



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Diplomarbeit

Hannes Osius

Entwicklung eines ubiquitären Media-Centers als
Set-Top-Box

Hannes Osius
Entwicklung eines ubiquitären Media-Centers als
Set-Top-Box

Diplomarbeit eingereicht im Rahmen der Diplomprüfung
im Studiengang Softwaretechnik
am Studiendepartment Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Kai von Luck
Zweitgutachter : Prof. Dr. rer. nat. Gunter Klemke

Abgegeben am 20. März 2006

Hannes Osius

Thema der Diplomarbeit

Entwicklung eines ubiquitären Media-Centers als Set-Top-Box

Stichworte

Disappearing Computer, .net, C#, Java, Media-Center, Set-Top-Box, ubiquitär, Medien

Kurzzusammenfassung

Die digitalen Medien drängen die analogen Medien immer mehr in der Hintergrund. Alle Unterhaltungsmedien (Audio, Video und Bilder) sind mittlerweile digital und können vom einem Standard PC wiedergegeben werden, so dass die herkömmlichen Abspielgeräte wie der CD-Spieler immer mehr ins Hintertreffen geraten. Wenn die Medien sowieso digital vorliegen, stellt sich die Frage ob man einen PC hinter der Fassade eines CD-Spielers (Set-Top-Box) verstecken kann. Diese Box wird an die Hifi-Anlage angeschlossen, und die Interaktion mit dem Benutzer wird über eine Fernbedienung realisiert. Diese Arbeit beschreibt den Weg einer solchen Vision, von der ersten Idee über einen Entwurf bis hin zu einem Prototypen.

Hannes Osius

Title of the paper

Deployment of a ubiquitous media centers as a set-top box

Keywords

disappearing computer, .net, c#, java, media center, set-top box, ubiquitous, media

Abstract

The digital media are eclipsing the analog media to an increasing extent. In the meantime all entertainment media (audio, video and pictures) has become digital and standard PCs are used for play-back. Conventional play-back devices like the CD player are becoming less important. If the media is already digital, it could make sense to hide a PC behind the front-panel of a CD player, to form a set-top box. This box is attached to the stereo system and the user can interact with it using a remote control. This work describes the way of such a vision, from the first draft to a prototype.

*Für Andrea, Anja, Helma, Kristine,
Carsten und Gerhard*

Danksagung

It's been a long road, getting from there to here.

An dieser Stelle möchte ich mich bei meinen betreuenden Professoren Herrn Prof. Dr. Kai von Luck und Herrn Prof. Dr. Gunter Klemke für ihren vielseitigen fachlichen Rat bedanken. Prof. Dr. von Luck möchte ich meinen besonderen Dank aussprechen, da mich in jeder Phase meiner Arbeit sehr sachkundig und richtungweisend begleitete. Ich weiß seine Unterstützung und seine Geduld zu schätzen. Im Weiteren möchte ich mich bei denjenigen bedanken, die mich bei dieser Arbeit unterstützt haben. Insbesondere danke ich Kristine, Roman, Roland und Carsten. Zum Schluss möchte ich mich bei Andrea und Gerhard bedanken. Ohne sie wäre ich nie so weit gekommen.

Dankeschön.

Inhaltsverzeichnis

Tabellenverzeichnis	9
Abbildungsverzeichnis	10
1. Einführung	12
1.1. Gliederung der Arbeit	14
2. Szenario	15
2.1. Einzelne Anwendungsfälle	16
2.1.1. Installation	16
2.1.2. Betrieb des Media Center	16
2.2. Akteure	18
2.3. Zusammenfassung	18
3. Analyse	19
3.1. Anforderungen an das Media-Center	19
3.2. Mögliche Lösungen	19
3.2.1. Feste Audio- und Videoverkabelung	20
3.2.2. Laptops als temporäre Multimedia-Zentrale	21
3.2.3. Media-Center	21
3.2.4. Streaming-Box	23
3.2.5. Tragbare Musikspieler	23
3.2.6. Zusammenfassung	25
3.3. Analyse des Szenarios	25
3.3.1. Interaktion mit dem Anwender	25
3.3.2. Speicherort der Musik	26
3.3.3. Wiedergabeliste	27
3.3.4. Zusammenfassung	28
3.4. Qualitätsmerkmale	28
3.4.1. Qualitätsmerkmale aus Anwendersicht:	29
3.4.2. Qualitätsmerkmale aus Entwicklersicht	34
3.5. Zusammenfassung	34

4. Design	36
4.1. Umgebung	36
4.2. Softwarearchitektur	37
4.3. Softwarekomponenten der Set-Top-Box	40
4.3.1. Die Kernkomponente (Core)	42
4.3.2. Komponenten des Benutzerinterfaces	48
4.3.3. Wiedergabekomponente	50
4.3.4. Komponente der Persistenzschicht	53
4.4. Hardware Architektur	53
4.5. Sequenzdiagramm	54
4.5.1. Starten der Wiedergabe	54
4.5.2. Erstellen einer Wiedergabeliste	55
4.5.3. Laden einer Wiedergabeliste	58
4.5.4. Einhängen eines neuen Datenverzeichnisses	60
4.6. Zusammenfassung	62
5. Realisierung	63
5.1. Machbarkeitsstudie	63
5.1.1. Hardware	64
5.1.2. Software	66
5.1.3. Zusammenfassung	68
5.2. Prototyp	69
5.2.1. Software	69
5.2.2. Die Anwendungslogik	70
5.2.3. Benutzerinterface	72
5.2.4. Persistenz	74
5.2.5. Sicherheit	77
5.2.6. Hardware	77
5.3. Zusammenfassung	82
5.3.1. Fazit	82
6. Ausblick	84
6.1. Zusammenfassung	84
6.2. Fazit	85
6.3. Ausblick	86
Literaturverzeichnis	88
A. Installation und Konfiguration des Media-Centers	93
A.1. Installation von Fedora Core 4	93
A.2. Konfiguration der Software	93

A.2.1. Netzwerk	93
A.2.2. Hardware Treiber	94
A.2.3. Java	98
A.2.4. Webmin	100
A.2.5. Zugriff auf die Netzwerkfeigaben	100
A.2.6. Media-Center Software	100
B. Platinen für den IOWarrior	101
B.1. Machbarkeitsstudie	101
B.2. Der Prototyp	101
B.3. Ansteuerungsplatine	101
B.3.1. Frontplatine	102
C. Umbau des Gehäuses	106
C.1. Stückliste	106
C.2. Montage W-Lan Karte	106
C.3. Umbau des Gehäuses	107
C.4. Winkel	107
D. Inhalt der CD	113
Glossar	114

Tabellenverzeichnis

4.1. Übersicht über die Schnittstellen der Kernkomponente	44
B.1. Prototyp: Stückliste der Basisplatine	102
B.2. Prototyp: Verdrahtungsplan	105
B.3. Prototyp: Stückliste der Frontplatine	105
C.1. Stückliste	106
D.1. Inhalt der CD	113

Abbildungsverzeichnis

3.1. Funküberträger von Conrad Electronic. Quelle:(Conrad Electronic, 2006) . . .	20
3.2. Der NOXON 2 von Terratec. Quelle: (TerraTec, 2006)	22
3.3. Die Squeezebox2 von Slim Devices. Quelle:(Devices, 2005)	24
3.4. Der iPod von Apple (fünfte Generation). Quelle:(Apple Computer, 2006a) . .	24
3.5. Alternative Client-Server-Anordnungen (Quelle:(Tanenbaum und van Steen, 2002))	33
4.1. Grobe Übersicht des Gesamtaufbaus	38
4.2. Die 3-Schichten-Architektur	39
4.3. Die Komponenten der Set-Top-Box	43
4.4. Ein Beispielbaum für das Menü	47
4.5. Das Zustandsdiagramm des Players	52
4.6. Starten der Wiedergabe	56
4.7. Eine Wiedergabeliste wird erstellt	57
4.8. Eine Wiedergabeliste wird über Menü geladen	59
4.9. Einhängen eines neuen Datenverzeichnisses	61
5.1. Das USBLCD Quelle:(Adams, 2005)	64
5.2. Das USBLCD Schnittstellen Modul. Quelle:(Adams, 2005)	64
5.3. Die Cw1602 von CwLinux. Quelle:(Cwlinux, 2005)	65
5.4. Die IO-Warrior24 von Code Mercenaries	65
5.5. Hardware: LC-Display, Taster, Ansteuerungsplatine und Fernbedienung . . .	67
5.6. Der Kern der Set-Top-Box: das VIA EPIA ME-6000 Mainboard	79
5.7. Die Ansteuerungsplatine mit dem IOWarrior24	80
5.8. Die fertig Set-Top-Box.	81
B.1. Schaltplan des Prototypen	103
B.2. Platine(von der Lötseite) des Prototypen	104
B.3. Bestückungsplan der Prototypplatine	104
B.4. Schaltplan der Frontplatine	104

C.1. Beigelegte Raiser Karte	107
C.2. Nachgekaufte WLAN-Karte1	107
C.3. Technische Zeichnung der Blende M 1:2	108
C.4. Technische Zeichnung der Frontplatte M 1:2	109
C.5. Bild der Blende	110
C.6. Die Set-Top-Box offen.	110
C.7. Der Winkel für die Lüftung	111
C.8. Gehäuse ohne Winkel	112
C.9. Der Winkel für die Lüftung	112

1. Einführung

Als Sony am ersten Oktober 1982 den ersten CD-Spieler (CDP-101) vorstellte, begann die Digitalisierung des Wohnzimmers. Bis dahin beherrschten die analogen Medien, wie die Schallplatte, Kassette und Tonbänder, das Wohnzimmer. Die Digitalisierung brachte den entscheidenden Vorteil, dass Medien immer wieder abgespielt werden können, ohne dass die Qualität abnimmt, auch die Lagerung über größere Zeiträume ist kein Problem¹. Später folgten dann mit DAT oder MiniDisk Möglichkeiten, auch digital aufzunehmen.

Sehr ähnlich verlief die digitale Entwicklung des Video: zunächst gab es nur die Möglichkeit, sich digitale Filme (Laserdisc, später dann DVD) anzuschauen, später dann auch die Möglichkeit, Filme selbst digital aufzunehmen (DV-Camcorder) und mit dem Heimcomputer weiter zu verarbeiten. Der Heimcomputer entwickelt sich in den letzten Jahren immer mehr in Richtung Multimedia-Zentrale.

Im Grunde kann man sagen, dass der Heimcomputer das Büro bzw. Arbeitszimmer in Richtung Wohnzimmer verlässt. Auf einem PC lässt sich heute, dank großer Festplatten und ausgereifter Kompressionsalgorithmen, die komplette Musiksammlung von mehreren Hundert Musik-CDs speichern. Die Speicherung am PC bietet den Vorteil, dass man die gesamte Musiksammlung immer im Zugriff hat. So kann man die gewünschte Musik mit ein paar Mausklicks anhören, ohne dass man erst die CD aus dem Regal heraussuchen und in den CD-Player einlegen muss. Die Zukunft für die Videosammlung sieht ähnlich aus: mit immer weiter steigender Festplattenkapazität ist es nur eine Frage der Zeit, bis man auch die gesamte Video- und DVD-Sammlung auf dem PC speichert.

Die CD war am Anfang das Transport- und Abspielmedium zugleich. Mittlerweile ist es nur noch das Transportmedium, mit dem die Musik verteilt wird. Nach dem Erwerb einer CD wird sie meist auf den heimischen PC überspielt, um sie in das digitale Archiv aufzunehmen. Es stellt sich die Frage, wann die CD als Transportmedium wegfällt und die Musik über das Internet gekauft wird. Dieser Markt verzeichnete in den letzten Jahren einen erheblichen Zuwachs, so dass zahlreiche Plattformen im Internet entstanden sind. Die bekannteste ist mit über 2 Millionen Musikstücken und über 1 Milliarde verkaufter Titeln seit dem Start im

¹ CDs lassen sich über mehrere Jahren lagern (siehe (NIST, 2006)), wie lange lässt sich nicht sagen, aber egal wie lange sich ein Speichermedium lagern lässt, man braucht immer noch ein Abspielgerät mit der passenden Software.

April 2003 (Stand Februar 2006) der „iTunes Music Store“ von der Firma Apple Computer (Apple Computer, 2006a).

Mit dem Beginn der digitalen Fernsehübertragung (DVB) über Satellit, Kabel und Antenne wurde auch der PC zum digitalen Fernseher und Videorekorder, der mehr Vorzüge als ein normaler Videorekorder bietet. So gibt es die Möglichkeit, seine Lieblingssendung aufzunehmen und, während die Aufnahme noch läuft, sie sich von Anfang an anzuschauen (Timeshift Aufnahme). Die mitübertragenen Metadaten erlauben es auch, dass Sendungen aufgenommen werden, die einen vielleicht interessieren könnten. Diese Geräte heißen „*Personal Video Recorder*“ (kurz PVR), der bekannteste Vertreter ist derzeit der Tivo (TiVo, 2006). Der Tivo Dienst ist nur in den USA erhältlich, MoreTV (moreTV, 2006) ist ein ähnlicher Dienst in Deutschland.

Auch bei den Fotoapparaten haben die digitalen Apparate den Durchbruch geschafft. Die Bilder werden auf dem PC gespeichert. Für eine Präsentation ohne Ausdruck fehlt es an den richtigen Anzeigengeräten: Zwar kann man die Fotos auf dem Fernseher oder Monitor darstellen, aber nicht in optimaler Auflösung. Aktuelle digitale Fotoapparate können Bilder mit mehr 8 Millionen Bildpunkten aufnehmen, der Fernseher kann ca 0,5 Millionen Bildpunkte darstellen, und ein normaler 19" TFT Monitor bei einer Auflösung von 1280*1024 Bildpunkten hat ca 1,3 Millionen Bildpunkte.

Es gibt also seit Jahren einen Trend der Digitalisierung der Unterhaltungsmedien, und ein Ende ist nicht in Sicht. Die Fachmessen CES (Association, 2006) und CeBIT (Deutsche Messe, 2006a) verzeichnen immer mehr Hersteller, die in diesem Bereich immer mehr Produkte anbieten. Parallel zur CeBIT 2006 findet die Sonderschau „*digital living*“ (Deutsche Messe, 2006b) statt. Diese behandelt nur den Bereich des digitalen Zuhauses. Diese Messen zeigen ganz klar, dass Computerwelt und Unterhaltungselektronik immer dichter zusammen rücken. So entsteht nach und nach eine Multimediawelt.

Mit der Digitalisierung und dem Zusammenspiel zwischen den Geräten steigt auch die Komplexität der Geräte und der Software. Es gibt kaum Geräte fürs Wohnzimmer, die eine einfache Handhabung ihrer gesamten Funktionen ermöglichen. Die größten Probleme gibt es mit der Konfiguration und bei der Interaktion mit anderen Geräten. Verschiedene Möglichkeiten des Wohnzimmer-PCs zeigt die Studienarbeit von Nicolas Botzet (Botzet, 2005). Hier wurden Media Systeme für Wohnzimmer getestet, mit Schwerpunkt auf die Bedienungs-freundlichkeit. Dort wurden Lösungen unter die Lupe genommen, die eine leistungsstarke Hardware voraussetzen. Zur Interaktion brauchen sie einen Fernseher und eine Fernbedienung oder Tastatur. Die Media-Center sind in der Lage, sowohl Bild, Musik als auch Video wiederzugeben. Das Resultat der Arbeit ist ernüchternd. Gerade weil die Geräte alles können, ist der Aufbau und Betrieb im Wohnzimmer nicht ohne Tiefe gehende PC-Kenntnisse möglich.

Dabei stellt sich doch die Frage, ob man es nicht schaffen kann, ein einfaches Gerät zu entwickeln, was nur für eine dieser drei Aufgabe gedacht ist? Man könnte einen PC hinter der Fassade eines CD-Spielers (Set-Top-Box) verstecken. Diese Box wird an die Hifi-Anlage angeschlossen und für die Interaktion mit dem Benutzer könnte man eine Fernbedienung benutzen.

Bei digitalen Archiven oder Media-Center kommt immer wieder die Frage der Legalität dieser Kopien auf. Die Aufgabe dieser Arbeit ist es nicht, das Recht auf Privatkopien (§53 UrhG) zu diskutieren. Es wird hier davon ausgegangen, dass alle Kopien, die im Heimnetzwerk freigegeben wurden, rechtmäßig erworben wurden und über das Media-Center abgespielt werden dürfen.

1.1. Gliederung der Arbeit

Im zweiten Kapitel wird das Szenario beschrieben, welches die Grundlage für die Anforderungen an das zu entwickelnde Media-Center beschreibt. Darauf aufbauend werden in Kapitel 3 die am Markt erhältlichen Produkte untersucht und bewertet, inwieweit sie die Anforderungen des Szenarios erfüllen. Dann wird festgelegt, wie ein Media-Center in ein bestehendes Netzwerk integriert werden kann, und auf welche Qualitätsmerkmale dabei zu achten ist. In Kapitel 4 wird dann die Umgebung, in der das Media-Center aufgebaut wird, konkretisiert und eine Soft- und Hardware Architektur entworfen, mit der sich die Anforderungen umsetzen lassen. Das entwickelte Design wird dann in Kapitel 5 umgesetzt. Erst wird mit einer Machbarkeitstudie überprüft, ob die Zielsetzung überhaupt möglich ist, dann wird ein Prototyp entwickelt. Zum Ende wird in Kapitel 6 die Arbeit zusammengefasst, ein Fazit gezogen und dann ein Ausblick auf künftige Entwicklungen gegeben.

2. Szenario

Anhand des folgenden Szenarios soll untersucht werden, inwieweit es möglich ist, eine gesamte Audiosammlung hinter der Fassade eines CD-Spielers zu verstecken. Als Grundlage des Szenarios dient die fiktive Familie Müller. Die Familie besteht aus Herrn und Frau Müller und ihren zwei Kindern. Die Familie wohnt in einem Einfamilienhaus in einem Vorort einer größeren Stadt. Im Besitz der Familie befinden sich zwei Computer und ein Laptop. Ein Computer steht im Arbeitszimmer, der andere im Spielzimmer der Kinder, den Laptop benutzt Frau Müller als mobile Arbeitsstation. Der Bürocomputer und der Computer der Kinder sind über ein kabelgebundenes Netzwerk an einen Router angeschlossen. Der Router stellt bei Bedarf eine Verbindung mit dem Internet her. Der Laptop ist bei Bedarf über ein W-Lan¹ mit dem Router verbunden. Die gemeinsamen Dokumente (z. B. Photos, Briefe usw.) werden auf einer Netzwerkfestplatte (kurz NAS²) gespeichert, damit sie von Laptop und PC immer zugänglich sind.

Familie Müller hat eine große Musiksammlung von über 500 CDs. Eine so große CD Sammlung bietet sicherlich für jeden Anlass die passende Musik, aber das Heraussuchen der richtigen Musik ist auch immer mit Aufwand verbunden. Die gesuchte CD muss in der Sammlung gefunden werden, nach dem abspielen, muss sie richtig zurücksortiert werden. Ein ziemlich hoher Aufwand, wenn man nur ein oder zwei Musikstücke von einer CD hören will. Leider ist es auch nicht ohne zusätzliche Komponenten, wie z. B. einen CD-Wechsler, möglich, eine Auswahl von Musikstücken von verschiedenen CDs anzuhören. Ein weiteres Problem ist der große und immer weiter wachsende Platzbedarf der Sammlung. Hier soll nun eine Lösung gefunden werden, die es ermöglicht, von der Couch aus per Fernbedienung durch die CD Sammlung zu navigieren und die Musik auszuwählen, oder eine vorher erstellte Auswahl abzuspielen. Herr und Frau Müller möchten eine Lösung, die ihren Geldbeutel nicht so stark belastet, das Limit liegt bei 500€. Die laufenden Kosten sollen sich auf die Stromkosten begrenzen, dabei soll der Stromverbrauch nicht den Verbrauch einer normalen Hifi-Komponente überschreiten. Die Müllers möchten nicht, dass das Gerät immer an ist, es

¹W-Lan (Wireless Local Area Network) ist die Bezeichnung für ein drahtloses lokales Funknetz. Genaueres unter (Sikora, 2001)

²NAS ist die Kurzform für Network Attached Storage. NAS sind reduzierter abgespeckte Dateiserver für den Heimbereich. Im Vergleich sind sie billiger und einfacher zu installieren, bieten aber meist keine Möglichkeit der Erweiterung (Speicherplatz oder Software). Durch die vereinfachte Administration bieten sie auch nur geringe Möglichkeiten, den Zugriff der Anwender auf Dateien einzuschränken.

muss über eine Stromsparfunktion (Stand-by) verfügen. Anhand dieser Vorgaben und den folgenden Szenarios soll ein möglichst gut passendes Media-Center für die Familie Müller gefunden oder entwickelt werden.

2.1. Einzelne Anwendungsfälle

Die Anwendungsfälle lassen sich in zwei Bereiche aufteilen:

1. Installation und Administration
2. Betrieb

In die Kategorie Installation gehört der einmalige Aufbau des Gerätes und das Anpassen an die örtlichen Gegebenheiten. Zur der Administration gehört die Einrichtung des Gerätes und die spätere Einrichtung von neuen Zusatzkomponenten. Zum Bereich Betrieb gehört die ständige Interaktion mit dem Benutzer.

2.1.1. Installation

Es handelt sich hier um eine Komponente, die in eine normale Hifi-Anlage im Wohnzimmer integriert werden soll. Man kann nicht davon ausgehen, dass die Benutzer in der Lage sind, komplexe technische Konfigurationen und Installationen durchzuführen. Auch in der Familie Müller hat niemand große Erfahrung mit dem Einrichten von Computern. Aus diesem Grund soll die Installation des Gerätes so einfach wie möglich sein und von Personen durchgeführt werden können, die keine Erfahrung mit Computern und ihrer Vernetzung haben. Die Installation soll bekannten Installationen, wie der eines CD-Spielers oder einer anderen Hi-Fi Komponente, möglichst ähnlich sein. Neben der einfachen Installation und Administration möchte Frau Müller, dass keine neuen Kabel im Haus verlegt werden müssen.

2.1.2. Betrieb des Media Center

Die Kernaufgabe des Gerätes ist die Wiedergabe von Multimediainhalten.

- Herr Müller möchte im Wohnzimmer in Ruhe Zeitung lesen und dazu Musik hören. Er wählt am Media-Center eine seiner vorher zusammen gestellten Favoritenlisten aus und lässt sie abspielen. Dabei will er noch die Möglichkeit haben, per Fernbedienung steuernd einzugreifen, um z. B. ein Lied weiter zuspringen oder die Wiedergabe anzuhalten, also Steuermöglichkeiten, die man von einem CD-Spieler kennt. Während der Wiedergabe sollen Information über das Musikstück abrufbar sein. Nachdem Herr

Müller die Zeitung zu Ende gelesen hat, möchte er eine neue CD in den Datenbestand des Media-Center aufnehmen. Die Musiksammlung wird also immer weiter wachsen. Wenn der vorhandene Speicherplatz voll, ist möchte Her Müller einfach eine weitere Speichereinheit in das Media-Center integrieren. Die Musik, die nicht mehr erwünscht ist, will Herr Müller entweder ins Archiv verschieben oder ganz vom Speichermedium des Media-Centers entfernen.

Das Vertrauen von Herrn Müller in die digitalen Speichermedien ist nicht sehr groß, deshalb möchte er regelmäßig eine Backup des Datenbestandes anfertigen können, und diese Sicherung bei einem Datenverlust (Ausfall eines Speichermedium oder versehentliche Löschung von Daten) wiederherstellen. Herr Müller geht davon aus, dass sich die Media-Center in Zukunft noch stark weiter entwickeln werden und mehr Funktionen bieten, er will sich deswegen nicht auf ein Gerät oder einen Hersteller festlegen. Somit muss das Gerät über eine Exportfunktion verfügen, oder die Musikdaten und Favoritenliste in einem standardisieren Format speichern.

- Frau Müller möchte einerseits, dass die vielen CDs aus dem Wohnzimmer verschwinden: andererseits möchte sie zukünftig einen schnellen Zugriff auf die Musik haben. Dabei ist ihr wichtig, dass sie leicht einen Interpreten oder ein Album auswählen kann. Wenn sie am Wohnzimmertisch sitzt und arbeitet, will sie das Media-Center bequem über den Laptop steuern. Frau Müller hat eine Vorliebe für klassische Musik, deshalb darf das Gerät nicht zuviel oder besser keinen Lärm verursachen. Außerdem ist es ihr wichtig, dass ein solches Media-Center immer zur Verfügung steht, nicht wie ein PC erst gestartet. Frau Müller kauft nicht nur CDs im Musikladen sondern, auch einzelne Musikstücke über das Internet, diese Musikstücke speichert sie auf dem Laptop. Das Media-Center sollte über eine Importfunktion verfügen, damit Frau Müller diese Musik auch hören kann, wenn der Laptop nicht zu Verfügung steht.
- Die Kinder habe ganz andere Anforderungen als ihre Eltern: sie wollen die Musik hören, die sie auf dem Kinder-PC oder auf ihrem MP3-Spieler gespeichert haben. Die Tochter erstellt die Wiedergabelisten für ihren MP3-Spieler mit einer Software auf dem PC, welche dann die Listen und die Musik auf den MP3-Spieler überspielt. So kann sie unterwegs leicht die Musik ihrer Stimmung anpassen, indem sie einfach eine der vorher erstellten Wiedergabelisten auswählt. Für das Media-Center im Wohnzimmer möchte sie etwas vergleichbares, sie will die vorher erstellten Wiedergabelisten einfach abrufen können. Die Kinder möchten außerdem über eine Internetradiostation die aktuellen Top 10 hören.

Ein paar der hier genannten Anforderungen lassen sich mit der bestehenden Hardware erledigen. Einige NAS-Lösungen bieten die Möglichkeit eine USB-Festplatte anschließen oder eine weitere Festplatte einbauen. So kann man die Kapazität erweitern, oder die zusätzliche Festplatte als Backupmedium benutzen.

2.2. Akteure

Aus den Anwendungsfällen in 2.1 ergeben sich zwei unterschiedliche Typen von Akteuren, die das Gerät benutzen: die Benutzer oder Anwender und die Administratoren.

- *Benutzer oder Anwender* sind Personen, die das Gesamtsystem benutzen, um Musik zu hören oder neue Musik in den Datenbestand aufzunehmen.
- *Administratoren* sind hier Personen, die mindestens minimale Erfahrungen mit Computern und Installation von Hardware und Software haben. Sie bauen das Gerät auf und richten es ein, so dass die Anwender es benutzen können.

2.3. Zusammenfassung

Das hier geschilderte Szenario soll die Grundlage für diese Arbeit sein. In den folgenden Kapiteln soll nun untersucht werden, ob und wie man die Wünsche der Familie Müller umsetzen kann.

3. Analyse

Die PCs haben bis jetzt noch nicht den Sprung vom Büro ins Wohnzimmer geschafft. Der Grund dafür dürfte wohl die komplexe Bedienung und Installation sowie die Laufgeräusche und das Design. Es gibt am Markt kaum Lösungen die es ermöglichen eine PC ohne Tastatur, Maus und (hochauflösenden) Monitor zu steuern. In diesem Kapitel soll untersucht werden, wie die im Szenario 2 geschilderten Anforderungen umgesetzt werden können. Dazu werden zunächst am Markt erhältliche Produkte auf ihre Eigenschaften untersucht und bewertet, wie weit sie die Anforderungen aus dem Szenario erfüllen.

In Abschnitt 3.3.1 wird analysiert, wie ein leichtgewichtiges Media-Center (siehe Abschnitt 3.2.3) die vorhandene Hardware benutzen kann, und wie man die Interaktion mit dem Benutzer ergonomisch gestalten kann. In Abschnitt 3.4 wird auf die Merkmale eingegangen, nach denen Anwender und Entwickler die Qualität eines System bewerten. In diesem Abschnitt wird untersucht, wie die Merkmale auf das Media-Center abzubilden sind und wie sie umgesetzt werden können.

3.1. Anforderungen an das Media-Center

Aus dem Szenario lässt sich ableiten, dass die Familie Müller ein Media-Center haben will, dessen Primärfunktion die Wiedergabe von Musik ist. Das Gerät soll ebenso einfach zu bedienen sein wie eine andere Hifi-Komponente oder ein Fernseher, über eine Fernbedienung oder am Geräte selbst. Während der Wiedergabe sollen Informationen über das Musikstück abrufbar sein, z.B. der Name des Musikstückes und die aktuelle Position, oder die Restzeit.

3.2. Mögliche Lösungen

Für die Lösung der im Szenario gestellten Anforderung gibt es mehrere Ansätze. In diesem Abschnitt sollen nun ein paar mögliche Lösungen vorgestellt und bewertet werden, wie sie in das Szenario passen. Mögliche Lösungen sind:

- Audio- und Videoverkabelung zwischen dem PC im Büro, und Hifi-Anlage und TV im Wohnzimmer.
- Laptops als temporäre Multimediazentrale.
- Ein zum Multimediasystem erweiterter PC
- Streaming-Box in Verbindung mit einem PC im Büro.
- Tragbarer Musikspieler.

3.2.1. Feste Audio- und Videoverkabelung

Dies ist wohl die einfachste Lösung, um die auf dem PC gespeicherten Daten im Wohnzimmer anzuhören und anzusehen. Problematisch sind bei dieser Lösung die Verkabelung und die Interaktion mit dem PC. Bei der Verkabelung haben Länge des Kabels und die örtliche Gegebenheiten Einfluss auf die Qualität. Während das Audiosignal nicht so stark unter langen Kabeln (10 bis 20m) leidet, ist die Qualität von Videosignalen abhängig von der Kabellänge. Mehr als 20m sind kaum zu vertreten. Falls es baulich keine Möglichkeit gibt, das Kabel vom Büro ins Wohnzimmer zu verlegen, kann man auf kabellose Funkübertragung (Abbildung 3.1) zurückgreifen. Manche der Funküberträger bieten auch die Möglichkeit, mit einem Infrarot-Funkwandler ein Infrarotsignal zurück zum Sender zu übermitteln.



Abbildung 3.1.: Funküberträger von Conrad Electronic. Quelle:(Conrad Electronic, 2006)

Da die Familie Müller einen PC im Arbeitszimmer hat, könnte man diesen über ein Stereo-Audio-Kabel mit der Hifi-Anlage im Wohnzimmer verbinden. Das widerspricht allerdings den Anforderungen von Frau Müller. Besser wären die schon erwähnten Funküberträger, welche um die 75€ kosten. Es müssten keine Kabel verlegt werden, und der lärmende PC würde nicht im Wohnzimmer stehen. Die meisten Anforderungen könnte man durch das Installieren von Software auf dem PC erfüllen, doch die Steuerung per Fernbedienung vom Sofa aus wäre ohne Rückmeldung an den Benutzer, man hätte also keine Informationen über das

laufende Musikstück. Ebenfalls müsste man für die Auswahl immer an den PC ins Arbeitszimmer gehen. Einen ähnlichen Ansatz verfolgt die Arbeit von Lars Wulff (Wulff, 2005): hier wird versucht einen Multimedia-Provider über einen Webservice zu realisieren. Der Provider bietet die Multimedia-Daten an, und Clients (PDA) können dann aus diesen Inhalten etwas auswählen.

3.2.2. Laptops als temporäre Multimedia-Zentrale

Wenn man im Besitz eines Laptops ist, kann man diesen zumindest als temporäre Multimediazentrale im Wohnzimmer benutzen, doch das bedeutet, dass man die Multimediaanlage immer wieder auf- und abbauen muss. Die Interaktion mit dem Benutzer ist hier schon besser als bei der in 3.2.1 beschriebenen Kabel-Lösung. Man muss zwar nicht immer bis ins Büro laufen, um ein anderes Lied zu hören, aber man muss zum Umschalten zum Laptop gehen, wenn sich dieser nicht von einer Fernbedienung steuern lässt.

Statt den Laptop immer auf- und abzubauen, könnte man ihn fest in die Hifi-Anlage integrieren. Für die hier gestellte Aufgabe der Musikwiedergabe würde auch ein älterer Laptop ausreichen. Für die Darstellung der Musikinformationen könnte man das Display des Laptops benutzen. Der Nachteil ist, dass Laptopdisplays derzeit noch sehr stark von Sichtwinkel abhängig sind. Weitere Nachteile sind die Laufgeräusche des Laptop und das notwendige hochfahren bevor man Musik hören kann.

3.2.3. Media-Center

Es gibt zwei Arten von Media-Centern: einmal die „*leichtgewichtigen*“ und die „*schwergewichtigen*“ Systeme.

Leichtgewichtige Media-Center

Zu den leichtgewichtigen¹ Media-Center zählen die Geräte, die versuchen, mit möglichst wenig Hardware auszukommen und so eine sehr kompakte Bauform zu nutzen. Zur Interaktion mit dem Benutzer kommen oft ein kleines Display und eine Fernbedienung zum Einsatz. Die Einrichtung und Wartung ist über eine Weboberfläche möglich. Die leichtgewichtigen Media-Center sind Spezialgeräte für nur eine Aufgabe: sie versuchen sich in bestehende Infrastrukturen zu integrieren und diese möglichst mitzunutzen.

¹Diese Art von Geräten ist auch unter dem Begriff „*disappearing computer*“ bekannt (Streitz und Nixon, 2005).

Ein extremes Leichtgewicht ist der NOXON 2 (Abbildung 3.2) aus dem Hause Terratec (TerraTec, 2006). Das Gerät ist für die Wiedergabe von Musik ausgelegt. Es verfügt deshalb über keinen eigenen Speicher für Musik oder Wiedergabelisten. Die Musik holt sich der NOXON 2 aus dem Netzwerk (Kabel oder W-Lan) oder von einer extern Festplatte (USB). Um die Musik im Heimnetzwerk zu finden, wird ein Media Server² vorausgesetzt.



Abbildung 3.2.: Der NOXON 2 von Terratec. Quelle: (TerraTec, 2006)

Schwergewichtige Media-Center

Ein schwergewichtiges Media-Center ist ein leistungsstarker PC, der mit zusätzlicher Hard- und Software ausgerüstet ist. Fast immer wird auf einen Monitor verzichtet, stattdessen wird der Fernseher zum Darstellen des grafischen Benutzerinterface verwendet. Die Software vereint dann Musik, Video, TV-Wiedergabe, TV-Aufnahme und Photo Verwaltung unter einem einfachen grafischen Interface. Durch den Umfang der Funktionen reicht bei der Steuerung meist die Fernbedienung nicht aus, so dass oft auf Maus oder Tastatur zurückgegriffen werden muss. Näheres zu den verschiedenen Media Centers findet sich in der Studienarbeit von Nicolas Botzet (Botzet, 2005). Diese Studienarbeit zeigt, dass es derzeit noch viele Probleme zwischen Hardware und Software gibt. Diese Probleme sollen laut Intel (Intel, 2006b) demnächst der Vergangenheit angehören. Intel hat mit dem *VII/V* Logo eine Art Qualitätssiegel für wohnzimmertaugliche Hardware eingeführt. Ähnlich wie bei dem Centrino³ Logo handelt

²Der Noxon nutzt das UPnP-AV Protokoll (Microsoft, 2006a) um die Multimedia Objekte im Netzwerk zu finden. Immer häufiger werden diese Media Server mit in den NAS integriert.

³Um das Centrino (Intel, 2006a) Logo zu bekommen, die Hardware muss eine bestimmte Kombination von Chipsatz, CPU und W-Lan-Chip entsprechen. Dabei handelt es um stromsparende Hardware, aus diesem Grund ist Centrino zu einem Synonym für leistungsstarke und stromsparende Laptops geworden.

es sich um eine Spezifikation für Hardware und Software, die zusammen einen festgelegten Funktionsumfang garantieren. Dazu zählt z. B. die Interaktion mit einer Fernbedienung oder der Transfer von Daten zwischen den verschiedenen Geräten. AMD hat unter dem Label *AMD LIVE* (AMD, 2006) etwas ähnliches veröffentlicht.

Fazit

Die schwergewichtigen Media-Center bieten alle die von der Familie Müller geforderten Funktionen, sind aber schwer aufzubauen und zu warten. Für die Interaktion ist es erforderlich, dass die Geräte an den Fernseher angeschlossen sind. Zur Auswahl der Musik muss dann immer der Fernseher eingeschaltet sein. Die leichtgewichtigen Media-Center wären da schon besser, da sie ohne Fernseher auskommen. Der NOXON 2 ist eine gute Plattform, allerdings ist er auf den Server angewiesen.

3.2.4. Streaming-Box

Ein anderer Lösungsansatz ist die Streaming-Box. Diese Lösung besteht aus zwei Komponenten, einem PC und einer Streaming-Box. Die Streaming-Box wird an eine vorhandene Hifi-Anlage angeschlossen und auf dem PC wird eine Serversoftware installiert. Beide Komponenten werden dann übers Netzwerk (Kabel oder W-Lan) miteinander verbunden. Die Serversoftware spielt die Musik dann ins Netzwerk, eine oder mehrere Streaming-Boxen empfangen die Musikdaten, decodieren diese und geben sie über die Hifi-Anlage wieder. Da auch mehrere Streaming-Boxen den gleichen Musikstream wiedergeben können, erweist sich die Steuerung der Wiedergabe als komplexer als beim Multimediasystem (siehe Abschnitt 3.2.3). Manche Streaming-Boxen bieten keine Bedienmöglichkeit. Man kann dann die Musik nur über den PC steuern, was bedeutet, dass man für eine Änderung immer zum PC gehen muss. Andere Streaming-Boxen bieten die Möglichkeit, die Musik über eine Fernbedienung zu steuern. Die Abbildung 3.3 zeigt die Squeezebox2 von Slim Devices.

Die Streaming-Box könnte man auch zu den leichtgewichtigen Media-Centern zählen, doch sie kommen nicht ohne zusätzliche Software aus. In die Anforderung der Familie Müller passt eine solche Box eigentlich ganz gut, doch bedeutet diese Lösung, dass man wie auch bei der festen Verkabelung immer erst einen PC starten muss, um Musik zu hören.

3.2.5. Tragbare Musikspieler

Möglich wäre auch, einen tragbaren Musikspieler wie den iPod an die Hifi-Anlage anzuschließen. Der iPod (siehe Abbildung 3.4) von Apple (Apple Computer, 2006a) ist der bekannt-



Abbildung 3.3.: Die Squeezebox2 von Slim Devices. Quelle:(Devices, 2005)

teste tragbare Musikspieler Spieler. Der Inhalt des iPod wird mit Hilfe der Software iTunes verwaltet. Die Wiedergabelisten werden auf dem PC erstellt und dann mit der Medienobjekten auf den iPod übertragen. Für iPod gibt es jede Menge Zubehör (z. B. Fernbedienung oder Docking Station für die Hifi-Anlage), die es ermöglicht den iPod in die Stereoanlage zu integrieren.



Abbildung 3.4.: Der iPod von Apple (fünfte Generation). Quelle:(Apple Computer, 2006a)

Der Vorteil dieser Lösung sind:

- keine Verkabelung quer durchs Zimmer.
- keine Wartezeit beim Einschalten.
- kaum hörbare Laufgeräusche.

Der Nachteil ist, dass der iPod erst an den PC angeschlossen werden muss, um neue Wiedergabeliste und Inhalte zu überspielen.

3.2.6. Zusammenfassung

Damit man im Wohnzimmer die digital gespeicherte Musik hören kann, gibt es weit mehr als nur eine Lösung. Die hier vorgestellten Lösungen verfolgen ganz unterschiedliche Ansätze. Die Festverkabelung, die Streaming-Box und der iPod setzen einen PC voraus, während die anderen Lösungen ein eigenständiger PC sind. Eine Sonderrolle spielen die leichtgewichtigen Media-Center, sie setzen auf eine vergleichsweise schwache Hardware. Da die Müllers keinen vollwertigen PC im Wohnzimmer haben möchten, und auch keinen Server, der 24 Stunden am Tag läuft, bleibt eigentlich nur das leichtgewichtige Media-Center als favorisierte Lösung übrig.

3.3. Analyse des Szenarios

Keine der im vorigen Abschnitt beschriebenen Lösungen erfüllt alle Anforderungen des Szenarios. Deshalb muss ein eigenes Media-Center entwickelt werden. Da die leichtgewichtigen Media-Center den Vorstellungen der Familie Müller recht nahe kommen, orientiere ich mich daran. Leichtgewichtige Media-Center besitzen keinen eignen Speicher für Medienobjekte, so dass ein anderer Speicherort für die Musik gefunden werden muss. In dem folgenden Abschnitt soll nun analysiert werden, wie ein Media-Center realisiert werden kann. Das Ziel ist ein Media-Center als Set-Top-Box, das in eine bestehende Hifi-Anlage integriert werden kann. Die Set-Top-Box soll einfach zu bedienen sein. Die Interaktion mit dem Anwender soll vergleichbar mit einem CD-Spieler sein. Die Box soll in der Lage sein, Mediendaten aus einer vorher angelegten Liste der Reihe nach wiederzugeben. Die Liste soll auch die vom CD-Spieler bekannten Funktionen wie zufällige Wiedergabe oder eine Wiederholung der Liste ermöglichen.

Def.: Eine **Set-Top-Box** ist ein Zusatzgerät für vorhandene Hardware (z. B. Hifi-Anlage oder Fernsehapparat), das deren Funktion erweitert. Eine Set-Top-Box verfügt meist über ein Statusdisplay, die Steuerung erfolgt über eine Fernbedienung. Eine Set-Top-Box hat einen kleinen Speicher für die Einstellungen, aber nicht für Wiedergabedaten. Eine Set-Top-Box ist im folgenden ein Synonym für ein leichtgewichtiges Media-Center.

3.3.1. Interaktion mit dem Anwender

Wenn das Gerät als Set-Top-Box realisiert wird, sind die Interaktionsmöglichkeiten mit dem Anwender begrenzt, da zur direkten Interaktion nur die Fernbedienung und eine Statusanzeige zur Verfügung stehen. Aus Sicht der Softwareergonomie ist über diesen Weg nur die

Steuerung der Wiedergabe vernünftig zu realisieren. Für die anderen Aufgaben (Verwalten des Datenbestands und Wiedergabelisten) muss eine andere Lösung gefunden werden. Anbieten würde sich eine Steuerung über externe Geräte, jedoch wünschen die Müllers keine weiteren Geräte im Wohnzimmer, so dass nur noch die Steuerung über den Fernseher oder über das Netzwerk übrig bleibt. Die Steuerung über den Fernseher würde aber auch nicht ohne Tastatur oder ein ähnliches Eingabegerät möglich sein. Die Müllers nutzen ein kabelloses Netzwerk für die Anbindung des Laptops, über dieses W-Lan könnte man auch den Zugriff auf die Set-Top-Box realisieren. So könnte man dann die administrativen Aufgaben von dem Büro-PC oder dem Laptop aus erledigen. Ein Nachteil bei der Verlagerung der administrativen Aufgaben ist, dass man dafür immer erst an den PC gehen muss. Der Vorteil ist, dass die Erledigung dieser Aufgabe vom PC aus mehr Komfort bietet (Maus, Tastatur und Monitor). Außerdem ist dem Anwender diese Aufteilung nicht fremd. Bei tragbaren digitalen Musikspielern gibt es eine identische Aufteilung (siehe Abschnitt 3.2.5), so ergibt sich eine vernünftige Trennung zwischen den Routine- und den Komfortfunktionen. Die Routine- und den Komfortfunktion werden wie folgt festgelegt:

Def.: Routinefunktion bzw. **Interaktionsfunktion** sind im Bezug auf die Set-Top-Box:

- Steuerung der Wiedergabe (play, stop, next, previous).
- Auswahl eine Wiedergabeliste.
- Navigation über den Inhalt.

Def.: Komfortfunktionen bzw. **Konfiguration** sind:

- Verwalten des Datenbestandes.
- Verwalten von Wiedergabelisten.
- Konfiguration der Set-Top-Box.

Diese Definition bildet die Grundlage für die Funktionsverteilung. Die Routinefunktionen sollen direkt am Geräte über Taster und Fernbedienung zugänglich sein. Die Komfortfunktionen sollen über das Netzwerk bereitgestellt werden. Die Routinefunktion wird so aufgebaut, wie man es von einem CD-Spieler her kennt, und die Komfortfunktionen könnte ähnlich realisiert werden wie die Administration eines Routers, z. B. über ein Webinterface. Die Interaktion mit einem Webinterface (z. B. Web-Mail oder Onlineshopping) ist heutzutage sehr verbreitet und kann somit als bekannt vorausgesetzt werden.

3.3.2. Speicherort der Musik

Da die Set-Top-Box normalerweise nur über einen sehr kleinen Speicher verfügt, muss die Musik woanders gespeichert werden. Es gibt folgende Möglichkeiten:

- Externer Speicher (siehe NOXON 2)
- Speichern im Netzwerk
 - auf dem PC bzw. Laptop
 - auf dem NAS

Die Speicherung der Musik auf einem externen Medium (z. B. USB-Festplatte), welches an die Set-Top-Box angeschlossen wird. Hat den Nachteil, dass die Musik immer erst vom PC oder Laptop auf das Media-Center überspielt werden muss und es so leicht zu ungleichen Datenbeständen führen kann, hat aber den Vorteil, dass man zum Musikhören keine Verbindung zum Heimnetz braucht. Die Speicherung im Netzwerk hat den Vorteil, dass alle Clients (Laptop, PC und Set-Top-Box) mit dem gleichen Datenbestand arbeiten. Die Speicherung auf dem PC macht nicht viel Sinn, da dieser dann immer an sein muss, wenn man Musik hören will. Die Netzwerkfestplatte ist immer an und bietet genug Platz für die Musik. Da die gemeinsamen Daten der Familie schon auf dem NAS gespeichert und verwaltet werden, ist also keine Umgewöhnung notwendig. Trotzdem soll es aber möglich sein, Daten, die auf einem externen Medium (CD, DVD oder USB-Medium) gespeichert sind, wiederzugeben. Die Medien werden bei Bedarf in die Set-Top-Box eingelegt bzw. angeschlossen und, nach der Wiedergabe wieder entfernt.

Da der Musikbestand noch nicht auf den NAS übertragen wurden, muss das noch passieren. Dazu wird die Musik von der CD aus über den PC auf das NAS gespielt. Das Einspielen der Musik über den PC bietet folgende Qualitätsmerkmale:

- Software-Ergonomie: Ein großer Datenbestand lässt sich einfacher über einen PC (Monitor, Maus und Tastatur) verwalten, als über die begrenzten Interaktionsmöglichkeiten der Set-Top-Box.
- Benutzerschnittstelle: Zum Verwalten des Datenbestandes kann auf bekannte Software zurückgegriffen werden. So kann der Anwender z. B. den Dateibrowser des Betriebssystems oder eine Musikverwaltungssoftware benutzen.
- Erweiterbarkeit: Falls die Musik später über einen anderen Weg (z. B. Internetshops) gekauft wird, muss diese Musik nur aus einem passenden Format auf dem NAS gespeichert werden.

3.3.3. Wiedergabeliste

Wiedergabelisten sind von den tragbaren MP3-Playern wie dem iPod von Apple bekannt. Beim iPod wird diese Wiedergabeliste mit Hilfe des Tools iTunes am PC erstellt und zusammen dann mit den Musikstücken auf das tragbare Gerät übermittelt. So ähnlich könnte es

dann auch bei der Set-Top-Box aussehen. Über eine Software (z. B. Webinterface) am Bürocomputer wird dann die Wiedergabeliste erstellt und verwaltet. Wenn die Musikdaten im Netzwerk gespeichert werden sollen, stellt sich die Frage, wo die Wiedergabelisten gespeichert werden. Wiedergabeobjekte sind unabhängig von der Set-Top-Box. Die Wiedergabelisten abhängig von der Sicht auf das Netzwerk, die Netzwerkfreigabe (Laufwerksbuchstaben) können z. B. am PC andere Name haben. Es macht also Sinn, die Wiedergabelisten lokal auf der Set-Top-Box zu speichern, während die Musik zentral auf dem NAS gespeichert wird. Die Speicherung der Listen auf dem NAS in einem Standardformat hätte den Vorteil, dass die Listen auch anderer Wiedergabesoftware zu Verfügung stehen. Der Anwender sollte bei der Konfiguration des Gerätes selber wählen, wo er die Listen speichern will. Die Wahl des Speicherortes der Wiedergabeliste ist dem Anwender vom Mobiltelefon bekannt, dort kann man wählen, ob man die Telefonnummern auf der SIM-Karte oder im Telefon speichern will. Wenn man die Nummer auf der SIM-Karte speichert, stehen einem nur die Standard-Felder „Name“ und „Nummer“ zur Verfügung, während bei der Speicherung im Telefonspeicher wesentlich mehr Felder zur Verfügung stehen. Genauso ist es bei den Wiedergabelisten: bei der Speicherung im Standardformat auf dem Netzwerk können keine eigenen Meta-Informationen gespeichert werden. Wenn man überlegt, dass die Familie Müller mehr als 500 CDs hat, könnte die Anzahl der Wiedergabelisten auch schnell auf mehrere 100 ansteigen. Es würde also Sinn machen, die Listen zu personalisieren oder in Kategorien aufzuteilen.

3.3.4. Zusammenfassung

In diesem Abschnitt wurde die Aufgaben in die Routinefunktionen und Komfortfunktionen unterteilt, um so dem Anwender eine einfache Interaktion mit dem Gerät zu ermöglichen. Die Routinefunktionen sind als Bedienelemente am Gerät zugänglich, während die Komfortfunktionen nur über das Netzwerk möglich sind. Da die angestrebte Lösung ein leichtgewichtiges Media-Center sein soll, das keinen eigenen Speicher für Medienobjekte hat, wird die Musik auf dem NAS gespeichert. Die Wiedergabelisten werden vom PC aus verwaltet, ob die Listen auf dem NAS oder Media-Center gespeichert werden sollen, bleibt dem Anwender überlassen.

3.4. Qualitätsmerkmale

Bei der Entwicklung von Software gibt es immer verschiedene Personen, die ihre Qualitätsmerkmale aus unterschiedlichen Sichtwinkeln formulieren. So gibt es bei der Softwareentwicklung die verschiedenen Anwender und das Entwickler-Team. Die beiden Gruppen haben ganz unterschiedliche Sichten auf das zu entwickelnde Produkt. So steht für einen Anwender

wohl die Funktionserfüllung, Robustheit und Ergonomie im Vordergrund, während der Entwickler neben diesen Punkten auch die Erweiterbarkeit, Wartbarkeit und die Portierbarkeit mit in den Vordergrund stellt. In dem folgendem Abschnitt soll nun kurz auf diese Qualitätsmerkmale eingegangen werden.

3.4.1. Qualitätsmerkmale aus Anwendersicht:

Aus Sicht des Anwenders muss ein System verschiedene Punkte erfüllen, damit er mit diesem System erfolgreich und effizient arbeiten kann.

Funktionserfüllung

Funktionserfüllung als Qualitätsmerkmal heißt, wie viele der geforderten Funktionen implementiert werden. Die Funktionen wurden in zwei Gruppen aufgeteilt (siehe 3.3.1), in die Routine- und Komfortfunktion. Damit der Anwender überhaupt mit dem System arbeiten kann, müssen die Routinefunktionen umgesetzt werden, während die Komfortfunktion dann in weiteren Ausbaustufen folgen können. Der Ausbau der Set-Top-Box verläuft wie folgt:

- Ausbaustufe 1:
 - Alle Routinefunktionen.
 - Wiedergabe von MP3-Codierter Musik
- Ausbaustufe 2:
 - Konfiguration der Set-Top-Box.
 - Anbindung von Netzwerkspeicher.
 - Erstellen und Löschen von Wiedergabelisten.
- Ausbaustufe 3:
 - Wiedergabe anderer Musikformate
 - Anbindungen von externen Speichermedien: CD, DVD und Festplatten.
 - Import und Export von Wiedergabelisten.
 - Import und Export der Konfiguration.
 - Wiedergabe von Internetradio.

Effizienz

Aus Sicht des Anwenders gibt es zwei wesentliche Messwerte für die Effizienz. Einmal möchte er eine reibungslose Wiedergabe der Musik, der andere Messwert ist die Reaktionszeit auf die Eingabe. Für die Wiedergabe muss sichergestellt werden, dass die Hardware genügend Performance hat, um die Musik zu dekomprimieren, und dass die Anbindungen an die Speichermedien schnell genug sind. Komprimierte Musik hat normalerweise eine Datenrate von 128 bis 256KBit/s. Unkomprimierte Musik (Audio-CD) hat eine Datenrate von 1200KBit/s. Wenn man davon ausgeht, dass der NAS direkt an dem Router angeschlossen ist und die Set-Top-Box über das W-Lan ebenfalls an den Router angeschlossen ist, dann stellt das W-Lan einen möglichen Engpass dar. Zum einen ist es langsamer als das Kabelnetz⁴ zum anderen teilen sich alle Clients die vorhandene Bandbreite. Im Haus der Familie Müller gibt es zwei W-Lan Clients: den Laptop und die Set-Top-Box die sich die Bandbreite teilen. Wenn z. B. jemand vom Laptop große Datenmengen auf den PC kopiert, könnte die Bandbreite nicht mehr ausreichen. Abhilfe könnte ein Router schaffen, der eine bestimmte Bandbreite für eine Dienst (traffic shaping⁵) garantiert, oder die Unterstützung von QoS⁶.

Zuverlässigkeit und Robustheit

Die Zuverlässigkeit schließt die Robustheit mit ein. Robust ist ein System, wenn es alle Fehleingaben erkennt und dann mit einer geeigneten Fehlermeldung zurückweist (vgl. (Raasch, 1992, S. 24)). Es dürfen nur Eingaben verarbeitet werden, bei denen kein Fehler gefunden wurden. Zuverlässigkeit heißt, dass das System die geforderten Funktionen immer wieder fehlerfrei erfüllt.

Bei der Wiedergabe von Medieninhalten kann es immer mal wieder zu Problem kommen. So kann es zu Beispiel vorkommen, dass ein Wiedergabeobjekt ausgewählt wurde, dessen Format der Set-Top-Box unbekannt ist, oder dass das ausgewählte Objekt nicht gefunden werden kann, weil es inzwischen gelöscht wurde. Da die Set-Top-Box die Musikdaten aus dem Netzwerk lädt, kann es auch vorkommen, dass diese Verbindung abreißt.

Ärgerlich wird es, wenn der Benutzer eine umfangreiche Wiedergabelist erstellt hat, dann etwas beim Speichern fehlschlägt geht und er die ganze Arbeit noch mal machen muss. Die Set-Top-Box muss robust gegenüber solchen Fehlern sein, da sie sonst für den Anwender wertlos ist. Wenn ein Fehler auftritt, muss die Software dafür sorgen, dass das Gerät in

⁴Angenommen wird ein 100-Base-TX Verkabelung (100MBit/s) (Tanenbaum, 2000), und ein IEEE 802.11g W-Lan (54MBit/s)(Sikora, 2001).

⁵Traffic shaping heißt, dass ein Gerät den Datenfluss so beeinflusst, dass es für bestimmte Dienste ein garantierte Untergrenze gibt.

⁶Quality of Service (QoS) stellt den Güte für einen Kommunikationsdienst sicher. Dabei werden beim Aufbau der Verbindungen eine garantierte Güte ausgehandelt (Bandbreite, Latenz usw.).

sicheren Zustand zurückkehrt, die Daten nicht verloren gehen, und den Anwender über das Auftreten des Fehlers informiert. So könnte die Software bei einem unbekanntem Format den Anwender informieren und dann mit der Wiedergabe des nächsten Objektes fortfahren. Problematisch wird das Verfahren, wenn die Verbindung zum Medium abgerissen ist und die Wiedergabeliste auf Wiederholung steht. Dann würde immer wieder eine Fehlermeldung ausgegeben. Eine solche endlos Wiederholung muss verhindert werden.

Software-Ergonomie

Im Abschnitt 3.3.1 wurde die Benutzerschnittstelle in zwei Bereiche aufgeteilt. Die Routinefunktionen sind über die Taster oder Fernbedienung am Gerät steuerbar, die Komfortfunktionen sind über das Netzwerk, z. B. als Weboberfläche, erreichbar. Die Steuerung soll einfach und intuitiv sein, ein abschreckendes Beispiel soll hier der Videorekorder sein: Die Wiedergabefunktionen eines Videorekorders kann jeder bedienen, aber an der Programmierung des Videorekorders scheitern viele. Das liegt daran, dass sie diese Funktion selten nutzen und schnell vergessen, wie es geht. Das Problem liegt hier beim Gerät: es unterstützt den Anwender bei dieser Komfortfunktion zu wenig.

Anforderungen an die Gerätesteuerung

Die Gerätesteuerung soll so einfach wie möglich sein. Dem Anwender soll ermöglicht werden, in die Wiedergabe einzugreifen, eine Wiedergabeliste auszuwählen oder direkt ein Musikstück auszuwählen. Dazu müssen passende Geräte zur Interaktion gefunden werden. Bei der Steuerung bieten sich eine Fernbedienung und Taster am Gerät an. Zur Darstellung der Statusinformationen und des Auswahlmenüs wird die Set-Top-Box mit einem Display ausgerüstet. Das Display muss groß genug sein, um neben der Wiedergabezeit auch noch Metainformationen wie z. B. Titel und Künstler anzuzeigen.

Kleine Flüssigkristallbildschirme (LC-Display), die an einen Computer angeschlossen werden können und Informationen über den Status des Computer darstellen, findet man immer häufiger. Gerade Server, Router oder Firewalls, die nicht direkt an einen Monitor angeschlossen sind, nutzen solche Displays für Statusanzeigen. Bei Desktop PCs werden diese Displays meistens dazu benutzt, um Informationen über die derzeit wiedergegebene Musik darzustellen, damit das Wiedergabeprogramm nicht immer sichtbar auf dem Monitor laufen muss, oder man die Informationen bei abgeschaltetem Monitor ablesen kann. Das LC-Display soll während der Wiedergabe die Metainformationen anzeigen. Damit das Display nicht mit Informationen überladen wird, werden während der Wiedergabe nur folgende Informationen dargestellt:

- Zustand der Set-Top-Box (play, pause, stop, repeat, shuffle)

- Name des Interpreten
- Name des Albums
- Name des Musikstücks
- Wiedergabeposition und Restzeit

Da an der Set-Top-Box auch die Auswahl einer Wiedergabeliste, oder eines oder mehrerer Musikstücke aus dem Verzeichnisbaum möglich sein soll, muss es eine Menüsteuerung geben. Die Steuerung soll über Taster und die Fernbedienung zugänglich sein, und der Inhalt des Menüs wird auf dem Display dargestellt. Auf eine Funktion für die Lautstärkeregelung wird verzichtet, da das Gerät eine Set-Top-Box für die Hifi-Anlage werden soll. Normalerweise wird dann die Lautstärke über den Verstärker geregelt. Denkbar wäre es, eine Lautstärkeregelung über die Fernbedienung anzubieten, aber dann sollte man die Lautstärke des Verstärkers steuern (lernfähige Fernbedienung). Da die Musik aus unterschiedlichen Quellen kommt, wäre eine Normalisierung bei der Wiedergabe sinnvoll. Über diese Funktion kann sichergestellt werden, dass die Musik mit dem gleichen Lautstärkeniveau wiedergegeben wird und so der Anwender nicht immer nachregeln muss.

Anforderungen an die Netzwerksteuerung

Die Komfortfunktion ist über das Netzwerk zu erreichen. Sinnvoll wäre es, hier nicht nur die Komfortfunktionen, sondern auch die Routinefunktionen über das Netzwerk anzubieten, dann könnte man jedes Gerät (PC, Laptop oder PDA) auch als Fernbedienung benutzen. Es gibt mehrere Möglichkeiten, die Funktionen über das Netzwerk (siehe Abbildung 3.5) zugänglich zu machen. Am sinnvollsten wäre es, die Funktionen über einen Webserver (Abbildung 3.5 b) zugänglich zu machen, dann könnte man von einem beliebigen PC aus über einen Webbrowser auf die Komfortfunktionen zugreifen. Eine andere Möglichkeit wäre die Steuerung über ein Programm vom Client PC (Abbildung 3.5 c) aus. Egal wie die Steuerung realisiert wird, aus Sicht der Software-Ergonomie einige Punkte müssen berücksichtigt werden.

- Die Benutzerschnittstelle muss übersichtlich und freundlich sein, damit sich auch ungeübte Anwender leicht zurechtfinden.
- Die Benutzerschnittstelle muss ein einheitliches Aussehen haben.
- Die Benutzerschnittstelle muss große Fehlertoleranz gegenüber Eingabe- und Bedienfehlern haben (siehe auch Abschnitt 3.4.1).
- Die Benutzerschnittstelle sollte über eine online Hilfe verfügen, um den Benutzer bei Fragen unterstützen zu können.

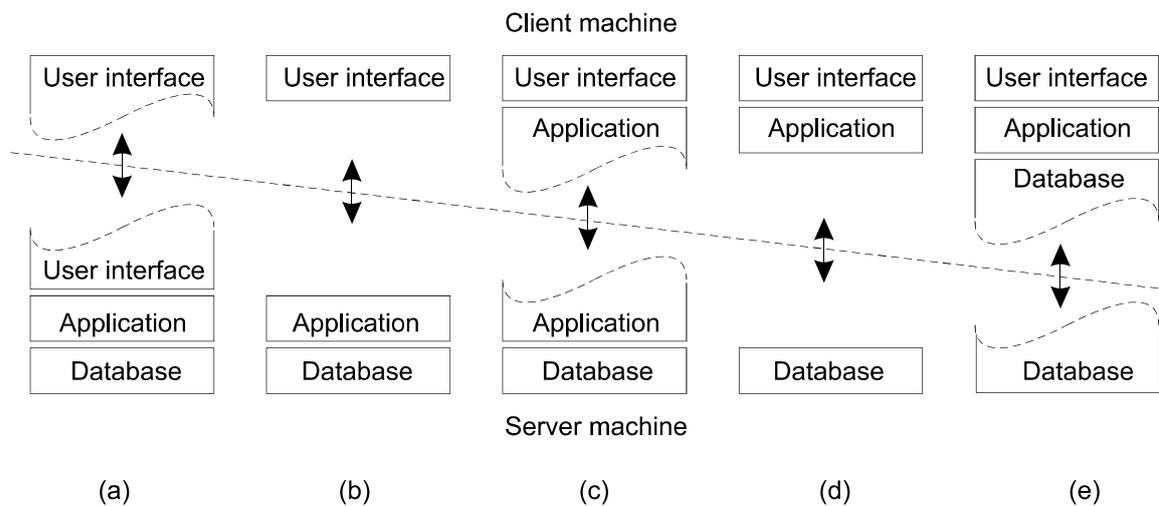


Abbildung 3.5.: Alternative Client-Server-Anordnungen (Quelle:(Tanenbaum und van Steen, 2002))

Da die Familie Müller schon beim Einrichten des NAS oder Routers mit der Steuerung über das Netzwerk mit Hilfe eines Webbrowsers vertraut ist, wäre es sinnvoll, sich bei der Entwicklung an einem solchen Webinterface zu orientieren. Die Konfrontation mit bekannten Metaphern soll dem Anwender die Angst vor einem neuen System nehmen, die er beim ersten Kontakt zu überwinden hat, also wäre es sinnvoll die Komfortfunktionen auch über ein Webinterface bereitzustellen.

Sicherheit

Allgemein versteht man unter dem Begriff Sicherheit nur, dass man das System vor dem Zugriff durch unbefugte Personen schützt. Aber um ein System sicher zu machen, braucht man mehr als nur eine Zugriffskontrolle (vgl. (Raasch, 1992, S.29)). Da es sich hier um eine Set-Top-Box für das private Wohnzimmer handelt, wird auf eine Benutzerverwaltung verzichtet, nur die Konfigurationsmasken (administrativen Aufgaben) werden durch eine Passwortprüfung geschützt. Verhindert werden muss aber, dass das System in einen gefährlichen Zustand gelangt sowie die Verfälschung des Programmes, des Systems oder der Einstellungen durch unbefugte Personen. Die Datensicherheit soll durch Sicherungs- und Wiederherstellungsfunktionen sichergestellt werden.

Da die Set-Top-Box über Funk auf das Heimnetz zugreift, wird eine sichere Verbindung vorausgesetzt. Informationen über die Sicherheit im W-Lan findet man unter (BSI, 2006).

3.4.2. Qualitätsmerkmale aus Entwicklersicht

Ein Entwickler sollte neben denen des vorangegangenen Abschnittes, auch die folgenden Qualitätsmerkmale bei der Entwicklung im Auge haben (vgl. (Raasch, 1992, S. 30-37)):

- Erweiterbarkeit: Das System sollte so entwickelt werden, dass es sich auch später noch möglichst leicht erweitern lässt, um so auch die zukünftigen Ansprüche der Anwender umsetzen zu können.
- Wartbarkeit: In einem laufenden System sollte ein Fehler schnell gefunden und behoben werden können.
- Übertragbarkeit, Portabilität: Das System sollte leicht auf eine andere Hardware oder auf ein anderes Betriebssystem portiert werden können.
- Wiederverwendbarkeit: Teile des System sollten in einem anderen System wieder verwendbar sein.

Um die o.g. Merkmal sicherzustellen, sollte bei der Entwicklung der Software auch darauf geachtet werden, dass die Software in einzelne Module aufgeteilt werden und dann über festgelegte Schnittstellen miteinander agieren. Mit der Benutzung von Design Patterns (siehe (Gamma u. a., 2002) und (Buschmann u. a., 2000)) sollen die Module voneinander entkoppelt werden und so die o.g. Merkmale sicherstellen.

3.5. Zusammenfassung

In diesem Kapitel wurden die Anforderungen an die Set-Top-Box aus dem Szenario (Kapitel 2) abgeleitet (Abschnitt 3.2). Erst wurde untersucht, ob man die Anforderungen der Familie Müller mit der Hilfe von bestehenden Lösungen umsetzen könnte (Abschnitt 3.2). Leider gibt es keine Lösung, die man direkt in das Wohnzimmer der Familie Müller stellen kann, so dass eine passende Lösung entwickelt werden muss. Dazu wurde das Szenario genauer analysiert und festgelegt, wie die Set-Top-Box in das bestehende Netzwerk integriert werden kann. Die Interaktion wurde in zwei Bereiche aufgeteilt. Die Wiedergabesteuerung erfolgt an der Set-Top-Box selbst, während die Einrichtung und die Erstellung der Wiedergabelisten über das Netzwerk erfolgen. So soll ein besserer Komfort für den Anwender sichergestellt werden, da die Interaktionsmöglichkeiten für die Komfortfunktionen (siehe 3.3.1) vom PC besser sind als die Interaktion über Taster. Da sich im Netzwerk schon ein Speicher (NAS) vorhanden ist, wurde festgelegt, dass die Musik dort gespeichert und der Set-Top-Box über das W-Lan bereitgestellt wird (siehe Abschnitt 3.3.2). Zum Abschluss des Kapitel wurde dann nochmal auf die Qualitätsmerkmale (Abschnitt 3.4) eingegangen. An diesen Merkmalen lässt sich bewerten, wie gut ein System wirklich ist und ob man damit arbeiten kann. Diese Merkmale wurden

unter dem Gesichtspunkt der im Wohnzimmer stehenden Set-Top-Box beleuchtet und durchgesprochen. Es wurde festgelegt, in welcher Reihenfolge die Funktionen umgesetzt werden (Abschnitt 3.4.1), in wie die Interaktion am Gerät und über das Netzwerk (Abschnitt 3.4.1 und 3.4.1) aussehen muss, und wie ein Weiterentwicklung (siehe 3.4.2) sichergestellt werden kann.

4. Design

Die gewonnenen Erkenntnisse aus der Analyse (Kapitel 3) sollen in diesem Kapitel weiter konkretisiert werden.

In diesem Kapitel wird als erstes die Umgebung, in der die Set-Top-Box aufgestellt wird, festgelegt. Basierend auf der Umgebung und den Erkenntnissen aus der Analyse (siehe Abschnitt 3.3) und dem Szenario (siehe Abschnitt 2) wird in diesem Kapitel dann die Software- und Hardware-Architektur der Set-Top-Box entwickelt. Die Software-Architektur wird in ihre Komponenten zerlegt und beschrieben. Wie diese Komponenten zusammenspielen, wird mit Hilfe von Sequenzdiagrammen verdeutlicht.

4.1. Umgebung

Die Grundlage der hier besprochenen Umgebung soll eine Standard „*Small office, Home office*“ (kurz SoHo¹) Installation sein. Die beschriebene Umgebung soll als Basis für weitere Betrachtungen dienen und die Umgebung festlegen, die im Haus der Familie Müller eingesetzt wird. Die Abbildung 4.1 auf Seite 38 zeigt den vereinfachten Gesamtaufbau des im Szenario (siehe Abschnitt 2) geschilderten Netzwerks. Das hausinterne Netzwerk der Familie Müller besteht aus einem Bürocomputer, einem Laptop, einer Netzwerkfestplatte und einem Router. In dieses Netzwerk soll die Set-Top-Box integriert werden. Der PC und der Laptop dienen Herrn und Frau Müller als Arbeitscomputer für Office-Anwendungen und zum Surfen im Internet bzw. Schreiben von E-Mails. Die erstellten Dokumente werden nicht nur auf dem PC bzw. Laptop, sondern auch auf der Netzwerkfestplatte (kurz NAS) im Keller gespeichert. Auf dem NAS sollen dann neben den Dokumenten auch die Musikdateien der Familie gespeichert werden. Der PC ist über ein Netzkabel ans Netzwerk angeschlossen, während der Laptop über Funk mit dem Hausnetz verbunden ist. Der Router im Wohnzimmer stellt bei Bedarf die Verbindung zum Internet her. Er ist die Funkgegenstelle für Clients, die kabellos auf das Netzwerk auf Netzwerk bzw. Internet zugreifen wollen. Die Set-Top-Box wird im Wohnzimmer aufgestellt und an die HiFi-Anlage angeschlossen. Da im Wohnzimmer

¹Mit SoHo (Small office /Home office) klassifiziert man kostengünstige Hard- und Software, die für den Einsatz in kleinen Firmen und im Heimbüro eingesetzt wird. SoHo Produkte sind leicht einzurichten, da auf eine komplexe Konfiguration verzichtet wird.

keine Möglichkeit besteht, die Set-Top-Box permanent per Kabel ans Netzwerk anzuschließen, und Frau Müller nicht möchte, dass neue Kabel verlegt werden (siehe Abschnitt 2.1.2), wird sie über das vorhandene Funknetzwerk ans Hausnetz angeschlossen. Über diese Anbindung erfolgt dann der Zugriff auf die Wiedergabedaten. Die Wiedergabesteuerung der Set-Top-Box erfolgt über die Fernbedienung.

4.2. Softwarearchitektur

Die Anforderungen an die Software der Set-Top-Box lassen sich in die Bereiche Benutzerinterface, Anwendungslogik und Persistenz aufteilen. In den Bereich des Benutzerinterface fällt unter anderem die Anzeige der Statusinformationen der Set-Top-Box und die Interaktion mit dem Anwender. Die Analyse (siehe Abschnitt 3.4.1) zeigt, dass die Interaktion über ein Statusdisplay am Gerät in Verbindung mit einer Fernbedienung oder dem Bedienelement der Set-Top-Box stattfinden wird. Eine andere Möglichkeit der Interaktion ist die Steuerung über das Netzwerk. Es gibt also verschiedene Möglichkeiten, mit dem Gerät zu interagieren. Jede dieser Möglichkeiten besteht immer aus einer Anzeige- und einer Eingabekomponente. Um den Betrieb mehrerer Anzeige- und Eingabekomponenten zu ermöglichen, wird hier bei der Softwareimplementierung das Model View Controller Pattern (Buschmann u. a., 2000, S. 124) eingesetzt. Jedes Anzeigeelement ist ein View, zu dem ein Controller gehört. Der Controller kümmert sich um das zu der Anzeige gehörende Eingabeelement und benutzt die Dienste des Models, um die Anwendung zu steuern. Das Model ist der eigentliche Kern der Software, es beherbergt die Anwendungslogik und Daten (Statusinformationen), und zählt damit zur Anwendungslogik. Jeder View und Controller meldet sich am Model an, und wird über den *change-propagation mechanism* vom Model benachrichtigt, wenn sich die Daten geändert haben. Jeder View und Controller hat die Möglichkeit, seine Daten zu aktualisieren und sie dann dem Benutzer darzustellen. In den Persistenzbereich fällt (siehe Abschnitt 3.3.2 und 3.3.3) das Laden der Medienobjekte vom NAS und das Laden bzw. Speichern der Wiedergabelisten in den Speicher der Set-Top-Box. Aus dieser Aufteilung lässt sich nun eine Software-Architektur ableiten, die 3-Schichten-Architektur (engl. „*three tier architecture*“). Jeder o.g. Bereich entspricht einer Softwarekomponente. Die Abbildung 4.2 auf Seite 39 verdeutlicht diese 3-Schichten-Architektur.

- Das Benutzerinterface kapselt die Interaktion mit dem Benutzer. Sie beherbergt den View und den Controller, ist somit für die Repräsentation der Daten und der Benutzereingabe zuständig.
- Die Logikschicht beinhaltet die Verarbeitungslogik, die Wiedergabeeinheit (Player) und die Statusinformationen.

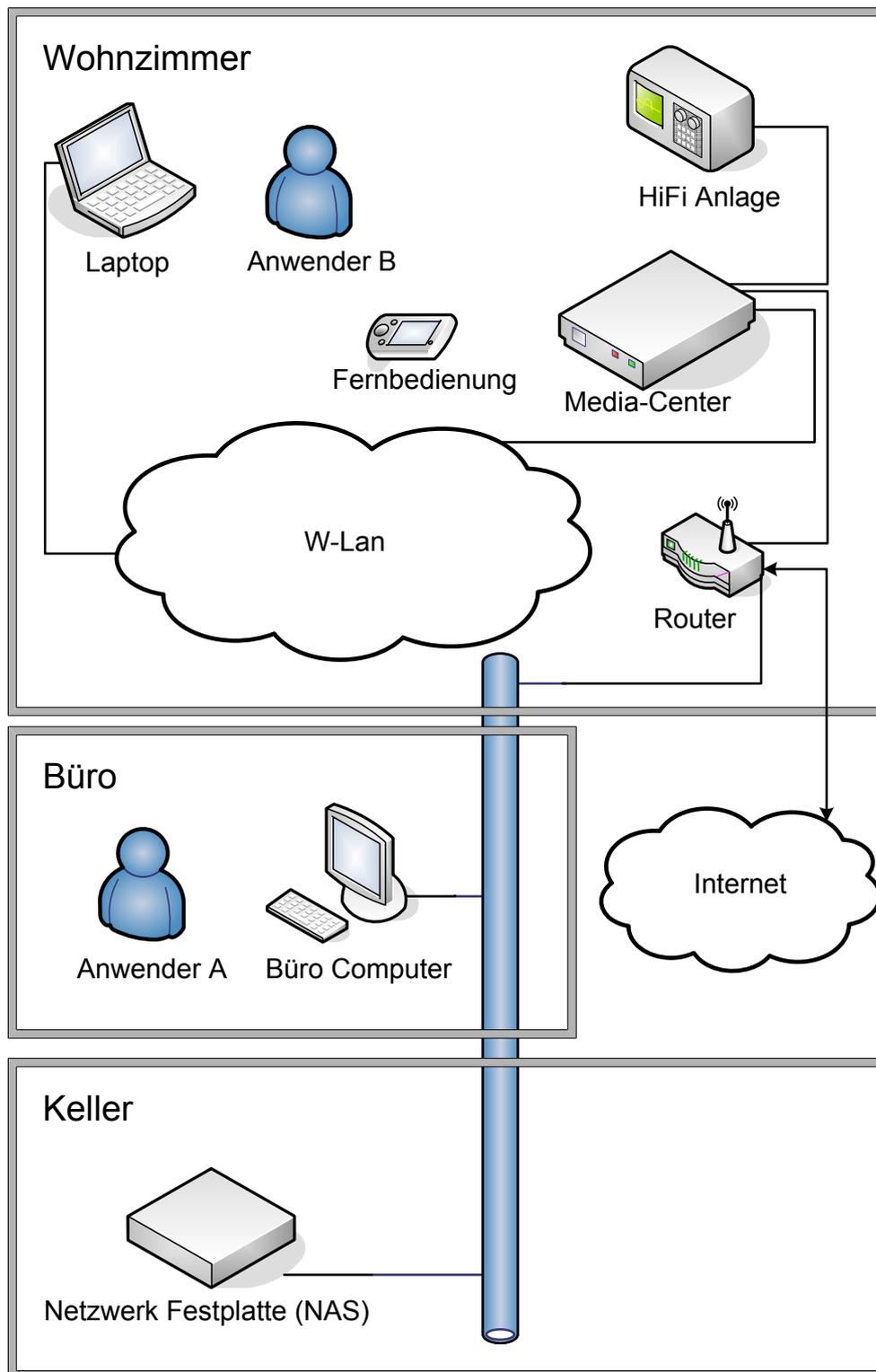


Abbildung 4.1.: Grobe Übersicht des Gesamtaufbaus

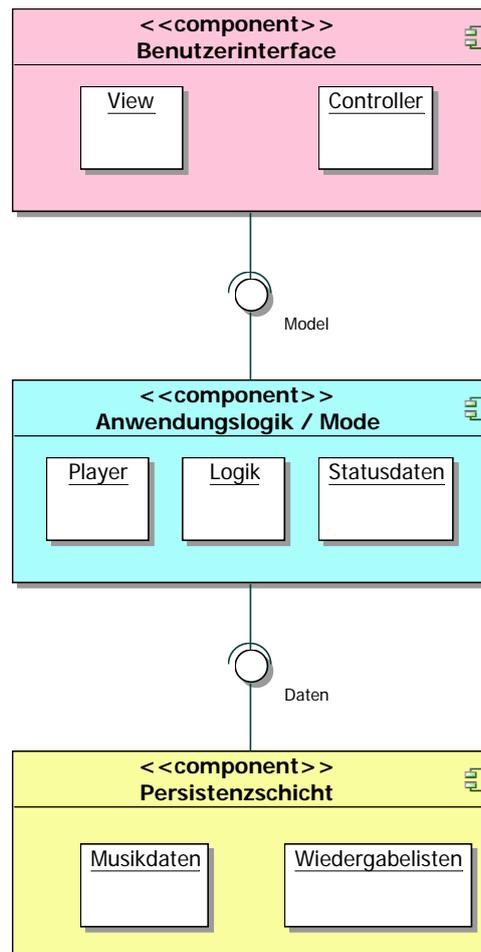


Abbildung 4.2.: Die 3-Schichten-Architektur

- Die Persistenzschicht ist für Speicherung und Laden der Daten (Wiedergabeobjekte und Wiedergabelisten) zuständig.

Jede Schicht bietet Dienste an, die von der darüber liegenden Schicht genutzt werden. Nach der Definition 4.4.1 (Kahlbrandt, 2001, S. 73) handelt es sich hier um eine geschlossene, vertikale Schichtung. Die Architektur ist *geschlossen*, weil die Schichten die Dienste der direkt unter ihr liegenden Schicht nutzen. *Offen* würde man dann sagen, wenn eine Schicht die Dienste aller darunter liegenden Schichten benutzt. Durch eine Schichten Architektur ergibt sich eine Struktur, die eine grobe Übersicht über die einzelnen Softwarekomponenten der Set-Top-Box ermöglicht. Die Unterteilung in Schichten bietet auch die Möglichkeit, eine Schicht durch eine erweiterte Version komplett auszutauschen. Dieser Vorteil könnte bei der späteren Weiterentwicklung wichtig werden, wenn die Set-Top-Box in Zukunft vielleicht neben Musik auch noch Bilder und Video wiedergeben soll.

4.3. Softwarekomponenten der Set-Top-Box

Um Software mit komplexen Anforderungen zu entwickeln, zerlegt man die Software in einzelne Komponenten. Diese Komponenten sind modulare Teile des Gesamtsystems, sie sind so aufgebaut, dass sie durch eine äquivalente Komponente ersetzt werden können. In der Softwareentwicklung spricht man von einer Softwarekomponente. Softwarekomponenten sind eine Ansammlung von Klassen, die Klassen stellen Schnittstellen bereit und benutzen Schnittstellen anderer Komponenten. Über die Schnittstellen ist es möglich, Komponenten mit anderen Komponenten in Beziehung zu setzen. Mit Hilfe der Beschreibungssprache UML kann man diese Beziehungen der Softwarekomponenten modellieren, und in einem Komponentendiagramm darstellen. UML wurde von der OMG² entwickelt und standardisiert. In diesem Abschnitt werden die einzelnen Komponenten der Set-Top-Box beschrieben und es wird erklärt, wie diese Komponenten realisiert werden sollen.

Wenn man also die in Kapitel 4.2 erläuterte 3-Schichten-Architektur in ein UML Diagramm überführt, zerfallen die einzelnen Schichten in Komponenten. So wird in der Analyse (siehe Abschnitt 3.3.1) zwischen den Interaktionsfunktionen (Routinefunktionen) und Konfigurationsfunktionen (Komfortfunktionen) unterschieden. Das heißt also, dass sich die Präsentationsschicht in zwei Komponenten aufteilen lässt, wobei sich das Interaktionselement wiederum in die Anzeige- und Bedienelemente aufteilen lässt. Diese Unterteilung findet deshalb statt, weil es auch möglich ist, dass ein Anzeigenelement ohne Bedienelement (Statusanzeige) vorkommt, oder umgekehrt ein Bedienelement ohne Anzeige (z. B. Fernbedienung). Bei

²OMG steht für Object Management Group. Die OMG ist ein Konsortium von derzeit über 800 Mitgliedern, das sich mit der Entwicklung von Standards für Objektorientierte Programmierung beschäftigt. Genauere Informationen findet man unter (OMG, 2005)

einem reinen Anzeigeelement wird bei der Implementierung auf einen Controller verzichtet, da es keine Benutzereingabe gibt. Durch den Wegfall des Controllers ergibt sich das Observer Pattern (Gamma u. a., 2002, S. 185). Bei den reinen Bedienelementen ist es ähnlich. Sie bestehen nur aus dem Controller, hier fällt der View weg, da ein reines Bedienelement die Daten des Models nicht braucht, sondern nur die Möglichkeit, Steuerbefehle abzugeben. Deshalb wird diese Schnittstelle mit dem Command Pattern realisiert.

Da das Interaktionselement aus einem Anzeigenelement und einem Bedienelement besteht, ergibt sich dafür das MVC Konzept. Sie haben einen View (die Anzeige) und einen Controller. Der Controller kann bei Änderung über die Informationsschnittstelle die Daten des Models abrufen und die Dienste des Models über die Bedienschnittstelle nutzen.

Die Logikschicht teilt sich in die Anwendungslogik der Set-Top-Box, die Wiedergabeliste, das Menü und die Steuerung der Wiedergabe auf. Die Daten beschafft die Persistenzschicht und speichert sie auch wieder. Daraus ergibt sich ein komponentenbasiertes System mit einem Kern, an dessen Schnittstellen sich die anderen Komponenten andocken. Die Aufspaltung in Komponenten bringt auch den Vorteil des Austauschens mit. Man kann einzelne Komponenten austauschen oder zwei Komponenten des gleichen Typs parallel betreiben, z. B. das Onbox Statusdisplaz und eine Anzeige an einem anderen PC. Diese Austausch- und Erweiterungsmöglichkeit ist einer der Gründe, warum ich mich für eine solche Architektur entschieden habe (siehe Abschnitt 3.4.2). Ein anderer Grund ist die damit verbundene lose Kopplung und das Information Hiding, die sich durch ein Komponentensystem ergeben. Die lose Kopplung ist in Bezug auf die Wiederverwendung von einzelnen Komponenten wichtig, denn wenn die Komponenten zu stark miteinander agieren und nicht über Schnittstellen arbeiten, ist es fast nicht möglich, sie aus dem System zu lösen und wiederzuverwenden oder sie auszutauschen. Also haben wir hier ein System mit einer Kernkomponente und Peripherie-Komponenten, die über Schnittstellen mit dem Kern interagieren. Daraus ergeben sich also folgende Schnittstellen für die Komponenten:

- Kernschnittstellen zum Benutzerinterface
 - Information: Bereitstellung der Statusinformationen
 - Befehl: Nimmt die Steuerkommandos an und gibt sie weiter
 - Konfiguration: Wird zu Konfiguration benutzt, z. B. Erstellen von Wiedergabelisten
- Kernschnittstellen innerhalb der Logikschicht
 - Menü: Enthält das Menü.
 - Wiedergabe: Steuerung einer externen Medienwiedergabesoftware.
 - Wiedergabeliste: Die Wiedergabeliste mit den Datenobjekt und den Metadaten.
- Kernschnittstellen zur Persistenzschicht

- Daten: Beschafft die Medienobjekte.
- Datenspeicherung: Speicherung von aufgenommenen Daten (z. B. Wiedergabelisten).

Daraus ergibt sich dann die in Abbildung 4.3 auf der nächsten Seite gezeigte Architektur. Neben dem Kern und seinen Schnittstellen zeigt sie auch mögliche Komponenten, die auf diese Schnittstellen zurückgreifen.

4.3.1. Die Kernkomponente (Core)

Die Kernkomponente (Core) ist das Herzstück der Set-Top-Box. Es verknüpft die anderen Komponenten zu einer Einheit. Die Kernkomponente stellt verschiedene Schnittstellen zur Verfügung, an denen die anderen Komponenten andocken können. Sie hält die Statusinformationen für die Anzeigeelemente bereit, reagiert auf die Kommandos und verwaltet die Wiedergabeliste und das Menü. Über das Menü lässt sich eine Wiedergabeliste oder ein Wiedergabeobjekt auswählen. In diesem Menü können Komponenten eigenen Einträge erzeugen oder einen vorhandenen Eintrag erweitern. Die Startparameter liest der Kern aus einer Konfigurationsdatei aus. Mit diesen Informationen werden dann die anderen Komponenten erzeugt oder mit ihnen Verbindung aufgenommen. Beim Erzeugen der Komponenten kommt das Builder Pattern (Gamma u. a., 2002, S. 97) zum Einsatz. Die Erbauerklasse selbst ruft zum Erstellen der Komponenten dann eine Factory Methode (Gamma u. a., 2002, S. 107) oder eine Abstrakte Fabrik (Abstract Factory Pattern (Gamma u. a., 2002, S. 87)) auf. Welches die Fabrikmethode oder Fabrikklasse ist, liest der Erbauer aus der Konfigurationsdatei. Damit wird sichergestellt, dass man nur die Konfigurationsdatei und nicht den Quellcode des Kerns ändern muss, um eine neue Komponente ins System zu integrieren. Die Tabelle 4.1 auf Seite 44 zeigt die vom Kern bereitgestellten Schnittstellen und die Beziehung. Bis auf die Wiedergabekomponente und die internen Kernkomponenten Menü und Wiedergabeliste, die nur einmal vorhanden sein dürfen, können sich an die anderen Schnittstellen beliebig viele Komponenten andocken.

Interaktionsschnittstellen

Zur Interaktion mit dem Benutzer gib es drei Schnittstellen:

1. Informationsschnittstelle: Sie ist die Schnittstelle für den View und Controller des Model View Controller Pattern. Sie stellt die Informationen bereit. Die bereitgestellten Informationen sind z. B.: Name, Größe, Länge... (siehe Abschnitt 3.4.1) oder Informationen über die Wiedergabelisten, also die Information, die der Anwender erwartet. Erwarten wird er die Informationen, die ihm Geräte wie ein CD-Spieler, DVD-Spieler

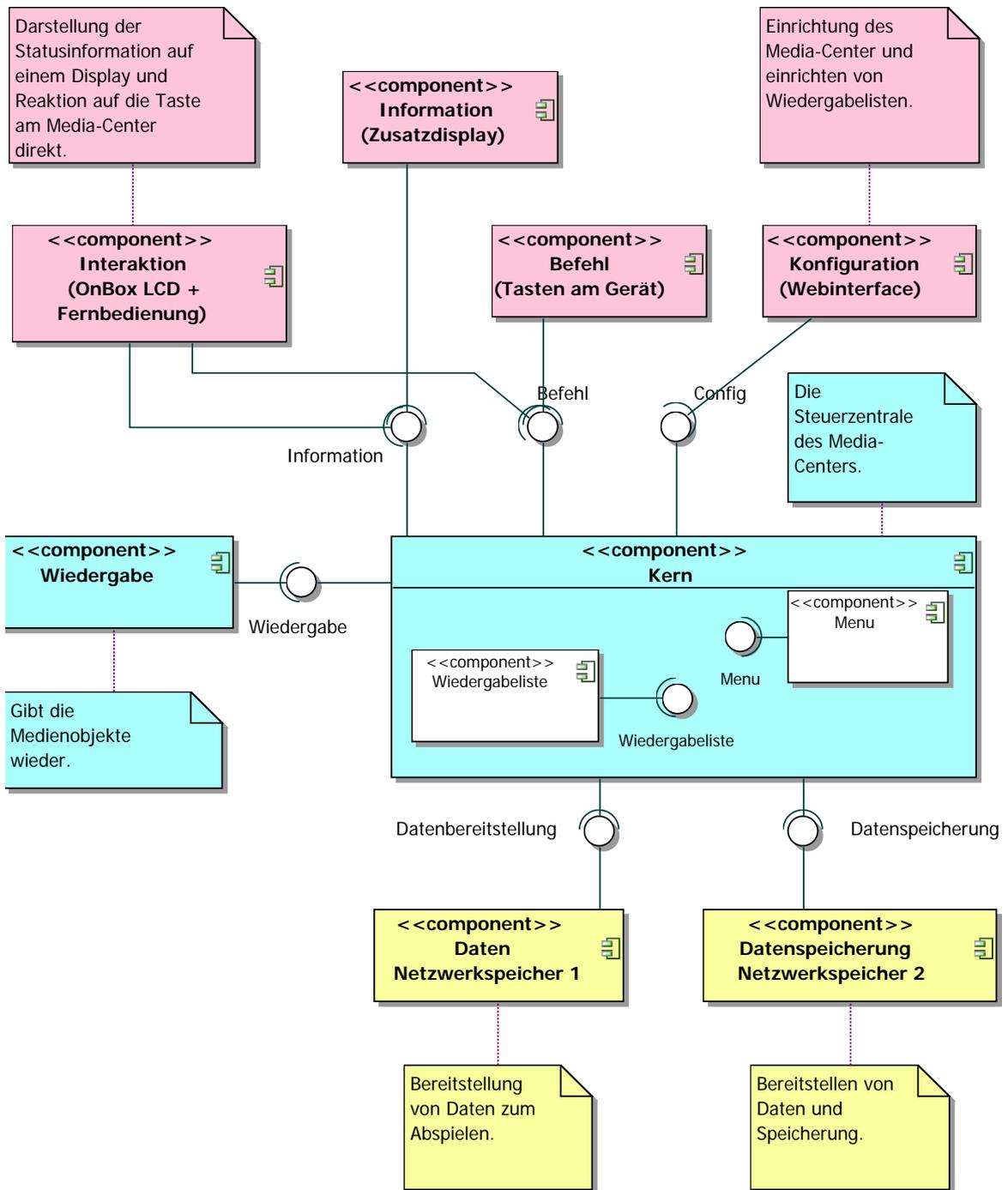


Abbildung 4.3.: Die Komponenten der Set-Top-Box

Schnittstelle	Bezeichnung	Richtung	Bemerkung
Information	0 zu N	Benutzerinterface	Statusinformationen für das LCD und andere Statusanzeigen
Befehl	1 zu N	Benutzerinterface	Entgegennehmen der Userbefehle
Konfiguration	0 zu N	Benutzerinterface	Zur Konfiguration der Kernkomponente
Wiedergabe	1 zu 1	Anwendungslogik	Steuerung der Wiedergabe
Wiedergabeliste	1 zu 1	Anwendungslogik	Beinhaltet eine List mit den Wiedergabeobjekten
Menü	1 zu 1	Anwendungslogik	Beinhaltet das Menü
Daten	1 zu N	Persistenz	Bereitstellung der Medienobjekte
Datenspeicherung	1 zu N	Persistenz	Erzeugung und Speicherung von Medienobjekten

Tabelle 4.1.: Übersicht über die Schnittstellen der Kernkomponente

oder ein iPod liefern. Bei einer Änderung der Statusinformationen werden alle registrierten Informationskomponenten über ein entsprechendes Ereignis benachrichtigt. Hier kommt bei der Implementierung das Observer Pattern (Gamma u. a., 2002, S. 185) zum Einsatz. Der Kern ist dabei das zu beobachtende Objekt (Subject bzw. Model), und die Anzeigekomponenten sind die Beobachter (Observer bzw. View und Controller).

2. Befehlsschnittstelle: Über diese Schnittstelle werden die Befehle des Benutzers entgegen genommen. Die Befehle teilen sich in zwei Gruppen auf: einmal die Steuerbefehle für die Wiedergabe, und Menü-Befehle. Sie werden an die Wiedergabekomponente weiter geleitet. Die Steuerbefehle werden z. B. von Tastern am Gerät ausgelöst und sie ermöglichen die Gerätesteuerung (Routinefunktion 3.3.1), ähnlich wie bei einem CD-Spieler. Die Menü-Befehle stehen dann über eine erweiterte Steuerungseinheit wie z. B. Fernbedienung bereit, vergleichbar mit der Aufnahmeprogrammierung eines Videorecorders. Diese Befehle werden an die interne Menükomponente im Kern weitergeleitet. Es gibt also eine Schnittstelle für die Steuerbefehle und ein Schnittstelle, die diese um die Menüfunktion erweitert. Hier werde ich mich bei der Implementierung an dem Command Pattern (Gamma u. a., 2002, S. 233) orientieren. Die Steuerbefehle werden an die Wiedergabekomponente weitergegeben, dafür wird das Adapter Pattern (Gamma u. a., 2002, S. 139) benutzt.

3. Konfigurationsschnittstelle: Die Schnittstelle Konfiguration wird benutzt, um die Set-Top-Box zu konfigurieren. Über die Schnittstelle werden neue Datenverzeichnisse ins System aufgenommen, Änderungen an den Parametern des Kerns vorgenommen oder Wiedergabelisten erstellt und bearbeitet. Da alle Funktionen schon im Kern vorhanden sind, aber in verschiedenen Schnittstellen, wird hier auf das Facade Pattern (Gamma u. a., 2002, S. 185) zurückgegriffen, um die Funktionen in einer Schnittstelle zu vereinen. Mit dieser Fassadenklasse soll der Zugriff auf den Kern aus Sicht der Konfigurationskomponente vereinfacht werden, denn die Komponente muss nur eine Schnittstelle des Kerns benutzen.

Wiedergabeschnittstelle

Def.: Eine Wiedergabekomponente beinhaltet eine Abspiel-Software, diese Software wird im folgende **Player** genannt.

Mit Hilfe der Wiedergabe-Schnittstelle wird mit den eigentlichen Wiedergabekomponenten interagiert. Über diese Schnittstelle werden die Steuerbefehle an den Player verschickt und die Informationen über die Wiedergabe empfangen. Die empfangenen Informationen werden vom Kern aufgearbeitet, und dann über die Informationsschnittstelle der Anzeigekomponente bereitgestellt. Zu diesen Informationen zählt z.B. die aktuelle Wiedergabeposition, Name des Stückes und Name des Interpreten. Da sich diese Informationen jederzeit ändern können, müsste die Kernkomponente immer wieder nachfragen. Um dieses „aktive Warten“ zu vermeiden, wird hier auf das Observer Pattern (Gamma u. a., 2002, S. 293) zurückgegriffen. Der Kern ist hier der Beobachter (Observer), der den Player (Subject) beobachtet. Da der Kern die Wiedergabeliste verwaltet, kann die Wiedergabekomponente über diese Schnittstelle immer das aktuelle Wiedergabeobjekt abholen.

Wiedergabelisteschnittstelle

Die Wiedergabeliste ist eine vom Anwender bestimmte Auswahl von Objekten, die der Reihe nach wiedergegeben werden (siehe Abschnitt 3.3). Der Aufbau soll sich an bekannten Metaphern orientieren, wie z.B. die Wiedergabeliste des Windows Media Player oder die Wiedergabeliste des iPods von Apple. Die Liste soll Funktionen wie zufällige Wiedergabe oder eine Wiederholung der Liste ermöglichen. Dazu wird hier das Iterator Pattern (Gamma u. a., 2002, S. 257) in Kombination mit dem State Pattern (Gamma u. a., 2002, S. 305) benutzt. Für die Wiedergabeliste werden vier verschiedene Iteratoren implementiert:

- Normal: Normale Wiedergabe, immer der Reihe nach bis zum Ende der Liste.
- Zufall: Zufällige Wiedergabe, bis jedes Objekt einmal wiedergegeben wurde.

- Wiederholung: Die Liste wird immer wiederholt.
- Zufall mit Wiederholung: Zufällige Wiedergabe aller Objekte mit mehrfacher Wiedergabe.

Diese vier Iteratoren sind die vier verschiedenen Zustände der State Klasse. Der interne Zustand ändert sich, wenn der Benutzer den Wiedergabemodus ändert, so wird die Anfrage fürs nächste Objekt immer an den passenden Iterator weitergeleitet. Neben den statischen Wiedergabelisten wäre noch eine dynamische Wiedergabeliste denkbar, also Wiedergabelisten, die am der Set-Top-Box direkt erstellt werden können. Diese Wiedergabeliste baut der Anwender mit der Zeit auf: wenn ihm z. B. das laufende Musikstück gefällt, drückt er auf der Fernbedienung die Taste „*add to Favorite*“ und nimmt so das Musikstück in die dynamische Wiedergabeliste auf. So ähnlich ist die Funktion „*On-The-Go*“ (Apple Computer, 2006b, S. 17) vom iPod. Sie ermöglicht es, unterwegs ohne PC eine Wiedergabelisten zu erstellen und später in die gespeicherten Listen zu überführen. Um die Funktion zufällige Wiedergabe zu verbessern, könnte man ein Bewertungssystem für die Wiedergabeliste einbauen, so könnte der Anwender die Musikstücke, die ihm gefallen, öfter hören als die, nicht so gut bewerteten. Ein solches Bewertungssystem wird in der Software WinAmp oder im iPod eingesetzt. Damit der Anwender nicht jedes Objekt einzeln bewerten muss, könnte man ihn bei der Bewertung unterstützen, indem man das Verhalten des Benutzers analysiert. So würde z. B. ein Song im Ranking³ steigen, wenn der Benutzer die Lautstärke erhöht. Negative Auswirkungen auf das Ranking wäre zum Beispiel das Überspringen mit der Taste *next*. Die Objekte in der Liste haben Metainformationen die über das Abstract Factory Pattern erzeugt werden. Beim Laden der Liste wird der Pfad des Objektes an diese Fabrik übergeben welche dann die Metadaten erzeugt. Der Grund dafür ist die Vielzahl von verschiedenen Wiedergabeobjekten, die alle verschiedenen System nutzen um die Information zu speichern. Wenn man ein neues Format für Informationen hinzufügen will, muss man die Fabrik erweitern.

Menüschnittstelle

Über die Menü-Schnittstelle soll es den Komponenten ermöglicht werden, das Auswahlmenü zu erweitern. So können hier die Datenbereitstellungskomponenten die auf ihnen gespeicherten Wiedergabelisten dem Menü hinzufügen oder das Menü um die Navigation durch ihren Speicher erweitern. Das Menü selbst wird im Composite Pattern (Gamma u. a., 2002, S. 163) realisiert. Die Menüeinträge werden als ein Baum von Objekten angelegt, jeder Eintrag ist entweder ein Untermenüpunkt oder ein Blatt im Baum. Die Navigation findet mit den folgenden Befehlen statt:

- *right*: nächster Eintrag auf der aktuellen Ebene.

³Ranking engl.: Einordnung in eine Rangliste.

- left: vorausgehender Eintrag auf der aktuellen Ebene.
- enter: Der Befehl Enter hat zwei Funktionen, je nachdem ob er auf einem Blatt oder Knoten ausgeführt wird.
 - Blatt: Befehl ausführen.
 - Knoten: eine Ebene tiefer.
- back: eine Ebene höher

Die Ebenen werden als Ring realisiert, um die Navigation zu erleichtern. Die Abbildung 4.4 zeigt einen Beispielbaum (vereinfacht) des Menüs. In der ersten Ebene befinden sich nur die zwei Knoten *Playlists* und *Content*. Der Knoten *Playlists* beherbergt schon erstellte Wiedergabelisten (*PL1 bis PL3*), und der Knoten *Content* bietet eine Navigation durch die verfügbaren Musikstücke. Die Musikstücke sind hier in folgender Reihenfolge sortiert: Interpret, Album, Titel Nr. auf dem Album. Ob das Menü nach der Auswahl einer *Playlist* oder Auswahl eines *Tracks* beendet wird oder nur ein anderer Knoten angewählt wird, legt das ausgeführte Befehlsobjekt fest. Der Anwender wird sicherlich mehrere Musikstücke auswählen wollen (temporäre Wiedergabeliste), bevor er die Wiedergabe startet. Die Befehle im Menü werden mit Hilfe des Command Pattern (Gamma u. a., 2002, S. 233) implementiert.

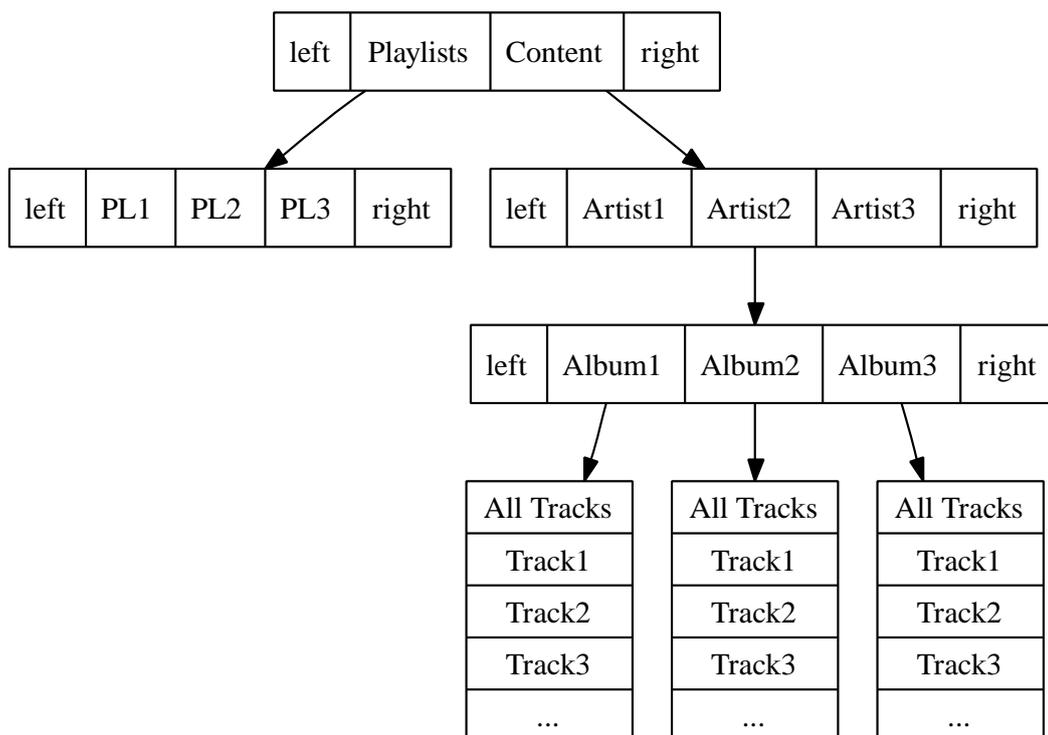


Abbildung 4.4.: Ein Beispielbaum für das Menü

Daten- und Datenspeicherungsschnittstelle

Die Daten- und die Datenspeicherungsschnittstelle sind die Schnittstellen zur Persistenzschicht. Über diese Schnittstellen beschafft sich der Kern die für die Wiedergabe benötigten Daten (z. B. Musik und Wiedergabeliste). Komponenten, die diese Schnittstellen implementieren, kann man als Adapter verstehen, die externe Datenquellen an die Set-Top-Box anschließen. Man könnte die Basisfunktion auch in den Kern integrieren, aber im Blick auf die Vielzahl von Datenquellen und die verschiedenen Protokolle wie z. B. interne Speicher, externe Speicher, Netzwerkspeicher, Streams usw. macht es mehr Sinn, diese Funktionen in einzelne Komponenten zu zerlegen und dann nur die benötigten zu nutzen. So ist es später leichter, noch weitere Datenprotokolle in die Software der Set-Top-Box zu integrieren. Über die Schnittstelle Daten lässt sich ein Objekt einlesen. Sie ist auf einen „*read only*“ Zugriff ausgelegt. Die Datenspeicherungsschnittstelle erweitert die Datenschnittstelle um die Funktion „*Schreiben*“. Über diese Schnittstelle soll es z. B. möglich sein, eine neue Wiedergabeliste zu speichern. Wo die Daten konkret gespeichert werden, bestimmt die Komponente selbst, zunächst wird davon ausgegangen, dass nur Wiedergabelisten gespeichert werden. Der Inhalt einer Liste ist pfadabhängig, so dass die Listen erstmal nur im Speicher der Set-Top-Box abgelegt werden (siehe Abschnitt 3.3.3).

4.3.2. Komponenten des Benutzerinterfaces

Es gibt vier verschiedene Komponenten, mit denen der Benutzer mit der Set-Top-Box interagiert.

1. Die Informationskomponente: Anzeige von Statusinformationen der Set-Top-Box.
2. Die Befehlskomponente: Steuerschnittstelle des Benutzers.
3. Die Interaktionskomponente: Zusammenfassung einer Informations- und Befehlskomponente.
4. Die Konfigurationskomponente: Administration der Set-Top-Box.

Die Aufteilung erfolgt, weil es eine Trennung zwischen Routine- und Komfortfunktionen gibt (siehe Abschnitt 3.3.1). Die Informations-, Befehls und Interaktionskomponente beinhalten die Routinefunktionen, und die Konfigurationskomponente enthält die Komfortfunktionen.

Informationskomponente

Die Informationskomponente ist für die Darstellung der Statusinformation der Set-Top-Box zuständig. Die Informationsschnittstelle der Kernkomponente stellt alle benötigten Informationen, wie z. B. den Titel des Musikstückes oder die aktuelle Position bereit. Eine Informationskomponente meldet sich bei der Schnittstelle an und wird dann bei jeder Änderung der Daten benachrichtigt (Observer Pattern). Dann holt sie die benötigten Informationen und stellt sie dem Benutzer zur Verfügung. Um die Informationen dem Benutzer darzustellen, sind verschiedene Möglichkeiten denkbar:

- onBox (ein kleines LCD)
- remote (z. B. als Weboberfläche)

Da die Informationsschnittstelle der Kernkomponente eine 1 zu N Beziehung ermöglicht, können auch mehrere Informationskomponenten gleichzeitig betrieben werden. Damit soll ermöglicht werden, dass von verschiedenen Orten Statusinformationen abrufbar sind.

Befehlskomponente

Die Befehlskomponente ist für die Interaktion mit dem Anwender zuständig. Sie nimmt die Befehle/Kommandos vom Anwender entgegen. Für die Interaktion mit dem Anwender gibt es verschiedene Szenarios, z. B. eine Interaktion über das Bedienfeld am Gerät, Fernbedienung oder über das Netzwerk (Weboberfläche oder Steuerungssoftware) von einem anderen Gerät. Jede dieser Möglichkeiten wäre eine eigenständige Komponente, die die Befehlsschnittstelle der Kernkomponente benutzt. Wie bei der Informationsschnittstelle ist auch hier eine 1 zu N Beziehung möglich, damit man die Set-Top-Box über mehrere Komponenten steuern kann.

Interaktionskomponente

Die Interaktionskomponente ist ein Zusammenschluss aus einer Informations- und einer Befehlskomponente, die zusammen als Einheit agieren. Die Zusammenfassung ist nötig, um eine komplexe Steuerung wie z. B. in Menüs zu ermöglichen. Das Menü soll zum Beispiel nur auf der Anzeige sichtbar sein, mit der der Anwender agiert. Hier kommt das Model View Controller Pattern (kurz MVC) (Buschmann u. a., 2000, S. 124) zum Einsatz. Jede Informationskomponente ist ein View, der einen eigenen Controller hat. Die Statusinformationen entsprechen dem Model und werden von der Kernkomponente verwaltet.

Konfigurationskomponente

Über die Konfigurationskomponente lassen sich Änderungen an den Einstellungen der Set-Top-Box vornehmen. Diese Änderungen teilen sich in zwei Bereiche auf: die Administration der Set-Top-Box und das Wiedergabelisten-Management. Dabei soll darauf geachtet werden, dass man die Anwender mit bekannten Paradigmen konfrontiert.

- Administrationskomponente: Über die Administrationskomponente soll es dem Benutzer ermöglicht werden, die Grundeinstellungen der Set-Top-Box zu verändern oder zu erweitern. Dazu zählt u. a. folgendes:
 - Einrichten des Netzwerkes (IP-Adresse).
 - Einrichten der W-Lan-Verschlüsselung.
 - Hinzufügen, Löschen und Ändern der Speicherorte von Musikdaten (siehe Abschnitt 3.3.2).
 - Speicherort und Format der Wiedergabelisten festlegen (siehe Abschnitt 3.3.3).
 - Komponentenmanagement: Konfigurieren und Löschen von anderen Komponenten.

Als Administrationskomponente würde sich hier ein Webinterface anbieten, das dem eines Routers nachempfunden ist. Aus Sicherheitsgründen (siehe Abschnitt 3.4.1) muss dieser Bereich mit einem Schutz gegen den Zugriff durch Unbefugten gesichert werden. Bei einem Webinterface ist ein Passwortschutz für die Administrationsmaske vorgesehen.

- Wiedergabelistenmanagement: Im Wiedergabelistenmanagement soll der Anwender der Möglichkeit haben, Wiedergabelisten zu bearbeiten, zu erstellen und zu löschen. Wie bei der Administrationskomponente soll der Anwender auch hier mit bekannten Paradigmen konfrontiert werden (siehe Abschnitt 3.3.3).

4.3.3. Wiedergabekomponente

Die Wiedergabekomponente gibt die Medienobjekte wieder. Die Kommunikation mit dem Kern erfolgt über die Wiedergabeschnittstelle. Der Kern löst über diese Schnittstelle Steuerereignisse wie Wiedergabe, Pause, Stop, nächstes Stück usw. aus. Diese Steuerbefehle werden vom Player verarbeitet. Wie der Player auf die Steuerbefehle reagiert, hängt vom aktuellen Zustand des Players ab. Aus dem Pause-Modus heraus muss er anders auf den Pause Befehl reagieren als aus dem Wiedergabe-Modus heraus. Im Wiedergabezustand würde der Player die Wiedergabe pausieren, während er im Zustand Pause die Wiedergabe fortsetzen

würde. Das Verhalten des Players auf die Befehle lässt sich in einem Zustandsdiagramm abbilden. Die Abbildung 4.5 auf der nächsten Seite zeigt das Zustandsdiagramm des Players. Der Player hat drei interne Zustände:

1. Play: In diesem Status werden Medieninhalte wiedergegeben. Wenn ein Medieninhalt zu Ende ist, wird der nächste aus der Wiedergabeliste wiedergegeben.
2. Pause: Die Wiedergabe wird pausiert.
3. Stop: Die Wiedergabe wird angehalten.

Jeder der Zustände muss auf die folgenden sechs Befehle reagieren:

1. play: Starten der Wiedergabe
2. pause: Wiedergabe pausiert, oder wird fortgesetzt.
3. playPause: Eine Kombination aus Play und Pause. Bedienelemente haben entweder einen PlayPause Taster oder einen Taster für Play und einen für Pause.
4. stop:Hält die die Wiedergabe an.
5. next: Schaltet die Wiedergabe ein Objekt weiter.
6. previous: Schaltet die Wiedergabe um ein Objekt zurück.

Um dieses Zustandsdiagramm in Software umzusetzen, habe ich mich für das State Pattern (Gamma u. a., 2002, 305) entschieden. Das State Pattern ermöglicht es einem Objekt, bei Änderungen des internen Zustandes, auch das Verhalten zu ändern. Um das Zustandsdiagramm in das State Pattern zu überführen, wird jeder Zustand zu einer eigenen Zustandsklasse und jeder Steuerbefehl zu einer Methode in diesen Zustandsklassen. Neben der Wiedergabe von Medieninhalten muss die Playerkomponente auch Informationen (siehe Abschnitt 3.4.1) des internen Zustandes und die Wiedergabezeit bereitstellen. Die anderen Metainformationen sind am Wiedergabeobjekt gespeichert (siehe Abschnitt 4.3.1). Wenn sich der Status des Players oder die Abspielposition ändert, schickt die Wiedergabekomponente eine Nachricht an den Observer (Kern), worauf dieser die neuen Daten beim Player abrufen. Da es bei Wiedergabeobjekten viele verschiedene Formate gibt und nicht jeder Player alle Formate unterstützen kann, wird der Player für ein Objekt über eine Fabrik (Abstract Factory Pattern) erzeugt. Damit soll sichergestellt werden, dass man nur die Fabrik erweitern muss, um die Software um neue Formate zu erweitern.

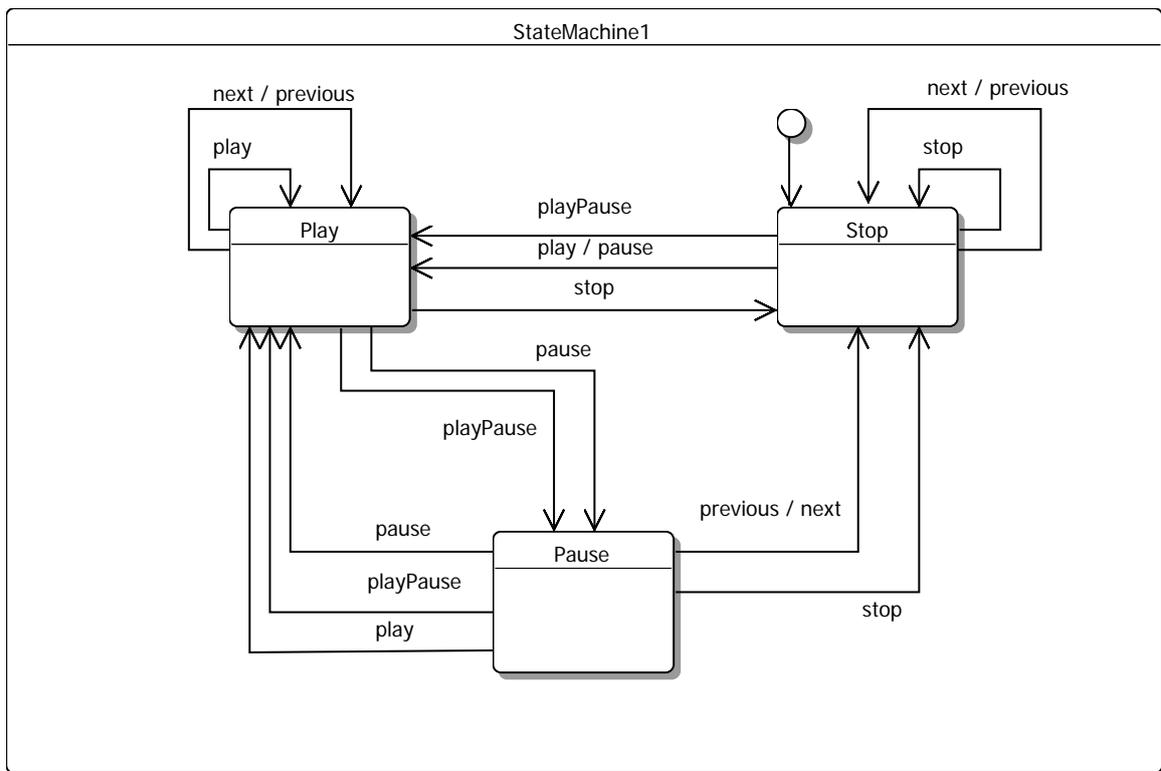


Abbildung 4.5.: Das Zustandsdiagramm des Players

4.3.4. Komponente der Persistenzschicht

In der Persistenzschicht gibt es die Daten- und die Datenspeicherungskomponente. Die Datenkomponente ist ein Adapter zwischen den gespeicherten Objekten und der Set-Top-Box. Sie stellt den Zugriff auf Medienobjekte und Wiedergabelisten bereit, die von der Set-Top-Box wiedergegeben werden. Diese Komponente ist für die Bereitstellung der Inhalte verantwortlich. Zu der Bereitstellung gehört auch die Navigation durch die Multimediainhalte, die Suche nach bestimmten Inhalten, die Filterung nach verschiedenen Auswahlkriterien sowie der hierarchische Aufbau der Medienobjekte. Eine Datenkomponente macht dazu die passenden Einträge im Menü der Set-Top-Box. Die eigentlichen Medienobjekte werden als einfache Dateien von einem Speichermedium abgerufen. Dabei ist es nicht zwingend erforderlich, dass ein Speichermedium in die Set-Top-Box eingebaut ist, die Daten sind ja im Netzwerk gespeichert oder werden über externe Medien (USB-Festplatte, CD oder DVD) der Set-Top-Box bereitgestellt (siehe Abschnitt 3.3.2). Die Set-Top-Box ist auch in der Lage, mit mehreren verschiedenen Datenbereitstellungskomponenten zu agieren. Die Datenspeicherungskomponente ist die Erweiterung der Datenkomponente, neben den Leseoperationen stellt sie die Möglichkeit bereit, neue Objekte zu erzeugen und zu speichern. Es soll möglich sein, Wiedergabelisten zu erstellen und dann zu speichern. Die Datenspeicherungskomponente legt die Daten dann auf einem beliebigen Speichermedium ab. Genauso wie bei der Datenkomponente muss der Speicher der Datenspeicherungskomponente nicht fest in die Set-Top-Box eingebaut sein, der Anwender muss dann bei der Konfiguration den Speicherort der Listen auswählen (siehe Abschnitt 4.3.2).

4.4. Hardware Architektur

Im Haus der Familie Müller befinden sich schon Komponenten, die man für die Set-Top-Box nutzbar machen kann. Es gibt ein W-Lan, einen Netzwerkspeicher und einen Computer. Das W-Lan bietet die Möglichkeit, die Set-Top-Box ans Netzwerk anzuschließen ohne Kabel zu verlegen, was eine der Anforderungen war (siehe Abschnitt 2.1.2). Es gibt zwei Wege der Interaktion (siehe Abschnitt 3.3.1) mit dem Gerät: direkte Steuerung am Gerät (Routinefunktionen) und die Steuerung über das Netzwerk von einem anderen PC (Komfortfunktionen). Da laut Analyse (siehe Abschnitt 3.4.1) auf herkömmliche Eingabe- und Anzeigergeräte verzichtet werden soll, müssen andere Elemente zur Interaktion gefunden werden. Eine Basissteuerung, wie man sie von CD-Spielern kennt, soll am Gerät durch Taster ermöglicht werden. Diese Steuerung soll durch eine Fernbedienung ergänzt werden. Die Metainformationen sollen auf einem LC-Display dargestellt werden (siehe Abschnitt 3.4.1). Die erhältlichen LCD Modelle teilen sich in zwei Gruppen auf: die grafischen Displays und die alphanumerischen Modelle. Bei den grafischen Displays lässt sich jeder Punkt auf dem Display einzeln ein- oder ausschalten. Die Ansteuerung erfolgt über eine Matrix der Bildpunkte, jeder Punkt lässt sich

über eine xy-Koordinate schalten. Text, der auf dem Display dargestellt werden soll, muss erst in Grafiken umgerechnet werden und kann dann dargestellt werden. Die alphanumerischen Displays haben einen festen Vorrat von darstellbaren Buchstaben und Symbolen. Da der Speicher der Display begrenzt ist haben sie normalerweise nur einen kleinen Zeichenvorrat (weniger als 256 Zeichen). Aus beiden Gruppen gibt es Displays mit Schnittstellen, die ein direktes Ansteuern vom PC aus ermöglichen. Da an der Set-Top-Box erst Einstellungen für die Anbindung an das W-Lan vorgenommen werden müssen (Name der Funkzelle und Verschlüsselung), wird das Gerät für diese Grundeinstellungen einmalig über das kabelgebundene Netzwerk angeschlossen. Die Wiedergabelisten und Musikstücke werden auf der Netzwerkfestplatte abgelegt.

4.5. Sequenzdiagramm

Anhand von vier Sequenzdiagrammen soll das Zusammenspiel der Komponenten verdeutlicht werden. Als Grundlage für die Sequenzdiagramme dienen vier der typischen Benutzerinteraktionen mit der Set-Top-Box.

- Starten der Wiedergabe
- Erstellen einer Wiedergabeliste
- Laden einer Wiedergabeliste
- Einhängen eines neuen Datenverzeichnisses

In allen diesen Beispielen wird davon ausgegangen, dass die Set-Top-Box konfiguriert und betriebsbereit ist.

4.5.1. Starten der Wiedergabe

In diesem Beispiel soll die Wiedergabe über die Taster am Gerät die Wiedergabe gestartet werden. Das Sequenzdiagramm 4.6 auf Seite 56 zeigt den Ablauf.

1. Der Anwender drückt die Taste *Play* am Gerät. Die Interaktionskomponente nimmt den Befehl entgegen.
2. Die Interaktionskomponente erstellt aus dem Befehl ein Objekt und gibt es an die Kernkomponente weiter.
3. Der Kern gibt den Steuerungsbefehl an die Wiedergabekomponente weiter.

4. Da die Wiedergabekomponente kein Objekt zur Wiedergabe hat, muss sie eins laden und fragt am Kern nach dem aktuellen Objekt.
5. Der Kern holt sich aus der Playlist die URL des aktuellen Eintrages.
6. Der Kern holt mit der URL das Objekt aus der Datenkomponente (Persistenzschicht).
7. Ein Wiedergabe-Task wird über die Fabrik erzeugt und gestartet.
8. Der Wiedergabetask benachrichtigt den Kern über den Fortschritt der Wiedergabe.
9. Der Kern benachrichtigt seinen Beobachter über die Datenänderung
10. Die Befehls- und Informationskomponente wird benachrichtigt.
11. Die Befehls- und Informationskomponente holt sich die neuen Daten.
12. Die Daten werden aus der Wiedergabekomponente ausgelesen.

4.5.2. Erstellen einer Wiedergabeliste

Das Sequenzdiagramm 4.7 auf Seite 57 soll den Vorgang der Erstellung einer Wiedergabeliste erläutern. Der Anwender wählt im Webinterface den Menüpunkt *erstelle neue Liste* aus. Danach wählt er aus der Liste die gewünschten Musikstücke aus. Vorgesehen ist eine mehrstufige Auswahl, wie Interpret -> Album -> Musikstück. Da dieses Diagramm das Zusammenspiel der Komponenten zeigen soll, wird hier angenommen, dass alle Musikstücke auf der obersten Ebene untergebracht sind.

1. Der Anwender öffnet den Erstellungsdialog.
2. Die Konfigurationskomponente lädt die Auswahl der Musik, die in der Set-Top-Box bekannt ist.
3. Da jede Datenkomponente ihren Inhalt ins Menü einträgt, wird dieser Teil des Menüs abgefragt und zurückgegeben.
4. Der Anwender wählt ein Objekt aus.
5. Das Objekt wird in die Liste aufgenommen.
6. Der Anwender speichert die Liste.
7. Die Liste wird zum Speichern an den Kern weiter gegeben.
8. Der Kern leitet die Liste an die ausgewählte Speicherkomponente weiter, die die Liste dann speichert.

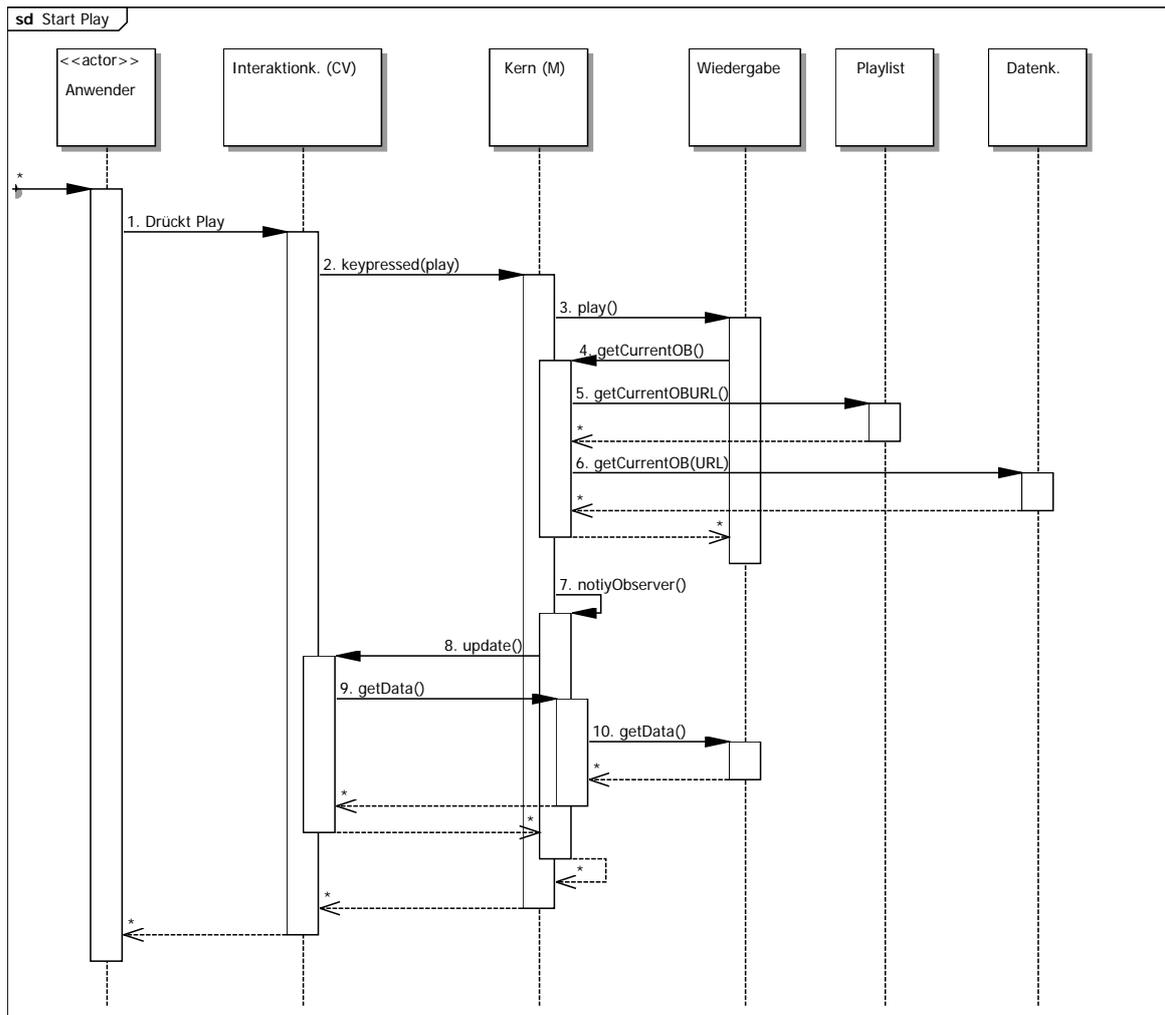


Abbildung 4.6.: Starten der Wiedergabe

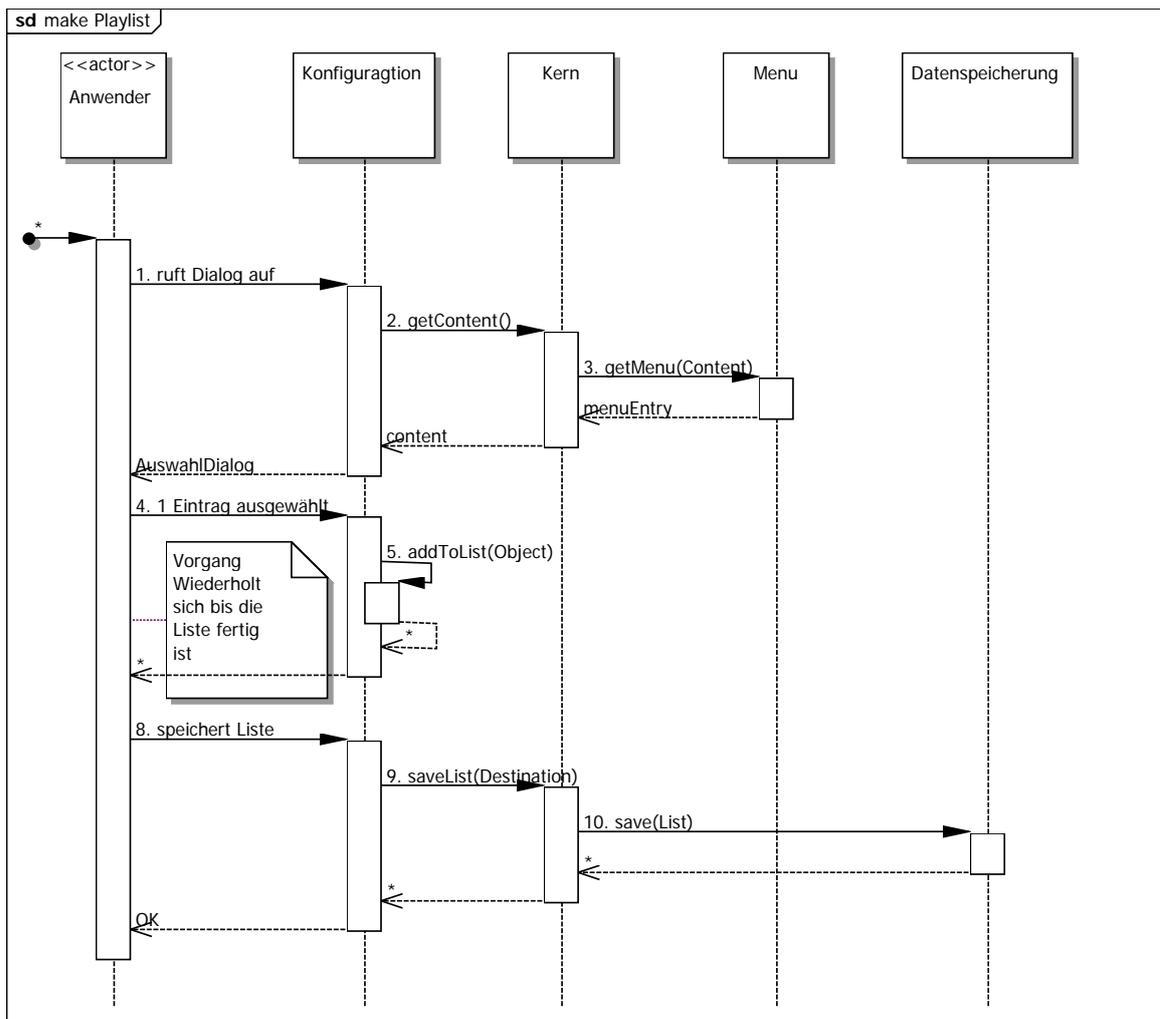


Abbildung 4.7.: Eine Wiedergabeliste wird erstellt

4.5.3. Laden einer Wiedergabeliste

Ein Anwender will eine vorher bestimmte Auswahl von Musik laden, die in der Set-Top-Box bereits als Wiedergabeliste vorhanden ist. Die Abbildung 4.8 zeigt den Ablauf. Für dieses Beispiel wird das Menü vereinfacht, es hat nur eine Ebene mit zwei Wiedergabelisten. Die gewünschte Wiedergabeliste ist der zweite Eintrag.

1. Der Anwender drückt die Taste *Menü* auf der Fernbedienung. Die Interaktionskomponente nimmt den Befehl entgegen.
2. Die Konfigurationskomponente ruft das aktuelle Menü vom Kern ab.
3. Der Kern leitet den Aufruf an die Menükomponente weiter.
4. Die Interaktionskomponente stellt die obersten Ebene des Menüs dar.
5. Der Anwender drückt die Taste *right* um die gewünschte Wiedergabeliste zu erreichen.
6. Der Befehl *right* wird an den gerade ausgewählten Menüeintrag gesendet.
7. Der nächste Eintrag wird ausgewählt.
8. Anwender drückt die Taste *enter*.
9. Der Befehl *enter* wird an den gerade ausgewählten Menüeintrag geschickt.
10. Der aktuelle Menüeintrag führt den Befehl aus. Es wird begonnen die Wiedergabeliste zu laden.
11. Dem Kern wird die ID der Wiedergabeliste übergeben.
12. Die Datei wird an die Wiedergabelistekomponente weiter gereicht, die aus den Datei die Wiedergabeliste aufbaut. Die Metadaten der Objekte werden über eine Fabrik erzeugt.
13. Der Kern benachrichtigt alle Observer über die Datenänderung (Wiedergabeliste wurde verändert)
14. Die Befehls- und Informationskomponente wird benachrichtigt.
15. Die Befehls- und Informationskomponente holt sich die neuen Daten.
16. Die Daten werden von der Wiedergabekomponente abgerufen.
17. Nach dem Laden der Wiedergabeliste wird das Menü beendet.

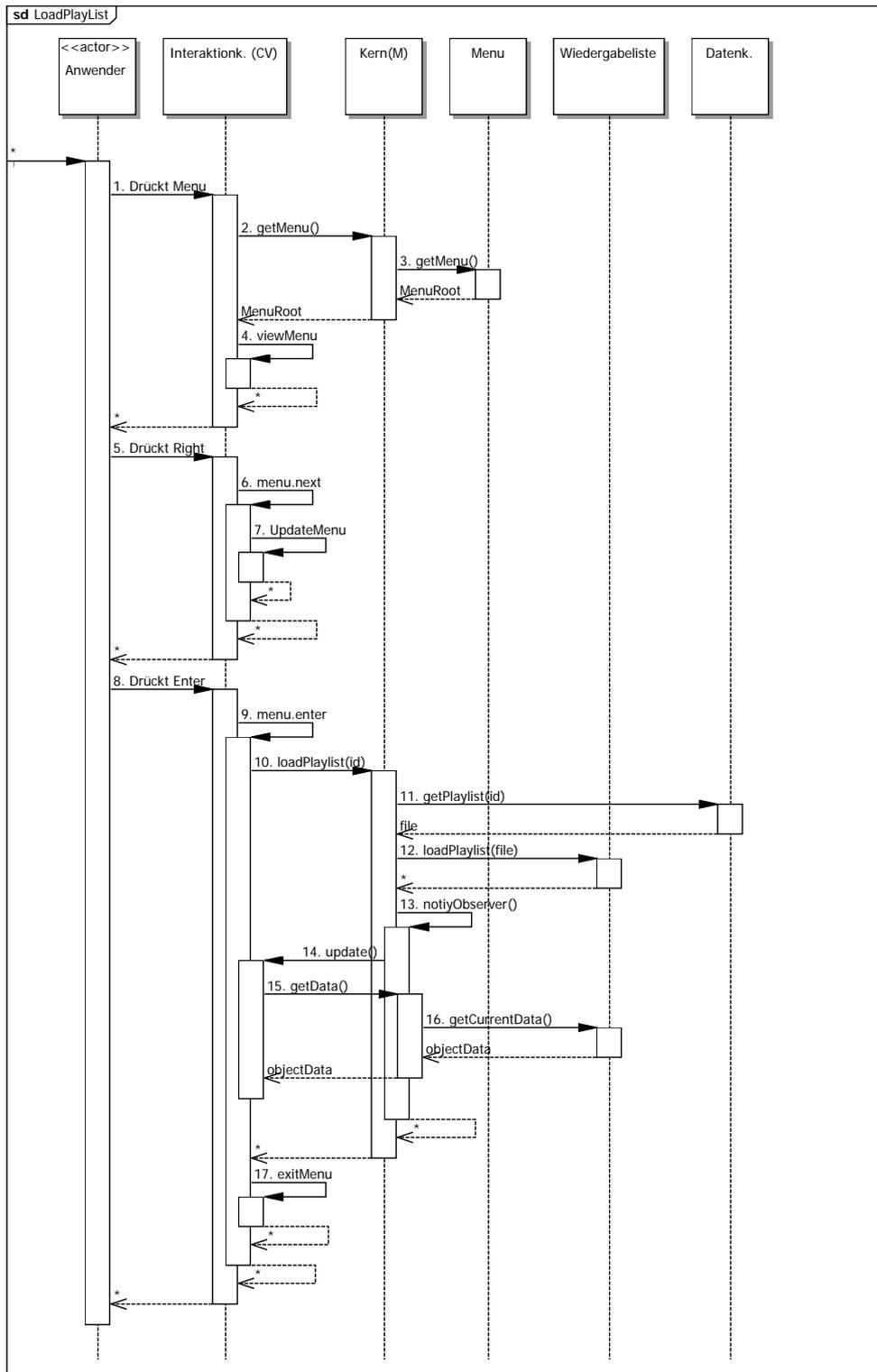


Abbildung 4.8.: Eine Wiedergabeliste wird über Menü geladen

4.5.4. Einhängen eines neuen Datenverzeichnisses

Das Sequenzdiagramm 4.9 auf der nächsten Seite soll das Zusammenspiel der Komponenten der Einbindung eines Datenverzeichnisses verdeutlichen. Das eigentliche Verzeichnis liegt auf einem Netzwerkspeicher und ist zum Lesen freigegeben. Um das Datenverzeichnis der Set-Top-Box bekannt zu machen, muss der Anwender über eine Konfigurationskomponente (z. B. ein Webinterface) eine neue Datenbereitstellungskomponente anlegen. Danach überprüft der Kern die Erreichbarkeit des Verzeichnisses. Wenn darauf zugegriffen werden kann, wird es in die Konfiguration übernommen, wenn nicht, wird dem Anwender eine Fehlermeldung gezeigt. Bei der erfolgreichen Erstellung wird die neue Komponente ins Menü übernommen.

1. Der Anwender öffnet den Dialog zum Einhängen des neuen Datenverzeichnisses.
2. Der Anwender führt den Dialog aus. Benötigte Informationen sind die Netzwerkadresse und der Pfad zu dem Verzeichnis.
3. Die Konfigurationskomponente setzt die beiden Informationen zu einer URL zusammen und übergibt diese an den Kern.
4. Der Kern erzeugt eine neue Datenkomponente und übergibt ihr die URL.
5. Der Kern lässt die URL von der Datenkomponente überprüfen.
6. Der Kern lässt das Verzeichnis auf Erreichbarkeit prüfen. Wenn es nicht lesbar ist wird eine *Exception* erzeugt, die den Vorgang abbricht und dem Anwender eine Fehlermeldung zeigt.
7. Wenn das Verzeichnis gelesen werden kann wird vom Kern die Initialisierung der Datenkomponente aufgerufen.
8. Die Datenkomponente erzeugt ihren Menüeintrag und gibt ihn an die Kernkomponente weiter.
9. Die Kernkomponente trägt den Menüeintrag ins Menü ein. Der Menüeintrag ist ein Baum, der dem aktuelle Menübaum hinzugefügt wird. Ist schon ein Knoten mit dem gleichen Namen vorhanden, so wird nur der Unterbaum zum Knoten zugefügt. So wird sichergestellt, dass es nicht zwei verschiedene Einträge mit dem gleichen Namen gibt (z. B. Wiedergabelisten).

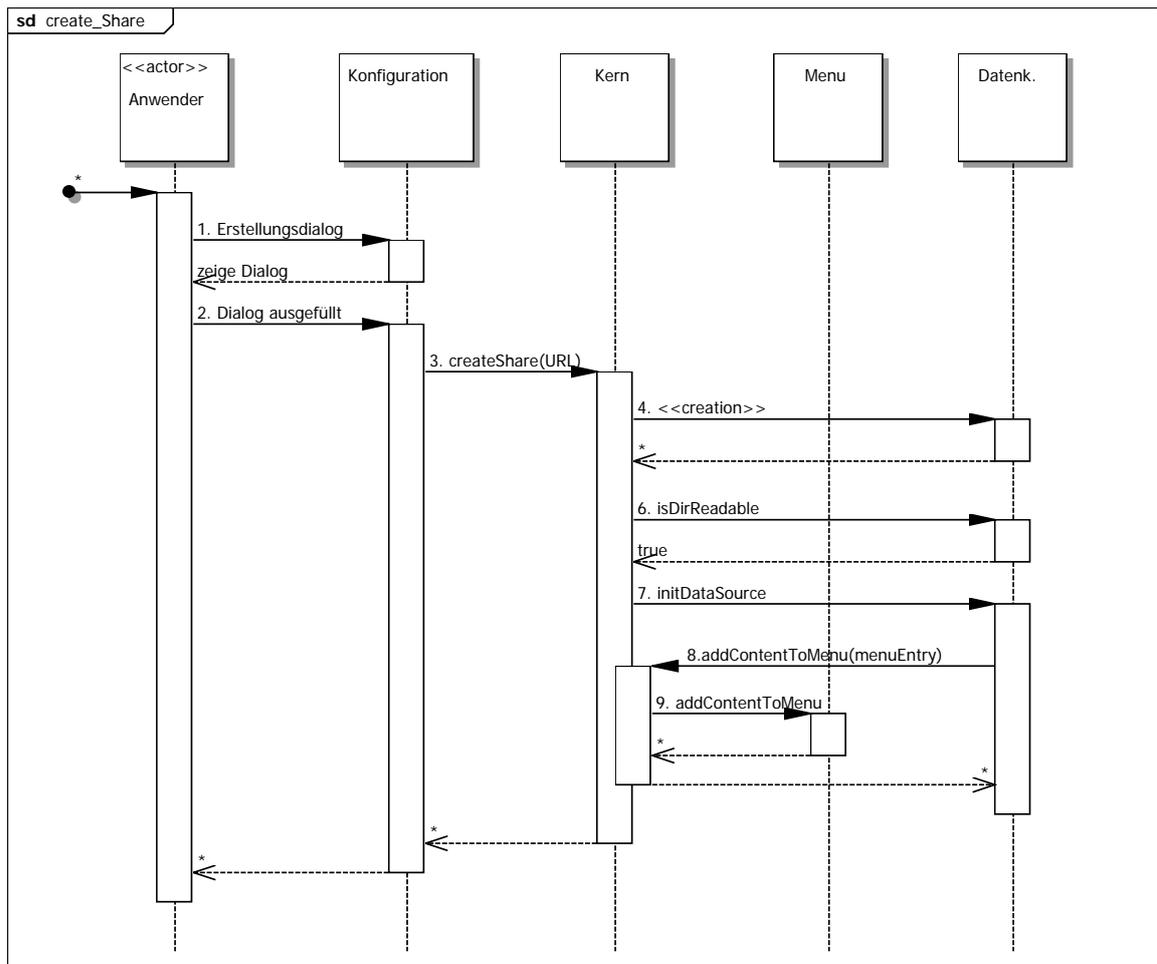


Abbildung 4.9.: Einhängen eines neuen Datenverzeichnisses

4.6. Zusammenfassung

Die Analyse (siehe Abschnitt 3.3) zeigt sehr komplexe Anforderungen an das System, dazu kommt der Wunsch nach einer späteren Erweiterung des Systems (siehe Abschnitt 3.4.2). Um diese Anforderungen zu erfüllen, wurde das System erst in die drei Schichten Benutzerinterface, Anwendungslogik und Persistenz aufgeteilt (vgl. Abbildung 4.2 auf Seite 39) und die Schichten dann in Komponenten zerlegt (vgl. Abbildung 4.3 auf Seite 43). Die Zerlegung des Systems folgt der Objektorientierten Zerlegungsstrategie (siehe (Raasch, 1992, S. 77)) auf der Basis von bekannten Design Patterns (Gamma u. a., 2002). Das daraus entstandene Design unterstützt Information Hiding und Polymorphie und wird so zu einem sehr flexiblen und anpassungsfähigen Design, das sich einfach um neue Komponenten erweitern lässt. Die Orientierung an bekannten Metaphern wie der Wiedergabeliste (iPod) oder dem Webinterface (Router) sollen dem Anwender die Interaktion mit dem Gerät erleichtern, und so die Akzeptanz fördern.

5. Realisierung

In diesem Kapitel soll der im Design (Kapitel 4) erdachte Entwurf umgesetzt werden. Um die Machbarkeit zu beurteilen, wird zuerst eine Machbarkeitsstudie durchgeführt. Mit dieser Studie soll herausgefunden werden, ob man die komplexe Steuerung des Gerätes auf eine Fernbedienung abbilden kann. Dazu wird zunächst eine geeignete Interaktionshardware entwickelt. Als Plattform für die Software dient ein handelsüblicher PC. Die Studie dient auch als Test für verschiedene Wiedergabekomponenten, die getestet und bewertet werden sollen. Mit der so gewonnenen Erfahrung aus der Machbarkeitstudie soll dann ein Prototyp entwickelt werden. Die Software für den Prototyp soll die in Abschnitt 4.2 genannte Architektur umfassen. Die in der Studie entwickelte Interaktionshardware soll dann, zusammen mit Standard PC Komponenten, in eine Set-Top-Box eingebaut werden. Bei der Auswahl der Hardwarekomponenten soll versucht werden, die in Abschnitt 4.4 gesetzten Anforderungen umzusetzen.

5.1. Machbarkeitsstudie

Mit der Machbarkeitsstudie sollte herausgefunden werden, ob und wie die Anforderungen aus der Analyse mit dem in Abschnitt 4 erdachten Design umsetzbar sind. Dabei wird das Hauptaugenmerk auf die Interaktion mit dem Benutzer gelegt. Die Studie soll diese Benutzerschnittstelle vorstellbar machen. Dazu wird ein Prototyp geschrieben, der folgende Routinefunktionen verwirklichen soll (siehe Abschnitt 3.4.1 Ausbaustufe 1):

- Interaktion über Taster und Fernbedienung
- Wiedergabe von Musik
- Anzeigen von Metainformationen.

Die genannten Punkte stellen die Kernpunkte der Set-Top-Box dar. In der Machbarkeitsstudie soll versucht werden, diese Punkte in Hard- und Software umzusetzen. Für die Studie muss erst eine passende Interaktionshardware gefunden werden. Im folgenden Abschnitt werden verschiedene Hardwarekomponenten auf ihre Tauglichkeit untersucht. Nachdem eine Hardwarekomponente ausgewählt wurde, wird die Softwareumgebung ausgewählt und beschrieben.

5.1.1. Hardware

Als Plattform für die Software dient ein normaler PC, an den das LCD, der Infrarotempfänger und die Taster angeschlossen werden. Da nur Textinformationen (siehe Abschnitt 4.4 und 3.4.1) dargestellt werden sollen, reicht ein alphanumerisches Display aus. Aus der großen Auswahl der Displays haben es drei genauer untersucht. Auswahlkriterien waren in erster Linie:

- Moderne Schnittstelle (USB)
- Mindestens 2 Zeilen mit je 16 Zeichen
- SDK für gängige Betriebssysteme wie z. B. Windows und Linux

Neben den hier formulierten Auswahlkriterien wäre es von Vorteil, wenn die Displays über eine Interaktionsmöglichkeit (Taster und Fernbedienung) verfügten, so dass dafür nicht noch eine externe Lösung gefunden werden muss.

USBLCD

Das USBLCD (Abbildung 5.1 und Abbildung 5.2) von Adams IT Services (Adams, 2005) besteht aus zwei Teilen: Schnittstellenmodul und Display. Das Schnittstellen Modul bietet die Möglichkeit, über USB ein Display von bis zu 80 Zeichen anzusteuern. Treiber gibt es für Linux und Windows. Der Windowstreiber ist eine Beta Version und trägt ein Datum von 2002. Leider gib es weder für Linux noch für Windows eine Beschreibung der API oder Programmierbeispiele.



Abbildung 5.1.: Das USBLCD
Quelle:(Adams, 2005)

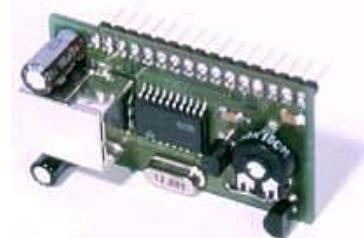


Abbildung 5.2.: Das USBLCD Schnittstellen Modul.
Quelle:(Adams, 2005)

CW1602

Das Cw1602 (Abbildung 5.3) von CwLinux (Cwlinux, 2005) ist ein LCD mit 2 Zeilen mit je 16 Zeichen. Das Modell hat eine USB Schnittstelle und verfügt über 6 Taster. Das Display ist für den Einbau in einen PC gedacht (5.25 Zoll, halbe Bauhöhe).



Abbildung 5.3.: Die Cw1602 von CwLinux. Quelle:(Cwlinux, 2005)

IO-Warrior24

Der IO-Warrior24 (Abbildung 5.4) von Code Mercenaries (Mercenaries, 2005) ist in erster Linie kein Display, sondern ein Ansteuerungs-Baustein. Der Baustein verfügt über eine USB Schnittstelle und über 16 Anschlüsse. Jeder dieser 16 Anschlüsse kann wahlweise Ein- oder Ausgang sein. Der IO-Warrior bietet die Möglichkeit zur Ansteuerung einer alphanumerischen LCD Moduls mit bis zu 80 Zeichen. Das IC bietet auch die Möglichkeit RC5-codierte¹ Infrarotsignale zu empfangen und zu decodieren. Wenn ein LCD-Modul und Infrarot-Empfänger an das IC angeschlossen sind, sind 3 Anschlüsse noch nicht beschaltet. Diese 3 Anschlüsse lassen sich dann als Eingänge nutzen, um Bedienelemente anzuschließen.

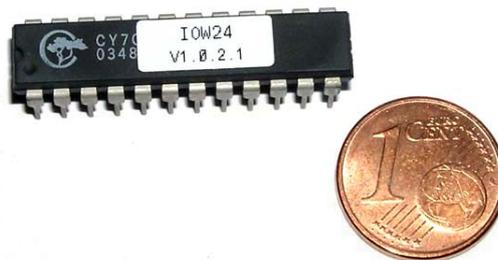


Abbildung 5.4.: Die IO-Warrior24 von Code Mercenaries

¹RC5 ist ein Standard zur Infrarot-Übertragung, der von Philips entwickelt wurde. Die übertragenen Daten ist 14 Bit lang, von denen 11 Bit Nutzdaten sind. Die Nutzdaten teilen sich in 5 Bit zum Adressieren, des Geräts und 6 Bit für die Codierung des Befehls.

Zusammenfassung

Das USBLCD bietet keine weiteren Features und die Treiber sind noch eine Beta Version. Das CW1602 hat zwar die Möglichkeit, Taster anzuschließen, aber das Display hat nur 2 Zeilen mit 16 Buchstaben, was ein wenig knapp bemessen ist, um die gesamten Metainformationen darzustellen. Da das USBLCD und das CW1602 Modul nicht ohne zusätzliche Hardware (Fernbedienung) auskommen, habe ich mich bei der Realisierung der Hardware für den IO-Warrior24 entschieden. Dieses IC vereint alle Funktionen, die hier gebraucht werden, und hat damit den Vorteil, dass man nur mit einem Software Development Kit (kurz SDK) arbeiten muss, was eine größere Auswahl bei dem Betriebssystem und der Programmiersprache garantiert, da man nicht auf die Schnittmenge von mehreren SDKs festgelegt ist. Nach dem Anschluss des Displays und des IR-Empfänger bleiben nur noch drei Eingänge frei, die aber nicht ausreichen, um die vier Steuertaster (Wiedergabe, Stopp, Vor und Zurück) direkt anzuschließen. Daher wird zwischen Taster und IO-Warrior eine Codierungseinheit (Decoder 8 auf 3) geschaltet, über die dann bis zu 7 Taster angeschlossen werden können. Die Taster werden an die 7 Eingänge des Decoder angeschlossen. Wenn ein Taster gedrückt wird, erzeugt der Decoder abhängig von der Wertigkeit des Eingangs einen Wert zwischen 1 und 7 (0 ist der Ruhezustand) und gibt ihn an den IO-Warrior (binärcodiert) weiter. Durch die Codierung ist es nicht möglich, zwei Taster gleichzeitig zu drücken. Daher wird ein Prioritäts-Decoder benutzt. Die Tasten werden für die Befehle Stopp, Wiedergabe/Pause, Vor, Zurück, Menü/Enter, Menü Links und Menü Rechts (sortiert nach Priorität) belegt. Die Steuerbefehle werden als Taster realisiert, während die Menübefehle über einen Drehimpulsgeber realisiert werden. Da der Infrarotempfänger in der Lage ist, RC5 codierte Infrarotsignale zu empfangen und zu decodieren, wird eine Universal-Fernbedienung benutzt. Damit diese mit dem Empfänger zusammenspielt, wird sie auf einen Philips Fernseher eingestellt. Der erste Prototyp der Ansteuerungsplatine wurde auf zwei Lochrasterplatinen aufgebaut (Abbildung 5.5). Auf der unteren Platine ist die Basisschaltung des IO-Warrior24 aufgebaut, auf der oberen die Codierungseinheit. Die Codierungseinheit ist hier ein 16 auf 4 Decoder, der aber nur als 8 auf 3 Decoder benutzt wird. Die Beschaltung des IO-Warrior24 wurde aus der Dokumentation der SDK entnommen. Der Schaltplan und das Platinenlayout befindet sich in Anhang B.

5.1.2. Software

Da die Anforderungen an die Software relativ klein gehalten wurden (3.4.1 Ausbaustufe 1), wurde in der Machbarkeitsstudie auf die Persistenzschicht ganz verzichtet, und es wird nur eine Interaktionskomponente (Display mit Fernbedienung) implementiert. Die Funktionen der Logikschicht sind auf ein Minimum eingeschränkt, so wird auf eine Wiedergabeliste und eine Menüsteuerung verzichtet. Als Betriebssystem wurde Windows XP gewählt, und als Pro-

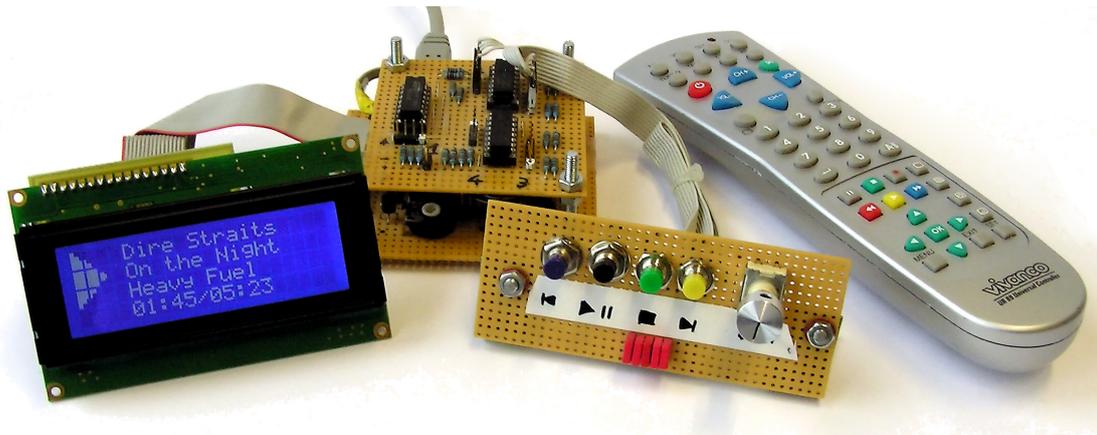


Abbildung 5.5.: Hardware: LC-Display, Taster, Ansteuerungsplatine und Fernbedienung

grammiersprache habe ich mich für C# entschieden. Die Programmiersprache C# ist Teil des .NET-Framework von Microsoft. Zur Ansteuerung des IO-Warrior kommt die von Christoph Schnedl (Schnedl, 2004) entwickelte Bibliothek IOW.NET zum Einsatz. Diese Bibliothek stellt die benötigten Services zur Darstellung von Text auf dem LC-Display und zur Abfrage des Infrarotsensors bereit. Auch die Wiedergabekomponente wurde nicht selber entwickelt, statt dessen wird ein eigenständiger Player ferngesteuert. Eigentlich sollte die Studie nur mit dem WinAmp als Player realisiert werden, aber da die Interaktion nicht wie vorgesehen möglich war, wurden auch andere Player in die Studie aufgenommen. Bei der Auswahl der eigenständigen Player wurden dann darauf geachtet, dass sie möglichst viele Musikformate wiedergeben können und sich die verschiedenen Player über andere Wege fernsteuern lassen. Dabei wurden folgende Player getestet:

1. WinAmp zählt mit zu den ersten Playern, die in der Lage waren, MP3 wiederzugeben. Die derzeit aktuelle Version 5.2 gibt alle gängige Musik- und Videoformate wieder. WinAmp lässt sich über „Windows Messages“² fernsteuern.
2. Der Zoomplayer aus dem Haus inmatrix (inmatrix, 2006) ist ein Video und DVD Player, kann aber auch mit alle gängigen Musikformaten abspielen. Der Zoomplayer lässt sich über eine TCP/IP Verbindung steuern.
3. Window Media Player (kurz WMP) ist wohl der bekannteste Player für Medieninhalte. Er lässt sich direkt aus dem .NET-Framework steuern. Voraussetzung ist die Installation des Window Media Player SDK von Microsoft.

²Mit Windows Messages kann man andere Anwendungen fernsteuern. In den Nachrichten kann werden Befehle verpackt. Die Befehle können normale Tastaturbefehle sein oder spezielle festgelegte Nachrichten.

Durch die Auswahl von möglichst verschiedenartige Playern, soll herausgefunden werden, welcher sich am besten für die hier gestellten Anforderungen eignet und welche Art von Interaktion sich am besten eignet.

5.1.3. Zusammenfassung

Mit der Machbarkeitsstudie sollte untersucht werden, ob es möglich ist, eine gesamte CD-Sammlung hinter der Fassade eines CD-Spielers zu verstecken. Dem Benutzer ist es möglich, mit der hier gewählten Hardware und Software auf eine akzeptablen Weise zu interagieren, und so die komplexe Steuerung über wenige Tasten durchzuführen. Bei der Umsetzung der Studie haben sich aber auch Schwächen des Hard- und Softwaredesigns gezeigt. Eine Softwareschwäche war die Implementierung der Bibliothek für die Ansteuerung des IO-Warrior im .NET-Framework. Diese Bibliothek reicht für die Studie aus, es fehlt aber die Möglichkeit, die Taster abzufragen, so dass sie für die Anwendung nicht ausreicht. Nach Aussage des Entwicklers wird sie auch nicht mehr weiterentwickelt, weiter fehlt die Dokumentation, und es gibt keinen Vermerk über die Lizenz, unter der sie veröffentlicht wurde. Nach Rücksprache mit dem Entwickler wurde die Bibliothek um die Tasterabfrage erweitert. Dabei stellte sich heraus, dass der IO-Warrior die Taster schnell genug abfragt, um auch das prellen³ mit zu übertragen, so kam es zur mehrfachen Ausführung der Befehle. Umgangen wurde das Problem vorläufig mit einer Softwarelösung, die einen Taster nach dem Betätigen für eine kurze Zeit sperrt.

Bei der Interaktion mit den Wiedergabekomponenten gab es verschiedene Probleme. Bei dem Versuch, WinAmp als Player zu benutzen, gab es Probleme mit der Kommunikation. Es stellte sich heraus, dass die Kommunikation erstmal nur in eine Richtung funktionierte: man kann WinAmp aus dem .NET-Framework so genannte „*Windows Messages*“ senden. Für eine beidseitige Kommunikation müsste man ein Plugin für WinAmp schreiben oder finden.

Die Interaktion mit dem Zoomplayer war auch nicht ganz einfach. Die Kommunikation lief über TCP/IP, so dass die ankommenden Daten immer erst mühsam zerlegt und umgewandelt werden mussten. Die aktuelle Spielposition, zum Beispiel, wurde als formatierte Zeichenkette übertragen, die nicht immer das gleichen Format⁴ hatte. Die Spielposition musste deshalb umständlich umgeformt werden. Ein weiteres Problem waren die Metadaten der Musikstücke. So war es weder möglich, diese Daten über das .NET-Framework auszulesen, noch ließen sich die Daten über den WinAmp beschaffen. Der Zoomplayer überträgt

³Prellen (engl. bouncing oder chatter): Bei elektrischen Tastern oder Schaltern kommt es beim Schließen oder Öffnen des Kontakts nicht sofort zu einem eindeutigen Zustand. Der Kontakt öffnet und schließt sich innerhalb eines Bruchteils einer Sekunde mehrere Male. Das führt dann zu Mehrfacheingaben. Dieser Effekt (entprellen, engl. debouncing) kann entweder mit Hilfe von Hardware oder Software unterdrückt werden.

⁴Die Spielposition wurde formatiert übertragen und es wurden die führenden Nullen weglassen (z. B. 5:45 (m:ss) oder 1:05:45 (s:mm:ss)).

die Metadaten bei einem Stückwechsel, aber man kann die Metadaten der Objekte in der Wiedergabeliste nicht abfragen. Wenn man die Wiedergabeliste abfragt, erhält man nur die Informationen über die Datei (Größe, Pfad, usw.). Die Spiellänge ist Null, bis der Spieler das Objekt einmal angespielt hat.

Da es für den Windows Media Player eine SDK für .NET-Framework gibt, ist der Player nicht direkt eine Stand-alone-Anwendung, sondern vollständig in die Wiedergabekomponente integriert. Dadurch ist es möglich, sehr einfach auf die Player Funktionen zuzugreifen und so alle benötigten Informationen auszulesen. Diese Studie zeigt, dass die Interaktion mit einem externen Player ohne SDK nicht geeignet ist, da nicht alle benötigten Informationen abgerufen werden, können oder umfangreich aufgearbeitet werden müssen.

5.2. Prototyp

Da die Machbarkeitsstudie zeigt, dass es möglich ist, die in der Anforderung gestellten Aufgaben zu erfüllen, soll nun mit einem Prototyp die zweite Ausbaustufe umgesetzt werden (siehe Abschnitt 3.4.1). Dabei sollen die Erkenntnisse, die bei der Entwicklung der Machbarkeitsstudie gemacht wurden, hier einfließen. Der geschriebene Quellcode soll allerdings nicht wieder verwendet werden. Die Hardware soll jetzt auf das Nötigste reduziert werden (siehe Abschnitt 4.4). In den Kapiteln 3 und 4 wurde eine offene, ausbaufähige Architektur entwickelt, diese soll nun im Prototyp umgesetzt werden. Dabei wird der Schwerpunkt auf die Routinefunktionen (siehe Abschnitt 3.4.1 Ausbaustufe 2) der Set-Top-Box gelegt. Die folgenden Abschnitte beschreiben wie die einzelnen Softwarekomponenten umgesetzt wurden und aus welcher Hardware die Set-Top-Box aufgebaut wurde.

5.2.1. Software

Für die Realisierung des Prototyps habe ich mich für Java als Programmiersprache und für Linux als Betriebssystem entschieden. Die Wahl ist primär aus zwei Gründen auf Linux gefallen. Einmal ist das Budget (siehe Kapitel 2) mit 500€ sehr knapp, so dass kein Geld für Softwarelizenzen ausgegeben werden soll, sondern eher für die Anschaffung der Hardware. Der zweite Grund ist die Modularisierung, die es ermöglicht, die Installation gut an die Bedürfnisse anzupassen. Um mich nicht von vornherein auf ein Betriebssystem festzulegen, habe ich mich für die plattformunabhängige Programmiersprache Java entschieden. Alternativ hätte ich die Anwendung im .NET-Framework entwickeln können und dann mit der Hilfe

von Mono⁵ auf Linux laufen lassen, aber es gibt kein plattformunabhängiges Multimedia Framework für .NET-Framework, so dass wieder auf eine fernsteuern Wiedergabekomponente angewiesen ist.

5.2.2. Die Anwendungslogik

Im Design (siehe Abschnitt 4.3) wurde die Anwendungslogik in die vier Teile Kern, Wiedergabe, Wiedergabeliste und Menü aufgeteilt. Hier wird jetzt genauer beschrieben, wie die einzelnen Teile implementiert werden und welchen Funktionsumfang sie bereitstellen sollen. Da die Schnittstellen für die Komponenten festgelegt sind, ist die Reihenfolge, in der die Komponenten implementiert werden, egal. Trotzdem wird eine logische Reihenfolge eingehalten. Erst wird das Grundgerüst des Kerns implementiert, dann die Wiedergabekomponente, danach das Benutzerinterface und anschließend die Persistenzschicht. Während die einzelnen Komponenten entwickelt werden, soll auch der Kern immer weiter ausgebaut werden.

Kernkomponente

Der Kern beherbergt die gesamte Anwendungslogik. Da die Logik immer vor den einzelnen Modulen entwickelt und getestet werden muss, wird hier auf Unit-Tests⁶ (Beck und Andres, 2004) und Mock-Objects⁷ ((Mackinnon u. a., 2000) und (MockObjects, 2006)) zurückgegriffen, um die Anwendungslogik zu testen und so die Zuverlässigkeit und Robustheit (siehe Abschnitt 3.4.1) sicherzustellen. Die Unit Tests werden benutzt, um über die Schnittstellen die Funktionalität der Anwendungslogik zu testen, und so sicherzustellen, dass der Kern korrekt auf die Eingabe reagiert. Die MockObjects werden benutzt, um den Zugriff auf externe Komponenten zu emulieren, wie zu Beispiel die Wiedergabekomponente oder die Datenspeicherungskomponente.

Im Prototyp werden alle im Design (siehe Tabelle: 4.1) beschriebenen Schnittstellen implementiert. Eine Schnittstelle für Fehlermeldungen wird nicht implementiert, hier wird auf den Logging Mechanismus der Java API zurückgegriffen. Der Kern registriert an dieser API eine Logdatei, in die dann alle Komponenten über ein statisches Objekt schreiben können.

⁵Genau wie das .NET-Framework Framework ist Mono eine Implementierung der „Common Language Infrastructure“(Microsoft, 2006c). Mono steht unter der GNU Public License und läuft auf allen gängigen Betriebssystemen.

⁶Unit-Tests sind Teil des Vorgehensmodells des Extreme Programming. Sie dienen zum Überprüfen, ob ein Softwaremodul korrekt arbeitet. Es werden vor und während der Entwicklung Testfälle geschrieben. Nach Änderungen am Quellcode lässt man dann die Testfälle durchlaufen.

⁷Mock-Objects sind Platzhalter für echte Objekte. Benutzt werden diese Platzhalter innerhalb vom Unit-Tests, um noch nicht geschriebene Module zu emulieren oder um Verhalten zu simulieren, das nur schwer nachzubilden ist (z.B. spontane Netzwerkfehler).

Unit-Tests und MockObjects werden nicht nur bei der Kernkomponente benutzt, sondern auch bei alle anderen Komponenten. So soll sichergestellt werden, dass keine fehlerhaften Module die Entwicklung oder die Weiterentwicklung behindern.

Wiedergabekomponente

Die Wiedergabekomponente muss die Steuerbefehle entgegennehmen und an den eigentlichen Player weitergeben. Für die Realisierung des Players gibt es mehrere Möglichkeiten. Einmal könnte man, wie bei der Machbarkeitsstudie (siehe Abschnitt 5.1.2) einen Stand-alone-Player benutzen. Die Stand-alone Player können zwar alle gängigen Audio-, Video- und Bildformate abspielen, aber sie versagen bei der Schnittstelle oder den Metainformationen. Da ich für den Prototypen das Betriebssystem Linux festgelegt habe und so die Windows Player nicht mehr zu Verfügung stehen, müssten außerdem neue Player evaluiert werden. Aus diesen Gründen soll eine andere Lösung gefunden werden. Eine mögliche Lösung wäre, die Komponente komplett selbst zu schreiben, aber die Neuentwicklung eines Players ist zeitlich sehr umfangreich, da es einfach zu viele verschiedene Komprimierungsformate für Musik gibt. Eine weitere Möglichkeit ist die Benutzung eines speziellen Frameworks zur Medienwiedergabe. Es gibt fast für alle Hochsprachen⁸ ein Multimedia Framework, welches die Wiedergabe von Video, Audio und Bildern ermöglicht. Anhand der Anforderungen (siehe Abschnitt 3.3) und Erfahrungen aus der Machbarkeitsstudie (siehe Abschnitt 5.1.3) habe ich mich für das „*Java Media Framework API (JMF)*“ entschieden. Die Entwicklung innerhalb eines Frameworks bietet den Vorteil, dass alle Grundfunktionen schon vorhanden sind. Die Funktionen müssen nur zu einer Softwarekomponente verknüpft, und mit einer Schnittstelle mit den gewünschten Funktionen versehen werden. Da die Wiedergabekomponente offen für Erweiterungen sein soll, wird der eigentliche Player über eine Fabrik (Abstract Factory Pattern) erzeugt (siehe Abschnitt 4.3.3). Die Fabrik wird erstmal für die von JMF unterstützten Musikformate (siehe (Sun Microsystems, 2005)) ausgelegt.

Wiedergabeliste

Da das „*Java Media Framework*“ keine Wiedergabeliste hat, muss eine eigene Komponente geschrieben werden. Die Wiedergabeliste wird aus einer Datei eingelesen und dann über eine abstrakte Fabrik (Abstract Factory Pattern) erzeugt. Die Metainformation werden aus der Musikdatei ausgelesen. Die Speicherung von Metadaten in der Musikdatei ist bei den gängigen Musikformaten ein normales Verfahren, diese Information können dann z. B. über

⁸Für Windows gibt es das „*DirectX*“ Framework von Microsoft für C/C++ und .NET-Framework oder das „*Java Media Framework API*“ (kurz JMF) von Sun Microsystem für Java

eine Bibliothek ausgelesen werden. Jedes Wiedergabeobjekt in der Liste besitzt eigene Metainformationenobjekte mit Informationen über sich selbst. Zu diesen Informationen zählen bei den Musikstücken z.B. Künstler, Name und Spielzeit. Die Metadaten werden über eine Fabrik erzeugt (siehe Abschnitt 4.3.1). Diese Fabrik liest die Metadaten aus der Musikdatei und erzeugt ein Metadatenobjekt. Da der Umfang der Metainformationen stark variiert (siehe (ID3v2, 2006)) werden die Metadaten als Schlüssel/Wertepaare mit Hilfe der Klasse Properties am Objekt gespeichert. Wenn die Datei keine Metainformationen hat oder die Fabrik das Format nicht unterstützt, wird nur der Dateiname als Metainformation gespeichert. Da der Schwerpunkt auf die Wiedergabe von MP3-codierter Musik (siehe Abschnitt 3.4.1) gelegt, wurde habe ich mich für die „ID3 Class Library“ von Paul Grebenc (Grebenc, 2006) entschieden, um die Metadaten aus den Dateien zu lesen.

Menü

Das Menü dient zur Auswahl der Wiedergabeliste und zur direkten Auswahl von Musikdateien. Über die Menüschnittstelle können andere Komponenten neue Menüeinträge erzeugen oder alte Einträge löschen. Die Komponenten werden, wie im Design besprochen, mit dem Composite Pattern umgesetzt (siehe Abschnitt 4.3.1). Es gibt eine abstrakte Klasse „Menu“, von dieser Klasse erben die Klasse „MenuEntry“ und „MenuCommand“. Die abstrakte Klasse beschreibt die Operationen, die die Unterklassen implementieren müssen, implementiert aber schon andere Teile, die für die Unterklassen gleich sind. Jeder Menüeintrag (Klasse: MenuEntry) hat eine Sammlung von Unterknoten, diese Unterknoten sind entweder weitere Menüeinträge oder Befehle (Klasse: MenuCommand). Diese Befehle wird über die Taste „Enter“ ausgelöst und führen eine beliebige kontextabhängige Aktion aus. In dem Menü kann jede andere Komponente Einträge machen. Der Anwender kann das Menü über den Menü-taste aufrufen.

5.2.3. Benutzerinterface

Das Benutzerinterface teilt sich in zwei Teile (siehe Abschnitt 3.3.1) auf, die Interaktionsfunktion am Gerät und die Konfigurationsfunktionen, die über ein Webinterface gesteuert werden. Da bei der Machbarkeitsstudie schon eine Interaktionskomponente implementiert wurde, soll auf dieser Erfahrung aufgebaut werden.

Interaktion

Bei Interaktion wird auf der in der Studie entwickelten Hardware aufgesetzt (siehe Abschnitt 5.1.3). Da in Java entwickelt wird, wird jetzt die Bibliothek von Thomas Wagner (Wagner,

2006) benutzt, sie stellt alle benötigten Funktionen bereit, um die Hardware aus Java heraus anzusprechen. Es ist möglich, gleichzeitig das LC-Display anzusprechen, den Infrarotempfänger zu betreiben und die Taster abzufragen. Die Fernbedienung, die Taster und das LC-Display werden zusammen als eine Interaktionskomponente implementiert. Das Trennen in die Taster als Befehlskomponente und die Fernbedienung zusammen mit dem LC-Display als Interaktionskomponente, macht hier aus zwei Gründen keinen Sinn. Ersten wird alles über einen Hardwarebaustein angesteuert, so dass das Trennen in der Software nicht möglich ist, sie müssen auf der gleichen Instanz einer Klasse arbeiten. Zweitens verfügen die Taster ebenfalls über eine Menüsteuerung, so dass sie auch eine Anzeigekomponente zum Anzeigen des Menüs brauchen. Da die Taster eine Untermenge der Taster der Fernbedienung darstellen, ist es kein Problem, die Taster und die Fernbedienung in einer Komponente zusammenzufassen. Erstmals wird nur eine Interaktionskomponente implementiert. Auf eine Steuerung per Netzwerk wird verzichtet, sie würde sich aber ohne Probleme nachrüsten lassen.

Konfiguration

Die Set-Top-Box soll ohne Monitor und Tastatur konfigurierbar sein, so dass die Konfiguration über ein Webinterface realisiert wird. Hierbei soll der Anwender mit bekannten Paradigmen konfrontiert werden (siehe Abschnitt 3.4.1). Es wird davon ausgegangen, dass die Set-Top-Box erst einmal über ein Kabel ans Hausnetz angeschlossen wird (siehe Abschnitt 4.4), um die Grundkonfiguration vorzunehmen (u. a. Einrichten des W-Lan), und erst dann im Wohnzimmer aufgebaut wird. Für die Konfiguration der Set-Top-Box wird eine externe Software eingesetzt. Mit der Software Webmin⁹ ist es möglich, das Media-Center bequem aus der Ferne zu konfigurieren. Über Webmin kann der Anwender dann Netzwerkeinstellungen ändern, Backups erstellen und Updates einspielen. Mit Webmin kann man fast fast jeden Dienst im System konfigurieren. Ein Anwender ohne Kenntnisse würde sich auf Grund des Umfangs nicht zurechtfinden, aber die Software ist stark modularisiert. Dadurch ist es möglich, dem Anwender nur die für ihn relevanten Module zu zeigen. Neben der Administration des Systems soll es aber auch möglich sein, die Media-Center Software zu konfigurieren. Diese Konfigurationskomponente wird als Adapter implementiert, der es externer Software ermöglichen soll, die Set-Top-Box über eine TCP/IP Verbindung zu konfigurieren. In diesem Prototypen soll dieser Adapter erstmal nur zwei Operationen ermöglichen: die Erstellung von neuen Wiedergabelisten und das Einbinden von Verzeichnissen. Diese beiden Operationen werden dann als Modul für das Webinterface realisiert, dazu werden diese Module in Perl implementiert. Die Konfiguration des kabellosen Netzwerks ist mit Webmin nicht möglich, so dass das W-Lan anders konfiguriert werden muss. Die Set-Top-Box wird für die Grundeinstellungen an

⁹Webmin (Webmin, 2006) ist ein Programmpaket zur Verwaltung eines Betriebssystems über ein Webinterface. Es unterstützt alle gängigen Betriebssysteme wie z. B. Unix basierte, Windows und Mac OS X.

einen Monitor und eine Tastatur angeschlossen. (siehe Anhang A). Später soll für die W-Lan Konfiguration ein Webmin Modul geschrieben werden, um dem Benutzer die Einstellungen zu erleichtern.

5.2.4. Persistenz

Innerhalb der Persistenzschicht werden zwei Komponenten implementiert: eine Komponente, die den Zugriff auf die Wiedergabeobjekte ermöglicht, und eine Komponente, die für die Speicherung und das Laden der Wiedergabelisten zuständig ist.

Wiedergabeobjekte

Im Abschnitt 3.3.2 wurde festgelegt, dass die Wiedergabeobjekte auf der Netzwerkfestplatte gespeichert sind. Die Set-Top-Box hat über das W-Lan Zugriff auf diese Daten. Die Datenkomponenten müssen sich um den Zugriff auf die Daten kümmern und sie dem Kern bereitstellen. Die Datenkomponente hat zwei Möglichkeiten, auf die im Netzwerk verteilten Daten zuzugreifen: Einmal gibt es die Möglichkeit, die Netzwerklaufwerke über das Betriebssystem einzubinden, oder man spricht die Netzwerklaufwerke direkt aus dem Framework an. Für Java gibt es ein entsprechendes Projekt: Die *JCIFS* Bibliothek (JCIFS, 2005) ermöglicht es, direkt auf die smb-Freigabe¹⁰ zuzugreifen. Der Nachteil an dieser Methode ist die Interaktion mit der Wiedergabekomponente und dem Auslesen der Metainformationen. Der Zugriff auf eine Datei über die *JCIFS* Bibliothek wird als Stream realisiert, während das Java Media Framework über die *DataSource* Klasse auf die Datei zugreift. Mit Hilfe der *JCIFS* Bibliothek könnte man eine *DataSource* Klasse für SMB-Protokoll realisieren (Sun Microsystems, 2005). Dann müsste man noch einen Adapter zum Auslesen der Metadaten schreiben. Die Metadaten von MP3-Dateien stehen meist am Ende (siehe (ID3v2, 2006)) der Datei, also müsste man den Stream einmal durchlaufen lassen, um an die Informationen zu kommen. Will man eine neue Datenkomponente für ein anderes Netzwerkprotokoll hinzufügen, muss man also auch Änderungen an anderen Komponenten machen, was gegen die Komponententechnologie spricht. Eine andere Variante wäre es, die Datei erst auf das lokale Dateisystem zu kopieren und dann nur den Pfad an die Wiedergabe weiterzugeben. Wenn man die Datei cachet, gib es zwei Probleme. Erstens könnte die Datei zu groß sein und zweitens braucht der Vorgang Zeit. So würden sich eventuell längere Pausen beim Wechseln der Musikstücke ergeben. Diese Probleme kann man umgehen, indem man die Dateien über das Betriebssystem in den lokalen Verzeichnisbaum einbindet. Da Linux eine Vielzahl von Dateisystemen und Netzwerkprotokollen unterstützt, habe ich mich für diese Lösung entschieden.

¹⁰smb: Server Message Block ist ein Kommunikationsprotokoll für Datei-, Druckerfreigaben und Namensdienste.

Um bei der Portierung auf ein anderes Betriebssystem nicht die gesamten Komponenten anpassen zu müssen, werden die betriebssystemabhängigen Klassen über eine Fabrik (Abstract Factory Pattern) erzeugt.

Das Einbinden von externen Speichermedien wie CDs, DVDs und USB-Speichermedien (siehe Abschnitt 3.4.1 Ausbaustufe 3) wird identisch abgewickelt wie die Netzwerkspeicherung. Das Betriebssystem bindet die Medien in den vorher festgelegten Ordner ein und verschickt ein Event an die Kernkomponente. Der Kern gibt dieses Event an die Datenkomponente weiter, die dann die Daten neu einliest. Beim Entfernen werden über den gleichen Mechanismus die Daten wieder ausgetragen. Die Datenkomponente trägt den bereitgestellten Inhalt (Musik und Wiedergabelisten) in das Menü des Media-Centers ein. Da die Navigation einfach gehalten werden soll, wird erstmal nur der Verzeichnisbaum des Speichermediums als Navigationsbaum eingetragen. Später könnte man einen Suchbaum auf Basis der Metadaten, sowie eine Sortierung nach den Metadaten (Interpret, Album, usw.) anbieten. Da die Musik nicht lokal gespeichert sondern über das Netzwerk eingebunden wird, ergibt sich das Problem der Datenveränderung, während das Media-Center darauf zugreift. Da es sich hier um eine Installation im einem privaten Haushalt handelt, wird dieses Problem erstmal zurückgestellt. Da auch neue Daten auf das NAS gespielt werden können, muss es eine Update Funktion geben, die dazu führt, dass das Menü aktualisiert wird. Diese Funktion wird als Menüpunkt im Menü angeboten und die Set-Top-Box sucht in regelmäßigen Abständen selbständig nach neuen Inhalten.

Wiedergabelisten

Neben der Wiedergabe ganzer Ordnerinhalte soll es auch möglich sein, eine vorher bestimmte Auswahl wiederzugeben (siehe 2.1.2). Diese Wiedergabelisten werden über die Konfigurationskomponente erstellt (siehe Abschnitt 3.3.1) und über die Persistenzschicht gespeichert und geladen. Da die Listen lokal gespeichert (siehe Abschnitt 3.3.3) werden, gibt es mehrere Möglichkeiten der Speicherung. Einmal könnte man das Objekt der Wiedergabeliste direkt auf die Festplatte schreiben. Da beim Serialisieren von Objekten auch alle Objektverweise rekursiv abgespeichert werden, also die Wiedergabeliste, das Wiedergabeobjekt und dessen Metadaten in der Liste mit gespeichert werden, würde das bei großen Wiedergabelisten zu einem großen Platzbedarf führen. Um den Platzbedarf zu verkleinern, sollte man vor dem Speichern die Daten aufräumen, also die Referenz auf die Metadaten lösen oder die Daten komprimieren. Eine andere Möglichkeit wäre die Speicherung der wichtigsten Daten in einer strukturieren Textdatei, in einer Datenbank oder in den Benutzereinstellungen (Java Preferences Object¹¹). Für den Prototypen habe ich mich für eine

¹¹Über das Paket *java.util.prefs* lässt sich unter Windows auf einen Teil der Registry zugreifen, unter Linux werden diese Daten in einer XML-Datei gespeichert. Es gibt zwei Gruppen, auf die man lesend und schreibend zugreifen kann. Diese Gruppen sind Benutzerumgebung und Systemumgebung.

Speicherung in einer XML-Datei¹² entschieden. Das bedeutet etwas mehr Aufwand beim Speichern und Einlesen im Vergleich zu der Serialisierung in der Programmiersprache oder der Speicherung der Daten in den Benutzereinstellungen, ist dafür aber nicht so aufwendig wie das Speichern und Auslesen aus einer Datenbank. Ein Problem ist die Pfadabhängigkeit der Daten. Wenn die Musikdaten auf dem Netzwerkspeicher verschoben wurden, ist es nicht mehr möglich, die Daten wiederzufinden. Wenn eine Datei beim Laden nicht gefunden wird, wird sie übergangen. Um diese Pfadabhängigkeit zu minimieren, wird in der Liste nicht der absolute Pfad sondern ein Verweis auf die Datenkomponente und der relative Pfad innerhalb dieser Komponente angegeben.

Jede Wiedergabeliste wird als eine eigene XML-Datei gespeichert. Zu jeder Wiedergabeliste können so allgemeine Informationen gespeichert werden. Erstmal wird nur der Name der Liste, die Größe und das Erstellungsdatum¹³ gespeichert. Später sollen hier auch noch Informationen des Benutzers gespeichert werden, wie zum Beispiel der Name des Erstellers, Bemerkung oder Kategorie. Für jeden Eintrag wird die Datenquelle, der Pfad innerhalb der Datenquelle und die Länge des Musikstücks gespeichert. Die Spiellänge wird deshalb in der Wiedergabeliste gespeichert, weil sie nicht immer in den Metadaten gespeichert ist. Ein Teil der exportierten Wiedergabeliste:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE preferences SYSTEM "playlist.dtd">
<playlist name="PlayList1">
  <size>"2"</size>
  <createDate>"1140288784759"</createDate>
  <!-- Sat Feb 18 19:53:04 CET 2006-->

  <entry DataSourceName="1" Path="Artist001/Album2/Track1.mp3" duration="185"/>
  <entry DataSourceName="NASKeller" Path="Interpret01/01.mp3" duration="245"/>
  <entry DataSourceName="1" Path="Artist001/Album2/Track2.mp3" duration="199"/>
  <entry DataSourceName="1" Path="Artist001/Album2/Track2.mp3" duration="220"/>
</playlist>
```

Dieses Listing zeigt eine Wiedergabeliste mit vier Einträgen.

¹²XML hat sich bei der Serialisierung und Deserialisierung von Objekten durchgesetzt. Mit XML ist es möglich Daten völlig unabhängig von Betriebssystem und Programmiersprache auszutauschen.

¹³Die Speicherung von Metadaten ermöglicht zusätzliche Funktionen. So könnte man z. B. in regelmäßigen Abständen alle Listen durchgehen und überprüfen, ob die in der Liste gespeicherten Einträge noch erreichbar sind. Wenn die Musikstücke nicht mehr auf dem NAS vorhanden sind, könnte man sie z. B. aus der Liste löschen oder erstmal nur markieren.

5.2.5. Sicherheit

In Abschnitt 3.4.1 und 4.3.2 wurde schon die Sicherheit der Software angesprochen. Diese wurde versucht, so gut wie möglich umzusetzen. Die Konfigurationssoftware Webmin (siehe Abschnitt 5.2.3) hat eine Benutzerverwaltung, so dass drei verschiedene Benutzer angelegt wurden:

1. root: hat Vollzugriff auf alle Module.
2. settopboxAdmin: darf die Konfiguration der Set-Top-Box ändern.
3. user: darf neue Wiedergabelisten anlegen.

So soll sichergestellt werden, dass kein Unbefugter Zugriff auf die Konfiguration erhält. Unter einer Standard-Linuxinstallation laufen mehr Dienste, als hier gebraucht werden. Um das System sicher zu machen, wurden die nicht gebrauchten Dienste stillgelegt. Um auszuschließen, dass die Software der Set-Top-Box das Betriebssystem durch Fehler oder Sicherheitslücken beeinträchtigt, läuft sie unter einem Account mit möglichst wenig Rechten.

5.2.6. Hardware

Das Hauptaugenmerk bei der Auswahl der Hardwarekomponenten lag auf deren Geräuschentwicklung, der Größe und dem Stromverbrauch (siehe Abschnitt 2.1.2). Die Komponente sollte nach Möglichkeit keine aktive Lüftung oder bewegliche mechanische Teile haben, damit das Gerät auch bei der Wiedergabe von leiser Musik nicht wahrgenommen wird.

Wenn man sich die Anforderungen der Analyse unter dem Gesichtspunkt der Hardwarearchitektur anschaut, ergibt sich das Bild Standard PC mit zwei Ausnahmen. Einmal soll das Gerät nicht hörbar sein, zweitens soll auf herkömmliche Eingabe- und Anzeigegeräte verzichtet werden. Die Geräusche eines normalen PCs werden hauptsächlich durch die Lüfter verursacht, die für die Kühlung der Komponenten verantwortlich sind. Dabei gilt: je mehr Leistung (Watt) ein PC umsetzt, umso mehr Abwärme produziert er. Also je größer die Stromaufnahme, desto aufwendiger und lauter die Kühlung. Da die Wiedergabe von Musik einen PC nicht sehr stark belastet, kann man auf weniger leistungsstarke Hardware zurückgreifen, die dafür aber leise ist. Also müsste es möglich sein, einen Computer passiv (ohne Lüfter) zu kühlen, wobei aber noch genug Performance abrufbar sein muss, um Musik wiederzugeben. Da die Set-Top-Box permanent eingeschaltet sein soll, muss trotzdem auf eine ausreichende Kühlung geachtet werden, auch in Hinblick auf einen niedrigen Stromverbrauch (siehe Abschnitt 4.4). Um das Laufgeräusch weiter zu minimieren, wäre es möglich, statt auf eine normale Festplatte auf eine leise Laptop-Festplatte oder sogar eine Speicherkarte zurückzugreifen. Da die Musikdaten auf der Netzwerkfestplatte gespeichert sind, braucht das Media-Center selbst nur einen Speicher für das Betriebssystem und die Software. Dafür würden

Speicherkarten ausreichen. Der Vorteil einer Speicherkarte ist, dass sie ohne bewegliche Teile auskommt, der Nachteil ist das schlechte Verhältnis zwischen Preis und der Größe des Speicher. Dieses Verhältnis ist bei einer Laptop-Festplatte zwar besser, aber dafür hat sie Laufgeräusche. Für den Prototyp habe ich mich für eine Laptop-Festplatte entschieden, da die größere Speicherkapazität es ermöglicht, mehrere Versionen der Software und verschiedene Betriebssysteme zu testen.

Daraus ergibt sich:

- Soundausgang
- Passive gekühlte CPU
- geringe Leistungsaufnahme (unter 50W)
- keine Laufgeräusche
- Speicherplatz für Software und Betriebssystem
- Netzwerkanschluß (W-Lan und LAN)
- Anzeigeelement
- Bedienelement

Diese Auflistung wurde als Basis für die Auswahl der Hardware genommen. In den folgenden Abschnitten wird die auf dieser Grundlage ausgewählte Hardware beschrieben.

Mainboard

Als Mainboard wählte ich ein VIA EPIA Mainboard. Die VIA EPIA Mainboards sind ultrakompakte PCs, die auf der Fläche von 17x17cm (Formfaktor: Mini-ITX) eine Standard PC unterbringen. Zur Wahl standen zwei verschiedene Mainboards, das VIA EP ML-6000EA und das VIA EPIA ME-6000. Das ML-6000EA würde zwar für die Anforderungen ausreichen, aber das ME-6000 bietet mehr Schnittstellen. Es verfügt über zwei FireWire Anschlüsse und einen Digital Sound Ausgang (SPDIF¹⁴). Ich habe mich für das ME-6000 Mainboard entschieden, denn es bietet einen digitalen Soundausgang. Die Eckdaten des ME-6000 Mainboards:

- Prozessor: 600MHz VIA EDEN
- VGA: VIA Unichrome AGP Grafik mit MPEG-2 Decoder
- Anschlüsse:

¹⁴SPDIF steht für Sony/Philips-Digital Interface, es bezeichnet einen digitale Übertragungs-Bus zwischen verschiedenen Audiogeräten.

- VGA, Composite und S-Video
- USB, Firewire
- Netzwerk (RJ-45)
- 3 Audiobuchsen: Line-Out, Line-In und Mic-In; oder Ausgang mit bis zu 6 Kanälen (5.1) Sound
- 1 PCI Steckplatz.

Das Mainboard wurde mit 256MB Hauptspeicher bestückt und bietet genug Platz für die Anwendung und das Betriebssystem. Für die kabellose Anbindung an das Netzwerk wurde eine PCI W-Lan-Karte von SMC mit einem prism54 Chipsatz eingebaut. Der prism54 Chipsatz wird direkt vom Linux Kernel unterstützt¹⁵.

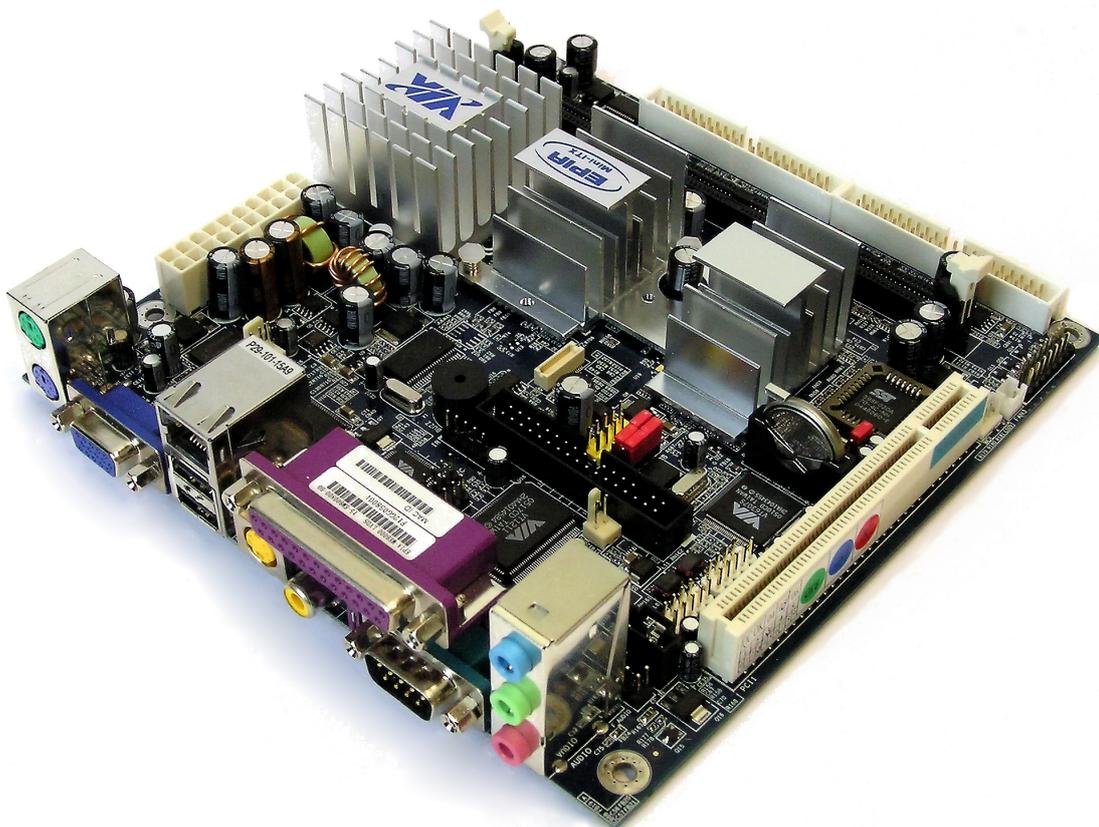


Abbildung 5.6.: Der Kern der Set-Top-Box: das VIA EPIA ME-6000 Mainboard

¹⁵Versucht wurde hier auch ein USB W-Lan-Dongle (LCS-8131G2) von Longshine mit einem ZyDAS Chipsatz (ZD1211), aber der Linuxtreiber (zd1211:2006, 2006) (Release r47) war nicht stabil genug. Die Einrichtung des W-Lan endete immer einem Absturz (Kernel Panic).

Display, Taster und Infrarot

Da sich die Schaltung mit dem IOWarrior24 vom Code Mercenaries bei der Machbarkeitsstudie bewährt hat, wurde dieser auch für den Prototyp benutzt. Auf dem Display sollen die Statusinformationen des Media-Center dargestellt werden (siehe Abschnitt 4.4). Hier soll das „*Look and Feel*“ eines MP3-Players nachempfunden werden, es werden also folgende Informationen dargestellt: Interpret, Album, Name des Musikstück, Spielposition, Gesamtzeit und der Zustand des Players (Play, Pause oder Stop). Dafür reicht ein textbasiertes LC-Display aus. Ich habe mich hier für ein 4 Zeilen LC-Display mit je 20 Buchstaben entschieden. Neben dem LC-Display werden noch vier Taster, ein Drehimpulsgeber und ein Infrarotempfänger ins Gehäuse eingebaut. Das Platinenlayout des ersten Prototypens dient als Basis für die Weiterentwicklung. Es wurde überarbeitet und damit erheblich kleiner, wodurch der Stromverbrauch sank. Um das Prellen zu verringern, wurden parallel zum Taster ein Kondensator und ein Widerstand in Reihe geschaltet.

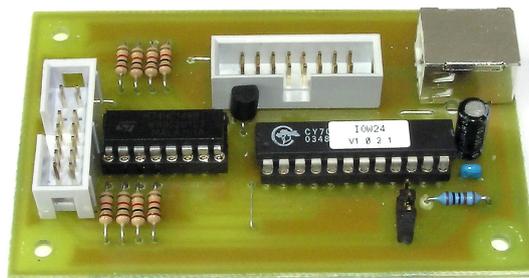


Abbildung 5.7.: Die Ansteuerungsplatine mit dem IOWarrior24

Gehäuse

Beim Gehäuse (siehe Abbildung 5.8) habe ich mich für das SilverStone LC02S (SilverStone, 2006) entschieden. Dieses Gehäuse ist zwar größer als benötigt, bietet aber deswegen genug Platz für den Einbau des Displays und der Steuerungselemente. Das Gehäuse eignet sich gut für den Einbau des LC-Displays, da auf die Frontplatte noch eine Blende geschraubt wird. Das Display selbst wird auf der Frontplatte montiert. In die Blende wird dann nur der Ausschnitt für das Sichtfenster des Displays gefräst. Für die Taster werden Löcher in die Blende und die Frontplatte gebohrt. Die Taster werden an der Blende befestigt. Der Drehimpulsgeber und der Infrarotempfänger werden auf eine Platine montiert, die auf der Frontplatte des Gehäuses montiert ist. Für den Empfänger wird ein Loch in die Blende gefräst, in das ein rote Plexiglasscheibe eingepasst wurde. Für die Welle des Drehimpulsgebers wurde ein Loch in die Blende gebohrt. In das Gehäuse wird noch eine Slimline DVD eingebaut,

um später auch Audio CDs abzuspielen (siehe Abschnitt 3.4.1). Die Wahl fiel auf ein DVD-Laufwerk, da es kaum noch Slimline CD-Laufwerke zu kaufen gibt und auch die Möglichkeit in Betracht gezogen wird, dass die Set-Top-Box später auch mal DVD abspielen soll. Bei der Halterung für die Festplatte des Gehäuses sind nur Bohrungen für den Einbau einer 3,5 Zoll Festplatte vorhanden, so dass für den Einbau der Laptop-Festplatte (2,5 Zoll) vier Löcher in die Montageplatte gebohrt werden. Die Festplatte wird dann bei der Montage mit der Hilfe von Gummidurchführungen vom Gehäuse entkoppelt, damit sich die Vibrationen der Festplatte nicht auf das Gehäuse übertragen. Diese Entkopplung sorgt dafür, dass die Laufgeräusche der Set-Top-Box verringert werden. Da das Gehäuse eigentlich für den Einbau eines ATX Mainboards vorgesehen ist, gab es Probleme beim Einbau der PCI W-Lan-Karte. Die Kombination Mini-ITX und der dem Gehäuse beigelegten PCI Adapter Karte passt nicht zusammen, so dass ein weiter Adapter gekauft werden musste (siehe Anhang C.2).



Abbildung 5.8.: Die fertig Set-Top-Box.

Das System wurde mehrere Tage im Dauerbetrieb getestet. Dabei wurde die Leistungsaufnahme gemessen. Die durchschnittliche Leistungsaufnahme lag bei 35 Watt. Da die Innentemperatur des Gehäuses während des Tests auf nicht über 38°C stieg, bei einer Umgebungstemperatur von 20°C, wurde der Gehäuselüfter ausgebaut, um zu testen, ob der Netzteil Lüfter ausreicht, um alle Komponenten ausreichend zu kühlen. Der Test musste nach wenigen Stunden abgebrochen werden, da es im Gehäuse zu warm wurde. Der Grund für die Überhitzung war ein Konstruktionsfehler im Netzteil. Der Lüfter des Netzteils führte die Abluft wieder zurück in das Gehäuse. Vorher hatte der Zusatzlüfter für genug Luftzirkulation gesorgt. Um den Lüfter nicht wieder einzubauen müssen, wurde ein Winkelblech im Gehäuse montiert, um die Abluft über den Netzteil Lüfter abzuführen. Mit dieser Konstruktion war es möglich, die Set-Top-Box über mehrere Tage laufen zulassen, ohne dass die Innentempera-

tur über 39°C stieg. Die technischen Zeichnungen für den Umbau des Gehäuses befinden sich im Anhang C.

5.3. Zusammenfassung

Mit einer Machbarkeitsstudie wurde untersucht, ob es überhaupt möglich ist, die Anforderungen an die Software umzusetzen, dazu wurde die Kernaufgaben (die Steuerung der Wiedergabe über eine Fernbedienung) umgesetzt. Die Machbarkeitsstudie hat gezeigt, dass es möglich ist, mit einer geeigneten Software eine gesamte CD-Sammlung hinter der Fassade eines CD-Spielers zu verstecken. Dazu wurde die Benutzerschnittstelle in zwei Teile aufgeteilt. Die Steuerung der Wiedergabe wurde über eine Fernbedienung realisiert, während der Aufbau der verschiedenen Wiedergabelisten von einem PC erledigt wird. Das Problem der Machbarkeitstudie war die Interaktion mit der Wiedergabesoftware (Player). Es war mit wesentlich mehr Aufwand verbunden, einen Player fernzusteuern, als ein Framework zu benutzen.

Mit den Erfahrungen aus der Entwicklung der Studie wurde dann ein Prototyp entwickelt. Dieser Prototyp wurde dann nicht mehr an einem Standard PC betrieben, sondern es wurde eine Hardwareplattform gefunden, die den Vorstellungen der Familie Müller sehr nahe kommt. Dabei wurde der Schwerpunkt auf die Geräuschentwicklung, Stromaufnahme und die Kosten gelegt. So wurde Set-Top-Box entwickelt, die 24 h am Tag läuft und dabei fast nicht hörbar ist.

5.3.1. Fazit

Ziel dieser Arbeit war es zu untersuchen, ob es möglich ist, eine größere CD-Sammlung über eine Fernbedienung sinnvoll zu steuern. Dabei sollte die Hardware möglichst leise sein und nicht über 500€ kosten. Die Realisierung hat gezeigt, dass es möglich ist, wenn man die Interaktion sinnvoll in zwei Teile zerlegt, und so die administrativen Aufgabe von den eigentlichen Wiedergabefunktionen trennt. Die Administration, das Erstellen von neuen Wiedergabelisten und das Einspielen von neuer Musik auf den NAS, wird von einem normalen PC aus erledigt, während man dann das System bequem vom Wohnzimmer aus steuern kann. Der hier entwickelte Prototyp bietet noch nicht die in Abschnitt 2 gesetzten Anforderungen. Das in Kapitel 4 entworfene Design ist aber eine offene Architektur, die sich leicht erweitern lässt. Über verschiedene Design Pattern wird sichergestellt, dass man die einzelnen Komponenten der Software erweitern kann, ohne das gesamte System zu verändern. So kann man später noch die Unterstützung für andere Musikformate oder andere Metadaten ins System einbauen.

Man könnte den Stromverbrauch noch weiter minimieren, da die Set-Top-Box derzeit permanent an ist könnte man sie in einen Stand-By-Modus versetzen wenn sie nicht benutzt wird. Eine Stand-by-Funktion nachzurüsten, ist kein großes Problem, denn es gibt Möglichkeiten, das über das Betriebssystem abzuwickeln. Wenn die Set-Top-Box allerdings über die Fernbedienung wieder aufgeweckt werden soll, müsste die Hardware noch erweitert werden, ein Wake-up über einen Taster stellt hingegen kein Problem dar. Um noch mehr Strom zu sparen, könnte man auf eine Speicherkarte statt einer Festplatte zurückgreifen. Die Speicherkarten haben zwar eine sehr geringe Kapazität, aber der Prototyp hat gezeigt, dass er mit einem minimalen Linux auskommen würde.

Mit der Auswahl der Programmiersprache wurde versucht, die Plattformabhängigkeit zu minimieren. Während der Entwicklung mussten Kompromisse gemacht werden, indem die Software auf die Plattform Linux festgesetzt wurde. Wenn die Software auf eine andere Plattform portiert werden soll, man die Komponenten der Persistenzschicht (Einhängen des Netzwerklaufwerkes) erweitern. Es muss eine neue Klasse geschrieben werden und dann die entsprechenden Fabrikklassen erweitert werden. Die Wiedergabe- und die Interaktionskomponente müssen auf Lauffähigkeit überprüft werden, da sie von Bibliotheken außerhalb von Java abhängig sind.

Das Speichern der Daten im Netzwerk bringt natürlich Probleme mit sich, so können z. B. die Daten im Netzwerk verschoben werden. Diese Probleme wurde versucht zu minimieren, indem nicht der absolute Pfad sondern der relative Pfad innerhalb des Netzwerkverzeichnis gespeichert wurde. Dies hilft aber nur, wenn der gesamte Netzwerkordner z. B. auf ein größeren NAS kopiert wird, und dann die neue Freigabe unter dem Name der alten eingebunden wird. Wenn nur ein Teil der Daten in ein anderes Verzeichnis kopiert, dann ist dieses Verfahren nicht anwendbar. Das Problem lässt sich auch kaum lösen, denn dazu müsste man das gesamte Management der Daten über eine Datenbank realisieren, die dann die Informationen über eine Datei speichert. In der Wiedergabeliste würde dann nur eine Identifikationsnummer (kurz: ID) gespeichert. Mit Hilfe der ID kann man dann die Information über die Datei aus der Datenbank lesen. Das Problem dabei ist, dass man nicht sicherstellen kann, dass alle Aktionen, die den Datenbestand verändern, auch die Datenbank aktualisieren. Um die Daten zu verwalten wird man sich nicht erst eine Software installiert, wenn das mit dem Dateibrowser des Betriebssystems geht. Ein mögliche Lösung wäre es, das Betriebssystem des NAS um eine Software zu erweitern. Diese Software könnte alle Änderungen des Benutzers an die Datenbank weitergeben, und so die Datenbank immer aktuell halten.

6. Ausblick

Hier soll noch einmal zusammengefasst werden und anschließend ein Fazit über die Arbeit gezogen werden.

6.1. Zusammenfassung

Ziel der Arbeit war die Entwicklung eines ubiquitären Media-Centers als Set-Top-Box. Als Grundlage für die Entwicklung diente die fiktive Familie Müller, die ihre CD-Sammlung durch ein Media-Center ersetzen will. Das Ziel dabei war, eine Lösung zu finden, wie man im Wohnzimmer einfach über den Inhalt der Sammlung navigieren oder eine vorher zusammengestellte Liste wiedergeben kann. Dabei sollte die Interaktion mit dem Gerät über eine Fernbedienung und ein Display realisiert werden. Die Anwender sollten dabei mit einem bekannten „*Look und Feel*“ konfrontiert werden, die Steuerung soll einem CD-Spielers bzw. MP3-Spielers nachempfunden sein, während die Konfiguration dem Webinterface eines Routers ähneln soll.

Zu Beginn wurde die vorhandene Situation im Haus der Familie Müller geschildert. Das Problem der Familie Müller ist zum einen die Größe der CD-Sammlung, die es kaum noch ermöglicht, den Überblick zu behalten und somit das Heraussuchen und das Zurücksortieren einer CD erschwert. Ein anderes Problem ist der steigende Platzbedarf der Sammlung. Auch mit der Steuerung der Anlage sind die Müllers nicht mehr zufrieden, denn es ist nicht möglich ein Auswahl abzuspielen, auch merkt sich der CD-Spieler die Favoriten auf einer CD nicht. Die Familie Müller möchte eine Lösung, die es ermöglicht, die gesamte Sammlung über eine Fernbedienung zu steuern, ohne immer wieder die CD wechseln zu müssen. Dabei sollte sich die neue Hardware in die Stereoanlage integrieren. Bei der Hardware war es der Familie wichtig, mögliches keine Laufgeräusche zu hören, und insgesamt nicht mehr als 500€ auszugeben.

Nachdem in der Analyse eine mögliche Lösung ausgewählt wurde, wurde dann ein konkretes Design entwickelt, um diese Lösung umzusetzen. Dazu wurde die Umgebung, in der die Set-Top-Box aufgestellt ist, konkretisiert (siehe Abbildung 4.1). Die Set-Top-Box wird an die Wohnzimmer-Stereoanlage angeschlossen und in das Heimnetzwerk der Familie Müller integriert. Aus den Anforderungen wurde dann eine Software-Architektur entwickelt, indem die

Anforderungen auf eine 3-Schichten-Architektur abgebildet wurde. Die einzelnen Schichten wurden dann in ihre Komponenten zerlegt. Während dieser Zerlegung wurden die Schnittstellen entwickelt. Dann wurden die einzelnen Komponenten entwickelt und beschrieben, mit welchen Pattern sie implementiert wurden.

Nachdem nun die Anforderungen in ein Design überführt wurden, wurde die Machbarkeit mit Hilfe einer Studie untersucht. Für diese Studie wurde eine Hardware zur Interaktion mit dem Anwender entwickelt. Die Hardware bestand aus einer Fernbedienung, Steuerungstaster und einen LC-Display und wurde zunächst an einem Standard PC betrieben. Innerhalb der Studie wurden auch verschiedene Wiedergabekomponenten untersucht. Dabei wurden die verschiedenen Wege der Interaktion mit externer Software beleuchtet. Da es mit den getesteten Stand-alone-Player Probleme mit der Interaktion gab, wurde auf eine Wiedergabe-Software zurückgegriffen, die über ein SDK ansprechbar ist. Die Studie hat gezeigt, dass es möglich ist, die gesamte CD-Sammlung über eine Fernbedienung zu steuern, wenn man das Benutzerinterface in zwei Teil teilt. Die Steuerung der Wiedergabe erfolgt über die Fernbedienung, das Einrichten der Wiedergabeliste und das Erweitern des Datenbestandes erfolgt von einem PC aus. Nach dem positiven Erfolg der Studie wurde dann der erste Prototyp realisiert. Der Prototyp sollte die offene Architektur, die im Kapitel 4 erdacht wurde, komplett umsetzen. Bei der Entwicklung der Set-Top-Box wurde auf Standard PC Komponenten gesetzt. Die Komponenten wurden dann mit der, in der Studie entwickelten, Interaktionshardware zusammen in ein Hifi-Komponentenähnliches Gehäuse eingebaut. Beim Prototyp wurde dann auch die Persistenzschicht und die Konfigurationskomponente realisiert. Über die Persistenzschicht können Metadaten und die Wiedergabelisten geladen werden. Die Konfigurationskomponente wurde mit Hilfe einer externen Software realisiert, für die eigene Module geschrieben wurden, um die Software zu konfigurieren.

Im Rückblick auf die gesetzten Ziele kann man zusammenfassend sagen, dass es gelungen ist, eine umfangreiche CD-Sammlung hinter der Fassade eines normalen CD-Spielers zu verstecken. Bei der Entwicklung des Designs wurde stark auf die Erweiterbarkeit der Software geachtet. So ist eine offene und erweiterbare Architektur entstanden, die es wiederum ermöglicht, die später kommenden Anforderungen umzusetzen. So können neue Musikformate oder andere Features nachträglich eingebaut werden. Abschließend lässt sich sagen, dass die wesentlichen Anforderungen (Ausbaustufe 2) umgesetzt wurden. Die optionalen Features (Ausbaustufe 3) umzusetzen (siehe Abschnitt 3.4.1), ist technisch möglich, geht aber über den gesetzten zeitlichen Rahmen hinaus.

6.2. Fazit

Diese Arbeit zeigt, dass es möglich ist, eine große CD-Sammlung hinter der Fassade eines CD-Spielers zu verstecken. Dazu wurde eine Set-Top-Box entwickelt, die über Fernbedie-

nung und über das Netzwerk gesteuert wird. Bei der Auswahl der Hardware wurde auf niedrigen Geräuschpegel und geringen Stromverbrauch geachtet, um so eine Set-Top-Box zu bauen, die immer eingeschaltet sein kann, und nicht erst wie ein PC oder Media-Center gestartet werden muss. Die so entstandene Set-Top-Box ist ein so genanntes „*Leichtgewicht*“, (siehe Abschnitt 3.2.3) da die Set-Top-Box weder über eine leistungsfähige Hardware noch über einen eigenen Speicher für die Mediendaten verfügt. Die Interaktion mit dem Anwender wurde auf zwei verschiedene Interfaces verteilt. Die Routinefunktionen (Steuerung der Wiedergabe) sind über die Fernbedienung und die Taster zugänglich, die Komfortfunktionen sind über ein Webinterface abrufbar. Diese Trennung erfolgt, da es einfach ergonomisch ist, alle Funktionen über ein kleines Display und Fernbedienung abzubilden. Alternativ hätte man diese Funktionen auf dem Fernseher darstellen können, mit Hilfe einer besseren Eingabemöglichkeit. Es stellt sich dann die Frage, ob jemand den Fernseher¹ anschaltet, um Musik zu hören. Deshalb wurde in dieser Arbeit eine Komponente entwickelt, die man in die Hifi-Anlage integrieren kann. Die Philosophie, für jeden Job genau ein Gerät einzusetzen, mag für den einen ein Vorteil, für den anderen ein Nachteil sein. Ich halte Spezialwerkzeuge für einen Vorteil gegenüber einem Multifunktionsstool, das alle Einsatzmöglichkeiten ein bisschen abdeckt, aber bei speziellen Einsätzen versagt.

6.3. Ausblick

Im „*Zuhause von Heute*“ finden sich immer mehr kleine Gadgets, die einem die Arbeit abnehmen sollen. Dieser Trend wird auch weiter anhalten. Es gibt schon heute sehr konkrete Vorstellungen, wie sich das Wohnen in den nächsten Jahren verändern wird. Eine der Vorstellungen ist das T-Com-Haus (Telekom, 2006) in Berlin, oder das inHaus (Fraunhofer, 2006) in Duisburg. In diesen Häusern sind alle Systeme miteinander vernetzt und werden über verschiedene Wege gesteuert, so dass man ein Gerät z. B. über eine Fernbedienung oder ein Touch Screen anspricht. Die Kommunikation unter den Geräten erfolgt dann per Web-Service. So ist es möglich, dass die Geräte nicht nur irgendwelche Schnittstellen haben, sondern auch genaue Informationen über die Parameter. Die Interaktionsgeräte können so auch alle Funktionen der Geräte anbieten. In eine solche Infrastruktur könnte man die hier entwickelte Set-Top-Box leicht integrieren, man müsste nur eine weitere Interaktionskomponente schreiben, die als Web-Service realisiert ist. Neben den Web-Service gibt es noch ein anderes Protokoll, welches das Zusammenspiel innerhalb eines Hausnetzwerkes verbessern soll, das „*Universal Plug and Play*“² (kurz UPnP) Protokoll. Innerhalb der UPnP gibt

¹Diese Art der Interaktion nutzen die „*Schwergewichtigen*“ Media-Centers fürs Wohnzimmer. Diese Geräte verfolgen die Devise: „*Ein Gerät für Wiedergabe und Aufnahmen*“ (Musik, Bilder und Video).

²Universal Plug and Play ist das erweiterte Plug and Play von Microsoft (Microsoft, 2006a). Es ist eine Schnittstelle zur Ad-hoc-Kommunikation zwischen den unterschiedlichsten Diensten und Geräten. Mehreren UPnP-

es eine Erweiterung für Audio und Video (UPnP-AV). Das UPnP-AV beschreibt drei verschiedene Geräteklassen: Media Renderer, Media Server und Control Point. Der Media Renderer ist ein Wiedergabegerät. Der Media Server bietet die Inhalte an und ermöglicht eine Sortierung. Der Control Point ist die Bedieneinheit, er ruft die Informationen beim Media Server ab und steuert ein oder mehrere Media Renderer. Einer dieser Media Server ist der „*twonkyvision*“ (TwonkyVision, 2006), einen solchen Server kann man auch auf einem NAS installieren. Man könnte eine Interaktionskomponente implementieren³, die es ermöglicht, die Set-Top-Box über UPnP-Geräte zu steuern oder eine Persistenzkomponente implementieren, die es ermöglicht, mit einem Media Server zu interagieren.

Ein andere Möglichkeit wäre es, die Set-Top-Box in ein anderes Gerät einzubauen, so wie man es derzeit bei den Fernsehern sieht. Geräte, die vorher eine Set-Top-Box (DVD-Player oder Digitalempfänger) waren, sind jetzt fest in den Fernseher integriert. Gleiches kann man sich auch für die hier entwickelte Set-Top-Box vorstellen, nur dass man sie nicht in der Fernseher einbaut, sondern in einen HiFi-Verstärker.

Man könnte auch mehrere Set-Top-Boxen in einem Netzwerk betreiben und diese dann über ein Webinterface verwalten. Eine solche Installation macht im Homebereich nicht viel Sinn, man könnte so aber ein Kaufhaus oder eine bzw. mehrere Messehallen mit Hintergrundmusik versorgen.

Geräten ist es möglich, sich selbst zu konfigurieren und so ohne Eingriff des Anwenders in Verbindung zu treten. So soll es möglich sein, mit einem PDA die Stereoanlage zu steuern ohne Software zu installieren.

³Ein mögliche SDK für Java und UPnP ist von Satoshi Konno geschrieben (Konno, 2006).

Literaturverzeichnis

- [zd1211:2006 2006] ZyDAS ZD1211 802.11b/g USB WLAN chipset Linux drivers. 02 2006. – URL <http://zd1211.ath.cx/>. – Zugriffsdatum: 26.06.2006 79
- [Adams 2005] ADAMS, IT S.: *USB LCD Site*. 2005. – URL <http://www.usblcd.de/>. – Zugriffsdatum: 15.12.2005 10, 64
- [AMD 2006] AMD, Inc.: *AMD LIVE*. 03 2006. – URL <http://amdlive.amd.com/>. – Zugriffsdatum: 05.03.2006 23
- [Apple Computer 2006a] APPLE COMPUTER, Inc.: *Apple - iPod + iTunes*. 02 2006. – URL <http://www.apple.com/itunes/>. – Zugriffsdatum: 28.02.2006 10, 13, 23, 24, 114
- [Apple Computer 2006b] APPLE COMPUTER, Inc.: *iPod Users Guide*. www.apple.com (Veranst.), 03 2006. – URL <http://www.apple.com/support/manuals/>. – Zugriffsdatum: 06.02.2006 46
- [Association 2006] ASSOCIATION, Consumer E.: *2006 International CES*. 02 2006. – URL <http://www.cesweb.org/>. – Zugriffsdatum: 05.02.2006 13
- [Bachelot 2006] BACHELOT, Xavier: *RPMs for Via Epia motherboards*. 02 2006. – URL <http://washington.kelkoo.net/epia/>. – Zugriffsdatum: 19.02.2006 95
- [Beck und Andres 2004] BECK, Kent ; ANDRES, Cynthia ; O'HAGAN, Don (Hrsg.): *Extreme Programming Explained*. Second Edition. Addison-Wesley, 2004. – ISBN 0-321-27865-8 70
- [Botzet 2005] BOTZET, Nicolas: *Multimedia Systeme: Analyse existierender Systeme und Visionen*, Hochschule für Angewandte Wissenschaften Hamburg, Studienarbeit, 2005. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/studien/botzet.pdf>. – Zugriffsdatum: 05.11.2005 13, 22
- [BSI 2006] BSI: *Sicherheit im Funk-LAN*. 03 2006. – URL <http://www.bsi.de/literat/doc/wlan/index.htm>. – Zugriffsdatum: 02.03.2006 33

- [Buschmann u. a. 2000] BUSCHMANN, Frank ; NEUNIER, Regine ; ROHNERT, Hans ; SOMMERLAND, Peter ; STAL, Michael: *Pattern-orientierte Softwarearchitektur*. Addison Wesley, 2000. – ISBN 3-8273-1282-5 34, 37, 49
- [Conrad Electronic 2006] CONRAD ELECTRONIC, GmbH: *Conrad Electronic*. 02 2006. – URL <http://www.conrad.de/>. – Zugriffsdatum: 05.02.2006 10, 20
- [Corley 2006] CORLEY, Jason: *JPackage Project*. 02 2006. – URL <http://www.jpakage.org/rpm.php?id=3366>. – Zugriffsdatum: 19.02.2006 98
- [Cwlinux 2005] CWLINUX, Limited: *CwLinux Limited, Products - Character Serial LCD*. 2005. – URL http://www.cwlinux.com/eng/products/products_cw1602.php. – Zugriffsdatum: 11.12.2005 10, 65
- [Deutsche Messe 2006a] DEUTSCHE MESSE, AG: *CeBIT*. 02 2006. – URL <http://www.cebit.de>. – Zugriffsdatum: 01.03.2006 13
- [Deutsche Messe 2006b] DEUTSCHE MESSE, AG: *Digital Living*. 02 2006. – URL <http://www.digital-living.de>. – Zugriffsdatum: 01.03.2006 13
- [Devices 2005] DEVICES, Slim: *Slim Devices: Squeezebox2*. 2005. – URL <http://www.slimdevices.com/index.html>. – Zugriffsdatum: 11.12.2005 10, 24
- [Ekman 2006] EKMAN, Lars: *vt1211 module for 2.6 kernel*. 02 2006. – URL <http://hem.bredband.net/ekmlar/vt1211.html>. – Zugriffsdatum: 19.02.2006 94
- [Fraunhofer 2006] FRAUNHOFER, Gesellschaft: *inhaus duisburg*. 02 2006. – URL <http://www.inhaus-duisburg.de/>. – Zugriffsdatum: 26.02.2006 86
- [Gamma u. a. 2002] GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph ; VLISSIDES, John: *Design Patterns*. Addison Wesley, 2002. – ISBN 0-201-633361-2 34, 41, 42, 44, 45, 46, 47, 51, 62
- [Grebenc 2006] GREBENC, Paul: *JID3 A Java™ ID3 Class Library Implementation*. 02 2006. – URL <http://jid3.blinkenlights.org/>. – Zugriffsdatum: 23.02.2006 72
- [ID3v2 2006] ID3V2: *ID3v2*. 02 2006. – URL www.id3.org. – Zugriffsdatum: 23.02.2006 72, 74, 100
- [inmatrix 2006] INMATRIX: *Zoomplayer*. 02 2006. – URL www.inmatrix.com/zoomplayer. – Zugriffsdatum: 10.02.2006 67
- [Intel 2006a] INTEL, Corporation: *Intel Centrino Mobile Technology*. 02 2006. – URL <http://www.intel.com/products/centrino/index.htm>. – Zugriffsdatum: 23.02.2006 22

- [Intel 2006b] INTEL, Corporation: *Intel VIIV Technology*. 02 2006. – URL <http://www.intel.com/products/viiv/index.htm>. – Zugriffsdatum: 23.02.2006 22
- [JCIFS 2005] JCIFS: *The Java CIFS Client Library*. 11 2005. – URL <http://jcifs.samba.org/>. – Zugriffsdatum: 26.02.2006 74
- [Kahlbrandt 2001] KAHLBRANDT, Bernd: *Software-Engineering mit der Unified Modeling Language*. Springer, 2001. – ISBN 3-540-41600-5 40
- [Konno 2006] KONNO, Satoshi: *CyberLink for Java*. 02 2006. – URL <http://www.cybergarage.org/net/upnp/java/>. – Zugriffsdatum: 26.02.2006 87
- [Mackinnon u. a. 2000] MACKINNON, Tim ; FREEMAN, Steve ; CRAIG, Philip: *EndoTesting: Unit Testing with Mock Objects*. 2000. – URL <http://www.connextra.com/aboutUs/mockobjects.pdf> 70
- [Mercenaries 2005] MERCENARIES, Code: *IO-Warrior Universeller IO-Controller am USB*. 2005. – URL <http://www.codemerics.com/IOWarriorD.html>. – Zugriffsdatum: 15.12.2005 65, 97, 101
- [Microsoft 2006a] MICROSOFT: *Welcome to the UPnP Forum!* 02 2006. – URL <http://www.upnp.org/>. – Zugriffsdatum: 26.02.2006 22, 86, 116
- [Microsoft 2006b] MICROSOFT, Corporation: *Microsoft .NET Homepage:*. 03 2006. – URL <http://www.microsoft.com/net/>. – Zugriffsdatum: 06.03.2006 114
- [Microsoft 2006c] MICROSOFT, Corporation: *MSDN .NET Framework Developer Center: ECMA C# and Common Language Infrastructure Standards:*. 03 2006. – URL <http://msdn.microsoft.com/netframework/ecma/>. – Zugriffsdatum: 06.03.2006 70, 114, 115
- [MockObjects 2006] MOCKOBJECTS: *MockObjects*. 02 2006. – URL <http://www.mockobjects.com>. – Zugriffsdatum: 13.02.2006 70
- [moreTV 2006] MORETV, Broadcasting G.: *moreTV*. 03 2006. – URL <http://www.more.tv/>. – Zugriffsdatum 01.03.2006 13
- [NIST 2006] NIST: *Government Information Preservation Working Group*. 02 2006. – URL <http://www.itl.nist.gov/div895/gipwog/index.html>. – Zugriffsdatum: 28.02.2006 12
- [OMG 2005] OMG: *Object Management Group*. 2005. – URL <http://www.omg.org/>. – Zugriffsdatum: 2005-09-19. – Zugriffsdatum: 06.02.2006 40, 115
- [prism54.org 2006] PRISM54.ORG: *The Prism54 Project*. 02 2006. – URL prism54.org/. – Zugriffsdatum: 19.02.2006 94

- [Raasch 1992] RAASCH, Jörg: *Systementwicklung mit Strukturierten Methoden*. 2. Hanser, 1992. – ISBN 3-446-17263-7 30, 33, 34, 62
- [redhat 2006] REDHAT: *Fedora Project, sponsored by Red Hat*. 02 2006. – URL <http://fedora.redhat.com/>. – Zugriffsdatum: 07.02.2006 93, 98
- [Schnedl 2004] SCHNEIDL, Christoph: *schnedl.at - IO-Warrior*. 04 2004. – URL <http://www.schnedl.at/iowarrior/>. – Zugriffsdatum: 06.02.2006 67
- [Sikora 2001] SIKORA, Axel: *Wireless LAN, Protokolle und Anwendungen*. Addison-Wesley, 2001. – ISBN 9-8273-1917-x 15, 30
- [SilverStone 2006] SILVERSTONE, Technology Co. L.: *SilverStone Technology Co., Ltd.* 03 2006. – URL <http://www.silverstonetek.com/>. – Zugriffsdatum: 06.03.2006 80, 107
- [Streitz und Nixon 2005] STREITZ, Norbert ; NIXON, Paddy: The Disappearing Computer. In: *Communications of the ACM* 48 (2005), März, Nr. Vol. 48, S. 33–35 21
- [Sun Microsystems 2005] SUN MICROSYSTEMS, Inc.: *Java Media Framework API*. 2005. – URL <http://java.sun.com/products/java-media/jmf>. – Zugriffsdatum: 06.02.2006 71, 74, 98, 99
- [Sun Microsystems 2006] SUN MICROSYSTEMS, Inc.: *Java Technology*. 02 2006. – URL <http://java.sun.com/>. – Zugriffsdatum: 19.02.2006 98
- [Tanenbaum und van Steen 2002] TANENBAUM, Adrew S. ; STEEN, Maarten van: *Distributed System Principles and Paradigms*. Prentice Hall, 2002. – URL <http://www.prenhall.com/tanenbaum/>. – ISBN 0-13-088893-1 10, 33
- [Tanenbaum 2000] TANENBAUM, Andrew S.: *Computernetzwerke*. 3. Pearson Studium, 2000. – ISBN 3-8273-7011-6 30
- [Telekom 2006] TELEKOM, Deutsche: *T-Com Haus*. 02 2006. – URL <http://www.t-com-haus.de>. – Zugriffsdatum: 01.03.2006 86
- [TerraTec 2006] TERRATEC, Electronic G.: *noxon 2 audio*. 03 2006. – URL <http://www.my-noxon.de/noxon.html>. – Zugriffsdatum: 01.03.2006 10, 22
- [TiVo 2006] TiVO, Inc: *TiVo.com | The TiVo Homepage*. 02 2006. – URL [TiVoInc..](http://www.tivo.com) – Zugriffsdatum: 28.02.2006 13
- [TwonkyVision 2006] TWONKYVISION: *TwonkyVision: All your media at your fingertips - any time, any place*". 02 2006. – URL <http://www.twonkyvision.de/>. – Zugriffsdatum: 28.02.2006 87

-
- [Wagner 2006] WAGNER, Thomas: *IO-Warrior-Java-Project*. 01 2006. – URL <http://www.wagner-ibw.de/iowj.html>. – Zugriffsdatum: 26.02.2006 72
- [Webmin 2006] WEBMIN: *Webmin*. 02 2006. – URL <http://www.webmin.com/>. – Zugriffsdatum: 26.02.2006 73, 100
- [Wulff 2005] WULFF, Lars: *Entwicklung eines verteilten Multimedia-Providers auf Basis von Web Services unter .NET (2005)*, Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit, 2005. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/diplom/wulff.pdf>. – Zugriffsdatum: 10.11.2005 21

A. Installation und Konfiguration des Media-Centers

A.1. Installation von Fedora Core 4

Fedora Core (redhat, 2006) ist der Nachfolger des Linux-Distribution Red Hat Linux. Die derzeit aktuelle Version ist die Fedora Core 4. Die Installation auf der im Design (siehe Abschnitt 5.2.6) ausgesuchten Hardware verlief ohne Probleme. Neben dem Standard-Paket wurden auch noch die Pakete für die Entwicklung von Software installiert. Die Festplatte wurde in fünf Teile (10GB,10GB,10GB,7GB,1GB) geteilt, um neben der Fedora Installation auch noch Platz für andere Betriebssysteme zu haben. Auf der zweiten Partition wurde für Testzwecke ein Windows XP installiert, die dritte Partition blieb leer und die vierte wurde als Tauschpartition eingerichtet. Die fünfte Partition ist die Swap Partition der Fedora. Nach der Installation wurde ein Online Update ausgeführt um das System auf den aktuellen Stand zu bringen.

A.2. Konfiguration der Software

In den folgenden Abschnitten wird beschrieben, wie ich das Betriebssystem (Fedora 4) angepasst habe, um die zusätzliche Hardware einzubinden, und die geschriebene Software darauf laufen zu lassen. Es wird auf die Konfiguration des Netzwerks eingegangen, danach werden erklärt, wie die Treiber für die Hardware, das Mainboard und den IOWarrior installiert wird. Danach wird auf die Installation der Media-Center Software eingegangen.

A.2.1. Netzwerk

Die Netzwerkkonfiguration bei der Distribution Fedora Core 4 ist Profil gestützt. Das Feature ist eigentlich mehr für Laptop gedacht, denn so kann man für verschiedene Standorte unterschiedliche Konfigurationen haben.

Die W-Lan Karte hat einen prism54 Chipsatz. Um diese Karte unter Linux zu installieren, braucht man die passende Firmware, die von der Webseite des prism Projekt (prism54.org, 2006) heruntergeladen werden kann. Die Firmware muss in „*lib/firmware/*“ gespeichert werden.

Um das Netzwerk zu konfigurieren, gibt es viele Möglichkeiten. Man kann eine GUI benutzen, die Befehle von Hand eintippen oder eine Konfiguration erstellen und dann hochladen. Da man für die Installation des Betriebssystem sowieso einen Monitor, Maus und Tastatur anschließen muss, kann man dann auch gleich das Netzwerk einrichten.

A.2.2. Hardware Treiber

Für das Übersetzen der Hardware Module werden die Header-Dateien des Kernels gebraucht, dieses RPM-Paket kann über yum (Paket „*kernel-devel.i586*“) installiert werden.

Hardware Monitor

Das VIA EPIA ME-6000 hat einen Sensor Chip, der es ermöglicht, die Temperatur, die Spannungen und die Drehzahl der Lüfter zu überprüfen. Leider wird dieser Chip im aktuellen Kernel nicht unterstützt, er wurde noch nicht von der alten Kernel Version zur neuen portiert. Lars Ekman (Ekman, 2006) hat die Änderungen an dem Modul gemacht, so dass man es unter dem Kernel 2.6.x einsetzen kann. Da diese Änderungen noch nicht in den Kernel aufgenommen wurden, muss das Modul von Hand erzeugt werden. Da zum Übersetzen mehrere Schritte nötig sind, habe ich ein kleines shell script geschrieben, das diese Arbeit erledigt.

```
#!/bin/sh
```

```
KERNEL_RE='uname -r '  
KERNEL_HEADER_HOME=/usr/src/kernels/${KERNEL_RE}-${KERNEL_ARCH}  
VT1211_HOME=/mediaCenter/vt1211  
VT1211_FILE=vt1211 - 2.6.14.2.patch
```

```
cd ${KERNEL_HEADER_HOME}
```

```
#patching the Kernel tree  
patch -p1 < ${VT1211_HOME}/${VT1211_FILE}
```

```
#make the oldconfig  
make oldconfig
```

```
#build the module
make drivers/hwmon/vt1211.ko

#if the dir does not ex. make it.
if [ ! -d /lib/modules/${KERNEL_RE}/extra ]; then
    mkdir /lib/modules/${KERNEL_RE}/extra/
fi
# remove old version
if [ -f /lib/modules/${KERNEL_RE}/extra/vt1211.ko ]; then
    rmmod vt1211.ko
    rm /lib/modules/${KERNEL_RE}/extra/vt1211.ko
fi
#copy the new module
cp drivers/hwmon/vt1211.ko /lib/modules/${KERNEL_RE}/extra/

#loading the module
depmod -ae
modprobe vt1211

#clean up
make clean
```

Mpeg2 Beschleuniger

Um zu testen, ob man auf dieser Hardware auch DVD abspielen kann, wurde auch die Software für den Mpeg2 Beschleuniger installiert. Die PRM-Pakete werden von Xavier Bachelot (Bachelot, 2006) gepflegt und sind über seine Webseite verfügbar. Damit die Pakete nicht immer erst von Hand heruntergeladen werden müssen, wird die Webseite einfach in das yum Repository aufgenommen. Dazu muss die Datei epia.repo, im Verzeichnis /etc/yum.repo.d, mit folgenden Inhalt angelegt werden:

```
[epia]
name=RPMs for Via Epia motherboards – Fedora Core $releasever
baseurl=http://washington.kelkoo.net/epia/FC$releasever
gpgcheck=0
enable=1
```

Da in diesem Repository nur die Direct Rendering Manager (kurz. DRM) Kernel Paket für die 686 Architektur verfügbar sind, aber das VIA EPIA ME-6000 Mainboard noch eine älteren C3 (Samuel 2) Prozessor hat, passt das Paket nicht und muss aus dem Quellpaket selber erzeugt werden. Da dieses Paket für jeden Kernel angepasst werden müsste, habe ich folgendes shell script geschrieben, um die Arbeit zu automatisieren.

```

#/bin/sh

TMP_DIR=/tmp
LOGFILE=/tmp/mod-update

KERNEL_ARCH=i586
KERNEL_RE='uname -r '

SRC_HOME=/usr/src
RPM_HOME=${SRC_HOME}/redhat

DRM_SRPM_DOWNLOAD=http://washington.kelkoo.net/epia/FC4/SRPMs/
DRM_SRPM_NAME=kernel-module-via-drm
DRM_SRPM_RE=2.6.14-1.1637_FC4
DRM_RE=2.7.4-1.ucr
DRM_SRPM=${DRM_SRPM_NAME}-${DRM_SRPM_RE}-${DRM_RE}

cd ${TMP_DIR}
echo -n "DRM: Download the DRM SRC: "
wget ${DRM_SRPM_DOWNLOAD}${DRM_SRPM}.src.rpm -a ${LOGFILE}
if [ ! -f ${DRM_SRPM}.src.rpm ]; then
    echo FAIL
    exit
fi
echo "OK"
rpmbuild --quiet --rebuild --target ${KERNEL_ARCH} --define "kernel ${DRM_SRPM_NAME}-${DRM_SRPM_RE}-${DRM_RE}" ${DRM_SRPM}.src.rpm

if [ -f ${RPM_HOME}/RPMS/${KERNEL_ARCH}/${DRM_SRPM_NAME}-${DRM_SRPM_RE}-${DRM_RE}.${KERNEL_ARCH}.rpm ]; then
    echo "DRM: The rpm was build"
    echo -n "DRM: Install RPM: "
    rpm -Uhv --quiet ${RPM_HOME}/RPMS/${KERNEL_ARCH}/${DRM_SRPM_NAME}-${DRM_SRPM_RE}-${DRM_RE}.${KERNEL_ARCH}.rpm
    echo "Done"
    echo -n "DRM: Now cleaning up: "
    rm -f ${RPM_HOME}/RPMS/${KERNEL_ARCH}/${DRM_SRPM_NAME}-${DRM_SRPM_RE}-${DRM_RE}.*
    rm ${DRM_SRPM}.src.rpm
    echo "Done"
else
    echo "DRM: The rpm was not build"
cd -

```

Über yum können nun die Pakete für die grafische Oberfläche und dem DVD-Player Xine installiert werden

```
yum update xorg-x11*
yum install xine-lib*
```

Kernel Module für den IOWarrior24

Um den IOWarrior unter Linux in Betrieb zu nehmen, muss ein Kernel-Modul erstellt werden, dazu braucht man das SDK für Linux des Herstellers Code Mercenaries (Mercenaries, 2005). Nach dem Auspacken wechselt man in das Verzeichnis „LinuxSDK/Kernel/Driver,“ übersetzt das Modul und installiert es.

```
cd LinuxSDK/Kernel/Driver
make
make install
```

Nach der Installation des Modules muss noch die Berechtigung für den Zugriff auf den IOWarrior gesetzt werden. Der Zugriff auf das Modul wird für normale Benutzer erlaubt (siehe Abschnitt 5.2.5).

```
echo "BUS=\"usb\", KERNEL=\"iowarrior[0-9]*\", NAME=\"usb/\\%k\" MODE=\"0666\" >> /etc/udev/rules.d/iowarrior.rules
```

Beim Testen stellte sich heraus, dass die aktuelle SDK (Datum 21.7.2005) nicht mit dem Linux Kernel 2.6.15.x zurecht kommt. Auf Nachfrage beim Hersteller erhielt ich die Antwort, dass man sich darum kümmern würde und, es demnächst ein Update geben wird.

Kernel update

Da die oben genannten Treiber immer auf den installierten Kernel angepasst werden müssen, habe ich ein kleines Shell Script geschrieben, was dies Schritte ausführt. Das Script ist auf der CD enthalten (siehe Anhang D). Das Script muss auf die lokalen Gegebenheiten angepasst werden. Es ist in zwei Bereiche aufgeteilt, dem Konfigurationsteil am Anfang der Datei und dem Ausführungsteil hinter der Konfiguration. Im Konfigurationsteil müssen die Variablen an die lokalen Gegebenheiten angepasst werden. Die Variablen zeigen auf die Verzeichnisse, in denen sich die Treiber befinden. Der Ausführungsteil ist eine Automatisierung der o.g. Schritte.

A.2.3. Java

Java gehört nicht zum Lieferumfang der Fedora Distribution. Die Anleitung bezieht sich auf das Java 2 Platform Standard Edition Development Kit 5.0 Update 6 (kurz JDK) und das Betriebssystem Fedora Core 4 (redhat, 2006). Die Voraussetzung zur Installation von Java ist das RPM Paket „rpm-build“, mit dem man RPM Paket erstellen kann. Installiert wird das Paket über yum.

```
yum install rpm-build
```

1. Die JDK von der sun Webseite (Sun Microsystems, 2006) herunterladen und in dem Verzeichnis `/usr/src/redhat/SOURCES/` speichern.
2. Um aus dem Binärfile eine RPM Paket zu erstellen braucht man das passende Quellpaket „java-1.5.0-sun“ von JPackage (Corley, 2006). Das Paket muss heruntergeladen und in ein beliebiges Verzeichnis gelegt werden.

3. Jetzt können die RPM Paket erstellt werden, mit dem Befehl:

```
rpmbuild --rebuild java-1.5.0-sun-1.5.0.06-1jpp.nosrc.rpm
```

4. Die Pakete werden im `/usr/src/redhat/RPMS/i586/` erzeugt und können dann mit folgendem Befehl installiert werden

```
rpm -ihv /usr/src/redhat/RPMS/i586/java-1.5.0-sun-*.rpm
```

5. Zu Schluß muß dann noch die Java Umgebung im System bekannt gemacht werden, das passiert mit dem Befehl:

```
/usr/sbin/alternatives --config java
```

Java Media Framework einrichten

Das Archiv „jmf-2_1_1e-linux-i586.bin“ von der Java Media Framework (kurz JMF) Webseite (Sun Microsystems, 2005) runterladen und in Verzeichnis `/usr/java/` kopieren. Die Datei muß ausführbar gemacht werden und ausgeführt werden.

```
cp jmf-2_1_1e-linux-i586.bin /usr/java/  
cd /usr/java/  
chmod u+x jmf-2_1_1e-linux-i586.bin  
./jmf-2_1_1e-linux-i586.bin
```

Nachdem das Installationsprogramm die Dateien ausgepackt hat, müssen die Bibliotheken und Java Archive noch ins System integriert werden. Um die Java Archive im System bekannt zu machen, gibt es zwei Möglichkeiten.

1. Das Java Archive „*jmf.jar*“ aus dem lib Verzeichnis des JMF in den Java \$CLASSPATH übernehmen.

```
export JMF_HOME="/usr/java/JMF-2.1.1e"
export CLASSPATH="JMF_HOME/lib/jmf.jar:${CLASSPATH}"
```

2. Alle Java Archive (*.jar) Dateien aus dem lib Verzeichnis des JMF in das Extension-Verzeichnis der Java Runtime Java Runtime Environment (kurz JRE) (hier /usr/java/jdk1.5.0_05/jre/lib/ext) kopieren.

Ich habe mich für die zweite Lösung entschieden, da nicht alle Programme (z. B. der Mozilla Firefox) die JMF Klassen gefunden haben, wenn sie nur im CLASSPATH eingetragen waren. Nach dem die Java Archive eingerichtet sind, müssen noch die JMF Bibliotheken im System eingerichtet werden, dazu muss der Pfad zum lib Verzeichnis des JMF in die Konfigurationsdatei des „*dynamic linker/loader*“ eingetragen werden.

```
export JMF_HOME="/usr/java/JMF-2.1.1e"
echo "$JMF_HOME/lib" >> /etc/ld.so.conf
export LD_LIBRARY_PATH = ${JMF_HOME}/lib:${LD_LIBRARY_PATH}
#änderungen im System bekannt machen
ldconfig
```

Zu guter letzt muss noch man die „*jmfdir*“ Datei im Home-Verzeichnis des Users anlegen mit dem Pfad zum JMF als Inhalt.

```
echo "JMF_HOME/lib" >> ~/.jmfdir
```

Ob das Einrichten erfolgreich war sieht man, wenn man mit dem Browser die JFM Diagnose Webseite (Sun Microsystems, 2005) aufruft. Damit man auch MP3 Datien wiedergeben kann, muss jetzt das Java MP3 Plugin installiert werden. Dazu muss das MP3 Plugin von der JMF Webseite heruntergeladen werden und die Datei „*mp3plugin.jar*“ ins Extension-Verzeichnis der Java JRE kopiert werden. Damit der Plugin mit dem JMF zusammenarbeitet, muss man ihn mit dem folgenden Befehl im Java Media Framework bekannt machen.

```
java com.sun.media.codec.audio.mp3.JavaDecoder
```

Mit der Java Anwendung „*JMStudio*“ kann nun überprüft werden, ob das JMF richtig installiert ist.

```
java JMStudio
oder mit
java -Dawt.toolkit=sun.awt.motif.MToolkit JMStudio $*
```

ID3Tags

Das Java Archiv (jar) von der Webseite der Entwickler ((ID3v2, 2006)) muss entweder in das Extension-Verzeichnis der JRE kopiert werden, oder es muss der Runtime über die Variable „*CLASSPATH*“ bekannt gemacht werden.

A.2.4. Webmin

Um Webmin zu installieren, muss das RPM von der Webseite (Webmin, 2006) heruntergeladen werden, danach wird das Paket mit rpm installiert.

```
rpm -i webmin-1.260-1.noarch.rpm
```

Jetzt kann man über einen beliebigen Browser auf das Webinterface zugreifen. Jetzt muss noch die Konfigurationssoftware für die Set-Top-Box installiert werden, das passiert über das Webinterface. Im Menüpunkt „*Webmin* → *Webmin Configuration* → *Webmin Modules*“ kann man die Module verwalten. Hier kann dann das Webmin-Modul von der CD installiert werden.

A.2.5. Zugriff auf die Netzwerkfreigaben

Auf die Netzwerkfreigabe wird über das smb Protokoll zugegriffen, deshalb muss das RPM-Paket „*samba-client*“ installiert sein. Um während der Entwicklung der Software den Datenaustausch mit der Set-Top-Box zu vereinfachen, wurde auf der Set-Top-Box ein Samba Server installiert. Freigegeben wurden die Partition, die für den Austausch eingerichtet wurde.

A.2.6. Media-Center Software

Während der Diplomarbeit wurden eine Machbarkeitsstudie und ein Prototyp entwickelt. Auf die Installation der Machbarkeitsstudie wird hier nicht weiter eingegangen. Der Quellcode der Studie befindet sich auf der CD (siehe Anhang D). Das Verzeichnis mit dem Prototypen muss von der CD auf die Festplatte kopiert werden. In dem Verzeichnis befindet sich das Start Skript „*mediaCenter.sh*“. Dieses Skript muss an die lokalen Gegebenheiten angepasst werden. Im Kopfbereich des Skriptes müssen z.B. die Variablen für die JRE gesetzt bzw. überprüft werden. Wenn das Media-Center bei Start des PC mit gestartet werden soll, muss das Skript in das Verzeichnis */etc/init.d* kopiert werden und dann mit dem folgenden Befehl bekannt gemacht werden.

```
chkconfig --level 345 mediaCenter.sh on
```

B. Platinen für den IOWarrior

Wären der Diplomarbeit habe ich zwei Version der Interaktionskomponente gebaut. Die erste Version basierte auf dem „*Starter Kit*“ der Herstellerfirma Code Mercenaries (Mercenaries, 2005). Die zweiten Version ist eine wWiterentwicklung der erst, hier wurden nur noch die nötige Funktionen eingebaut..

B.1. Machbarkeitsstudie

Die erste Version war für die Machbarkeitsstudie, diese Schaltung basiert auf dem „*Starter Kit*“ von Code Mercenaries. Das Modul besteht aus zwei Platinen, der Basisplatine und dem Decoderplatine. Der Schaltplan der Basisplatine ist identisch mit dem Kit, das daraus erstellte Platinenlayout habe ich selbst entwickelt. Die Pläne für die Machbarkeitsstudie befinden sich auf der CD (siehe Anhang D).

B.2. Der Prototyp

Der Prototyp ist die Version, die in das Gehäuse fest eingebaut wurde. Da zum Zeitpunkt der Weiterentwicklung noch nicht klar war, wie die Platine, das Display und die Taster ins Gehäuse eingebaut werden können, habe ich mich wieder für eine Lösung mit 2 Platinen entschieden. Auf der Ansteuerungsplatine sitzt der IO-Warrior und auf der zweite Platine (für die Frontplatte) ist der Infrarotempfänger und die Entprellschaltung für die Taster untergebracht.

B.3. Ansteuerungsplatine

Die Abbildung B.1 zeigt den reduzierten Schaltplan der Machbarkeitsstudie. An JP1 wird die Platine mit dem USB verbunden. JP2 muss ein Jumper auf die Pins 1 und 2 oder 2-3 gesetzt werden. 1-2 ist der low-power-mode, dann reserviert der IO-Warrior bei der USB

Hub 100mA für sich, bei den Pins 2-3 (high-power-mode) wären es 500mA. Wie der Jumper gesetzt wird, kommt auf den Stromverbrauch des eingestetzten Displays an. Die Pinleiste JP3 ist der Anschluß für die Frontplatine und JP4 ist das Anschluss für das Display.

Bezeichnung	Wert	Anzahl
C1	100 η F (RM:2,5)	1
C2	10 μ F (RM:2,5)	1
IC1	IO-WARRIOR24	1
IC2	74HC148N	1
JP1	USB-Einbaubuchse	1
JP2	1X03	1
JP3	2X07	1
JP4	2X08	1
Q1	BC307	1
R1	1,3k Ω (RM:10)	1
R2	4,7 Ω (RM:10)	1
R11-R18	1k Ω (RM:10)	8

Tabelle B.1.: Prototyp: Stückliste der Basisplatine

B.3.1. Frontplatine

An dieser Platine werden die Taster und der Drehimpulsgeber angeschlossen. Aus Zeitgründen wurde die Platine für die Frontplatte auf einer Lochrasterplatine aufgebaut. Den Schaltplan zeigt die Abb. B.4. Für die Verdrahtung siehe Tabelle B.2.

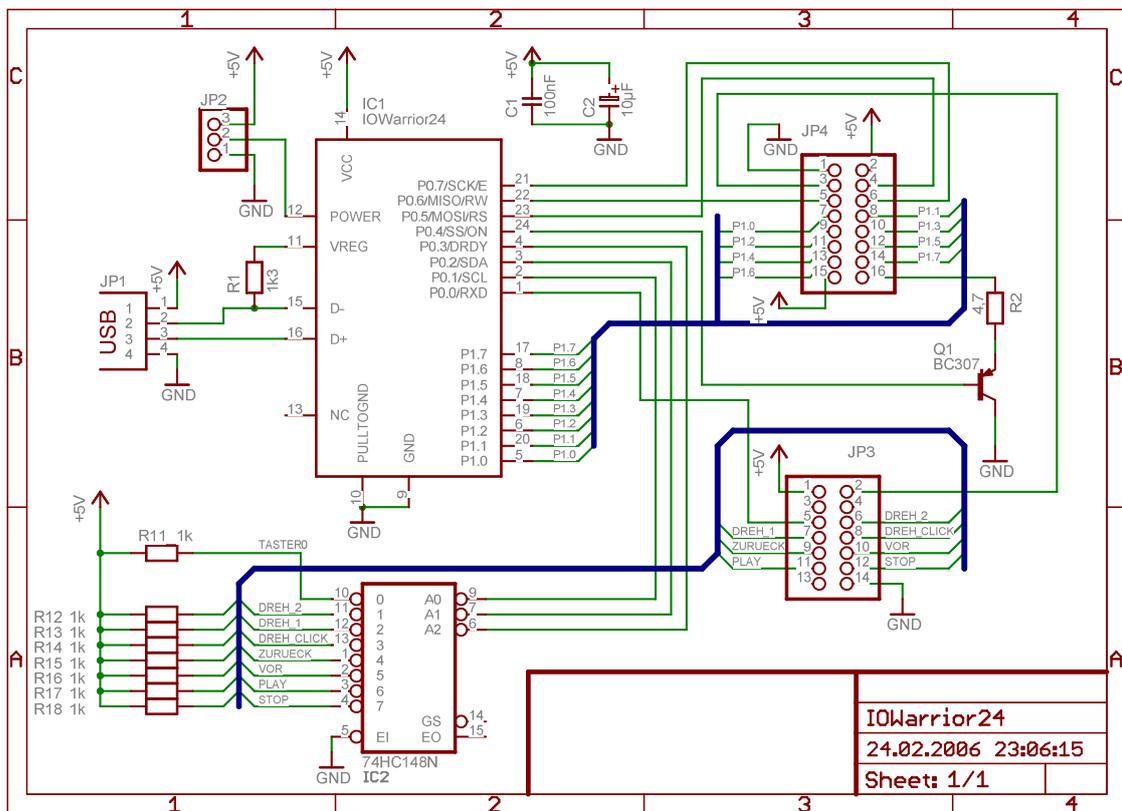


Abbildung B.1.: Schaltplan des Prototypen

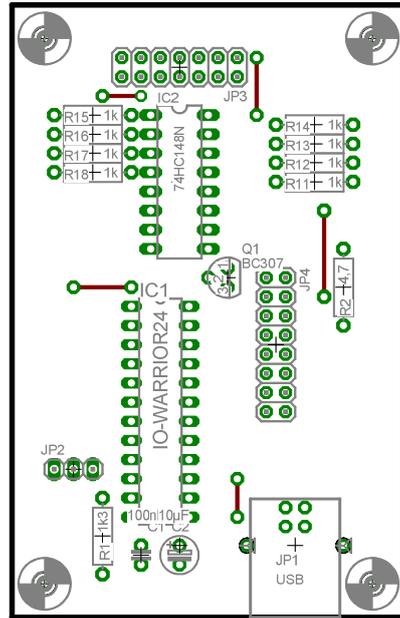
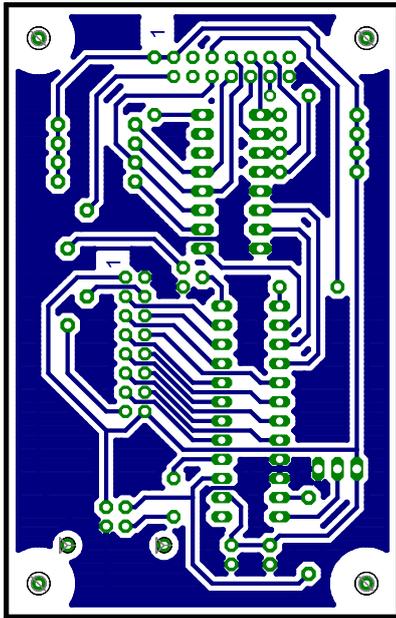


Abbildung B.2.: Platine (von der Lötseite) des Prototypen

Abbildung B.3.: Bestückungsplan der Prototypplatine

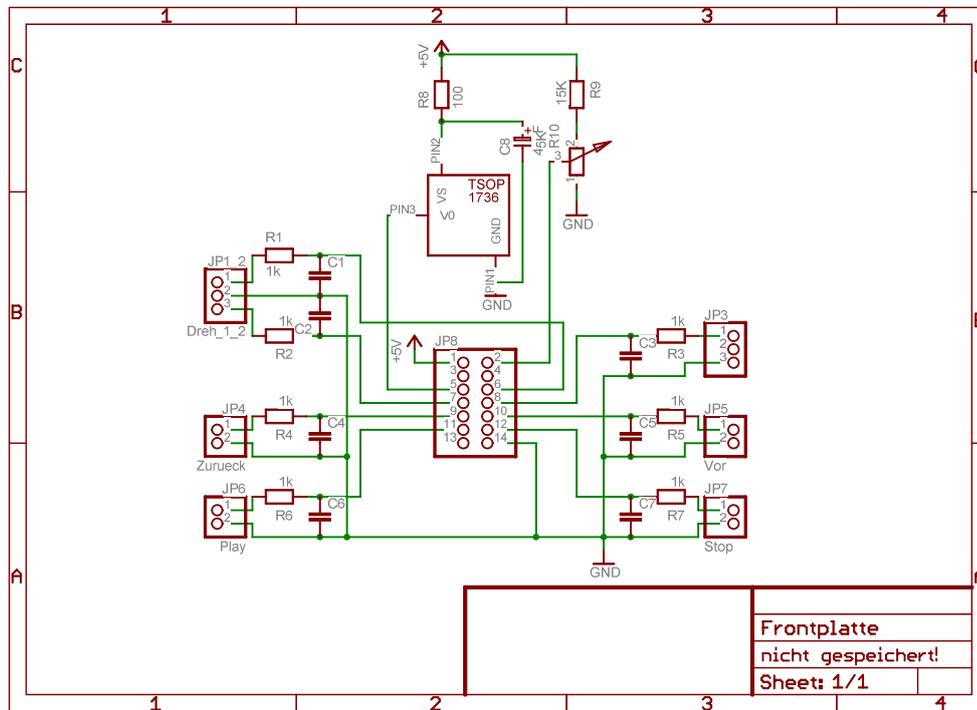


Abbildung B.4.: Schaltplan der Frontplatte

Steckverbinder	Pin	Bauteil	Pin
JP1_2	1	Drehimpulsgeber	A
JP1_2	2	Drehimpulsgeber	C
JP1_2	3	Drehimpulsgeber	B
JP3	1	Drehimpulsgeber	P1
JP3	2	Drehimpulsgeber	P2
JP4	1	Taster Zurück	1
JP4	2	Taster Zurück	2
JP5	1	Taster Vor	1
JP5	2	Taster Vor	2
JP6	1	Taster Play	1
JP6	2	Taster Play	2
JP7	1	Taster Stop	1
JP7	2	Taster Stop	2

Tabelle B.2.: Prototyp: Verdrahtungsplan

Bezeichnung	Wert	Anzahl
C1-C7	100nF (RM:2,5)	7
C8	4,7µF (RM:2,5)	1
JP1_2, JP3	1X03	2
JP3-J8	1X02	6
R1-R7	1kΩ (RM:10)	7
R8	100Ω (RM:10)	1
R9	15kΩ (RM:10)	1
R10	5kΩ	1
IC1	TSOP1736	1

Tabelle B.3.: Prototyp: Stückliste der Frontplatine

C. Umbau des Gehäuses

C.1. Stückliste

In der folgenden Tabelle sind die benötigten Teile aufgeführt.

Anzahl	Bezeichnung	Bermerkung
1	Gehäuse	SilverStone LC02S
1	Mainboard	VIA EPIA ME-6000
1	Laptop-Festplatte	40GB
1	DVD-Laufwerk	Silm-Line Bauform
1	W-Lan-PCI-Karte	SMC 54MBit/s prism54 Chipsatz
1	Raiser-Karte	siehe Anhang C.2
1	Winkel	siehe Anhang C.4
1	LC-Display 20x4	Electronic Assembly: EA W204B-NLW
1	LCD-RAHMEN	
1	Frontplatine	siehe Anhang B.3.1
1	Ansteuerungsplatine	siehe Anhang B.3
4	Taster	APEM AV0630C90
1	Drehimpulsgeber 427	ddm hopt schuler:427-021101AL001
1	Flachbandkable 14-pol.	Ansteuerungsplatine ↔ Frontplatine
1	Flachbandkable 16-pol.	Ansteuerungsplatine ↔ LC-Display
1	USB-Kabel	Ansteuerungsplatine ↔ Mainboard
x	Montagematerial	Schraube, Mutter, Abstandhalter und Kabel

Tabelle C.1.: Stückliste

C.2. Montage W-Lan Karte

Beim Einbau der W-Lan Karte gab es Probleme mit den dem Gehäuse beigelegten Raiser Platinen (siehe Bild). Da das Gehäuse für den Einbau eine mini-ATX Mainboard mit AGP-Steckplatz gedacht war, passten die beigelegten Raiser Platien (siehe Abbildung C.1) nicht

für den PCI Steckplatz, Abhilfe schaffte eine Raiser Karte mit flexiblem Kabel (siehe Abbildung C.2).

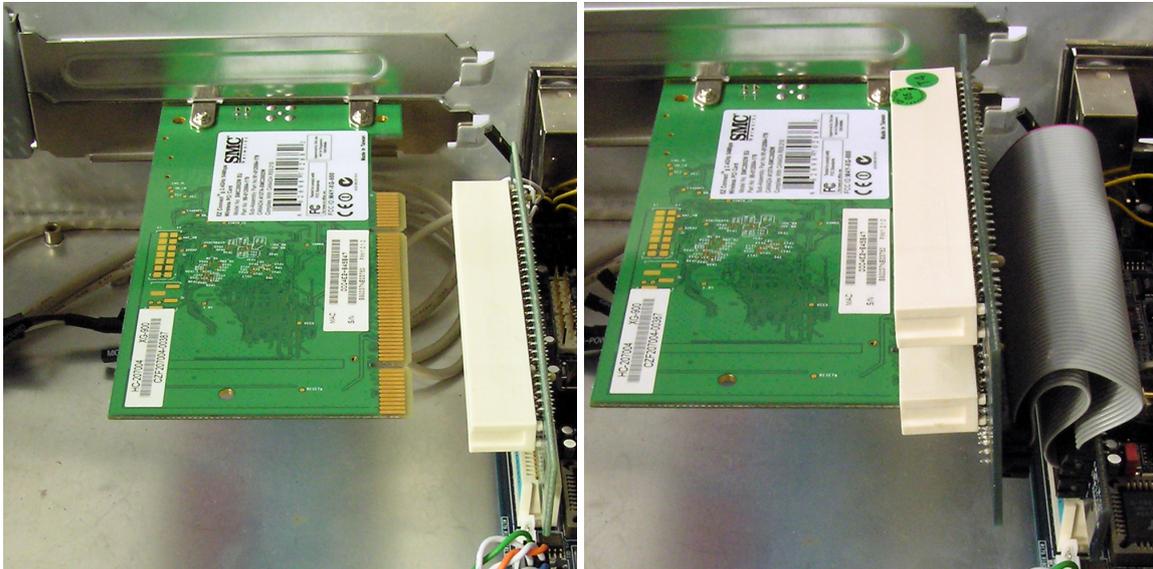


Abbildung C.1.: Beigelegte Raiser Karte

Abbildung C.2.: Nachgekaufte WLAN-Karte 1

C.3. Umbau des Gehäuses

Wie schon in Abschnitt 5.2.6 beschrieben, mussten ein paar Änderungen am SilverStone LC02S Gehäuse (SilverStone, 2006) gemacht werden. In die Blende mussten Bohrungen für die Taster und ein Ausschnitt für Display gefräst werden. Die Zeichnung C.3 zeigt die Blende. In die Frontplatte mussten Aussparung für das Display, die Taster und die Frontplatte gemacht werden (Zeichnung C.4). Die fertige Blende zeigt die Abbildung C.5. Die Abbildung C.6 zeigt das offene Gehäuse der Set-Top-Box.

C.4. Winkel

Wie schon in Abschnitt 5.2.6 geschrieben, gab es Temperaturprobleme nach dem der Gehäuselüfter ausgebaut wurde. Um das Problem zu umgehen, wurde ein Winkelblech ins Gehäuse eingepasst. Die Abbildung C.7 zeigt die Technische Zeichnung des Winkels. Abbildung C.8 zeigt das Gehäuse ohne Winkel und C.9 mit.

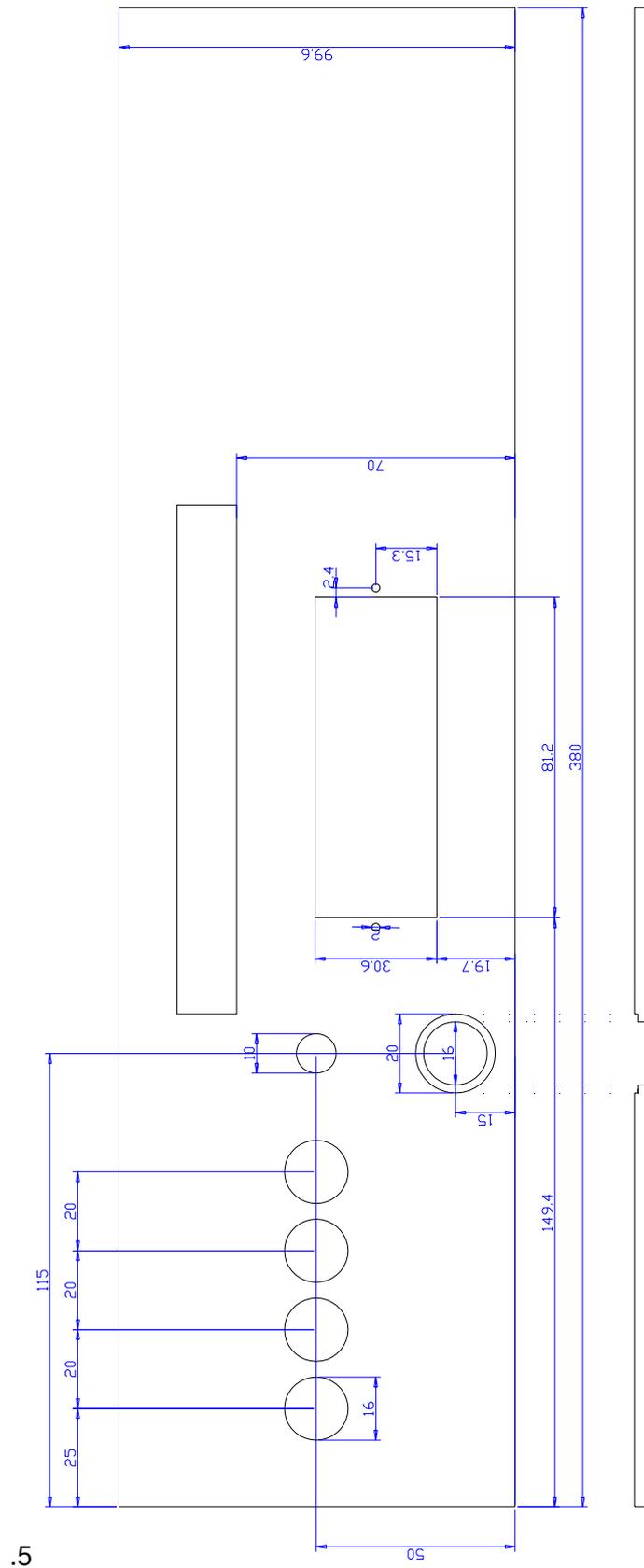


Abbildung C.3.: Technische Zeichnung der Blende M 1:2

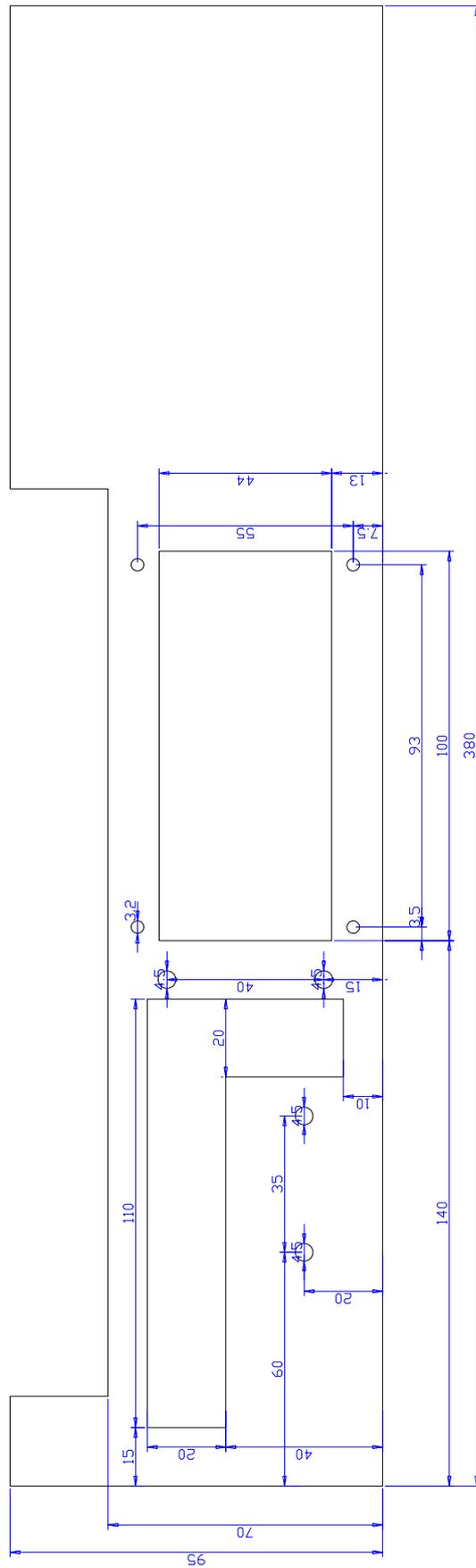


Abbildung C.4.: Technische Zeichnung der Frontplatte M 1:2



Abbildung C.5.: Bild der Blende

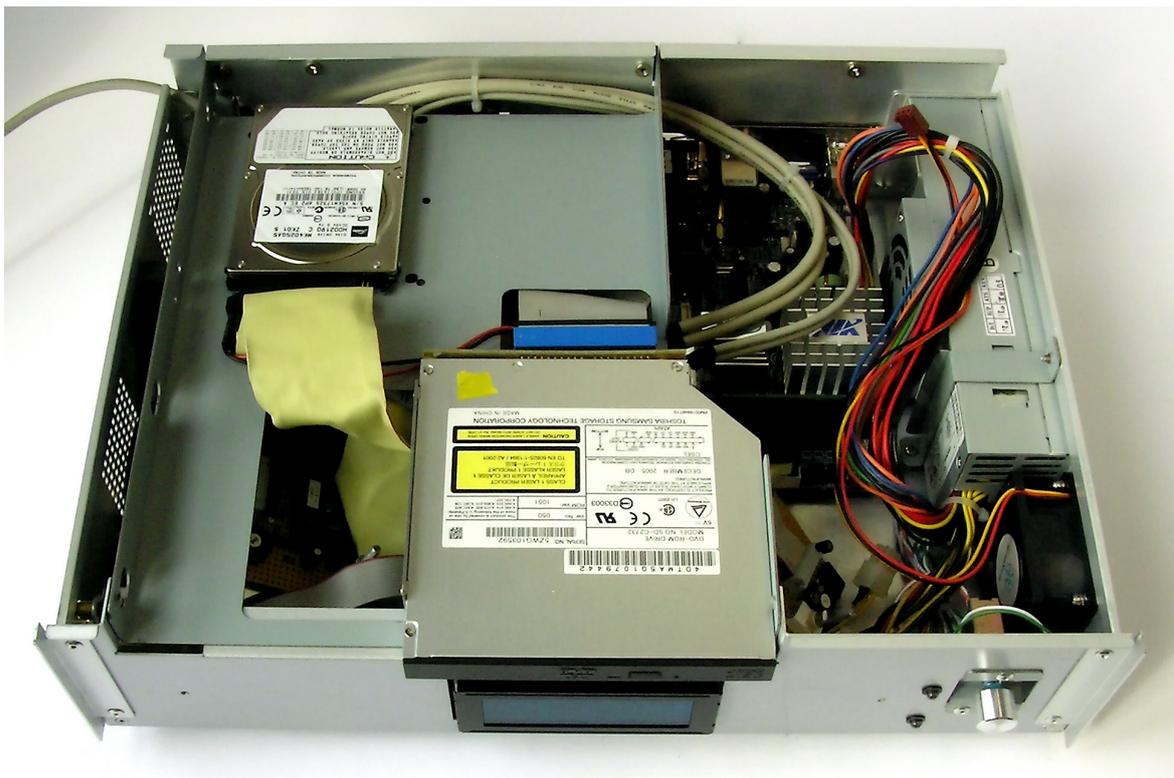


Abbildung C.6.: Die Set-Top-Box offen.

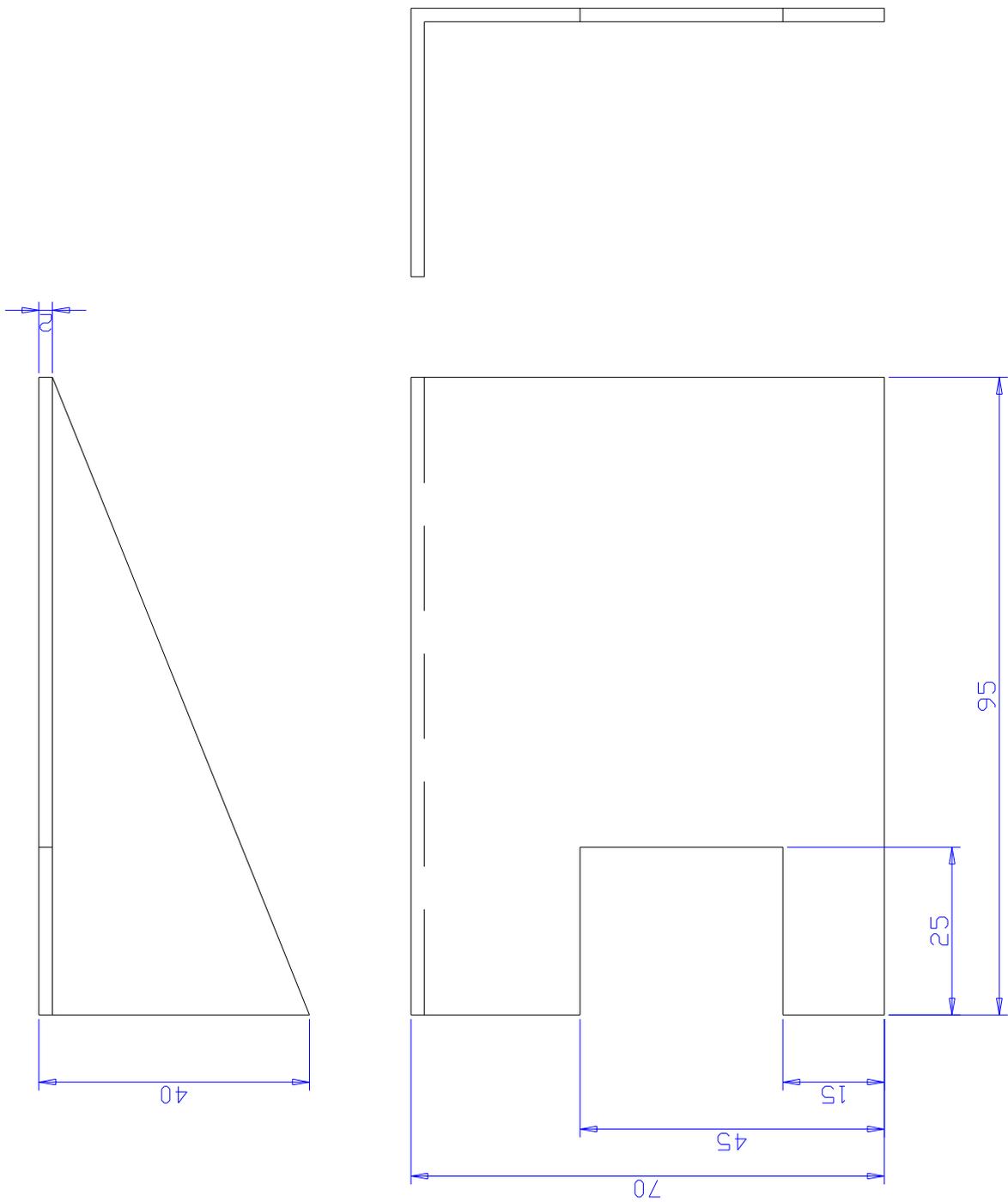


Abbildung C.7.: Der Winkel für die Lüftung

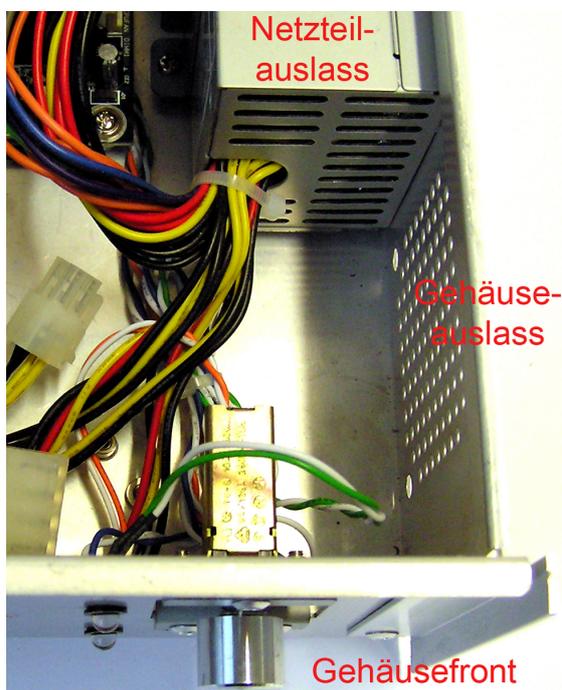


Abbildung C.8.: Gehäuse ohne Winkel

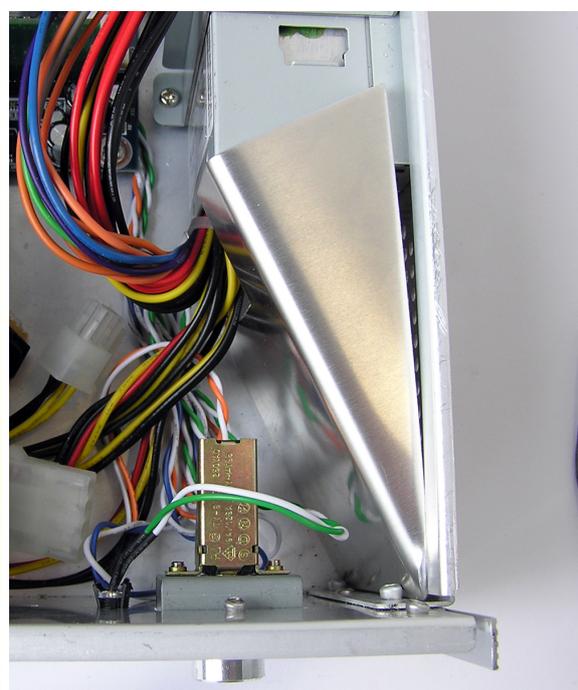


Abbildung C.9.: Der Winkel für die Lüftung

D. Inhalt der CD

Die beigelegte CD ist wie folgt aufgeteilt:

Ordner	Inhalt
Diplomarbeit	Enthält diese Diplomarbeit als PDF-Dokument
Machbarkeitstudie	Machbarkeitstudie
..src	Quellcode
..lib	benötigte Bibliotheken
..pic	Bilder
..doc	Dokumentation
..tech	Schaltpläne und Layouts der Platine
Prototyp	Prototyp
..src	Quellcode
..lib	benötigte Bibliotheken
..pic	Bilder
..doc	Dokumentation
..tech	Schaltpläne und Layouts der Platine
..MediaCenter	Jar des Media-Centers
Linux	
..sh	Skripte für die Treiber
..software	Java, JFM, ID3, IOWarrior SDK, Webmin und vt1211

Tabelle D.1.: Inhalt der CD

Glossar

.NET Framework Das .NET-Framework ist eine Implementierung der CLI durch den Hersteller Microsoft. Damit stellt das Framework die Basis bereit zur Ausführung von Programmen, die in verschiedenen Programmiersprachen geschrieben wurden. Weitere Infos unter (Microsoft, 2006b)

CLI Die CLI Common Language Infrastructure ist ein Standard, der ein System spezifiziert, das es ermöglicht, Anwendungen auszuführen, die in unterschiedlichen Programmiersprachen geschrieben wurden. (Microsoft, 2006c)

disappearing computer Der Computer verschwindet vor den Augen der Anwender. Ein Synonym für den versteckten Computer. Computer tauchen in immer mehr alltäglichen Geräten auf, ohne dass die Bediener es bemerken.

iPod Ein tragbarer Musikspieler der Firma Apple (Apple Computer, 2006a). Der Begriff iPod hat sich zu einem Synonym für tragbare Musikspieler entwickelt.

LCD Ein LCD (liquid cystal display bzw. Flüssigkristallbildschirm) ist ein elektrisches Anzeigegerät, das mit Hilfe von Flüssigkristallen Symbole darstellen kann. Hier ist LC-Display ein Synonym für ein kleines Display, mit dessen Hilfe Metainformationen für den Anwender dargestellt werden.

leichtgewichtige Media-Center sind Media-Center, die mit möglichst wenig Hardware auskommen und so eine sehr kompakte Bauform nutzen. Dabei wird bestehende Hardware (Media-Server) vorausgesetzt.

Media-Center Ein Gerät, das in der Lage ist, digitale Inhalte wiederzugeben.

Mock-Objects Mock-Objects sind Platzhalter für echte Objekte. Benutzt werden diese Platzhalter innerhalb von Unit-Tests, um noch nicht geschriebene Module zu emulieren oder um Verhalten zu simulieren, das nur schwer nachzubilden ist (z. B. spontane Netzwerkfehler).

- mono** Genau wie das .NET-Framework Framework ist Mono eine Implementierung der „Common Language Infrastructure“ (Microsoft, 2006c). Mono steht unter der GNU Public License und läuft auf allen gängigen Betriebssystemen.
- NAS** NAS ist die Kurzform für Network Attached Storage. NAS sind reduzierter abgespeckte Dateiserver für den Heimbereich. Im Vergleich sind sie billiger und einfacher zu installieren, bieten aber meist keine Möglichkeit der Erweiterung (Speicherplatz oder Software). Durch die vereinfachte Administration bieten sie auch nur geringe Möglichkeiten, den Zugriff der Anwender auf Dateien einzuschränken.
- OMG** OMG steht für Object Management Group. Die OMG ist ein Konsortium von derzeit über 800 Mitgliedern, das sich mit der Entwicklung von Standards für Objektorientierte Programmierung beschäftigt. Genauere Informationen findet man unter (OMG, 2005)
- PDA** Ein Personal Digital Assistant (engl. persönlicher digitaler Assistent) ist ein kleiner tragbare Computer mit persönlichen Daten (z. B. Termine, Daten)
- Player** Eine eigenständige Software, die Medieninhalte wiedergibt.
- Playlist** Ein Liste von Medienobjekten, die von einem Player in einer bestimmten Reihenfolge wiedergegeben werden.
- Schwergewichtiges Media-Center** Ein Schwergewichtiges Media-Center ist ein leistungsstarker PC, der mit zusätzlicher Interaktionshardware ausgerüstet ist. Es wird dabei versucht, auf Eingabekomponenten wie Tastatur und Maus zu verzichten. Als Anzeige dient meist der Fernseher.
- SDK** Ein Software Development Kit ist eine Ansammlung von Bibliotheken, Dokumentation und Beispielprogrammen. Diese Sammlung soll es Entwicklern ermöglichen, eigene Anwendungen oder Erweiterung einer Anwendung auf Basis dieser SDK zu schreiben.
- Set-Top-Box** Hier: Ein Synonym für ein leichtgewichtiges Media-Center
- SoHo** Mit SoHo (Small office /Home office) klassifiziert man kostengünstige Hard- und Software, die für den Einsatz in kleinen Firmen und im Heimbüro eingesetzt wird. SoHo Produkte sind leicht einzurichten, da auf eine komplexe Konfiguration verzichtet wird.

Unit-Tests Unit-Tests sind Teil des Vorgehensmodells des Extreme Programming. Sie dienen zum Überprüfen, ob ein Softwaremodul korrekt arbeitet. Es werden vor und während der Entwicklung Testfälle geschrieben. Nach Änderungen am Quellcode lässt man dann die Testfälle durchlaufen.

UPnP Universal Plug and Play ist das erweiterte Plug and Play von Microsoft (Microsoft, 2006a). Es ist eine Schnittstelle zur Ad-hoc-Kommunikation zwischen den unterschiedlichsten Diensten und Geräten. Mehreren UPnP-Geräten ist es möglich, sich selbst zu konfigurieren und so ohne Eingriff des Anwenders in Verbindung zu treten. So soll es möglich sein, mit einem PDA die Stereoanlage zu steuern ohne Software zu installieren.

W-lan Ein W-Lan (Wireless Local Area Network) ist ein „drahtloses“ lokales Netzwerk.

Wiedergabeliste siehe Playlist.

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 20. März 2006

Ort, Datum

Unterschrift