuer" type="sensor" template="door" posAndSize ="..."/>
6.       <AALItem id="302" name="Bewegungsmelder" type="locator" template="motion"
           posAndSize ="..."/>
7.       <AALItem id="304" name="Heizung" type="actuator" template="heater"
           posAndSize =..." adjustedValue="0"/>
8.       <AALItem id="305" name="Licht" type="actuator" template="light"
           posAndSize =..." adjustedValue="0" />
9.     </AALItem>
10.   <AALItem id="22" name="Alarm" type="sensor" template="alarm" posAndSize ="..."/>
11. </AALItem>
12. </AALItem>
```

Listing 1: Ein Beispiel für die Abbildung des Settings mit Hilfe einer XML-Datei

Die erforderliche Eltern-Kind-Beziehung, wird durch die XML-Struktur abgedeckt. Die Parameter Identifikation, Typus, Position und Größe werden durch die Attributknoten des jeweiligen XML-Elementes definiert. Der verbleibende Parameter Geometrie wird in diesem Listing durch das Attribut `templateName` abgedeckt und stellt eine Referenz auf ein bereits bestehendes Element dar. Templates werden im Abschnitt 4.7.3.2 genauer erläutert.

Obwohl die lokale Abbildung die Unabhängigkeit von einem bestimmten AAL-Backend bietet, müssen die daraus resultierenden Nachteile hier auch erwähnt werden. Bei dieser Methode wird die Abbildung des Settings dupliziert und eine Änderung erfordert die Korrektur sowohl am Backend als auch am Frontend. Dieses Argument wird jedoch nicht strak gewichtet, da eine Änderung des Settings in den meisten Fällen selten vorkommt und die doppelte Anlernung für gewöhnlich nur bei der Erstinstallation erfolgt.

4.7.3. Der Einsatz von Scalable Vector Graphics für die Abbildung des Settings

Die vorherigen zwei Abschnitte behandelten die Abbildung von einzelnen Setting-Objekten mit Hilfe von AALItems und die Persistierung mit Hilfe einer lokalen XML-Datei.

Zu diesem Zeitpunkt müssen noch folgende zwei Fragestellungen behandelt werden:

- Wie wird das Frontend angelernt, bzw. wie wird die XML-Datei erstmals erstellt?
- Wie kann die Geometrie eines AAL-Items beschrieben werden?

Beide Fragestellungen können durch die Verwendung des Scalable Vector Graphics (SVG)-Formats beantwortet werden. SVG ist ein vom W3C Konsortium entwickelter freier

Standard für eine Sprache, die zweidimensionale Vektorgrafiken und gemischte Vektor-/Rastergrafiken beschreiben kann (The World Wide Web Consortium (W3C), 2011) (Mahle, 2004, S.8-9). Somit kann eine SVG-Datei die Geometrie einzelner AALItems als eine zweidimensionale Vektorgrafik beschreiben und gestattet zudem die Erstellung der lokalen XML-Datei aus der XML-basierten Struktur des Fileformats.

4.7.3.1. Die Vorteile des SVG-Formats für die Abbildung des Settings

Das SVG-Format bietet im Hinblick auf die Beschreibung eines Settings folgende Vorteile:

- **XML-basierte Struktur**

Die Eltern-Kind-Beziehung kann direkt aus dem SVG-Format entnommen werden, da die hierarchische Struktur durch die Verschachtelung von einzelnen SVG-Elementen vorgegeben ist.

- **Automatische Ermittlung der Position und Größe**

Die Position und die Größe eines AALItems muss nicht explizit angegeben werden, sondern kann direkt aus den SVG-Elementen, dessen Attributen und durch das Vorwissen von Elternelementen berechnet werden.

- **Skalierungsunabhängige Vektorgrafik**

Wie bereits im Abschnitt 4.5.2 erwähnt, werden Rastergrafiken für die Darstellung von Symbolen benötigt. Da das SVG eine Vektorgrafik und dadurch skalierungsunabhängig ist, kann es jederzeit dazu verwendet werden, um Rastergrafiken in der jeweils benötigten Auflösung zu generieren.

- **Keine direkte Abhängigkeit vom AAL-Backend**

Das SVG-File kann vollkommen unabhängig vom Backend erstellt werden. Existierende SVG-Editoren wie z.B. Inkscape²⁸ ermöglichen sogar das computerunterstützte Zeichnen des Settings. Der offene Standard würde aber auch einen Export aus dem jeweiligen AAL-Backend erlauben. Dazu müsste allerdings das AAL-Backend dementsprechend erweitert werden.

4.7.3.2. Definition eines AALItems innerhalb der SVG-Datei

Durch die XML-basierte Struktur des SVG-Formats können die AALItems leicht als zusätzliche XML-Attribute innerhalb von SVG-Tags definiert werden. Listing 2 zeigt einen Ausschnitt eines SVG-Files, das zwei AALItems definiert. In der Zeile 4 wird ein neuer Namensraum aal definiert (Vergl. Becher, 2009, S.79ff). In den Zeilen 12 - 15 und 19 - 22 werden

²⁸ Inkscape: Open-Source-Vektorgrafikeditor. Siehe <http://inkscape.org>

dann zusätzliche AAL-Attribute in ein gewöhnliches SVG-Group-Element²⁹ eingefügt, die ein SVG-Element schließlich als ein AALItem kennzeichnen.

```
1. <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2. <!-- Created with Inkscape (http://www.inkscape.org/) -->
3. <svg
4.   xmlns:aal="http://www.sili.at/aal/"
5.   xmlns:dc="http://purl.org/dc/elements/1.1/"
6.   ...
7.   style="display:inline">
8.   ...
9.   <g
10.     inkscape:groupmode="layer"
11.     id="layer10"
12.     aal:id="300"
13.     aal:type="room"
14.     aal:name="Wohnzimmer"
15.     aal:templateName="livingRoom">
16.     ...</g>
17.   ...
18.   <g
19.     aal:id="1120"
20.     aal:type="aktuator"
21.     aal:name="Lichtschalter"
22.     aal:templateName="lightSwitch">
23.     ...</g>
24.   ...
25. </svg>
```

Listing 2: Definition eines AAL-Items innerhalb eines SVG-Group-Elements mit Hilfe von zusätzlichen Attributen des Namenraums aal.

Die vier Attribute `aal:id`, `aal:type`, `aal:name` und `aal:templateName` sind ausreichend, um ein AALItem, wie es im Abschnitt 4.7.1 definiert wurde, zu beschreiben. Die Größen- und Positionsangaben sowie die Eltern-Kind-Beziehungen müssen nicht explizit angegeben werden, sondern können aus den übrigen SVG-Attributen ermittelt werden. Das Attribut `aal:templateName` kennzeichnet, dass aus dieser 2D Vektorgrafik eine Rastergrafik erstellt werden muss, die im späteren Verlauf im Frontend angezeigt wird. Die Erstellung geschieht allerdings nur dann, wenn noch keine Rastergrafik mit dem gleichen Namen im Filesystem vorhanden ist. Wie der Name Template schon vermuten lässt, wird dieses Element nur einmal erstellt und mehrmals verwendet. Wie im Listing 2 zu sehen ist, definieren die Zeilen 19 bis 22 einen Lichtschalter mit dem TemplateNamen „lightSwitch“. Dieser TemplateName kann im SVG-File mehrmals verwendet werden. Das einmal erstellte lightSwitch-Template wird in diesem Fall auch im Frontend mehrmals verwendet. Die Verwendung von Templates hat einen weiteren Vorteil. So können die verschiedenen Einheiten, Aktionsnamen und Zustandsnamen durch ein Template definiert werden. Die Tabelle 4 verdeutlicht diese Vielfältigkeit an Einheiten und Namen anhand von Aktuatoren für fünf

²⁹ SVG-Group-Element: Siehe <http://www.w3.org/TR/SVG/struct.html#Groups>

verschiedene Einsatzgebiete. Diese Eigenschaften werden benötigt, um aussagekräftige Sätze über den Zustand eines Aktuators/Sensors zu generieren. Die Sätze wie „Das Wohnzimmerlicht ist ausgeschaltet. Wollen Sie es einschalten?“ oder „Die Raumtemperatur beträgt 22° C“ sind nur einige Beispiele dafür.

	Werte- bereich	Einheit	Minimum Name	Maximum Name	Erhöhen	Verringern
Lichtschalter	0..1	-	ausgeschaltet	eingeschaltet	einschalten	ausschalten
Lichtdimmer	0..100	%	aus	voll aufgedreht	heller stellen	dunkler stellen
Jalousie	0..100	%	unten	oben	rauf	runter
Temperatur	5..30	°C	5°C	30°C	erhöhen	verringern
Fenster	0..2	-	geschlossen	geöffnet	öffnen	schließen

Tabelle 4: Beispiele für verschiedene Aktionsnamen (Einschalten, Erhöhen, Öffnen usw.) und Zustandsnamen (offen, eingeschaltet, 27°C usw.) bei Aktuatoren für verschieden Einsatzgebiete.

4.7.4. Die SVG - Verarbeitungskette

Nachdem das SVG als das „Lernformat“ für das AAL-Frontend ausgewählt wurde, muss das Frontend auch so gestaltet werden, dass es mit dem SVG-Format umgehen kann. Folgende Schritte bilden die SVG-Verarbeitungskette des AAL-Frontends:

- AALItems aus dem SVG-File extrahieren**

Der Import einer SVG-Datei beginnt mit der Extraktion von AALItems. Dazu wird jedes Element des SVG-Files nach Attributen des Namenraums aal durchsucht. Die gefundenen AALItems werden intern als eine Baumstruktur abgebildet.
- Position und Größe ermitteln**

Die Ermittlung der Position und der Größe jedes einzelnen AALItem geschieht automatisch aus den SVG-Attributen. Dabei reicht es, wenn die Positionen und die Größe relativ zum Elternobjekt gespeichert werden, da mit Hilfe dieser Informationen eine vollständige Rekonstruktion möglich ist.
- Erstellung von Template-Rastergrafiken**

Falls im System noch keine Template-Rastergrafik mit dem angegebenen Namen existiert, dann wird aus dem SVG-Element eine solche Rastergrafik generiert.

- **Persistierung als ein internes XML-File**

Die im Speicher vorliegende Baumstruktur wird, wie im Abschnitt 4.7.2.2 beschrieben, persistiert. So kann sie unabhängig von der SVG-Datei beim Starten des Systems wieder geladen und intern als eine Baumstruktur abgebildet werden.

4.8. Sicherheit

Das AAL-Frontend ist aus der Sicht von sicherheitstechnischen Aspekten ein Zugang zu dem dahinter befindlichem AAL-Backend. Ein Eindringling könnte durch diesen Zugang Befehle an das Backend senden und so z.B. Aktuatoren ein- oder ausschalten. Aus diesem Grund ist es wichtig, dass die Kommunikation zwischen dem Frontend und dem Backend möglichst gegen Fremdeingriffe geschützt wird.

Für das hier zu entwickelnde AAL-Frontend wurde schon bei der Wahl des Webserver darauf geachtet, dass gewisse sicherheitstechnische Features abgedeckt werden. So unterstützt der im Frontend verwendete Mongoose-Webserver das Verschlüsselungsprotokoll Secure Sockets Layer (SSL), das die Kommunikation zwischen dem Frontend und dem Backend verschlüsseln und so gegen Lauschangriffe schützen kann. Auch die Verwendung von Access Control Lists (ACL) wird von Mongoose ermöglicht. Dadurch ist es möglich, das Frontend so zu konfigurieren, dass nur Anfragen von einer bestimmten IP-Adresse, z.B. von der des AAL-Backends, wahrgenommen werden. Alle anderen Anfragen werden ignoriert. Anstatt einer WLAN-Verbindung könnte auch nur die LAN-Verbindung verwendet werden, um auf diese Weise die Sicherheit zu erhöhen. In diesem Fall wäre das System nur durch eine bestehende Internetverbindung, oder durch ein physikalisches Netzkabel erreichbar. Zusammenfassend lässt sich sagen, dass eine Anbindung an das Internet immer ein gewisses Sicherheitsrisiko darstellt. Das System kann jedoch auch ohne eine Internetverbindung, als ein gekapseltes System, in einer LAN-Umgebung betrieben werden und dadurch einen unbefugten Fremdzugriff weitestgehend vermeiden.

4.9. Usability und Anwenderfreundlichkeit

Wie bereits im Abschnitt 2.2.2 erwähnt, nimmt die Anwenderfreundlichkeit eine essentielle Rolle in jedem einzelnen Designschritt des Softwaredesigns ein. Zusammengefasst wurden in diesem Kapitel folgende Designentscheidungen vorgenommen, die unter anderem auch eine Steigerung der Usability und der Anwenderfreundlichkeit bezwecken:

- **Bedienung mit Hilfe von Softkey- und Ziffernblock-Tasten**

Die Bedienung erfolgt nicht durch externe PC-Tastaturen und/oder Zusatzgeräte, sondern mit Hilfe der Fernbedienung und zudem mit immer den gleichen 14 Tasten.

- **Dynamische und fixe Hotkeys**

Einzelne Elemente wie z.B. Aktuatoren oder Sensoren eines Wohnraumes müssen auf der graphischen Benutzeroberfläche nicht umständlich durch einen „Mauszeiger“ angesteuert und ausgewählt werden, sondern sind direkt mit Hilfe der Ziffernblock-Tasten 0-9, den sogenannten Hotkeys, ansprechbar. Die dynamische Hotkey-Zuordnung ermöglicht eine Mehrfachausnutzung. Im Bedarfsfall besteht die Möglichkeit auch eine fixe Zuordnung vorzunehmen.

- **Level-of-details Darstellung mit fixen Bereichen**

Die hierarchische Darstellung mit mehreren level-of-details vermeidet die andernfalls notwendigen Scrollbalken. Die Darstellungsart schafft zudem genügend Raum für einen fixen Inhaltsbereich, in dem wichtige Meldungen immer an der gleichen Stelle visualisiert werden. Der ebenfalls fixierte Navigationsbereich im unteren Drittel des OSDs ermöglicht die rasche Auslegung der vier Steuertasten.

- **Meldungsausgabe mit Hilfe von Text + Icon sowie von TTS**

Die Ausgabe von Alarm- und Warnmeldungen kann sowohl visuell als auch akustisch erfolgen. Die visuelle Textausgabe wird mit repräsentativen Icons bzw. Symbolen untermalen. Der dargestellte Text kann im Bedarfsfall von der verwendeten TTS-Engine akustisch ausgegeben werden.

- **Verständliche Aktions- und Zustandsnamen**

Unverständliche Sensorwerte und Aktuatorzustände werden mit Zuhilfenahme von Templates auf verständliche Sätze abgebildet. Aus einer empfangenen Meldung: „device id=25 value=0;“ wird ein, für den Nutzer verständlicher, Satz: „Das Licht im Wohnzimmer ist ausgeschaltet.“

- **Exit Button**

Das angezeigte Frontend kann stets mit der blauen bzw. der 4'ten Softkey-Taste ausgeblendet werden.

- **Parametrisierung**

Die Parametrisierung wurde bereits am Anfang dieser Diplomarbeit erwähnt und sollte hier vollständigkeitshalber nochmals angeführt werden. Benutzerspezifische Anpassungen an die Schriftgröße, Schriftfarbe, Hintergrundfarbe, Lautstärke usw. ermöglichen sowohl die Lesbarkeit von visuellen als auch die Verständlichkeit von akustischen Meldungen zu steigern.

- **Time-To-Live Parameter für Meldungen mit einer niedrigen Priorität**

Meldungen mit einer niedrigeren Priorität besitzen einen Time-To-Live Parameter. Sie werden nach Ablauf einer bestimmten Zeitspanne automatisch, ohne jegliches Zutun seitens des Nutzers, wieder vom TV-Bildschirm entfernt.

5. Implementierung

Dieser Abschnitt behandelt die konkrete Implementierung der Software als ein VDR-Plugin, das sich in einer ReelBox einsetzen lässt und eine AAL-Funktionalität auf dem Fernsehbildschirm ermöglicht. Es werden der verwendete Model-Viewer-Controller-Ansatz, die verschiedenen Klassen, die verwendeten Bibliotheken sowie die Installation auf der ReelBox behandelt. Um eine Unterscheidung zwischen dem beschreibenden Text und den Eigennamen für Klassen, Klassenattributen und Aufzählungstypen zu ermöglichen, werden die Eigennamen kursiv geschrieben. Der hier verwendete Begriff **vdr4aal** bezieht sich im Folgenden auf das entwickelte VDR-Plugin, das ein TV-basiertes Frontend für ein AAL-Backend erlaubt.

5.1. Der Model-Viewer-Controller - Ansatz

Das vdr4aal-Plugin ist nach dem Model-Viewer-Controller (MVC) Designmuster gestaltet, bei dem die Softwarekomponenten in drei Bereiche (Model, Viewer und Controller) getrennt werden (Daniel Liebhart, 2007). Das Model bestimmt die Daten, die in der Software verarbeitet werden. Der Viewer (die Präsentationsschicht) bestimmt wie die Model-Daten dargestellt bzw. ausgegeben werden sollen. Der Controller (die Steuerung) ist für Änderungen an Model-Daten zuständig. Abbildung 12 verdeutlicht den MVC-Ansatz. In der klassischen Variante (linke Abbildung) existiert keine direkte Kommunikation zwischen dem Controller und dem Viewer. Eine Änderung am Modell wird an beiden Seiten nur indirekt durch sogenannte Observer (Beobachter) wahrgenommen. Die rechte Abbildung stellt eine Variante dar, die sehr oft in der Praxis verwendet wird und eine direkte Kommunikation zwischen dem Controller und dem Viewer erlaubt. Das Plugin verwendet diese Variante. Eine vom Benutzer ausgelöste Aktion, z.B. das Einschalten der Raumbelichtung, wird von der Präsentationsschicht direkt an den Controller geleitet und dort weiter verarbeitet. Die Kommunikation kann auch umgekehrt erfolgen. So werden interne Ereignisse, wie z.B. eine aufkommende, vom AAL-Backend ausgelöste Warnmeldung, vom Controller direkt an den Viewer geleitet und dort dem Benutzer präsentiert.

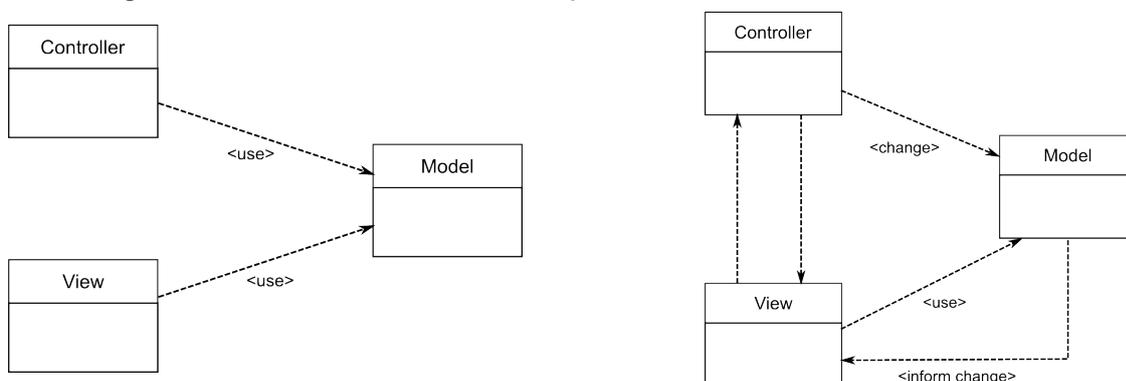


Abbildung 12: Links: Der klassische MVC-Ansatz. Rechts: Der MVC-Ansatz in der Praxis (aus Daniel Liebhart, 2007)

Die Tabelle 5 listet die einzelnen Klassen des MVC-Designpatterns im vdr4aal-Plugin auf. Das Modell wird von den Klassen *Template*, *Property* und *AALItem* gebildet. Die Präsentation erfolgt auf dem OSD durch die *Visualizer*-Klasse bzw. den darin enthaltenen *Pane*-Klassen. Die Controller-Schicht wird von der *Controller*-Klasse und 8 *Factory*-Klassen gebildet. Der Controller ist dabei eigentlich nur ein Dispatcher (Verteiler), der alle ankommenden Befehle bzw. Aufgaben an die 8 *Factory*-Klassen leitet. Je nach Art der Aufgabe fühlen sich eine oder mehrere *Factory*-Klassen angesprochen und verarbeitet/-en die Aufgabe.

Modell	Viewer	Controller
<i>AALItem</i>	<i>Visualizer</i>	<i>Controller</i>
<i>AALSystemItem</i>	<i>DrawPane</i>	<i>SystemFactory</i>
<i>AALLocatorItem</i>	<i>ControlPane</i>	<i>PropertyFactory</i>
<i>AALSensorItem</i>		<i>AudioFactory</i>
<i>AALActorItem</i>		<i>TemplateFactory</i>
<i>Template</i>		<i>VisFactory</i>
<i>Property</i>		<i>LogFactory</i>
		<i>HttpServer</i>
		<i>HttpClient</i>

Tabelle 5: Einteilung der Klassen und der HTML-Webseite in die Model-Viewer-Controller Designpatterns

5.2. Die Kommunikation mit dem AAL-Backend

Wie bereits im Abschnitt 4.2 erwähnt, benötigt das vdr4aal-Plugin sowohl einen HTTP-Webserver als auch einen HTTP-Client. Mit Hilfe des HTTP-Servers kann ein Backend aktuelle Sensor- und Aktuator-Werte sowie Alarmer und Warnungen an das Frontend senden. Umgekehrt wird der HTTP-Client verwendet wenn das Frontend Nachrichten an das Backend schicken soll. Das kann eine Anweisung sein, die z.B. einen Aktuator einschaltet. In beiden Fällen wird eine HTTP-GET-Anfrage für den Nachrichtenaustausch verwendet.

5.2.1. Frontend zu Backend Kommunikation

Der HTTP-Client des Frontends sendet folgende HTTP-GET-Anfrage an das eingesetzte AAL-Backend um einen Aktuator zu steuern:

```
1. http://<host>:<port>/vdr?action=setValue&itemId=<id>&value=<value>
```

Die Strings in den spitzen Klammern repräsentieren Parameter, die folgende Werte annehmen können:

- **host**
Der DNS-Name oder die IP-Adresse, unter der das AAL-Backend erreichbar ist. Der Parameter ist in der `property.xml`-Datei, die sich im Konfigurationsverzeichnis befindet, einstellbar.
- **port**
Die Portnummer, unter der das AAL-Backend Anfragen entgegennimmt. Der Parameter ist in der `property.xml`-Datei, die sich im Konfigurationsverzeichnis befindet, einstellbar.
- **id**
Die AALItem-Id eines Aktuators wie sie im SVG-File beim Import definiert wurde. Die Identifikationsnummer wird automatisch in die HTTP-GET-Anfrage eingefügt und entspricht der AALItem-Id des Aktuators, der aktuell im Frontend angezeigt wird.
- **value**
Die AALItem-adjustedValue eines Aktuators. Der im Frontend eingestellte Wert für einen Aktuator. Dieser Wert wird ebenfalls automatisch in die HTTP-GET-Anfrage eingefügt.

5.2.2. Backend zu Frontend Kommunikation

Der HTTP-Server am AAL-Frontend erwartet sich folgende Anfrage vom eingesetztem AAL-Backend:

```
1. http://<host>:8081/doAction?Action=<type>&message=<message>&itemId=<id>
```

Die Strings in den spitzen Klammern repräsentieren Parameter, die folgende Werte annehmen können:

- **host**
Der DNS-Namen oder die IP-Adresse der ReelBox.
- **type**
Bestimmt welche Aktion am Frontend ausgeführt werden soll. Folgende Werte sind möglich: *info*, *show*, *warn*, *alarm* und *changeValue*.
- **message**
Bei Aktionen vom Typ *info*, *show*, *warn* und *alarm* wird ein Message-String wie z.B. „Die Eingangstuer ist offen!“ erwartet. Bei Aktion vom Typ *changeValue* wird ein Integerwert erwartet, der dem *currentValue*-Parameter eines *AALLocatorItems*, *AALSensorItems* oder *AALActorItems* entspricht.
- **itemId**
Die AALItem-Id eines AALItems, wie sie im SVG-File beim Import definiert wurde.

5.3. Klassendiagramm

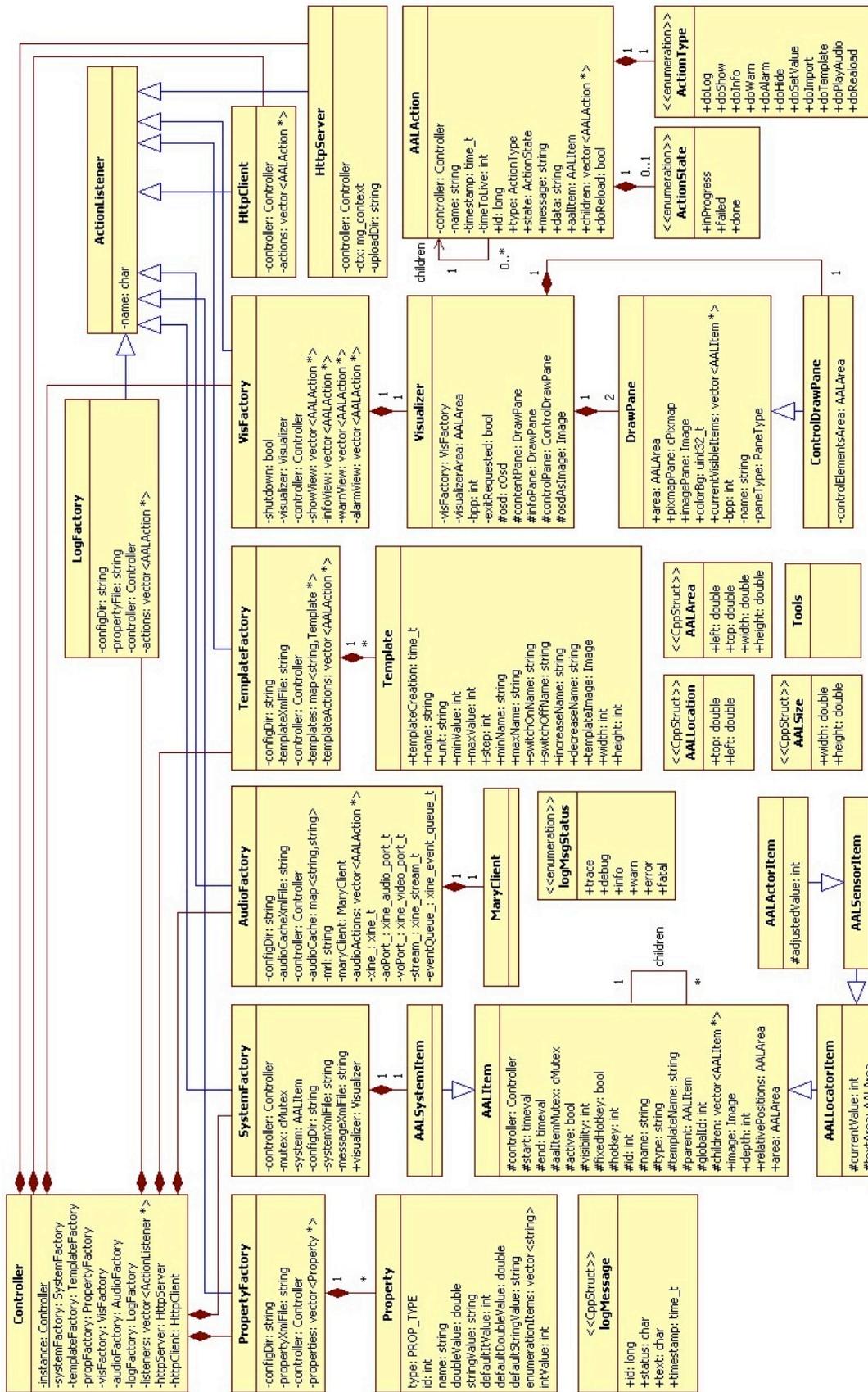


Abbildung 13: Klassendiagramm des vdr4aal-Plugins

5.4. Klassenbeschreibung

Der folgende Abschnitt beschreibt die wichtigsten Attribute, Funktionen und Eigenschaften der einzelnen Klassen, die das vdr4aal-Plugin bilden.

cPluginAalSystem

Die *cPluginAalSystem*-Klasse erweitert die VDR *cPlugin*-Klasse und ist somit die Startklasse des Plugins, welches ein AAL-Frontend im VDR-Programm erlaubt. In dieser Klasse geschieht recht wenig. Die Klasse instanziiert einen *Controller* und leitet alle weiteren Tätigkeiten an diese *Controller*-Instanz weiter.

Controller

Die *Controller*-Klasse bildet mit den 8 *Factory*-Klassen die Controller-Schicht des MVC-Designmusters. Beim Start des Systems werden hier die *Factory*-Klassen instanziiert und in die interne Liste von *ActionListeners* eingefügt. Umgekehrt enthält auch jede *Factory*-Instanz eine Referenz der *Controller*-Instanz. Dadurch wird es den *Factory*-Klassen ermöglicht, eine *AALAction* an den *Controller* und infolge an alle anderen *Factory*-Klassen zu senden. Der *Controller* fungiert - wie bereits erwähnt - als ein Dispatcher, der die *AALAction* empfängt und an alle Mitglieder der internen *ActionListeners*-Liste weiterleitet. Die *Controller*-Klasse bietet auch direkten Zugang zu einigen *Factory*-Klassen wie z.B. *PropertyFactory*. Dies ermöglicht anderen *Factory*-Klassen *Property*-Instanzen zu empfangen.

ActionListener

Die *ActionListener*-Klasse ist eigentlich ein reines Interface, das es anderen Klassen ermöglicht, auf ankommende *AALActions* zu reagieren. Alle 8 *Factory*-Klassen implementieren dieses Interface und ermöglichen somit dem *Controller* *AALActions* zu verteilen. Je nach Art der Aktion fühlen sich eine oder mehrere *Factory*-Klasse angesprochen und verarbeitet/-en intern die Aktion.

ActionType

Der Aufzählungstyp *ActionType* definiert die möglichen Typen von Aktionen im vdr4aal-Plugin. Folgende Aktionstypen werden unterstützt: *doLog*, *doShow*, *doInfo*, *doWarn*, *doAlarm*, *doHide*, *doSetValue*, *doImport*, *doTemplate*, *doPlayAudio*, *doReload*.

ActionState

Der Aufzählungstyp *ActionState* definiert die möglichen Status von Aktionen. Aktionen können folgenden Status aufnehmen: *inProgress*, *failed*, *done*.

AALAction

Die *AALAction*-Klasse kapselt eine auszuführende Aktion vom Typ *ActionType* innerhalb des *vdr4aal*-Plugins. Je nach Zustand des Systems wird eine Aktion entweder sofort ausgeführt, oder in eine Verarbeitungsschlange einer *Factory*-Klasse eingereiht und zu einem späteren Zeitpunkt abgearbeitet. Aktionen besitzen grundsätzlich einen Status vom Typ *ActionState*. So kann dem Aufrufer signalisiert werden, ob die Aktion gerade ausgeführt wird, bereits gescheitert ist, oder erfolgreich ausgeführt wurde. Dies ist notwendig, da Aktionen nicht immer synchron, sondern manchmal in eigenen Threads abgearbeitet werden. Aktionen besitzen einen Timer, der darüber entscheidet wie lange die Aktion noch gültig ist. So können Benachrichtigungen ausgegeben werden, die nur eine bestimmte Zeit am TV-Gerät sichtbar sind und danach automatisch ausgeblendet werden. Aktionen sind immer an ein *AALItem* gebunden. Auf diese Weise wird auch gleichzeitig angegeben auf welches *AALItem* sich eine Aktion bezieht. Sollte kein *AALItem* vorhanden sein, dann wird automatisch das Root-Element, also das *SystemAALItem*, verwendet.

Property

Die *Property*-Klasse definiert Benutzereigenschaften wie z.B. die zu verwendete Schriftfarbe als ein Schlüssel-Werte-Paar im *vdr4aal*-Plugin. Der hier definierte Aufzählungstyp *PROP_TYPE* listet die möglichen Stereotypen auf, die eine Benutzereigenschaft annehmen kann. Folgende Stereotypen werden unterstützt: *INTEGER*, *STRING*, *DOUBLE*, *ENUMARATION*, *ARGBCOLOR*. Eine Benutzereigenschaft hat einen Standardwert der verwendet wird, falls kein anderer Wert vom Designer definiert wurde.

PropertyFactory

Die *PropertyFactory*-Klasse persistiert, lädt und stellt *Property*-Instanzen für andere Klassen zur Verfügung. *Property*-Instanzen werden beim Herunterfahren des Systems mit Hilfe des Apache Xerces XML-Parsers als eine *property.xml*-Datei persistiert und beim Starten des Systems wieder rekonstruiert.

AALItem

Die *AALItem*-Klasse ist das zentrale Element für die Abbildung eines Settings. Wie bereits im Abschnitt 4.7.1 beschrieben, stellt ein *AALItem* ein auswählbares und am Frontend visualisierbares Objekt des Settings dar und kann z.B. ein Zimmer, ein einzelner Sensor oder eine ganze Wohnung bzw. auch ein Haus sein. Ein *AALItem* hat für gewöhnlich ein *AALItem*-Elternelement und keine, eine oder mehrere *AALItem*-Kinderelemente. Dies ermöglicht eine hierarchische Abbildung des Settings wie z.B. „Wohnung-Raum1-Sensor1“ oder „Wohnung-Raum2-Sensor13“. Ein *AALItem* speichert seine Position innerhalb des Settings nicht absolut, sondern relativ zu seinem Elternelement. Dies erlaubt eine dynami-

sche Re-Skalierung, sodass das aktuell anzeigende *AALItem* immer mit der möglichst größten Auflösung am TV-Bildschirm zu sehen ist. Da das *AALItem* ein auswählbares Objekt ist, besitzt es während der Anzeigedauer einen Hotkey, das als eine eingekreiste Nummer am Frontend visualisiert wird. Wie bereits beschrieben, existieren fixe und dynamische Hotkeys. Das *AALItem* kann durch die Eltern-Kind-Beziehung und einen rekursiven Aufruf automatisch die dynamischen Hotkeys für die aktuell anzeigenden Elemente generieren.

SystemAALItem

Die *SystemAALItem*-Klasse ist eine Spezialisierung der *AALItem*-Klasse. Sie hat keine zusätzlichen Attribute, sondern bestimmt lediglich durch den Klassennamen das Root-Element des gesamten Systems, das kein Elternelement hat. Da dieses *AALItem* das Root-Element ist, kann es dazu verwendet werden, ein beliebiges Kind-Element im System zu erhalten. Das *SystemAALItem* wird auch bei *AALActions* verwendet, die kein spezielles *AALItem* besitzen wie z.B. bei einer *AALAction* vom Typ *doLog*.

AALLocatorItem

Die *AALLocatorItem*-Klasse ist eine Spezialisierung der *AALItem*-Klasse mit den Zusatzattributen *currentValue* und *textArea*. Dieses spezielle *AALItem* kann im Frontend nicht ausgewählt werden. Es entstand aus der Anforderung, einzelne Bewegungsmelder innerhalb eines Raumes anzuzeigen, die nicht ausgewählt werden dürfen. Bewegungsmelder sind anders als Sensoren und Aktuatoren nur für kurze Zeit aktiv, z.B. während einer Bewegung. Eine gezielte Auswahl des Bewegungsmelders im Frontend wäre unpraktisch, weil die Bewegungsdetektion auch direkt in der Visualisierung des Raumes dargestellt werden kann. Da das *AALLocatorItem* das erste *AALItem* ist, das einen Wert annehmen kann (Bewegung vorhanden oder nicht) besitzt es auch das Attribut *currentValue*.

AALSensorItem

Die *AALSensorItem*-Klasse ist eine Spezialisierung der *AALLocatorItem*-Klasse mit keinen weiteren Attributen. Im Unterschied zu einem *AALLocatorItem* kann ein *AALSensorItem* im Frontend ausgewählt und der aktuelle *currentValue*-Wert betrachtet werden. Werte eines Sensors können im Frontend nicht geändert werden. Sie geben nur Auskunft über einen Zustand, z.B. dass die Eingangstür nicht verschlossen ist. Der Nutzer muss den Zustand jedoch selbst ändern.

AALActorItem

Die *AALActorItem*-Klasse ist eine Spezialisierung der *AALSensorItem*-Klasse mit dem Zusatzattribut *adjustedValue*, das einen vom Benutzer voreingestellten Wert darstellt. Im Gegensatz zu einem *AALSensorItem* kann ein *AALActorItem* „Wunschwerte“ beinhalten. Ein Beispiel dafür wäre die Raumtemperatur mit einem voreingestellten Wert (*adjustedValue*) von 22°C und einem aktuellen Wert von (*currentValue*) 18°C. Eine Heizung ist dann so lange aktiv bis sich die zwei Werte angeglichen haben.

SystemFactory

Die *SystemFactory*-Klasse persistiert, lädt, importiert und stellt die *AALSystemItem*-Instanz und dadurch alle *AALItem*-Kind-Elemente zur Verfügung. Die *AALSystemItem*-Instanz wird beim Herunterfahren des Systems mit Hilfe des Apache Xerces XML-Parsers als eine *system.xml*-Datei persistiert und beim Starten des Systems wieder rekonstruiert. Der Import geschieht mit Hilfe des *ActionListeners*, der auf ankommende *AALActions* vom Typ *doImport* hört. Nachdem so eine *AALAction* erfasst wurde, sendet die *SystemFactory*-Klasse eine *doHide* Action aus, die das Frontend dazu veranlasst, alle aktuell anzeigenden *AALItems* zu verbergen. Dies ist notwendig, da der Import auch eine Änderung in der aktuellen Anzeige mit sich führen könnte. Das zu importierende SVG-File wird ebenfalls mit Hilfe des Apache Xerces XML-Parsers geparkt und der daraus resultierende *AALItem*-Baum wird in die *AALSystemItem*-Instanz als ein Kind-Element eingefügt.

Template

Die *Template*-Klasse definiert einerseits die anzuzeigende Rastergrafik und ist andererseits ein Behälter für die Einheit, die Wertebereiche, die Aktions- und die Zustandsnamen von *AALItems*. Die Klasse speichert zudem eine *Magick-Image* Instanz der Rastergrafik, die im Frontend gerändert wird.

TemplateFactory

Die *TemplateFactory*-Klasse persistiert, lädt, generiert und stellt *Template*-Instanzen für das Rendering des Frontends zur Verfügung. *Template*-Instanzen werden beim Herunterfahren des Systems mit Hilfe des Apache Xerces XML-Parsers als eine *templates.xml*-Datei persistiert und beim Starten wieder rekonstruiert. Die Erzeugung von *Templates* geschieht mit Hilfe des *ActionListeners*, der auf ankommende *AALActions* vom Typ *doTemplate* hört. Die so erfassten *AALActions* werden in eine interne *templateActions*-Liste zwischengespeichert und in einem späteren Verarbeitungsschritt in einem separaten Thread abgearbeitet. Dies ist notwendig, weil die Generierung von *Template*-Rastergrafiken eine gewisse Zeit in Anspruch nimmt. Eine sofortige Verarbeitung im Mainthread würde das VDR-Programm zu stark beeinträchtigen und die Reaktionszeit z.B. beim Wechseln des

Fernsehkanales deutlich herabsetzen. Eine *doTemplate*-Aktion beinhaltet den Namen eines SVG-Files, das sich bereits auf der Festplatte der ReelBox befindet. Dieses SVG-File wird mit Hilfe der Inkscape-Applikation geparkt und aus jedem SVG-Element mit einer AAL-Kennung wird eine eigene PNG-Bilddatei generiert. Die PNG-Bilddatei wird im Template-Verzeichnis des vdr4aal-Plugins abgelegt. Gleichzeitig wird auch eine *Template*-Instanz erstellt und in der internen *templates*-Mappe abgelegt. Falls ein *Template* mit dem gleichen Namen in der *templates*-Mappe existiert, dann wird diese von der neuen Template-Instanz überschrieben.

AudioFactory

Die *AudioFactory*-Klasse persistiert, lädt, verwaltet, generiert und gibt Audiodateien aus. Wie bereits im Abschnitt 4.6 erläutert, verwendet das System WAV-Dateien, die von der TTS-Engine generiert werden. Eine WAV-Datei ist dabei die akustische Repräsentation eines Message-Strings und wird im Filesystem unter einem Dateinamen gespeichert, der zufällig generiert wurde. Die *AudioFactory*-Klasse verwaltet in einer internen Mappe die Information welcher Message-String zu welcher im Filesystem abgelegten Audiodatei gehört. Diese Mappe wird beim Herunterfahren des Systems mit Hilfe des Apache Xerces XML-Parsers als eine *audioCache.xml*-Datei persistiert und beim Starten des Systems wieder rekonstruiert. Je nach eingesetzter TTS-Engine ist die Klasse auch für die Generierung von Audiodateien zuständig. Die Wiedergabe bzw. die Erzeugung der Audiodatei startet mit Hilfe des *ActionListeners*, der auf ankommende *AALActions* vom Typ *doPlayAudio* hört. Diese Action wird in die interne *audioActions*-Liste eingefügt und in einem separaten Thread abgearbeitet. Dies ist deswegen notwendig, da die Generierung von Audiodateien eine gewisse Zeit in Anspruch nimmt. Falls für die auszugebende Message noch keine Audiodatei existiert, dann wird diese zuerst mit der ausgewählten TTS-Engine erzeugt und im System abgelegt, bevor sie ausgegeben wird. Im anderen Fall wird sie direkt ausgegeben. Für die Ausgabe verwendet das System die xine-Multimedia Library des Ubuntu-Systems.

MaryClient

Die *MaryClient*-Klasse ermöglicht die Kommunikation mit der Mary TTS-Engine, die als ein Server entweder direkt am lokalen Rechner, oder an einem anderen Rechner im lokalen Netzwerk ausgeführt wird. Die Angabe welcher Server und welcher Port verwendet werden soll erfolgt über den Parameter der *maryQuery*-Methode. Die *AudioFactory*-Instanz instanziiert die *MaryClient*-Klasse nur dann, wenn die Mary als TTS-Engine verwendet werden soll. Dies kann in der *property.xml*-Datei unter dem key *audioType* eingestellt werden.

LogFactory

Die *LogFactory*-Klasse ist indirekt für das Generieren von Logdateien zuständig. Logeinträge werden durchgehend im ganzen System erstellt und helfen dem Entwickler bei der Rekonstruktion von ungewünschten Situationen wie z.B. einem Abbruch des Programms. Für das Logging wird die Apache log4cpp Bibliothek (Apache Software Foundation, 2008) verwendet. Die *LogFactory*-Klasse erstellt dabei die log4cpp - Konfigurationsdatei mit dem Namen `log4cpp.properties` im Konfigurationsverzeichnis des `vdr4aal`-Plugins. Diese Datei definiert einen `log4cpp-Fileappender`³⁰ und veranlasst, dass jeder Logeintrag im Filesystem unter der Datei `vdr4aal/logs/aalsystem.log` gespeichert wird.

AALLocation, AALSize und AALArea

Die drei Klassen definieren eine Position, eine Größe und einen Bereich eines Objektes innerhalb des `vdr4aal`-Plugins. Sie werden von verschiedenen anderen Klassen wie z.B. der *AALItem*-Klasse verwendet.

Tools

Die *Tools*-Klasse stellt vorwiegend allgemeine statische Methoden für Extraktion und Wertekonvertierungen zur Verfügung und wird von den anderen Klassen im `vdr4aal`-Plugin verwendet.

HttpClient

Die *HttpClient*-Klasse ist für das Senden von Nachrichten an das AAL-Backend zuständig. Das Format der Nachricht und die Parameter wurden im Abschnitt 5.2.1 erläutert.

HttpServer

Die *HttpServer*-Klasse empfängt und verarbeitet Nachrichten vom AAL-Backend. Das Format der erwarteten Nachricht vom AAL-Backend und die Parameter wurden im Abschnitt 0 erläutert.

VisFactory

Die *VisFactory*-Klasse instanziiert und verwaltet die *Visualizer*-Instanz, die das OSD darstellt. Sie ist zudem für das im Abschnitt 4.3.2 beschriebene Regelwerk zuständig. Hier werden die einzelnen Listen für die *showView*-, *infoView*-, *warnView*- und *alarmView*-Meldungen verwaltet. Ein Aufruf der Methode *getNextAction()* führt dazu, dass zuerst alle Alarmnachrichten, gefolgt von allen Warnmeldungen, gefolgt von der Informationsmeldung

³⁰ FileAppender: Ein Appender definiert eine Destination, also ein Ziel, das einen Logeintrag speichern bzw. ausgeben kann. Der FileAppender ist in diesem Zusammenhang eine im Filesystem definierte Datei, die Logeinträge speichert.

und schließlich die Anzeigemeldungen der Home-Control-Funktion retourniert werden. Die Klasse ist ebenfalls für das Entfernen von nicht mehr benötigten Anzeigen, die entweder durch einen Timeout, oder durch andere Benutzereingaben als ungültig markiert wurden. Die *VisFactory*-Klasse implementiert ebenfalls den *ActionListeners* und wird bei Aktionen vom folgenden Typ aktiv: *doInfo*, *doShow*, *doWarn*, *doAlarm*, *doHide* und *doReload*.

Visualizer

Die *Visualizer*-Klasse stellt das OSD des VDR-Plugins dar. Die Klasse definiert die drei im Abschnitt 4.5.1 beschriebenen Bereiche. Den Übersichtsbereich (*contentPane*), den Inhaltsbereich (*infoPane*) und den Steuerbereich (*controlPane*). Der Visualizer bietet diese Bereiche mit Hilfe von Getter-Methoden für die zu rendernden *AALItems* an. So weiß z.B. ein *AALSensorItem*, dass es sich im Übersichtsbereich nur als ein Symbol innerhalb eines Raumes darstellen soll und im Informationsbereich als eine Textzeile, die den aktuellen Wert darstellt (Vergl. Abbildung 10 d und 10 e). Nachdem alle Bereiche der *Visualizer*-Klasse aktualisiert wurden, wird eine *Graphickmagick-Image*-Instanz aus den Bereichen erstellt, quantisiert und schließlich als eine *cBitmap*-Instanz dem OSD-Instanz für die Visualisierung übergeben. Der Umweg über eine *Graphickmagick-Image*-Instanz ist notwendig, da die *ReelBox* nur 8bpp³¹ für *cBitmaps* erlaubt. So muss die Anzahl der Farben auf 256 herunterquantisiert werden, damit keine Fragmente am OSD sichtbar werden.

DrawPane

Die *DrawPane*-Klasse wird in der *Visualizer*-Instanz für den Übersichtsbereich und den Informationsbereich verwendet. Sie besitzt die *DrawImage(...)*, *DrawText(...)* und *DrawHotkey(...)* Methoden, die eine vollständige Visualisierung von *AALItems* erlauben.

ControlPane

Die *ControlPane*-Klasse wird in der *Visualizer*-Instanz für den Steuerbereich verwendet. Sie ist eine Spezialisierung der *DrawPane*-Klasse und stellt die Methode *drawControl(...)* zur Verfügung, die das Rendern der vier Steuerbereiche, wie sie im Abschnitt 4.4.2 vorgestellt wurden, erlaubt.

³¹ 8bpp: Abkürzung für 8 bits per pixel und definiert die Farbtiefe mit $2^8 = 256$ Farbwerten

5.5. Verwendete Bibliotheken und Applikationen

In der bisherigen Ausarbeitung wurden schon einige Programmbibliotheken oder Applikationen vorgestellt, die für den Betrieb des vdr4aal-Plugins notwendig sind. Um eine vollständige Übersicht bereitzustellen, werden hier alle verwendeten Libraries und Applikationen aufgelistet und ihre Aufgabengebiete näher erläutert.

5.5.1. Apache log4cxx

Die Apache log4cxx Bibliothek wird - wie bereits erwähnt - durchgehend in jeder Factory-Klasse verwendet und ist für die Erstellung von Logeinträgen zuständig. Die Bibliothek lässt sich mit dem folgenden Befehl in der Kommandozeile der ReelBox installieren.

```
1. reel@reelBox: ~$ sudo apt-get install liblog4cxx10-dev
```

5.5.2. GraphicsMagick

GraphicsMagick (GraphicsMagick Group, 2011) ist ein Bildverarbeitungssystem, das auf der ReelBox schon vorinstalliert ist. Sollt dies wegen eines neuen Betriebssystems für die ReelBox nicht der Fall sein, dann lässt sich die Library mit dem folgenden Befehl in der Kommandozeile der ReelBox nachinstallieren.

```
1. reel@reelBox: ~$ sudo apt-get install libgraphicsmagick++1-dev
```

GraphicsMagick wird einerseits für das Laden und die Skalierung von Template-Bilddateien verwendet und andererseits für die interne Generierung einer Bildinstanz aus dem OSD (Vergl. *Visualizer*-Klasse).

5.5.3. Apache Xerces XML-Parser

Der Apache Xerces XML-Parser (The Apache Software Foundation, 2011) wird für die Persistierung und die Rekonstruktion von DOM-Bäumen verwendet, die im vdr4aal-Plugin dann als eine verkettete Liste oder eine Mappe von C++ Instanzen abgebildet werden. Jede einzelne Property-Datei, die sich im Konfigurationsverzeichnis des Plugins befindet, wurde mit Hilfe des Xerces Parsers erstellt. Der Parser lässt sich mit dem folgenden Befehl in der Kommandozeile der ReelBox installieren.

```
2. reel@reelBox: ~$ sudo apt-get install libxerces-c2-dev
```

5.5.4. Inkscape Applikation

Inkscape ist ein Open-Source-Vektorgrafikeditor (Inkscape Community, 2011), der beim Import von SVG-Grafiken verwendet wird. Die Applikation wird im Hintergrund aufgerufen und ermöglicht den Export von einzelnen SVG-Elementen mit einer *AALItem*-Kennung. Auch die Berechnung der Größe und der Position des SVG-Elementes erfolgt mit Hilfe der Inkscape Applikation. Inkscape lässt sich mit dem folgenden Befehl in der Kommandozeile der ReelBox installieren.

```
1. reel@reelBox: ~$ sudo apt-get install inkscape
```

5.5.5. Mongoose Webserver

Der Mongoose Webserver (Lyubka, 2011) wurde bereits im Abschnitt 4.2.2 näher erläutert. Der Webserver bedarf keiner externen Installation, da der Quellcode im *vdr4aal*-Plugin inkludiert ist und automatisch mitübersetzt (kompiliert) wird.

5.5.6. eSpeak und MARY

eSpeak (Jonathan Duddington, 2011) ist eine Open-Source-Software für Sprachsynthese. Die Installation auf einem Ubuntu-basierten Betriebssystem ist relativ unkompliziert und kann den Installationsinstruktionen der Webseite des Projektes entnommen werden.

MARY ist ebenfalls ein TTS-System (Deutsches Forschungszentrum für Künstliche Intelligenz, 2010) das unter der MIT-Lizenz zum Download bereitgestellt wird. Anders als bei der eSpeak-Software sind die Installationsdateien über 1GB groß. Auch hier sollten die genaueren Installationsschritte der Webseite des Projektes entnommen werden.

Sowohl die eSpeak-Software als auch die MARY-Plattform werden von der *AudioFactory*-Klasse für die Generierung von TTS-WAV-Dateien verwendet.

5.5.7. Libcurl

Libcurl ist eine Programmbibliothek, die verschiedenste Protokolle wie HTTP, POP3, FTP usw. beherrscht (cURL Community, 2011). Die Bibliothek wird von der *HttpClient*-Klasse verwendet, um Anfragen über die HTTP-GET-Methode an das AAL-Backend zu senden. Libcurl lässt sich mit dem folgenden Befehl in der Kommandozeile der ReelBox installieren.

```
1. reel@reelBox: ~$ sudo apt-get install libcurl4-openssl-dev
```

5.6. Quellcode und die Installation auf der ReelBox

Nachdem die benötigten Bibliotheken und Applikationen auf der ReelBox installiert wurden, kann das vdr4aal-Plugin ebenfalls übersetzt und installiert werden. Der folgende Abschnitt beschreibt die Einzelschritte, die durchgeführt werden müssen, um das Plugin auf der ReelBox zu installieren.

5.6.1. Quellcode

Der Quellcode des vdr4aal-Plugin kann entweder der CD, die dieser Diplomarbeit beigelegt ist, entnommen werden oder online aus dem vdr4aal - Google Code Projekt geladen werden. Die Webadresse des Projektes lautet <http://code.google.com/p/vdr4aal/>. Die nachfolgende Installationsanleitung verwendet die Online-Ressourcen, die mit Hilfe des Versionsverwaltungsprogramms SVN ausgecheckt werden kann.

5.6.2. Installation auf der ReelBox

Um das Plugin übersetzen zu können, werden zuerst die Sourcecode-Dateien des ReelBox-VDR-Programms, sowie die passenden Entwicklerpakete benötigt. Die Entwicklerpakete können mit dem folgenden Befehl in der Kommandozeile der ReelBox installiert werden.

```
1. reel@reelBox: ~$ sudo aptitude install reelbox-devel
```

Danach kann der VDR-Sourcecode mit Hilfe von SVN ausgecheckt werden. Zuvor sollte in ein entsprechendes Verzeichnis z.B. /usr/src gewechselt werden.

```
1. reel@reelBox: ~$ cd /usr/src
2. reel@reelBox: usr/src$ svn co svn://reelbox.org testing
```

Nach dem checkout des VDR-Programmes kann das vdr4aal-Plugin ausgecheckt werden. Zuvor muss in das vdr-plugins Verzeichnis des VDR-Programms gewechselt werden.

```
1. reel@reelBox: ~$ cd reelbox.org/testing/src/vdr-plugins/src
2. reel@reelBox: usr/src/reelbox.org/testing/src/vdr-plugins/src$ svn co http://vdr4aal.googlecode.com/svn/trunk/plugin vdr4aal
```

Gegebenenfalls muss das Makefile des soeben ausgecheckten Plugins angepasst werden. Vor allem der VDRDIR-Eintrag und die APIVERSION müssen angepasst werden.

VDRDIR muss auf das Verzeichnis des VDR-Programms zeigen. Der APIVERSION-Eintrag muss mit der Versionsnummer des ausgecheckten VDR-Programms übereinstimmen.

1. VDRDIR = /usr/src/reelbox.org/testing/src/vdr-1.7
2. APIVERSION = "1.7.18.5"

Nach der Modifikation kann in das vdr4aal Verzeichnis gewechselt und die Übersetzung mit dem folgenden Befehl gestartet werden:

1. reel@reelBox: usr/src/reelbox.org/testing/src/vdr-plugins/src/vdr4aal \$ make all

Die resultierende Shared Object Bibliothek befindet sich anschließend im Verzeichnis /usr/lib/vdr und trägt den Namen libvdr-vdr4aal.so.1.7.18.5. Die hintere Versionsnummer entspricht dem Wert, der zuvor in der APIVERSION eingetragen wurde.

Um das Plugin beim Starten der ReelBox automatisch mitzuladen, bedarf es noch einer Anpassung. Im Verzeichnis /etc/reel/ muss eine Datei mit dem Namen p.vdr4aal.conf und dem folgenden Inhalt erstellt werden.

1. #
2. # configuration file for vdr4aal plugin
3. #
4. LOAD_PLUGIN=true
5. PLUGIN_OPTIONS="-Pvdr4aal"

Nach dem neuerlichen Starten der ReelBox sollte das AALSystem-Plugin verfügbar sein und mit dem Befehl aufgelistet werden.

1. reel@reelBox: ~\$ vdr -V
vdr (1.7.18.5/1.7.18.5) - The Video Disk Recorder
vdr4aal (1.0.0) - VDR plugin 4 AAL frontend
...

5.7. vdr4aal-Plugin im Einsatz

Die nachfolgenden Bilder zeigen das vdr4aal-Plugin im Betrieb auf der Reelbox.



Abbildung 14: vdr4aal-Plugin im Einsatz auf der ReelBox. Visualisierung der Home-Control-Funktion. Unter dem TV-Gerät ist die ReelBox erkennbar.

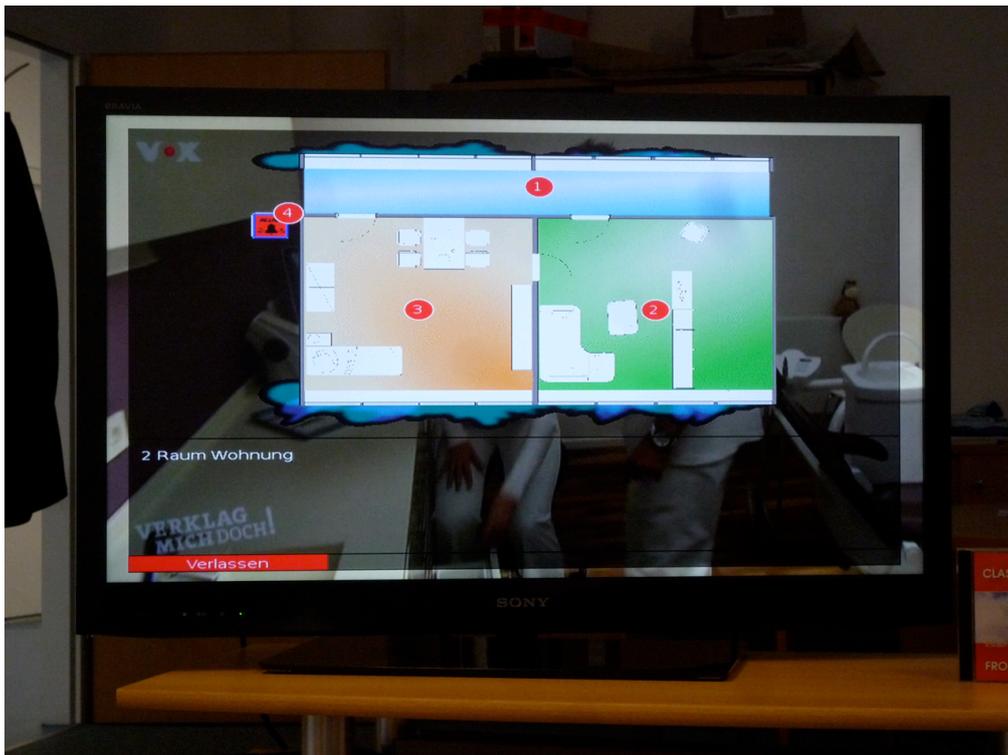


Abbildung 15: Ebenfalls eine Darstellung der Home-Control-Funktion.

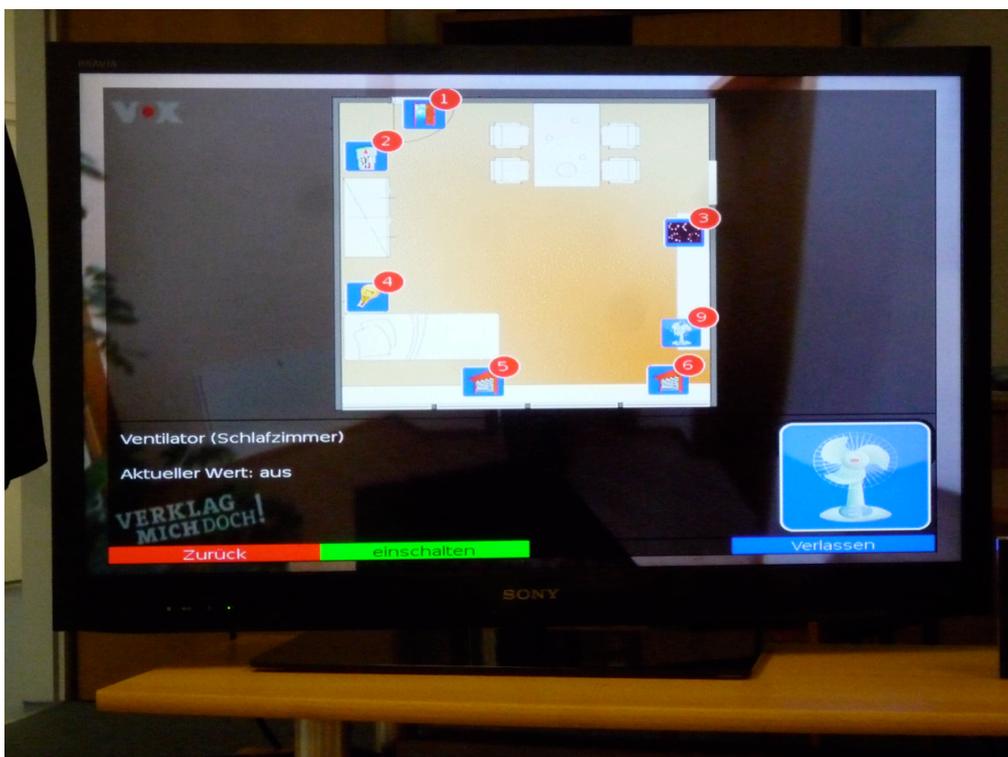


Abbildung 16: Steuerung eines Ventilators in der Home-Control-Funktion.

6. Zusammenfassung

Das Ziel dieser Diplomarbeit war die Ausarbeitung einer alternativen, Set-Top-Box-basierten Interaktionsmöglichkeit für bestehende AAL-Systeme, die bislang eine Benutzerinteraktion nur mit Hilfe eines PCs bzw. eines Touchscreen-PCs erlaubten.

Nach der Erläuterung der Ziele, die eine Ambient Assisted Living-Umgebung verfolgt und der Aufzählung der typischen Anwendungsfelder einer AAL-Plattform, wurden vier existierende, im Einsatz befindliche AAL-Systeme vorgestellt. Bei der Wahl der Vergleichssysteme wurde darauf geachtet, dass sowohl TV-basierte, als auch nicht-TV-basierte Systeme verwendet wurden. In einem direkten Vergleich wurden die Gemeinsamkeiten, aber auch die Unterschiede der vorgestellten Plattformen aufgezeigt. Dieser Vergleich lieferte die Basis für die Festlegung von funktionalen und nichtfunktionalen Softwareanforderungen. Neben der Home-Monitoring-Funktionalität und der Home-Control-Funktionalität wurde die einfache Integration in bereits bestehende AAL-Systeme als eine funktionale Softwareanforderung fixiert. Anschließend wurden die heutzutage üblichen Empfangsmöglichkeiten für TV-Signale sowie die Standards im Bereich interaktives Fernsehen genauer untersucht. Dieser Schritt listete die notwendigen Empfangstechnologien auf, welche die einzusetzende Set-Top-Box beherrschen sollte, um möglichst an vielen Orten einsetzbar zu sein. Auch auf den relativ neuen Standard für interaktives Fernsehen, den HbbTV-Standard, wurde kurz eingegangen. Für den Betrieb der Software wurden die Eigenschaften eines PCs, von einigen Smart-TV-Geräten sowie die von zwei Set-Top-Boxen untersucht. Die Wahl fiel auf die ReelBox Set-Top-Box, die einige Vorteile gegenüber der Dreambox aufweist. Im Softwaredesign wurde die Software entsprechend den funktionalen und den nichtfunktionalen Anforderungen ausgearbeitet. Dabei wurde auf die Eigenschaften der verwendeten ReelBox und des VDR-Programmes, in dem das System betrieben wird, Rücksicht genommen. Auf die Ergebnisse der Diplomarbeit wurde in dem letzten Abschnitt, in der konkreten Implementierung, eingegangen.

Diese Diplomarbeit bietet eine Übersicht über aktuelle Entwicklungen, aufkommende und auch gescheiterte Standards auf dem Gebiet interaktives Fernsehen und verdeutlicht, dass dieses Themengebiet relativ neuartig und durchaus dynamisch ist. Die konkrete Implementierung zeigt, dass eine Verwendung des TV-Gerätes als ein Ein- und Ausgabegerät für das dahinter befindliche AAL-Backend durchaus denkbar und auch realisierbar ist. Welcher Standard bzw. Technik sich in Zukunft etablieren werden, kann jedoch aus heutiger Sicht noch nicht klar beantwortet werden. Die hier verwendete ReelBox und die Implementierung der Software als ein VDR-Plugin bietet eine aktuell einsetzbare Lösung die

in Zukunft eventuell noch mehr auf die Bedürfnisse der Kunden/innen und auf die Änderungen am Marktsegment angepasst werden muss.

Literaturverzeichnis

Österreichischer Rundfunk. (06 2008). *Start von DVB-H: Mit ORF kommt EURO 08 auf Handy-Bildschirm*. Abgerufen am 16. 10 2011 von http://digital.orf.at/show_content2.php?s2id=530

Österreichischer Wirtschaftsverlag GmbH. (12. 04 2011). *Wiener DVB-T2 Testbetrieb wird verlängert*. Abgerufen am 26. 10 2001 von HbbTV-Pilotversuch auf Kanal 60: <http://www.elektrojournal.at/ireds-112068.html>

A1 Telekom Austria AG. (26. 09 2011). *A1 baut A1 TV in Bundesländern weiter aus*. Abgerufen am 20. 10 2011 von <http://www.a1.net/newsroom/2011/09/a1-baut-a1-tv-in-bundeslaendern-weiter-aus/>

AIT - Austrian Institute of Technology GmbH. (11. 05 2011). *HOMER - HOME Event Recognition System*. Abgerufen am 28. 11 2011 von <http://homer.aaloo.org>

Andrushevich, A., Kistler, R., Bieri, M., & Klapproth, A. (2009). *ZigBee/IEEE 802.15.4 Technologies in Ambient Assisted Living Applications*. Lucerne University of Applied Sciences and Arts, CEESAR-iHomeLab. München: 3rd European ZigBee Developers' Conference (EuZDC).

Anthony, S.-C. (2009). *HBB & HbbTV in the MHP environment Status and Prospects*. Frankfurt am Main, Deutschland.

Apache Software Foundation . (2008). *log4cxx - Short introduction to Apache log4cxx*. Abgerufen am 15. 11 2011 von <http://logging.apache.org/log4cxx/>

Becher, M. (2009). *XML* (1. Auflage Ausg.). Deutschland: W3I Akademie.

BEKO Engineering & Informatik AG. (2011). *BEKO Smart Home > HOMEBUTLER*. Abgerufen am 28. 11 2011 von <http://www.beko.at/index.php?id=1036>

Beutel, J. (2009). *Interaktives Fernsehen; Neue Werbeformen des interaktiven Fernsehens* (1. Auflage Ausg.). Deutschland: IGEL Verlag.

Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V. (08. 04 2011). *Computernutzung nimmt weiter zu*. Abgerufen am 01. 11 2011 von http://www.bitkom.org/de/presse/30739_67616.aspx

Burkner, D. F. (11. 09 2011). *German TTS-systems*. Abgerufen am 13. 11 2011 von <http://ttsamples.syntheticspeech.de/>

CEIT Central European Institute of Technology gemeinnützige GmbH. (01. 12 2009). *Assistive Smart-Home-Lösung "eHome"*. Abgerufen am 28. 11 2011 von <http://deutsch.ceit.at/news-events/assistivesmarthomeloesungehome>

Colaitis, F., & Jourdan, P. (1995). *MHEG and its profile for ITV applications. IEE Colloquium on Interactive Television*, (S. 3/1 - 3/8). London.

cURL Community. (06. 12 2011). *cURL and libcurl*. Abgerufen am 06. 12 2011 von <http://curl.haxx.se/>

- Daniel Liebhart, G. S. (2007). *Architecture blueprints: Ein Leitfaden zur Konstruktion von Softwaresystemen ...* (2. Auflage Ausg.). Hanser Verlag.
- Deutsche Forschungszentrum für Künstliche Intelligenz. (12 2010). *The MARY Text-toSpeech System*. Abgerufen am 13. 11 2011 von <http://mary.dfki.de/>
- Die Presse. (14. 11 2011). Abgerufen am 23. 11 2011 von Google TV: Logitech steigt aus, Google rüstet auf: http://diepresse.com/home/techscience/internet/google/708342/Google-TV_Logitech-steigt-aus-Google-ruestet-auf?from=simarchiv
- Digitales Fernsehen Förder GmbH. (10. 10 2011a). *DVB-T; Das digitale Antennenfernsehen*. Abgerufen am 10. 10 2011 von Allgemeine Information: <http://www.dvb-t.at/was-ist-das/allgemeine-information.html>
- Digitales Fernsehen Förder GmbH. (26. 10 2011). *DVB-T; Das Digitale Antennefernsehen*. Abgerufen am 26. 10 2011 von MHP: <http://www.dvb-t.at/glossar/m-o.html#c253>
- Dream Multimedia GmbH. (2011). *Produktübersicht | dream-multimedia-tv.de*. Abgerufen am 23. 11 2011 von <http://www.dream-multimedia-tv.de/produktuebersicht>
- Dream Multimedia GmbH. (23. 11 2011 a). *Startseite - Dreamboard*. Abgerufen am 23. 11 2011 von <http://www.dream-multimedia-tv.de/board/>
- DVB Project. (Mai 2010). *Introduction to the DVB Project*. Abgerufen am 19. 10 2011 von http://www.dvb.org/technology/fact_sheets/DVB-Project_Factsheet.pdf
- DVB Project. (2008). *MHP*. Abgerufen am 15. 10 2011 von Introduction to MHP & GEM: <http://www.mhp.org/introduction.htm>
- DVB Project. (Mai 2011). *Multimedia Home Platform fact sheet*. Abgerufen am 26. 10 2011 von http://www.dvb.org/technology/fact_sheets/DVB-MHP_Factsheet.pdf
- Fischer, W. (2009). *Digitale Fernseh- und Harfunktechnik in Theorie und Praxis* (2. Auflage Ausg.). München, Deutschland: Springer.
- Gerdes, S. (2003). *Das Internet als Distributionskanal*. Wiesbaden, Deutschland: Deutscher Universitäts-Verlag GmbH.
- Google Inc. (2011). *Get It - Google TV*. Abgerufen am 23. 11 2011 von <http://www.google.com/tv/get.html>
- GraphicsMagick Group. (2011). *GraphicsMagick Image Processing System*. Abgerufen am 28. 11 2011 von <http://www.graphicsmagick.org/>
- HbbTV Infoportal. (10 2011). *HbbTV - Infos*. Abgerufen am 10 2011 von Alles über HbbTV - dem interaktiven TV!: <http://www.hbbtv-infos.de/sender/hbbtv-senderliste.php>
- Heise Zeitschriften Verlag. (14. 10 2010). *DVB-H auch in Österreich gescheitert*. Abgerufen am 16. 10 2011 von <http://www.heise.de/mobil/meldung/DVB-H-auch-in-Oesterreich-gescheitert-1108096.html>
- Hester Bornman, S. v. (1993). Hypermedia, multimedia and hypertext: definitions and overview. (M. U. Ltd, Hrsg.) *The Electronic Library*, S. 259-268.

- Inkscape Community. (30. 11 2011). *Inkscape. Draw Freely*. Abgerufen am 05. 12 2011 von <http://inkscape.org/>
- Jonathan Duddington. (14. 03 2011). *eSpeak:Speech Synthesizer*. Abgerufen am 13. 11 2011 von <http://espeak.sourceforge.net/>
- Julien Maisonneuve, M. D. (05. 05 2009). An Overview of IPTV Standards Development. (I. B. Society, Hrsg.) *IEEE Transactions on Broadcasting* .
- Lambert M Surhone, M. T. (2010). *Set-Top Box*. VDM Verlag Dr. Miller AG & Co. Kg.
- Laposa, T. (2005). *Der Fußballkommentar im deutschen Fernsehen: Analyse des Endspiels bei der EURO 2004*. Norderstedt, Deutschland: GRIN Verlag.
- Lugmayer, A., Niiranen, S., & Kalli, S. (2004). *Digital Interactive TV and Metadata*. Finland: Springer Verlag.
- Lyubka, S. (16. 11 2011). *Mongoose - easy to use web server*. Abgerufen am 18. 11 2011 von <http://code.google.com/p/mongoose/>
- Mahle, N. (2004). *Das technische und strategische Potential von SVG als standardisierte Grafik* (1. Auflage Ausg.). Norderstedt, Deutschland: GRIN Verlag.
- Mahr, H. *Zaubermaschine interaktives Fernsehen?* (1. Auflage Ausg.). (C. z. Salm, Hrsg.) Gabler Verlag.
- Maier, M. (2008). *WebTV- Analyse eines neuen Geschäftsfelds von Zeitungsverlagen am Beispiel der Heilbronner Stimme*. Norderstedt, Deutschland: GRIN.
- NovaHome-Konsortium. (2011). *NovaHome*. Abgerufen am 28. 11 2011 von <http://www.novahome.at/>
- P. Panek, W. L. (2008). A Living Lab for Ambient Assisted Living in the Municipality of Schwechat. In J. K. Klaus Miesenberger, *Computers Helping People with Special Needs* (S. 1008–1015). Linz, UpperAustria, Austria: Springer-Verlag Berlin Heidelberg.
- Philips Deutschland GmbH. (09 2011). *Philips erweitert sein Smart TV-Angebot um weitere interessante Funktionen*. Abgerufen am 04. 10 2011 von http://www.newscenter.philips.com/de_de/standard/news/consumerlifestyle/20110901_philips_erweitert_sein_smart_tv-angebot_um_weitere_interessante_funktionen.wpd
- Philips Nederland B.V. (04. 11 2011). Abgerufen am 04. 11 2011 von About Philips NetTV partnerships: <http://www.philips.nl/c/over-philips-nettv-partnerships/12063/cat/nl/>
- Probst, M. (12. 04 2011). Hybrid Broadcast Broadband Television – HbbTV. München, Deutschland.
- Reel Multimedia Vertriebs GmbH. (10. 10 2011). *Hauptseite - RMM-Wiki*. Abgerufen am 23. 11 2011 von <http://wiki.reel-multimedia.com/index.php/Hauptseite>
- Reel Multimedia Vertriebs GmbH. (13. 10 2011). *Reel-Multimedia*. Abgerufen am 23. 11 2011 von http://www.reel-multimedia.com/de/shop_reelbox_satellitenfernsehen_technische_daten.php

- Reimers, U. (2008). *DVB - digitale Fernsehtechnik: Datenkompression und Übertragung* (3. Auflage Ausg.). Berlin, Deutschland: Srpinger Verlag.
- Riegamer, R. (2008). *IPTV und WebTV- Der neue Fernsehmarkt und seine Wettbewerber in Deutschland*. Norderstedt, Deutschland: GRIN.
- RNCOS Market Research. (2011). *Global IPTV Market Forecast to 2014*. INDIEN.
- RTR - GmbH. (31. 05 2011). Abgerufen am 26. 10 2011 von RTR-GmbH fördert ORF-Projekt für hybrides Fernsehen: <http://www.rtr.at/en/pr/PI31052011Medien>
- RTR - GmbH. (27. 06 2006). *RTR-GmbH fixiert Endgeräteförderung für DVB-T, das digitale Antennenfernsehen*. Abgerufen am 27. 10 2011 von <http://www.rtr.at/de/pr/PInfo27062006RF>
- Samsung Electronics GmbH . (03. 11 2011). Abgerufen am 03. 11 2011 von Samsung Smart TV Apps: <http://samsung.de/de/microsites/smarttvapps/default.aspx#>
- Samsung Electronics GmbH. (03. 11 2011 a). *Create with Samsung D Forum*. Abgerufen am 03. 11 2011 von <http://www.samsungdforum.com/>
- Schill, A., & Springer, T. (2007). *Verteilte Systeme: Grundlagen und Basistechnologien*. Dresden, Deutschland: Springer Verlag.
- Schmidinger, K. (18. 10 2009). *The Video Disk Recorder*. Abgerufen am 23. 11 2011 von <http://www.tvdr.de/index.htm>
- Scolik, R. (2007). *Was ist neu am neuen Fernsehen?* (J. Wippersberg, Hrsg.) Deutschland: LIT Verlag.
- Screen Digest. (2010). *The Global Transmission Market*.
- Sebastian, T. (2009). *Mobile TV: Das Handy als audiovisuelles Medium der Zukunft* (1. Auflage Ausg.). Deutschland: Igel Verlag.
- Seniorenzentrum Schwechat. (2011). *Demo-Wohnung im Seniorenzentrum*. Abgerufen am 28. 11 2011 von Hilfsmittel für ein leichteres Leben: <http://www.schwechat.gv.at/fs1/cs1/home/lebeninschwechat/senioren/demowohnung>
- SES Astra. (21. 10 2011). Abgerufen am 21. 10 2011 von Alle Fragen und Antworten zur Einstellung des analogen Fernsehens: <http://www.astra.de/4003575/faq>
- SevenOne Media GmbH. (10 2011). *HbbTV Trailer*. Abgerufen am 10 2011 von So funktioniert HbbTV: <http://www.sevenonemedia.de/web/sevenone/plattformen/hbbtv>
- Spanoudakis, N., Grabner, B., Kotsiopolou, C., Lymperopoulou, O., Moser-Siegmeth, V., Pantelopoulos, S., et al. (2010). A novel architecture and process for Ambient Assisted Living - the HERA approach. *10th IEEE International Conference. on Information Technology and Applications in Biomedicine*.
- Statistik Austria. (11. 11 2011). *STATISTIK AUSTRIA - Bevölkerungsprognosen*. Abgerufen am 25. 11 2011 von http://statistik.gv.at/web_de/statistiken/bevoelkerung/demographische_prognosen/bevoelkerungsprognosen/index.html

T. Fuxreiter, C. M. (2010). A modular platform for event recognition in smart homes. *e-Health Networking Applications and Services (Healthcom)* (S. 1-6). Lyon: 12th IEEE International Conference.

Technische Universität Wien, Institut 'integriert studieren' . (29. 09 2011). *Die e-Home Homepage*. Abgerufen am 28. 11 2011 von <http://www.is.tuwien.ac.at/ehome/>

The Apache Software Foundation. . (09. 06 2011). *The Apache Xerces Project - xerces.apache.org*. Abgerufen am 01. 12 2011 von <http://xerces.apache.org/>

The World Wide Web Consortium (W3C). (16. 08 2011). Scalable Vector Graphics (SVG) 1.1 (Second Edition).

United Nations. (2010). *World Population Ageing 2009*. Department of Economic and Social Affairs, Population Division, New York.

VDR Wiki. (26. 03 2011). Abgerufen am 07. 11 2011 von <http://www.vdr-wiki.de/wiki/index.php/Hauptseite>

VDR Wiki. (26. 03 2011). *VDR Wiki*. Abgerufen am 23. 11 2011 von <http://www.vdr-wiki.de/wiki/index.php/Hauptseite>

Versteegen, G. (2002). *Software-Management: Beherrschung Des Lifecycles*. Berlin, Deutschland: Springer Verlag.

Walter, F. (2009). *Digitale Fernseh- und Harfunktechnik in Theorie und Praxis* (2. Auflage Ausg.). Berlin, Deutschland: Springer Verlag.

Wang, C., & Shen, H.-W. (2005). Hierarchical Navigation Interface: Leveraging Multiple Coordinated Views for Level-of-Detail Multiresolution Volume Rendering of Large Scientific Data Sets. *Information Visualisation, 2005. Proceedings. Ninth International Conference on* .

Zier, T. (2007). *IPTV - Die Zukunft des Fernsehens ?* Norderstedt, Deutschland: GRIN.

Zlotnik, M. H. *World Population Ageing 2009*. Department of Economic and Social Affairs, Population Division. New York: United Nations Secretariat.

Abbildungsverzeichnis

Abbildung 1:	Links: HOMER Benutzeroberfläche die aktuell den Energieverbrauch des Hauses darstellt. Quelle: http://www.novahome.at/musterhaus/galerie . Rechts: Konfigurationsbereich der HOMER Plattform Quelle: http://homer.aaloo.org/redmine/projects/homer-core/wiki	12
Abbildung 2:	Links: Benutzerinteraktion mit der eHome-Plattform. Rechts: Konfigurationsbereich der eHome-Plattform. Quelle: http://www.is.tuwien.ac.at/ehome/e-HomeErgebnisse.html	13
Abbildung 3:	Links: HOMEBUTLER Benutzeroberfläche. Rechts: Übersicht der Dienste die von der HOMEBUTLER Plattform angeboten werden. Quelle: http://www.beko.at/index.php?id=1048	14
Abbildung 4:	Aufbau der HERA - Plattform. Quelle: http://www.math-info.univ-paris5.fr/~nikos/HERA_silo_wp5_d522_final.pdf	15
Abbildung 5	Interaktives Fernsehen anhand des HbbTV Standards. Verdeutlicht werden die verschiedenen Übertragungsmedien für lineare und nichtlineare TV-Inhalte (Quelle: HbbTV Infoportal)	24
Abbildung 6:	Benutzeroberfläche des Programms Enigma2. Abbildung links: Die Wahl des TV-Senders, Abbildung rechts: Darstellung von Informationen zum aktuellen TV-Programm. Quelle: http://www.saunalahti.fi/~znark/dreambox/dm7025/kuvagalleria/enigma2/	36
Abbildung 7:	Benutzeroberfläche des Programms VDR auf der ReelBox. Abbildung links: Die Wahl des TV-Senders, Abbildung rechts: Elektronischer Programmführer auf der ReelBox. Quelle: http://wiki.reel-multimedia.com	37
Abbildung 8:	Client/Server-Modell der AAL-Plattform. In der Home-Control-Funktion fungiert das AAL-Frontend als Client und in der Home-Monitoring-Funktion als Server. Analoges gilt für das AAL-Backend.	40
Abbildung 9:	Die drei Visualisierungsbereiche des AAL-Frontends	46
Abbildung 10:	Zwei Möglichkeiten für verschieden Startanzeigen des Übersichtsbereiches (a) und (b). Dynamische Namensgebung und die Auswahl eines Lichtaktuators (b), (c) und (d). Dynamische Namensgebung und die Auswahl der Heizungssteuerung (b), (c) und (e).	49
Abbildung 11:	Ablauf der TTS-Audioausgabe für Alarime und Warnmeldungen. Die Schritte 1-3 stellen eine direkte Ausgabe, die von der TTS-Engine gesteuert wird, dar. Die Schritte 1,2,4,5 und 6 stellen eine indirekte Ausgabe, die durch das AAL-Plugin gesteuert wird, dar.	52
Abbildung 12:	Links: Der klassische MVC-Ansatz. Rechts: Der MVC-Ansatz in der Praxis (aus Daniel Liebhart, 2007)	64
Abbildung 13:	Klassendiagramm des vdr4aal-Plugins	68
Abbildung 14:	vdr4aal-Plugin im Einsatz auf der ReelBox. Visualisierung der Home-Control-Funktion. Unter dem TV-Gerät ist die ReelBox erkennbar.	80
Abbildung 15:	Ebenfalls eine Darstellung der Home-Control-Funktion.	81
Abbildung 16:	Steuerung eines Ventilators in der Home-Control-Funktion.	81

Tabellenverzeichnis

Tabelle 1: Übersicht verschiedener Smart-TV-Gerätehersteller hinsichtlich Erweiterungsmöglichkeiten für eigene TV-Apps.	33
Tabelle 2: Regelwerk für die festgelegten Meldungskategorien	43
Tabelle 3: Funktionsbelegungen und Sichtbarkeit der vier Softkey-Tasten	50
Tabelle 4: Beispiele für verschiedene Aktionsnamen (Einschalten, Erhöhen, Öffnen usw.) und Zustandsnamen (offen, eingeschaltet, 27°C usw.) bei Aktuatoren für verschieden Einsatzgebiete.	61
Tabelle 5: Einteilung der Klassen und der HTML-Webseite in die Model-Viewer-Controller Designpatterns	65

Sourcecode Listings

- Listing 1: Ein Beispiel für die Abbildung des Settings mit Hilfe einer XML-Datei 58
- Listing 2: Definition eines AAL-Items innerhalb eines SVG-Group-Elements mit
Hilfe von zusätzlichen Attributen des Namenraums aal..... 60

Abkürzungs- und Akronymverzeichnis

AAL	Ambient Assisted Living
ACL	Access Control Lists
API	Application Programming Interface
App	Application
ARGB	Alpha Red Green Blue
ATSC	Advanced Television System Committee
CD	Compact Disk
CE-HTML	Consumer Electronic - HTML
CSS	Cascading Style Sheets
DNS	Domain Name System
DOM	Document Object Model
DTMB	Digital Terrestrial Multimedia Broadcast
DVB	Digital Video Broadcasting
DVD	Digital Versatile Disc
EPG	Electronic Program Guide
ETSI	European Telecommunications Standards Institute
FIFO	First In First Out
FTP	File Transfer Protocol
GNU	General Public License
HbbTV	Hybrid Broadcast Broadband TV
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
IPTV	Internet Protocol Television
ISDB	Integrated Service Digital Broadcasting
ISO	International Standardization Organization
JPEG	Joint Photographic Experts Group
JS	Java Script
LAN	Local Area Network
LIFO	Last In First Out
MHEG	Multimedia and Hypermedia Experts Group
MHP	Multimedia Home Platform
MPEG	Moving Picture Experts Group

MVC	Model Viewer Controller
OSD	On Screen Display
OSGi	Open Services Gateway initiative
PC	Personal Computer
PNG	Portable Network Graphics
POP	Post Office Protocol
RAM	Random Access Memory
RIFF	Resource Interchange File Format
SDK	Software Development Kit
SSL	Secure Sockets Layer
STB	Set-Top-Box
SVG	Scalable Vector Graphics
SVN	Apache Subversion
TCP/IP	Transmission Control Protocol / Internet Protocol
TTS	Text To Speech
TV	Television
TV-App	Television Application
URL	Uniform Resource Locator
USB	Universal Serial Bus
VDR	Video Disk Recorder
WAVE	Waveform Audio File Format
WLAN	Wireless Local Area Network
XML	Extensible Markup Language