

DISSERTATION

Dynamische Kopplung von Gerätenetzwerken im Heimbereich

ausgeführt zum Zwecke der Erlangung des akademischen Grades
Doktor der technischen Wissenschaften

Erstbegutachter

O. Univ. Prof. Dipl.-Ing. Dr. techn. Dietmar Dietrich
Institut für Computertechnik, TU Wien

Zweitbegutachter

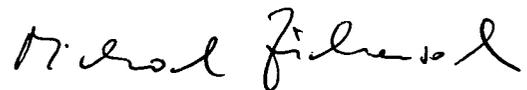
Ao. Univ. Prof. Dipl.-Ing. Dr. techn. Wolfgang Kastner
Institut für Rechnergestützte Automation, TU Wien

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

von

Mag. Dipl.-Ing. Michael Ziehensack
Jägerstraße 69/2/20, 1200 Wien
Matr. Nr. 9225075

Wien, März 2004



Kurzfassung

Die Zielsetzung dieser Arbeit ist die Untersuchung der verschiedenen Möglichkeiten zur dynamischen Kopplung von Gerätenetzwerken im Heimbereich. Auf Grund der verschiedenen Anwendungsbereiche im Heim und der daraus resultierenden Anforderungen gibt es unterschiedliche Gerätenetzwerke, wie beispielsweise den Europäischen Installations Bus (EIB), das IEEE 1394 basierte Home Audio/Video interoperability (HAVi) Netzwerk oder das TCP/IP basierte Universal Plug and Play (UPnP) Netzwerk. Entsprechend der Fortführung des Vernetzungsgedankens, demzufolge der Nutzen eines Gerätenetzwerks größer ist als die Summe des Nutzens der einzelnen Gerätefunktionen, kann durch die Kopplung der Gerätenetzwerke ein zusätzlicher Nutzen (einfachere Bedienbarkeit, gemeinsame Nutzung von Ressourcen, neue Anwendungen, ...) erzielt werden. Da es jedoch kein allgemeines Modell zur Kopplung von Gerätenetzwerken im Heimbereich gibt, ist der Vergleich von Kopplungen beziehungsweise die Suche nach geeigneten Implementierungen sehr aufwendig, und es fehlen Richtlinien oder Kriterien für die Entwicklung von neuen Kopplungen.

In dieser Arbeit wird ein allgemeines Modell zur Kopplung von Gerätenetzwerken im Heimbereich definiert. Wesentlicher Bestandteil des entwickelten Kopplungsmodells ist eine Gerätearchitektur, welche die für die Vernetzung notwendigen Dienste definiert und diese in drei Schichten gruppiert. Eine Kopplung kann durch verschiedene Gateways (System-Gateway, Service-Gateway, UI-Gateway) zur Umsetzung auf einer dieser Schichten oder ein nichtumsetzendes, mehrfach angebundenes Gerät erfolgen. Außerdem können verschiedene Arten von Kopplungen (einseitig/zweiseitig, direkt/indirekt, statisch/dynamisch, ...) unterschieden werden. Des Weiteren ergibt der gemeinsame Einsatz von mehreren Kopplungen in Form eines Kopplungssystems weitere Kopplungsvarianten (AB-Kopplung, C-Kopplung, T-Kopplung). Eine geeignete Kopplung muss jedenfalls dynamisch sein, weil nur diese allgemein einsetzbar ist und ohne wesentlichen Konfigurationsaufwand auskommt. Schließlich werden anhand des Modells siebzehn Kopplungen analysiert und am Beispiel der Unterhaltungselektronik sechs verschiedene dynamische Kopplungen – drei erstellte Konzepte und drei durchgeführte Implementierungen – beschrieben.

Stichwörter: Netzwerk, Kopplung, Modell, Gateway, HAVi, intelligentes Heim

Abstract

The thesis deals with dynamic interconnection of networks in the home environment. Due to the different demands on networks in the home there exist several kinds of networks, like the European Installation Bus (EIB), the IEEE 1394 based Home Audio/Video interoperability (HAVi) network or the TCP/IP based Universal Plug and Play (UPnP) network. Concluding the idea of networking, that the value of a network is more than the sum of the values the devices provide independently, interconnecting networks adds value (better usability of devices, resource sharing, novel applications, ...). As there is no common model for the interconnection of networks in the home, it is hard to compare interconnections and to find appropriate implementations. Furthermore there are no guidelines and criteria for the development of new interconnections.

Therefore a common model for the interconnection of networks in the home has been defined. An integral part of this model makes a three layered device architecture which specifies the features of home networks. According to this three types of gateways (System-Gateway, Service-Gateway, UI-Gateway) are defined which can be used to interconnect networks at one of these layers by performing layer specific transformations if required. For the sake of completeness a multihomed non transforming device is mentioned as fourth method of interconnection. In addition different types of interconnection (one-way/two-way, direct/indirect, static/dynamic, ...) are specified. Using multiple interconnections in the form of an interconnection system result in further interconnection variants (AB-interconnection, C-interconnection, T-interconnection). Anyway, an appropriate interconnection must be dynamic because only this is all-purpose and does not require additional set-up effort. Finally seventeen network interconnections will be analysed according to the model and six developed interconnections – three implementations and three concepts – will be described.

Keywords: network, interconnection, model, gateway, HAVi, smart home

Vorwort

Die Motivation für diese Arbeit entstand aus der Beschäftigung mit Gerätenetzwerken im Heimbereich am Institut für Computertechnik an der TU Wien beginnend mit der Mitarbeit am EU-Projekt InHoMNet (In Home High Speed Multimedia Network based on IEEE 1394). Wie gezeigt wird gibt es unterschiedliche Konzepte zur Vernetzung von Geräten im Heimbereich und so interessierte mich, auf welchen grundlegenden Mechanismen diese Gerätenetzwerke basieren und wie man diese geeignet koppeln kann, um mit dem dadurch entstehenden Kopplungsgewinn einen höheren Gesamtnutzen zu erreichen.

Die Arbeit ist wie folgt aufgebaut: In der Einleitung wird ein Überblick über das vernetzte Heim hinsichtlich der Geräte, der verschiedenen Anwendungsbereiche, der daraus resultierenden Anforderungen an die Gerätenetzwerke und der neuen Möglichkeiten durch die Kopplung von Gerätenetzwerken gegeben und der Beitrag der Arbeit erläutert.

In Kapitel 2 wird der Stand der Technik der Gerätevernetzung im Heimbereich anhand von verschiedenen Vernetzungsstandards bzw. -spezifikationen beschrieben und neben bereichsübergreifenden Anwendungen eine Reihe von bereits publizierten Konzepten und Implementierungen von Kopplungen analysiert.

In Kapitel 3 wird ein allgemeines Modell zur Kopplung von Gerätenetzwerken im Heimbereich definiert. Wesentlicher Bestandteil des Kopplungsmodells ist eine Gerätearchitektur, welche die für die Vernetzung notwendigen Mechanismen definiert und diese in drei Schichten gruppiert. Anhand dieser werden unterschiedliche Kopplungsmethoden, -arten und -systeme definiert und mögliche Kopplungsverluste behandelt.

In Kapitel 4 werden am Beispiel der Unterhaltungselektronik sechs verschiedene dynamische Kopplungen – drei erstellte Konzepte und drei durchgeführte Implementierungen – beschrieben.

Abschließend werden in Kapitel 5 die Kernaussagen der Arbeit anhand von sechs Thesen ausgeführt und ein Ausblick über weitere Untersuchungen, Implementierungen und Maßnahmen bezüglich der Kopplung von Gerätenetzwerken im Heimbereich gegeben.

Ich hatte Gelegenheit, im Rahmen des Doktoratsstudiums und der Tätigkeit an der TU-Wien viele interessante Menschen zu treffen und möchte mich speziell bei meinen Betreuern, Prof. Dietmar Dietrich und Prof. Wolfgang Kastner, bei meinen Kollegen der

„IEEE 1394“-Gruppe am Institut für Computertechnik, Rene Bauer, Christian Gillesberger, Peter Skiczuk und Manfred Weihs für die gute Zusammenarbeit und bei meinen Studenten für die Mitarbeit bei der Umsetzung von verschiedenen Ideen zur Kopplung von Gerätenetzwerken bedanken.

Größter Dank gilt meiner Familie und insbesondere meiner lieben Frau Birgit, die mich stets auf meinem Weg bestärkt haben - Vielen Dank.

Michael Ziehensack, Wien, März 2004.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Das vernetzte Heim	1
1.2	Beitrag der Arbeit	5
2	Stand der Technik	7
2.1	Gerätevernetzung im Heimbereich	8
2.1.1	Heim-Automatisierung (HA)	8
2.1.2	Unterhaltungselektronik (CE)	15
2.1.3	Small Office/Home Office (PT)	29
2.2	Kopplung von Gerätenetzwerken im Heim	48
2.2.1	Bereichsübergreifende Anwendungen	49
2.2.2	Kopplungen	50
3	Modell zur Kopplung von Gerätenetzwerken	53
3.1	Gerätearchitektur	53
3.1.1	System Layer	54
3.1.2	Service Layer	58
3.1.3	User Interface Layer	60
3.2	Einstufung der Gerätenetzwerke	61
3.2.1	Heim-Automatisierung (HA)	62
3.2.2	Unterhaltungselektronik (CE)	62
3.2.3	Small Office/Home Office (PT)	62
3.3	Kopplungsmethoden	62
3.3.1	Multihomed Non Transforming Device	62
3.3.2	System Gateway	70
3.3.3	Service Gateway	72
3.3.4	User Interface Gateway	75
3.4	Kopplungsarten	77
3.4.1	Transparente Kopplung	77
3.4.2	Direkte und indirekte Kopplung	77
3.4.3	Einseitige und zweiseitige Kopplung	78
3.4.4	Statische und dynamische Kopplung	80
3.5	Kopplungssysteme	82
3.5.1	AB-Kopplung	82

3.5.2	C-Kopplung	83
3.5.3	T-Kopplung	84
3.6	Kopplungsverluste	84
3.7	Ergänzende Anmerkungen	86
4	Dynamische Kopplung am Beispiel der Unterhaltungselektronik	89
4.1	Unterhaltungselektronik – Heim-Automatisierung	90
4.1.1	Umsetzung der Service-Schicht	91
4.1.2	HA-Erweiterung zu HAVi 1.1	94
4.1.3	HAVi-LON Service-Gateway	95
4.1.4	HAVi-EIB Service-Gateway	97
4.1.5	Ergänzende Anmerkungen	99
4.2	Kopplung von Unterhaltungselektronikclustern	99
4.2.1	HAVi-HAVi Kopplung via TCP/IP-Tunnel	99
4.2.2	HAVi-AV/C Kopplung	105
4.3	Unterhaltungselektronik – Small Office/Home Office	109
4.3.1	UPnP-HAVi/DDI UI-Gateway	110
4.3.2	UPnP-HAVi/App UI-Gateway	113
5	Bewertung und Ausblick	115
5.1	Bewertung	115
5.2	Ausblick	121
	Abbildungsverzeichnis	123
	Tabellenverzeichnis	125
	Literaturverzeichnis	126

Kapitel 1

Einleitung

Die Computertechnik ist in immer mehr Geräten des modernen Heims enthalten. Dadurch werden die Geräte intelligenter und bieten immer mehr Funktionen [Mes03, Hec03]. Der digitale Anrufbeantworter kommt ohne Tonband aus, die DVD erlaubt die Wiedergabe in verschiedenen Sprachen, das 100-Hz Fernsehgerät liefert ein augenschonenderes Bild oder die Heizungsanlage unterstützt verschiedene Ziel-Temperaturen in Abhängigkeit der Zeit oder anderer Vorgaben. Durch die Vernetzung dieser intelligenten Geräte entsteht ein vernetztes Heim¹, das dem Bewohner je nach Geräten und Vernetzungsgrad einen höheren Komfort, eine bessere Kommunikation, mehr Sicherheit, Information und Unterhaltung verspricht und über den intelligenten Einsatz der Geräte auch die Energiekosten senken kann [Mil01, BHP02].

1.1 Das vernetzte Heim

Die Gerätevernetzung im Heimbereich erfolgt jedoch aufgrund der unterschiedlichen Anforderungen der Geräte an das Netzwerk zumeist jeweils nur innerhalb eines Anwendungsbereichs und nicht bereichsübergreifend [Hom03, ZP02, EUR99, Ros03]. So gibt es Unterschiede hinsichtlich der benötigten Bandbreite und Ausdehnung des Netzwerks, der möglichen Geräteanzahl und der Teilnehmerdynamik. Weitere Merkmale sind die geforderte Datensicherung und Betriebssicherheit, das Übertragungsmedium, die Art der Gerätenutzung oder der zulässige Konfigurationsaufwand. Je nach Fokussierung können unterschiedliche Anwendungsbereiche identifiziert werden. In Bezug auf die Gerätevernetzung werden hier drei Bereiche mit jeweils gleichen oder zumindest ähnlichen Anforderungen unterschieden²:

(1) Heim-Automatisierung

Zu diesem Bereich gehören vor allem Geräte für Licht, Jalousie, Heizung, Warmwasser, Klima, Lüftung, Alarmanlage, Schließenanlage, Melder (Feuer, Gas, Wasser), Bedienelemente (Schalter, Taster, einfache LCD-Anzeige), Zähler (Gas, Wasser,

¹intelligentes Heim, Smart Home

²Die zu jedem Bereich angeführten Beschreibungen dienen vorwiegend zur Vorstellung des jeweiligen Bereichs sollen jedoch nicht als abschließende Analyse verstanden werden.

Strom) und Geräte der Weißen Ware, wie Mikrowelle, Eiskasten, Tiefkühltruhe, Geschirrspüler, Herd, Waschmaschine, Wäschetrockner, usw.

Das Gerätenetzwerk muss eine große Anzahl an Teilnehmern (> 100) aufnehmen können und eine große Ausdehnung ($> 100\text{ m}$) ermöglichen. Für die Kommunikation ist nur eine geringe Datenrate ($\approx 78\text{ kbit/s}$) notwendig, die Übertragung muss jedoch gesichert erfolgen (bestätigte Übertragung mit Wiederholungen bei Fehlern). Die Anforderungen an die Betriebssicherheit sind sehr hoch. Da diese Geräte zumeist einen konstanten Einsatzort haben, ist der benötigte Konfigurationsaufwand für das Netzwerk nicht sehr entscheidend.

Typische Anwendungsbeispiele der Heim-Automatisierung sind: die Heizungssteuerung mit mehreren Heizzonen und raumweiser Temperaturabsenkung bei Abwesenheit, die Anwesenheitssimulation im Urlaub, die Beleuchtung anhand vordefinierter Lichtszenen, das Lastmanagement zur Energiekosteneinsparung, die Mehrfachverwendung von Sensoren (z. B. Fensterkontakt und Anwesenheitssensor für Heizungssteuerung und Alarmanlage) oder die gemeinsame Ansteuerung mehrerer Geräte durch Mehrfachaktionen (Licht aus, Heizung auf Abwesenheit, Jalousien schließen) [KDP02, Mil01, Sch03, Tel01, Hau03].

(2) Unterhaltungselektronik

Zum Bereich Unterhaltungselektronik gehören vor allem Geräte wie Fernsehapparat, Radio-Tuner, Verstärker, CD/MD-Player, DVD-Player/Recorder, Videorekorder (Kassette, Harddisk), Set-Top-Box, Sat-Receiver, Videokamera, Lautsprecher, Spielekonsole.

Die Anforderungen an ein Netzwerk für digitale Geräte der Unterhaltungselektronik sind primär die digitale Übertragung von Audio/Video-Daten zwischen beliebigen Geräten des Netzwerks und die Ansteuerung der Geräte über das Netzwerk. Dies erfordert eine hohe Bandbreite des Netzwerks und eine geringe Schwankung der Übertragungsdauer. In Tabelle 1.1 sind typische Bitraten von verschiedenen Audio/Video-Quellen dargestellt. Da das Netzwerk vom Benutzer selbst verändert wird, soll das Hinzufügen und Entfernen von Geräten im laufenden Betrieb möglich sein und ohne Administration durch den Benutzer auskommen.

Die Vorteile gegenüber der „alten“ Unterhaltungselektronik liegen dabei in der Vielzahl an neuen Möglichkeiten wie das zeitversetzte oder interaktive Fernsehen, die Bereitstellung von Informationen zu Audio/Video Daten (EPG, Content-Navigator, ...), die gemeinsame Nutzung von Ressourcen über das Netzwerk (SAT- und Kabel-Empfang, Speichermedien, Internetanschluss, ...), die automatische Synchronisation der Geräteuhren und vieles andere mehr [Zie03b, Mes03].

(3) Small Office/Home Office (SOHO)

Zu diesem Bereich gehören Geräte wie: PC, Drucker, Scanner, externe Laufwerke (CD, DVD, HD, ...), Fax, Modem (analog, xDSL, Kabel), ISDN-Adapter, Telefon (fix, schnurlos), Anrufbeantworter, Telefonanlage.

Die primäre Anforderung an das PC-Netzwerk ist die gemeinsame Nutzung von Ressourcen, wie ein gemeinsamer Zugriff auf einen Drucker oder auf Dateien und vor allem ein gemeinsamer Zugang zum Internet [Wür03, Foc03]. Das PC-Netzwerk im Heim ist IP-basiert, besteht aus wenigen Teilnehmern und hat eine geringe Ausdehnung. Aufgrund der vielseitigen Einsetzbarkeit von IP steht eine große Anzahl von Übertragungssystemen zur Verfügung. Es kommen vor allem kabelgebundene Ethernet-LANs (IEEE 802.3 [IEE02b]) oder kabellose WLAN-LANs (IEEE 802.11 [IEE99a]) mit Übertragungsraten je nach Übertragungssystem von 2 bis 100 Mbit/s zum Einsatz (Best-Effort, kein QoS)³. Da das Netzwerk vom Benutzer selbst verändert wird, soll das Hinzufügen und Entfernen von Geräten möglich sein und ohne wesentliche Administration durch den Benutzer auskommen. Die Computerperipherie ist entweder über das Netzwerk oder direkt mit dem PC über eine der „lokalen“ PC-Schnittstellen (serielle oder parallele Schnittstelle, USB, PS/2, IrDA, Bluetooth, ...) verbunden. In Bezug auf die Heim-Vernetzung von Geräten zur Telekommunikation sind vor allem das kabelgebundene ISDN (Integrated Services Digital Network) und das schnurlose DECT (Digital Enhanced Cordless Technologies) zu erwähnen.

Audio/Video Daten	typische Bitrate
Compact Disc (Stereo)	1,41 Mbit/s
MPEG Layer 3 „MP3“ (Stereo)	0,020-0,128 Mbit/s
TV (PAL, unkomprimiert)	250 Mbit/s
DV (Digital Video)	25 Mbit/s
MPEG2 Video (VHS Quality)	2 Mbit/s
MPEG2 Video (PAL Broadcast Quality)	4 Mbit/s
MPEG2 Video (DVD Quality)	10 Mbit/s
MPEG2 Video (DV Quality)	15 Mbit/s
Sat-Transponder (ASTRA 1H, tp71 9xTV, 10xRadio)	38 Mbit/s

Tabelle 1.1: Typische Bitraten von Audio/Video-Daten [Zie03b]

Einerseits bestehen zwischen einzelnen Anforderungen der unterschiedlichen Bereiche Zielkonflikte, wie beispielsweise zwischen einer hohen Bandbreite und einer großen Ausdehnung oder zwischen einer hohen Betriebssicherheit und einer großen Teilnehmerdynamik. Andererseits werden zur Erreichung möglichst niedriger Kosten hauptsächlich nur die primär notwendigen Anforderungen einer Gerätevernetzung realisiert. So wird für die Vernetzung von Geräten der Unterhaltungselektronik eine zeitlich garantierte Übertragung von großen Datenmengen gefordert, während bei der Heim-Automatisierung eine gesicherte Übertragung von kleinen Datenmengen über große Distanzen zwischen vielen

³Die maximale Übertragungsraten von WLANs ist derzeit 54 Mbit/s (IEEE 802.11a [IEE99b], IEEE 802.11g [IEE03]); Gigabit-Ethernet ermöglicht eine Übertragungsraten von 1000 Mbit/s (IEEE 802.3z [IEE02b]).

im ganzen Heim verteilten Geräten notwendig ist. Anstatt die Heterogenität der Gerätevernetzung im Heimbereich durch ein neues „Super-Netzwerk“, das viele Kompromisse aufgrund des großen Einsatzspektrums eingehen muss, zu überwinden, sollen die spezialisierten und branchenbezogenen Netze gekoppelt werden.

Durch die Kopplung der branchenspezifischen Netzwerke könnten die Grenzen zwischen den verschiedenen Netzwerken überwunden und Dienste bzw. Funktionen, die in einem Netz verfügbar sind, auch Geräten in einem anderen Netzwerk zugänglich gemacht werden. So könnte durch eine Kopplung zwischen der Unterhaltungselektronik und der Heim-Automatisierung ein TV-Gerät als komfortable Benutzerschnittstelle für Geräte der Heim-Automatisierung dienen, z. B. zur Einstellung der Heizungsanlage oder der Bedienung der Jalousien. Durch eine Kopplung von Netzwerken der Unterhaltungselektronik könnte wiederum deren beschränkte Ausdehnung überwunden werden und z. B. die Programmierung eines VCR oder einer AV Hard Disk auch von einem entfernten Netzwerk aus erfolgen. Des Weiteren werden durch die Kopplung neue, bereichsübergreifende Anwendungen möglich. Beispiele dafür sind programmierbare Szenarien zur vordefinierten Einstellung verschiedener Geräte, wie „TV-Prime-Time“ (Licht im Wohnzimmer gedimmt auf 20%, TV-Audio über HiFi-Anlage mit Lautstärke auf Stufe 5, TV-Videoeinstellungen Standard, ...) oder „Wake-Up“ (bei Erreichen der Weckzeit: Radio ein, Jalousie öffnen, persönliche Verkehrs- und Wetterinfos über TV bereitstellen, ...), und erweiterte Einstellmöglichkeiten durch die Verknüpfung von Informationen aus mehr als einem Netzwerk, wie die Programmierung der Heizung entsprechend der über das TV-Gerät abrufbaren Wetterdaten. Die Kopplung der Gerätenetze ist die konsequente Fortführung des Vernetzungsgedankens, demzufolge der Nutzen eines Gerätenetzes größer ist als die Summe des Nutzens der einzelnen Gerätefunktionen.

Die Zielsetzung dieser Arbeit ist die Untersuchung der verschiedenen Möglichkeiten zur dynamischen Kopplung von Gerätenetzen im Heimbereich.

Zur Verbindung von Gerätenetzen im Heimbereich sind bereits eine Reihe von Konzepten und Implementierungen von Kopplungen publiziert worden (s. Abschnitt 2.2). Die meisten der Kopplungen unterscheiden sich jedoch nicht nur hinsichtlich der zu verbindenden Netzwerke, sondern auch hinsichtlich der Zielsetzung der Verbindung (Überwindung von Beschränkungen eines Gerätenetzes, Ansteuerung von Geräten über eine andere API als die ursprüngliche API, Benutzung der Geräte über eine Benutzerschnittstelle eines anderen Netzwerks, ...). Da es jedoch kein allgemeines Modell zur Kopplung von Gerätenetzen im Heimbereich gibt, ist der Vergleich von Kopplungen bzw. die Suche nach geeigneten Implementierungen sehr aufwendig und es fehlen Richtlinien oder Kriterien für die Entwicklung von neuen Kopplungen.

Abhilfe soll die Definition eines allgemeinen Modells zur Kopplung von Gerätenetzen im Heimbereich schaffen, das sich einerseits zur Analyse und Beschreibung von bestehenden und andererseits als Grundlage zum Design von neuen Kopplungen eignet.

1.2 Beitrag der Arbeit

Diese Arbeit behandelt die dynamische Kopplung von Gerätenetzwerken im Heimbereich. Basierend auf der Untersuchung und dem Vergleich von sechs verschiedenen Gerätenetzwerken des Heimbereichs und von mehreren veröffentlichten Kopplungen von Gerätenetzwerken wird ein allgemeines Modell zur Kopplung von Gerätenetzwerken im Heimbereich definiert. Als wesentlicher Bestandteil des Kopplungsmodells wird eine Gerätearchitektur spezifiziert, welche die zur Vernetzung notwendigen Dienste definiert und diese in drei Schichten gruppiert. Eine Kopplung von Gerätenetzwerken wird dann durch eine umsetzende Verbindung auf einer der Schichten der Gerätearchitektur über ein entsprechendes Gateway oder durch eine einfache Anbindung über ein nichtumsetzendes, mehrfach angebundenes Gerät abstrahiert. Das Modell unterscheidet verschiedene Arten von Kopplungen (einseitig/zweiseitig, direkt/indirekt, statisch/dynamisch, ...) und berücksichtigt die speziellen Aspekte des gemeinsamen Einsatzes von mehreren Kopplungen durch die Beschreibung von Kopplungssystemen. Außerdem werden mögliche Kopplungsverluste identifiziert und ein Vergleich des entwickelten Modells mit anderen Modellen der Gerätekommunikation durchgeführt.

Die Beschreibung einer Kopplung anhand des Modells erfordert Antworten über die wesentlichen Punkte der Kopplung. Dadurch wird ein einfacher Vergleich mit anderen Kopplungen oder das Auffinden geeigneter Realisierungen möglich. Das Ergebnis von insgesamt siebzehn anhand des Modells analysierten Kopplungen ist in Tabelle 5.2 zusammengefasst. Die letzte Spalte der Tabelle bezeichnet jeweils den Abschnitt in dieser Arbeit, welcher die Analyse der Kopplung beschreibt. Die untersuchten Kopplungen stellen einen guten Querschnitt dar. So sind alle sechs in Abschnitt 2 beschriebenen Gerätenetzwerke aus den drei Bereichen des Heims, alle vier Kopplungsmethoden, alle drei Kopplungssysteme, einseitige, zweiseitige, direkte, indirekte, transparente sowie dynamische Kopplungen vertreten.

Neben der Analyse von Kopplungen liefert das Modell Richtlinien und Kriterien für die Entwicklung von neuen Kopplungen. Voraussetzung für ein Design von Kopplungen ist eine Analyse der zu verbindenden Gerätenetzwerke. Dies kann anhand der dreischichtigen Gerätearchitektur erfolgen. Da das Kopplungsmodell unterschiedliche Methoden und Arten von Kopplungen beschreibt, kann es als eine Art „Werkzeugkasten“ für neue Kopplungen dienen. Ausgehend von der Zielsetzung der zu erstellenden Kopplung kann so eine geeignete Kopplungsmethode und -art gewählt werden. So wurden am Beispiel der Unterhaltungselektronik insgesamt sechs verschiedene dynamische Kopplungen – drei Konzepte und drei Implementierungen – anhand des Modells entwickelt. Sie sind in Kapitel 4 beschrieben und repräsentieren einen Querschnitt von Verbindungen zwischen den Gerätenetzwerken, weil Kopplungen zu den drei Bereichen des Heims (Heim-Automatisierung, Unterhaltungselektronik, Small Office/Home Office) durchgeführt wurden und alle drei Gateway-Typen des Modells zur Anwendung kamen.

Wie in den Abschnitten 3.4.4 und 5.1 gezeigt wird, muss eine geeignete Kopplung

jedenfalls „dynamisch“ sein, weil nur diese allgemein einsetzbar ist und ohne wesentlichen Konfigurationsaufwand auskommt. Nur durch die allgemeine Einsetzbarkeit der Kopplung können die unterschiedlichen im Heimbereich eingesetzten Anwendungen und Geräte unterstützt werden. Ein geringer Konfigurationsaufwand ist wiederum entscheidend für den Nutzen einer Kopplung. So sind alle sechs im Rahmen dieser Arbeit entwickelten Kopplungen dynamisch.

Kapitel 2

Stand der Technik

Die Gerätevernetzung im Heimbereich erfolgt zumeist jeweils nur innerhalb eines Anwendungsbereichs und nicht bereichsübergreifend. Dies liegt, wie bereits im vorhergehenden Kapitel ausgeführt wurde, vor allem an den unterschiedlichen Anforderungen der Geräte an das Netzwerk. So gibt es Unterschiede hinsichtlich der benötigten Bandbreite und Ausdehnung des Netzwerks, der möglichen Geräteanzahl und vielen anderen mehr. Der Heimbereich stellt damit in Bezug auf die Gerätevernetzung ein heterogenes Umfeld dar [Hom03, ZP02, Pfe99, O'D01, Ros03, KDP02].

Zur Strukturierung der Gerätevernetzung im Heimbereich existieren verschiedene Ansätze. In der im Rahmen des Projekts HINE (Heterogeneous Inhouse Networking Environment) [EUR99] durchgeführten Analyse zur Ermittlung von „Inhouse Technologies and Products“ wurde eine Unterteilung in vier Anwendungsbereiche vorgenommen. Im einzelnen sind das die Bereiche „Home Automation“, „Small Office, Home Office (SOHO)“, „Entertainment“ und „Telephony“. Im Gegensatz dazu werden in [BHP02] fünf so genannte Anwendungsfelder integrierter Gebäudesysteme unterschieden: „Gebäudesicherheit“, „Komfort und Wohlfühlen - Energie und Klima“, „Entertainment“, „Health Care - Social Care“ und „Smart Office“. Während die erste Unterteilung Geräte und Netzwerke mit gleichen oder verwandten Anwendungen und Anforderungen in Anwendungsbereiche gruppiert, werden bei der zweiten Art der Strukturierung mehrere Anwendungsfelder aus mitunter auch sehr unterschiedlichen einzelnen Anwendungen¹ gebildet. Beispielsweise ist das Anwendungsfeld „Health Care - Social Care“ besonders auf die Gruppe der Senioren zugeschnitten und verwendet Sensoren zur Aktivitätsüberwachung, eine Internetverbindung zur Telediagnose bzw. für Krankenfernbetreuungsdienste oder beinhaltet die Verwendung eines WebPads als universale Fernbedienung für TV, Licht und Heizung. Die Perspektive des Anwendungsfelds ist demnach primär nutzerorientiert und nicht geräte- bzw. netzwerkorientiert. Damit eignet sich die Unterteilung in Anwendungsfelder gut zur Entwicklung neuer (bereichsübergreifender) Anwendungen, jedoch nicht zur Analyse der verschiedenen Gerätenetze der einzelnen Anwendungsbereiche.

Angelehnt an die erste Strukturierungsvariante werden in Abschnitt 2.1 typische Vertreter von Gerätenetzen aus folgenden drei Anwendungsbereichen mit jeweils gleichen oder zumindest ähnlichen Anforderungen beschrieben:

¹Die einzelnen Anwendungen können aus unterschiedlichen Anwendungsbereichen stammen.

- (1) Anwendungsbereich HA: Heim-Automatisierung („Home Automation“)
- (2) Anwendungsbereich CE: Unterhaltungselektronik („Consumer Electronics“)
- (3) Anwendungsbereich PT: Small Office/Home Office („PC and Telecommunication“)

Nach der Darstellung der Gerätevernetzung im Heimbereich werden in Abschnitt 2.2 bereichsübergreifende Anwendungen und eine Reihe von publizierten Konzepten und Implementierungen von Kopplungen besprochen.

2.1 Gerätevernetzung im Heimbereich

Entsprechend der Unterteilung in die drei Anwendungsbereiche Heim-Automatisierung (Abschnitt 2.1.1), Unterhaltungselektronik (Abschnitt 2.1.2) und Small Office/Home Office (Abschnitt 2.1.3) werden für jeden Bereich jeweils zwei typische Standards bzw. Spezifikationen zur Vernetzung von Geräten beschrieben. Ziel der Ausführungen ist die Funktionsweise und wesentlichsten Eigenschaften der Gerätenetzwerke darzustellen. Sie bilden damit die Grundlage für die Analyse der wesentlichen Mechanismen zur Vernetzung von Geräten im Heimbereich (siehe Kapitel 3, Gerätearchitektur).

2.1.1 Heim-Automatisierung (HA)

Das *Local Operating Network (LON)* [DLS97] und der *Europäische Installations Bus (EIB)* [DKS00] sind zwei verbreitete Feldbussysteme und damit zwei typische Vertreter für Standards zur Vernetzung von Geräten der Heim-Automatisierung.

2.1.1.1 Local Operating Network (LON)

Ende der 80er Jahre wurde von der amerikanischen Firma Echelon² das so genannte *Local Operating Network* zur Vernetzung von Geräten der Gebäude- und Prozessautomatisierung entwickelt, das bis heute unter dem eingetragenen Markenzeichen *LonWorks* vermarktet und stetig weiterentwickelt wird. Die LON-Technologie [DLS97] besteht im Wesentlichen aus den folgenden Elementen:

- *LonTalk*-Protokoll
Das LonTalk-Protokoll behandelt alle sieben Schichten des OSI-Modells und ist das Kommunikationsprotokoll des LON. Seit 1998 ist es als ANSI/EIA Standard 709.1 („Control Network Specification“) [EIA98, EIA02] veröffentlicht.
- *Neuron-Chip*
Der Neuron-Chip ist ein Mikrocontroller³, auf welchem das LonTalk-Protokoll implementiert ist und ausgeführt wird. Des Weiteren ermöglicht er die Ausführung

²Homepage: www.echelon.com

³Der Neuron-Chip besteht aus drei miteinander verbundenen Prozessoren, Speicher und verschiedenen Ein- und Ausgabeschchnittstellen. Zwei Prozessoren sind für das LonTalk-Protokoll und eine für die Abarbeitung der Anwendung zuständig.

von ereignisgesteuerten Anwendungen. Nahezu jeder LON-Knoten enthält einen Neuron-Chip und damit eine einheitliche Implementierung des LonTalk-Protokolls.

- **Transceiver**
Der Transceiver ist die Anbindung des Neuron-Chips an das Übertragungsmedium [Lon02b]. Es werden unterschiedliche Übertragungsmedien wie Twisted-Pair, Koaxialkabel, Powerline, Funk und Lichtwellenleiter unterstützt. Die standardisierten Übertragungsraten hängen vom verwendeten Übertragungsmedium ab und reichen von 300 bit/s bis zu 1,25 Mbit/s, weit verbreitet sind 78 kbit/s.
- **Tools**
Zur Entwicklung, Installation und Wartung von LON-Geräten und Netzen stehen verschiedene Software- und Hardwarewerkzeuge zur Verfügung: LonBuilder, NodeBuilder, LNS⁴, LonMaker, ...

Je nach Übertragungsmedium werden unterschiedliche Topologien wie Linie, Ring, Stern oder Baum unterstützt. Durch den Einsatz von LON-Repeater, -Bridges oder -Router können (auch unterschiedliche) physikalische Übertragungswege miteinander verbunden und Netze mit beliebiger Topologie realisiert werden. Als Buszugriffsverfahren wird predictive p-persistent CSMA eingesetzt. Dabei kann der Buszugriff erfolgen, wenn (a) der Knoten keine Aktivität am Bus feststellt und (b) auch nach Ablauf einer zufällig gewählten Wartezeit keine neue Aktivität am Bus erkannt wird (p-persistent). Zur weiteren Vermeidung von Kollisionen wird die zufällig gewählte Wartezeit in Abhängigkeit der zu erwartenden Buslast - eine bestätigte Übertragung verursacht durch das Senden der Bestätigungen eine höhere Buslast als eine unbestätigte Übertragung - erhöht (predictive).

Ein LON-Knoten besteht zumeist aus einem Neuron-Chip, einer entsprechenden Ein- und Ausgabebeschaltung zu Sensoren oder Aktoren, einem Transceiver-Modul zur Anbindung an das Übertragungsmedium, einer ereignisgesteuerten Anwendung zur Anbindung der Sensoren oder Aktoren und einem Versorgungsteil (Stromversorgung, Quarz, ...).

Die Adressierung der LON-Knoten kann auf unterschiedliche Arten erfolgen. Jeder LON-Knoten besitzt eine weltweit eindeutige physikalische 48-Bit-Adresse, die bei der Herstellung fest zugewiesen und als *Neuron-ID* bezeichnet wird. Die Neuron-ID wird generell nur zur Installation und für Diagnosezwecke verwendet. Für den normalen Betrieb werden logische Adressen benutzt, die während der Installation vergeben werden. Die logischen Adressen basieren auf einer hierarchischen Adressstruktur, die aus folgenden Teilen besteht:

- *Domain*
In einer Domäne („Domain“) werden Knoten nach ihrem Aufgabengebiet (Beleuchtung, Sicherheit, Energie, ...) oder ihrer physikalischen Lage (Gebäude, Stockwerk,

⁴LNS ist die Abkürzung für „LonWorks Networks Services“.

Zimmer, ...) zusammengefasst. Knoten können ohne Gateway nur innerhalb einer Domäne kommunizieren. Ein Knoten kann in bis zu zwei⁵ Domänen Mitglied sein.

- *Subnet, Node*
Innerhalb einer Domäne kann es bis zu 255 Teilnetze („Subnets“) geben, welchen jeweils bis zu maximal 127 Knoten („Nodes“) angehören können. Zwischen den Knoten eines Teilnetzes wird nicht geroutet.
- *Group, Member*
Gruppen („Groups“) dienen zur Zusammenfassung von maximal 64 Knoten einer Domäne zur gemeinsamen Adressierung („Multicast“). Innerhalb einer Domäne können maximal 256 Gruppen gebildet werden, wobei ein einzelner Knoten in maximal 15⁶ Gruppen Mitglied („Member“) sein kann.

Jedes LonTalk-Paket, das über das Netzwerk übertragen wird, enthält die logische Knotenadresse (Subnet-Node) des übertragenden Knotens (Quelladresse), die Zieladresse des oder der Empfänger und die Domain-ID. Abhängig von der Adressierungsart besteht die Zieladresse aus der physikalischen Neuron-ID, der logischen Knotenadresse (Subnet-Node), einer Gruppenadresse oder einer Broadcastadresse.

Die Anwendungen der Neuron-Chip basierten LON-Knoten werden in einer an ANSI-C angelehnten Sprache (*Neuron C* [Ech03a, Ech03b]) erstellt, die speziell auf den Neuron-Chip zugeschnitten ist. Der Scheduler des Neuron-Chips ermöglicht ereignisgesteuerte Anwendungen, d. h. ereignisabhängig werden bestimmte Programmteile abgearbeitet. Die einzelnen Programmteile werden durch Anweisungen der Form:

```
when (event)
  task
```

definiert. Das Ereignis („event“) kann entweder ein vordefiniertes Ereignis wie z. B. *msg_arrives* oder *timer_expires*, oder ein Neuron-C Ausdruck wie z. B. $x < y$ sein. Die einzelnen „when“-Klauseln werden nacheinander⁷ ausgewertet und bei Eintritt des Ereignisses wird der zugehörige Programmteil („task“) ausgeführt.

Die Kommunikation der Anwendungen basiert fast ausschließlich auf so genannten *Netzwerkvariablen (NV)*, die logische Verbindungen zwischen den Anwendungen der LON-Knoten darstellen. Es gibt Eingangs- und Ausgangsnetzwerkvariablen und sie werden im Quelltext der Anwendung mit der Speicherklasse *network* definiert. Im Rahmen der Installation erfolgt, zumeist durch ein Netzwerk-Management-Werkzeug wie z. B. dem LonMaker, das Verbinden (*Binding*) der Eingangs- und Ausgangsnetzwerkvariablen. Dazu wird für jede Verbindung eine geeignete Adressierungsart und ein eindeutiger 14-Bit großer NV-Selektor, der zur Identifizierung der zusammengehörenden NV dient,

⁵Nach ANSI/EIA-709.1-B-2002 kann ein Knoten in bis zu 65.535 Domänen Mitglied sein.

⁶Nach ANSI/EIA-709.1-B-2002 kann ein Knoten in bis zu 65.535 Gruppen Mitglied sein.

⁷Dies gilt nur für nicht priorisierte Programmteile, die vom Scheduler jeweils erst berücksichtigt werden, wenn keine priorisierten Programmteile mehr ausgeführt werden sollen.

gewählt und damit die entsprechenden internen Tabellen der zu den NV gehörenden Knoten konfiguriert. Weist die Anwendung einer Ausgangsnetzwerkvariablen einen Wert zu, wird dieser automatisch vom Neuron-Chip als Nachricht an alle betroffenen Knoten gesendet, dort empfangen und den verbundenen Eingangsnetzwerkvariablen zugewiesen. Aus der Sicht der Anwendung können die NV wie normale Variablen verwendet werden. Die NV-Kommunikation wird als *Implicit Message* bezeichnet, weil das Senden und Empfangen automatisch erfolgt. Im Gegensatz dazu wird bei der als *Explicit Message* bezeichneten zweiten Art der Kommunikation von der Anwendung selbst die Nachricht gebildet und für das Senden der Nachricht und Empfangen der Antwort gesorgt.

Die Basis für interoperable Anwendungen bilden neben dem LonTalk-Protokoll die LonMark-Richtlinien. Die LonMark Interoperability Association⁸ wurde 1994 gegründet und hat drei Hauptaufgaben: (1) Definition von LonMark-Richtlinien für interoperable Geräte, (2) Zertifizierung von Geräten und (3) Förderung der Vermarktung zertifizierter Geräte. Damit mehrere Geräte unabhängig vom Hersteller zusammenarbeiten können, ist eine einheitliche Interpretation der Daten einer Nachricht und eine standardisierte Geräteschnittstelle notwendig.

Einerseits werden dazu standardisierte Typen für Netzwerkvariablen, die *Standard Network Variable Types (SNVTs)* [Ech02b] und standardisierte Typen für Konfigurationsparameter („Configuration Properties“, CP)⁹, die *Standard Configuration Parameter Types (SCPTs)* [Ech02a], definiert. Zum Beispiel definiert der Typ `SNVT_length_kilo` eine zwei Byte große vorzeichenlose Ganzzahl, die eine Länge in Kilometern bezeichnet.

Andererseits werden in so genannten *Functional Profiles* verschiedene Gerätefunktionen („Thermostat“, „Switch“, „Lamp Actuator“, „Analog Input“, „Pump Controller“, ...) durch die Festlegung von vorgeschriebenen und optionalen NVs und CPs, deren Typen (SNVTs, SCPTs) und deren Funktion standardisiert¹⁰. Die Implementierung eines Functional Profiles muss zumindest die vorgeschriebenen NVs und CPs enthalten und wird *Functional Block* genannt, wobei ein Gerät aus einem oder mehreren Functional Blocks besteht [Lon02a]. Im Functional Profile „Thermostat“ [Lon96b] wird zum Beispiel festgelegt, dass die fünfte Netzwerkvariable eine Ausgangsnetzwerkvariable vom Typ `SNTV_hvac_status` ist und den Status des Gerätes jedes Mal ausgibt, wenn er sich ändert, oder in Intervallen, soweit dies über den Konfigurationsparameter `nciSndHrtBt` eingestellt wurde.

Zur Selbstbeschreibung der Geräte können Informationen über die enthaltenen Function Blocks und der zugehörigen Netzwerkvariablen und Konfigurationsparameter in speziell aufgebauten Strings (*Self-Documentation (SD) String*, *Self-Identification (SI) String*) im Gerätespeicher abgelegt werden. Die Strings werden automatisch vom Neuron-C

⁸Homepage: www.lonmark.org

⁹Die Konfigurationsparameter dienen zur Gerätekonfiguration und können als Netzwerkvariablen oder durch im Gerätespeicher abgelegte Konfigurationsdateien, auf welche über ein Netzwerk-Management-Tool zugegriffen werden kann, implementiert werden.

¹⁰Von der LonMark Interoperability Association standardisierte Functional Profiles werden als *LonMark Profiles* bezeichnet, alle anderen als *User Functional Profiles*.

Compiler ab Version 2 aus dem Quellcode der Anwendung generiert. Bei älteren Compilern und für Ergänzungen müssen die Strings manuell über eine Compiler-Direktive¹¹ beziehungsweise über einen Modifier der Netzwerkvariablen-Deklaration¹² gesetzt werden. Ein Netzwerk-Management-Werkzeug kann diese Strings aus dem Gerät auslesen und für die Installation des Gerätes verwenden. Die Beschreibung des Gerätes kann jedoch auch in einer speziellen Datei, dem *Device Interface File (XIF)*, gespeichert werden. Die XIF-Datei ist in [Lon03] definiert und kann automatisch von einem Netzwerk-Management-Werkzeug erstellt werden.

2.1.1.2 Europäischer Installations Bus (EIB)

Das Hauptanwendungsgebiet des Feldbus *Europäischer Installations Bus (EIB)* ist die Gebäudeautomation und -leittechnik [DKS00]. Zur Standardisierung, Zertifizierung und Vermarktung des EIB wurde 1990 von führenden europäischen Herstellern von Installationsgeräten und -systemen die EIB Association (EIBA)¹³ gegründet. Die EIBA hält das EIB-Logo als eingetragenes Markenzeichen und ist Herausgeber der EIB-Spezifikation (EIB Handbook Series, Version 3) [EIB99].

Beginnend mit 1996 wurde der Konvergenz-Prozess für einen einheitlichen Standard der Heim- und Gebäudeautomation durch die drei europäischen Feldbus-Vereinigungen BatiBUS Club International (BCI)¹⁴, European Home System Association (EHSA)¹⁵ und der EIBA gestartet und 2000 die Vereinigung *Konnex*¹⁶ gegründet. Der gemeinsame Standard *KNX* basiert auf dem EIB mit Erweiterungen von BatiBUS und EHS hinsichtlich neuer Übertragungsmedien und einfacherer Inbetriebnahmemöglichkeiten. Mit Juni 2003 wurde ein erster Teil von KNX (KNX-Protokoll, KNX-Übertragungsmedium Twisted-Pair) Teil der europäischen Norm EN 50090¹⁷.

Angelehnt an die Struktur eines Gebäudes sind EIB-Netzwerke hierarchisch aufgebaut. Das Grundelement des Netzwerks bildet die Linie, über welche bis zu 256 Geräte verbunden werden können. Bis zu 15 Linien können über Linienkoppler zusammengeslossen werden und bilden einen Bereich. Über Bereichskoppler können wiederum bis zu 15 Bereiche verbunden werden.

Jedes EIB-Gerät besitzt eine eindeutige strukturierte (ortsabhängige) physikalische Adresse, die sich aus dem Bereich, der Linie und einer laufenden Nummer innerhalb der Linie zusammen setzt und bei der Geräteinstallation zugewiesen wird. Über diese

¹¹`#pragma set_node_sd_string "sdString"`

¹²`sd_string`

¹³Homepage: www.eiba.org

¹⁴Homepage: www.batibus.com

¹⁵Homepage: www.ehsa.com

¹⁶Homepage: www.konnex.org

¹⁷EN 50090-3-2, EN 50090-4-1, EN 50090-4-2, EN 50090-7-1, EN 50090-5-2

Adresse kann ein Gerät eindeutig angesprochen werden. Zur gemeinsamen Adressierung mehrerer Teilnehmer werden logische Adressen (Gruppenadressen) verwendet. Die Gruppen stellen die Verbindung zwischen den einzelnen Gerätefunktionen (Sensoren und Aktoren) her. Sollen beispielsweise mehrere Aktoren gemeinsam angesprochen werden, so werden sie in einer Gruppe zusammengefasst. Zur besseren Übersicht bestehen die ortsunabhängigen Gruppenadressen aus den Elementen Hauptgruppe und Untergruppe. So können Funktionsbereiche wie zum Beispiel die Beleuchtung, Klimatechnik oder Sicherheitstechnik anhand der Hauptgruppe und die zugehörigen Gerätefunktionen anhand der Untergruppe unterschieden werden. Jedes Datenpaket enthält sowohl die Quell- als auch die Zieladresse. Während die Quelladresse immer eine physikalische Adresse ist, kann die Zieladresse auch eine Gruppenadresse sein, was sie im Normalfall auch ist.

Ein EIB-Knoten besteht aus einem Kommunikationsmodul, das die Abwicklung des EIB-Protokolls übernimmt und auf das verschiedene Anwendungsmodul, wie zum Beispiel Schalter, Taster, Anzeigen oder Schnittstellen zu PCs, aufgesteckt werden können:

- *Kommunikationsmodul*

Das Kommunikationsmodul (Bus Coupling Unit, BCU) umfasst einen Sende- und Empfangsteil (Transceiver), eine Verarbeitungseinheit (Mikrocontroller) und eine genormte Anwendungsschnittstelle (AST)¹⁸ zur Anbindung des Anwendungsmoduls. Je nach Implementierung kann das Kommunikationsmodul auch ein Anwendungsprogramm enthalten, das bei der Installation in den nichtflüchtigen Speicher geladen wird und über die AST auf das Anwendungsmodul zugreifen kann.

- *Anwendungsmodul*

Das Anwendungsmodul kann entweder über das Anwendungsprogramm im Kommunikationsmodul (internes Programm) angebunden werden oder auch ein eigenes Anwendungsprogramm enthalten (externes Programm). Für einfache Anwendungen, wie zum Beispiel einem Schalter, wird meist ein internes Programm verwendet, das die AST als einfachen Eingabeport verwendet und den Zustand über eine Nachricht den zugehörigen Aktoren (Licht, Jalousie, Heizung, ...) mitteilt. Die Konfiguration der AST wird durch einen Kommunikationstyp festgelegt, der über den Spannungsabfall an einem im Anwendungsmodul enthaltenen Typwiderstand bestimmt wird. Beispielsweise definiert Typ 16 eine serielle, asynchrone Kommunikation über die AST zur Anbindung eines PCs oder Mikrocontrollers. Die Software-Schnittstelle zwischen externer Anwendung und lokaler BCU bildet das *External Message Interface (EMI)*. Damit ist ein Zugriff der externen Anwendung auf verschiedene EIB-Kommunikationsschichten der BCU möglich.

Als Übertragungsmedium werden vor allem Twisted-Pair und Powerline unterstützt. Die ursprüngliche Variante des EIB ist Twisted-Pair (TP), bei der die Busleitungen beliebig verzweigt werden dürfen (Bus, Baum, Stern) und keine Abschlusswiderstände benötigt werden, die eine Übertragungsrate von 9.600 bit/s bietet und bei der die Geräte

¹⁸engl. Physical External Interface (PEI)

über die Busleitung ferngespeist werden. Als Buszugriffsverfahren wird bei TP das Verfahren Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) eingesetzt. Bei Powerline erfolgt die Datenübertragung eigenleitungslos über das 230 V-Stromnetz mit 1.200 bit/s, wobei der Buszugriff über eine Art p-persistent CSMA erfolgt und die Datenrahmen der TP-Variante erweitert wurden. So dient beispielsweise eine zusätzliche System-ID zur Identifizierung von logisch zusammengehörenden EIB-Netzwerken, die aufgrund der durch das installierte Stromnetz vorgegebenen Topologie nicht elektrisch voneinander getrennt werden können.

Verglichen mit dem OSI-Modell implementiert EIB Dienste und Protokolle entsprechend den ersten vier Schichten und der Schicht 7 des OSI-Modells. Während die unteren vier Schichten die Übertragung von Daten über Broadcast- und Multicast-, sowie über verbindungsorientierte und verbindungslose Unicast-Dienste ermöglichen, bietet die Schicht 7 im Wesentlichen Netzwerkmanagementdienste (Adresszuweisung, Zugriff auf den lokalen Gerätespeicher, ...) und Dienste zum Austausch von Daten über Objekte. Die Kommunikation bei EIB ist daten- und nicht befehlsorientiert. Anwendungen tauschen Daten über Kommunikationsobjekte oder Anwendungsobjekte aus. Für das Netzwerkmanagement werden Systemobjekte (Geräte-Objekt, Adresstabellen-Objekt, ...) verwendet.

- *Gruppen-Kommunikationsobjekte*
Kommunikationsobjekte bestehen aus drei Teilen: einem Objektwert, einer Objektbeschreibung und einem für die Kommunikation relevanten Satz von Flags. Wird dem Kommunikationsobjekt eine Gruppenadresse zugeordnet, kann der Objektwert über das Netzwerk gelesen oder geschrieben werden (Multicast-Übertragung). Für die Zuordnung gilt jedoch die Einschränkung, dass ein Kommunikationsobjekt Daten über verschiedene Gruppenadressen empfangen kann, jedoch nur über eine einzige Gruppenadresse seine Daten senden darf.
- *Anwendungsobjekte, Systemobjekte (EIB-Objekte)*
Anwendungsobjekte und Systemobjekte besitzen einen eindeutigen Kennzeichner (Objekt-Id) und bestehen aus einem oder mehreren *Property*s. Ein *Property* setzt sich aus einer Beschreibung und einem Wert (Datenarray) zusammen und besitzt einen innerhalb des Objekts eindeutigen Kennzeichner (Property-Id). Der Zugriff auf ein *Property* erfolgt über die physikalische Adresse des Gerätes, die Objekt-Id und die Property-Id (Unicast-Übertragung).

Zwischen Anwendung und der Schicht 7 wurde noch eine weitere Schicht, der *User-Layer* eingefügt. Er entlastet die Anwendung von den allgemeinen EIB-Kommunikationsaufgaben, indem er den Zugriff auf Kommunikationsobjekte, Anwendungsobjekte und Systemobjekte selbständig behandelt. Der User-Layer agiert als eine Art Objekt-Server, er verwendet die Dienste der Anwendungsschicht und stellt „höherwertige“ Dienste der Anwendung zur Verfügung. Aufgabe der internen oder externen Anwendung bleibt jedoch die Übertragung von Objektwerten zu veranlassen und diese Werte entsprechend

zu verarbeiten.

Für die standardisierte Interpretation der ausgetauschten Daten wurden die *EIB Interworking Standards* (EIS) definiert [EIB99]. So gibt es 15 EIS-Typen („switching“, „dimming“, „time“, ...) für Kommunikationsobjekte. Für jeden Typ ist die Größe des Wertes und dessen Aufbau (z. B. 3 Byte groß: die höherwertigsten drei Bits des ersten Bytes enthalten den Tag, die verbleibenden 5 Bits des ersten Bytes die Stunde, ...) sowie dessen Funktion beziehungsweise Bedeutung („on/off“, „move up/down“, Temperatur in Grad Celsius, Länge in Meter, Zählerstand, ...) festgelegt. Zur Standardisierung der EIB-Objekte (IDs, Datentypen, Aufbau) sind die *Object Interworking Standards* (ObIS) definiert worden. Darauf aufbauend wurden im Rahmen des neuen Standards KNX verschiedene Anwendungsmodelle beschrieben und dazu in Form von so genannten *Functional Blocks* festgelegt, welche Funktionen ein konformes Gerät besitzen muss und über welche Kommunikationsobjekte oder Anwendungsobjekte diese genutzt werden können [KON02].

Einziges Planungs- und Inbetriebnahmewerkzeug ist die von der EIBA entwickelte *EIB Tool Software* (ETS). Die Anwendungsprogramme der EIB-Geräte sind zumeist in Assembler geschrieben und werden vom Gerätehersteller gemeinsam mit der Gerätebeschreibung und den möglichen Konfigurationsparametern über die Produktdatenbank der ETS bereitgestellt. Zur Inbetriebnahme wird das entsprechende Gerät aus der Produktdatenbank ausgewählt (z. B. Taster 2-fach), eine physikalische Adresse entsprechend des Installationsortes des Gerätes festgelegt (z. B. 2.1.2), eines der verfügbaren Anwendungsprogramme (z. B. Taster 2-fach: Ein/Aus) sowie die zugehörigen Konfigurationsparameter (z. B. Wippenkontakt oben: drücken EIN, Wippenkontakt unten: drücken AUS, ...) gewählt und die Verbindung der Kommunikationsobjekte durch Zuweisung von Gruppenadressen (z. B. 1/0/2) erstellt. Da nur von der EIBA zertifizierte Geräte in die Produktdatenbank aufgenommen werden, ist gewährleistet, dass nur standardkonforme EIB-Geräte verwendet werden. Die aktuelle Version der ETS ist „ETS2 Version 1.3“.

2.1.2 Unterhaltungselektronik (CE)

Bisher können Geräte der Unterhaltungselektronik von verschiedenen Herstellern nur in einem sehr beschränkten Maß Steuerinformationen austauschen. Eine Kommunikation scheidet oft an unterschiedlichen meist proprietären Protokollen. Jedes Gerät bietet als Einzelkämpfer dem Benutzer ausschließlich die eigenen Dienste an. Durch die herstellernabhängigen Spezifikationen Audio Video Control (AV/C) [13901a] und Home Audio Video interoperability (HAVi) [HAV01] soll erreicht werden, dass digitale Unterhaltungselektronikgeräte verschiedener Hersteller miteinander über einen gemeinsamen Standard kommunizieren können. Beide basieren auf dem seriellen Hochgeschwindigkeitsbus IEEE 1394, welcher Plug-and-Play zur einfachen Vernetzung von Geräten bietet und Datenraten von bis zu 1600 Mbit/s unterstützt. Während AV/C auf dem Client/Server-Prinzip basiert, bietet HAVi echtes Peer-to-Peer-Networking. In den folgenden Unterabschnitten

werden die Standards bzw. Spezifikationen IEEE 1394, AV/C und HAVi beschrieben.

2.1.2.1 IEEE 1394

Der serielle Hochgeschwindigkeitsbus IEEE 1394 (FireWire, iLink) wurde ursprünglich bei der Firma Apple entwickelt und 1995 zum IEEE-Standard IEEE 1394-1995 [IEE95, And99]. Die Eigenschaften und seine Funktionsweise sollen hier kurz behandelt werden. Einige seiner markantesten Leistungsmerkmale sind:

- maximal 63 Knoten pro Bus, max. 1023 Busse (P1394.1)¹⁹
- Punkt-zu-Punkt Kabelverbindung, jeder Knoten agiert als Repeater (maximale Distanz zwischen zwei Knoten 4,5 m; IEEE 1394b bis zu 100 m)
- automatische Konfiguration aller Geräte am Bus (Plug-and-Play)
- An- und Abstecken der Geräte während des Betriebs (Hot-Plug)
- verschiedene Datenraten 100, 200 und 400 Mbit/s (IEEE 1394b bis 1600 Mbit/s)
- zwei Übertragungsarten: asynchronous mode (bestätigte Übertragung) und isochronous mode (garantierte Bandbreite)
- konsistent zu ISO/IEC 13213:1994 (IEEE Std. 1212-1991 Control und Status Register Standard) fixe 64-Bit Adressierung
- optionale Versorgung über das Buskabel (8-40 VDC / 1,5 A)

An einen Bus können bis zu 63 Geräte (Knoten) angesteckt werden. Die einzelnen Geräte besitzen dazu einen oder mehrere Anschlüsse und können über eine maximal 4,5 m lange Punkt-zu-Punkt Kabelverbindung verbunden werden. Das 6-adrige Verbindungskabel besteht aus zwei Signalladerpaaren und einem Paar, das die Stromversorgung über den Bus (8-40 VDC / 1,5 A) ermöglicht. Die einzelnen Knoten agieren als Repeater zwischen den Knoten und simulieren damit einen einzigen logischen Bus.

Der serielle Bus verwendet zur Adressierung das fixe 64-bit Adressierungsmodell der CSR-Architektur [IEE02a]. In Abbildung 2.1 ist der Adressbereich dargestellt. Die ersten 10 Bit bezeichnen den Bus (bus_ID, 0..1022, lokaler Bus 1023), die nächsten 6 Bit den Knoten am Bus (physical_ID, 0..62, Broadcast 63). Diese ersten 16 Bit werden auch als *NodeID* bezeichnet. Die restlichen 48 Bit bilden den lokalen (virtuellen) Speicherbereich. Gemäß der CSR-Architektur²⁰ sind die letzten 256 MB des lokalen Adressbereichs für Register vorgesehen (ab 0xFFFFF0000000). In diesem Registerbereich sind in den ersten 512 Byte die Core-Control- und Status-Register enthalten, in den folgenden 512 Byte die Serial-Bus-Register und danach das 1 kByte große Configuration Rom (*ConfigRom*), welches unter anderem zur Beschreibung des Gerätes verwendet wird (z. B. HAVi-IAV-Gerät mit Stream Manager).

¹⁹Die Verbindung der einzelnen IEEE 1394 Busse wird in einem eigenen Standard IEEE 1394.1 [Joh02] behandelt.

²⁰CSR steht für „Control and Status Register“.

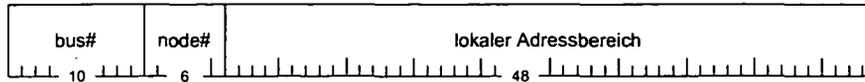


Abbildung 2.1: IEEE 1394-Adressierung

Wird ein Gerät an den Bus gesteckt, so wird ein Bus-Reset ausgelöst und der Bus automatisch neu konfiguriert. Diese Businitialisierung besteht aus den drei Phasen *Reset*, *Tree Identify* und *Self Identify*. In der ersten Phase werden alle Knoten am Bus vom Bus-Reset informiert, in der zweiten Phase wird die Topologie des Busses ermittelt und in der letzten Phase wird unter anderem jedem Gerät die *physical_ID* zugewiesen. Die *physical_ID* - und damit auch die *NodeID* - des Gerätes kann sich also bei jedem Bus-Reset ändern. Zur dauerhaft eindeutigen Adressierung eines Gerätes dient der 64 Bit große Extended Unique Identifier (EUI-64), der auch als Global Unique Identifier (*GUID*) bezeichnet und am Beginn des ConfigRoms gespeichert wird.

Die Übertragung von Daten kann auf zwei Arten erfolgen:

- *Asynchronous Mode*
Bestätigte Übertragung von Paketen variabler Größe zu einer bestimmten Adresse am Bus (Knoten). Hierbei ist die garantierte Auslieferung der Pakete wichtiger als der Übertragungszeitpunkt. Wird ein Paket fehlerhaft übertragen, so kann die Übertragung wiederholt werden. Für diese Übertragungsart stehen drei Transaktionstypen zur Verfügung: *read*, *write*, *lock*.
- *Isochronous Mode*
Unbestätigte Übertragung von Broadcast-Paketen variabler Größe innerhalb eines garantierten Zeitintervalles ($125 \mu s$) über einen logischen Kanal (0..63). Dieser Modus eignet sich beispielsweise zur Übertragung von Audio/Video-Daten. Eine Übertragung zu garantierten Zeitpunkten ist hierbei wichtiger als eine garantierte Auslieferung. Die Wiederholung eines einzelnen fehlerhaft übertragenen Videobildes ist nutzlos, weil der geforderte Darstellungszeitpunkt in der Vergangenheit liegt.

Auf Grund der ausgeführten Leistungsmerkmale eignet sich der IEEE 1394-Bus ausgezeichnet zur Vernetzung von Geräten der Unterhaltungselektronik. Er bietet jedoch nur die Basis für Protokolle höherer Schichten, wie z. B. AV/C oder HAVi. Diese werden in den nächsten beiden Abschnitten erläutert.

2.1.2.2 Audio Video Control (AV/C)

Die Spezifikation des AV/C-Protokolls setzt sich aus mehreren von der 1394 Trade Association herausgegebenen Spezifikationen zusammen. Als grundlegende Spezifikation definiert die AV/C Digital Interface Command Set General Specification [13901a] ein allgemeines Befehl-Antwort-Verfahren zur Gerätekommunikation, *Units* und *Subunits*

zur Abstraktion eines Gerätes und dessen funktionellen Komponenten, und einen Satz von Unit- und Subunit-Befehlen zur Ansteuerung der Geräte. Die Beschreibung des Gerätes und der Funktionen erfolgt einerseits über in [13999b] spezifizierte Einträge im ConfigRom des Gerätes und andererseits durch den in [13902a] definierten so genannten *Descriptor Mechanism*. Letzterer definiert hierarchische und verschachtelungsfähige Datenstrukturen und Befehle zum Erstellen, Lesen und Schreiben dieser Datenstrukturen. Die einzelnen Gerätetypen sind in einer Reihe von weiteren Dokumenten spezifiziert, wie beispielsweise „Disc Subunit“ [13902c, 13902b], „Tape Recorder/Player Subunit“ [13901b], „Monitor Subunit“ [13999a] oder „Tuner Subunit“ [13998b, 13998a, 13999c].

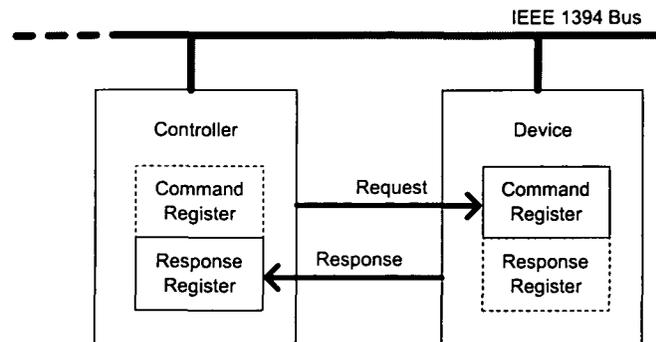


Abbildung 2.2: AV/C-Kommunikation

Das Befehl-Antwort-Verfahren zur Gerätekommunikation basiert auf dem im IEC-Standard IEC 61883-1 [IEC98] definierten *Function Control Protocol* (FCP), welches ein Paketformat und einen Adressbereich für die asynchrone Übertragung von Befehlen und Antworten über IEEE 1394 definiert. Damit FCP von mehreren höheren Protokollen verwendet werden kann, enthalten die ersten vier Bits des FCP-Rahmens den Command/Transaction Set Code (*CTS-Code*), welcher das verwendete höhere Protokoll bezeichnet. Eine AV/C-Transaktion besteht aus mindestens zwei Teilen. Den ersten Teil bildet das Schreiben des Befehls in das 512 Byte große *FCP_COMMAND*-Register des Zielgerätes (Abbildung 2.2, Request). Kann der Befehl innerhalb von 100 ms im Zielgerät durchgeführt werden, so wird im zweiten Teil der Transaktion die Antwort in das 512 Byte große *FCP_RESPONSE*-Register des Steuergerätes geschrieben (Abbildung 2.2, Response). Anderenfalls wird die Antwort *INTERIM* gesendet, und in einem dritten Teil erfolgt nach Abarbeitung des Befehls das Senden der endgültigen Antwort.

Ein AV/C-Gerät besteht aus einer Unit und einer oder mehreren Subunits, welche die speziellen Gerätefunktionen repräsentieren. Die Subunits werden über den Subunit-Type und die Subunit-ID innerhalb des AV/C-Gerätes adressiert. In Abbildung 2.3 ist der Aufbau eines IEEE 1394-Block-Write-Packets zum Senden eines AV/C-Befehls dargestellt. Den ersten Teil des Pakets bildet der IEEE 1394-Header, der unter anderem die Ziel- und Quell-NodeID der beteiligten AV/C-Geräte und als „destination_offset“ die Adresse des *FCP_COMMAND*-Registers (0xFFFFF000B00) enthält. Am Beginn des

AV/C-Befehls steht der CTS-Code für AV/C (0000b). Der „ctype“ bezeichnet den Befehlstyp, der „subunit.type“ und die „subunit_ID“ (subID) adressieren die Subunit, der „opcode“ bezeichnet den auszuführenden Befehl und die darauf folgenden „operands“ sind die Parameter des Befehls. Aufgrund der Größe der FCP-Register beträgt die maximale Befehlsgröße 512 Bytes.

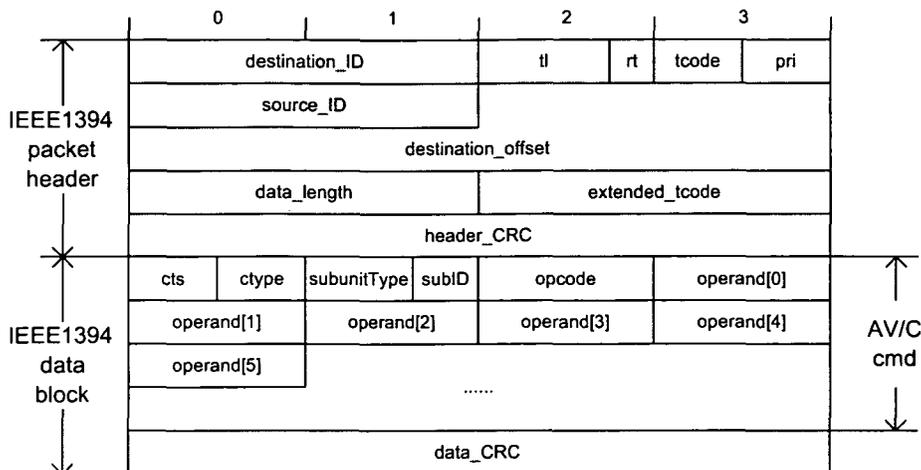


Abbildung 2.3: AV/C-Request-Packet

In AV/C sind fünf Befehlstypen (CONTROL, STATUS, SPECIFIC INQUIRY, NOTIFY, GENERAL INQUIRY) und sieben Antworttypen (NOT IMPLEMENTED, ACCEPTED, REJECTED, IN TRANSITION, IMPLEMENTED/STABLE, CHANGED, INTERIM) definiert. Ein Benachrichtigungsmechanismus ist durch den Befehlstyp NOTIFY spezifiziert. Damit ist es der Kontrollanwendung am Steuergerät möglich, auf Änderungen im Zielgerät zu reagieren. Die Kontrollanwendung sendet dazu den entsprechenden Befehl mit NOTIFY als Befehlstyp und erhält unmittelbar als Zwischenantwort (INTERIM) den aktuellen Status, wenn das Gerät für diesen Befehl den Befehlstyp NOTIFY unterstützt. Bei Änderung des Zustandes im Zielgerät wird die endgültige Antwort CHANGED mit dem veränderten Zustand an das Steuergerät gesendet. Zu beachten ist, dass dieser Benachrichtigungsmechanismus nur eine „one-shot operation“ ist und zum Empfangen einer weiteren Zustandsänderung der Befehl erneut zu senden ist. Außerdem ist auch nach einem Bus-Reset der Befehl erneut zu senden, da der AV/C-Stack die Geräte nur anhand der dynamischen zugewiesenen NodeID und nicht anhand der statischen (dauerhaften) GUID identifiziert. Eine eindeutige Identifizierung des Steuergerätes wäre nicht möglich, weil die NodeIDs bei einem Bus-Reset neu vergeben werden.

2.1.2.3 Home Audio Video interoperability (HAVi)

HAVi ist die Abkürzung für Home Audio Video interoperability. Acht große Unternehmen der Unterhaltungselektronik-Industrie²¹ haben unter diesem Namen im Mai 2001 eine Spezifikation zur Vernetzung von digitalen Unterhaltungselektronikgeräten herausgegeben [HAV01]. HAVi ist eine Middleware und bildet die mittlere Schicht zwischen einer plattform-abhängigen API (OS, IEEE 1394-Treiber, ...) und einer von HAVi spezifizierten plattformunabhängigen API (*Interoperability API*).

Um den Anforderungen des Marktes zu entsprechen definiert HAVi vier Gerätetypen. Einerseits müssen bestehende Geräte (z.B. Geräte ohne IEEE 1394-Anschluss) in das Heim-Netz integrierbar sein (LAV) und andererseits müssen Geräte mit unterschiedlichen Funktionalitäten für verschiedene Preissegmente verfügbar sein (BAV, IAV und FAV):

- FAV (Full AV Device)
Ein FAV enthält einen vollständigen HAVi-Stack einschließlich einer Java Virtual Machine zum Ausführen von Anwendungen und Gerätemodulen in Java Bytecode. Typische Vertreter für FAV sind Set-Top-Boxen und digitale Fernsehgeräte.
- IAV (Intermediate AV Device)
Ein IAV enthält einen HAVi-Stack, aber keine Java Virtual Machine.
- BAV (Base AV Device)
BAVs sind Geräte mit IEEE 1394-Anschluss und einer HAVi konformen Beschreibung des Gerätes im Configuration Rom des IEEE 1394-Adapters. Auf diesen Geräten wird kein HAVi-Stack ausgeführt, eine Ansteuerung erfolgt über einen FAV oder IAV. Der FAV kann dazu die Software zur Ansteuerung des Gerätes (DCM) vom BAV laden oder wie ein IAV eine bereits enthaltene DCM für das Gerät verwenden.
- LAV (Legacy AV Device)
Dieser Gerätetyp repräsentiert die derzeitigen Geräte der Unterhaltungselektronik; sie besitzen keine Kenntnis der HAVi-Architektur und benötigen einen FAV oder IAV, der eine passende Software zur Ansteuerung des Gerätes (DCM) enthält, um Dienste im Heim-Netz anbieten zu können.

Die Kommunikation zwischen einem Gerät mit HAVi-Stack (FAV oder IAV) und einem Gerät ohne HAVi-Stack (BAV oder LAV) ist herstellerabhängig und erfordert die Umsetzung der HAVi-Befehle in entsprechende Befehle des BAVs bzw. LAVs.

In der HAVi-Architektur stellen Software-Elemente (SE) einander Dienste über eine definierte Schnittstelle (API) zur Verfügung. Die Dienste können in drei Gruppen eingeteilt werden: Systemdienste, Gerätedienste, Dienste für die Benutzerschnittstelle. Jene

²¹Grundig AG, Hitachi Ltd., Matsushita Electrical Industrial Co. Ltd. (Panasonic), Royal Philips electronics, Sharp Corp., Sony Corp., Thomson Multimedia und Toshiba Corp.

SE, die ausschließlich Systemdienste zur Verfügung stellen, werden als System-SE bezeichnet. Dazu gehören: *Communication Media Manager* (Cmm1394), *Messaging System* (MS), *Registry* (REG), *Event Manager* (EM), *Stream Manager* (SM), *Resource Manager* (RM) und der *Device Control Module Manager* (DCMM). Zu den SE, die Gerätedienste bereitstellen, gehören das *Device Control Module* (DCM) zur Abstraktion eines Gerätes und die *Functional Component Modules* (FCM) zur Abstraktion der einzelnen funktionellen Komponenten eines Gerätes, wie z. B. die Tuner-FCM, AV-Display-FCM oder VCR-FCM. Diese SE können im Gegensatz zu den System-SE in einem HAVi-Stack auch mehrfach vorkommen. Jedem SE wird bei der Anmeldung im Messaging System ein eindeutiger *Software Element Identifier* (SEID) zugewiesen. Die zehn Byte große SEID setzt sich aus der GUID des lokalen Gerätes (8 Bytes) und einem 2 Byte großen *Software-Handle* (swHandle) zusammen. Abbildung 2.4 zeigt überblicksartig die Anordnung der einzelnen SE im HAVi-Stack. Daraus ist auch ersichtlich, dass HAVi die mittlere Schicht zwischen einer plattformabhängigen API (Platform Specific API) und einer plattformunabhängigen API (Interoperability API) bildet. Nachfolgend werden die Aufgaben der einzelnen SE beschrieben.

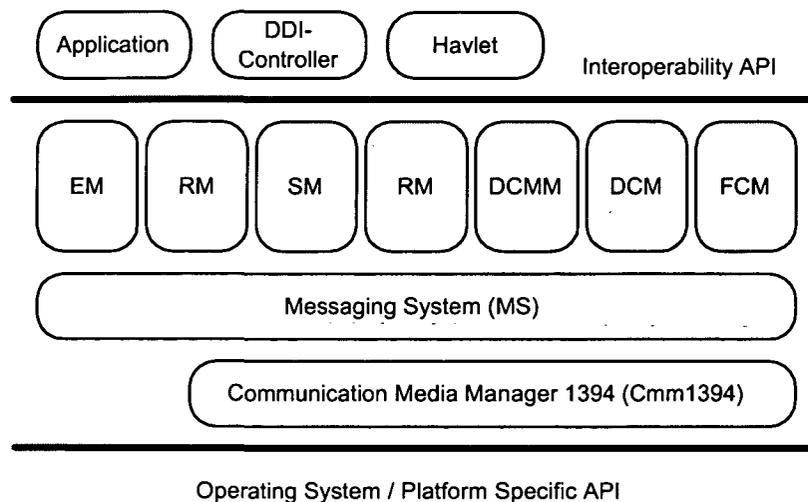


Abbildung 2.4: HAVi-Software-Architektur

Communication Media Manager (Cmm1394): Der Cmm1394 dient als Schnittstelle zum Übertragungsmedium, dem IEEE 1394-Bus. Er stellt zwei Arten von Diensten den lokalen SE zur Verfügung. Zur ersten Art gehören die Dienste zur Kommunikation mit anderen Geräten. Das sind die Funktionen Read, Write, Lock und die Funktionen DropIndication und EnrollIndication. Letztere ermöglicht das Buchen eines lokalen Adressbereichs, um von jedem Zugriff auf diesen Bereich benachrichtigt zu werden. Die zweite Art von Diensten dient zur Abstrahierung des IEEE 1394-Busses. Dazu verwaltet der Cmm1394 Netzwerkinformationen und aktualisiert diese bei jedem Bus-Reset.

Er arbeitet mit dem Event Manager zusammen, um Topologie-Änderungen des Busses (z. B. An- oder Abstecken von Geräten) zu melden.

Messaging System (MS): Das MS ist verantwortlich für den Nachrichtenaustausch zwischen den Software-Elementen (SE). Damit SE den verbindungslosen Übertragungsdienst nutzen können, müssen sie sich beim lokalen MS anmelden. Dabei übergeben sie diesem eine Callback-Funktion (CB) und erhalten eine SEID, mit welcher sie im gesamten Netzwerk eindeutig adressierbar sind. Die CB-Funktion wird vom lokalen MS aufgerufen, um eine Nachricht an das SE zu übergeben. Die Kommunikation zwischen den SE erfolgt über Nachrichten und ist in Abbildung 2.5 nochmals dargestellt.

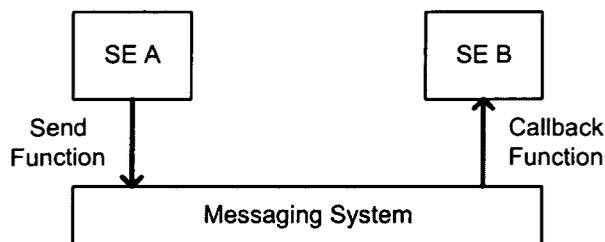


Abbildung 2.5: Software-Element-Kommunikation

Zur Übertragung von Nachrichten über den IEEE 1394-Bus wird das in IEC 61883-1 [IEC98] spezifizierte Function Control Protocol (FCP) verwendet. FCP definiert ein Paketformat für die asynchrone Übertragung von Befehlen und Antworten. Im Gegensatz zum Standard IEC 61883-1 wird nur das 512 Byte große *FCP_COMMAND*-Register verwendet, in welches alle Nachrichten geschrieben werden (sowohl Befehle als auch Antworten). In Abbildung 2.6 ist die allgemeine Struktur eines HAVi-Nachrichtenpaketes dargestellt.

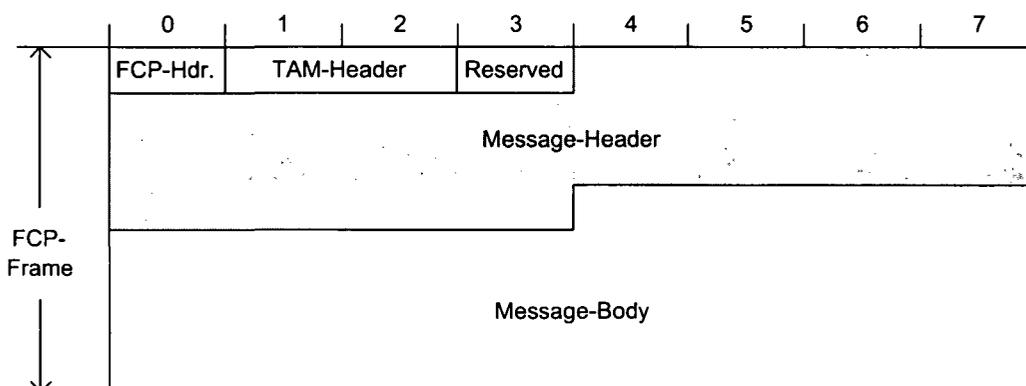


Abbildung 2.6: Allgemeine Struktur einer HAVi-Nachricht

Das erste Byte besteht aus dem 4 Bit großen CTS-Code (0011b) und 4 von HAVi reservierten Bits. Die maximale Paketgröße beträgt 512 Bytes (Abbildung 2.6, FCP-Frame). Um auch größere Nachrichten versenden zu können, ist eine Zerstückelung in einzelne Pakete notwendig. Damit der Empfänger die Pakete wieder richtig zusammenstellen kann, enthält der TAM-Header eine Kennzeichnung des Pakettyps und eine laufende Paketnummer. Der Message-Header enthält unter anderem die SEID des Empfänger-SE, des Absender-SE und die gesamte Länge der Nachricht. Der Message-Body enthält die Nachricht für das SE.

Für jede Funktion der API ist ein eindeutiger Operation Code definiert. Wenn ein SE eine API-Funktion eines anderen SE aufrufen möchte, dann muss es die Funktionsparameter in eine Nachricht umwandeln und gemeinsam mit dem entsprechenden Operation Code über das MS an das Ziel-SE senden. Ausgenommen davon sind die MS-Funktionen, da diese über einen normalen Funktionsaufruf aufgerufen werden. Beim Empfänger wird die Nachricht wieder in einen Funktionsaufruf umgewandelt. Hat die aufgerufene Funktion einen Rückgabewert, so wird dieser ebenfalls in eine Nachricht umgewandelt und zurück an den Aufrufer gesendet. Durch diese Vorgangsweise gibt es für das SE keinen Unterschied zwischen einer Kommunikation mit einem lokalen oder einem entfernten SE. Das Senden der Nachricht erfolgt immer durch Aufruf einer Sendefunktion des MS. Dieses sendet die Daten an das MS des Ziel-SE. Dort wird die Nachricht durch Aufruf der Callback-Funktion des Ziel-SE übergeben. Zum Senden von Nachrichten stellt das MS zwei Funktionen zur Verfügung:

- `MsgSendSimple`: Senden ohne Empfangsbestätigung
- `MsgSendReliable`: Senden mit Empfangsbestätigung

Für das oben beschriebene Verfahren zum Aufruf von Funktionen anderer SE bietet das MS die Funktionen `MsgSendRequestSync` und `MsgSendRequest`. Diese Funktionen ermöglichen die Zuordnung der Antwort zu einem Funktionsaufruf durch Vergabe einer Transaktionsnummer (`transactionID`). Zusätzlich wird vor die Nachricht ein Funktions-Header hinzugefügt, der den Operation Code der Funktion, die Transaktionsnummer und die Control-Flags enthält. Letzteres kennzeichnet die Nachricht als Funktionsanfrage oder -antwort. `MsgSendRequestSync` ist eine Funktion zur Durchführung eines synchronen Nachrichtentransfers, d. h. der Aufrufer wird so lange blockiert bis eine Antwort auf die gesendete Anfrage eingetroffen oder eine definierte Zeit abgelaufen ist. Die Funktion `MsgSendRequest` ist eine Funktion zur Durchführung eines asynchronen (nicht blockierenden) Nachrichtentransfers, d. h. der Aufrufer wird nicht blockiert und erhält für die Zuordnung der später eintreffenden Antwort eine Transaktionsnummer. Beide Funktionen verwenden zum Senden die Funktion `MsgSendReliable`. Zum Senden einer Funktionsantwort dient die Funktion `MsgSendResponse`, die sowohl über `MsgSendSimple` als auch über `MsgSendReliable` erfolgen kann. Ein Beispiel für einen synchronen Nachrichtentransfer ist in Abbildung 2.7 dargestellt.

Bei der Umwandlung der Funktionsparameter in eine Nachricht müssen alle Funktionsparameter in einen „genormten“ Bytestrom umgewandelt werden. Man nennt dies

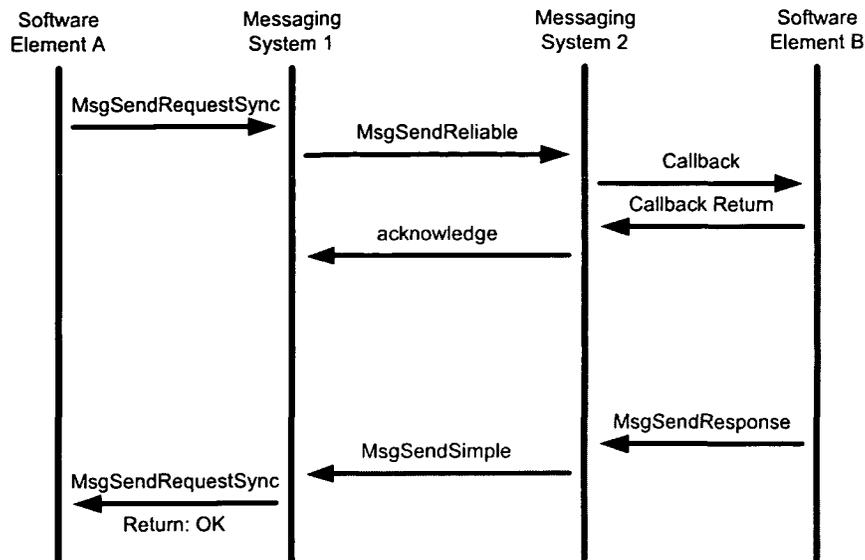


Abbildung 2.7: Beispiel für einen synchronen Nachrichtentransfer

Marshalling und deren Rückwandlung *Unmarshalling*. Dieser Vorgang ist für die Kommunikation zwischen Geräten unterschiedlicher Hersteller zur richtigen Interpretation der Daten unerlässlich. Die Umwandlung erfolgt in Anlehnung an den „Common Data Representation (CDR)“-Übertragungssyntax [Obj98].

Event Manager (EM): Der *Event Manager* ist ein Ereignis-Mitteilungsdienst. Ein Event ist eine Zustandsänderung eines SE (z. B. Videorekorder wechselt in den Wiedergabemodus oder ein neues Gerät wird an den Bus gesteckt). Es ist die Aufgabe des SE bei einer Zustandsänderung das entsprechende Event auszulösen. Dazu sendet das SE eine Nachricht an den EM, welche das auszulösende Event und die Information, ob dieses nur lokal oder im ganzen Netzwerk (global) gemeldet werden soll, enthält. SE können verschiedene Events bei ihrem lokalen EM abonnieren. Für die Auslieferung der Events ist der EM verantwortlich. Er führt dazu für jedes Event eine Liste von interessierten SE. Wird ein Event ausgelöst, dann benachrichtigt er alle lokalen SE der Abonnentenliste und, wenn das Event global gemeldet werden soll, auch alle anderen EM im Netzwerk. Empfängt ein EM eine Benachrichtigung eines anderen EM, so benachrichtigt er wiederum alle an diesem Event interessierten lokalen SE. Die Benachrichtigung der SE erfolgt durch ein so genanntes *Message Back* (MB). Das heißt die interessierten SE geben beim Abonnieren von Events einen Operation Code einer selbst definierten Funktion zur Event-Behandlung an. Wird ein abonniertes Event ausgelöst, dann sendet der EM („Server“) eine Nachricht bestehend aus dem registrierten Operation Code und einigen Eventinformationen an das SE („Client“). Diese Kommunikationsart wird *Message Back* bezeichnet, weil das Senden der Nachricht vom Server und nicht vom Client initiiert wird.

Registry (REG): Die *Registry* verwaltet ein Verzeichnis von lokalen SE. Sie stellt Dienste zum Registrieren lokaler SE und zum Suchen aller im HAVi-Netzwerk vorhandenen SE zur Verfügung. Jedes SE kann seine Eigenschaften (*Attributes*) unter Angabe der eigenen SEID in der Registry speichern, um diese im Netzwerk bekannt zu machen. Es gibt Einträge, die für bestimmte SE-Typen (System-SE, DCM/FCM, Anwendungen) verpflichtend sind. Die Tabelle 2.1 zeigt dazu einen Auszug aus der Liste der vordefinierten Attribute. Die Angabe des SE-Typs ist beispielsweise für alle SE zwingend. Das Attribut ATT_AV_LANG ist hingegen optional.

Attributname	Typ	verpflichtend für
ATT_SE_TYPE	SoftwareElementType	alle
ATT_VENDOR_ID	VendorId	DCM/FCM/APP
ATT_HUID	HUID	DCM/FCM/APP
ATT_AV_LANG	AvLanguage	-
ATT_USER_PREF_NAME	UserPreferredName	DCM/FCM

Tabelle 2.1: Vordefinierte Attribute der Registry (Auszug)

Sucht ein SE andere SE mit bestimmten Eigenschaften, so sendet es eine Suchanfrage (Query) an seine lokale Registry (Funktion `GetElement` oder `MultipleGetElement`). Diese sucht lokal nach passenden SE (identifiziert durch die SEID) und leitet die Suchanfrage auch an alle anderen im Netzwerk vorhandenen Registries weiter. Aus allen Antworten erstellt die lokale Registry eine gemeinsame Liste mit allen gefundenen SEIDs und liefert diese an den Aufrufer zurück.

Device Control Module (DCM), Functional Component Module (FCM): Ein Gerät wird durch ein *Device Control Module* und alle seine funktionellen Komponenten durch *Functional Component Modules* repräsentiert (z. B. Tuner, Display, VCR). Zur Ansteuerung eines Gerätes ist im Netzwerk eine DCM gemeinsam mit ihren FCMs notwendig. Jede DCM und jede FCM ist ein eigenständiges SE und speichert die grundlegenden Eigenschaften in der lokalen Registry. Zusätzlich zur SEID ist jeder DCM und FCM ein dauerhaft eindeutiger *HAVi Unique Identifier* (HUID) zugewiesen. Während die SEID nach jedem Einschalten des Gerätes für alle SE, ausgenommen der System-SE, neu vergeben wird, ist die HUID dauerhaft mit einer DCM bzw. FCM verbunden. Zusätzlich ist für eine benutzerfreundlichere Identifikation eines Gerätes vorgesehen, für jedes Gerät einen so genannten *User Preferred Name* zu wählen (z. B. „Wohnzimmer TV“ oder „Michi's DVD Player“). Dieser Name wird sowohl in der Registry als auch in der DCM gespeichert²². Die Kommunikation mit dem Gerät erfolgt immer über die DCM bzw. FCM, aber niemals direkt mit dem Gerät. Dazu sind derzeit für die Gerätefunktionen „Tuner“, „VCR“, „Clock“, „Camera“, „AV Disc“, „Amplifier“, „Display“, „AV Display“,

²²Im Falle eines IEEE 1394-Gerätes wird der User Preferred Name auch noch ins lokale ConfigRom des Gerätes eingetragen

„Modem“ und „Web Proxy“ API-Funktionen spezifiziert. Durch dieses Konzept ist es möglich, dass Geräte verschiedener Hersteller über den gleichen Befehl angesteuert werden können.

Eine *DCM-Code-Unit* (DCU) besteht aus einem Initialisierungs- und Beendigungsteil, einer DCM und den zugehörigen FCMs. Eine DCU kann in Java oder in „native Code“ geschrieben werden. Die erste Variante hat den Vorteil, dass die Programmausführung unabhängig von der verwendeten Plattform in einer Java Virtual Machine (JVM) möglich ist. Sie wird auch als „uploadable DCU“ bezeichnet und kann von jedem FAV aus dem ConfigRom des fremden Gerätes oder von einer URL geladen und ausgeführt werden. Eine DCU, die in „native Code“ geschrieben wurde, kann von einem FAV oder IAV ausgeführt werden, muss jedoch bereits am Zielgerät vorhanden sein und wird deshalb auch als „embedded DCU“ bezeichnet. Für die Installation von DCMs fremder Geräte ist ein DCM-Manager notwendig. Ein FAV enthält immer einen DCM-Manager, ein IAV nur, wenn eine DCM eines fremden Gerätes ausgeführt werden soll. Die Installation einer DCM für das lokale Gerät ist dem Gerätehersteller überlassen.

Device Control Module Manager (DCM-Manager): Der *DCM-Manager* ist verantwortlich für die Installation und Deinstallation von DCM-Code-Units auf dem lokalen Gerät zur Ansteuerung von BAV- oder LAV-Geräten mit IEEE 1394-Anschluss. Die Installation einer DCM-Code-Unit beinhaltet das Laden der Code-Unit und der Installation der DCM und ihrer FCMs. Eine „uploadable DCU“ wird gegenüber einer „embedded DCU“ bevorzugt, im ganzen Netzwerk darf für ein Gerät nur genau eine DCM installiert werden. Dazu bilden alle DCM-Manager im HAVi-Netzwerk ein DCM-Management-System. Beim Ein- oder Ausstecken eines Gerätes wird ein Bus-Reset ausgelöst. Die DCM-Manager reagieren deshalb auf das vom Cmm1394 ausgelöste Event „Network-Reset“ und beginnen mit der Identifizierung des *Initial Leader*. Dazu wird für jeden DCM-Manager eine Kennzahl bestimmt und jener mit der höchsten Kennzahl gewählt. Die Kennzahl wird aus der bitweise umgedrehten GUID gebildet. Da die ersten 3 Bytes der GUID den Hersteller kennzeichnen, würde sonst ein bestimmter Hersteller bevorzugt werden. Der Initial Leader holt von allen anderen DCM-Managern, den so genannten *Initial Followers*, Informationen über deren Fähigkeiten und aktuellen Auslastung ein und entscheidet nach einer vorgegebenen Rangordnung, welcher DCM-Manager der *Final Leader* wird, indem er an diesen eine entsprechende Nachricht sendet. Der Final Leader teilt allen anderen Knoten mit, dass sie *Final Follower* und damit dem Final Leader untergeordnet sind. Die Auswahl eines geeigneten FAV bzw. IAV zur Installation der DCM-Code-Unit erfolgt nunmehr unter der Leitung des Final Leaders.

Stream Manager (SM): Der *Stream Manager* bietet allen lokalen SE benutzerfreundliche Funktionen zum Auf- und Abbau von Verbindungen zwischen zwei FCMs zur isochronen Datenübertragung. Über weitere Funktionen stellt er allen SE Verbindungsinformationen zur Verfügung. Es werden drei Transporttypen unterschieden: INTERNAL,

IEC61883 und CABLE. Der erste Typ bezeichnet eine Verbindung zwischen zwei FCM-Anschlüssen innerhalb eines Gerätes, die anderen beiden eine Verbindung eines DCM-Anschlusses zum IEEE 1394-Bus (IEC61883) oder einer anderen Kabelverbindung (CABLE). Quelle und Senke von Audio/Video-Daten ist immer eine FCM, die Anschlüsse nach außen werden jedoch über die DCM verwaltet, da diese das gesamte Gerät und nicht nur eine einzelne Komponente betreffen. Für eine Verbindung zwischen zwei Geräten ist deshalb immer auch eine Verbindung zwischen einem FCM-Anschluss und einem DCM-Anschluss bei jedem Gerät notwendig, welche auch als *Attachment* bezeichnet wird. Eine FCM kann mit einer anderen FCM über eine gebundene (*Point-to-Point*) Verbindung oder über zwei ungebundene (*Broadcast*) Verbindungen verbunden sein. Beim Aufbau einer isochronen Verbindung übernimmt der SM entsprechend den Funktionsparametern:

- die Kompatibilitätsprüfung (Richtung, Format, Transport, ...) der Anschlüsse der beteiligten FCMs und DCMs,
- die Konfiguration der geräteinternen Verbindungen zwischen den FCM-Anschlüssen, im Falle einer INTERNAL Verbindung, bzw. zwischen den FCM- und DCM-Anschlüssen, im Falle einer externen Verbindung (IEC61883 oder CABLE),
- die Reservierung von Übertragungsressourcen (IEEE 1394-Bandbreite, IEEE 1394-Kanal und IEC 61883-Plug-Control-Register), im Falle einer IEC61883-Verbindung.

Dabei nutzt er Funktionen der an der Verbindung beteiligten DCMs und FCMs. Zu jeder Verbindung vergibt der SM eine global eindeutige Nummer, den *Connection Identifier*. Über diese Nummer können SE eindeutig auf die Verbindung verweisen. Auch für die Wiederherstellung von IEC61883-Verbindungen nach einem Bus-Reset ist der SM verantwortlich. Der Stream Manager führt zur Verwaltung der Verbindungen eine Liste mit allen von ihm erstellten Verbindungen und ihren Eigenschaften (beteiligte SE, Anschlüsse, Format, Transport, ...). Er überwacht die erstellten Verbindungen und baut unterbrochene Verbindungen automatisch ab (z. B. wenn eine an der Verbindung beteiligte FCM nicht mehr verfügbar ist oder die Wiederherstellung nach einem Bus-Reset fehlschlägt).

Resource Manager (RM): Der *Resource Manager* bietet Funktionen zum Reservieren und Freigeben von Gerätere Ressourcen (FCMs) und Strategien zur Lösung von Zugriffskonflikten. Durch die Reservierung soll sichergestellt werden, dass SE, die eine FCM reserviert haben, nicht durch andere SE gestört werden. Eine Gruppe von FCMs wird nur reserviert, wenn alle FCMs reserviert werden können. Außerdem bietet der RM Funktionen für geplante Aktionen (*Scheduled Actions*, wie z. B. dem Aufnehmen eines Fernsehprogramms durch einen Videorecorder).

Die Vorgangsweise zum Reservieren einer FCM und die Lösung damit verbundener Zugriffskonflikte wird anhand des Beispiels aus Abbildung 2.8 beschrieben. SE, die eine FCM reservieren, werden hierin als „Clients“ bezeichnet. Client2 hat die FCM VCR2 reserviert. Client1 sucht nach einem VCR im HAVi-Netzwerk und fragt dazu seine lokale

Registry (Pfeil 1, Registry1), welche die SEID der FCM VCR2 zurück gibt. Über den lokalen RM (RM1) versucht der Client1 die FCM VCR2 zu reservieren (Pfeil 2). Dazu fragt RM1 die FCM VCR2 nach dem aktuellen Reservierungsstatus. Entsprechend dem Status wird die Reservierung durchgeführt oder dem Client1 gemeldet, dass die FCM nicht frei ist. Im letzteren Fall kann Client1 den RM1 beauftragen, mit dem aktuellen Benutzer des VCR2 zu verhandeln (Pfeil 4). Die Verhandlung erfolgt, wie schon beim Event Manager, über ein Message Back, d. h. der RM1 erhält über eine erneute Anfrage an VCR2 die SEID und den Operation Code der Verhandlungsfunktion von Client2. Das Ergebnis der Verhandlungsanfrage (Pfeil 5) ist entweder eine Zurückweisung oder die Erlaubnis zur Benutzung der FCM VCR2. Entsprechend der Antwort kann Client1 eine Reservierung von VCR2 über RM1 durchführen (Pfeil 2). Bei der Verhandlung über FCMs gibt es zwei Arten von Clients: *User* und *System*. Für diese Clients ist definiert, wie sie sich bei einer Verhandlung verhalten können und ob sie trotz einer Zurückweisung eine Reservierung erzwingen dürfen.

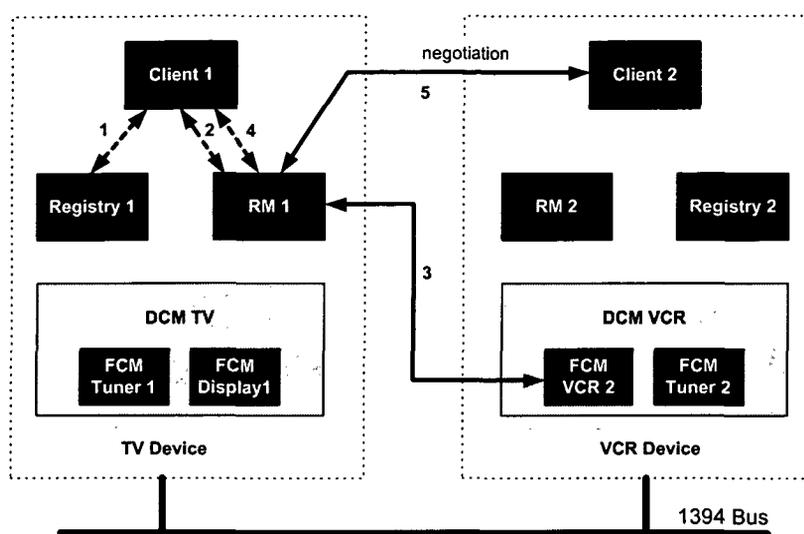


Abbildung 2.8: FCM-Reservierung

Application Module: Das Softwareelement *Application Module* ist konzeptionell einer DCM sehr ähnlich. Es besitzt genauso eine HUID und kann in Java oder in „native Code“ geschrieben werden. Wie ein FAV die Anwendung ladet und installiert, ist genauso wie die Installation einer „native Code“-Anwendung auf einem IAV bzw. FAV nicht festgelegt. Mit einer Anwendung können verschiedene Benutzeranforderungen realisiert werden, beispielsweise eine EPG oder eine Anwendung für ein digitales Fernsehgerät, die automatisch alle Geräte am IEEE 1394-Bus anzeigt und über eine eigene Benutzerschnittstelle ansteuert.

Benutzerschnittstelle (DDI, Havlet): Die HAVi-Architektur definiert zwei spezielle Mechanismen zur geräteunabhängigen Beschreibung einer Benutzerschnittstelle:

- Level 1 UI, *Data Driven Interaction* (DDI)
Das HAVi-UI-L1 setzt nicht Java voraus und ist deshalb speziell aber nicht ausschließlich für IAVs gedacht. Dabei ermöglicht ein SE einem Benutzer, ein anderes SE zu bedienen. Ersteres SE wird als *DDI-Controller* bezeichnet, letzteres als *DDI-Target*. Die Beschreibung der Benutzerschnittstelle erfolgt plattformunabhängig über DDI-Elemente (Icon, Button, Text, ...). Der DDI-Controller lädt diese DDI-Elemente vom DDI-Target und stellt diese entsprechend seinen Möglichkeiten (großer Bildschirm, kleines LCD-Display, ...) dem Benutzer dar. Der DDI-Controller sendet bei Benutzereingaben definierte Nachrichten an das DDI-Target und dieses antwortet mit einer Liste von zu ändernden DDI-Elementen. Außerdem reagiert der DDI-Controller auch auf Nachrichten vom DDI-Target, die aufgrund einer Zustandsänderung des zugehörigen SE gesendet werden (z. B. VCR: Wiedergabe beendet). Die Kommunikation zwischen DDI-Controller und DDI-Target wird als DDI-Protokoll bezeichnet.
- Level 2 UI, *Havlets*
Das Level 2 UI besteht aus Java-Packages, die auf einer Teilmenge der Java Abstract Windowing Toolkit (AWT) 1.1 basieren und im Wesentlichen Erweiterungen für die speziellen Benutzerschnittstellen von Unterhaltungselektronikgeräten enthalten. Ein Havlet ist eine Java HAVi-Anwendung, die zur Darstellung einer Benutzerschnittstelle Funktionen aus den HAVi-UI-L2-Packages verwendet. Durch die Standardisierung des HAVi-UI-L2 ist die Entwicklung und Ausführung einer graphischen Benutzerschnittstelle nicht mehr an die speziellen Fähigkeiten eines bestimmten Gerätes gebunden.

2.1.3 Small Office/Home Office (PT)

Das PC-Netzwerk im Heim ist IP-basiert²³ und aufgrund der vielseitigen Einsetzbarkeit von IP steht damit eine große Anzahl von Übertragungssystemen zur Verfügung. Dies sind vor allem kabelgebundene Ethernet-LANs (IEEE 802.3 [IEE02b]), Buszugriffsverfahren: CSMA/CD²⁴)²⁵ mit Übertragungsraten je nach Übertragungssystem von 10 bis 1000 Mbit/s oder Wireless-LANs [Ros02b] (IEEE 802.11 [IEE99a])²⁶, Buszugriffsverfahren: CSMA/CA²⁷ und ETSI HiperLAN2 [ETS00, Joh99], Buszugriffsverfahren: TDMA²⁸) mit Übertragungsraten je nach Übertragungssystem von 2 bis 54 Mbit/s.

Als zwei typische Vertreter für Standards/Spezifikationen zur Vernetzung von Geräten des Bereichs Small Office/Home Office wird in Abschnitt 2.1.3.1 Universal Plug and Play (UPnP) [Mic00] und in Abschnitt 2.1.3.2 Jini [Sun03b] beschrieben.

²³Internet Protocol (IP) [Pos81a]; Transmission Control Protocol (TCP) [Pos81b]; User Datagram Protocol (UDP) [Pos80]

²⁴CSMA/CA steht für Carrier Sense Multiple Access with Collision Detection

²⁵10Base2, 10Base-T, 100Base-TX, 1000Base-T, ...

²⁶IEEE 802.11a [IEE99b], IEEE 802.11b [IEE99c, IEE01], IEEE 802.11g [IEE03]

²⁷CSMA/CA steht für Carrier Sense Multiple Access with Collision Avoidance

²⁸TDMA steht für Time Division Multiple Access

2.1.3.1 Universal Plug and Play (UPnP)

Zur spontanen Vernetzung von IP basierenden Geräten entwickelte Microsoft die Universal Plug and Play (UPnP) Device Architecture [Mic00]. Die Idee dabei ist basierend auf den verbreiteten Internet-Standards²⁹ wie dem Internet Protocol (IP), dem User Datagram Protocol (UDP), dem Transmission Control Protocol (TCP), dem Hypertext Transfer Protocol (HTTP), der Extensible Markup Language (XML) und dem Simple Object Access Protocol (SOAP), geeignete Mechanismen für eine flexible und wartungsfreie Peer-to-Peer Vernetzung von Geräten zu definieren. Dazu wurden einerseits verschiedene Vorschläge bei der Internet Engineering Task Force (IETF) als so genannte Internet Drafts eingereicht und andererseits im Juni 2000 die Version 1.0 der UPnP Device Architecture (UDA) Spezifikation herausgegeben. Die UDA beschreibt die angewendeten Mechanismen anhand von sechs Stufen, die in folgender Auflistung zusammengefasst sind:

- Stufe 0: Addressing
Grundvoraussetzung für eine IP basierende Vernetzung ist die Konfiguration der Geräte mit einer IP-Adresse. Für den automatischen Bezug einer IP-Adresse wird das Dynamic Host Configuration Protocol (DHCP) und, falls kein DHCP-Server im Netzwerk vorhanden ist, ein auf dem IETF Draft Auto-IP („Dynamic Configuration of IPv4 link-local addresses“) basierendes Verfahren eingesetzt.
- Stufe 1: Discovery
Nachdem ein Gerät mit einer gültigen IP-Adresse konfiguriert ist (Stufe 0), bildet Discovery die Stufe 1 der UPnP-Vernetzung. Hierbei wird die Bekanntmachung der Geräte und die Suche nach Geräten durch das Senden von speziellen HTTP-Nachrichten über Multicast-UDP und die Beantwortung von Suchanfragen über Unicast-UDP geregelt. Discovery beruht auf den IETF Drafts HTTPU/MU („Multicast and Unicast UDP HTTP Messages“) und SSDP („Simple Service Discovery Protocol“).
- Stufe 2: Description
Nach der Entdeckung von geeigneten Geräten können in der Stufe 2 der UPnP-Vernetzung genauere Eigenschaften der Geräte über eine aus dem Discovery-Prozess bekannten URL ausgelesen werden. Die Beschreibung der Geräteeigenschaften und Dienste erfolgt in Form von XML-Dokumenten [W3C00a] (Device Description und Service Description), deren Aufbau gemäß XML-Schema [W3C01a, W3C01b] in der UDA-Spezifikation festgelegt ist.
- Stufe 3: Control
Die Stufe 3 der UPnP-Vernetzung behandelt die Ansteuerung der Geräte. Zum Aufruf von Aktionen der Dienste eines Gerätes wird das Simple Object Access

²⁹Der Terminus Internet-Standards bezeichnet hierbei durch die Internet Engineering Task Force (IETF) und das World Wide Web Consortium (W3C) herausgegebene und oft als Request for Comments (RFC) oder Technical Reports bezeichnete technische Dokumente.

Protocol (SOAP) [W3C00b] eingesetzt. Die dabei notwendige „control URL“ des anzusprechenden Dienstes ist in der Device Description (Stufe 2) des zugehörigen Gerätes enthalten.

- Stufe 4: Eventing
Eventing bildet die Stufe 4 der UPnP-Vernetzung und behandelt das Versenden von Benachrichtigungen bei Zustandsänderungen der Geräte und die Registrierung für solche Benachrichtigungen beim Gerät. Dazu werden entsprechend dem IETF Draft GENA („General Event Notification Architecture Base: Client to Arbiter“) HTTP Nachrichten mit erweiterten Methoden über TCP/IP gesendet. Die für die Registrierung notwendige „event subscription URL“ des entsprechenden Dienstes ist in der Device Description (Stufe 2) des zugehörigen Gerätes enthalten.
- Stufe 5: Presentation
Stufe 5 der UPnP-Vernetzung definiert eine HTML-basierende Benutzerschnittstelle zur Ansteuerung und Wiedergabe des Zustands eines Gerätes. Die zum Herunterladen der HTML-Seite notwendige „presentation URL“ ist in der Device Description (Stufe 2) des zugehörigen Gerätes enthalten.

Zur Verbreitung und Weiterentwicklung von UPnP sowie der Definition von Geräteschnittstellen wurde das UPnP Forum gegründet, dem mit Ende Mai 2003 bereits mehr als 520 Unternehmen angehören³⁰. Im Mai 2003 wurde vom UPnP Forum die aktualisierte Version 1.0.1 der UPnP Device Architecture Spezifikation [Con03] herausgegeben. Neben einigen Korrekturen beseitigt die neue Version die ungeeigneten³¹ Referenzen auf IETF Drafts, indem die darin vorgeschlagenen und von der UDA verwendeten Mechanismen in die UDA-Spezifikation übernommen wurden und nunmehr darin abschließend definiert sind.

Abbildung 2.9 zeigt die UPnP-Gerätearchitektur. Die einzelnen Blöcke sind derart angeordnet, dass ein Block auf der Funktionalität des jeweils darunterliegenden Blocks aufbaut. Beispielsweise beruht Control auf SOAP, dieses verwendet HTTP über TCP über IP. Unterhalb der dicken Trennlinie sind die eingesetzten Internet Protokolle inklusive deren Erweiterungen durch die UDA-Spezifikation dargestellt. Der obere Teil besteht aus den eigentlichen drei UPnP-Schichten: *UPnP Device Architecture*, *UPnP Forum* und *UPnP Vendor*. Die erste Schicht repräsentiert die gemeinsame Basis aller UPnP-Geräte, d. h. die grundlegenden Mechanismen und Eigenschaften, die für alle UPnP-Geräte gelten (Discovery, Description, Presentation, ...). Im Gegensatz dazu werden auf der darüber liegenden Schicht (UPnP Forum) nur mehr Vorgaben für bestimmte Gerätetypen und deren Dienste über so genannte *device templates* und *service templates* des UPnP Forums festgelegt. Die oberste Schicht (UPnP Vendor) steht für alle herstellereigenspezifischen Eigenschaften eines konkreten UPnP-Gerätes, wobei die Gestaltungsfreiheit jedoch durch die Vorgaben der darunter liegenden Schichten begrenzt ist.

³⁰Für die Mitgliedschaft am UPnP Forum sind keine Mitgliedsgebühren zu zahlen. Dies mag einer der Gründe für die große Mitgliederanzahl sein. UPnP Forum Homepage: www.upnp.org

³¹IETF Internet Drafts sind nur für eine Dauer von 6 Monaten gültig und unterliegen kurzfristigen Änderungen [Bra96]. Deshalb kann auf sie nicht dauerhaft referenziert werden.

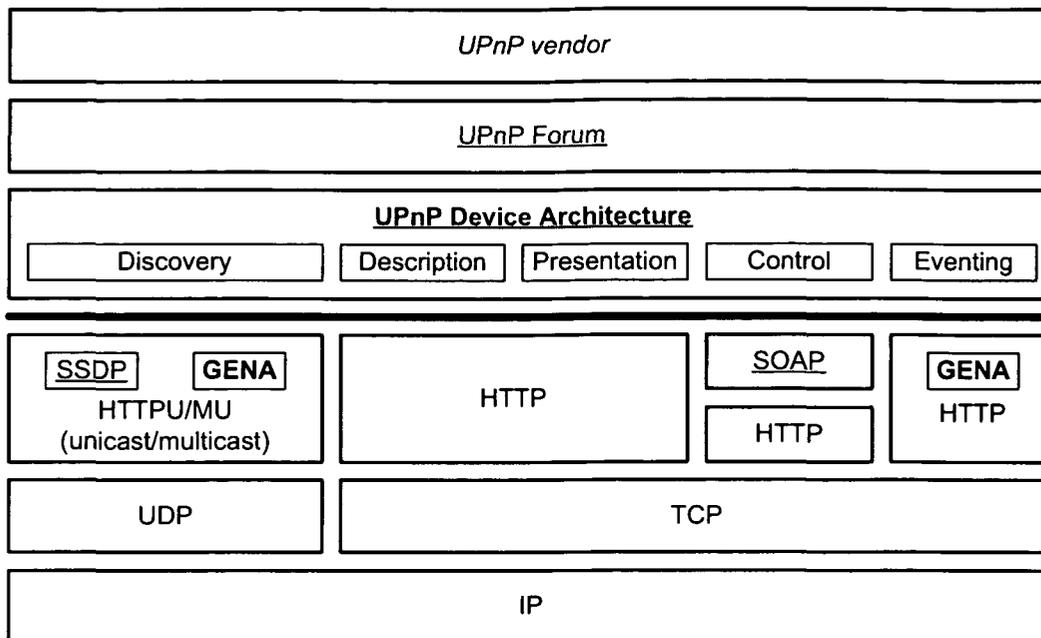


Abbildung 2.9: UPnP-Architektur

Der Stil (normal, kursiv, fett, unterstrichen) und die Farbe der Bezeichner (Draft, Standard, Forum, ...) in Abbildung 2.9 wird in den folgenden Abbildungen zur Beschreibung von UPnP-HTTP-Nachrichten bzw. XML-Dokumenten zur Kennzeichnung der Herkunft der entsprechenden Elemente (HTTP-Methode, HTTP-Header-Eintrag, XML-Element, ...) beibehalten.

Die UPnP Device Architecture Specification (UDA-Spezifikation) definiert zwei Arten von Geräten: *UPnP Devices*, die Dienste bereitstellen, und *UPnP Control Points*, die Dienste nutzen. Den Aufbau eines UPnP-Gerätes zeigt Abbildung 2.10. Ein physikalisches Gerät wird in der UDA durch ein oder mehrere UPnP-Geräte (*root devices*) abstrahiert. Ein „root device“ besteht aus 0 oder mehreren Diensten und 0 oder mehreren eingebetteten Geräten. Geräte besitzen einen eindeutigen Gerätenamen (*Unique Device Name, UDN*) und sind einem Gerätetyp zugeordnet. Dienste stellen Funktionen (*Actions*) bereit und enthalten Zustandsvariablen (*State Variables*). Sie entsprechen einem bestimmten Dienstyp und besitzen eine innerhalb des zugehörigen Gerätes eindeutige Nummer (*serviceId*). Als Gerätename wird ein dauerhaft eindeutiger 128 Bit großer Wert (Universally Unique Identifier, UUID) verwendet, der vom Gerätehersteller vergeben wird. Die Kennzeichnung des Geräte- und Dienstyps erfolgt über einen URN³² und wird für standardisierte Typen vom UPnP Forum und für nicht standardisierte Typen vom Gerätehersteller vergeben.

³²Ein Uniform Resource Name (URN) dient zur dauerhaft eindeutigen und ortsunabhängigen Bezeichnung von Ressourcen [Moa97].

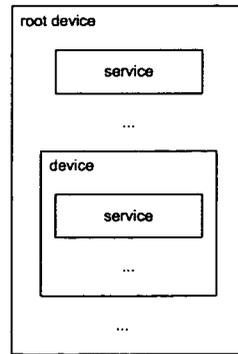


Abbildung 2.10: UPnP-Geräteaufbau

Addressing: Die Konfiguration der Geräte mit einer IP-Adresse erfolgt primär über DHCP (Dynamic Host Configuration Protocol [Dro97]). Ist kein DHCP-Server erreichbar, dann wird die automatische Konfiguration mittels Auto-IP durchgeführt. Hierbei wird eine Adresse aus dem Bereich 169.254.1.0 bis 169.254.254.255 über einen Zufallsalgorithmus ausgewählt. Der Zufallsalgorithmus soll die Wahrscheinlichkeit, dass zwei Geräte die selbe Adresse auswählen, minimieren. Die Überprüfung der IP-Adresse erfolgt über das Senden eines ARP-Requests (Address Resolution Protocol [Plu82]) mit der MAC-Adresse des Gerätes als Source-HW-Adresse, der Source-IP-Adresse 0.0.0.0, der Destination-HW-Adresse 00-00-00-00-00-00 und der zu überprüfenden IP-Adresse als Destination-IP-Adresse. Dieser spezielle ARP-Request wird auch als ARP-Probe bezeichnet. Wird auch nach mehrmaligem Senden der Anfrage keine Antwort empfangen, dann gilt die Adresse als frei und kann vom Gerät verwendet werden. Anderenfalls wird der Auswahlprozess erneut gestartet. Zur Vermeidung von Adresskonflikten muss das Gerät zu jeder Zeit auf ARP-Pakete, welche die eigene IP-Adresse als Source-IP-Adresse aber eine fremde Source-HW-Adresse enthalten, mit einem ARP-Request antworten, der als Source-HW-Adresse die MAC-Adresse des Gerätes, als Destination-HW-Adresse 00-00-00-00-00-00 und die eigene IP-Adresse sowohl als Source- als auch als Destination-IP-Adresse enthält. Dadurch wird die IP-Adresse verteidigt und ein anderes Gerät davon abgehalten, diese IP-Adresse auszuwählen. Bei Verwendung einer Auto-IP-Adresse muss jedoch weiterhin (z. B. alle 5 Minuten) überprüft werden, ob ein DHCP-Server verfügbar ist und gegebenenfalls die Konfiguration über DHCP durchgeführt werden. Unterstützt das Gerät mehrere gleichzeitige IP-Adressen, kann die Auto-IP-Adresse beibehalten werden, anderenfalls muss sie nach Konfiguration über DHCP freigegeben werden.

Discovery: Die Bekanntmachung von und die Suche nach Geräten und Diensten erfolgt über das Senden von Multicast-UDP-Nachrichten. UPnP besitzt im Gegensatz zu anderen Vernetzungskonzepten keine zentrale oder verteilte Dienstdatenbank (vgl. Lookup-Dienst bei Jini oder Registry bei HAVi). Die Bekanntmachungen (*ssdp:alive*) erfolgen erstmalig nach dem Start eines Gerätes und enthalten eine Gültigkeitsdauer von mindestens 1800 Sekunden. Vor Ablauf der Gültigkeitsdauer muss jeweils eine

erneute Bekanntmachung gesendet werden. Die Bekanntmachungen erlöschen damit automatisch mit Ablauf der Gültigkeitsdauer. Dieses Verfahren gewährleistet, dass ein Control Point die abrupte Entfernung bzw. den Ausfall eines Gerätes spätestens mit Ablauf der Gültigkeitsdauer der Bekanntmachung erkennt. Wenn ein Gerät abgeschaltet oder vom Netzwerk entfernt wird, sollte es jedoch vorher - soweit dies möglich ist - die gesendeten „ssdp:alive“-Bekanntmachungen durch das Senden von so genannten „ssdp:byebye“-Bekanntmachungen zurücknehmen. Dies ermöglicht einem Control Point noch vor Ablauf der Gültigkeitsdauer der Bekanntmachungen, die nicht mehr verfügbaren Geräte zu erkennen.

Zum Versenden von Nachrichten verwendet UPnP das Hypertext Transfer Protocol (HTTP [FGM⁺99]). Eine HTTP-Nachricht besteht im Wesentlichen aus einer *Start-Line*, 0 oder mehreren *Message-Header* Zeilen, einer leeren Zeile und optional aus einem *Message-Body*. Die Start-Line ist entweder eine *Request-Line* oder eine *Status-Line*. Erstere enthält die auszuführende HTTP-Methode, eine *Request-URI*³³ zur Bezeichnung der Ressource, an welche dieser HTTP-Request gerichtet ist, und die HTTP-Versionsnummer der Nachricht. Die Status-Line enthält die HTTP-Versionsnummer der Nachricht, einen numerischen *Status-Code* und die zugehörige textuelle *Reason-Phrase* zur Beschreibung des Ergebnisses des bearbeiteten HTTP-Requests.

```
00: NOTIFY * HTTP/1.1
01: HOST: 239.255.255.250:1900
02: CACHE-CONTROL: max-age = seconds until advertisement expires
03: LOCATION: URL for UPnP device description
04: NT: search target
05: NTS: ssdp:alive
06: SERVER: OS/version UPnP/1.0 product/version
07: USN: advertisement UUID
08:
```

Legende: **GENA** **SSDP** **UPnP Device Architecture** **Geraetehersteller**

Abbildung 2.11: UPnP Discovery: Alive-Nachricht

Abbildung 2.11 zeigt eine „ssdp:alive“-Bekanntmachung. Die Legende beschreibt die Herkunft der verschiedenen Elemente. So enthält die Request-Line (Zeile 00) die für Benachrichtigungen definierte neue HTTP-Methode NOTIFY. Dazu gehören die beiden HTTP-Header-Fields NT (Notification Type) und NTS (Notification Sub Type). NT (Zeile 04) bezeichnet den Typ des bekanntzumachenden Gerätes oder Dienstes („root device“, Geräte-Name, Gerätetyp oder Dienstyp), nach dem auch gesucht werden kann. NTS (Zeile 05) enthält den Typ der Bekanntmachung (ssdp:alive). Die Benachrichtigung erfolgt über die Multicast-Adresse 239.255.255.250 und den Port 1900³⁴ (Zeile 01). Zur eindeuti-

³³Ein Uniform Resource Identifier (URI) besteht aus einer Anzahl an Zeichen zur Identifizierung von abstrakten oder physischen Ressourcen. Ein Uniform Resource Locator (URL) und ein Uniform Resource Name (URN) sind beides URIs [BLFM98].

³⁴Die Multicastadresse und der Port sind bei der IANA (Internet Assigned Numbers Authority) für SSDP (Simple Service Discovery Protocol) reserviert.

gen Identifizierung des bekanntzumachenden Objekts (Gerät oder Dienst) dient der USN (Unique Service Name, Zeile 07). Die Gültigkeitsdauer der Bekanntmachung steht in Zeile 02 und die URL der Gerätebeschreibung steht in Zeile 03. Auf Bekanntmachungen wird nicht geantwortet. Für ein Gerät werden mehrere verschiedene Bekanntmachungen gesendet:

- 1 x für das „root device“ mit NT gleich `upnp:rootdevice`
- 1 x für jedes Gerät mit dem Gerätenamen als NT
- 1 x für jeden Gerätetyp mit dem Gerätetyp als NT
- 1 x für jeden unterschiedlichen Diensttyp mit dem Diensttyp als NT

Die Suche erfolgt ebenfalls über das Senden von Multicast-HTTP-UDP-Nachrichten. Abbildung 2.12 zeigt das Format einer Suchanfrage und deren Antwort. Die Request-Line (Suchanfrage: Zeile 00) enthält die zur Suche definierte neue HTTP-Methode `M-SEARCH`. Dazu gehört das HTTP-Header-Field `ST` (Search Target), das zur Angabe des Suchkriteriums dient. Neben dem Suchkriterium `ssdp:all` zum Auffinden aller Geräte und Dienste kann `upnp:rootdevice`, ein Gerätenamen, ein Gerätetyp oder ein Dienstyp als Suchkriterium verwendet werden (vgl. Header-Field `NT` aus den Bekanntmachungen). Das Header-Field `MX` (Maximum Wait, Suchanfrage: Zeile 03) definiert die maximale Anzahl an Sekunden, um welche ein Gerät das Senden der Antwort verzögern soll. Dieser Mechanismus dient zur Vermeidung der Überlastung eines Control Points durch viele gleichzeitige Antworten.

```

00: M-SEARCH * HTTP/1.1                               Suchanfrage
01: HOST: 239.255.255.250:1900
02: MAN: "ssdp:discover"
03: MX: seconds to delay response
04: ST: search target
05:

00: HTTP/1.1 200 OK                                     Antwort
01: CACHE-CONTROL: max-age = seconds until advertisement expires
02: DATE: when response was generated
03: EXT:
04: LOCATION: URL for UPnP device description
05: SERVER: OS/version UPnP/1.0 product/version
06: ST: search target
07: USN: advertisement UUID
08:

```

Legende: SSDP UPnP Device Architecture *Geraetehersteller*

Abbildung 2.12: UPnP Discovery: Suchanfrage und Antwort

Die Antwort auf eine Suchanfrage wird als Unicast-UDP-Nachricht gesendet und wirkt wie eine Bekanntmachung. So enthält sie eine Gültigkeitsdauer (Antwort: Zeile 01), einen Typ (Antwort: Zeile 06), einen USN (enthält Gerätenamen als Präfix, Antwort: Zeile 07) und eine URL für die Gerätebeschreibung (Antwort: Zeile 04).

Description: Die Beschreibung der Geräteeigenschaften und Dienste erfolgt in Form von XML-Dokumenten [W3C00a]. Für jedes „root device“ gibt es ein *Device Description* XML-Dokument und für jeden Dienst ein *Service-Description-XML*-Dokument. Beide können mit einem normalen HTTP-GET-Request über TCP/IP geladen werden. Die URL für die Device Description ist aus dem Discovery-Prozess bekannt und die URL der Service Description ist in der Device Description des zugehörigen Gerätes enthalten.

Die Device Description besteht im Wesentlichen aus den zwei verpflichtenden Elementen `specVersion` zur Bezeichnung der UPnP Version und `device` zur eigentlichen Beschreibung des Gerätes. Das Element `device` enthält die folgenden Informationen:

- Herstellerspezifische Informationen:
Modellname, -nummer, -beschreibung, Seriennummer, ...
- Gerätetyp und Gerätename
- Liste aller enthaltener Dienste:
Enthält für jeden Dienst den Diensttyp, die *serviceId*, die URL für das Service-Description-XML-Dokument, die URL für Control und die URL für Eventing
- Liste aller eingebetteter Geräte:
Enthält für jedes Gerät ein eigenes `device`-Element
- User Interface Informationen:
URL für Icons und HTML-Seite (Presentation)

Im Wesentlichen besteht die Service Description aus den drei Elementen `specVersion`, `actionList` und `serviceStateTable`. Die `actionList` definiert die Namen der aufrufbaren Funktionen und deren Parameter (Name, Richtung, zugeordnete Zustandsvariable). Der Zustand eines Gerätes wird über so genannte *State Variables* (Zustandsvariablen) abstrahiert. Alle Zustandsvariablen eines Dienstes sind in der `serviceStateTable` definiert (Datentyp, vorgegebener Wert, erlaubte Werte, ...). Über die Zuordnung einer Zustandsvariablen zu jedem Funktionsparameter erfolgt die Festlegung des Datentyps und der erlaubten Werte der Funktionsparameter.

Der Aufbau und zwingende Inhalt der XML-Dokumente sowie die Datentypen werden durch eine vom XML-Schema (Part1: Structures [W3C01a], Part2: Datatypes [W3C01b]) abgeleitete *UPnP Template Language* festgelegt. Damit ist auch die automatisierte Überprüfung der Geräte- und Dienstbeschreibungen (XML-Dokumente!) möglich. Die Dokumente werden vom UPnP-Gerätehersteller geschrieben und basieren meist auf so genannten *Device Templates* für standardisierte Gerätetypen bzw. auf *Service Templates* für standardisierte Dienste vom UPnP Forum.

Control: Für das entfernte Aufrufen von Funktion der Dienste eines Gerätes wird das Simple Object Access Protocol (SOAP) [W3C00b] eingesetzt. Ein Funktionsaufruf erfolgt dabei durch das Senden einer SOAP-Nachricht mit einem HTTP-POST-Request über TCP/IP. Abbildung 2.13 zeigt den in eine SOAP-Nachricht umgewandelten Funktionsaufruf.

```

00: POST path to control URL HTTP/1.1
01: HOST: host of control URL:port of control URL
02: CONTENT-LENGTH: bytes in body
03: CONTENT-TYPE: text/xml; charset="utf-8"
04: SOAPACTION: "urn:schemas-upnp-org:service:serviceType:v#actionName "
05:
06: <?xml version="1.0"?>
07: <s:Envelope
08:     xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
09:     s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" >
10:   <s:Body>
11:     <u:actionName xmlns:u="urn:schemas-upnp-org:service:serviceType:v">
12:       <argumentName>in arg value</argumentName>
13:     </u:actionName>
14:   </s:Body>
15: </s:Envelope>

```

Legende: SOAP 1.1 UPnP Device Architecture UPnP Forum Geraetehersteller

Abbildung 2.13: UPnP Control: Funktionsaufruf

Der Name der auszuführenden Aktion steht in dem von SOAP definierten HTTP-Header-Field SOAPACTION (Zeile 04). Die für den Aufruf notwendige „control URL“ des Dienstes (Zeilen 00, 01) ist in der Device Description des zugehörigen Gerätes enthalten. Nach der Leerzeile (Zeile 05) folgt der HTTP-Body, der ein XML-Dokument mit allen Eingangsparametern der auszuführenden Aktion enthält (Zeilen 06 bis 15).

Die Beantwortung des Funktionsaufrufs erfolgt durch das Senden der HTTP-Response über TCP/IP und ist in Abbildung 2.14 dargestellt. Die Rückgabewerte der Funktion sind im HTTP-Body der Nachricht als XML-Dokument enthalten (Zeilen 06 bis 16). Sowohl die Ausführung der Aktion als auch das Senden der Antwort dürfen zusammen nicht länger als 30 Sekunden dauern, anderenfalls soll das Gerät eine vorzeitige Antwort schicken und den Abschluss der Aktion über eine spätere Event-Benachrichtigung (siehe Eventing) signalisieren. Tritt bei der Ausführung der Aktion ein Fehler auf, so enthält die Antwort anstatt des XML-Dokuments mit den Rückgabewerten ein XML-Dokument mit der Beschreibung des aufgetretenen Fehlers (`errorCode`, `errorDescription`).

Zur Ermittlung des aktuellen Zustands eines Gerätes können Control Points die Zustandsvariablen über geeignete Aktionen auslesen oder sich für eine im nächsten Abschnitt beschriebene Ereignisbenachrichtigung registrieren³⁵.

³⁵Die in der Version 1.0 der UDA-Spezifikation definierte allgemeine Aktion `QueryStateVariable` zum Auslesen des Wertes einer einzelnen Zustandsvariablen wurde in der neuen Version 1.0.1 als veraltet gekennzeichnet und soll nun nicht mehr verwendet werden.

```

00: HTTP/1.1 200 OK
01: CONTENT-LENGTH: bytes in body
02: CONTENT-TYPE: text/xml; charset="utf-8"
03: DATE: when response was generated
04: SERVER: OS/version UPnP/1.0 product/version
05:
06: <?xml version="1.0"?>
07: <s:Envelope
08:   xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
09:   s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
10:   <s:Body>
11:     <u:actionNameResponse
12:       xmlns:u="urn:schemas-upnp-org:service:serviceType:v">
13:       <argumentName>out arg value</argumentName>
14:     </u:actionNameResponse>
15:   </s:Body>
16: </s:Envelope>

```

Legende: SOAP 1.1 UPnP Device Architecture UPnP Forum Geraetehersteller

Abbildung 2.14: UPnP Control: Funktionsantwort

Eventing: Zustandsvariablen repräsentieren den aktuellen Zustand eines Gerätes. Ein Event beschreibt die Änderung eines Zustands. Für jede Zustandsvariable ist in der Service Description definiert, ob bei einer Änderung der Zustandsvariablen eine Benachrichtigung gesendet wird. Control Points können den Empfang von Event-Benachrichtigungen über die URL für Eventing (`eventSubURL`) des Dienstes für eine bestimmte Zeitspanne abonnieren und auch wieder abbestellen. Die `eventSubURL` ist in der Dienstbeschreibung der Device Description des zugehörigen Gerätes enthalten. Um die Registrierung aufrecht zu erhalten, muss diese vor Ablauf der Zeitspanne erneuert werden. An- und Erneuerungsmeldungen sowie Abmeldungen werden mit HTTP über TCP/IP gesendet.

Abbildung 2.15 zeigt eine Nachricht zur Registrierung von Event-Benachrichtigungen und die zugehörige Antwort. Die Request-Line enthält die für An- und Erneuerungsmeldungen definierte neue HTTP-Methode `SUBSCRIBE` (Anfrage: Zeile 00). Zu ihr gehören die HTTP-Header-Fields `CALLBACK` (Anfrage: Zeile 02) zur Angabe der URL, an welche die Benachrichtigungen gesendet werden sollen, und `NT` (Anfrage: Zeile 03) mit dem fixen Wert `upnp:event`. Das Feld `TIMEOUT` (Anfrage: Zeile 04) enthält die vom Control-Point gewünschte Dauer des Abonnements. Die Antwort enthält ein weiteres zur HTTP-Methode `SUBSCRIBE` gehörendes Header-Field, den `SID` (Subscription Identifier, Antwort: Zeile 03), welcher zur eindeutigen Identifizierung des erfolgreich eingerichteten Abonnements dient. Die vom Gerät zugewiesene Gültigkeitsdauer des Abonnements steht im Feld `TIMEOUT` (Antwort: Zeile 04).

Erneuerungsmeldungen sind der Nachricht zur Anmeldung ähnlich, anstatt der Header-Fields `CALLBACK` und `NT` wird jedoch die nunmehr bekannte `SID` gesendet. Abmeldungen verwenden die eigens dafür definierte neue HTTP-Methode `UNSUBSCRIBE` und das Header-Field `SID` zur Bezeichnung des abzumeldenden Abonnements.

Event-Benachrichtigungen (siehe Abbildung 2.16) werden ebenfalls mit HTTP über

```

00: SUBSCRIBE eventSubURL path HTTP/1.1
01: HOST: eventSubURL host:eventSubURL port
02: CALLBACK: <delivery URL>
03: NT: upnp:event
04: TIMEOUT: Second-requested subscription duration
05:

00: HTTP/1.1 200 OK
01: DATE: when response was generated
02: SERVER: OS/version UPnP/1.0 product/version
03: SID: uuid:subscription-UUID
04: TIMEOUT: Second-actual subscription duration
05:

```

Anfrage

Antwort

Legende: **GENA** UPnP Device Architecture *Geraetehersteller*

Abbildung 2.15: UPnP Eventing: Subscribe-Nachricht

TCP/IP gesendet und enthalten im HTTP-Body (Zeilen 09-15) ein XML-Dokument mit allen seit der letzten Benachrichtigung geänderten Zustandsvariablen und ihre neuen Werte. Wie auch schon für die Bekanntmachungen beim Discovery-Prozess wird für Benachrichtigungen die neue HTTP-Methode NOTIFY (Zeile 00) verwendet. Die zugehörigen Header-Fields NT, NTS und SID enthalten den fixen Wert `upnp:event` (Zeile 04), den fixen Wert `upnp:propchange` (Zeile 05) bzw. den eindeutigen Kennzeichner des Abonnements. Das von der UDA definierte Header-Field SEQ (Zeile 07) dient als *Event Key* zur Kennzeichnung der Reihenfolge der gesendeten Event-Benachrichtigungen. Dazu enthält es einen 32 Bit großen „unsigned integer“, der mit 0 initialisiert wird und bei einem Überlauf bei 1 anstatt bei 0 fortsetzt.

Gleich nach erfolgreicher Registrierung erfolgt eine initiale Benachrichtigung (mit einem SEQ-Wert gleich 0), die alle Zustandsvariablen enthält und damit dem Empfänger ein vollständiges Abbild des aktuellen Zustands liefert. Benachrichtigungen werden jeweils an alle Abonnenten gesendet, es ist also keine individuelle Registrierung für den Empfang von Benachrichtigungen bei der Änderung einer bestimmten Zustandsvariablen möglich.

Presentation: Die UDA definiert eine HTML-basierende Benutzerschnittstelle zur Ansteuerung und Wiedergabe des Zustands eines Gerätes. Dazu ist in der Device Description das optionale Element `presentationURL` definiert, das eine URL zu einer HTML-Seite (*Presentation-Page*) enthält. Mit dieser URL kann ein Control-Point über ein normales HTTP-GET über TCP/IP die Presentation-Page laden und in einem Browser als Benutzerschnittstelle des Gerätes darstellen. Abgesehen von der Vorschrift der Verwendung von HTML (HyperText Markup Language) Version 3.0 oder höher für die Presentation-Page, macht die UDA-Spezifikation keine weiteren Vorgaben. Wie die An-

```

00: NOTIFY delivery path HTTP/1.1
01: HOST: delivery host:delivery port
02: CONTENT-TYPE: text/xml
03: CONTENT-LENGTH: Bytes in body
04: NT: upnp:event
05: NTS: upnp:propchange
06: SID: uuid:subscription-UUID
07: SEQ: event key
08:
09: <?xml version="1.0"?>
10: <e:propertyset xmlns:e="urn:schemas-upnp-org:event-1-0">
11:   <e:property>
12:     <variableName>new value</variableName>
13:   </e:property>
14:   Other variable names and values (if any) go here.
15: </e:propertyset>

```

Legende: **GENA** UPnP Device Architecture UPnP Forum Geraetehersteller

Abbildung 2.16: UPnP Eventing: Benachrichtigung

steuerung und Wiedergabe des Zustands eines Gerätes erfolgt, ist nicht festgelegt und damit dem Hersteller des Gerätes überlassen.

UPnP-Geräte und -SDKs: Bis Ende Mai 2003 wurden vom UPnP Forum Geräteschnittstellen (Device/Service-Templates) für fünf verschiedene Gerätetypen standardisiert [UPnP03]: Internet Gateway Device (IGD) V1.0, MediaServer/MediaRenderer V1.0, Printer Device V1.0, Scanner V1.0 und HVAC³⁶ V1.0. Die Zertifizierung von UPnP-Geräten erfolgt über die *UPnP Implementers Cooperation (UIC)*³⁷, deren Aufgabe die Vermarktung des Einsatzes der UPnP-Technologie in Geräten ist und die das UPnP-Markenzeichen hält. Eine Mitgliedschaft beim UIC kostet pro Jahr 5000 Dollar und setzt eine Mitgliedschaft beim UPnP Forum voraus³⁸. Für eine Zertifizierung muss der Gerätehersteller Mitglied bei der UIC sein, 2000 Dollar für die Registrierung eines Gerätes bezahlen, die vorgeschriebenen Tests durchführen und es müssen die eingesendeten Testprotokolle vom Test-Reviewer der UIC positiv begutachtet worden sein. Im Gegenzug erlaubt die UIC die Verwendung des UPnP-Logos auf zertifizierten Geräten.

Für UPnP sind mehrere Software Development Kits (SDK) verfügbar, einige davon sind für interne Evaluierungszwecke kostenlos (z. B. Intel UPnP SDK [SH03] oder Siemens UPnP SDK [Wis01]).

³⁶HVAC steht für Heating, Ventilation, Air Conditioning

³⁷UIC Homepage: www.upnp-ic.org

³⁸Mit Ende Mai 2003 hatte die UIC 31 Mitglieder.

2.1.3.2 Jini

Im Jänner 1999 wurde von der Firma Sun Microsystems die erste Version von Jini³⁹ zur dynamischen Vernetzung von Geräten vorgestellt [Edw00]. Die wohl kürzeste und prägnanteste Beschreibung von Jini läßt sich der aktuellen Jini Architecture Specification (Version 2.0Beta) [Sun03b] entnehmen: „The Jini system extends the Java application environment from a single virtual machine to a network of machines.“

Jini ist Java [GJSB00] basierend und im Gegensatz zu anderen Vernetzungsstandards, wie beispielsweise UPnP (siehe Abschnitt 2.1.3.1), werden nicht nur Daten, sondern auch Programmcode übertragen. So werden bei Jini die Dienste der Geräte des Netzwerks über so genannte Dienst-Proxys zur Verfügung gestellt. Ein *Dienst-Proxy* ist der lokale Stellvertreter für einen (entfernten) Dienst. Da der Dienst-Proxy die selbe Schnittstelle wie sein Dienst bereitstellt und in der lokalen Java Virtual Machine (JVM) des Dienstnutzers ausgeführt wird, kann der Dienst wie jedes andere lokale Objekt genutzt werden. Soweit der Proxy die geforderte Funktionalität nicht bereits selbst bietet, leitet er die Anfrage an den entfernten Dienst weiter.

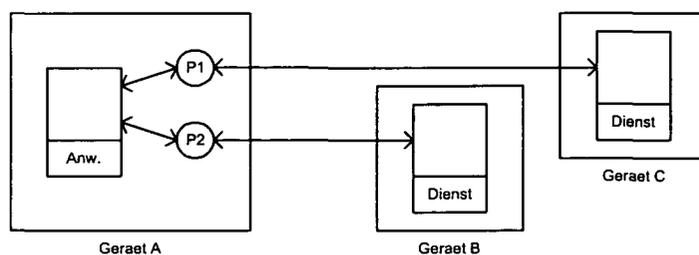


Abbildung 2.17: Dienst und Dienst-Proxy

In Abbildung 2.17 ist dieses Konzept nochmals dargestellt. Die Anwendung (Anw.) auf dem Gerät A nutzt über den lokalen Dienst-Proxy P2 den Dienst des Gerätes B und über P1 den Dienst von Gerät C. Die Kommunikation zwischen dem Dienst-Proxy und dem Dienst ist dabei für die Anwendung nicht sichtbar und wurde deshalb von Jini nicht festgelegt. Der Dienst-Proxy muss dabei vom Anbieter des Dienstes zur Verfügung gestellt werden. Zum Auffinden von Diensten und der Bereitstellung der Dienst-Proxys werden ein oder mehrere *Lookup-Dienste* eingesetzt. Der Lookup-Dienst selbst wird über ein mit *Discovery* bezeichnetes Verfahren gefunden. Der Mechanismus zur Teilnahme von Diensten am Jini Netzwerk, welcher unter anderem die Bereitstellung des Dienst-Proxy und von Attributen zur Beschreibung des Dienstes beim Lookup-Dienst beinhaltet, wird mit *Join* bezeichnet. Schließlich erfolgt die Suche nach den Diensten und das Herunterladen der Dienst-Proxys nach einem mit *Lookup* bezeichneten Verfahren. Zur Unterteilung des Jini-Netzwerks werden Gruppen verwendet. Eine Gruppe wird durch einen Namen bezeichnet und besteht aus einem oder mehreren Lookup-Diensten und allen dort für diese Gruppe registrierten Diensten. Eine Gruppe wird auch als *Djinn* bezeichnet.

³⁹Ogleich Jini eigentlich kein Akronym ist, wird Jini vielfach als Kurzform für „Java Intelligent Network Infrastructure“ bezeichnet.

Discovery: Jini definiert zum Auffinden von Lookup-Diensten drei verschiedene Discovery-Protokolle, die nachfolgend beschrieben und in Abbildung 2.18 dargestellt werden.

- **Multicast-Request-Protokoll**
Dieses Protokoll wird von Anwendungen oder Diensten zum Auffinden von Lookup-Server bestimmter Gruppen verwendet. Die Anfrage erfolgt dabei über ein UDP-Multicast-Paket an die Multicast-IP-Adresse 224.0.1.85 und den Ziel-Port 4160. Die Antwort erfolgt über ein TCP-Unicast-Paket an die in der Anfrage angegebene Adresse und enthält den Dienst-Proxy vom antwortenden Lookup-Dienst.
- **Multicast-Announcement-Protokoll**
Mit diesem Protokoll machen sich Lookup-Dienste in regelmäßigen Abständen⁴⁰ bei allen anderen Diensten und Anwendungen bekannt. Damit können bereits laufende Anwendungen und Dienste von neuen oder wiederhergestellten Lookup-Diensten benachrichtigt werden. Die Benachrichtigung erfolgt dabei über ein UDP-Multicast-Paket an die Multicast-IP-Adresse 224.0.1.84 und den Ziel-Port 4160 und enthält bloß eine eindeutige Bezeichnung des Lookup-Dienstes. Um den Dienst-Proxy zu erhalten, muss der Empfänger das Unicast-Discovery-Protokoll verwenden.
- **Unicast-Discovery-Protokoll**
Über dieses Protokoll kann eine Anwendung oder ein Dienst mit einem bereits bekannten Lookup-Dienst direkt kommunizieren. Die eindeutige Bezeichnung erfolgt dabei über den Host-Namen und die Port-Nummer oder als URL der Form `jini://host:port`. Die Anfrage wird dabei über ein TCP-Unicast-Paket an den bekannten Host des Lookup-Dienstes gesendet. Die Antwort erfolgt über ein TCP-Unicast-Paket an das anfragende Gerät und den Ziel-Port 4160 und enthält den Dienst-Proxy vom gewünschten Lookup-Dienst.

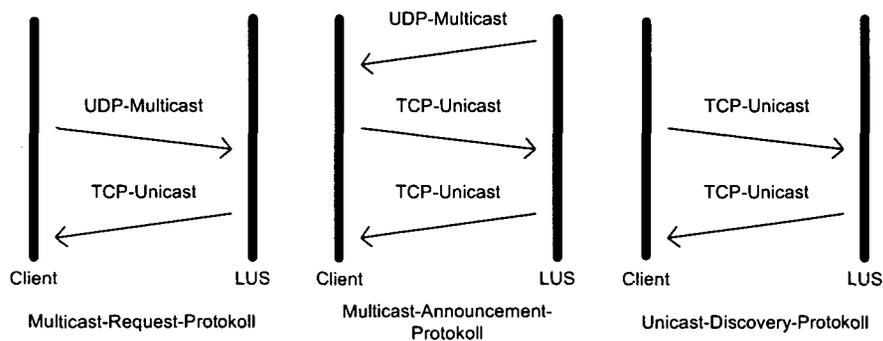


Abbildung 2.18: Discovery-Protokolle

⁴⁰Die Jini Technology Core Platform Specification empfiehlt ein 120s Intervall

Join: Damit ein Dienst gefunden und dessen Dienst-Proxy heruntergeladen werden kann, muss eine Registrierung des Dienstes bei einem Lookup-Dienst der gewünschten Gruppe erfolgen. Zur Registrierung wird ein *ServiceItem* übergeben, der aus den folgenden Elementen besteht: (1) einer *ServiceID*⁴¹, (2) dem Dienst-Proxy und (3) den Attributen des Dienstes. Ein Dienst kann an mehreren Gruppen teilnehmen und zur Fehlertoleranz auch bei mehreren Lookup-Diensten einer Gruppe registriert werden.

Lookup: Der Lookup-Dienst ermöglicht die Suche nach Diensten anhand der ServiceID, der implementierten Schnittstellen und der Attribute des Dienstes. Die Suche erfolgt durch Übergabe einer Vergleichsmaske (*ServiceTemplate*) mit den Eigenschaften des gesuchten Dienstes, wobei Felder mit dem Wert null beim Vergleich ignoriert werden. Entsprechend der verwendeten Suchfunktion wird das ServiceItem eines oder mehrerer passender Dienste zurück geliefert. Da ein Dienst auch bei mehreren Lookup-Diensten der selben Gruppe registriert werden kann, liefert jeder dieser Lookup-Dienste bei einer entsprechenden Suche den ServiceItem vom selben Dienst. Bei einer Anwendung ist deshalb zur Vermeidung der Mehrfachzählung eines Dienstes der Abgleich mit bereits gefundenen Diensten anhand der ServiceID notwendig.

Die Funktionsweise von Join und Lookup ist in Abbildung 2.19 nochmals graphisch dargestellt. Hierbei hat die Anwendung auf Gerät A und der Dienst auf Gerät B bereits über eines der Discovery-Protokolle den Lookup-Dienst-Proxy P1 erhalten. Über P1 registriert der Dienst seinen Proxy P2 beim Lookup-Dienst (Pfeil 1). Den Proxy P2 erhält die Anwendung als Antwort auf eine Suchanfrage nach dem neuen Dienst über den Lookup-Dienst-Proxy P1 auf dem Gerät A (Pfeil 2). Über den Proxy P2 kann die Anwendung den von Gerät B angebotenen Dienst nutzen (Pfeil 3).

Remote Method Invocation: Zur Kommunikation zwischen Lookup-Dienst-Proxy und Lookup-Dienst wird Remote Method Invocation (RMI) [Sun02] verwendet. Jedes Objekt, das Methoden enthält, die über entfernte Anwendungen⁴² aufgerufen werden können, wird als *Remote Object* bezeichnet. Die Kommunikation der entfernten Anwendung mit dem Remote Object erfolgt über so genannte *Stubs* und *Skeletons*. Ein Stub bietet die gleiche Schnittstelle wie das Remote Object und wird auf der entfernten Seite eingesetzt, um bei einem Methodenaufruf (1) den Aufbau einer Verbindung zum Remote Object, (2) die Übertragung der Methodenparameter in serialisierter⁴³ Form, (3) den Empfang des Rückgabewertes oder Ausnahmewertes und (4) die Rückgabe des Wertes oder der

⁴¹Die ServiceID ist ein 128 Bit Wert, der zur (ausreichend) eindeutigen Identifizierung des Dienstes verwendet wird. Bei der erstmaligen Registrierung des Dienstes wird, bei Übergabe von Null, diese vom Lookup-Dienst erzeugt.

⁴²Als entfernte Anwendung wird hier eine Anwendung bezeichnet, die auf einer anderen JVM ausgeführt wird.

⁴³Die Java Object Serialization [Sun98] definiert die Konvertierung eines Objekts in einen und aus einem genormten Datenstrom, der zur Speicherung oder Netzwerkübertragung verwendet werden kann.

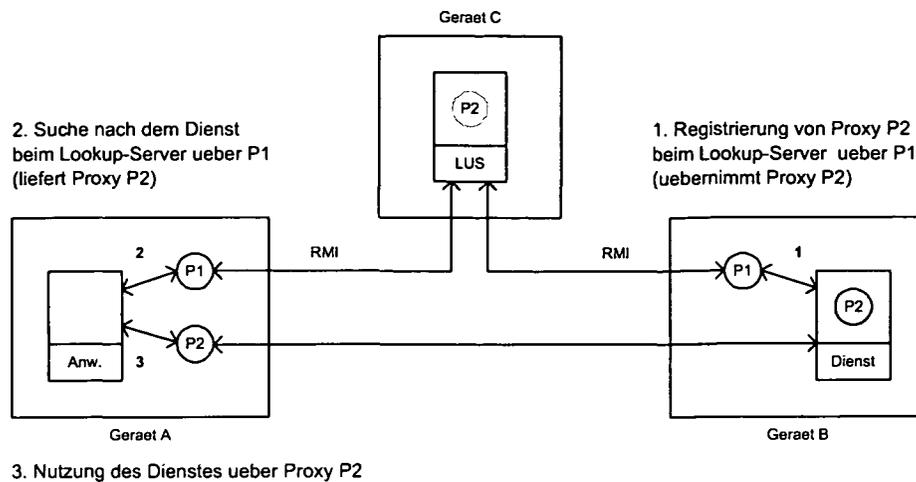


Abbildung 2.19: Join und Lookup

Ausnahme an den Aufrufer durchzuführen. Das Skeleton wird auf der Seite des Remote Objects verwendet, um (1) entfernte Anfragen zu empfangen, (2) die Methodenparameter wiederherzustellen (Deserialisierung), (3) die gewünschte Methode aufzurufen und (4) den Rückgabewert oder den Ausnahmewert in serialisierter Form an den Aufrufer zurück zu senden. Sowohl der Stub als auch das Skeleton⁴⁴ kann direkt aus der Remote-Object-Klasse über den Java RMI-Stub-Compiler *rmic* erzeugt werden. Da bei der Java Objekt-Serialisierung [Sun98] die serialisierte Form eines Objekts keinen Programmcode enthält, muss dieser entweder bereits beim Empfänger vorhanden sein oder von der von RMI beim Serialisieren hinzugefügten „codebase URL“ geladen werden. Meist wird dazu der Programmcode über einen HTTP-Server verfügbar gemacht.

Lease: Jini definiert einen als *Leasing* bezeichneten Mechanismus, der die automatische Freigabe von nicht mehr benötigten Ressourcen ermöglicht. Eine *Lease* ist eine zeitlich befristete Erlaubnis zur Inanspruchnahme einer Ressource eines Dienstes. Sie wird unter Angabe einer gewünschten Lease-Dauer beim Lease-Geber beantragt. Der Lease-Geber entscheidet, ob und für welche Dauer er sie gewährt⁴⁵. Benötigt ein Lease-Nehmer eine Ressource länger als für die gewährte Lease-Dauer, so muss er vor Ablauf der Lease versuchen diese beim Lease-Geber zu verlängern. Läuft jedoch eine Lease aus, so kann die zugehörige Ressource freigegeben werden. Dies ist eine notwendige Voraussetzung für langlebige Dienste, weil die Freigabe von nicht mehr benötigten Ressourcen durch den Nutzer nicht vorausgesetzt werden kann. So könnte z. B. ein Nutzer vor Ausführung einer geeigneten Abmeldung vom Netzwerk getrennt werden. Ein weiterer Aspekt des

⁴⁴Ab Java Version 1.2 ist es möglich einen Stub zu erzeugen, der mit einem allgemeinem Skeleton kommuniziert. In diesem Fall wird dann kein spezielles Skeleton zu jedem Remote Object mehr benötigt.

⁴⁵Die gewährte Lease-Dauer darf hierbei aber nicht größer als die gewünschte Lease-Dauer sein.

Leasings ist die Betrachtung der Anfragen zur Verlängerung einer Lease als „keep alive“-Nachrichten. Ein Beispiel dafür ist der Lookup-Dienst, der für die Speicherung der Einträge der Dienste Leases vergibt. Die gewünschte Lease-Dauer wird dazu bei der Registrierung eines Dienstes übergeben. Damit ein Dienst in der Jini-Gruppe sichtbar bleibt, d. h. der Eintrag im Lookup-Dienst nicht gelöscht wird, muss er die Lease vor Ablauf verlängern. Da ein Dienst, abgesehen von der Änderung seiner Gruppenmitgliedschaft, dauerhaft an der Teilnahme der Jini-Gruppe interessiert ist, wird er immer versuchen die Lease zu verlängern. Jede empfangende Verlängerungsanfrage kann deshalb als „keep alive“-Nachricht betrachtet werden. Nicht mehr verfügbare Dienste, d. h. Dienste, für welche keine Verlängerungsanfrage innerhalb der Lease-Dauer empfangen wurde („keep alive“-Nachricht ausständig), werden automatisch⁴⁶ vom Lookup-Dienst gelöscht und damit von der Jini-Gruppe entfernt. Leasing sorgt also für eine Art „Selbstheilung“ des Netzwerks, indem nicht mehr benötigte Ressourcen automatisch freigegeben und nicht mehr verfügbare Dienste von den Jini Gruppen entfernt werden.

Remote-Events: Eine Zustandsänderung eines Objekts wird als *Ereignis* bezeichnet. Damit ein Objekt auf ein Ereignis eines (entfernten) Objektes reagieren kann, wird von Jini ein auf RMI basierendes Ereignis-Benachrichtigungsmodell definiert. Jenes Objekt, das die Ereignis-Benachrichtigung an alle registrierten Empfänger sendet, wird als *Event Generator* bezeichnet. Das an dem Ereignis interessierte Objekt wird *Remote Event Listener* genannt. Hauptaufgabe des Remote Event Listeners ist der Empfang von Benachrichtigungen des Event Generators. Hierzu verfügt jeder Listener über die Methode *notify*, welcher als Parameter das *Remote Event*-Objekt übergeben wird. Das Remote-Event-Objekt enthält (1) die *eventID*⁴⁷, welche das aufgetretene Ereignis bezeichnet; (2) die Referenz auf das Objekt, in welchem das Ereignis aufgetreten ist; (3) die Folgenummer⁴⁸ des Ereignisses, die zur Ermittlung der Reihenfolge der Ereignisse dient und (4) das *handback*-Objekt, das beim Registrieren übergeben wurde und dem Listener als Kontext dient. Die Registrierung für den Empfang von Remote Events ist bis auf wenige Anforderungen, wie die Verwendung von Leasing für die Registrierung, von Jini nicht festgelegt. Die Registrierung eines Remote Event Listeners kann auch von einem dritten Objekt durchgeführt werden. Dadurch können zur Erweiterung der Funktionalität so genannte *Agents* zwischen dem Event Generator und dem Remote Event Listeners eingefügt werden. Beispielsweise könnte ein Store-and-forward-Agent als einziger Remote Event Listener des Event Generators registriert sein, der die empfangenen Benachrichtigungen an alle nunmehr bei ihm registrierten Remote Event Listener gesichert überträgt.

⁴⁶Die maximale Zeit zum Erkennen nicht mehr verfügbarer Dienste wird durch die vom Lookup-Dienst gewährte Lease-Dauer bestimmt.

⁴⁷Der Ereignisbezeichner ist nur innerhalb eines Objekts eindeutig.

⁴⁸Die Folgenummer muss aufsteigend sein und darf sich für zwei Remote Objects mit der gleichen *eventID* und Objektreferenz nur dann unterscheiden, wenn das Ereignis aus unterschiedlichen Vorkommnissen resultiert.

Transaktionen: Das von Jini definierte Transaktionsmodell basiert auf dem „*Two-Phase-Commit*“-Protokoll. Eine Reihe von Operationen eines Dienstes oder mehrerer Dienste können gemeinsam eine Transaktion bilden. Eine Transaktion ist erfolgreich, wenn alle Operationen erfolgreich durchgeführt werden konnten. Schlägt auch nur eine Operation fehl, so wird der Zustand vor Durchführung aller Operationen wiederhergestellt. Von Diensten, die Transaktionen unterstützen, wird gefordert, dass sie die Schnittstelle für die Teilnahme an den Transaktionen („*TransactionParticipant*“) implementieren und die ACID⁴⁹ Eigenschaften einhalten. Der Transaktionsmanager koordiniert die Dienste, indem er ihnen signalisiert, was sie im Rahmen des „*Two-Phase-Commit*“-Protokolls tun sollen. Vereinfacht kann der Ablauf wie in Abbildung 2.20 dargestellt werden: (1) Eine Anwendung erzeugt einen Transaktionsmanager („*TransactionManager*“); (2) die Anwendung ruft unter Angabe des erzeugten Transaktionsmanagers die einzelnen Operationen der Dienste auf; (3) die Dienste registrieren sich bei diesem Transaktionsmanager; (4) die Anwendung ruft die Methode `commit` des Transaktionsmanagers auf, um die Transaktion durchzuführen; (5) der Transaktionsmanager ruft die Methode `prepare` aller für diese Transaktion registrierten Dienste auf und die Dienste versuchen die Operation durchzuführen; (6a) sind alle Dienste bereit zum „*commit*“, d. h. alle Operationen konnten erfolgreich durchgeführt werden, dann ruft der Transaktionsmanager die Methode `commit` aller Dienste auf und diese schließen die Operation ab; (6b) ist auch nur eine einzige Operation fehlgeschlagen, ruft der Transaktionsmanager die Methode `abort` aller Dienste auf und alle Dienste müssen den Ausgangszustand wiederherstellen.

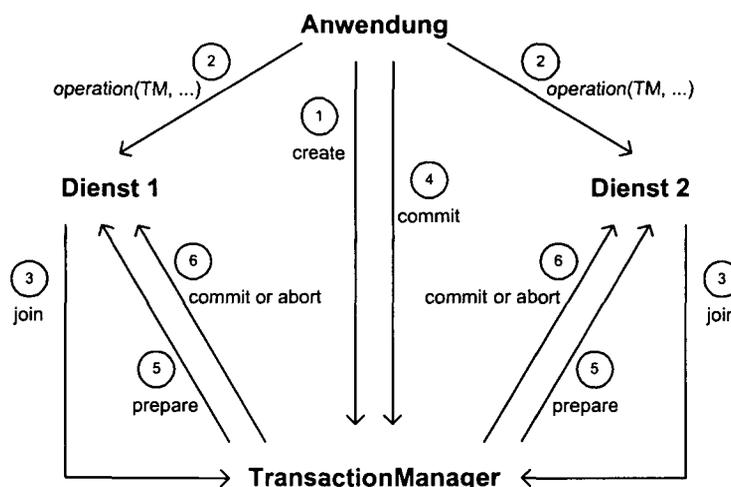


Abbildung 2.20: Transaktionen

⁴⁹Atomicity, Consistency, Isolation, Durability

Gerätearchitekturen [Sun03c]: Für die Bereitstellung und Nutzung von Diensten über Jini benötigt ein Gerät eine JVM, Jini-Klassen, einen Dienst-Proxy und ein Dienstprogramm zur Ansteuerung der Hardware des Dienstes. Besitzt ein Gerät eine eigene JVM, dann erfolgt die Bereitstellung und Nutzung des Dienstes, wie in Abbildung 2.19 gezeigt wurde. Die Ansteuerung der Hardware des Dienstes erfolgt dabei durch das in der JVM des Gerätes ausgeführte Dienstprogramm. Die Kommunikation zwischen der Hardware und dem Dienstprogramm ist privat. Die notwendigen Geräteresourcen können durch Verwendung einer angepassten Version der JVM, z. B. ohne UI-Klassen, gesenkt werden. Eine weitere Möglichkeit zur Senkung der Gerätekosten besteht in der gemeinsamen Nutzung einer JVM durch mehrere Geräte. Dabei erlaubt ein Gerät mit JVM den Anschluss von Geräten ohne JVM. Die Bereitstellung und Nutzung des Dienstes erfolgt über das Gerät mit der JVM. Die Kommunikation zwischen dem Dienstprogramm und dem angeschlossenen Gerät kann dabei sehr aufwendig werden, weil je nach Anschlussart die Erkennung und Verwaltung des angesteckten Gerätes durchgeführt werden muss. Eine Variante des vorhergehenden Typs ist die Verwendung von Dienst-Proxys, die direkt mit dem Gerät ohne JVM kommunizieren können. Hierbei dient das Gerät mit JVM also rein zur Bereitstellung des Dienst-Proxys.

Zur Vereinfachung der Entwicklung von Jini-Anwendungen und Diensten gibt es eine Reihe von Hilfsklassen, die Standardaufgaben übernehmen können [Sun03a]. Als Beispiel werden die folgenden drei Klassen vorgestellt:

- **LookupDiscoveryManager**
Die Klasse „LookupDiscoveryManager“ (`net.jini.discovery`) erledigt alle Discovery-spezifischen Aufgaben, wie z. B. die Suche nach und Speicherung von allen Lookup-Diensten der angegebenen Gruppen oder die Aktualisierung der Liste der Lookup-Dienste bei Änderung der Gruppen.
- **LeaseRenewalManager**
Die Klasse „LeaseRenewalManager“ (`net.jini.lease`) kann zur automatischen Verlängerung von Leases verwendet werden.
- **JoinManager**
Die Klasse „JoinManager“ (`net.jini.lookup`) dient Diensten zur Durchführung des Join-Verfahrens, wie die Registrierung des Dienstes bei allen angegebenen Lookup-Diensten und die Verlängerung aller Lookup-Dienst-Leases durch Verwendung des „LeaseRenewalManagers“. Die Angabe der Lookup-Dienste kann dabei sehr einfach durch Übergabe eines „LookupDiscoveryManager“-Objekts erfolgen.

Neben zahlreichen weiteren Hilfsklassen von Sun, wie beispielsweise die Klassen des Pakets `com.sun.jini.landlord`, die Diensten zur Implementierung von Leases dienen, bietet Sun eine Implementierung aller Jini-Dienste und eine Menge von Beispielprogrammen zum Download⁵⁰ an.

⁵⁰Homepage: www.jini.org

2.2 Kopplung von Gerätenetzwerken im Heim

Wie bereits in der Einleitung angemerkt wurde und nunmehr auch aus der Darstellung der Funktionsweise und wesentlichen Eigenschaften der Gerätenetzwerke aus den verschiedenen Bereichen hervorgeht, bestehen zwischen den einzelnen Anwendungsbereichen im Heim in Bezug auf die Gerätenetzwerke Zielkonflikte, wie beispielsweise zwischen einer hohen Bandbreite und einer großen Ausdehnung (Übertragungsmedium, Buszugriff, ...) oder zwischen einer hohen Betriebssicherheit und einer großen Teilnehmerdynamik (Installation über ein Management-Werkzeug vs. Plug-and-Play, voreingestellte Kommunikationsteilnehmer vs. freie Kommunikation, ...). Andererseits werden zur Erreichung möglichst niedriger Kosten hauptsächlich nur die primär notwendigen Anforderungen einer Gerätevernetzung realisiert. So wird für die Vernetzung von Geräten der Unterhaltungselektronik eine zeitlich garantierte Übertragung von großen Datenmengen gefordert, während bei der Heim-Automatisierung eine gesicherte Übertragung von kleinen Datenmengen über große Distanzen zwischen vielen im ganzen Heim verteilten Geräten notwendig ist [TW02]. Anstatt die Heterogenität der Gerätevernetzung im Heimbereich durch ein neues „Super-Netzwerk“, das viele Kompromisse aufgrund des großen Einsatzspektrums eingehen muss, zu überwinden, sollen die spezialisierten und branchenbezogenen Netze gekoppelt werden.

Durch die Kopplung der branchenspezifischen Netzwerke können die Grenzen zwischen den verschiedenen Netzwerken überwunden und Dienste bzw. Funktionen, die in einem Netz verfügbar sind, auch Geräten in einem anderen Netzwerk zugänglich gemacht werden. Die Kopplung der Gerätenetzwerke ist somit die konsequente Fortführung des Vernetzungsgedankens, demzufolge der Nutzen eines Gerätenetzwerks größer ist als die Summe des Nutzens der einzelnen Gerätefunktionen. Ein weiterer Aspekt von Kopplungen ist, dass durch die Möglichkeit einer Kopplung das Investitionsrisiko in eine bestimmte Vernetzungstechnologie abgefedert wird beziehungsweise bestehende Investitionen in Geräte und Infrastruktur nicht verloren gehen.

Kopplungen zwischen den Gerätenetzwerken im Heimbereich bringen so einen Mehrwert für den Nutzer. Sie ermöglichen einerseits die einfachere Bedienbarkeit der Geräte durch die gemeinsame Nutzung von Benutzerschnittstellen (z. B. HAVi-EIB Kopplung: Verwendung eines HAVi-TV zur Einstellung einer über EIB angebotenen Heizungsanlage) und andererseits die Überwindung von Beschränkungen des Netzwerks, wie z. B. dessen maximale Ausdehnung (z. B. HAVi-HAVi Kopplung: Programmierung eines HAVi-VCR im Heim über ein entferntes via TCP/IP angebotenes HAVi-Netzwerk). Des Weiteren werden durch die Kopplung neue, bereichsübergreifende Anwendungen möglich. Eine Kopplung von Gerätenetzwerken ist jedoch unweigerlich auch mit Verlusten behaftet (siehe Abschnitt 3.6).

Ganz entscheidend für die Verwendung von Kopplungen im Heimbereich ist, dass diese allgemein einsetzbar sind und ohne wesentlichen Konfigurationsaufwand auskommen. Nur durch die allgemeine Einsetzbarkeit der Kopplung können die unterschiedli-

chen im Heimbereich eingesetzten Anwendungen und Geräte unterstützt werden. Seitens der Nutzer wird außerdem kein oder nur ein sehr geringer Konfigurationsaufwand der Kopplung zur Inbetriebnahme und bei Änderungen in den beteiligten Netzwerken (Hinzufügen und Entfernen von Geräten) akzeptiert [Ros02a]. In Abschnitt 3.4.1 werden die genannten Forderungen präzisiert und Kopplungen, welche diese Forderungen erfüllen als *dynamische* Kopplungen bezeichnet. So kann hier bereits festgehalten werden, dass eine geeignete Kopplung von Gerätenetzwerken im Heimbereich dynamisch sein muss.

2.2.1 Bereichsübergreifende Anwendungen

Bereichsübergreifenden Anwendungen nutzen Funktionen, die aus unterschiedlichen Anwendungsbereichen des Heims stammen. Beispiele dafür sind programmierbare Szenarien zur vordefinierten Einstellung mehrerer Geräte aus unterschiedlichen Bereichen, wie „TV-Prime-Time“ (Licht im Wohnzimmer gedimmt auf 20%, TV-Audio über HiFi-Anlage mit Lautstärke auf Stufe 5, TV-Videoeinstellungen Standard, ...) oder „Wake-up“ (bei Erreichen der Weckzeit: Radio ein, Jalousie öffnen, persönliche Verkehrs- und Wetterinfos über TV bereitstellen, ...), und erweiterte Einstellmöglichkeiten durch die Verknüpfung von Informationen aus mehr als einem Bereich, wie die Programmierung der Heizung entsprechend der über das TV-Gerät oder dem PC abrufbaren Wetterdaten.

Soweit die programmierten Szenarien nicht durch den Benutzer direkt ausgelöst werden (z. B. TV-Prime-Time), sondern deren Ausführung vom Eintreten definierter Ereignisse abhängen (z. B. Wake-up: Weckzeit), ist zur Überwachung der Ein- und Ausschaltbedingungen eine eigene Anwendung nötig. Ein Beispiel für eine solche Anwendung ist eine Rule-Engine, die eine regelbasierende Ansteuerung von Geräten ermöglicht. Eine Regel besteht dabei aus einer Bedingung und einer Aktion, die bei Eintreten der Bedingung veranlasst wird. Hierbei besteht (kurz gefasst) eine Bedingung aus einer oder mehreren logisch verknüpfbaren Zeit- oder Statusbedingungen (Geräte- bzw. Netzwerkzustand) und eine Aktion aus einer oder mehreren aufzurufenden Gerätefunktionen. Detailliertere Informationen über die Rule-Engine und die Beschreibung einer Implementierung sind in [WT04] zu finden.

Ein Aspekt, der zwar schon bei der homogenen Vernetzung von Geräten zu berücksichtigen ist, jedoch durch die größere Anzahl der Geräte und Funktionen durch Ankoppeln zusätzlicher (bereichsfremder) Netzwerke noch verstärkt wird, sind die gestiegenen Anforderungen an die Benutzerschnittstelle zur Bedienung bzw. Verwaltung der Geräte. Während die Bedienung eines einzelnen herkömmlichen TV-Gerätes noch relativ übersichtlich ist, erfordert ein digitales TV-Gerät, das zur Steuerung verschiedener über einen IEEE 1394-Bus angeschlossener Geräte dienen kann, eine wesentlich aufwendigere Benutzerschnittstelle. So muss etwa für jede Anforderung des Benutzers erst ermittelt werden, an welches Gerät diese Anforderung geleitet werden soll: eine Änderung der Lautstärke kann z.B. am TV-Gerät selbst, an der angeschlossenen Dolby-Surround-Anlage oder einem TV-Gerät eines anderen Zimmers erfolgen. Ein weiteres Beispiel ist

die Behandlung der vielen möglichen Audio/Video-Verbindungen: als Datenquelle für das erwähnte TV-Gerät kommen nicht nur der integrierte Tuner, AV-1 oder AV-2 in Frage, sondern prinzipiell alle direkt oder indirekt am IEEE 1394-Bus angeschlossenen Audio/Video-Datenquellen (Set-Top-Box, PC, DVD-Player, AV-Hard-Disk, ...) [ICT04]. Der Benutzerschnittstelle kommt so eine wesentliche Bedeutung zu, da viele der neuen Möglichkeiten durch Gerätenetze und deren Kopplungen erst über diese nutzbar gemacht werden können.

2.2.2 Kopplungen

Zur Verbindung von Gerätenetzen im Heimbereich sind bereits eine Reihe von Konzepten und Implementierungen publiziert worden. Einige davon werden in diesem Abschnitt vorgestellt und dazu wie folgt grob unterteilt:

- Kopplungen zwischen zwei Netzwerken:
 - Jini, EIB [RHB02, KK99]
 - Jini, LON [KL01]
 - UPnP, EIB [WG00]
 - HAVi, HAVi [WUB01]
 - Jini, HAVi [Ski02, Rah01]
 - Jini, UPnP [ACGI03]
 - WAP, HAVi [NMV02, NMV03]
 - UPnP, HAVi [BGSS03]

Die einzelnen Kopplungen unterscheiden sich nicht nur hinsichtlich der zu verbindenden Netzwerke, sondern auch hinsichtlich der Zielsetzung der Verbindung, wie beispielsweise die Überwindung von Beschränkungen eines Gerätenetzes [WUB01], die Ansteuerung von Geräten über eine andere API als die ursprüngliche API [RHB02] oder die Benutzung der Geräte über eine Benutzerschnittstelle eines anderen Netzwerks [NMV03].

- Konzepte zur Kopplung mehrerer Netzwerke:
 - „Framework for the composition and interoperation of the home appliances based on heterogeneous middleware in residential networks“ [Cho02]: Für die Kopplung von verschiedenen angeschlossenen Gerätenetzen sorgt ein zentrales Gerät, das „Residential-Gateway“. Es besteht aus jeweils einem Middleware Manager für jedes angekoppelte Netzwerk und einem Middleware Broker, der für die Kommunikation zwischen den Middleware Managern verantwortlich ist.

- „A Virtual Overlay Network for Integrating Home Appliances“ [NSA02]: Über ein zentrales als Virtual Overlay Network bezeichnetes Netzwerk können verschiedene Gerätenetzwerke gekoppelt werden. Das zentrale Netzwerk besteht aus einzelnen miteinander über TCP/IP verbundenen Geräten, den so genannten Application Level Gateways (ApGW), die untereinander über HTTP kommunizieren und für die Umsetzung zwischen der allgemeinen API des zentralen Netzwerks und der API des jeweils angebotenen Netzwerks sorgen.
- Verbindungen zwischen verschiedenen Übertragungssystemen und -medien:
 - „Transparent Bridges for IEEE 1394 Networks with Hiperlan2 Interconnection between IEEE 1394 Portals“ [BPS⁺02]: Die transparenten Brücken und ihre drahtlosen Verbindungen sorgen für eine Verbindung der angeschlossenen IEEE 1394-Cluster derart, dass sie für alle dadurch angebotenen IEEE 1394-Geräte wie ein einziger großer IEEE 1394-Bus wirkt.
 - „Overview of IEEE 1394 serial bus bridging solutions“ [Bra02]: Im Gegensatz zu IEEE 1394.1-Brücken ermöglichen transparente Brücken die Weiterverwendung von bestehenden Geräten ohne Änderung ihrer Hard- oder Software. Nachteile der transparenten Brücken sind allerdings die Limitierung der maximalen Geräteanzahl auf die eines einzelnen IEEE 1394-Busses (63) und eine mangelnde Bandbreitenausnutzung auf Grund der fehlenden Segmentierung der einzelnen Busse.
 - „Übertragung von IEEE1394-Datenverkehr über IP-Verbindungen“ [Anh02]: Es wird gezeigt, wie eine für höhere Protokollschichten weitestgehend transparente Nachbildung eines IEEE 1394-Busses auf der Basis von IP und dessen Verbindung mit einem „echten“ IEEE 1394-Bus erfolgen kann und damit 1394-basierte Netzwerkstandards wie z. B. HAVi für IP-basierte Endgeräte verfügbar gemacht bzw. letztere in 1394-Netzwerke integriert werden können.
 - „Media Access Control Bridges“ [IEE98]: Die MAC-Brücken arbeiten unterhalb der Grenze der 802.3-MAC-Dienst-Schicht und sind transparent in Bezug auf Protokolle, die oberhalb dieser Grenze arbeiten, wie beispielsweise IP. Sie ermöglichen so die Verbindung von unterschiedlichen IEEE 802.3 [IEE02b] (CSMA/CD LANs) oder IEEE 802.11 (Wireless LANs) Netzen.
- Produktankündigungen bzw. Prototypenpräsentationen:
 - „Loewe Wireless Home Automation“ [Loe03]: Der deutsche High-End TV-Gerätehersteller Loewe bietet als Zubehör für seine TV-Geräte ein „Wireless Home Automation Starter Set“ zur Steuerung von Geräten der Heim-Automatisierung über die normale TV-Fernbedienung und das On-Screen-Display an. Als Heim-Automatisierungs-Netzwerk wird ein Funk-Bus der Firma Gira eingesetzt und das TV-Gerät mit einem entsprechenden Funk-Sendemodul aufgerüstet.

- „HAVi Network Adapter - IEEE 1394.1 / HAVi-Bridge“ [Fro02]: Ein im Rahmen des europäischen Forschungsprojekts HomeNet2Run (HN2R) [Hom03] vorgestellter Prototyp der Firma Canon. Er ermöglicht unter anderem die Verbindung von mehreren HAVi-Netzwerken über IEEE 1394.1-Bridges. Als HAVi-Stack wird der in Kapitel 4 beschriebene Stack der Firma dmn Software-Entwicklung GmbH verwendet.
- „Streaming Set-Top-Box with HAVi-UPnP Gateway Functionality“ [Rie02]: Ein im Rahmen des europäischen Forschungsprojekts HomeNet2Run (HN2R) [Hom03] entwickelter Prototyp der Firma Thomson, der eine Kopplung zwischen HAVi und UPnP in beide Richtungen ermöglicht.

Wie bereits angesprochen unterscheiden sich die meisten der genannten Kopplungen nicht nur hinsichtlich der zu verbindenden Netzwerke, sondern auch hinsichtlich der Zielsetzung der Verbindung. In keiner der oben zitierten Publikationen wird jedoch ein allgemeines Modell zur Kopplung von Gerätenetzwerken im Heimbereich beschrieben oder referenziert. Einzig auf das OSI-Referenzmodell [IT94, Zim80, Tan00] wird bei einigen zur Beschreibung der grundsätzlichen Datenkommunikation der einzelnen Gerätenetze Bezug genommen. Wie in Abschnitt 3.7 noch ausgeführt wird, können jedoch Aspekte wie die automatische Bekanntmachung und Beschreibung der Geräte im Netzwerk (zentrale oder verteilte Datenbank mit Abfragemöglichkeiten, periodisches Senden von Alive-Nachrichten, Configuration Rom, ...), die Geräteansteuerung (Schnittstelle zur Geräte-HW/SW) oder die Benutzerschnittstelle durch das OSI-Referenzmodell und die verfügbaren Application Layer Services (X.217, X.218, X.219, ...) nicht geeignet beschrieben werden. Auf Grund des Fehlens eines allgemeinen Modells zur Kopplung von Gerätenetzwerken im Heimbereich ist der Vergleich von Kopplungen bzw. die Suche nach geeigneten Implementierungen sehr aufwendig und es fehlen Richtlinien oder Kriterien für die Entwicklung von neuen Kopplungen.

Die Zielsetzung des nächsten Kapitels ist, ein allgemeines Modell zur Kopplung von Gerätenetzwerken im Heimbereich zu definieren, das sich einerseits zur Analyse und Beschreibung von bestehenden und andererseits als Grundlage zum Design von neuen Kopplungen eignet.

In den nächsten Kapiteln werden die zitierten Kopplungen noch eingehender behandelt und, soweit ausreichende Informationen veröffentlicht wurden, anhand des im nächsten Kapitel definierten Kopplungsmodells bewertet.

Kapitel 3

Modell zur Kopplung von Gerätenetzwerken

Die Vernetzung von Geräten im Heimbereich erfordert im Wesentlichen Mechanismen zur Adressierung, Übertragung, Bekanntmachung, Beschreibung, Ansteuerung, Benachrichtigung und Darstellung der Geräte beziehungsweise ihrer Kommunikationsobjekte. Obwohl verschiedene Konzepte zur Vernetzung von Geräten entwickelt wurden, basieren alle auf diesen grundlegenden Mechanismen. Zum Vergleich der unterschiedlichen Vernetzungskonzepte und der Beschreibung von Kopplungen zwischen den einzelnen Netzwerken wird in diesem Kapitel ein Kopplungsmodell entwickelt. Wesentlicher Bestandteil dieses Modells ist die in Abschnitt 3.1 definierte Gerätearchitektur, welche die für die Vernetzung notwendigen Dienste definiert und diese in drei Schichten gruppiert. In Abschnitt 3.2 erfolgt eine Einstufung von jeweils zwei Gerätenetzwerken der drei Bereiche „Heim-Automatisierung“, „Unterhaltungselektronik“ und „Small Office/Home Office“ entsprechend dieser Gerätearchitektur. Hierbei wird für jedes Gerätenetzwerk analysiert, wie die grundlegenden Mechanismen der Gerätevernetzung realisiert werden. Eine Kopplung von Gerätenetzwerken erfolgt durch Umsetzung auf einer der Schichten der Gerätearchitektur, nach welcher auch das entsprechende Gateway benannt wird (Abschnitt 3.3). Die unterschiedlichen Arten von Kopplungen (einseitig/zweiseitig, direkt/indirekt, statisch/dynamisch, ...) werden im darauf folgenden Abschnitt 3.4 beschrieben. Der Einsatz von mehreren Kopplungen (Kopplungssystemen) wird im Abschnitt 3.5 behandelt. Den Abschluss des Kapitels bilden die Behandlung von möglichen Kopplungsverlusten (Abschnitt 3.6) und ein Vergleich des hier beschriebenen Modells mit anderen Modellen der Gerätekommunikation (Abschnitt 3.7).

3.1 Gerätearchitektur

Wie die Analyse der im vorhergehenden Kapitel vorgestellten Gerätenetzwerke nachfolgend zeigen wird, können als wesentliche Mechanismen zur Vernetzung von Geräten im Heimbereich die folgenden 6 Dienste identifiziert werden: *Communication*, *Discovery*, *Description*, *Control*, *Notification* und *User Interface*. Des Weiteren können die Dienste entsprechend ihres Aufgabenbereichs auf die drei Schichten *System Layer*, *Service Layer* und *User Interface Layer*, wie in Abbildung 3.1 dargestellt, verteilt werden. Hierbei ver-

wendet eine höhere Schicht jeweils nur die Dienste der direkt darunterliegenden Schicht. Die Bildung der drei Schichten erfolgte dabei in Anlehnung an [Zim80] anhand der folgenden Grundprinzipien: (1) Es sollen nur so viele Schichten wie unbedingt notwendig erstellt werden; (2) Offensichtlich unterschiedliche Funktionen sollen in unterschiedlichen Schichten sein; (3) Ähnliche Funktionen sollen in einer Schicht zusammengefasst werden; (4) Die Anzahl an Interaktionen zwischen den Schichten soll möglichst minimiert werden; (5) Funktionen einer Schicht sollen nur von der direkt darüberliegenden Schicht benutzt werden.

Der *Transport* repräsentiert das Übertragungssystem und -medium, wie z. B. den IEEE 1394-Bus oder ein IP-basierendes Netzwerk (IP/Ethernet, IP/WLAN, ...). Oberhalb der Gerätearchitektur befindet sich der Benutzer, der über den User Interface Layer die Dienste des Gerätenetzwerks nutzen kann.¹

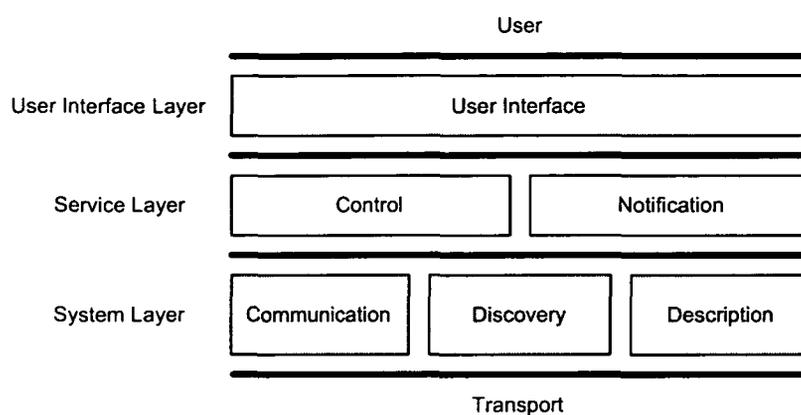


Abbildung 3.1: Dreischichtige Gerätearchitektur

3.1.1 System Layer

Der *System Layer* liegt oberhalb des Transports und bildet damit die erste Schicht der Gerätearchitektur. Die Gerätekommunikation erfolgt über so genannte *Kommunikationsobjekte*, die Quelle oder Ziel von Nachrichten sind und zur Kommunikation, Bekanntmachung und Beschreibung die Dienste dieser Schicht (Communication, Discovery und Description) nutzen.

Communication ist verantwortlich für die Kommunikation zwischen Kommunikationsobjekten und besteht aus den Teilen Addressing, Transmission und Interaction:

- *Addressing*

Addressing ist verantwortlich für die Adressierung von Kommunikationsobjekten.

¹Eine Gegenüberstellung der Gerätearchitektur mit den Schichten des OSI-Modells wird in Abschnitt 3.7 durchgeführt.

Dies beinhaltet die Vergabe einer Adresse für ein Kommunikationsobjekt und soweit notwendig auch die Zuordnung dieser Adresse zu einer Adresse des Transports für die Übertragung im Netzwerk. Die wesentliche Anforderung an die Adresse eines Kommunikationsobjekts ist, dass es über diese Adresse angesprochen werden kann². Zumeist besteht sie aus mehreren Teilen und setzt sich aus der Geräteadresse und einer lokalen Adresse zusammen. Ein Beispiel dafür ist der HAVi Software Element Identifier (SEID), welcher aus dem Global Unique Identifier (GUID) des Gerätes und einem lokalen HAVi Software Element Handle (swHandle) besteht. Ein weiteres Beispiel ist die Adresse einer LON-Netzwerkvariable, sie setzt sich aus einem 14-Bit langen Netzwerkvariablenselektor und einer Geräteadresse zusammen. Soweit ein Netzwerk aus mehreren Teilnetzen bestehen kann, muss die Adresse nicht nur eindeutig sondern auch strukturiert³ sein. Wird vom Transport eine andere Adresse zur Übertragung im Netzwerk eingesetzt, so ist eine Umsetzung der Adresse des Kommunikationsobjekts in eine entsprechende Adresse des Transports notwendig. Beispiele dafür sind der „Domain Name System (DNS)“-Dienst [Moc87a, Moc87b], der eine DNS-Adresse einer IP-Adresse zuordnet oder der HAVi-Systemdienst Communication Media Manager 1394 (Cmm1394), der für die Umsetzung der statischen GUIDs in die dynamische IEEE 1394-NodeIDs verantwortlich ist. Die genannten Beispiele veranschaulichen auch einige der Gründe für die Verwendung von unterschiedlichen Adressen auf den verschiedenen Ebenen, so ist die alphanumerische DNS-Adresse benutzerfreundlicher als die rein numerische IP-Adresse, oder die Handhabung der dauerhaften GUID einfacher als die nach jedem IEEE 1394-Bus-Reset neu zugewiesene NodeID. Ein weiterer, ganz wesentlicher Grund für unterschiedliche Adressen ist die notwendige Unteradressierung zur Unterscheidung von mehreren Kommunikationsobjekten eines Gerätes (z. B. HAVi: SEID = GUID + swHandle).

- *Transmission*

Transmission ist verantwortlich für die Übertragung von Nachrichten zwischen Kommunikationsobjekten. Das beinhaltet (1) die Definition eines Nachrichtenformats, (2) Mechanismen zum Senden und Zustellen von Nachrichten und (3) in Abhängigkeit vom Transport auch die Fragmentierung und bestätigte Übertragung von Nachrichten.

- *Interaction*

Für die Interaktion der Kommunikationsobjekte sind vor allem der zweiteilige Anfrage-Antwort-Mechanismus und die einfache Mitteilung von Bedeutung. Der *Anfrage-Antwort-Mechanismus* dient zum entfernten Aufrufen von Funktionen und

²Eindeutigkeit kann aufgrund der oft verwendeten Gruppenadressierung (EIB-Kommunikationsobjekte werden über Gruppenadressen angesprochen, mehrere LON-Eingangs-Netzwerkvariablen können mit der selben LON-Ausgangs-Netzwerkvariablen verbunden sein, ...) nicht gefordert werden.

³Als strukturierte Adresse wird hier eine Adresse bezeichnet, die Informationen über das zugehörige Netzwerk (z. B. Netzwerk-ID oder Bus-ID) enthält. Diese Information ist für die Weiterleitung von Nachrichten zwischen einzelnen Teilnetzen (Routing) notwendig.

besteht aus zwei Teilen: der Anfrage, welche die auszuführende Funktion bezeichnet und die Eingangsparameter enthält, und der dazugehörigen Antwort, welche den Rückgabewert und die Rückgabeparameter enthält. Der entfernte Funktionsaufruf wird als *synchroner Funktionsaufruf* bezeichnet, wenn der Aufrufer bis zum Eintreffen der Antwort oder des Verstreichens eines vorher bestimmten Zeitintervalls blockiert wird, anderenfalls als *asynchroner Funktionsaufruf*. Im Gegensatz zum Anfrage-Antwort-Mechanismus besteht die *einfache Mitteilung* nur aus einer Nachricht ohne Antwort. In beiden Fällen kann die Übertragung bestätigt oder unbestätigt erfolgen.

Discovery ist verantwortlich für die Bekanntmachung der Kommunikationsobjekte des Gerätes. Die Bekanntmachung soll hierbei nur die notwendigsten Informationen enthalten, eine genauere Beschreibung erfolgt durch die weiter unten beschriebene Description. Damit zwei Kommunikationsobjekte miteinander kommunizieren können, müssen zumindest die Adressen der Kommunikationspartner bekannt sein. Hierzu existieren verschiedene Ansätze:

- *manuelle Bekanntmachung*
Bei der rein manuellen Bekanntmachung werden die Adressen der Kommunikationspartner manuell den Kommunikationsobjekten zugewiesen. Dies erfolgt zumeist durch Einträge in entsprechende Tabellen der Geräte. Die manuelle Konfiguration der Geräte wird oft dadurch erleichtert, dass die Geräte „auf Knopfdruck“ eine Rundsende-Nachricht mit der eigenen Adresse senden können (z. B. Service Pin bei LON-Knoten).
- *direkte Bekanntmachung*
Bei der direkten Bekanntmachung erfolgt die Bekanntmachung des Kommunikationsobjekts durch das periodische Senden von so genannten Alive-Nachrichten als Rundsendungen. Vor Beendigung eines Kommunikationsobjekts werden die Bekanntmachungen durch das Senden so genannter ByeBye-Nachrichten zurückgenommen. Damit auch nicht mehr verfügbare Kommunikationsobjekte durch die Entfernung eines Gerätes vom Netzwerk erkannt werden, gibt es zwei Möglichkeiten: Entweder der Transport kann das Entfernen eines Gerätes erkennen, wie dies z. B. beim IEEE 1394-Bus der Fall ist, und damit kann auch die Entfernung der Kommunikationsobjekte des entfernten Gerätes erkannt werden, oder nicht mehr vorhandene Kommunikationsobjekte werden aufgrund des Ausbleibens der regelmäßigen Alive-Nachrichten erkannt. Im zweiten Fall ist allerdings die maximale Erkendauer gleich dem Sendeintervall der Alive-Nachrichten. Sowohl das Senden als auch die Verarbeitung der Bekanntmachungsnachrichten erfolgt entweder durch das Kommunikationsobjekt selbst oder durch einen Bekanntmachungsdienst.
- *indirekte Bekanntmachung* über einen Registrierungsdienst
Hierbei meldet sich ein neues Kommunikationsobjekt bei einem zentralen oder verteilten Registrierungsdienst. Dieser speichert neben anderem die Adresse des

neuen Kommunikationsobjekts und stellt den Zugriff auf diese Informationen bereit. Kommunikationsobjekte können diese Informationen abrufen und je nach Vernetzungsstandard auch über die Existenz von neuen Kommunikationsobjekten benachrichtigt werden. Vorausgesetzt wird hierbei jedoch, dass die Adresse des zentralen oder verteilten Registrierungsdienstes bekannt ist. Dies kann durch manuelle Konfiguration oder über die direkte Bekanntmachung erfolgen. Wird ein Kommunikationsobjekt beendet, meldet es sich zuvor beim Registrierungsdienst ab, der je nach Vernetzungsstandard daran interessierte Kommunikationsobjekte benachrichtigt. Für das Erkennen von nicht mehr verfügbaren Kommunikationsobjekten durch die Entfernung eines Gerätes vom Netzwerk gibt es wieder zwei Möglichkeiten: Entweder der Transport kann das Entfernen eines Gerätes erkennen und damit den Registrierungsdienst veranlassen die Kommunikationsobjekte des entfernten Gerätes vom Registrierungsdienst zu entfernen, oder es wird eine regelmäßige Erneuerung des Eintrag eines Kommunikationsobjekts im Registrierungsdienst gefordert, sodass Einträge von nicht mehr vorhandenen Kommunikationsobjekten bei Ausbleiben der Erneuerung automatisch entfernt werden. Im zweiten Fall ist allerdings die maximale Erkendauer gleich dem Erneuerungsintervall des Eintrags.

Description ist verantwortlich für die Bereitstellung von Informationen über Kommunikationsobjekte bzw. deren zugrunde liegenden Dienste und Funktionen des Geräte Netzwerks. Zu den wesentlichen Informationen gehören:

- Adressinformationen⁴
- Beschreibung der Fähigkeiten und Eigenschaften
Diese erfolgt zumeist in zusammengefasster Form über eine standardisierte Typenbezeichnung. Beispiele dafür sind der HAVi Software Element Type VCR_FCM oder TUNER_FCM, das LON Mark Functional Profile „Thermostat“ [Lon96b] oder „Switch“ [Lon96a] und das UPnP Device „Printer“ [AHLZ02] oder „Internet Gateway“ [IW01].
- Dauerhaft eindeutiger Kennzeichner
Sofern die Adresse für einen dauerhaft eindeutigen Kennzeichner nicht ausreichend ist, weil sie sich nach einem Neustart, dem An-/Abstecken des Gerätes vom Netzwerk oder sonst einem Grund ändern kann, ist ein solcher notwendig (z. B. HAVi: HUID, Jini: ServiceId, UPnP: UUID).
- Herstellerinformationen
Informationen, wie z. B. Herstellername, Seriennummer, Gerätebezeichnung oder HW/SW-Versionsnummer.

Neben der manuellen Bereitstellung der Informationen durch Verweis auf Standards, Spezifikationen oder die Gerätedokumentation gibt es verschiedene Konzepte der Selbst-

⁴siehe auch Dienst Communication, Teil Addressing

beschreibung. Eine Möglichkeit ist die Speicherung der Informationen im Gerät an vorgegebenen oder über Discovery bekanntgemachten Stellen, die über das Netzwerk ausgelesen werden können. Beispiele dafür sind das Configuration Rom eines IEEE 1394-Gerätes oder die XML-basierte Device Description eines UPnP-Gerätes. Die Informationen können aber auch direkt durch das Kommunikationsobjekt selbst bereitgestellt werden, wobei der Zugriff über eine standardisierte API wie beim weiter unten beschriebenen Mechanismus Control erfolgt (z. B. `GetHUID`, `GetServiceId`, `GetDeviceManufacturer`). Weitere Möglichkeiten ergeben sich mit den beim Discovery eingesetzten Mechanismen. Dies ist einerseits der Registrierungsdienst, auf welchen ebenfalls über eine standardisierte API zugegriffen werden kann, oder die Verlautbarung der Beschreibung als Teil der Alive-Nachrichten. Im letzteren Fall ist allerdings die Datenmenge an Informationen stark zu beschränken, weil die Alive-Nachrichten periodisch gesendet werden und sonst unverhältnismäßig viel Bandbreite beanspruchen würden.

Zusammenfassend lässt sich sagen, dass Discovery Kommunikationsobjekte sichtbar macht, Description diese beschreibt und Communication die Kommunikation zwischen diesen ermöglicht. Die genannten Mechanismen unterstützen die nächst höhere Schicht bei der Realisierung von Diensten für ein vernetzte Heim. Ist die Beschreibung der Dienste und Protokolle dieser Schicht nicht öffentlich verfügbar, so gelten diese Schicht und die darüberliegenden Schichten als proprietär. In diesem Fall gibt es keine Interoperabilität, weil zur Bereitstellung und Nutzung von Diensten proprietäre Informationen notwendig sind.

3.1.2 Service Layer

Der *Service Layer* ermöglicht die Bereitstellung und Nutzung von Diensten des GeräteNetzwerks durch Kommunikationsobjekte. Jedes Gerät bietet verschiedene Funktionen, wie z. B. das Abspielen einer CD, die Auswahl eines Fernsehprogramms, das Einscannen einer Seite, das Ausdrucken einer Seite, das Dimmen des Lichts auf 50%, das Ausschalten der Heizung, usw. Die Funktionen werden durch Geräte des Netzwerks, wie z. B. CD-Player, TV-Gerät, Scanner, Drucker, Lampe, Heizung bereitgestellt. Ein *Dienst des GeräteNetzwerks* besteht aus einer oder mehreren Funktionen, die auch von unterschiedlichen Geräten bereitgestellt werden können. Beispielsweise bilden die Funktionen `Print`, `SetPrintMode` eines Druckers einen einfachen Druck-Dienst, die Funktionen `Scan`, `SetScanMode` eines Scanners einen einfachen Scan-Dienst und mit den Funktionen beider Geräte kann ein einfacher Kopier-Dienst gebildet werden. Die Bereitstellung und Nutzung der Dienste durch Kommunikationsobjekte erfordert einerseits die Abbildung der Dienste und ihrer Funktionen auf Kommunikationsobjekte und andererseits die Bereitstellung einer API zur Nutzung der Dienste.

Control und *Notification* bilden gemeinsam die Schnittstelle zur Gerätehardware und -software (siehe Abbildung 3.2). *Control* ist für die Ansteuerung der Geräte durch andere (entfernte) Kommunikationsobjekte verantwortlich. Die Kommunikation wird hierbei

vom Benutzer des Dienstes initiiert. Für die andere Kommunikationsrichtung, d. h. für die Meldung von Statusänderungen oder von anderen Informationen des Dienstes an daran interessierte Kommunikationsobjekte, ist Notification verantwortlich. In diesem Fall wird die Kommunikation vom Dienstbereitsteller initiiert.

Es werden zwei Arten von Notification unterschieden: die *direkte Benachrichtigung*, bei der die Registrierung beim und die Benachrichtigung durch den Dienstbereitsteller selbst erfolgt, und die *indirekte Benachrichtigung*, bei der die Registrierung bei und die Benachrichtigung durch einen zwischen dem Dienstbereitsteller und dem Dienstanutzer dazwischengeschalteten Verteildienst erfolgt. Die Benachrichtigung des Verteildienstes kann wiederum auf eine der beiden Arten erfolgen, wobei bei einer Kette von Verteildiensten der Erste über die direkte Benachrichtigung und alle Weiteren über indirekte Benachrichtigungen benachrichtigt werden. Die Registrierung des zu benachrichtigenden Kommunikationsobjekts erfolgt entweder manuell, meist durch Speicherung der Empfänger-Adressen in entsprechenden Tabellen, oder über einen entfernten Funktionsaufruf.

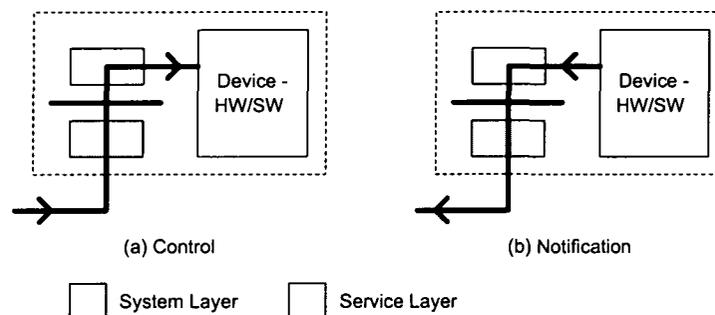


Abbildung 3.2: Control und Notification

Die Nutzung der durch Control und Notification bereitgestellten Dienste erfolgt durch entfernte Funktionsaufrufe oder einfachen Mitteilungen, welche in entsprechenden API-Definitionen beschrieben sind. Für entfernte Funktionsaufrufe besteht die API-Definition aus der Beschreibung der Eingangsparameter, der Rückgabeparameter, der auszuführenden Funktion und des Rückgabewerts. Die Beschreibung der API-Definition erfolgt über den weiter oben beschriebenen Dienst Description durch Verweis auf Standards, Spezifikationen oder Gerätedokumentation oder aber auch durch Selbstdokumentation, z. B. in Form eines XML-Dokuments gemäß XML-Schema, wie dies bei UPnP geschieht. Für Control und Notification ist weiters eine standardisierte Interpretation der Daten Voraussetzung. Beispiele dafür sind: die HAVi-Spezifikation, welche für entfernte Funktionsaufrufe definiert, wie HAVi-Datentypen in einen byteweisen Datenstrom konvertiert (Marshalling) und wiederhergestellt (Unmarshalling) werden, oder die LON Standard Network Variable Types (SNVT), welche standardisierte Typen für die über einfache Mitteilungen gesendeten LON-Netzwerkvariablen definieren. Eine weitere Kommunikationseigenschaft ist, ob die Kommunikation frei oder gebunden ist, d. h. ob die Kommuni-

kationsteilnehmer frei gewählt werden oder voreingestellt sind. Control basiert zumeist auf einer freien Kommunikation, während bei Notification die zu benachrichtigenden Kommunikationsobjekte durch eine vorhergehende Registrierung vorgegeben sind.

Für die Verwendung von Geräten unterschiedlicher Hersteller im Netzwerk ist eine standardisierte Schnittstellendefinition notwendig, die verschiedene Dienst- oder Gerätetypen und die dazugehörigen Funktionen und Eigenschaften definiert. Ist diese nicht öffentlich verfügbar, so bedeutet dies den Verlust der Interoperabilität, weil die Dienste nicht von Anwendungen anderer Hersteller verwendet werden können. In manchen Fällen kann dieser Mangel durch eine geeignete Benutzerschnittstelle mit bekannten Bezeichnungen, wie „play“, „stop“ oder „on“, „off“ oder Symbolen auf den Schaltflächen überwunden werden, jedoch ist auch in diesem Fall die nicht interaktive Nutzung der Dienste nahezu unmöglich.

Zusammengefasst: Control und Notification ermöglichen die Bereitstellung und Nutzung von Diensten für das vernetzte Heim und bilden damit den Service Layer.

HAVi Functional Component Modules (FCMs) sind ein Beispiel für Kommunikationsobjekte, die sowohl die Aufgaben von Control als auch von Notification wahrnehmen. Im HAVi-Netzwerk bilden FCMs die Schnittstelle zur Gerätehardware und -software. Die Ansteuerung der Geräte (Control) basiert auf entfernten Funktionsaufrufen gemäß der standardisierten API der FCM (`AvDisc::Play`, `Tuner::SelectService`, ...). Auch die Registrierung für und Bekanntmachung von Statusänderungen (`AvDiscStateChanged`, `TunerServiceChanged`, ...) erfolgt über entfernte Funktionsaufrufe gemäß der standardisierten API der FCM für eine direkte Benachrichtigung (`Fcm::SubscribeNotification`, `<Client>::FcmNotification5`) bzw. des Event Managers als Verteildienst bei der indirekten Benachrichtigung (`EventManager::Subscribe`, `<Client>::EventManagerNotification`).

3.1.3 User Interface Layer

Die Benutzerschnittstelle (*User Interface*) definiert die Schnittstelle zum Benutzer. Sie ermöglicht dem Benutzer die Bedienung der Dienste des Gerätenetzwerks durch die Verwendung von Ein-/Ausgabeeinheiten zur Darstellung von Menüs, Auswahlfeldern, Schaltflächen, Bildern, Zeichen, Texten, die Ausgabe von Sprache, Musik, Video, die Aufnahme von Benutzereingaben via Fernbedienung, Maus, Kamera, Sprache und dergleichen mehr. Die Verbindung der Benutzerschnittstelle mit den Diensten des Gerätenetzwerks erfolgt über eine Anwendung der Service-Schicht. Für die Benutzerschnittstelle gibt es verschiedene Konzepte. Die hier wesentlichen Unterscheidungen lassen sich anhand der Antworten auf die folgenden drei Fragen beschreiben:

- (1) Welche Anforderungen werden an die Ein-/Ausgabefähigkeiten der Geräte gestellt?

⁵Der Vollständigkeit halber sei angemerkt, dass die Benachrichtigung des HAVi Event Managers über `EventManager::PostEvent` erfolgt und keiner vorhergehenden Registrierung des Event Managers bei der FCM bedarf.

- (2) Wo erfolgt die Beschreibung der dem Benutzer darzustellenden Objekte?
- (3) Wo werden die Eingaben der Benutzer verarbeitet?

Sind die Anforderungen an die Ein-/Ausgabefähigkeiten standardisiert, so kann die Anwendung in Bezug auf die Benutzerschnittstelle unabhängig von der verwendeten Geräteplattform realisiert werden. Bei der Verwendung von Java ist die Anwendung sogar komplett unabhängig von der Zielplattform und kann damit über das Netzwerk geladen und auf dem entsprechenden Zielgerät ausgeführt werden. HAVi Havlets sind ein Beispiel für dieses Konzept. Kurz gefasst sind Havlets in Java geschriebene Anwendungen, die standardisierte Ein-/Ausgabefähigkeiten verwenden und auf entsprechenden HAVi-Geräten⁶ ausgeführt werden können. Im Gegensatz dazu können Anwendungen, die proprietär auf Fähigkeiten einer bestimmten Geräteplattform zurückgreifen, die besonderen Fähigkeiten der Geräte besser nutzen. In beiden Fällen erfolgt die Beschreibung der Darstellung und Verarbeitung der Benutzereingaben in der lokal ausgeführten, wenn auch zuvor heruntergeladenen, Anwendung. Ein anderes Konzept basiert auf einem Renderer, der die Beschreibung der darzustellenden Objekte bei Bedarf ladet und Eingaben nicht selbst verarbeitet, sondern diese an das zur Benutzerschnittstelle gehörende Kommunikationsobjekt weiterleitet. Die Kommunikation erfolgt hierbei auf der UI-Schicht. Ein Internet-Browser, der HTML-Seiten über HTTP/GET ladet und Benutzereingaben über HTTP/POST sendet, ist ein Beispiel für dieses Konzept. Ein weiteres Beispiel dafür ist HAVi Data Driven Interaction (DDI, HAVi User Interface Level 1).

3.2 Einstufung der Gerätenetzwerke

Ausgangsbasis einer Kopplung von Gerätenetzwerken bildet eine Untersuchung der einzelnen Netze hinsichtlich der Realisierung der im Abschnitt 3.1 (Gerätearchitektur) beschriebenen grundlegenden Mechanismen der Gerätevernetzung und des Transports. Eine Kopplung von Gerätenetzwerken erfordert die Vermittlung zwischen den unterschiedlichen Realisierungsarten. In diesem Kapitel werden hierfür einige Gerätenetzwerke der drei Bereiche „Heim-Automatisierung“, „Unterhaltungselektronik“ und „Small Office/Home Office“ entsprechend der Kriterien Transport, Addressing, Transmission, Interaction, Control, Notification und User Interface eingestuft. So können losgelöst vom spezifischen Anwendungsumfeld des einzelnen Standards dessen wesentlichen Eigenschaften und Funktionsweisen erfasst werden. Eine weitere Vorgabe für die Einstufung ist, dass sie nicht länger als eine Seite umfassen soll, um eine Beschränkung auf die wesentlichen Punkte zu erreichen. Basierend auf der Untersuchung der unterschiedlichen Vernetzungsstandards anhand der selben Kriterien kann eine direkte Gegenüberstellung erfolgen, anhand welcher der notwendige Aufwand zur Kopplung der einzelnen Netzwerke abgeleitet werden kann.

⁶HAVi Full Audio/Video (FAV) Geräte mit „User Interface Level 2 (UI-L2)“-Unterstützung

3.2.1 Heim-Automatisierung (HA)

Als zwei typische Vertreter für Standards der Heim-Automatisierung werden in Tabelle 3.1 das Local Operating Network (LON) und in Tabelle 3.2 der Europäische Installations Bus (EIB) anhand der oben beschriebenen Kriterien untersucht.

3.2.2 Unterhaltungselektronik (CE)

Zur Einstufung von Gerätenetzwerken der Unterhaltungselektronik wurde das bereits weit verbreitete⁷ Audio Video Control (AV/C) Protokoll und der wesentlich mächtigere, aber erst am Beginn seiner Anwendung stehende Vernetzungsstandard Home Audio Video interoperability (HAVi) gewählt. Tabelle 3.3 behandelt AV/C und Tabelle 3.4 HAVi.

3.2.3 Small Office/Home Office (PT)

Für die spontane Vernetzung des Bereichs „Small Office/Home Office“ wurden die beiden Spezifikationen Universal Plug and Play (UPnP) und Jini berücksichtigt. Die Einstufung von UPnP zeigt Tabelle 3.5, jene von Jini ist in Tabelle 3.6 dargestellt.

3.3 Kopplungsmethoden

Eine Kopplung von Gerätenetzwerken erfolgt durch Umsetzung auf einer der Schichten der in Abschnitt 3.1 eingeführten Gerätearchitektur, nach welcher auch das entsprechende Gateway benannt wird: System Gateway, Service Gateway und UI Gateway. Daneben gibt es noch das so genannte Multihomed Non Transforming Device, das Teilnehmer zweier oder mehrerer Gerätenetzwerke ist, jedoch keine Umsetzung zwischen diesen ermöglicht.

3.3.1 Multihomed Non Transforming Device

Die Kopplung von Gerätenetzwerken ermöglicht Anwendungen von Geräten eines Netzwerks die Benutzung von Diensten, die von Geräten eines anderen Netzwerks bereitgestellt werden. Dieser Ansatz darf jedoch nicht mit einem Ansatz verwechselt werden, bei welchem ein Gerät gleichzeitig Teilnehmer an zwei oder mehreren Netzwerken ist und lokale Anwendungen die Dienste der Geräte mehrerer direkt angeschlossener Netzwerke nutzen. Dieser Gerätetyp wird im Folgenden als *Multihomed Non Transforming Device* bezeichnet und ist in Abbildung 3.3 dargestellt.

⁷siehe 1394 Trade Association (<http://www.1394ta.org>)

Aufgabe	Realisierung
Transport	Unterstützung unterschiedlicher Übertragungsmedien wie Twisted-Pair, Koaxialkabel, Powerline, Funk und Lichtwellenleiter durch entsprechende Transceiver. Übertragungsraten bis zu 1,25 Mbit/s.
Addressing	Jeder LON-Knoten besitzt eine fest zugewiesene, weltweit eindeutige 48-Bit-Adresse (Neuron-ID). Für den normalen Betrieb werden zur Adressierung der Knoten logische Adressen basierend auf einer hierarchischen Adressstruktur (Domain, Subnet, Node, Group und Member) verwendet, deren Zuweisung erfolgt manuell im Rahmen der Geräteinstallation. Zur Adressierung der Netzwerkvariablen eines Knotens (Unteradressierung) dient ein 14-Bit langer Netzwerkvariablen-Selektor.
Transmission und Interaction	Die Kommunikation erfolgt durch Netzwerkvariablen (NV) oder Explicit Messages. NV stellen logische Verbindungen zwischen den Anwendungen der LON-Knoten dar und müssen dazu im Rahmen der Installation verbunden werden. Weist die Anwendung einer Netzwerkvariablen einen neuen Wert zu, so werden die verbundenen Netzwerkvariablen durch das Senden einer einfachen Mitteilung automatisch aktualisiert. Explicit Message bildet die zweite Art der Kommunikation, die neben einfachen Mitteilungen auch den Anfrage-Antwort-Mechanismus unterstützt. Hierbei muss die Anwendung selbst die Nachricht bilden und das Senden und gegebenenfalls auch das Empfangen der Antwort veranlassen.
Discovery	Manuelle Bekanntmachung durch ein Netzwerk-Management-Werkzeug: Verbinden von Netzwerkvariablen und Zuweisung der Zieladresse bei gebundenen Explicit Messages. Unterstützung der Konfiguration durch Service-Pin-Nachricht.
Description	Selbstbeschreibung durch Self-Documentation (SD) und Self-Identification (SI) Zeichenketten im Gerätespeicher oder über ein extern bereitgestelltes Device Interface File (XIF).
Control und Notification	Gerätefunktionen werden über ladbare, ereignisgesteuerte Anwendungen in den LON-Knoten bereitgestellt. Für Control und Notification werden vorwiegend NV eingesetzt, die Kommunikation ist gebunden. Die einheitliche Interpretation der Daten einer Nachricht beruht auf standardisierten Typen für NV (SNVT) und Konfigurationsparameter (SCPT). Gerätefunktionen sind in öffentlich verfügbaren Functional Profiles durch die Festlegung von vorgeschriebenen und optionalen NV und Konfigurationsparametern, deren Typen (SNVTs, SCPTs) und deren Funktion standardisiert.
User Interface	Spezielle und einfach zu bedienende Hardware, wie z. B. Schalter, Dimmer, Taster oder einfache LCD Displays mit Knöpfen.

Tabelle 3.1: Local Operating Network (LON)

Aufgabe	Realisierung
Transport	Als Übertragungsmedien werden Twisted-Pair (9.600 bit/s) und Powerline (1.200 bit/s) unterstützt, zusätzlich werden Transceiver für Funk und Infrarot definiert.
Addressing	Jeder Knoten erhält bei der Installation eine eindeutige, strukturierte (ortsabhängige) Adresse. Zur gemeinsamen Adressierung von mehreren Knoten werden Gruppenadressen verwendet. Kommunikationsobjekte sind einer oder mehreren Gruppenadressen zugeordnet, wobei das Senden auf eine Gruppenadresse beschränkt ist. Die Adressierung von EIB-Objekten erfolgt über die eindeutige Adresse des Knotens sowie über eine Objekt- und Property-Id.
Transmission und Interaction	Die Kommunikation ist daten- und nicht befehlsorientiert, Anwendungen tauschen Daten über Objekte aus. Die Objekte werden über einen User-Layer (Objekt-Server) bereitgestellt, der Zugriffe auf diese automatisch behandelt. Kommunikationsobjekte enthalten bis zu 14 Byte große Daten und sind an zumindest eine Gruppenadresse gebunden, der Zugriff erfolgt über Multicast-Nachrichten. EIB-Objekte bestehen aus einem oder mehreren Datenarrays (Properties), auf die über Unicast-Nachrichten zugegriffen wird.
Discovery	Manuelle Bekanntmachung durch die EIB Tools Software (ETS): Zuweisung der eindeutigen Geräteadresse (Broadcast-Nachricht und Programmier Taste am Zielknoten gedrückt oder Seriennummer bekannt), Verbinden von Kommunikationsobjekten durch Zuweisung von gemeinsamen Gruppenadressen.
Description	Die Produktdatenbank der ETS enthält die wesentlichen Informationen für jedes zertifizierte EIB-Gerät. Selbstbeschreibung: Systemobjekte enthalten grundlegende Geräteinformationen (Seriennummer, Herstellerkennung, ...), zu jedem Objekt werden Konfigurationsinformationen (Zugriff, Datentyp, ...) gespeichert.
Control und Notification	Ein EIB-Knoten besteht aus einem Kommunikationsmodul, auf das verschiedene Anwendungsmodule über eine Anwendungsschnittstelle (AST) aufgesteckt werden können. Gerätefunktionen werden über eine ladbare, interne Anwendung, die entweder proprietär oder über die AST auf die spezielle Geräte-HW/SW zugreift, bereitgestellt. Über die AST kann auch eine externe Anwendung angebunden werden. Die Verbindung der Gerätefunktionen erfolgt zumeist über Kommunikationsobjekte, die an eine gemeinsame Gruppenadresse gebunden sind. Zur standardisierten Interpretation der Daten wurden für Kommunikationsobjekte die EIB Interworking Standards (EIS) und für EIB-Objekte die Object IS (ObIS) definiert.
User Interface	Spezielle und einfach zu bedienende Hardware, wie z. B. Schalter, Dimmer, Taster oder einfache LCD Displays mit Knöpfen.

Tabelle 3.2: European Installation Bus (EIB)

Aufgabe	Realisierung
Transport	Serieller Bus IEEE 1394 („FireWire“). Automatische Konfiguration aller Geräte am Bus (Hot-Plug-and-Play). Übertragungsraten bis zu 1600 Mbit/s (IEEE 1394b).
Addressing	Jeder IEEE 1394-Knoten besitzt eine fest zugewiesene, weltweit eindeutige 64-Bit-Adresse (GUID). Zur Übertragung von Paketen über den Bus wird die automatisch zugewiesene, 16-Bit große NodeID verwendet, die sich allerdings bei jeder Änderung am Bus (Bus-Reset) durch die automatische Neuvergabe ändern kann. Die Unteradressierung der einzelnen Gerätefunktionen erfolgt über den Subunit-Type und die Subunit-Id.
Transmission und Interaction	Die Kommunikation basiert auf dem Anfrage-Antwort-Mechanismus und erfolgt über das Function Control Protocol (FCP). Hierbei werden Anfragen in das FCP-Befehlsregister und Antworten in das FCP-Antwortregister der entsprechenden Knoten geschrieben.
Discovery	Eine Änderung am IEEE 1394-Bus wird durch einen so genannten Bus-Reset (BR) angezeigt. Nach einem BR wird dieser automatisch neu konfiguriert, d. h. jeder Knoten erhält eine NodeID und sendet Basisinformationen. Dadurch sind alle Knoten am Bus identifiziert und hinzugefügte oder entfernte Geräte können erkannt werden.
Description	Selbstbeschreibung: IEEE 1394-ConfigRom (Geräteinformationen); AV/C-Befehle von Units oder Subunits, wie z. B. die Unit-Befehle UNIT INFO und SUBUNIT INFO (Anzahl und Typ der vorhandenen Subunits, Hersteller-ID) oder der „Descriptor and Info Block Mechanism“, der Datenstrukturen und Befehle zum Erstellen, Lesen und Schreiben dieser Datenstrukturen definiert. Die grundsätzlichen Fähigkeiten einer Subunit werden durch den Subunit-Typ (Monitor, Audio, Disc, Tuner, ...) festgelegt.
Control	Geräte werden durch Units und ihre funktionellen Komponenten durch Subunits abstrahiert. Für Geräte der Unterhaltungselektronik ist eine umfassende API in Form von standardisierten Befehl- und Antwort-Paketen an Units oder Subunits definiert worden. Sie setzt sich neben einem allgemeinen Dokument aus einer Unmenge an einzelnen Subunit-spezifischen Dokumenten zusammen.
Notification	Anfrage mit verzögerter Antwort, wobei diese Art der direkten Benachrichtigung nur eine „one-shot“-Operation ist und nach einer Antwort oder einem BR die Anfrage erneut gesendet werden muss (es gibt keine Abonnentenverwaltung).
User Interface	Anwendungen, welche spezielle GUI-Eigenschaften eines Gerätes nutzen; AV/C-Panel-Subunit, welche die standardisierte Bereitstellung von GUI-Informationen und deren Wiedergabe auf UI-Controllern definiert.

Tabelle 3.3: Audio Video Control (AV/C)

Aufgabe	Realisierung
Transport	Serieller Bus IEEE 1394 („FireWire“). Automatische Konfiguration aller Geräte am Bus (Hot-Plug-and-Play). Übertragungsraten bis zu 1600 Mbit/s (IEEE 1394b).
Addressing	Jeder IEEE 1394-Knoten besitzt eine fest zugewiesene, weltweit eindeutige 64-Bit-Adresse (GUID). Zur Übertragung von Paketen über den Bus wird die automatisch zugewiesene 16-Bit große NodeID verwendet. Die Adressierung der Software Elemente eines Gerätes erfolgt durch den Software Element Identifier (SEID), der aus der GUID des Gerätes und einem lokalen „swHandle“ (Unteradressierung) gebildet wird. Die Umsetzung zwischen der statischen GUID und der dynamischen NodeID erfolgt durch das HAVi-Systemmodul Communication Media Manager 1394 (Cmm1394).
Transmission und Interaction	Die Kommunikation basiert auf Software-Elementen (SE), die Nachrichten über das HAVi Messaging System (MS) austauschen. Das MS basiert auf dem Function Control Protocol und unterstützt neben der Fragmentierung und bestätigten Übertragung von Nachrichten den Anfrage-Antwort-Mechanismus zum entfernten Aufruf von Funktionen.
Discovery	Das Hinzufügen und Entfernen von Geräten wird vom Cmm1394 durch die Unterstützung des IEEE 1394-Busses erkannt und über den Event Manager (EM) berichtet. Die Bekanntmachung der SE erfolgt indirekt über die Registry als Registrierungsdiens und den EM zur Verteilung der Bekanntmachung.
Description	Selbstbeschreibung: IEEE 1394-ConfigRom (Geräteinformationen), HAVi-Registry (verteilte Datenbank, die Informationen über SE enthält und Suchfunktionen bietet). Die grundsätzlichen Fähigkeiten eines SE werden durch den SE-Typ (Tuner, Camera, Amplifier, Display, Application, ...) festgelegt. Da sich der SEID nach einem Neustart ändern kann, gibt es einen eigenen dauerhaft eindeutigen Kennzeichner (HUID).
Control	Für Geräte der Unterhaltungselektronik enthält die HAVi-Spezifikation eine umfassende API. Geräte werden durch Device Control Modules (DCM) und ihre funktionellen Komponenten durch Functional Component Modules (FCM) repräsentiert. Anwendungen, DCMs und FCMs sind SE und kommunizieren über Nachrichten (MS).
Notification	Indirekte Benachrichtigung mit dem Event Manager als Verteilungsdienst. Direkte Benachrichtigung durch SE, wie z. B. die FCM-Notification.
User Interface	Anwendungen, welche spezielle GUI-Eigenschaften eines Gerätes nutzen; spezielle GUI-Konzepte (Havlets, DDI), welche auf standardisierten GUI-Fähigkeiten der Geräte basieren.

Tabelle 3.4: Home Audio/Video interoperability (HAVi)

Aufgabe	Realisierung
Transport	IP basierende Netze. Aufgrund der vielseitigen Einsetzbarkeit von IP steht eine große Anzahl von Übertragungssystemen zur Verfügung: Ethernet, Fast-Ethernet, WLAN, ...
Addressing	Jedes Gerät benötigt eine IP-Adresse, die über Auto-IP oder DHCP zugewiesen wird. Zur Unteradressierung dient eine Portnummer und darüber hinaus ein lokaler Pfad. Die gesamte Adresse wird als URL (<code>http://host:port/path</code>) angegeben.
Transmission und Interaction	Die Kommunikation basiert auf HTTP/TCP/IP. Zusätzlich wird zur Geräte-Bekanntmachung und -Suche eine Unicast- und Multicast-Variante von HTTP (HTTP-U, HTTP-MU) über UDP/IP eingesetzt. Die Interaktion basiert auf dem Anfrage-Antwort-Mechanismus (HTTP-Request/-Response).
Discovery	Direkte Bekanntmachung durch periodisches Senden von Multicast-Alive-Nachrichten bzw. Senden von Multicast-Byebye-Nachrichten bei Beendigung. Nicht mehr vorhandene Kommunikationsobjekte werden außerdem durch das Ausbleiben der regelmäßigen Alive-Nachrichten erkannt.
Description	Selbstbeschreibung: Beschreibung der Geräteeigenschaften und Funktionen in Form von XML-Dokumenten (Device Description, Service Description), deren Aufbau und zwingender Inhalt gemäß XML-Schema festgelegt und die jeweils über eine URL via HTTP verfügbar sind. Die URL der Gerätebeschreibung wird im Rahmen der direkten Bekanntmachung verlautbart oder ist in der Antwort auf eine Multicast-Query enthalten. Die URL der Dienstbeschreibungen (API) sowie die URL für Control, Notification und Presentation stehen in der zugehörigen Gerätebeschreibung. Als dauerhaft eindeutiger Kennzeichner dient der Geräteiname.
Control	Ein physikalisches Gerät wird durch ein oder mehrere „root devices“ abstrahiert, welche aus Diensten und eingebetteten Geräten bestehen. Die Geräteschnittstellen sind für verschiedene Gerätetypen (Internet Gateway, Media Server/Renderer, Scanner, ...) in öffentlich verfügbaren Device-/Service-Templates standardisiert. Für das entfernte Aufrufen von Funktionen der Dienste eines Gerätes wird das auf HTTP basierende Simple Object Access Protocol verwendet.
Notification	Direkte Benachrichtigung durch Dienste anhand von Zustandsvariablen, die den aktuellen Zustand eines Gerätes repräsentieren. Die Abonnieung und Benachrichtigung kann nur pauschal für alle Zustandsvariablen eines Dienstes erfolgen und basiert auf HTTP mit erweiterten HTTP-Methoden.
User Interface	Anwendungen, welche spezielle GUI-Eigenschaften eines Gerätes nutzen oder auf HTML basieren (Browser).

Tabelle 3.5: Universal Plug and Play (UPnP)

Aufgabe	Realisierung
Transport	IP basierende Netze. Aufgrund der vielseitigen Einsetzbarkeit von IP steht eine große Anzahl von Übertragungssystemen zur Verfügung: Ethernet, Fast-Ethernet, WLAN, ...
Addressing	Jedes Gerät benötigt eine IP-Adresse, die über Auto-IP oder DHCP zugewiesen wird. Zur Unteradressierung dient eine Portnummer. Die Adressierung erfolgt jedoch zumeist implizit, d. h. ein entferntes Objekt wird über ein lokales Objekt (Proxy) angesprochen. Das Netzwerk kann in einzelne Gruppen (Djinn) unterteilt werden, die aus einem oder mehreren Lookup-Diensten und den dort registrierten Diensten bestehen und einen eindeutigen Namen besitzen.
Transmission und Interaction	Die grundlegende Kommunikation basiert auf TCP/IP und UDP/IP. Jini stellt im Wesentlichen eine Erweiterung einer einzelnen JVM auf ein Netzwerk von JVMs dar. Die Nutzung der Java Objekte erfolgt jeweils lokal. Dazu werden für entfernte Objekte lokale Stellvertreter (Proxys) zur Verfügung gestellt, welche für die Kommunikation mit dem entfernten Objekt verantwortlich sind. Die Kommunikation zwischen Proxy und dem entfernten Objekt ist verborgen, beruht jedoch typischerweise auf Remote Method Invocation (RMI).
Discovery	Indirekte Bekanntmachung durch Registrierung bei einem oder mehreren, jeweils unabhängigen, zentralen Registrierungsdiensten (Lookup-Service, LUS). Zur Erkennung von nicht mehr vorhandenen Objekten müssen die Einträge regelmäßig erneuert werden. Die Ermittlung der LUS erfolgt über direkte Bekanntmachung, Suchanfragen oder manuell über vorkonfigurierte Einstellungen.
Description	Selbstbeschreibung: Beschreibung der Geräteeigenschaften und -funktionen durch Attribute und die zugehörigen Proxies als Einträge am LUS, der die Suche nach Einträgen unterstützt. Als dauerhaft eindeutiger Kennzeichner eines Dienstes dient die ServiceId.
Control	Geräte und ihre Funktionen werden durch Dienste repräsentiert. Die Nutzung eines Dienstes erfolgt über einen Dienst-Proxy, der die selbe Schnittstelle wie der zugehörige Dienst bereitstellt und soweit der Proxy die geforderte Funktionalität nicht bereits selbst bietet, die Anfragen an den entfernten Dienst weiterleitet. Vor Nutzung eines Dienstes ist jeweils der entsprechende Proxy vom LUS zu laden. Abgesehen von den eigentlichen Jini-Diensten sind im Wesentlichen keine standardisierten Geräteschnittstellen verfügbar.
Notification	Direkte oder indirekte Benachrichtigung basierend auf RMI. Der Empfang von Benachr. ist über die Methode <code>notify</code> festgelegt, die Registrierung für Benachr. ist nicht standardisiert.
User Interface	Java-Anwendungen, die teilw. spezielle GUI-Pakete voraussetzen.

Tabelle 3.6: Jini

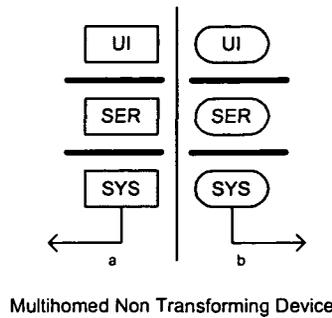


Abbildung 3.3: Multihomed Non Transforming Device

Ein digitales Fernsehgerät, welches einen HAVi-Stack zur Anbindung von Unterhaltungselektronikgeräten, ein EIB-Modul zur Anbindung von Geräten der Heim-Automatisierung und eine gemeinsame Benutzerschnittstelle zur Benutzung der Dienste der Geräte beider Netzwerke besitzt, ist ein Beispiel für einen solchen Gerätetyp (siehe Abbildung 3.4)⁸. Die Benutzung von Geräten unterschiedlicher Netzwerke ist allerdings auf das mehrfach angebundene Gerät beschränkt. Im Gegensatz dazu ist der wesentliche Punkt bei der Kopplung von Gerätenetzwerken, dass Dienste von Geräten entfernter Netzwerke wie Dienste von Geräten des eigenen, lokalen Netzwerks verwendet werden können. Ein hierin als Gateway bezeichnetes Gerät zur Kopplung von Gerätenetzwerken führt dazu eine Umsetzung auf einer der Schichten der Gerätearchitektur durch. Ein Multihomed Non Transforming Device führt keine Umsetzung durch und wird deshalb nicht als Gateway bezeichnet.

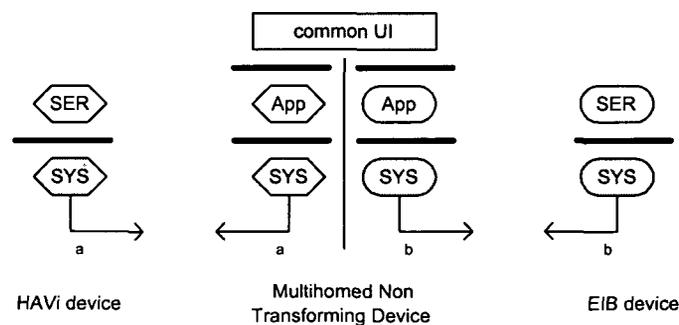


Abbildung 3.4: Multihomed Non Transforming Device - Beispiel

⁸Einen ersten Schritt in diese Richtung hat der deutsche High-End TV-Gerätehersteller Loewe gemacht, indem er ein „Wireless Home Automation Starter Set“ zur Steuerung von Geräten der Heim-Automatisierung über die normale TV-Fernbedienung und das On-Screen-Display anbietet. Als Heim-Automatisierungs-Netzwerk wird ein Funk-Bus der Firma Gira eingesetzt und das TV-Gerät mit einem entsprechenden Funk-Sendemodul ausgerüstet [Loe03].

3.3.2 System Gateway

Ein *System Gateway* ist ein Gerät, das eine Kopplung von zwei Netzwerken durch Umsetzung der System-Schicht der in Kapitel 3.1 vorgestellten Gerätearchitektur durchführt. Beispielsweise können zwei System-Gateways zur Kopplung von zwei entfernten HAVi-Netzwerken über einen TCP/IP-Tunnel dienen. Abbildung 3.5 zeigt ein einzelnes System-Gateway und das angesprochene Anwendungsbeispiel mit zwei System-Gateways.

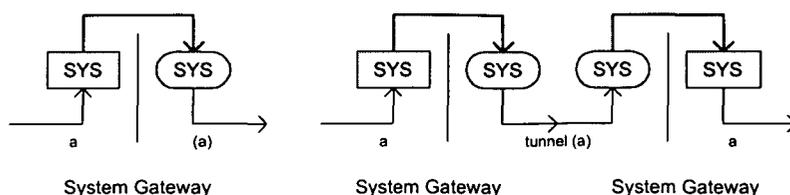


Abbildung 3.5: System Gateway

Sofern vom Transport der System-Schicht die gleichen Dienste angeboten werden, können auch Netze mit unterschiedlichem Transport ohne System-Gateway gekoppelt werden. Beispiele hierfür sind die Verwendung von Wireless-1394-Verbindungen über HyperLan2 basierte transparente Brücken [BPS⁺02, Bra02] an einem IEEE 1394-Bus oder der Mischbetrieb eines IEEE 1394-Busses von IEEE 1394a TP-Medien und IEEE 1394b POF-Medien, die an einem IEEE 1394b-Knoten über so genannte IEEE 1394b-*Bilingual-Plugs* verbunden sind [ICT04]. Weitere Beispiele sind die Verbindungen von unterschiedlichen IEEE 802.3 [IEE02b] (CSMA/CD LANs) oder IEEE 802.11 [IEE99a] (Wireless LANs) Netzen über transparente Brücken gemäß IEEE 802.1D [IEE98] (Media Access Control Bridges). Die MAC-Brücken arbeiten unterhalb der Grenze der 802.3-MAC-Dienst-Schicht und sind transparent⁹ in Bezug auf Protokolle, die oberhalb dieser Grenze arbeiten, wie beispielsweise IP.

Auch bei der einheitlichen Verwendung eines anderen als durch den betreffenden Netzungsstandard spezifizierten Transports oder Protokolls der System-Schicht ist unter der Voraussetzung der Einhaltung der jeweiligen Dienste kein System-Gateway nötig. Die System-Schicht ist jedoch in einigen Fällen eng mit dem Transport verbunden, d. h. hierbei werden spezielle Eigenschaften des Transports genutzt, die von einem anderen Transport nur umständlich zu emulieren sind. Ein typisches Beispiel hierfür ist IEEE 1394 als Transport für HAVi oder AV/C [Anh02] (siehe auch Tabelle 3.3 und 3.4).

Die Kopplung auf der System-Schicht wird vor allem zur Überwindung von Beschränkungen eines Gerätenetzwerks hinsichtlich der Netzwerkausdehnung, der Geräteanzahl oder des Transportmediums oder auch zur Erweiterung eines Gerätenetzwerks auf neue

⁹In Bezug auf QoS sind die MAC-Brücken nicht vollständig transparent.

Anwendungsgebiete eingesetzt. Die Verwendung des im Automobilbereich verbreiteten Vernetzungsstandards MOST (*Media Oriented Systems Transport*) anstelle von IEEE 1394 zur Vernetzung von HAVi FAV/IAV-Geräten ist ein Beispiel für die Erweiterung von HAVi zur Erschließung neuer Anwendungsgebiete¹⁰ (HAVi over MOST [DR02]). Wird gleichzeitig auch ein Anschluss von IEEE 1394-basierten HAVi-Geräten gefordert, z. B. zum Anschluss portabler AV-Geräte an das MOST-basierte HAVi-Netz, so ist dafür ein System-Gateway nötig. Beispiele zur Überwindung der beschränkten Ausdehnung des IEEE 1394-basierten HAVi-Netzwerks sind die in Abschnitt 4.2.1 beschriebene transparente Kopplung von HAVi-Netzwerken über einen TCP/IP-Tunnel oder die in [WUB01] beschriebene Anbindung von IP-basierten Geräten an das HAVi-Netzwerk zur entfernten Ausführung von HAVi-Anwendungen. Letzteres Beispiel ist in Abbildung 3.6 dargestellt.

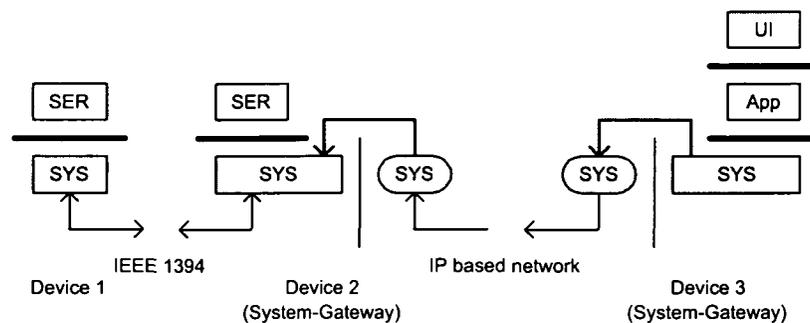


Abbildung 3.6: Entfernte Ausführung von HAVi-Anwendungen auf IP-Geräten

Hierbei kann eine HAVi-Anwendung auf einem IP-basierten Gerät (Device-3) über die beiden System-Gateways die Dienste der Geräte des HAVi-Netzwerks (Device-1, Device-2) nutzen. Die System-Gateways sorgen dabei für die wechselseitige Umsetzung zwischen der HAVi System-Schicht und der auf verbreiteten Internetprotokollen (SOAP, HTTP, TCP/IP) basierten System-Schicht zur Anbindung von IP-basierten Geräten¹¹.

Ein weiteres Beispiel für den Einsatz von paarweisen System-Gateways ist die Kopplung von entfernten UPnP-Netzen über einen TCP/IP-Tunnel. Obwohl UPnP auf IP basiert ist die Ausdehnung des UPnP-Netzwerks beschränkt. Der Grund dafür liegt in der Verwendung von Multicast-Nachrichten für das Discovery. Multicast-Nachrichten werden von IP-Routern, soweit diese nicht extra dafür konfiguriert wurden, nicht weitergeleitet. Die Weiterleitung der Bekanntmachungen von einem lokalen zu einem entfernten UPnP-Netz könnte über zwei System-Gateways und einen dazwischen liegenden TCP/IP-Tunnel erfolgen. Das eine System-Gateway empfängt dabei die Bekanntmachungen und leitet diese über den Tunnel an das System-Gateway des entfernten UPnP-

¹⁰Ogleich sich IEEE 1394 mit Plastic Optical Fibre (POF, IEEE 1394b) als Transportmedium hervorragend für den Einsatz im Auto eignet, kann die Verwendung von MOST aufgrund dessen Verbreitung und der damit verbundenen einfacheren Integration bestehender Geräte sinnvoll sein.

¹¹Im Gegensatz zu der in Abschnitt 4.2.1 vorgestellten Kopplung ist diese Kopplung auf das entfernte Ausführen von HAVi-Anwendungen beschränkt, eine Nutzung von Diensten der entfernten Geräte (Device-3 und an Device-3 direkt angeschlossene Geräte) durch das lokale HAVi-Netzwerk (Device-1 und Device-2) ist nicht vorgesehen.

Netzes weiter. Letzteres sorgt für die Bekanntmachung im entfernten Netz anhand der über den Tunnel empfangenen Daten. Da die restliche UPnP-Kommunikation auf TCP basiert, ist die Verwendung von Geräte-Proxys nicht nötig, soweit die UPnP-Geräte routbare IP-Adressen verwenden. Das heißt, die IP-Adressen der Geräte dürfen weder aus dem privaten Adressbereichen [RMK⁺96] stammen noch zu den von Auto-IP verwendeten Link-Local-Adressen [Chi02] gehören.

3.3.3 Service Gateway

Geräte, die eine Kopplung von zwei Netzwerken durch Umsetzung der Service-Schicht der in Kapitel 3.1 vorgestellten Gerätearchitektur durchführen, werden als *Service Gateway* bezeichnet. Dieser Typ ist in Abbildung 3.7 dargestellt und repräsentiert den wichtigsten Typ, weil eine Umsetzung der Service-Schicht die Ansteuerung von Geräten über eine andere API als die ursprüngliche API ermöglicht. So können die Dienste von Geräten eines anderen Netzwerks, wie die Dienste von Geräten des eigenen Netzwerks, genutzt werden. Viele Gateway-Implementierungen sind von diesem Typ. Einerseits ermöglichen sie die Einbeziehung von Geräten anderer Bereiche, wie beispielsweise die Anbindung der Feldbusse LON und EIB an Netze der Unterhaltungselektronik (HAVi-EIB-Gateway [Abschnitt 4.1], HAVi-LON-Gateway [Abschnitt 4.1]) oder des Bereichs Small Office/Home Office (Jini-LON-Gateway [KL01], UPnP-EIB-Gateway [WG00]) oder die Kopplung der Bereiche Unterhaltungselektronik und Small Office/Home Office (UPnP-HAVi-Gateway [Abschnitt 4.3]). Andererseits finden sie auch Anwendung zur Kopplung von unterschiedlichen Netzwerken des gleichen Bereichs, wie beispielsweise das Jini-UPnP-Gateway [ACGI03] oder HAVi-AV/C-Gateway [Abschnitt 4.2.2] zeigt.

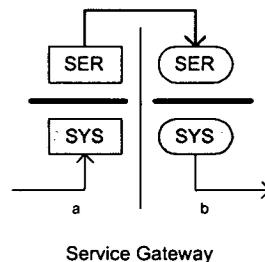


Abbildung 3.7: Service Gateway

Zur Umsetzung der unterschiedlichen Service-Schichten ist eine Kommunikation zwischen diesen Schichten nötig. Da die Service-Schichten die jeweils eigenen System-Schichten verwenden, erfolgt keine direkte Umsetzung der System-Schicht. Zu den typischen Aufgaben eines Service-Gateways gehören die:

- (1) Abbildung der oft sehr unterschiedlichen Geräte- bzw. Funktionsabstraktionen. HAVi repräsentiert Geräte und deren Funktionen durch Device Control Modules (DCM) und Functional Component Modules (FCM), bei LON sind Gerätefunktionen über Netzwerkvariablen und bei EIB über Gruppenadressen nutzbar. Bei der in Abschnitt 4.1 beschriebenen Realisierung einer Anbindung der Feldbusse LON und EIB an HAVi wird der gesamte Feldbus einer DCM und alle Netzwerkvariablen bzw. Gruppenadressen, die gemeinsam einen Dienst (Licht, Heizung, Waschmaschine, ...) bilden, einer FCM zugeordnet. Ein anderes Beispiel ist die Zuordnung von AV/C-Units zu HAVi-DCMs und AV/C-Subunits zu HAVi-FCMs (siehe Abschnitt 4.2.2).
- (2) Abbildung der unterschiedlichen Anwendungsprogramm-Schnittstellen (API-Mapping). Soweit beide Seiten über eine entsprechende API verfügen, kann eine $n : m$ -Umsetzung der vorhandenen Funktionen erfolgen. So ist etwa die API zur Ansteuerung eines Tuners sowohl bei HAVi (Tuner-FCM) als auch bei AV/C (Tuner-Subunit) definiert. Die Auswahl eines Fernseh- oder Radioprogramms bei einem AV/C-Tuner durch eine HAVi-Anwendung erfordert im Wesentlichen die Umsetzung der aufgerufenen HAVi-Funktion `Tuner::SelectService` auf die AV/C-Funktion `TunerSubunit::DirectSelectInformationType`. Oft ist die Umsetzung nicht 1 : 1 möglich, weil die Anwendungsprogramm-Schnittstellen sich hinsichtlich der Anzahl der Funktionen und ihrer Parameter unterscheiden oder bestimmte für die Ausführung einer Funktion notwendige Informationen nicht im Gateway gespeichert werden und zuerst vom Gerät gelesen werden müssen.

Während bei der Kopplung von Gerätenetzwerken des selben Bereiches häufig vergleichbare APIs verfügbar sind, fehlt bei der Kopplung von unterschiedlichen Bereichen zumeist eine entsprechende API. Beispielsweise enthält die HAVi-Spezifikation keine API zur Ansteuerung von Geräten der Heim-Automatisierung. In diesen Fällen muss erst eine geeignete API definiert werden, wobei die übliche Nutzungsart der Dienste und Funktionen durch Anwendungen des „eigenen“ Netzwerks im Vordergrund stehen soll und nicht eine möglichst einfache Umsetzung der API des anderen Netzwerks. Zu der in Abschnitt 4.1 beschriebenen Anbindung von Geräten der Heimautomatisierung an ein HAVi-Netzwerk wurde deshalb die *HAVi HA Extension* [Zie03a] entwickelt.

Steht bloß die Ausführung einer einfachen Benutzerschnittstelle zur Ansteuerung von Geräten eines anderen Netzwerks im Vordergrund, so kann die API-Umsetzung auf eine einfache Schnittstelle zu einer Anwendung des anderen Netzwerks, welche nur die von der Benutzerschnittstelle benötigten Funktionen bereitstellt, beschränkt werden. Diese über das Service-Gateway im lokalen Netz bereitgestellten Funktionen können von der Benutzerschnittstelle eines Gerätes über dessen Service-Schicht-Anwendung genutzt werden, um nicht lokale Geräte anzusteuern. Ein Beispiel dafür ist das als „Jini-HAVi-Bridge“ [Ski02, Rah01] bezeichnete Service-Gateway zur Ansteuerung von HAVi-Geräten durch die entfernte Ausführung einer einfachen Benutzerschnittstelle auf Jini-Geräten. Die Abstim-

mung des Funktionsumfangs der Anwendung auf die von der Benutzerschnittstelle benötigten Funktionen verhindert jedoch eine allgemeine Einsetzbarkeit dieser Kopplung. Wird der Funktionsumfang der bereitgestellten Funktionen erweitert, fällt der Vorteil der einfachen Anbindung weg. Ein weiterer Nachteil dieser Kopplung ist, dass sie nicht transparent ist, weil die entfernten Geräte nur über die speziellen Funktionen der Anwendung und nicht wie lokale Geräte genutzt werden können. Soll eine Kopplung möglichst allgemein eingesetzt werden, muss deshalb unbedingt die übliche Nutzungsart der Dienste und Funktionen durch Anwendungen des lokalen Netzwerks im Vordergrund stehen und nicht eine möglichst einfache Umsetzung der API des anderen Netzwerks.

- (3) Kommunikation zwischen den unterschiedlichen Service-Schichten. Sie erfolgt lokal, innerhalb des Gateways über Mechanismen der Interprozesskommunikation, wie z. B. Named Pipes, Shared Memory oder Dateien. Wird das Gateway als einzelner Prozess realisiert, können auch direkte Funktionsaufrufe verwendet werden. Jedenfalls ist eine geeignete Schnittstelle zwischen den Service-Schichten bzw. auch ein Protokoll für den Datenaustausch zu definieren. Beispielsweise besteht das bereits mehrfach erwähnte HAVi-EIB-Gateway aus zwei Prozessen, die über zwei Named Pipes, eine für jede Kommunikationsrichtung, über ein eigenes Protokoll kommunizieren und beim Start eine gemeinsame Konfigurationsdatei auslesen. Im Gegensatz dazu vereint das HAVi-AV/C-Gateway beide Teile (HAVi-Stack, AV/C-Stack) in einem einzigen Prozess. Die Aufteilung des Gateways in unterschiedliche Prozesse beruht nicht immer auf einer rein technischen Entscheidung, sondern kann speziell aufgrund von Lizenzinkompatibilitäten der eingesetzten Software gefordert werden. Beispielsweise unterliegt der vom HAVi-EIB-Gateway verwendete EIB-Treiber der *General Public License* (GPL) [Fre91], die eine Einbindung (Verwendung in einem gemeinsamen Prozess) in eine nicht der GPL unterliegenden Software, wie dem eingesetzten HAVi-Stack, verbietet und damit eine Aufteilung in zwei Prozesse notwendig macht.
- (4) Ergänzung fehlender Mechanismen einer System-Schicht. Das Service-Gateway nutzt die jeweilige System-Schicht zur Ermittlung der Kommunikationspartner, zur Erlangung von Informationen über diese und zur Kommunikation mit diesen. Wenn jedoch die System-Schichten nicht vergleichbare Mechanismen unterstützen, dann müssen die fehlenden Informationen gesondert zur Verfügung gestellt werden. Am Beispiel des HAVi-EIB-Gateways stellt sich die Situation wie folgt dar: HAVi bietet mit der verteilten HAVi-Registry für alle SE die Möglichkeit zur umfangreichen Selbstbeschreibung und zur Suche nach SE. Im Gegensatz dazu ist die Selbstbeschreibung bei EIB sehr beschränkt und die Bekanntmachung der Kommunikationsteilnehmer erfolgt durch Zuweisung von Gruppenadressen im Rahmen der Netzwerkkonfiguration. Um der HAVi-Seite des Service-Gateways alle EIB-Geräte und deren Eigenschaften bekannt zu machen, werden diese Informationen in einer dem HAVi-Prozess bekannten Konfigurationsdatei bereitgestellt. Diese Datei wird bei der EIB-Netzwerkkonfiguration als Beiprodukt erstellt und vom HAVi-

Teil nach Aufforderung durch die EIB-Seite gelesen.

3.3.4 User Interface Gateway

Eine Kopplung von zwei Netzwerken durch Umsetzung der User Interface-Schicht der in Abschnitt 3.1 vorgestellten Gerätearchitektur wird mit einem so genannten *User Interface Gateway* durchgeführt. Dieser Typ ist in Abbildung 3.8 dargestellt. Ein Beispiel dafür ist das WAP-DDI-Gateway, das den Zugriff auf HAVi-Geräte über ein WAP-fähiges Gerät ermöglicht und in [NMV02, NMV03] beschrieben ist. Weitere Beispiele sind die in Abschnitt 4.3 beschriebenen Konzepte zur Kopplung von UPnP und HAVi über die UI-Schicht.

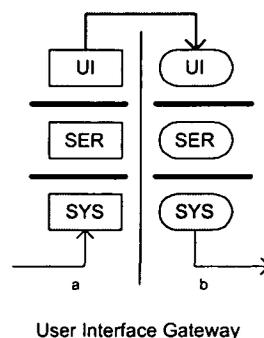


Abbildung 3.8: User Interface Gateway

Die unterschiedlichen UI-Konzepte können anhand der Antworten auf die drei, in Abschnitt 3.1.3 vorgestellten Fragen gruppiert werden. In Abbildung 3.9 ist diese Gruppierung dargestellt. Als erstes Abgrenzungsmerkmal dient die Art der Beschreibung der Benutzerschnittstelle. So wird unterschieden, ob sie standardisiert ist und damit von den zu ihrer Ausführung vorgesehenen Geräten standardisierte Ein-/Ausgabefähigkeiten gefordert werden, oder ob sie proprietär ist (vgl. Frage 1). In der zweiten Ebene wird unterschieden, ob die Benutzerschnittstelle einteilig oder zweiteilig ist. Eine Benutzerschnittstelle wird hierin als zweiteilig bezeichnet, wenn deren Beschreibung und Ausführung getrennt erfolgt (vgl. Frage 2). Die Unterteilung der dritten Ebene unterscheidet, wo die Eingaben der Benutzer verarbeitet werden (vgl. Frage 3).

Ein Beispiel für eine standardisierte, einteilige Benutzerschnittstelle mit lokaler Verarbeitung der Benutzereingaben sind HAVi Havlets. Havlets sind in Java geschriebene HAVi-Anwendungen, die zur Ausführung einer Benutzerschnittstelle die standardisierten HAVi UI-Klassen verwenden. Die Beschreibung der Benutzerschnittstelle ist im Programmcode enthalten und wird nicht von außen geladen. Die Eingaben der Benutzer werden durch das Havlet selbst verarbeitet, indem geeignete HAVi-Funktionen (Registry-Abfragen, DCM-/FCM-Befehle, ...) über das HAVi Messaging System aufgerufen werden. Im Gegensatz dazu gehören Anwendungen, die proprietär auf GUI-Fähigkeiten einer

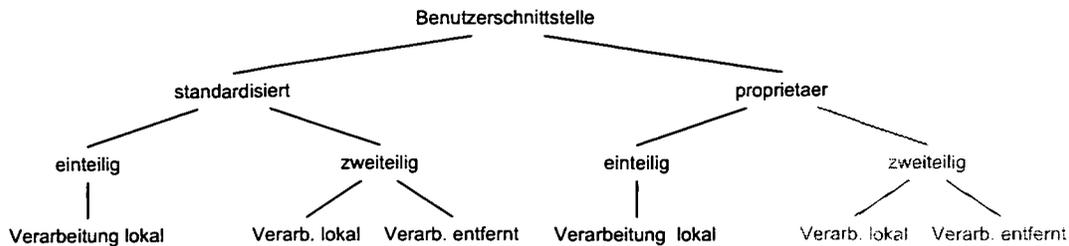


Abbildung 3.9: Benutzerschnittstellen - Merkmale

bestimmten Geräteplattform zurückgreifen, zur Kategorie der proprietären, einteiligen Benutzerschnittstellen mit lokaler Verarbeitung der Benutzereingaben.

HTML ist ein Beispiel für eine standardisierte, zweiteilige Benutzerschnittstelle. Ist im HTML-Code ausführbarer Programmcode (Java-Skripts, Visual-Basic-Skripts, ...) enthalten, so können die Benutzereingaben auch lokal verarbeitet werden. Werden hingegen die Eingaben der Benutzer an den Server weitergeleitet, dann erfolgt eine entfernte Verarbeitung im Server. HAVi-DDI ist ein typisches Beispiel für eine standardisierte (DDI-Elemente), zweiteilige (DDI-Controller ↔ DDI-Target) Benutzerschnittstelle mit entfernter Verarbeitung der Benutzereingaben (Eingabe am Controller → Verarbeitung im Target). Da die Aufteilung einer Benutzerschnittstelle in einen Beschreibungsteil und einen Teil zur Ausführung eine Kommunikation zwischen diesen Teilen erfordert, kann eine nicht standardisierte Benutzerschnittstelle nicht zweiteilig sein. In Abbildung 3.9 ist deshalb dieser Bereich nur blass dargestellt.

Eine Kopplung auf UI-Schicht ist nur bei zweiteiligen Benutzerschnittstellen möglich, weil nur bei diesen eine Kommunikation auf der UI-Schicht (z. B. HAVi DDI-Protokoll, HTML, ...) stattfindet. Eine einteilige Benutzerschnittstelle führt die Ansteuerung der Geräte nicht selbst, sondern mittels einer Anwendung auf der Service-Schicht durch. Um über die einteilige Benutzerschnittstelle Geräte anderer Netzwerke zu nutzen, ist deshalb ein Service-Gateway notwendig und nicht ein UI-Gateway. Im Gegensatz dazu können bei einer zweiteiligen Benutzerschnittstelle Geräte anderer Netzwerke durch Verwendung eines UI-Gateways zur Umsetzung der auf der UI-Schicht ablaufenden Kommunikation genutzt werden. So können beispielsweise UPnP-Geräte mit HTML-Browser HAVi-Geräte mit DDI-Target ansteuern, indem ein dazwischenliegendes UI-Gateway die Informationen aus den HAVi DDI-Targets in entsprechende HTML-Seiten umsetzt und über einen UI-Proxy im UPnP-Netzwerk zur Verfügung stellt [Abschnitt 4.3.1].

Es ist jedoch nicht zwingend notwendig, dass auf beiden Seiten des UI-Gateways die Ansteuerung der Geräte über die UI-Schicht erfolgt. Lediglich die Verbindung zum Benutzer muss über eine zweiteilige Benutzerschnittstelle erfolgen. Ein Beispiel dafür ist das zweite in Abschnitt 4.3 beschriebene UI-Gateway zur Kopplung von UPnP und HAVi. Hierbei werden die HAVi-Geräte vom UI-Gateway durch Aufruf geeigneter DCM- bzw. FCM-Funktionen über eine HAVi-Anwendung, also durch eine Kommunikation auf Service-Schicht, angesteuert, während auf der UPnP-Seite eine zweiteilige Benutzer-

schnittstelle verwendet wird [Abschnitt 4.3.2].

3.4 Kopplungsarten

In diesem Abschnitt werden verschiedene Arten von Kopplungen definiert. Kann zumindest ein Gerät eines Netzwerks A ein Gerät eines Netzwerks B nutzen oder umgekehrt, so besteht zwischen den Gerätenetzwerken eine Kopplung. Es werden transparente, einseitige und zweiseitige, direkte und indirekte, statische und dynamische Kopplungen unterschieden.

3.4.1 Transparente Kopplung

Eine Kopplung zwischen Gerätenetzwerken wird als *transparent* bezeichnet, wenn beliebige Geräte eines Netzwerks Geräte des angekoppelten Netzwerks wie jene des eigenen Netzwerks nutzen können. Die an der Kopplung beteiligten Geräte (Gateways) sind in Bezug auf die Interaktion zwischen den Endgeräten durchsichtig (transparent), d. h. die Geräte des fremden Netzwerks erscheinen wie Geräte des lokalen Netzwerks. Dies ist auch der wesentliche Vorteil der transparenten Kopplung, weil an die Endgeräte keine zusätzlichen Anforderungen gestellt werden und normale Anwendungen zur Gerätenutzung verwendet werden können. Bei einer nicht transparenten Kopplung ist die Nutzung der Geräte des fremden Netzwerks nur für spezielle Geräte des lokalen Netzwerks möglich.

Entsprechend der oft unterschiedlichen primären Nutzungsmöglichkeiten der Geräte und der damit verbundenen unterschiedlichen Anforderungen an die einzelnen Netzwerke in Bezug auf Bandbreite, Echtzeit-Übertragungsfähigkeit, Ausfallssicherheit und dergleichen, ist die Transparenz nicht immer in allen Teilbereichen gegeben. Die fehlende Transparenz für bestimmte Nutzungsformen wird dann zu den Kopplungsverlusten gezählt (siehe auch Abschnitt 3.6).

3.4.2 Direkte und indirekte Kopplung

Erfolgt die Kopplung von zwei Gerätenetzwerken über zumindest ein dazwischenliegendes Netzwerk C, so wird die Kopplung als *indirekt* bezeichnet, anderenfalls wird sie als *direkte* Kopplung bezeichnet. Die Begriffe „Netzwerk“ und „dazwischenliegend“ sind hier primär systemspezifisch (auf den Vernetzungsstandard bezogen) zu verstehen und beziehen sich nur bedingt auf die physikalische Netzwerktopologie. Beispielsweise kann eine Kopplung zwischen mehreren Netzwerken physikalisch auf einem einzigen Bus erfolgen, wie dies bei der Kopplung zwischen den Vernetzungsstandards HAVi und AV/C der Fall ist.

Eine indirekte Kopplung erfordert jeweils transparente Kopplungen zwischen den beteiligten Netzwerken, anderenfalls ist die Nutzung der Geräte des indirekt gekoppelten

Netzwerks auf spezielle Geräte, die eine genaue Kenntnis über die an der indirekten Kopplung beteiligten Geräte und Netzwerke besitzen, beschränkt. Während die direkte Kopplung nur einen Umsetzungsschritt erfordert, sind bei indirekt gekoppelten Netzwerken zumindest zwei Umsetzungsschritte erforderlich¹². Da die indirekte Kopplung von zwei Netzwerken über verschiedene dazwischenliegende Netzwerke erfolgen kann, sind auch mehrfache Kopplungen zwischen zwei Netzwerken mit unterschiedlicher Anzahl an Umsetzungsschritten möglich. Der direkteste Weg zwischen zwei Netzwerken, d. h. der Weg mit den wenigsten Umsetzungsschritten, wird hierin als *Kopplungsdistanz* der beiden Netzwerke bezeichnet (siehe auch Gleichung 3.4).

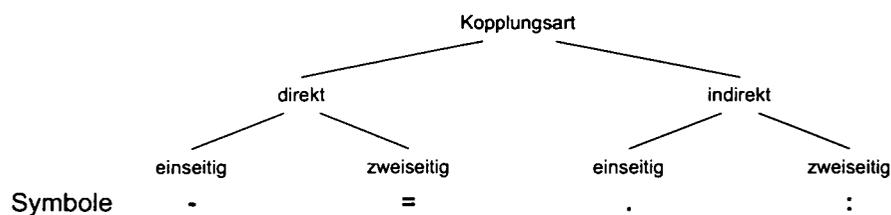


Abbildung 3.10: Kopplungsarten - Symbole

3.4.3 Einseitige und zweiseitige Kopplung

Eine weitere Kopplungseigenschaft ist die Richtung der Netzwerk-Kopplung. Wenn Geräte eines Netzwerks A Geräte eines Netzwerks B nutzen können, so besteht zumindest eine *einseitige* Kopplung zwischen dem Gerätenetzwerk A und B. Können zusätzlich auch Geräte des Netzwerks B Geräte des Netzwerk A nutzen, so ist die Kopplung zwischen den beiden Netzwerken *zweiseitig*. Von der Richtung der Kopplung, die sich an der Gerätenutzung orientiert, ist die Richtung der Kommunikation zu unterscheiden, sie ist zumeist auch bei einseitigen Kopplungen in beide Richtungen notwendig.

Bei einer zweiseitigen, transparenten Kopplung ist darauf zu achten, dass die eigenen an das gekoppelte Netzwerk exportierten Geräte nicht zurück importiert werden. Wenn es der Importmechanismus nicht verhindert, kommt es sonst zur *Mehrfachrepräsentation* der Geräte und je nach Exporteinstellungen zum *Mehrfachexport*, d. h. zum erneuten Export bereits exportierter Geräte. Ein Beispiel dazu zeigt Tabelle 3.7. Hier wird das Gerät-1.1 aus dem Netz-1 in das Netz-2 exportiert und dort als Gerät-2.1 verfügbar gemacht. Wird dieses (virtuelle) Gerät-2.1 (zurück) ins Netz-1 exportiert und dort als Gerät-1.2 verfügbar gemacht (Rückimport aus der Sicht von Netz-1), so führt dies zu einer Mehrfachrepräsentation des Gerätes-1.1 im Netz-1. Ein Export des (virtuellen) Gerätes-1.2 in das Netz-2, wo es als Gerät-2.2 verfügbar gemacht wird, stellt dann einen

¹²Hierbei wird von einer einseitigen Kopplung ausgegangen, für eine zweiseitige Kopplung sind auch noch die Umsetzungsschritte für die andere Kopplungsrichtung zu berücksichtigen (siehe Abschnitt 3.4.3)

Mehrfachexport des Gerätes-1.1 dar. In dieser Folge des Exports und Rückimports werden fortlaufend immer neue Mehrfachrepräsentationen erzeugt. Dieser Prozess nimmt erst auf Grund einer Ressourcenbeschränkung eines an der Kopplung beteiligten Gerätes ein Ende.

Netzwerk 1		Netzwerk 2	Anmerkung
Gerät 1.1	→	Gerät 2.1	Export von Gerät 1.1
Gerät 1.2	←	Gerät 2.1	Rückimport von Gerät 1.1
Gerät 1.2	→	Gerät 2.2	Mehrfachexport von Gerät 1.1
...

Tabelle 3.7: Beispiel für Rückimport und Mehrfachexport

Durch ein Nicht-Exportieren importierter Geräte werden diese Probleme zwar vermieden, jedoch ist dann die indirekte, transparente Kopplung von Netzwerken nicht mehr möglich. Eine allgemeinere Möglichkeit zur Vermeidung der Mehrfachrepräsentation ist die bedingte Importierung auf Grund eines eindeutigen Herkunftsbezeichners, wie beispielsweise einer eindeutigen Kennung des Ursprungsnetzwerks.

Das Problem der Mehrfachrepräsentation tritt auch bei mehrfach, indirekt und transparent gekoppelten Netzwerken auf, weil hier über unterschiedliche Netzwerke Geräte importiert werden, die ursprünglich aus dem selben Netzwerk stammen (*Mehrfachimport*). Auch hierbei kann das Problem durch eine bedingte Importierung vermieden werden (siehe auch Abschnitt 3.5).

Zur einfachen Beschreibung der in diesem und im vorherigen Unterabschnitt behandelten Kopplungseigenschaften werden die in Abbildung 3.10 dargestellten Symbole¹³ verwendet. Bei einseitigen Kopplungen kennzeichnet der erste Netzwerkbezeichner jenes Netzwerk, dessen Geräte die Geräte des anderen Netzwerks nutzen können. Im Falle von zweiseitigen Kopplungen ist die Reihenfolge der Netzwerkbezeichner ohne Bedeutung. So bezeichnet eine „HAVi-EIB“-Kopplung eine direkte, einseitige Kopplung zwischen HAVi und EIB. Hierbei können von Geräten des HAVi-Netzwerks Dienste der Geräte des EIB genutzt werden, jedoch nicht umgekehrt. Im Gegensatz dazu bezeichnet eine „Jini:UPnP“-Kopplung eine indirekte, zweiseitige Kopplung zwischen Jini und UPnP.

3.4.3.1 Formale Beschreibung einseitiger Kopplungen

Formal sind einseitige Kopplungen durch die Gleichungen 3.1 - 3.3 beschrieben. Hierbei bezeichnet $K^1(A, B)$ eine einseitige Kopplung zwischen dem Netzwerk A und B .

¹³Ein Symbol mit zwei Elementen kennzeichnet eine zweiseitige Kopplung. Die im Vergleich zur indirekten Kopplung „stärkere“ direkte Kopplung wird durch ein breites Symbol dargestellt.

$$K^1(A, B) \Leftrightarrow K_D^1(A, B) \vee K_I^1(A, B) \quad (3.1)$$

$$K_D^1(A, B) \Leftrightarrow \text{Netz } B \text{ ist an das Netz } A \text{ direkt gekoppelt} \quad (3.2)$$

$$K_I^1(A, B) \Leftrightarrow \exists X : K_D^1(A, X) \wedge K^1(X, B) \quad (3.3)$$

Die Indizes $_D$ und $_I$ bezeichnen eine direkte bzw. indirekte Kopplung. Die Kopplungsdistanz $d_K^1(A, B)$ der einseitigen Kopplung zwischen dem Netzwerk A und B wird in Gleichung 3.4 definiert. Im Falle einer mehrfachen, indirekten Kopplung ist sie die kurzeste Distanz zwischen A und B .

$$d_K^1(A, B) = \begin{cases} 0 & : A = B \\ 1 & : K_D^1(A, B) \\ 1 + \min(d_K^1(X, B)) & : K_I^1(A, B) \\ \infty & : \text{sonst} \end{cases} \quad (3.4)$$

3.4.3.2 Formale Beschreibung zweiseitiger Kopplungen

Formal sind zweiseitige Kopplungen durch die Gleichungen 3.5 - 3.7 beschrieben. Hierbei bezeichnet $K^2(A, B)$ eine zweiseitige Kopplung zwischen den Netzwerken A und B .

$$K^2(A, B) \Leftrightarrow K_D^2(A, B) \vee K_I^2(A, B) \quad (3.5)$$

$$K_D^2(A, B) \Leftrightarrow K_D^1(A, B) \wedge K_D^1(B, A) \quad (3.6)$$

$$K_I^2(A, B) \Leftrightarrow K^1(A, B) \wedge K^1(B, A) \wedge \neg(K_D^2(A, B)) \quad (3.7)$$

Die Kopplungsdistanz $d_K^2(A, B)$ zwischen den zweiseitig gekoppelten Netzwerken A und B wird in Gleichung 3.8 definiert. Besteht die zweiseitige Kopplung aus zumindest einer indirekten Kopplung, so ist die zweiseitige Kopplungsdistanz gleich der langeren der beiden einseitigen Kopplungsdistanzen zwischen A und B .

$$d_K^2(A, B) = \begin{cases} 0 & : A = B \\ d_K^1(A, B) & : (d_K^1(A, B) > d_K^1(B, A)) \wedge K^2(A, B) \\ d_K^1(B, A) & : (d_K^1(B, A) > d_K^1(A, B)) \wedge K^2(A, B) \\ \infty & : \text{sonst} \end{cases} \quad (3.8)$$

3.4.4 Statische und dynamische Kopplung

Dieser Unterabschnitt behandelt die allgemeine Einsetzbarkeit und den notwendigen Konfigurationsaufwand einer Kopplung. Eine Kopplung wird als *dynamisch* bezeichnet, wenn sie:

- (1) transparent ist,
- (2) nicht auf eine spezielle Umgebung bzw. Geratekonfiguration beschrankt ist und

- (3) keinen oder einen nur sehr geringen Konfigurationsaufwand im Verhältnis zum Installationsaufwand der gekoppelten Netzwerke erfordert.

Wird auch nur eines dieser Kriterien nicht erfüllt, so wird die Kopplung als *statisch* bezeichnet.

Die ersten beiden Forderungen beziehen sich auf die allgemeine Einsetzbarkeit der Kopplung. Sie ist Grundvoraussetzung für eine Kopplung zwischen Netzwerken, die sich spontan ändern können. Die Forderung der Transparenz ist damit begründet, dass bei nicht transparent gekoppelten Netzwerken nur über „eingeweihte“ Geräte die Dienste und Funktionen der Geräte des angekoppelten Netzwerks genutzt werden können. Für eine Nutzung der angekoppelten Geräte müssten die bestehenden Geräte erweitert werden und im Falle von indirekt gekoppelten Netzwerken ist zusätzlich eine genaue Kenntnis über die an der indirekten Kopplung beteiligten Geräte und Netzwerke erforderlich. Zusätzliche Anforderungen an die Geräte stehen jedoch einer allgemeinen Einsetzbarkeit der Kopplung entgegen. Während die erste Forderung die Nutzung angekoppelter (importierter) Geräte behandelt, bezieht sich die zweite Forderung auf die zu exportierenden Geräte. Eine Kopplung, die eine spezielle Umgebung voraussetzt bzw. nur eine bestimmte Anzahl von Geräten oder gar nur einzelne Geräte berücksichtigt, ist nicht allgemein einsetzbar.

Die dritte Forderung bezieht sich auf den notwendigen Aufwand zur Einrichtung der Kopplung. Ist bei jeder Änderung der beteiligten Netzwerke (Hinzufügen und Entfernen von Geräten) ein wesentlicher zusätzlicher Aufwand neben der netzwerkabhängigen Installation des Gerätes nötig, so ist die Kopplung nur für eine nicht veränderliche (statische) Umgebung geeignet.

Die in Abschnitt 4.1.4 beschriebene transparente HAVi-EIB Service-Kopplung ermöglicht den Geräten des HAVi-Netzwerks die Nutzung von Funktionen, die von Geräten des EIB-Netzwerks bereitgestellt werden. Zur Darstellung der EIB-Geräte im HAVi-Netzwerk benötigt das Service-Gateway zusätzliche Informationen über die EIB-Geräte, wie Gerätename, Einsatzort oder Gerätetyp. Da diese Informationen ohne großen Aufwand im Zuge der bei EIB benötigten manuellen Installation gesammelt werden können, kann der zusätzlich zur netzwerkabhängigen Installation des Gerätes nötige Aufwand als gering eingestuft und die Kopplung als dynamisch bezeichnet werden. Die Bewertung des zusätzlichen Installationsaufwandes einer Kopplung ist jedoch strenger, wenn die betreffenden Netzwerke ohne manuellen Eingriff verwaltet werden, d. h. abgesehen von der physikalischen Verbindung der Geräte bei kabelgebundenen Netzwerken die Installation automatisch (durch die Geräte selbst) erfolgt. Infolgedessen wird eine transparente HAVi-AV/C Service-Kopplung als statische Kopplung eingestuft werden, wenn sie zusätzliche Informationen für die Kopplung vom Benutzer beim Hinzufügen eines neuen AV/C-Gerätes erfordert.

3.5 Kopplungssysteme

Verfügt ein Netzwerk über mehr als eine Kopplung, so werden die einzelnen Kopplungen aller Netze gemeinsam als *Kopplungssystem* bezeichnet. In diesem Abschnitt werden neben dem Grundelement des Kopplungssystems (AB-Kopplung) zwei spezielle Systeme (C-Kopplung und T-Kopplung) vorgestellt.

3.5.1 AB-Kopplung

Eine *AB-Kopplung* bezeichnet eine direkte Kopplung zwischen zwei unterschiedlichen Netzwerken. Sie ist das Basiselement eines Kopplungssystems und sorgt für die Umsetzung einer der drei Schichten der in Abschnitt 3.1 vorgestellten Gerätearchitektur.

Es gibt verschiedene Ansätze mehrere Netzwerke zu koppeln. Die naheliegendste Möglichkeit ist, die benötigten Netzwerke jeweils über eine eigene AB-Kopplung anzubinden. Hierbei ist die Kopplungsdistanz minimal ($= 1$), die Anzahl an Kopplungen jedoch maximal. Im ungünstigsten Fall werden bei n verschiedenen Netzwerken $n \cdot (n - 1)$ einseitige Kopplungen benötigt¹⁴. Da jedoch für jedes Netzwerk unterschiedliche Kopplungspräferenzen zu anderen Netzwerken vorliegen, liegt die praktisch notwendige Kopplungsanzahl weit unterhalb der theoretisch möglichen. Anhand bestimmter globaler Nutzungsprofile im Heimbereich könnte ermittelt werden, zwischen welchen Netzwerken eine Kopplung notwendig ist und mit welcher Häufigkeit.

So könnte beispielsweise für die vier Gerätenetzwerke HAVi, UPnP, AV/C und LON festgestellt werden, dass eine AV/C-HAVi, AV/C-LON, AV/C-UPnP, LON-AV/C, LON-HAVi und LON-UPnP Kopplung nie oder nur sehr selten benötigt wird; eine HAVi-AV/C oder UPnP-LON Kopplung häufiger benötigt wird als eine HAVi-LON oder UPnP-AV/C Kopplung; und eine HAVi-UPnP, HAVi-AV/C, UPnP-HAVi und UPnP-LON Kopplung oft benötigt wird. In diesem Fall liegt die praktisch notwendige Kopplungsanzahl bei 6 ($2 + 4$; eine zweiseitige und vier einseitige Kopplungen) und die maximal notwendige bei 12. Für einen bestimmten Anwendungsort mag diese Reduktion ausreichend sein, global gesehen existieren jedoch mehr als vier unterschiedliche Vernetzungsstandards für Gerätenetzwerke im Heimbereich und damit ist die benötigte Kopplungsanzahl noch immer sehr hoch.

Ein anderer Ansatz zur Kopplung von mehreren Netzwerken basiert auf der stärkeren Berücksichtigung von häufig benötigten Kopplungen entsprechend bestimmter globaler Nutzungsprofile durch Beschränkung auf diese Kopplungen und die Nutzung von indirekten Kopplungen. Die häufig benötigten Verbindungen werden als transparente AB-Kopplungen realisiert und können so indirekte Kopplungen für weniger gefragte Verbindungen bereitstellen. Entsprechend dem obigen Beispiel könnte das Kopplungssystem auf folgende AB-Kopplungen reduziert werden: HAVi-AV/C, UPnP=HAVi und UPnP-LON. Damit ergeben sich folgende einseitige Kopplungsabfolgen: UPnP-HAVi-

¹⁴Zum Vergleich wird eine zweiseitige Kopplung als zwei einseitige Kopplungen gezählt.

AV/C und HAVi-UPnP-LON. Die Kopplungsanzahl kann hier dementsprechend auf 4 ($2 + 2$, eine zweiseitige und zwei einseitige Kopplungen) gesenkt werden. Auf Grund der Verwendung von indirekten Kopplungen ist jedoch die Kopplungsdistanz höher als beim vorigen Ansatz und es besteht eine Abhängigkeit von den „Verbindungsnetzwerken“ entsprechend den Kopplungsabfolgen.

Ein weiterer Ansatz ist die im nächsten Unterabschnitt vorgestellte C-Kopplung, die eine Reduktion der Kopplungen durch Verwendung eines zentralen Netzwerks bietet.

Sind zwei Netzwerke über mehr als eine Kopplung miteinander verbunden, so besteht zwischen den Netzwerken eine *mehrfache* Kopplung. Diese kann entweder direkt durch mehrere AB-Kopplungen zu einem zweiten Netzwerk oder aber durch eine Reihe von AB-Kopplungen zwischen mehr als zwei Netzwerken entstehen.

Bei mehrfachen, transparenten Kopplungen ist darauf zu achten, dass es nicht zum *Mehrfachexport* der Geräte kommt, d. h. zum erneuten Export bereits exportierter Geräte, weil sonst die Geräte mehrfach repräsentiert werden (*Mehrfachrepräsentation*, siehe auch Abschnitt 3.4.3). Eine Möglichkeit zur Vermeidung der Mehrfachrepräsentation ist die bedingte Importierung auf Grund eines eindeutigen Herkunftsbezeichners, wie beispielsweise einer eindeutigen Kennung des Ursprungsnetzwerks.

3.5.2 C-Kopplung

Eine *C-Kopplung*¹⁵ ist eine indirekte Kopplung von mehreren Netzwerken über ein zentrales Netzwerk, an welches die einzelnen Netzwerke jeweils zweifach, direkt und transparent gekoppelt sind. Ein Unterschied zu zwei normalen AB-Kopplungen ist die spezielle Rolle des zentralen Netzwerks. Die Idee bei dieser Kopplung ist die Reduktion der Anzahl an Kopplungen, indem jedes Netzwerk einzig an das zentrale Netzwerk direkt und über dieses mit allen anderen indirekt gekoppelt ist.

Wie bereits im vorherigen Unterabschnitt erwähnt, ist die Anzahl an benötigten einseitigen Kopplungen im ungünstigsten Fall bei n verschiedenen Netzwerken gleich $n * (n - 1)$ ¹⁶. Erfolgt die Verbindung der einzelnen Netzwerke über eine C-Kopplung, so kann die Anzahl der Kopplungen auf den Wert $(n - 1) * 2$ reduziert werden, weil nur noch jeweils die direkte, zweifache und transparente Kopplung zu einem zentralen Netzwerk bereitgestellt werden muss. Die Kopplungsdistanz der über eine C-Kopplung indirekt gekoppelten Netzwerke ist konstant zwei, d. h. es sind immer zwei Umsetzungsschritte nötig. Ein Problem der C-Kopplung ist, dass die Reduktion der Kopplungsanzahl erst dann voll greift, wenn die Hersteller sich auf einen Vernetzungsstandard für das zentrale Netzwerk einigen. Andererseits kann die C-Kopplung als Teil eines gesamten Kopplungssystems bereits zu einer Reduktion der zu unterstützenden Kopplungen führen, wenn zumindest bereichsweise ein zentrales Netzwerk auserkoren wird und diese direkt und transparent gekoppelt sind, z. B. HAVi in der Unterhaltungselektronik und UPnP im

¹⁵Das C im Begriff C-Kopplung steht für „Central“.

¹⁶Zum Vergleich wird eine zweiseitige Kopplung als zwei einseitige Kopplungen gezählt.

Bereich Small Office/Home Office.

Das in [Cho02] vorgeschlagene Home Network Framework basiert auf einer C-Kopplung, wobei das zentrale Netzwerk als einzelnes Gerät realisiert ist. Das als „Residential-Gateway“ bezeichnete Gerät besteht aus jeweils einem Middleware Manager für jedes angekoppelte Netzwerk und einem Middleware Broker (\approx zentrales Netzwerk), der für die Kommunikation zwischen den einzelnen Middleware Managern verantwortlich ist. Ein potentiell Problem dieser Realisierung ist, dass das zentrale Netzwerk als einzelnes Gerät (Residential-Gateway) realisiert ist, weil damit die gesamte Kommunikation nur über ein Gerät erfolgt.¹⁷ Ein weiteres Beispiel für den Einsatz einer C-Kopplung ist in [NSA02] beschrieben. Hier wird das zentrale Netzwerk als Virtual Overlay Network bezeichnet und besteht aus einzelnen miteinander über TCP/IP verbundenen Geräten, den so genannten Application Level Gateways (ApGW). Die ApGW kommunizieren untereinander über HTTP und sorgen für die Umsetzung der UI- oder Service-Schicht zum jeweils angeschlossenen Netzwerk.¹⁸

3.5.3 T-Kopplung

Eine *T-Kopplung*¹⁹ ist eine indirekte Kopplung auf der System-Schicht von gleichen Netzwerken über ein *Transitnetzwerk*. Die Netzwerke sind an das zentrale Netzwerk (Transitnetzwerk) nur insoweit gekoppelt, dass zwischen ihnen insgesamt eine indirekte und transparente Kopplung gewährleistet ist, jedoch am Transitnetzwerk die Geräte der angekoppelten Netzwerke nicht genutzt werden können. Ein Beispiel für dieses Kopplungssystem ist die in Abschnitt 4.2.1 vorgestellte Kopplung von zwei entfernten HAVI-Netzwerken über einen TCP/IP-Tunnel.

3.6 Kopplungsverluste

Die Kopplung von Gerätenetzwerken durch eines der beschriebenen Gateways ist unweigerlich auch mit Verlusten behaftet. Selbst bei einer dynamischen Kopplung von Gerätenetzwerken ist zumindest eine durch die Bearbeitungszeit des Gateways verursachte Verzögerung der Datenkommunikation zu berücksichtigen. Die mit der Kopplung von Gerätenetzwerken zusammenhängenden Einbußen gegenüber einem homogenen Gerätenetzwerk werden in Folge als Kopplungsverluste bezeichnet.

¹⁷Inwieweit das Home Network Framework eine dynamische Kopplung bietet, lässt sich anhand der vorliegenden Informationen nicht abschließend feststellen.

¹⁸Die durch das Virtual Overlay Network erreichte Kopplung ist nicht dynamisch, weil die Registrierung der Geräte bei den ApGWs am Netzwerk der Nutzer jeweils manuell eingetragen werden müssen (Konfigurationsaufwand!). Des Weiteren sind am zentralen Netzwerk nur wenige Befehle für bestimmte Anwendungen, jedoch keine ausreichende API für Dienste und Funktionen von Geräten, definiert worden (Beschränkung auf spezielle Umgebung!).

¹⁹Das T im Begriff T-Kopplung steht für „Tunnel“.

- **Zeit:** Durch die Kopplung von Gerätenetzwerken entsteht durch die Bearbeitungszeit aller an der Kopplung beteiligten Gateways und der Übertragungszeit der Daten über etwaige Transfernetze bei indirekt gekoppelten Netzen (Kopplungsdistanz größer 1) eine Verzögerung.
- **Betriebssicherheit:** Einen weiteren Kopplungsverlust stellt die Verminderung der Betriebssicherheit dar. Wenn Geräte nicht direkt sondern über Gateways verbunden sind, dann steigt gegenüber einer direkten Verbindung der Geräte die Ausfallanfälligkeit mit jeder nicht redundanten dazwischenliegenden Kopplung. Ein besonderes Problem stellen in diesem Zusammenhang C-Kopplungen dar, deren zentrales Netzwerk als einzelnes Gerät realisiert ist (z. B. das „Residential Gateway“ des Home Network Framework [Cho02]).
- **Funktionalität:** Den größten Anteil an Kopplungsverlusten bilden wohl Einschränkungen infolge einer unvollständigen Umsetzung der jeweiligen Schicht durch das Gateway. Ein typisches Beispiel dafür ist das API-Mapping bei Service-Gateways: In manchen Fällen ist eine spezielle Funktionalität, wie die Ressourcenverwaltung bei HAVi (Resource Manager) oder die Transaktionen bei Jini, auf Grund des Fehlens entsprechender Funktionen im Importnetz nicht oder nur mit sehr großem Aufwand umsetzbar.
- **Sicherheit²⁰:** Eine Verminderung des Schutzes vor unerlaubtem Zugriff auf Geräte durch eine Kopplung stellt eine weitere Art von Kopplungsverlusten dar. Einerseits führt die mit einer Kopplung verbundene Erweiterung bzw. Öffnung des Netzwerks für einen größeren Anwenderkreis auch zu einer größeren Angriffsfläche. Andererseits ist der Schutz vor fremden Zugriffen stark vom eingesetzten Übertragungssystem abhängig. So verändert sich das Risikopotential vor einem unerlaubten Zugriff bei einem drahtgebundenen Feldbus zur Vernetzung von Geräten der Heim-Automatisierung schlagartig, wenn dieser über ein Gateway mit einem drahtlosen WebPad verbunden ist. Erfolgt die drahtlose Übertragung ohne Verschlüsselung, so stellt dies einen Kopplungsverlust durch das höhere Sicherheitsrisiko dar.
- **Transport:** Weitere Kopplungsverluste entstehen auf Grund von Beschränkungen im Transport (Übertragungssystem und -medium) des Importnetzes im Vergleich zum Exportnetz: niedrigere Datenrate, Best-Effort (z. B. bei IEEE 802.3) an Stelle von garantierter Bandbreite (z. B. bei IEEE 1394), größere Ausfallanfälligkeit, keine Unterstützung von Prioritäten, usw.

Für die Echtzeit-Übertragung von Audio/Video-Daten (AV-Streaming) sind Unterschiede im Transport besonders heikel. Ist die Bandbreite im Importnetz geringer, so kann unter Umständen durch Umkodierung (Transcoding) des AV-Datenstroms in ein stärker komprimierendes Format eine Übertragung (bei eventuell merkbar verminderter AV-Daten-Qualität) erreicht werden. Im Extremfall ist eine

²⁰Hierbei ist Sicherheit im Sinne von „Security“, d. h. dem Schutz vor unerlaubtem Zugriff gemeint. Sicherheit im Sinne von „Safety“ wurde unter dem Punkt Betriebssicherheit behandelt.

geeignete Übertragung nicht möglich. Da das AV-Streaming zwischen unterschiedlichen Netzwerken die Behandlung einer Reihe von weiteren Aspekten erfordert (siehe auch [WZ03, Zie00] und Abschnitt 4.2.2) wäre es auch durchaus denkbar, das AV-Streaming als eigenständigen Bereich in einer Erweiterung des Kopplungsmodells aufzunehmen.

3.7 Ergänzende Anmerkungen

Die beiden grundlegenden Zielsetzungen des in den vorhergehenden Abschnitten dieses Kapitels entwickelten allgemeinen Modells zur Kopplung von Gerätenetzwerken im Heimbereich waren ein Modell zu definieren, dass sich einerseits für die Analyse und Beschreibung von bestehenden Kopplungen und andererseits als Grundlage zum Design von neuen Kopplungen eignet. Diese sich ergänzenden Forderungen führten nach Analyse der wesentlichen Eigenschaften und Funktionsweisen verschiedener Gerätenetzwerke und Kopplungen zur Definition einer eigenen Gerätearchitektur und darauf aufbauend zur Definition der unterschiedlichen Kopplungsmöglichkeiten.

Die Gerätearchitektur definiert die für eine Vernetzung von Geräten im Heimbereich notwendigen Dienste und ordnet sie entsprechend ihres Aufgabenbereichs jeweils einer der drei Schichten System Layer, Service Layer und User Interface Layer zu. Eine Kopplung wird dann als Umsetzung auf einer der Schichten der Gerätearchitektur (Gateway) oder als nichtumsetzende, mehrfache Anbindung beschrieben.

In Abbildung 3.11 ist eine Gegenüberstellung der einzelnen Schichten der erstellten Gerätearchitektur zu den Schichten des OSI-Referenzmodells [IT94, Zim80, Tan00] dargestellt²¹. Hierbei steht im Wesentlichen dem Hauptteil der Gerätearchitektur nur der OSI-Layer-7 (Application Layer) gegenüber²². Das OSI-Referenzmodell und die vorhandenen Application Layer Services (X.217, X.218, X.219, ...) bieten jedoch keine ausreichende Möglichkeit zur Behandlung der Aspekte wie die automatische Bekanntmachung und Beschreibung der Geräte im Netzwerk (zentrale oder verteilte Datenbank mit Abfragemöglichkeiten, periodisches Senden von Alive-Nachrichten, Configuration Rom, ...), die Geräteansteuerung (Anwendungsprogramm-Schnittstelle zur Geräte-HW/SW) oder die Benutzerschnittstelle.

Eine weitere Vergleichsmöglichkeit ist, die einzelnen im Kopplungsmodell unter Kopplungsarten definierten Gateways (System-Gateway, Service-Gateway und UI-Gateway) den im Zusammenhang mit „Lokalen Netzen“ verwendeten Begriffen für Verbindungselemente (Repeater, Bridge, Router und Gateway) gegenüber zu stellen. Obgleich die

²¹Im Gegensatz zum OSI-Referenzmodell, bei dem das Physical Media nicht Teil der ersten Schicht (Physical Layer) ist [IT94, Els91], werden die verwendbaren physikalischen Medien innerhalb der Standards IEEE 1394 und IEEE 802.3 festgelegt. Bei der Gerätearchitektur werden sie dem Transport zugerechnet.

²²Die Anwendung, der User und das Physical Media sind nicht Teil des OSI-Referenzmodells [IT94] und hier nur der Vollständigkeit halber dargestellt.

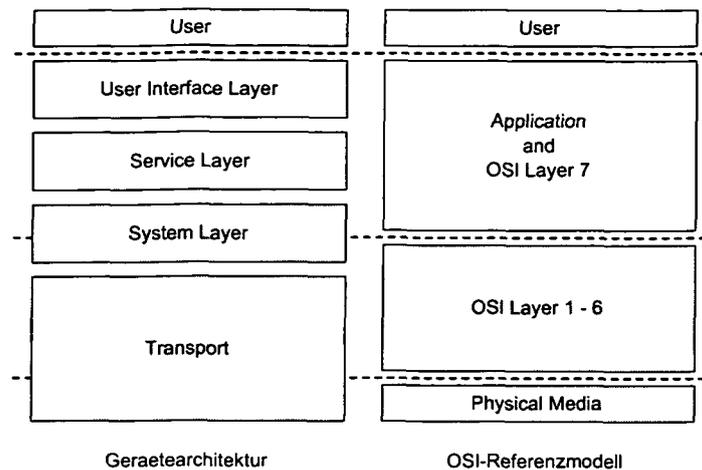


Abbildung 3.11: Vergleich: Gerätearchitektur vs. OSI-Referenzmodell

genannten Begriffe nicht ganz einheitlich verwendet werden, kann zumindest Folgendes grundsätzlich festgehalten werden [Lip98, Lin01, As03]:

- (1) Repeater verbinden auf der OSI-Schicht-1 Netzsegmente unmittelbar und ermöglichen durch eine Signalregenerierung eine Vergrößerung der Ausdehnung des Netzes.
- (2) Brücken (Bridges) verbinden auf der OSI-Schicht-2 Netzsegmente, wobei sie den lokalen Datenverkehr der einzelnen Segmente durch die Filterung von Rahmen anhand der physikalischen Adressen (MAC-Adressen) voneinander trennen.
- (3) Router verbinden auf der OSI-Schicht-3 einzelne Netze (Subnet) zu einem gesamten Netz und sorgen anhand der logischen Adressen für die Weiterleitung von Paketen an das Ziel-Subnet.
- (4) Gateways verbinden unterschiedliche Netze bzw. inkompatible Systeme durch Protokollumsetzung der entsprechenden OSI-Schicht (zumeist OSI-Schicht-7).

Die einzelnen im Kopplungsmodell definierten Gateways entsprechen demnach dem bei „Lokalen Netzen“ verwendeten Begriff „Gateway“. Da die zugehörigen OSI-Schichten der Verbindungselemente „Repeater“, „Bridge“ und „Router“ bei der Gerätearchitektur dem Transport zugeordnet sind, werden diese Verbindungselemente im engeren Sinn des Kopplungsmodells (System-, Service- und UI-Schicht) nicht verwendet. Jedoch bezeichnen sie im erweiterten Sinn des Kopplungsmodells - wie im Abschnitt über System-Gateways ausgeführt wurde - ihrer Entsprechung nach drei Möglichkeiten zur Verbindung von Netzen mit unterschiedlichem Transport.

Basierend auf dem in diesem Kapitel definierten allgemeinen Modell zur Kopplung von Gerätenetzwerken im Heim werden im nächsten Kapitel am Beispiel der Unterhaltungselektronik sechs verschiedene dynamische Kopplungen – drei erstellte Konzepte

und drei durchgeführte Implementierungen – zu den drei Bereichen des Heims (Heim-Automatisierung; Unterhaltungselektronik; Small Office/Home Office) beschrieben, wobei alle drei Gateway-Typen sowie eine T-Kopplung und mehrere AB-Kopplungen zur Anwendung kommen.

Kapitel 4

Dynamische Kopplung am Beispiel der Unterhaltungselektronik

Für Kopplungen von Gerätenetzwerken im Heimbereich ist entscheidend, dass diese allgemein einsetzbar sind und ohne wesentlichen Konfigurationsaufwand auskommen. Nur durch die allgemeine Einsetzbarkeit der Kopplung können die unterschiedlichen im Heimbereich eingesetzten Anwendungen und Geräte unterstützt werden. Seitens der Benutzer wird ein möglichst geringer Konfigurationsaufwand der Kopplung zur Inbetriebnahme und bei Änderungen in den beteiligten Netzwerken (Hinzufügen und Entfernen von Geräten) gefordert, wobei hier das Verhältnis des Aufwands für die Kopplung zum Installationsaufwand des Netzwerks bzw. der Geräte wesentlich ist. Eine geeignete Kopplung von Gerätenetzwerken im Heimbereich muss deshalb dynamisch¹ sein.

In diesem Kapitel werden am Beispiel der Unterhaltungselektronik Konzepte und Realisierungen für verschiedene dynamische Kopplungen im Heimbereich beschrieben. Der Abschnitt 4.1 beschreibt zwei Implementierungen für eine dynamische Kopplung zwischen der Unterhaltungselektronik und Heim-Automatisierung auf der Service-Schicht. In Abschnitt 4.2 wird die dynamische Kopplung zwischen Netzwerken der Unterhaltungselektronik auf System- und Service-Schicht behandelt. Der letzte Abschnitt (4.3) in diesem Kapitel ist der dynamischen Kopplung der Bereiche Unterhaltungselektronik und Small Office/Home Office auf der UI-Schicht gewidmet.

Als Netzwerk für die Unterhaltungselektronik wird in den nachfolgend beschriebenen Implementierungen vor allem HAVi eingesetzt. Die Software-Architektur des dabei verwendeten HAVi-Stacks² ist in Abbildung 4.1 dargestellt.

Der HAVi-Stack wurde in ANSI C [ISO99] geschrieben und ist sehr modular aufgebaut. Er basiert auf zwei Abstraktions-Schichten, dem ALOS [dmn03] zur Abstrahierung der Betriebssystem-spezifischen Funktionen und dem AL1394 [dmn02a, dmn02c, dmn02b] zur Abstrahierung des Zugriffs auf den IEEE 1394-Bus. Damit ist eine Portierung auf andere Betriebssysteme und die Verwendung anderer IEEE 1394-Treiber ohne großen Aufwand

¹siehe Abschnitt 3.4.4

²Der HAVi-Stack ist Teil eines Software-Development-Kits (HAVi-SDK [dmn02e, dmn02d]) des von mir und vier Studienkollegen im Jahr 2001 gegründeten Unternehmens *dmn Software-Entwicklung GmbH*. Als technischer Leiter dieses Unternehmens war ich sowohl für das System-Design als auch für die Implementierung des Stacks verantwortlich.

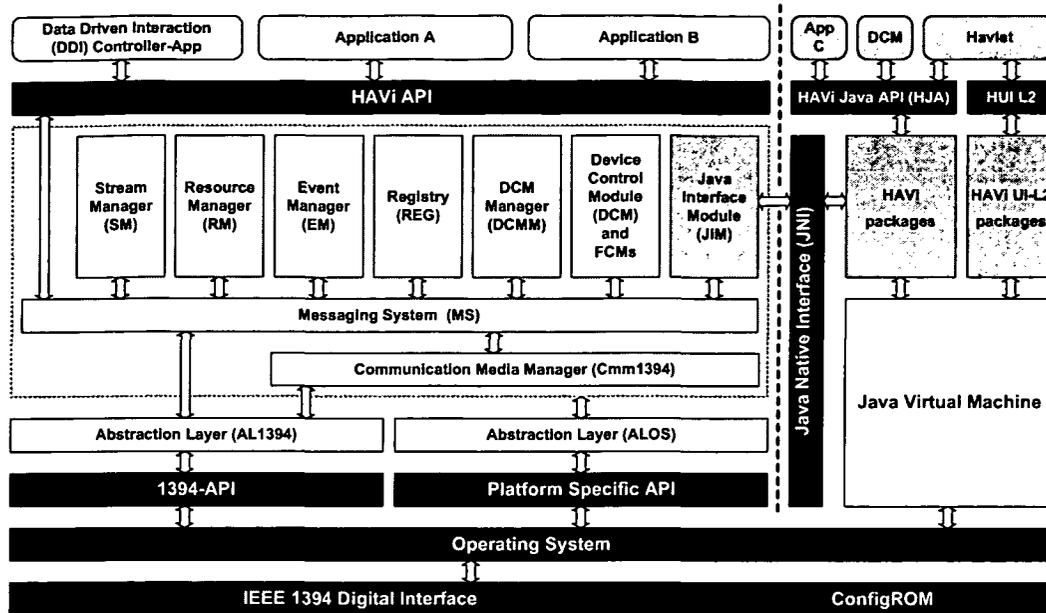


Abbildung 4.1: HAVi-Stack

möglich³. Die Module im grauen Bereich stellen den Kern des HAVi-Stacks dar, jene gelben Module oberhalb des grauen Bereichs repräsentieren die HAVi-Anwendungen in C-Code. Das Java Interface Module (JIM) ist für die Verbindung zur „Java-Welt“ verantwortlich und ermöglicht gemeinsam mit den HAVi-Packages (Java-Code) und der Java Virtual Machine die Ausführung von Java-basierten HAVi-Anwendungen und -Geräte-modulen entsprechend der HAVi Java API (HJA) [HAV01].

4.1 Unterhaltungselektronik – Heim-Automatisierung

In diesem Abschnitt werden zwei Implementierungen einer dynamischen Kopplung zur Anbindung von Geräten der Heim-Automatisierung an ein Netzwerk der Unterhaltungselektronik beschrieben. Die erste Realisierung ist eine HAVi-EIB und die zweite eine HAVi-LON Kopplung. Beide Kopplungen sind direkt, einseitig, transparent und erfolgen auf der Service-Schicht, sie ermöglichen HAVi-Geräten die Nutzung von EIB- bzw. LON-Geräten. In Abbildung 4.2 ist die Kopplungsmethode graphisch dargestellt. Das Service-Gateway sorgt für die Umsetzung der Service-Schichten (eckig → rund) und ermöglicht damit die Ansteuerung der angekoppelten Geräte über eine andere als die ursprüngliche API.

³Derzeit werden die Betriebssysteme Linux, Windows, QNX und RTEMS unterstützt.

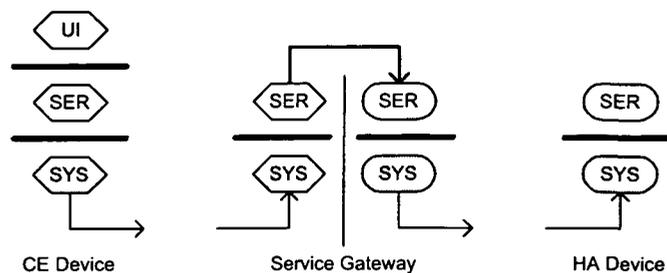


Abbildung 4.2: CE-HA Service-Gateway

4.1.1 Umsetzung der Service-Schicht

Die Service-Schicht von LON basiert im Wesentlichen auf der Kommunikation mit Standard-Netzwerkvariablen (SNVT)⁴ und jene von EIB erfolgt zumeist über Kommunikationsobjekte gemäß EIS, die über Gruppenadressen verbunden sind⁵. Die Verbindung der Netzwerkvariablen (Binding) bzw. die Zuweisung der Gruppenadressen erfolgt im Rahmen der Netzwerkinstallation. Im HAVi-Netzwerk werden Geräte als DCMs und ihre funktionellen Komponenten als FCMs repräsentiert⁶. Für die Umsetzung der Service-Schicht ist damit eine Abbildung der über LON-Netzwerkvariablen bzw. EIB-Gruppenadressen verfügbaren Funktionen auf HAVi DCM und FCM erforderlich.

Bei der in [WG00] beschriebenen UPnP-EIB-Gateway Implementierung wird jedes EIB-Gerät als UPnP-Gerät repräsentiert, das UPnP-Dienste entsprechend den allgemeinen Gerätefunktionen (z. B. Relais, analoger Eingang, analoger Ausgang) bietet. Während diese Abbildung vom Standpunkt eines Technikers sinnvoll erscheint, ist sie nicht sehr benutzerfreundlich. Der Benutzer erwartet vielmehr Dienste und möchte sich nicht mit Geräten, Netzwerken und ihrer inneren Struktur befassen [Ros02a]. Ein Gerät der Heim-Automatisierung, das vier Relais besitzt und zum Schalten von Lampen eingesetzt wird, sollte durch vier eigenständige Licht-Dienste repräsentiert werden und nicht als ein Gerät, das vier Relais zum Schalten von Verbrauchern enthält. Ein Heim-Automatisierungs-Dienst sollte dem Benutzer Funktionen der Heim-Automatisierung bieten anstelle von Gerätefunktionen, mit welchen letztendlich Heim-Automatisierungsfunktionen realisiert werden.

Das Heim-Automatisierungs-Netzwerk (HA-Netzwerk) kann in drei Stufen unterteilt werden. In Abbildung 4.3 ist diese dreistufige Unterteilung dargestellt. Die unterste Stufe bilden die einzelnen Funktionen der Heim-Automatisierung, welche im Wesentlichen über LON-Netzwerkvariablen bzw. über EIB-Gruppenadressen genutzt werden können. Dienste der Heim-Automatisierung (z. B. Licht, Waschmaschine, Klimaanlage, Feuer-

⁴siehe Abschnitt 2.1.1.1 und Abschnitt 3.1

⁵siehe Abschnitt 2.1.1.2 und Abschnitt 3.2

⁶siehe Abschnitt 2.1.2.3 und Abschnitt 3.4

melder) basieren auf einer oder mehreren dieser Funktionen und bilden die zweite Stufe. Das HA-Netzwerk als Ganzes wird durch die dritte Stufe repräsentiert.

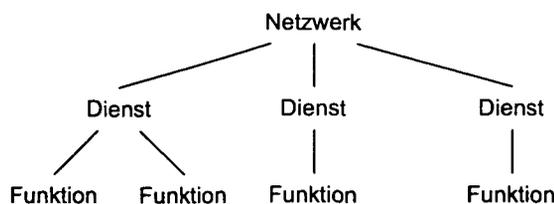


Abbildung 4.3: Unterteilung des Heim-Automatisierungs-Netzwerks

Tabelle 4.1 zeigt anhand der dreistufigen Unterteilung des HA-Netzwerks unterschiedliche Möglichkeiten für die Abbildung der einzelnen Elemente des HA-Netzwerks auf HAVi-DCM und -FCMs. Eine Bewertung dieser Möglichkeiten erfolgt in nachfolgender Auflistung:

Stufe\Option	1	2	3	4	5	6
HA-Netzwerk	D/F	D	D	-	-	-
HA-Dienst	-	F	-	D/F	D	-
HA-Funktion	-	-	F	-	F	D/F

Tabelle 4.1: Abbildung des HA-Netzwerks auf HAVi-DCM (D) und -FCMs (F)

- Bei Option 1, 4 und 6 gehört jeweils zu einer DCM eine FCM, welcher das gesamte HA-Netzwerk (1) bzw. ein HA-Dienst (4) bzw. eine HA-Funktion (6) zugeordnet ist. Durch die fehlende Hierarchie zwischen DCM und FCM sind jedoch keine stufenübergreifenden Funktionalitäten und Informationen möglich.
- Bei Option 2 ist jeder HA-Dienst einer FCM und das gesamte HA-Netzwerk einer DCM zugeordnet. Die Auflistung aller FCMs der DCM entspricht damit zugleich einer Auflistung aller verfügbaren HA-Dienste.
- Bei Option 3 und 5 ist jede HA-Funktion einer FCM zugeordnet und diese einer einzelnen DCM, die das gesamte HA-Netzwerk repräsentiert (3) bzw. entsprechend der Zugehörigkeit zu einem HA-Dienst einer DCM, die einen HA-Dienst repräsentiert (5). Da gemäß HAVi-Spezifikation eine FCM jedoch nur einer DCM angehören kann, ermöglicht Option 5 keine Verwendung von HA-Funktionen durch mehr als einen HA-Dienst.
- Da bei Option 1 das gesamte HA-Netzwerk durch eine einzige DCM und FCM repräsentiert wird, kann die HAVi-Registry nicht zum Suchen nach dem Typ, Hersteller, oder Einsatzort eines HA-Dienstes genutzt werden. Dafür wäre eine eigene Abfrage-Schnittstelle notwendig.

- Bei Option 1, 3 und 6 ist eine Zuordnung von HA-Funktionen zu HA-Diensten nicht möglich, weil die zweite Stufe nicht auf DCMs oder FCMs abgebildet wird.
- Option 3, 5 und 6 führen zu einer hohen Anzahl an HAVi-Software-Elementen. Die Benutzerschnittstelle eines HAVi-Fernsehgerätes wird typischerweise über ein Auswahlfeld zur Anzeige aller Geräte (DCMs) im Netzwerk verfügen, die jedoch bei einer unnötig hohen Anzahl an DCMs sehr leicht unübersichtlich wird. Einen Extremfall stellt Option 6 dar, wo jede Lampe im Heimbereich durch eine eigene DCM und FCM repräsentiert wird.
- Da bei Option 4, 5 und 6 das HA-Netzwerk als Ganzes nicht auf DCM und FCMs abgebildet wird, können keine netzwerkweiten Einstellungen vorgenommen oder Informationen abgerufen werden. Beispielsweise lässt sich kein gemeinsames DDI-Target (HAVi-UI-Level 1) für das gesamte HA-Netzwerk bereitstellen.

Entsprechend dieser Bewertung ist Option 2 am besten zur Abbildung der Elemente des HA-Netzwerks auf HAVi-DCMs und -FCMs geeignet. Hierbei ist das gesamte HA-Netzwerk einer DCM (HA-DCM) und alle HA-Funktionen, die gemeinsam einen HA-Dienst bilden, sind einer FCM (HA-FCM) zugeordnet. Name, Typ und Einsatzort eines HA-Dienstes sind einige der Attribute, welche die HA-FCM bei der HAVi-Registry speichert. Damit können die standardisierten HAVi-Abfragemechanismen benutzt werden, um nach bestimmten Diensten zu suchen.

Das Service-Gateway nutzt die jeweilige System-Schicht zur Ermittlung der Kommunikationspartner, zur Erlangung von Informationen über diese und zur Kommunikation mit diesen. HAVi bietet mit der verteilten HAVi-Registry für alle Software-Elemente die Möglichkeit zur umfangreichen Selbstbeschreibung und zur Suche nach Software-Elementen. Im Gegensatz dazu ist die Selbstbeschreibung bei LON und EIB sehr beschränkt und die Bekanntmachung der Kommunikationsteilnehmer erfolgt durch die Verbindung der LON-Netzwerkvariablen bzw. die Zuweisung von EIB-Gruppenadressen im Rahmen der Netzwerkkonfiguration. Um der HAVi-Seite des Service-Gateways alle LON- bzw. EIB-Geräte und deren Eigenschaften bekannt zu machen, müssen die fehlenden Informationen dem Service-Gateway gesondert bereitgestellt werden. Entsprechend der im folgenden Unterabschnitt dargestellten HA-Erweiterung zur HAVi-Spezifikation werden für jede HA-FCM die folgenden Informationen benötigt:

1. Name, Einsatzort, Typ und Service-ID des HA-Dienstes und
2. alle für einen HA-Dienst notwendigen LON-Netzwerkvariablen bzw. EIB-Gruppenadressen und zu jeder einen passenden Funktionsnamen und Funktionstyp.

Da diese Daten im Rahmen der normalen, manuellen LON- bzw. EIB-Installation gesammelt werden können, kann der für die Kopplung notwendige Konfigurationsaufwand im Verhältnis zum LON- bzw. EIB-spezifischen Installationsaufwand durchaus vernachlässigt werden. Soll eine Kopplung mit einem bereits installierten Heim-Automatisierungs-Netzwerk eingerichtet werden, so kann die gesonderte Ermittlung der Daten

für die bestehende Installation allerdings einen wesentlichen Aufwand darstellen.

4.1.2 HA-Erweiterung zu HAVi 1.1

Da die HAVi-Spezifikation keine API für HA-Dienste definiert, wurde eine HA-Erweiterung zu HAVi 1.1 erstellt [Zie03a]. Die Erweiterung definiert eine Heim-Automatisierungs-FCM (HA-FCM), welche HA-Dienste anbietet und das HA-Netzwerk auf DCM und FCMs entsprechend der Option 2 von Tabelle 4.1 abbildet.

Während ein einfacher HA-Dienst zum Ein- und Ausschalten einer Lampe nur eine HA-Funktion benötigt, bestehen manche HA-Dienste aus mehreren oder komplexen HA-Funktionen, wie z. B. in [KCK⁺02] für Weiße Ware (Waschmaschine, Klimanlage, Mikrowelle, ...) beschrieben ist. Als Voraussetzung für die Interoperabilität der Geräte müssen die HA-Dienste der Geräte standardisiert sein. In der HA-Erweiterung wird die grundlegende Schnittstelle für alle HA-Dienste definiert, die Definition der spezifischen Eigenschaften der HA-Dienste und deren HA-Funktionen hat jedoch über zusätzliche HA-Dienst-Spezifikationen zu erfolgen (z. B. [Zie02]).

Service	Comm Type	Locality	Access	Resv Prot
HA::GetServiceInfo	M	global	all	
HA::GetServiceLocation	M	global	all	
HA::GetServiceFunctions	M	global	all	
HA::GetServiceState	M	global	all	
HA::GetValue	M	global	all	
HA::SetValue	M	global	all	yes
HAStateChanged	E	global	HA-FCM (all)	

Tabelle 4.2: HAVi-Dienste der HA-Erweiterung

Die grundlegende Schnittstelle der HA-FCM zeigt Tabelle 4.2, wobei die übliche Konvention zur Beschreibung der API eines Software-Elements von der HAVi-Spezifikation übernommen wurde. Die ersten drei HAVi-Dienste dienen zur Abfrage von HA-Dienst-Eigenschaften: dem Diensttyp, dem Einsatzort und den zugehörigen HA-Funktionen. Zumindest die ersten beiden Informationen können auch über eine Abfrage bei der HAVi-Registry ermittelt werden. Jeder HA-Dienst ist einem bestimmten, in der HA-Erweiterung definierten Dienst-Typ zugeordnet. Der Einsatzort wird über eine benutzerdefinierte Zeichenkette und optional auch über die Auswahl bestimmter vordefinierter Ortsangaben (Küche, Wohnzimmer, Vorzimmer, ...; Stockwerk; Raumnummer) bezeichnet. Eine HA-Funktion besitzt die Eigenschaften Name (Zeichenkette) und Typ (Aktor oder Sensor). Mit `GetServiceState` kann der aktuelle Zustand des HA-Dienstes abgefragt werden. Die nächsten beiden HAVi-Dienste, `GetValue` und `SetValue`, dienen zum Auslesen und Setzen von HA-Funktionswerten (Zugriff auf das HA-Netzwerk). Da ein

HA-Dienst aus mehreren HA-Funktionen bestehen kann, definiert der erste Parameter die gewünschte HA-Funktion (Unteradressierung). Diese Dienste können über das HAVi Messaging System über Nachrichten (Communication-Type = M) von beliebigen (Access = all) auch entfernten (Locality = global) Software-Elementen aufgerufen werden. Da mittels (SetValue) der Status der HA-FCM geändert werden kann, unterliegt dieser Befehl dem HAVi-Reservierungsschutz⁷. Der System-Event `HASStateChaged` soll von der HA-FCM (Access = HA-FCM) bei jeder Statusänderung über den HAVi Event Manager an alle Abonnenten im HAVi-Netzwerk (Locality = global, Access = (all)) gesendet werden.

In einer HA-Dienst-Spezifikation werden alle HA-Funktionen des betreffenden Dienstes definiert und der Dienst einem entsprechend Diensttyp⁸ der HA-Erweiterung zugeordnet. Neben der Anzahl und Reihenfolge der HA-Funktion wird für jede ein entsprechender Werttyp (`OCTET_SEQ`, `INT32`, `FLOAT`, `WSTRING`) festgelegt. Zur eindeutigen Identifikation des Dienstes wird von einer zentralen Stelle⁹ ein eindeutiger Kennzeichner (*Service-ID*) ausgestellt und der HA-Dienst-Spezifikation zugewiesen. Die Service-ID wird der HA-FCM beim Start, gemeinsam mit anderen Daten des HA-Gerätes, übergeben und ermöglicht dadurch die Ermittlung der dazugehörenden HA-Dienst-Spezifikation.

Eine HA-FCM ermöglicht damit jedem HAVi-Software-Element die Nutzung von HA-Geräten unabhängig vom Gerätehersteller. Beispielsweise kann eine HAVi-Anwendung alle Lampen eines Zimmers einschalten, indem bei der HAVi-Registry nach HA-FCMs gesucht wird, die Attribute `Location` und `ServiceType` mit dem gewünschten Zimmer und HA-Diensttyp verglichen werden und bei jeder Übereinstimmung die betreffende Lampe über `SetValue` eingeschaltet wird.

Ein HAVi-Software-Entwickler benötigt zumindest drei verschiedene Spezifikationen zur Erstellung einer HAVi-Anwendung, die einen HA-Dienst nutzen möchte: (1.) die HAVi-Spezifikation, (2.) die HA-Erweiterung und (3.) die entsprechende HA-Dienst-Spezifikation.

4.1.3 HAVi–LON Service-Gateway

Eine erste Realisierung dieser Kopplung ist auch in [ZW03] beschrieben. Die HW/SW-Architektur des Gateways ist in Abbildung 4.4 dargestellt. Die Hardware des Gateways besteht aus einem Windows 2000-PC mit einer Echelon LON-PCI-Karte mit Free-

⁷Im Wesentlichen sorgt der Reservierungsschutz dafür, dass nach einer Reservierung der HA-FCM, diese bei jedem Aufruf eines geschützten Befehls überprüft, ob das aufrufende Software-Element der aktuelle „Eigentümer“ der HA-FCM ist und den Zugriff für andere Software-Elemente ablehnt.

⁸Der `ServiceType` setzt sich aus einer Kennnummer für die Kategorie, `CatCode` (`ENERGY`, `WHITE_GOODS`, `LIGHTING`, ...), und einer Kennnummer für die Unterkategorie, `SubCatCode` (`HEATING`, `FRIDGE`, `LIGHT`, ...), zusammen.

⁹Ein geeigneter Kandidat für eine zentrale Vergabestelle der Service-ID ist HAVi Inc., da diese Herausgeber der HAVi-Spezifikation ist und auch die Weiterentwicklung von HAVi koordiniert (Technical Steering Group, Technical Maintenance Working Group, ...).

Topology-Transceiver [Ech01] und einer OHCI IEEE 1394-PCI-Karte. Die Software des Gateways besteht aus einem HAVi-Teil (linke Seite) und einem LON-Teil (rechte Seite).

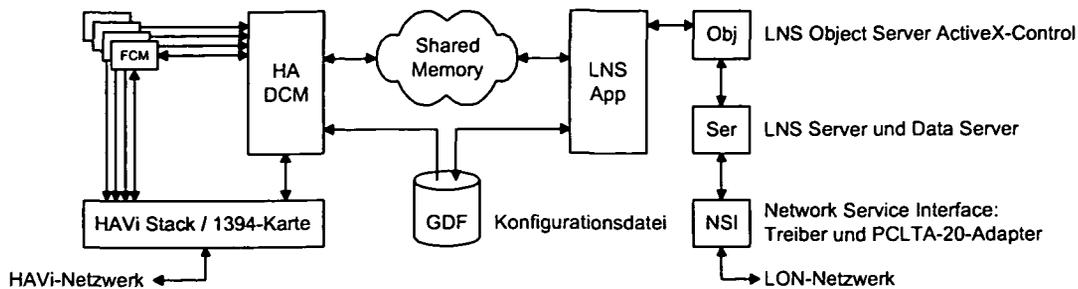


Abbildung 4.4: HAVi-LON Service-Gateway

Die Kommunikation zwischen den beiden als eigenständige Prozesse realisierten Teilen erfolgt einerseits über Shared-Memory, synchronisiert über Semaphore [Sta98, CW98], und andererseits über eine Konfigurationsdatei (*Gateway-Definition-File*), welche bei der Installation des LON-Netzwerks erstellt und bei Hinzufügen und Entfernen von LON-Geräten aktualisiert wird. Dadurch ist es möglich, die Informationen zur LON-Netzwerkkonfiguration auch unabhängig vom Gateway erstellen zu können, z. B. durch ein LON-Netzwerk-Management-Werkzeug. Die beiden Teile im Einzelnen:

1. LON-Teil

Der LON-Teil besteht aus einer LNS-Anwendung, die den Echelon LNS-Server¹⁰ [Ech00] für den Zugriff auf den LON-Bus nutzt. Der Zugriff auf den LNS-Server erfolgt über ein ActiveX-Control. Die LNS-Anwendung ist verantwortlich für (1) die Benachrichtigung der HA-DCM über hinzugefügte und entfernte LON-Geräte, (2) die Benachrichtigung der entsprechenden HA-FCM (via HA-DCM) von Netzwerkvariablen-Aktualisierungen, (3) das Lesen des aktuellen Wertes von einer bestimmten Netzwerkvariablen bei Anfrage durch eine der zugehörigen HA-FCMs (via HA-DCM) und (4) das Schreiben eines neuen Wertes an eine bestimmte Netzwerkvariable bei Anfrage durch eine der zugehörigen HA-FCMs (via HA-DCM). Für das Überwachen und Senden von Netzwerkvariablen-Aktualisierungen wird die Single-Point-Monitoring-Funktion des LNS-Servers verwendet. Die Netzwerkvariable ist durch die HA-FCM und HA-Funktion genau festgelegt.

2. HAVi-Teil

Der HAVi-Teil besteht aus einer HAVi-Anwendung, dem HAVi-Stack und einer embedded DCM und FCM, die mehrfach instanziiert werden können. Unmittelbar nach dem Start des HAVi-Stacks erfolgt die Installation der HA-DCM, die ihrerseits für das Starten und Entfernen von geeigneten HA-FCMs entsprechend den im HA-Netzwerk vorhandenen HA-Diensten verantwortlich ist. Jeder HA-FCM werden bei der Installation die HA-spezifischen Daten übergeben. Die dazu notwen-

¹⁰LNS steht für LonWorks Network Services

digen Informationen¹¹ werden über die Konfigurationsdatei (Gateway-Definition-File) bereitgestellt.

Der Test der Kopplung erfolgte anhand eines Demo-Aufbaus, bestehend aus (1) zwei PCs zur Emulation eines HAVi-TV und einer HAVi-SetTop-Box, (2) dem HAVi-LON Service-Gateway und (3) mehreren LON-Knoten zum Schalten von Lampen, Steuern einer Jalousie und zum Melden eines Wasserlecks.

4.1.4 HAVi-EIB Service-Gateway

Die HW/SW-Architektur des Gateways ist in Abbildung 4.5 dargestellt. Die Hardware des Gateways besteht aus einem Linux PC mit angeschlossenem EIB-RS232-Anwendungsmodul auf einer BCU1 [Sie01] und einer OHCI IEEE 1394-Karte. Die Software des Gateways besteht aus einem HAVi-Teil (linke Seite) und einem EIB-Teil (rechte Seite).

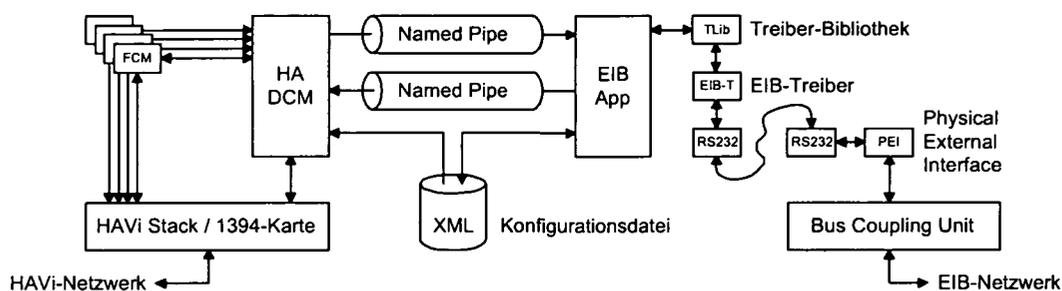


Abbildung 4.5: HAVi-EIB Service-Gateway

Die Kommunikation zwischen den beiden als eigenständige Prozesse realisierten Teilen erfolgt einerseits über Named-Pipes [Sta98, Her99] und andererseits über eine XML-basierte Konfigurationsdatei, welche bei der Installation des EIB-Netzwerks erstellt und bei Hinzufügen und Entfernen von EIB-Geräten aktualisiert wird. Dadurch ist es möglich, die Informationen zur EIB-Netzwerkconfiguration auch unabhängig vom Gateway erstellen zu können, z. B. mit der EIB Tool Software (ETS-2). Die beiden Teile im Einzelnen:

1. EIB-Teil

Der EIB-Teil besteht aus einer EIB-Anwendung, einem Linux EIB-Treiber [Tha01], einem RS232-Anwendungsmodul und einer BCU1. Die EIB-Anwendung kommuniziert als externe (ausserhalb der BCU ausgeführte) Anwendung unter Nutzung des EIB-Treibers asynchron, seriell über das Physical External Interface (PEI-Typ 16) gemäß dem External Message Interface (EMI) [EIB99] mit der BCU. EMI ermöglicht einer externen Anwendung den Zugriff auf einzelne EIB-Kommunikationsschichten der BCU. Für die Kommunikation mit den anderen EIB-Knoten wird

¹¹siehe Abschnitt 4.1.1

die BCU in den Operationsmodus „Data-Link-Layer, Normal-Mode“ gesetzt (`PC_Set_Value.req`) und mittels der Data-Link-Dienste `L_Data.req`, `L_Data.ind` EIB-Pakete gesendet und empfangen. Die Zusammenstellung geeigneter Pakete (z. B. Gruppentelegramm zum Schreiben des Objektwertes eines Gruppen-Kommunikationsobjekts) und die Analyse empfangender Pakete erfolgt in der EIB-Anwendung. Damit das Gateway die Änderungen von Gruppen-Kommunikationsobjekten aller EIB-Knoten mitbekommt, muss die BCU entweder Teilnehmer aller Gruppen sein oder in den Operationsmodus „Data-Link-Layer, Bus-Monitor-Mode“ gesetzt werden. Ersteres ist jedoch auf Grund einer Einschränkung der BCU-Adresstabelle auf 254 Gruppen beschränkt und zweiteres erfordert das häufige Umschalten zwischen Bus-Monitor-Mode und Normal-Mode, weil im Bus-Monitor-Mode keine Pakete gesendet werden können. Abhilfe schafft jedoch die Möglichkeit, die automatische Filterung der empfangenen Pakete nach der physikalischen Adresse und den angehörenden Gruppenadressen der BCU abzuschalten, indem die Länge der BCU-Adresstabelle auf Null gesetzt wird.

Die EIB-Anwendung ist verantwortlich für (1) die Benachrichtigung der HA-DCM über hinzugefügte und entfernte EIB-Geräte, (2) die Benachrichtigung der entsprechenden HA-FCM (via HA-DCM) bei Änderung des Wertes eines Kommunikationsobjektes und (3) das Schreiben eines neuen Wertes eines Kommunikationsobjektes bei Anfrage durch eine der zugehörigen HA-FCMs (via HA-DCM). Die Gruppenadresse ist durch die HA-FCM und HA-Funktion genau festgelegt.

2. HAVi-Teil

Der HAVi-Teil besteht aus einer HAVi-Anwendung, dem HAVi-Stack und einer embedded DCM und FCM, die mehrfach instanziiert werden können. Unmittelbar nach dem Start des HAVi-Stacks erfolgt die Installation der HA-DCM, die ihrerseits für das Starten und Entfernen von geeigneten HA-FCMs entsprechend den im HA-Netzwerk vorhandenen HA-Diensten verantwortlich ist. Jeder HA-FCM werden bei der Installation die HA-spezifischen Daten übergeben. Die dazu notwendigen Informationen¹² werden über die XML-basierte¹³ Konfigurationsdatei bereitgestellt.

Eine Realisierung des Gateways durch zwei Prozesse begründet sich unter anderem darin, dass der EIB-Treiber der General Public License (GPL) unterliegt und diese eine Einbindung (Verwendung in einem gemeinsamen Prozess) in eine nicht der GPL unterliegenden Software, wie dem verwendeten HAVi-Stack, verbietet.

Der Test der Kopplung erfolgte anhand eines Demo-Aufbaus, bestehend aus zwei PCs zur Emulation eines HAVi-TV und einer HAVi-SetTop-Box, dem HAVi-EIB Service-Gateway und mehreren EIB-Knoten zum Schalten und Dimmen mehrerer Lampen und zum Melden eines Wasserlecks.

¹²siehe Abschnitt 4.1.1

¹³Zum Lesen und Schreiben der Konfigurationsdatei wird die XML-Bibliothek Libxml2 [lib] verwendet.

4.1.5 Ergänzende Anmerkungen

Der HAVi-Teil könnte unabhängig vom angekoppelten Feldbus sein, wenn die Schnittstelle zwischen den Prozessen standardisiert ist, wenn also z. B. generell Named-Pipes und eine XML-basierte Konfigurationsdatei (anstelle von Shared Memory und einem proprietären Gateway-Definition-File) verwendet werden.

Im Gegensatz zur CE-HA-Kopplung sind zur Anbindung von Geräten der Heim-Automatisierung an Netzwerke des Bereichs Small Office/Home Office verschiedene Vorschläge publiziert worden: Jini-EIB [RHB02], UPnP-EIB [WG00], Jini-LON [KL01], Jini-EIB [KK99].

Die Anbindungen sind als transparente Service-Gateways realisiert, indem jedes EIB- bzw. LON-Gerät durch eine entsprechende Repräsentation (Jini-Service, UPnP-Device) auf der Jini- bzw. UPnP-Seite vertreten ist und über diese wie ein Gerät des eigenen Netzwerks genutzt werden kann. Bei allen Anbindungen ist jedoch die API-Umsetzung auf eine sehr einfache Funktionalität beschränkt. Die ersten beiden Anbindungen können außerdem als dynamische Kopplungen klassifiziert werden, weil bei der ersten Kopplung ein *Service-Controller* für die automatische Instanziierung von geeigneten Jini-Diensten oder deren Entfernung bei Hinzufügen oder Entfernen von EIB-Geräten sorgt und bei der zweiten Kopplung die jeweils benötigten UPnP-Dienste automatisch aus der Projektdatenbank der EIB Tool Software (ETS) erzeugt werden. Die anderen beiden Kopplungen sind als statisch zu klassifizieren, weil sie auf eine spezielle Umgebung bzw. Gerätekonfiguration beschränkt sind.

4.2 Kopplung von Unterhaltungselektronikclustern

In diesem Abschnitt werden ein Konzept für eine dynamische HAVi=HAVi System-Kopplung zur Verbindung von entfernten HAVi-Netzwerken über ein TCP/IP-basiertes Transitnetzwerk und eine Implementierung einer dynamischen HAVi-AV/C Service-Kopplung zur Nutzung von AV/C-Geräten über ein HAVi-Netzwerk beschrieben.

4.2.1 HAVi-HAVi Kopplung via TCP/IP-Tunnel

Die Zielsetzung dieser Kopplung ist die transparente Verbindung von HAVi-Netzwerken über ein TCP/IP-basiertes Transitnetzwerk (siehe auch Abbildung 4.6). Da hierbei die zu verbindenden HAVi-Netzwerke und das gemeinsame Transitnetzwerk unterschiedliche System-Schichten besitzen und am Transitnetz keine Nutzung der Dienste der angeschlossenen HAVi-Netzwerke vorgesehen ist, kann die Kopplung jeweils über System-Gateways an den Anbindungsstellen zum Transitnetz erfolgen. Wie in Abbildung 4.7 dargestellt, erfolgt der Einsatz der System-Gateways jeweils paarweise, d. h. in Abhängigkeit von der Kommunikationsrichtung ist ein System-Gateway jeweils für die Umsetzung der HAVi System-Schicht in die System-Schicht des Transitnetzwerks und ein anderes für

die Rückumsetzung in die HAVi System-Schicht verantwortlich. Um von jedem beliebigen Gerät eines angebundenen HAVi-Netzwerks die Dienste eines anderen angebundenes HAVi-Netzwerks nutzen zu können, müssen die System-Gateways außerdem zweiseitig und transparent sein. Für die Kommunikation auf der Service-Schicht ist keine Umsetzung notwendig, da alle an einer Kommunikation beteiligten Geräte entweder über die selbe Service-Schicht verfügen oder aber mit ihnen keine Kommunikation auf der Service-Schicht vorgesehen ist. In Abbildung 4.7 ist dies durch eine durchgehende Linie (zwischen App und DF, HAVi Service API) dargestellt.

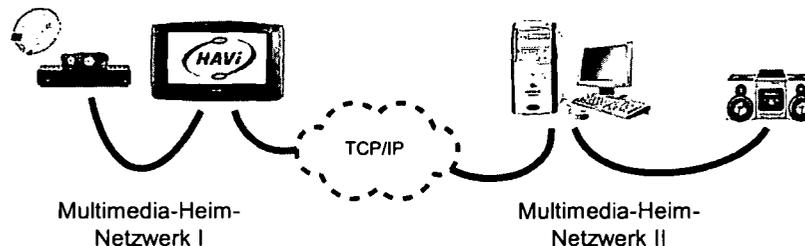


Abbildung 4.6: HAVi-HAVi Kopplung via TCP/IP-Tunnel

Mögliche Benutzerszenarien für diese HAVi-HAVi Kopplung sind:

- Verbindung mehrerer HAVi-Netze
 - im Heimbereich (Wohnung, Haus, Auto)
 - von unterschiedlichen Orten, z. B. für den Datenaustausch oder Spiele.
- Verbindung eines einzelnen HAVi-Gerätes mit einem entfernten HAVi-Netzwerk
 - für den Zugriff auf das Heimnetzwerk von außen, z. B. vom Büro, Auto oder Hotel.
 - als Wireless Erweiterung des lokalen HAVi-Netzwerks, z. B. intelligente Fernbedienung, PDA oder Handy.

4.2.1.1 Softwarearchitektur der System-Gateways

In Abbildung 4.8 sind die wesentlichen SW-Komponenten des System-Gateways dargestellt. Die Verbindung über das Transitnetz erfolgt über den *Interconnection-Manager* (IMa). Die IMa der einzelnen Gateways kommunizieren untereinander über ein so genanntes „Buddy-Protokoll“, das über Mechanismen zum Aufbau (JOIN), zur Aufrechterhaltung (KEEP ALIVE) und zum Abbau (LEAVE) von permanenten Punkt-zu-Punkt Verbindungen und zum Weiterleiten von HAVi-Nachrichten (FORWARD) über diese Verbindungen verfügt. Zu den Verbindungen sind unterschiedliche Zugriffsstufen definiert und für jede Verbindung wird ein Anschlussstatus gewartet. Der *Export-Manager* (EMa)

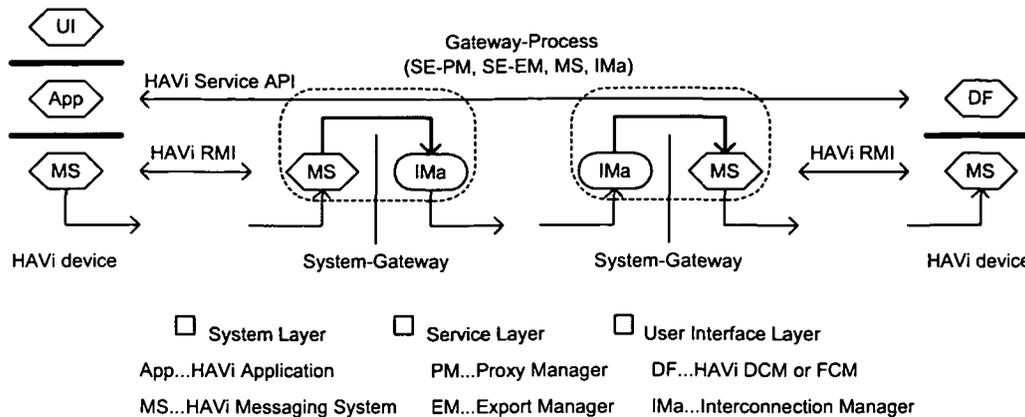


Abbildung 4.7: HAVi-HAVi Kopplung über System-Gateways

ist für den Export lokaler HAVi Software-Elemente (SE) in entfernte HAVi-Netzwerke zuständig. Er kontrolliert damit einerseits, auf welche lokale SE von welchen entfernten HAVi-Netzwerken zugegriffen werden kann, und andererseits, welche lokale SE mit welchen entfernten SE kommunizieren dürfen. Damit hier keine umfangreiche manuelle Konfiguration notwendig ist, sind verschiedene Standardeinstellungen verfügbar. Beispielsweise legt die Einstellung `EXPORT_DCMFCM_TRUSTED_ONLY` fest, dass alle lokalen DCMs und FCMs in allen angeschlossenen HAVi-Netzwerken mit Anschlussstatus `CONNECTED_TRUSTED` bekannt gemacht werden sollen und von dort auf sie zugegriffen werden kann. Entsprechend dieser Einstellungen exportiert der EMa die SE in die entfernten HAVi-Netze. Dazu wird in den entsprechenden entfernten HAVi-Netzwerken für jedes zu exportierende SE ein *SE-Proxy* (P) erzeugt, d. h. ein P im entfernten Netz instantiiert und dort beim Messaging-System und der Registry angemeldet. Die Attribute der SE erhält der EMa über die Registry im Heimatnetz und fügt als zusätzliches Attribut (`EXPORT_MANAGER`) die eigene SEID hinzu. Dieses Attribut dient als *Netzwerk-kennung* des Heimatnetzes und wird von den EMa benutzt, um einen P zu erkennen und dadurch zu vermeiden, dass dieses SE erneut exportiert wird. Ohne diese Vorkehrung könnte es zu einem fortlaufenden wechselseitigen Export ein und des selben SE kommen („Ping-Pong-Spiel“ zwischen den EMa) bis der System-Speicher in den Gateways voll ist. Die Verwaltung der Proxy und deren Verbindung zum IMa erfolgt über den *SE-Proxy-Manager* (PMa). Aus Geschwindigkeitsgründen erfolgt die Kommunikation zwischen Proxy - PMa, PMa - IMa, PMa - EMa und EMa - IMa direkt, d. h. nicht über das HAVi Messaging System. Die Kommunikation zu einem entfernten PMa erfolgt über Befehlscodes, beispielsweise signalisiert ein EMa mit `CREATE_PROXY`, dass ein neuer Proxy erzeugt werden soll.

Die von einem entfernten HAVi-Netzwerk durch jeweils einen SE-Proxy (P) importierten SE stehen lokal wie jedes andere SE zur Verfügung. Der P ist unabhängig vom jeweils

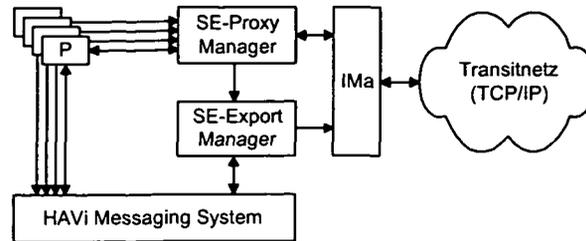


Abbildung 4.8: HAVi-HAVi Kopplung: Komponenten

repräsentierten SE¹⁴, er leitet an ihn gerichtete Anfragen ohne Analyse des Nachrichteninhalts über den PMA an das zugehörige Original-SE weiter und die über den PMA empfangenen Antworten oder Benachrichtigungen an den lokalen Aufrufer zurück bzw. an den lokalen Abonnenten weiter. Erkennt der EMA, dass ein exportiertes SE entfernt wird, so veranlasst er die Entfernung der zugehörigen SE-Proxys in den entfernten HAVi-Netzen. Damit funktioniert auch die Überwachung von nicht lokalen SE (Message Supervision, *MsgWatchOn*). Um den exportierten SE auch die Überwachung und die Ermittlung der Zugriffsberechtigung (*trusted*, *untrusted*, *owner*, ...) ihrer nicht lokalen Aufrufer zu ermöglichen, müssen diese aufrufenden SE in das Heimatnetz des aufgerufenen entfernten SE durch einen P importiert werden. Der dafür verantwortliche EMA veranlasst, soweit das SE nicht ohnehin bereits exportiert wurde, die Erzeugung eines P, welcher jedoch nicht bei der Registry bekannt gemacht wird und auch als *Client-Proxy* (CP) bezeichnet wird.

4.2.1.2 Kommunikationsablauf

Der Kommunikationsablauf zwischen einer Anwendung und einer exportierten DCM ist in Abbildung 4.9 als Beispiel dargestellt. Eine Anwendung (APP, linke Seite, oben) sendet über das HAVi Messaging System eine Nachricht an den SE-Proxy (DcmP) der entfernten DCM. Dieser kennt die SEID und die *Netzwerkennung* (SEID vom EMA des Heimatnetzwerks) der zugehörigen DCM und übergibt diese gemeinsam mit der Anfrage an den PMA. Der PMA stellt über den EMA sicher, dass im Heimatnetzwerk der DCM ein Client-Proxy für die aufrufende Anwendung (AppP) existiert, gegebenenfalls veranlasst der EMA den PMA im Heimatnetzwerk der DCM einen zu erzeugen. Dann leitet der PMA die Daten an den IMA zur Übertragung über das Transitnetz weiter. Der IMA ist über Punkt-zu-Punkt-Verbindungen mit den IMA anderer HAVi-Netze verbunden und speichert zu jeder Verbindung die Netzwerkennung des angeschlossenen HAVi-Netzwerks, den Anschlussstatus und Verbindungsinformationen. Anhand dieser Informationen sendet er die Daten über den TCP/IP-Tunnel an den IMA des Zielnetzwerks. Dieser leitet die empfangenen Daten an den dortigen PMA weiter. Der PMA verwaltet eine Liste über alle von ihm erzeugten P, die jeweils die SEID des P und eine Referenz auf den P enthält. Anhand der Referenz kann er die Anfrage direkt an den SE-Proxy des Aufrufers (AppP)

¹⁴Eine Ausnahme davon bilden die System-Software-Elemente, wie weiter unten noch ausgeführt wird.

übergeben. Der AppP speichert die HAVi-Transaktionsnummer (*transId*) der erhaltenen Anfrage für das spätere Zurücksenden der Antwort und sendet die Anfrage über das HAVi Messaging System an die DCM (rechte Seite, oben) und wartet auf Antwort (*MsgSendRequestSync*).

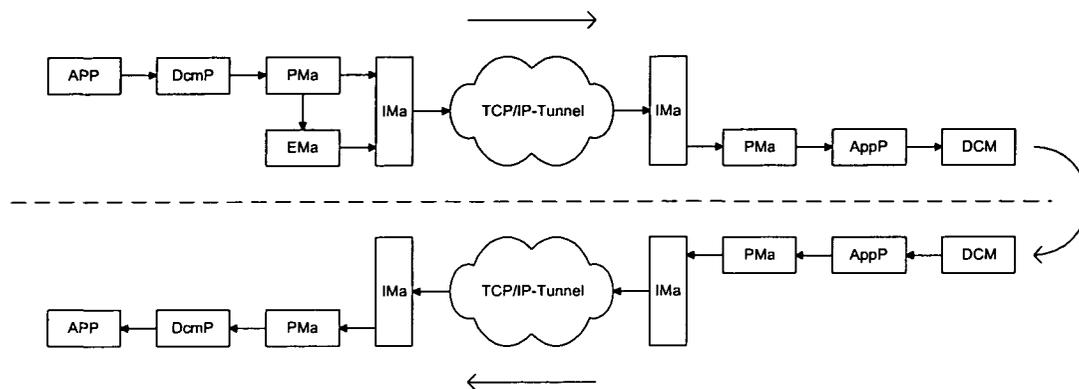


Abbildung 4.9: HAVi-HAVi Kopplung: Kommunikationsablauf

Die DCM verarbeitet die Anfrage und sendet eine Antwort an den AppP über das HAVi Messaging System zurück (rechte Seite, unten). Nach Austausch der HAVi-Transaktionsnummer leitet der AppP die Antwort über den lokalen PMa, die beiden IMa und den PMa an den DcmP weiter. Als letzter Teil der Transaktion sendet der DcmP die erhaltene Antwort über das HAVi Messaging System an die Anwendung (APP, linke Seite, unten).

4.2.1.3 Spezialfall System-Software-Elemente

Die HAVi-Spezifikation definiert für alle Dienste, welche SE diese benutzen dürfen (*system*, *trusted*, *all*, *Stream Manager*, ...). Speziell die Zugriffsbeschränkungen auf bestimmte System-SE (*Stream Manager*, *Resource Manager*, ...) sind gesondert zu berücksichtigen, weil die Beschränkung im betreffenden SE anhand des Software-Handles (*swHandle*) der SEID bestimmt wird und der Client-Proxy (in Abbildung 4.9 als AppP bezeichnet) als vermeintliche Quelle der Anfrage deshalb keine beliebige SEID besitzen darf. Soll beispielsweise eine Reservierung einer FCM erfolgen, so muss die Anfrage *Fcm::Reserve* eines Resource Managers an eine entfernte FCM im Heimnetzwerk der FCM durch den Resource Manager am System-Gateway gesendet werden. Anderenfalls würde die FCM die Anfrage mit dem Fehlercode *EACCESS_VIOLATION* abweisen. Darüber hinaus ist es in manchen Fällen nötig, dass das System-SE Kenntnis von der in seinem Namen (Quell-SEID) gesendeten Anfrage erhält, weil sonst ein inkonsistenter Zustand eintreten könnte. Dies soll durch das folgende Beispiel veranschaulicht werden.

Die Anfrage *Dcm::Connect*, die nur von Stream Managern aufgerufen werden darf, bewirkt, dass eine Geräteverbindung innerhalb einer DCM erstellt wird und der aufrufende

Stream Manager in der DCM für diese Verbindung gespeichert wird. Wird vom selben Stream Manager die gleiche Verbindung ein zweites Mal erstellt, so wird die Verbindung in der DCM überlagert, der Stream Manager aber nicht erneut gespeichert. Es liegt in der Verantwortung des Stream Managers, zu jeder von ihm erstellten Geräteverbindung einen eigenen Verbindungszähler zu verwalten und nur beim Abbau der jeweils letzten Geräteverbindung (Verbindungszähler gleich Null) den Abbaubefehl `Dcm::Disconnect` an die entsprechende DCM zu senden. Für den Fall, dass ein Stream Manager *SM-A* im HAVi-Netz *A* eine Geräteverbindung in einer DCM *DCM-A* des selben Netzes erstellt und später ein Stream-Manager *SM-B* im HAVi-Netz *B* die selbe Verbindung über den Proxy der an das Netz *B* exportierten *DCM-A*, weiteren Kopplungskomponenten und letztendlich durch den *SM-A* als Client-Proxy des *SM-B* erstellt, wird die Verbindung ordnungsgemäß überlagert, jedoch der Verbindungszähler für diese Geräteverbindung im *SM-A* nicht erhöht. Beim Abbau der ursprünglich vom *SM-A* veranlassten Geräteverbindung der *DCM-A* wird der Verbindungszähler auf Null erniedrigt und damit die *DCM-A* veranlasst die Geräteverbindung abzubauen, obwohl die vom *SM-B* veranlasste Geräteverbindung weiterbestehen sollte¹⁵.

Das angesprochene Problem kann jedoch dadurch verhindert werden, dass System-SE nicht nur die Anfragen als einfache Client-Proxys für entfernte System-SE weiterleiten, sondern die erhaltenen Anfragen auch analysieren und entsprechend darauf reagieren (z. B. im Fall des Stream Managers den Verbindungszähler aktualisieren).

Für eine vollständige transparente Kopplung ist es deshalb nötig, dass einerseits die System-SE am System-Gateway als spezielle Client-Proxys für entfernte System-SE agieren und andererseits für nicht am System-Gateway vorhandene System-SE Client-Proxys mit entsprechenden System-SEIDs erzeugt werden. Die Kommunikation zwischen dem PMA und den als Client-Proxy agierenden System-SE läuft, wie auch die andere interne Kommunikation des System-Gateways, direkt und nicht über das HAVi Messaging System.

4.2.1.4 System-Schicht des Transitnetzwerks

Die Kommunikation zwischen den Interconnection-Manager der System-Gateways erfolgt über TCP/IP anhand des weiter oben beschriebenen „Buddy“-Protokolls. Das Nachrichtenformat ist in Tabelle 4.3 dargestellt¹⁶. Als `DestSEID` wird die SEID des EMA vom Zielnetz und als `SourceSEID` die SEID des lokalen EMA angegeben. Die am Gateway auszuführende Aktion (`JOIN`, `KEEP ALIVE`, `FORWARD`, ...) wird durch den `OperationCode` festgelegt. Über die `ControlFlags` wird eine Anfrage und eine Antwort unterschieden. Die `TransactionId` dient für die Zuordnung einer Antwort zu ihrer Anfrage, sie wird dazu bei der Anfrage gewählt und bei der Antwort zurückgesendet. Das Feld `ReturnErrCode` ist Teil der Antwort und gibt den Status des Befehls an. `ParamLength` definiert

¹⁵In diesem Beispiel wird ohne Beschränkung der Allgemeinheit davon ausgegangen, dass *SM-A* der einzige Stream Manager im Netz *A* ist, der die angesprochene Geräteverbindung in der *DCM-A* erzeugt hat.

¹⁶*uimsbf* steht für „unsigned integer most significant bit first“

die Länge der Parameter in Bytes. Die Parameter der Funktionen werden zur einheitlichen Interpretation in einen genormten Bytestrom gemäß HAVi 1.1 umgewandelt. Dazu werden die HAVi-Marshal- bzw. -Unmarshal-Funktionen verwendet.

Feld	Bits	Übertragung
DestSEID	80	uimsbf
SourceSEID	80	uimsbf
OperationCode	24	uimsbf
ControlFlags	8	uimsbf
TransactionId	32	uimsbf
ReturnErrCode	16	uimsbf
ParamLength	32	uimsbf
Parameter	–	uimsbf

Tabelle 4.3: Gateway-Kommunikation: Nachrichtenformat

Die Bekanntmachung der IP-Adressen der Interconnection-Manager der anzubindenden Netzwerke erfolgt schon aus Sicherheitsgründen manuell. Als Netzwerkkennung wird die SEID der Export-Manager verwendet, die bei erfolgreichem Verbindungsaufbau zurückgeliefert wird. Über Alive-Nachrichten (*KEEP ALIVE*) kann die Existenz der angebotenen Netze überwacht werden. Die maximale Erkendauer der Trennung von einem entfernten Netz ist dabei gleich dem beim Verbindungsaufbau ausgehandelten Sendeintervall für Alive-Nachrichten.

4.2.2 HAVi–AV/C Kopplung

Die Zielsetzung dieser Kopplung ist die Einbindung von AV/C-Geräten eines bestimmten Typs (z. B. DVB-S Tuner, Audio/Video-Hard-Disc-Drive) in ein HAVi-Netzwerk. Die AV/C-Geräte sollen ohne zusätzliche Konfigurationen durch den Benutzer unmittelbar nach dem Anstecken an den IEEE 1394-Bus über das HAVi-Netzwerk verwendbar sein.

Aus der Zielsetzung folgt, dass sich für diese Art der Verbindung eine dynamische HAVi–AV/C Service-Kopplung eignet, denn diese transparente und einseitige Kopplung auf der Service-Schicht ermöglicht den Geräten des HAVi-Netzwerks die Nutzung der AV/C-Geräte über die HAVi-API und sorgt für die dazu notwendige API-Umsetzung (HAVi → AV/C). Aus der Gegenüberstellung der beiden Vernetzungskonzepte anhand der Tabellen 3.3 und 3.4 folgt, dass beide eng auf den IEEE 1394-Bus abgestimmt sind und auf der Service-Schicht AV/C-Units durch HAVi-DCMs und AV/C-Subunits durch HAVi-FCMs repräsentiert werden können. Durch die gemeinsame Verwendung des IEEE 1394-Busses können die für eine dynamische Kopplung neben der Transparenz notwendigen Bedingungen (Unabhängigkeit von einer speziellen Umgebung und ein niedri-

ger Konfigurationsaufwand) grundsätzlich ohne größeren Aufwand erfüllt werden.

4.2.2.1 Softwarearchitektur des Service-Gateways

Ein AV/C-Gerät wird gemäß HAVi-Spezifikation als „1394 Legacy Audio Video (LAV)“-Gerät eingestuft, weil es über den IEEE 1394-Bus angeschlossen ist, aber kein HAVi-konformes Configuration-Rom (SDD) besitzt. Als Service-Gateway kann jedes IAV- oder FAV-Gerät agieren, das eine entsprechende DCM-Code-Unit (DCU¹⁷) für das anzubindende AV/C-Gerät laden und ausführen kann. Die von der DCU installierte DCM und die dazugehörigen FCMs sorgen für die Umsetzung der an sie gerichteten HAVi-Befehle in geeignete AV/C-Befehle an die Unit bzw. Subunit des entsprechenden AV/C-Gerätes. Die Installation der DCU für Geräte mit IEEE 1394-Anschluss erfolgt durch einen DCM-Manager. Einen DCM-Manager besitzt jedes FAV-Gerät und jedes IAV-Gerät, das eine DCU für ein fremdes IEEE 1394-Gerät ausführen kann. Da im ganzen Netzwerk für ein Gerät nur genau eine DCM installiert werden darf, bilden alle DCM-Manager im HAVi-Netzwerk ein verteiltes DCM-Management-System, welches die Einhaltung dieser Bedingung für IEEE 1394-Geräte gewährleistet.

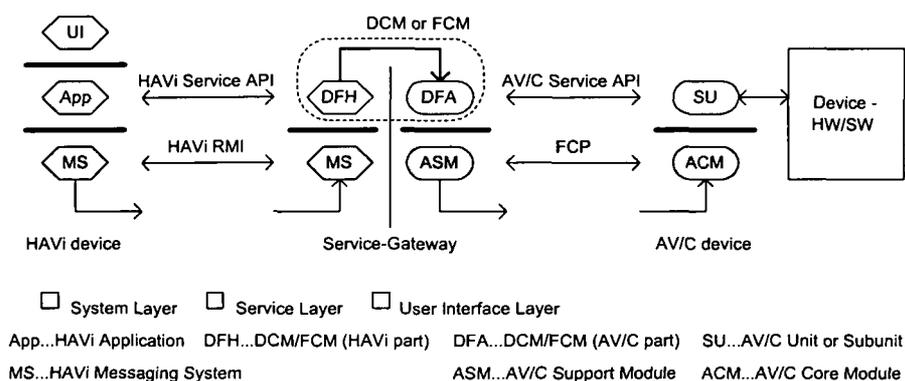


Abbildung 4.10: HAVi-AV/C Service-Gateway

Abbildung 4.10 zeigt den Kommunikationsablauf zwischen einem HAVi-Nutzer und dem benutzten AV/C-Gerät über das implementierte Service-Gateway. Als Gateway wird ein IAV-Gerät verwendet und dort für jeden zu unterstützenden AV/C-Gerätetyp eine „embedded DCU“ (d. h. hierbei eine in ANSI-C geschriebene DCU) bereitgestellt. Über die von den embedded DCU jeweils bereitgestellten Funktionen `DCU_IsGuest-Supported`, `DCU_InstallDCU`, `DCU_UninstallDCU` kann der DCM-Manager des IAV ermitteln, ob ein neues Gerät durch eine der vorhandenen DCU unterstützt wird und in Abstimmung mit den anderen DCM-Managern im Netzwerk diese auch installieren bzw. deinstallieren. Die Ermittlung des unterstützten AV/C-Gerätetyps kann von den DCU anhand von mehreren Merkmalen erfolgen:

¹⁷siehe auch Seite 25

- AV/C-Unit-Directory im IEEE 1394-Configuration-Rom → AV/C-Gerät
- AV/C-Unit-Befehl `UnitInfo` → Unit-Typ
- AV/C-Unit-Befehl `SubUnitInfo` → Anzahl und Typ der enthaltenen Subunits

Da das Gateway als einzelner Prozess realisiert ist, erfolgt die Kommunikation zwischen den unterschiedlichen Service-Schichten über direkte Funktionsaufrufe. Die DCM und die zugehörigen FCMs können in einen HAVi-Teil (DFH) und einen AV/C-Teil (DFA) unterteilt werden. Zur einfacheren Kommunikation mit den AV/C-Geräten wird ein so genanntes AV/C-Support-Module (ASM) eingesetzt. Dieses besteht wiederum aus mehreren Komponenten: einem AV/C-Core-Modul (ACM), das Funktionen zum Senden und Empfangen von AV/C-Paketen bietet und darauf aufbauend Unit- und Subunit-spezifische Module, die Wrapper-Funktionen für die Kommunikation mit AV/C-Units bzw. -Subunits bereitstellen. Die Wrapper besitzen als Funktionsparameter neben den Adressinformationen (GUID des Zielknotens und soweit notwendig Subunit-Typ und Subunit-ID zur Unteradressierung) und einem Timeout-Wert die Eingangs- und Ausgangsparameter des jeweiligen AV/C-Befehls und dessen Antwort. Aus den Eingangsparametern wird ein geeignetes AV/C-Befehls-Paket erstellt und über den IEEE 1394-Bus an das adressierte AV/C-Gerät gesendet, auf die Antwort gewartet und anhand dieser die Ausgangsparameter gesetzt.

4.2.2.2 API-Umsetzung

Da HAVi und AV/C dem selben Anwendungsbereich entstammen, gibt es für die meisten Geräte des Bereichs auf beiden Seiten entsprechende APIs und es kann grundsätzlich eine $n : m$ -Umsetzung der vorhandenen Funktionen erfolgen. So ist etwa die API zur Ansteuerung eines Tuners sowohl bei HAVi (Tuner-FCM) als auch bei AV/C (Tuner-Subunit) definiert. Die Auswahl eines Fernseh- oder Radioprogramms bei einem AV/C-Tuner durch eine HAVi-Anwendung erfordert im Wesentlichen die Umsetzung der HAVi-Funktion `Tuner::SelectService` auf die AV/C-Funktion `TunerSubunit::DirectSelectInformationType`. Oft ist die Umsetzung jedoch nicht 1 : 1 möglich, weil die Schnittstellen sich hinsichtlich der Anzahl der Funktionen und ihrer Parameter unterscheiden oder bestimmte für die Ausführung einer Funktion notwendige Informationen nicht im Gateway gespeichert werden und zuerst vom AV/C-Gerät gelesen werden müssen. Die API-Umsetzung von einigen HAVi-AvDisc-Befehlen [HAV01] in entsprechende AV/C-Befehle zur Ansteuerung einer AV/C-Disc-Subunit [13902c, 13902b, 13902a] ist in Tabelle 4.4 aufgelistet. Je nach Funktionsparameter kommen alle oder auch nur Teile der angeführten AV/C-Befehle zur Anwendung.

4.2.2.3 Ergänzende Anmerkungen

Das AV-Streaming (Echtzeit-Übertragung von Audio/Video-Daten) zwischen AV/C- und HAVi-Geräten ist grundsätzlich ohne Einschränkung auf Grund der gemeinsamen

HAVi-Funktion	AV/C-Funktion
GetItemList	OpenDescriptor, ReadDescriptor, ReadInfoBlock
Play	SetPlugAssociation, Play
Record	SetPlugAssociation, Record
VariableForward	Play
VariableReverse	Play
Stop	Stop
RecPause	Record
Skip	OpenDescriptor, ReadInfoBlock, SetPlugAssociation, Search, Play
GetState	OpenDescriptor, ReadInfoBlock
GetPosition	OpenDescriptor, ReadInfoBlock
Erase	Erase
PutItemList	OpenDescriptor, ReadInfoBlock, WriteInfoBlock

Tabelle 4.4: AvDisc-API-Umsetzung: HAVi-AV/C (Auszug)

Verwendung des IEEE 1394-Busses möglich. Da AV/C und HAVi auf dem Plug-Modell des Standards IEC 61883-1 [IEC98] basieren, ist die Verwaltung von isochronen Verbindungen zwischen AV/C- und HAVi-Geräten für den HAVi Stream-Manager transparent. Der HAVi Stream-Manager bietet benutzerfreundliche Funktionen zum Auf- und Abbau von Verbindungen zwischen zwei FCMs zur isochronen Datenübertragung und sorgt für die Wiederherstellung von Punkt-zu-Punkt-Verbindungen nach einem Bus-Reset. Neben der Kompatibilitätsprüfung (Richtung, Format, Transport, ...) der Anschlüsse aller beteiligten DCMs und FCMs und der Konfiguration der geräteinternen Verbindungen zwischen den DCM- und FCM-Anschlüssen, sind die Reservierung der Übertragungsressourcen (IEEE 1394-Bandbreite, IEEE 1394-Kanal) beim IRM (Isochronous Resource Manager des IEEE 1394-Busses) sowie die Konfiguration der IEC 61883-Plug-Control-Register bei der AV-Datenquelle und -Senke notwendig. Während die ersten beiden entsprechend der API-Umsetzung zwischen HAVi und AV/C über die DCM bzw. die FCM am Gateway erfolgen, wird auf die Plug-Control-Register direkt am AV/C-Gerät zugegriffen. Die GUID des AV/C-Gerätes erhält der Stream-Manager aus der HUID (`HUID.targetId.guid`)¹⁸ der das AV/C-Gerät repräsentierenden DCM [Zie00].

Der Test der Kopplung erfolgte anhand eines Demo-Aufbaus, bestehend aus (1) zwei HAVi-TVs (emuliert durch Linux-PCs), (2) dem HAVi-AV/C Service-Gateway, (3) einem AV/C DVB-S-Tuner [Pro02] und (4) zwei AV/C AV-Hard-Disk-Drives [Mat02a, Mat02b].

¹⁸Die HUID enthält die GUID des AV/C-Gerätes und nicht die des Gateways.

Abschließend ist als weiteres Beispiel für eine Service-Kopplung von unterschiedlichen Netzwerken des gleichen Bereichs die in [ACGI03] beschriebene Kopplung zwischen Jini und UPnP zu erwähnen. Hierbei handelt es sich um eine zweiseitige, dynamische Kopplung, weil sowohl Jini-Dienste als auch UPnP-Geräte automatisch der jeweils anderen Seite in Form von „virtuellen“ UPnP-Geräten bzw. Jini-Diensten bereitgestellt werden. Wie auch bei der HAVi-AV/C Kopplung muss allerdings für jeden Gerätetyp ein entsprechendes Modul für die API-Umsetzung extra entwickelt werden.

4.3 Unterhaltungselektronik – Small Office/Home Office

Ein typisches Anwendungsbeispiel für eine Kopplung zwischen zwei Netzwerken dieser beiden Bereiche ist in Abbildung 4.11 dargestellt. Nachdem in den anderen Abschnitten bereits System- und Service-Gateways verwendet wurden, sollen in diesem Abschnitt UI-Gateways zum Einsatz kommen. So werden zwei Konzepte für eine dynamische UPnP-HAVi UI-Kopplung beschrieben, welche die Nutzung von HAVi-Geräten (z. B. Sat-Receiver) über ein UPnP-Gerät mit HTML-Browser (z. B. PC) erlauben:

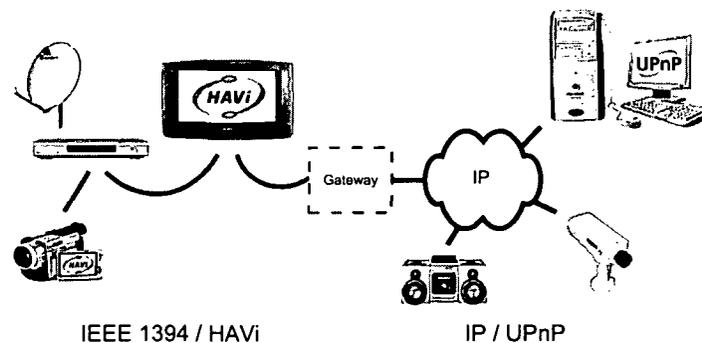


Abbildung 4.11: UPnP-HAVi Kopplung

- (1) Ansteuerung von HAVi-Geräten mit DDI-Target (HAVi-User-Interface-Level 1) über ein UPnP-Gerät mit HTML-Browser durch Umsetzung der DDI-Targets in entsprechende HTML-Seiten und Weiterleitung von Benutzereingaben am UPnP-HTML-Browser an das zugehörige DDI-Target (vgl. WAP-HAVi/DDI UI-Gateway [NMV02, NMV03]);
- (2) Ansteuerung von HAVi-Geräten bestimmter Typen über ein UPnP-Gerät mit HTML-Browser durch die dynamische Erzeugung und Bereitstellung von geeigneten HTML-Seiten auf UI-Ebene und Einsatz einer HAVi-Anwendung (Service-Schicht) zum Aufruf entsprechender Funktionen der DCM/FCM des zugehörigen HAVi-Gerätes zur Behandlung von am UPnP-HTML-Browser erfolgten Benutzereingaben (basierend auf [BGSS03]).

Zum Vergleich der in weiterer Folge beschriebenen UI-Gateways mit einem Service-Gateway zur Kopplung von UPnP und HAVi werden in Abbildung 4.12 die Kommunikationsbeziehungen zwischen einem UPnP-Gerät (als Nutzer) und einem HAVi-Gerät (als Bereitsteller einer gewünschten Funktion) über ein dazwischenliegendes Service-Gateway dargestellt.

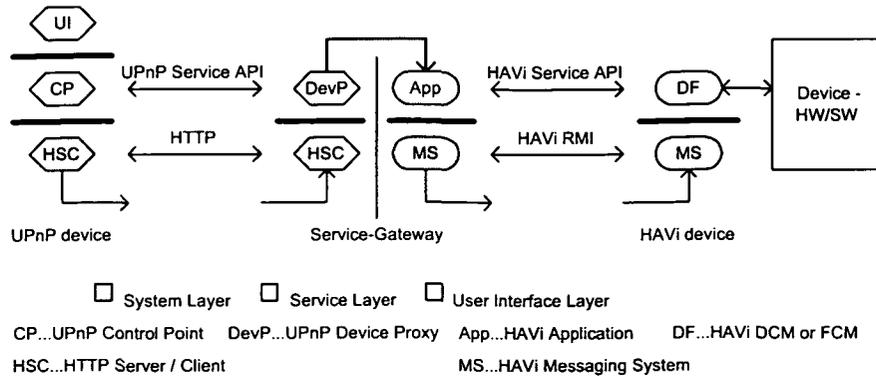


Abbildung 4.12: UPnP-HAVi Service-Gateway

4.3.1 UPnP-HAVi/DDI UI-Gateway

Das HAVi-User-Interface-Level 1 (Data Driven Interaction) definiert die Bereitstellung einer plattformunabhängigen Benutzerschnittstelle (DDI-Target), die von einem DDI-Controller geladen und dargestellt werden kann. Benutzereingaben werden vom Controller an das Target weitergeleitet und die dargestellte Benutzerschnittstelle bei Aufforderung des Targets aktualisiert. Wie auch im rechten Teil von Abbildung 4.13 dargestellt ist, erfolgt die Kommunikation zwischen Controller (DDI-C) und Target (DDI-T) über das DDI-Protokoll, wobei hierbei eine HAVi-Anwendung dem DDI-Controller und eine HAVi-DCM dem DDI-Target zum Senden und Empfangen von Nachrichten über das Messaging System (MS, HAVi RMI) dient.¹⁹ Zu beachten ist, dass bei DDI die Ansteuerung der Geräte nicht über die Service-API sondern über das DDI-Target erfolgt. Der Controller teilt dem Target lediglich mit, welche Eingabe erfolgt ist (z. B. DDI-Element-Nummer: 7, Benutzeraktion: Taste gedrückt), weiß jedoch nicht, welche Aktion am Gerät (z. B. Audio/Video-Display: Ton lauter, Bild heller, Bild schärfer, ...) damit verbunden ist. Die Interpretation obliegt alleine dem DDI-Target, das dann die entsprechende Aktion an der Geräte-HW/SW vornimmt (Pfeil von DDI-T zu Device-HW/SW).

Die UPnP Device Architecture definiert eine HTML-basierende Benutzerschnittstelle zur Ansteuerung und Wiedergabe des Zustands eines Gerätes. Dazu ist in der UPnP-Device-Description das optionale Element `presentationURL` definiert, das die URL zu

¹⁹Grundsätzlich könnte auch für eine Anwendung ein DDI-Target zur Verfügung gestellt werden. In diesem Fall dient dann die HAVi-Anwendung dem DDI-Target zum Senden und Empfangen von Nachrichten.

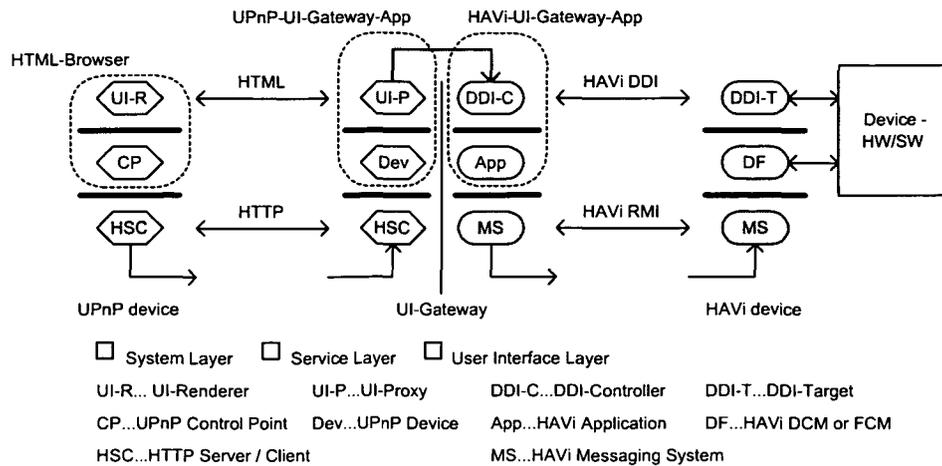


Abbildung 4.13: UPnP-HAVi/DDI UI-Gateway

der HTML-Seite des Gerätes (*Presentation-Page*) enthält. Wie der linke Teil der Abbildung 4.13 zeigt, kann ein Control-Point (CP) anhand dieser URL über ein normales HTTP-GET die *Presentation-Page* laden und durch einen UI-Renderer (UI-R) die HTML-Seite als Benutzerschnittstelle des Gerätes darstellen.

Damit UPnP-Geräte mit einem HTML-Browser über ihre HTML-basierte Benutzerschnittstelle auch HAVi-Geräte ansteuern können, sorgt das UI-Gateway für die Umsetzung der DDI-Targets in entsprechende HTML-Seiten und die Weiterleitung von Benutzereingaben an das zugehörige DDI-Target. Dazu werden vom DDI-Controller am UI-Gateway die DDI-Targets aller HAVi-Geräte geladen (abonniert) und für jedes geeignete HTML-Seiten erzeugt (siehe Tabelle 4.5) und dem UI-Proxy übergeben. Dieser sorgt für die Erstellung einer HTML-Seite, die für jedes DDI-Target einen Link mit Namen des HAVi-Gerätes und den Verweis auf die dem DDI-Root-Panel²⁰ entsprechende HTML-Seite enthält. Diese Seite dient als Einstiegsseite zur Ansteuerung von HAVi-Geräten und wird vom UPnP-Device des UI-Gateways als *Presentation-Page* im UPnP-Netz bekannt gemacht. Die HTML-Seiten werden über den HTTP-Server des UPnP-Stacks bereitgestellt. Über die *Presentation-Page* des UI-Gateways kann ein UPnP-Control-Point die in HTML umgewandelten DDI-Targets laden und damit die Benutzerschnittstelle der HAVi-Geräte anzeigen.

Der UI-Proxy stellt nicht nur die HTML-Seiten der DDI-Targets zur Verfügung, er behandelt auch die von UPnP-HTML-Browsern gesendeten Benutzereingaben, indem er für deren Weiterleitung an den DDI-Controller sorgt. Dieser sendet über das DDI-Protokoll die der Benutzereingabe entsprechende DDI-Aktion an das zugehörige DDI-

²⁰Jedes DDI-Target besteht aus zumindest einem organisatorischen Element, dem Root-Panel, über das alle weiteren DDI-Elemente entweder direkt oder über im Root-Panel enthaltene organisatorische Elemente (Panel, Group) erreicht werden können.

Target, welches die Benutzereingabe interpretiert und letztendlich für die Ausführung der dafür vorgesehenen Aktion am HAVi-Gerät sorgt. Eine damit verbundene Aktualisierung der Benutzerschnittstelle (z. B. DVD-Player beginnt das Abspielen einer DVD → Status ändert sich von „STOP“ auf „PLAY“) kann über die Antwort auf die vom UPnP-HTML-Browser gesendete Anfrage erreicht werden.

Der DDI-Controller ist mit dem DDI-Target über Benachrichtigungen synchronisiert und aktualisiert die zugehörigen, über den UI-Proxy für das UPnP-Netz bereitgestellten HTML-Seiten. Da HTML ein „PULL“-Dienst ist - kurz gefasst keine unaufgeforderten Nachrichten empfangen kann - kann es vorkommen, dass die im Browser eines UPnP-Gerätes dargestellte Seite nicht mehr aktuell ist. Erst durch ein erneutes Laden der Seite, veranlasst entweder durch explizite Aufforderung oder als Folge der Antwort auf eine gesendete Benutzereingabe, wird der zu diesem Zeitpunkt aktuelle Zustand des HAVi-Gerätes angezeigt. Das Verlassen eines DDI-Targets vom HAVi-Netzwerk wird vom DDI-Controller über die HAVi-Message-Supervision erkannt und dem UI-Proxy mitgeteilt. In diesem Fall löscht der UI-Proxy den zugehörigen Eintrag aus der Presentation-Page (Einstiegsseite) des UI-Gateways und die zugehörigen Daten (HTML-Seiten, Verwaltungsinformationen).

DDI-Element	HTML
Panel	HTML-Seite
Group	Fieldset
PanelLink	Link auf HTML-Seite
Button	Forms (Button)
Toggle	Skript
Animation	Skript
ShowRange	Image (je nach Wert)
SetRange	Skript
Entry	Forms (Input/Text und OK-Button)
Choice	Forms (Select und OK-Button)
Text	Paragraph
Status	Skript
Icon	Image

Tabelle 4.5: DDI-zu-HTML-Umsetzung (Übersicht)

Die Umsetzung der DDI-Targets in entsprechende HTML-Seiten [W3C99] kann prinzipiell entsprechend Tabelle 4.5 erfolgen. Während die meisten DDI-Elemente ohne größeren Aufwand umgesetzt werden können, sind die DDI-Elemente Animation, Toggle, SetRange und Status ohne Skriptunterstützung des Browsers schwer zu realisieren. Beispielsweise müsste SetRange bei der Realisierung durch Forms (Button mit Image) bei jeder Wertänderung ein Image, das den aktuellen Wert repräsentiert, vom Server laden. Ein weiterer zu berücksichtigender Aspekt sind die große Anzahl an verschiedenen At-

tributen der DDI-Elemente.

4.3.2 UPnP-HAVi/App UI-Gateway

Geräte werden im HAVi-Netzwerk durch DCMs und ihre funktionellen Komponenten durch FCMs repräsentiert. Für die Benutzerschnittstelle gibt es unterschiedliche Möglichkeiten, wie DDI, Havlets oder Anwendungen mit proprietärer GUI-Nutzung. Letztere wird hierin als *UI/HAVi-Anwendung* bezeichnet und bestehen aus zwei Teilen: einem UI-Teil, der spezielle GUI-Fähigkeiten²¹ eines Gerätes zur Ausführung einer vordefinierten Benutzerschnittstelle nutzt, und einem HAVi-Teil (HAVi-Anwendung), der zur Ansteuerung der HAVi-Geräte durch Aufruf entsprechender Funktionen der zugehörigen DCM/FCMs genutzt wird. Die Ansteuerung der Geräte erfolgt hierbei, im Gegensatz zum vorhergehenden Abschnitt (DDI), nicht auf der UI-Schicht sondern auf der Service-Schicht und deshalb können auch HAVi-Geräte ohne eigener Software für die UI-Schicht genutzt werden.

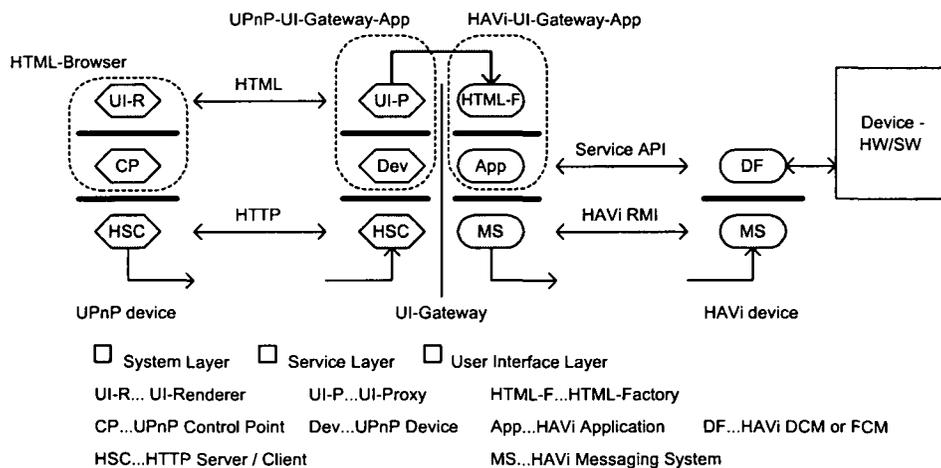


Abbildung 4.14: UPnP-HAVi/App UI-Gateway

Die Idee bei diesem Gateway ist, eine UI/HAVi-Anwendung zu verwenden, deren UI-Teil anstelle der Ausführung einer lokalen Benutzerschnittstelle für jedes HAVi-Gerät eine geeignete Benutzerschnittstellenbeschreibung für das angekoppelte Netzwerk erzeugt, im vorliegenden Fall demnach für jedes HAVi-Gerät entsprechende HTML-Seiten erstellt und diese den angebotenen UPnP-Geräten mit HTML-Browser zur Darstellung der Benutzerschnittstelle der HAVi-Geräte bereitstellt. Abbildung 4.14 zeigt das UI-Gateway und dessen Kommunikationswege. Die UI/HAVi-Anwendung wird hier als

²¹Je nach Betriebssystem und Programmiersprache sind unterschiedliche GUI-Pakete verfügbar: Win32, MFC, GTK, Qt, Swing, ...

HAVi-UI-Gateway-App und deren UI-Teil als HTML-Factory bezeichnet. Die HTML-Factory erzeugt

- für jede FCM, deren FCM-Typ unterstützt wird, eine HTML-Seite, welche die Nutzung der allgemeinen Funktionen des jeweiligen FCM-Typs erlaubt und
- für jede DCM eine HTML-Seite mit Geräteinformationen (Device-Icon, User-Preferred-Name, ...) und Links zu den Seiten der zugehörigen FCMs. Für nicht unterstützte FCM-Typen wird statt dem Link nur der Name des FCM-Typs in der HTML-Seite der DCM angeführt.

Ähnlich zu dem im vorhergehenden Abschnitt beschriebenen UI-Gateway werden die HTML-Seiten einem UI-Proxy (UI-P) übergeben. Dieser sorgt für die Erstellung einer HTML-Seite, die Links zu den HTML-Seiten aller verfügbaren DCMs enthält und als Einstiegsseite für die HTML-basierte Benutzerschnittstelle der HAVi-Geräte dient. Die Seite wird vom UPnP-Device des UI-Gateways (Dev) als Presentation-Page im UPnP-Netz bekannt gemacht und wie auch die anderen HTML-Seiten über den HTTP-Server des UPnP-Stack bereitgestellt. UI-Proxy und UPnP-Device bilden zusammen die Gateway-Anwendung der UPnP-Seite (UPnP-UI-Gateway-App). Über die Presentation-Page des UI-Gateways kann jeder UPnP-Control-Point die HTML-basierte Benutzerschnittstelle der HAVi-Geräte laden.

Die von HTML-Browsern gesendeten Benutzereingaben werden dem UI-Proxy zugestellt, der diese an die HTML-Factory weiterleitet. Dort werden über die HAVi-Anwendung (App) die entsprechenden Funktionen der zugehörigen DCM bzw. FCM aufgerufen und anhand der Antwort die HTML-Seiten des betroffenen HAVi-Gerätes aktualisiert und dem UI-Proxy als Antwort übergeben. Anhand dieser Daten kann der UI-Proxy eine geeignete Antwort an den betreffenden HTML-Browser zurücksenden.

Für die Synchronisation der bereitgestellten HTML-Seiten mit dem Zustand der HAVi-Geräte und die Erkennung der Entfernung von HAVi-Geräten gilt entsprechendes wie für das im vorhergehenden Abschnitt beschriebene UI-Gateway.

Kapitel 5

Bewertung und Ausblick

Die Zielsetzung dieser Arbeit war die Untersuchung der verschiedenen Möglichkeiten zur dynamischen Kopplung von Gerätenetzwerken im Heimbereich. Nach einem Überblick über das vernetzte Heim (Geräte, Bereiche, Netzwerke, Anwendungen, Anforderungen, Probleme, ...) wurde die Gerätevernetzung im Heimbereich anhand der Unterteilung des Heims in die drei Anwendungsbereiche „Heim-Automatisierung“, „Unterhaltungselektronik“ und „Small Office/Home Office“ beschrieben, der Stand der Technik dargestellt und neue Anforderungen wie die Kopplung von Gerätenetzwerken vorgestellt. Zum Vergleich der unterschiedlichen Vernetzungskonzepte und der Beschreibung von Kopplungen zwischen den einzelnen Netzwerken wurde ein allgemeines Modell zur Kopplung von Gerätenetzwerken im Heimbereich definiert. Wesentlicher Bestandteil dieses Modells ist eine dreischichtige Gerätearchitektur, welche die für die Vernetzung notwendigen Dienste definiert und diese in drei Schichten gruppiert. Eine Kopplung von Gerätenetzwerken erfolgt durch Umsetzung auf einer der Schichten der Gerätearchitektur, nach welcher auch das entsprechende Gateway benannt wurde. Des Weiteren wurden unterschiedliche Arten von Kopplungen definiert und der Einsatz von mehreren Kopplungen in Form von Kopplungssystemen behandelt. Am Beispiel der Unterhaltungselektronik wurden schließlich sechs verschiedene Realisierungen bzw. Konzepte für dynamische Kopplungen im Heimbereich entwickelt. Im Einzelnen sind das ein HAVi-EIB und HAVi-LON Service-Gateway, die Geräten der Unterhaltungselektronik die Nutzung von Geräten der Heim-Automatisierung ermöglichen, ein HAVi=HAVi System-Gateway zur Verbindung von entfernten HAVi-Netzwerken über ein TCP/IP-basiertes Transitnetzwerk, ein HAVi-AV/C System-Gateway zur Nutzung von AV/C-Geräten über das HAVi-Netzwerk und zwei UI-Gateways (UPnP-HAVi/DDI und UPnP-HAVi/App), welche die Nutzung von HAVi-Geräten über ein UPnP-Gerät mit HTML-Browser erlauben.

In Abschnitt 5.1 sind die Kernaussagen der Arbeit zusammengefasst und in Abschnitt 5.2 wird ein Ausblick über weitere Untersuchungen, Implementierungen und Maßnahmen bezüglich der Kopplung von Gerätenetzwerken im Heimbereich gegeben.

5.1 Bewertung

Anhand von sechs Thesen (Heterogenes Umfeld, Kopplungsgewinn, allgemeine Gerätearchitektur, Kopplungsmodell zur Analyse, Kopplungsmodell zum Design, dynamische Kopplung) werden nachfolgend die Kernaussagen der Arbeit ausgeführt. Eine Bewer-

tung der sechs entwickelten und im vorhergehenden Kapitel beschriebenen dynamischen Kopplungen erfolgt in Tabelle 5.2.

These 1 (Heterogenes Umfeld) *Im Heimbereich werden mehrere, unterschiedliche Gerätenetzwerke eingesetzt. Der Heimbereich ist ein heterogenes Umfeld.*

In Kapitel 2 wurde dargelegt, dass der Heimbereich in zumindest drei unterschiedliche Anwendungsbereiche unterteilt werden kann: Heim-Automatisierung, Unterhaltungselektronik und in den Bereich Small Office/Home Office. Da jeder dieser Bereiche unterschiedliche Anforderungen an ein Gerätenetzwerk hinsichtlich Bandbreite, Ausdehnung, Geräteanzahl, Datensicherung, Betriebssicherheit, Konfigurationsaufwand, Gerätenutzung, Übertragungsmedium, Teilnehmersdynamik und anderer Eigenschaften stellt, gibt es kein einzelnes, homogenes im realen Heimbereich - das heißt ausserhalb des reinen Laboraufbaus unter ökonomischen Rahmenbedingungen (Kostenfaktor!) - einsetzbares Gerätenetzwerk.

Einerseits bestehen zwischen einzelnen Anforderungen Zielkonflikte, wie beispielsweise zwischen einer hohen Bandbreite und einer großen Ausdehnung oder zwischen einer hohen Betriebssicherheit und einer großen Teilnehmersdynamik. Andererseits werden zur Erreichung möglichst niedriger Kosten hauptsächlich nur die primär notwendigen Anforderungen einer Gerätevernetzung realisiert. So wird für die Vernetzung von Geräten der Unterhaltungselektronik eine zeitlich garantierte Übertragung von großen Datenmengen gefordert, während bei der Heim-Automatisierung eine gesicherte Übertragung von kleinen Datenmengen über große Distanzen zwischen vielen im ganzen Heim verteilten Geräten notwendig ist. Die Verwendung von Plastic Optical Fibre als Übertragungsmedium mit einer möglichen Bandbreite von 200 Mbit/s und der Einsatz von Mechanismen zur automatischen Konfiguration der Teilnehmer sind zur Ansteuerung einer dauerhaft angebrachten Lampe ökonomisch wohl nicht vertretbar. Die Zielkonflikte zwischen einzelnen Anforderungen und vor allem die ökonomischen Rahmenbedingungen begründen damit den Einsatz bereichsspezifischer Gerätenetzwerke.

These 2 (Kopplungsgewinn) *Kopplungen zwischen den Gerätenetzwerken im Heimbereich bringen einen Mehrwert für den Nutzer.*

Wie in Abschnitt 2.2 und bei den im Kapitel 4 beschriebenen dynamischen Kopplungen ausgeführt wurde, ermöglicht die Kopplung von Gerätenetzwerken einerseits die einfachere Bedienbarkeit der Geräte durch die gemeinsame Nutzung von Benutzerschnittstellen (z. B. HAVi-EIB-Kopplung: Verwendung eines HAVi-TV zur Einstellung einer über EIB angebotenen Heizungsanlage) und andererseits die Überwindung von Beschränkungen des Netzwerks, wie z. B. dessen maximale Ausdehnung (z. B. HAVi-HAVi-Kopplung: Programmierung eines HAVi-VCR im Heim über ein entferntes via TCP/IP angebotenes HAVi-Netzwerk). Des Weiteren werden durch die Kopplung neue, netzübergreifende Anwendungen möglich. Beispiele dafür sind programmierbare Szenarien zur vordefinierten Einstellung verschiedener Geräte, wie „TV-Prime-Time“ (Licht im Wohnzimmer gedimmt auf 20%, TV-Audio über HiFi-Anlage mit Lautstärke auf Stufe 5,

TV-Videoeinstellungen Standard, ...) oder „Wake-Up“ (bei Erreichen der Weckzeit: Radio ein, Jalousie öffnen, persönliche Verkehrs- und Wetterinfos über TV bereitstellen, ...), und erweiterte Einstellmöglichkeiten durch die Verknüpfung von Informationen aus mehr als einem Netzwerk, wie die Programmierung der Heizung entsprechend der über das TV-Gerät abrufbaren Wetterdaten. Die Kopplung der Gerätenetze ist die konsequente Fortführung des Vernetzungsgedankens, demzufolge der Nutzen eines Geräte-netzwerks größer ist als die Summe des Nutzens der einzelnen Gerätefunktionen. Daraus ergibt sich der Kopplungsgewinn als Differenz zwischen dem Gesamtnutzen gekoppelter Gerätenetze und der Summe des Nutzens der einzelnen Gerätenetze.

These 3 (Allgemeine Gerätearchitektur) *Die unterschiedlichen Gerätenetze im Heimbereich können anhand einer allgemeinen 3-schichtigen Gerätearchitektur grundsätzlich beschrieben und verglichen werden.*

Die Analyse der verschiedenen in Abschnitt 2.1 beschriebenen Konzepte zur Vernetzung von Geräten zeigt, dass diese im Wesentlichen Mechanismen zur Adressierung, Übertragung, Bekanntmachung, Beschreibung, Ansteuerung, Benachrichtigung und Darstellung der Geräte beziehungsweise ihrer Kommunikationsobjekte erfordern. Basierend auf diesen allgemeinen Mechanismen zur Vernetzung wurde in Abschnitt 3.1 eine 3-schichtige Gerätearchitektur definiert. Die Gerätearchitektur besteht aus den 6 grundlegenden Diensten *Communication (Addressing, Transmission, Interaction), Discovery, Description, Control, Notification* und *User Interface*, wobei die Dienste entsprechend ihres Aufgabenbereichs auf die drei Schichten *System Layer, Service Layer* und *User Interface Layer* verteilt sind. Unterhalb der System-Schicht ist der *Transport*, der das Übertragungssystem und Übertragungsmedium repräsentiert. Oberhalb der dritten Schicht ist der Benutzer, der über den *User Interface Layer* die Dienste des Gerätenetzes nutzen kann.

Aufgabe	Realisierung
Transport	...
Addressing	...
Transmission und Interaction	...
Discovery	...
Description	...
Control	...
Notification	...
User Interface	...

Tabelle 5.1: Schablone zur Beschreibung von Gerätenetzen

Den Nachweis der allgemeinen Anwendbarkeit der Gerätearchitektur zur Beschreibung der Kernmechanismen eines Netzwerks wurde in Abschnitt 3.2 durch die Beschreibung

der 6 verschiedenen Gerätenetze LON (Tabelle 3.1), EIB (Tabelle 3.2), HAVi (Tabelle 3.4), AV/C (Tabelle 3.3), UPnP (Tabelle 3.5) und Jini (Tabelle 3.6) auf jeweils maximal einer Seite entsprechend der in Tabelle 5.1 gezeigten Schablone erbracht. Die Beschränkung der Beschreibung auf eine Seite dient zur Fokussierung auf die wesentlichen Fähigkeiten und Eigenschaften eines Netzwerks. Durch die strikte und anwendungsbereichsunabhängige Struktur der Beschreibung wird eine grundsätzliche Vergleichbarkeit der unterschiedlichen Netzwerke ermöglicht, deren Anwendbarkeit durch die Nutzung der oben genannten Tabellen für den Vergleich der Netzwerke zur Erstellung der in Kapitel 4 beschriebenen Kopplungen belegt wurde.

These 4 (Kopplungsmodell zur Analyse) *Die verschiedenen Verbindungen der Gerätenetze im Heimbereich können anhand eines allgemeinen Modells zur Kopplung von Gerätenetzen im Heimbereich beschrieben werden.*

Zur Verbindung von Gerätenetzen im Heimbereich sind eine Reihe von Kopplungen publiziert worden ([WUB01, WG00, KL01, RHB02, Cho02, NMV02], ...). Die meisten der Kopplungen unterscheiden sich jedoch nicht nur hinsichtlich der zu verbindenden Netzwerke, sondern auch hinsichtlich der Zielsetzung der Verbindung (Überwindung von Beschränkungen eines Gerätenetzes, Ansteuerung von Geräten über eine andere API als die ursprüngliche API, Benutzung der Geräte über eine Benutzerschnittstelle eines anderen Netzwerks, ...). Die Suche nach einer geeigneten Kopplung und der Vergleich der Kopplungen ist deshalb schwierig. Vielfach sind wesentliche Informationen klausuriert in der Beschreibung enthalten oder fehlen überhaupt. Abhilfe schafft das in Kapitel 3 definierte allgemeine Modell zur Kopplung von Gerätenetzen im Heimbereich. Es ermöglicht, die Aufgabe einer Kopplung und deren grundsätzliche Funktionsweise einfach zu beschreiben.

Wesentlicher Bestandteil des Kopplungsmodells ist die in Abschnitt 3.1 definierte Gerätearchitektur, welche die für die Vernetzung notwendigen Dienste definiert und diese in drei Schichten gruppiert. Eine Kopplung von Gerätenetzen erfolgt durch Umsetzung auf einer der Schichten der Gerätearchitektur, nach welcher auch das entsprechende Gateway benannt wird: *System-Gateway*, *Service-Gateway*, *UI-Gateway* (siehe Kopplungsmethoden in Abschnitt 3.3). In Ergänzung zu diesen Gateways wird ein Gerät, das gleichzeitig Teilnehmer an zwei oder mehreren Netzwerken ist und lokalen Anwendungen die Nutzung von Diensten und Geräten mehrerer direkt angeschlossener Netzwerke ermöglicht, als *Multihomed Non Transforming Device* bezeichnet. Die unterschiedlichen Arten von Kopplungen (einseitig/zweiseitig, direkt/indirekt, statisch/dynamisch, ...) wurden in Abschnitt 3.4 beschrieben. Verfügt ein Netzwerk über mehr als eine Kopplung, so werden die einzelnen Kopplungen aller Netze gemeinsam als *Kopplungssystem* bezeichnet (siehe Abschnitt 3.5).

Anhand des Kopplungsmodells wurden insgesamt 17 Kopplungen analysiert und das Ergebnis in Tabelle 5.2 zusammengefasst. Die Beschreibung einer Kopplung anhand des Modells erfordert Antworten über die wesentlichen Punkte der Kopplung. Dadurch wird

Methode	Art	System	Tra.	Dyn.	Referenz	Abs.
NT	HAVi,EIB	-	-	-	[Abschnitt 3.3.1]	3.3.1
SY	HAVi=HAVi	T	j	j	[Abschnitt 4.2.1]	4.2.1
SY	HAVi-HAVi	T	j	j	[WUB01]	3.3.2
SE	UPnP-EIB	AB	j	j	[WG00]	4.1.5
SE	Jini-LON	AB	j	n	[KL01]	4.1.5
SE	Jini-EIB	AB	j	j	[RHB02]	4.1.5
SE	Jini-EIB	AB	j	n	[KK99]	4.1.5
SE	HAVi-EIB	AB	j	j	[Abschnitt 4.1]	4.1
SE	HAVi-LON	AB	j	j	[Abschnitt 4.1]	4.1
SE	HAVi-AV/C	AB	j	j	[Abschnitt 4.2.2]	4.2.2
SE	Jini-HAVi	AB	n	n	[Ski02, Rah01]	3.3.3
SE	Jini=UPnP	AB	j	j	[ACGI03]	4.2.2
SE	HAVi,UPnP	C	j	j	[Cho02]	3.5.2
SE	HAVi,UPnP,Jini	C	j	n	[NSA02]	3.5.2
UI	WAP-HAVi/DDI	AB	j	j	[NMV02, NMV03]	3.3.4
UI	UPnP-HAVi/DDI	AB	j	j	[Abschnitt 4.3.1]	4.3.1
UI	UPnP-HAVi/App	AB	j	j	[Abschnitt 4.3.2]	4.3.2

Tabelle 5.2: Analyse von Kopplungen

ein einfacher Vergleich mit anderen Kopplungen oder das Auffinden geeigneter Realisierungen möglich.

In der ersten Spalte ist die Kopplungsmethode angeführt. Hierbei wurde ein Multihomed Non Transforming Device mit NT, ein System-Gateway mit SY, ein Service-Gateway mit SE und ein UI-Gateway mit UI abgekürzt. Für die in der zweiten Spalte bezeichnete Kopplungsart wurde die in Abschnitt 3.4 festgelegte Konvention verwendet (siehe auch Abbildung 3.10). Hierbei bezeichnet „N1-N2“ eine einseitige, direkte Kopplung zwischen dem Netzwerk N1 und N2 und „N1=N2“ eine zweiseitige, direkte Kopplung zwischen N1 und N2. Im Falle der Kopplungsart NT oder einer C-Kopplung werden die verbundenen Netzwerke durch einen Beistrich getrennt angeschrieben. In der Spalte System wird mit T eine T-Kopplung, mit AB eine AB-Kopplung und mit C eine C-Kopplung abgekürzt. Wenn die Kopplung transparent ist, steht in der Spalte Tra ein „j“, anderenfalls ein „n“. Ist die Kopplung darüber hinaus auch dynamisch, so steht auch in der Spalte Dyn ein „j“, anderenfalls ein „n“. Die Spalte Referenz enthält einen Verweis auf die Beschreibung der Kopplung. Die letzte Spalte bezeichnet den Abschnitt, in welchem die Analyse der Kopplung beschrieben ist.

Die siebzehn untersuchten Kopplungen stellen einen guten Querschnitt dar. So sind alle sechs in Abschnitt 2 beschriebenen Gerätenetze aus den drei Bereichen des Heims, alle vier Kopplungsmethoden, alle drei Kopplungssysteme, einseitige, zweiseitige, direkte, indirekte, transparente sowie dynamische Kopplungen vertreten.

These 5 (Kopplungsmodell zum Design) *Ein allgemeines Modell zur Kopplung von Gerätenetzwerken im Heimbereich eignet sich als Grundlage für das Design von Kopplungen.*

Voraussetzung für ein Design von Kopplungen ist eine Analyse der zu verbindenden Gerätenetzwerke. Dies kann anhand der dreischichtigen in Abschnitt 3.1 definierten Gerätearchitektur entsprechend der in Tabelle 5.1 gezeigten Schablone erfolgen. Da das Kopplungsmodell unterschiedliche Methoden und Arten von Kopplungen beschreibt, dient es als eine Art „Werkzeugkasten“ für neue Kopplungen. Ausgehend von der Zielsetzung der zu erstellenden Kopplung (Überwindung von Beschränkungen eines Gerätenetzwerks, Ansteuerung von Geräten über eine andere API als die ursprüngliche API, Benutzung der Geräte über eine Benutzerschnittstelle eines anderen Netzwerks, ...) kann so entsprechend den Abschnitten 3.3 und 3.4 eine geeignete Kopplungsmethode und -art gewählt werden.

System-Gateways werden vor allem zur Überwindung von Beschränkungen eines Gerätenetzwerks hinsichtlich der Netzwerkausdehnung, der Geräteanzahl oder des Transportmediums oder auch zur Erweiterung eines Gerätenetzwerks auf neue Anwendungsgebiete eingesetzt. Im Gegensatz dazu ermöglicht ein Service-Gateway die Ansteuerung von Geräten über eine andere API als die ursprüngliche API, d. h. die Dienste und Geräte des angebundenen Netzwerks können wie Dienste und Geräte des eigenen Netzwerks genutzt werden. Eine Kopplung auf UI-Schicht ermöglicht wiederum die Benutzung von Geräten des angekoppelten Netzwerks über eine zweiteilige Benutzerschnittstelle des lokalen Netzwerks. In Ergänzung zu den schichtumsetzenden Gateways bietet das Multihomed Non Transforming Device, das gleichzeitig Teilnehmer zweier oder mehrerer Gerätenetzwerke ist, lokalen Anwendungen die Benutzung der Dienste und Geräte aller angeschlossenen Netzwerke. Neben der Kopplungsmethode ist die Auswahl der Kopplungsart ein weiteres Designkriterium: Soll die Kopplung transparent, einseitig, zweiseitig, direkt, indirekt oder dynamisch sein?

Entsprechend der gewählten Kopplungsmethode und -art sind die wesentlichen Teilaufgaben zur Erstellung der Kopplung vorgegeben (siehe Abschnitte 3.3 bis 3.5). Beispielsweise gehört zu den Teilaufgaben des System-Gateways die Abbildung der oft sehr unterschiedlichen Geräte- bzw. Funktionsabstraktionen, die Abbildung der unterschiedlichen Anwendungsprogramm-Schnittstellen (API-Mapping), die Kommunikation zwischen den unterschiedlichen Service-Schichten und soweit erforderlich auch die Ergänzung fehlender Mechanismen einer der den Service-Schichten zugrundeliegenden System-Schichten. Bei Vorliegen bestimmter globaler Nutzungsprofile im Heimbereich kann der gezielte Einsatz von Kopplungssystemen zur Reduktion der Anzahl an Kopplungen angedacht werden (siehe Abschnitt 3.5). Einen Nachteil können allerdings die dabei entstehenden indirekten Kopplungen darstellen.

Bei den in Kapitel 4 beschriebenen entwickelten Kopplungen diente das Kopplungsmodell aus Kapitel 3 als Grundlage für das Design. So konnten insgesamt sechs verschiede-

dene dynamische Kopplungen, drei Konzepte und drei Realisierungen, erstellt werden. Diese repräsentieren einen guten Querschnitt von Verbindungen zwischen Gerätenetzwerken, weil am Beispiel der Unterhaltungselektronik Kopplungen zu den drei Bereichen des Heims (Heim-Automatisierung, Unterhaltungselektronik, Small Office/Home Office) durchgeführt worden sind und hierbei alle drei Gateway-Typen sowie eine T-Kopplung und mehrere AB-Kopplungen zur Anwendung kamen.

These 6 (Dynamische Kopplung) *Geeignete Kopplungen müssen dynamisch sein.*

Gemäß Abschnitt 3.4.4 wird eine Kopplung als dynamisch bezeichnet, wenn sie (1) transparent ist, (2) nicht auf eine spezielle Umgebung bzw. Gerätekonfiguration beschränkt ist und (3) keinen oder einen nur sehr geringen Konfigurationsaufwand im Verhältnis zum Installationsaufwand der gekoppelten Netzwerke erfordert.

Die ersten beiden Forderungen beziehen sich auf die allgemeine Einsetzbarkeit der Kopplung. Die Forderung der Transparenz (siehe Abschnitt 3.4.1) ist damit begründet, dass bei nicht transparent gekoppelten Netzwerken nur über „eingeweihte“ Geräte die Dienste und Funktionen der Geräte des angekoppelten Netzwerks genutzt werden können. Für eine Nutzung der angekoppelten Geräte müssten die bestehenden Geräte erweitert werden, und im Falle von indirekt gekoppelten Netzwerken ist außerdem eine genaue Kenntnis über die an der indirekten Kopplung beteiligten Geräte und Netzwerke erforderlich. Zusätzliche Anforderungen an die Geräte stehen jedoch einer allgemeinen Einsetzbarkeit der Kopplung entgegen. Während die erste Forderung die Nutzung angekoppelter (importierter) Geräte behandelt, bezieht sich die zweite Forderung auf die zu exportierenden Geräte. Die dritte Forderung bezieht sich auf den notwendigen Aufwand zur Einrichtung der Kopplung. Wird auch nur eines dieser Kriterien nicht erfüllt, so kann die Kopplung entsprechend dem in dieser Arbeit entwickelten Kopplungsmodell nicht als dynamisch bezeichnet werden.

Für den Einsatz von Kopplungen im Heimbereich ist entscheidend, dass diese allgemein einsetzbar sind und ohne wesentlichen Konfigurationsaufwand auskommen. Nur durch die allgemeine Einsetzbarkeit der Kopplung können die unterschiedlichen im Heimbereich eingesetzten Anwendungen und Geräte unterstützt werden. Seitens der Benutzer wird ein möglichst geringer Konfigurationsaufwand der Kopplung zur Inbetriebnahme und bei Änderungen in den beteiligten Netzwerken (Hinzufügen und Entfernen von Geräten) gefordert [Ros02a]. Eine geeignete Kopplung von Gerätenetzwerken im Heimbereich muss deshalb dynamisch sein.

5.2 Ausblick

Wie die vorliegende Arbeit gezeigt hat, gewinnt die Kopplung von Gerätenetzwerken mit zunehmender Vernetzung in den unterschiedlichen Anwendungsbereichen des Heims an Bedeutung. Hierbei wären vor allem die folgenden Untersuchungen, Erweiterungen, Implementierungen und Maßnahmen von Bedeutung:

- Entwicklung einer dynamischen HAVi=UPnP Service-Kopplung: Die Relevanz einer solchen Kopplung begründet sich durch die starke Position von Microsoft am PC-Markt und deren Bekenntnis zur Weiterentwicklung von UPnP und des Einsatzes von UPnP in PCs oder PC-nahen Produkten [Nix03, Geh03] sowie die durch Media-Server und normale Flachbildschirme mit eingebautem TV-Tuner voranschreitende Konvergenz der PC- und Unterhaltungselektronik-Welt.
- Erweiterung des Kopplungsmodells hinsichtlich AV-Streaming, Transaktionen und Ressourcenverwaltung: In der derzeitigen Fassung des Modells werden diese Aspekte noch nicht eigenständig berücksichtigt und die dadurch auftretenden Verluste als Kopplungsverluste behandelt.
- Analyse von weiteren Kopplungen: Die weitere Analyse von publizierten Kopplungen anhand des Kopplungsmodells ermöglicht die Erweiterung oder Festigung des Modells.
- Weitere Untersuchungen bezüglich der Kopplungssysteme: Von Interesse wäre die Untersuchung des Gewinns einer Reduktion der Kopplungsanzahl durch den gezielten Einsatz von Kopplungssystemen im Vergleich zur unkoordinierten Verwendung von direkten Kopplungen. Durch die Ermittlung von globalen Nutzungsprofilen könnten gewünschte Kopplungen und deren Relevanz (Häufigkeit der gewünschten Kopplung pro Anwendungsszenario) erfasst werden. Anhand dieser Daten könnte eine Kopplungsmatrix mit einer Gewichtung jeder Kopplung erstellt werden. Daraus ließen sich dann Implementierungsempfehlungen ableiten: Kopplungen mit hohen Gewichtungen, d. h. häufig benötigte Kopplungen, sollen bevorzugt als direkte Kopplungen und Kopplungen mit niedrigen Gewichtungen in Form von indirekten Kopplungen (über zwei oder mehrere Kopplungen mit hohen Gewichtungen) realisiert werden.
- Implementierung eines HAVi=HAVi System-Gateways entsprechend dem in Abschnitt 4.2.1 beschriebenen Konzept: Durch dieses System-Gateway könnten entfernte HAVi-Netzwerke über ein TCP/IP-basiertes Transitnetzwerk zusammengeschlossen werden und so jedem HAVi-Gerät auch die Nutzung der HAVi-Geräte aller angebundenen HAVi-Netzwerke ermöglichen.
- Implementierung der beiden UPnP-HAVi UI-Gateways entsprechend den in Abschnitt 4.3 beschriebenen Konzepten: Durch die beiden UI-Gateways (UPnP-HAVi/DDI und UPnP-HAVi/App) könnten HAVi-Geräte über ein UPnP-Gerät mit HTML-Browser angesteuert werden.

Abbildungsverzeichnis

2.1	IEEE 1394-Adressierung	17
2.2	AV/C-Kommunikation	18
2.3	AV/C-Request-Packet	19
2.4	HAVi-Software-Architektur	21
2.5	Software-Element-Kommunikation	22
2.6	Allgemeine Struktur einer HAVi-Nachricht	22
2.7	Beispiel für einen synchronen Nachrichtentransfer	24
2.8	FCM-Reservierung	28
2.9	UPnP-Architektur	32
2.10	UPnP-Geräteaufbau	33
2.11	UPnP Discovery: Alive-Nachricht	34
2.12	UPnP Discovery: Suchanfrage und Antwort	35
2.13	UPnP Control: Funktionsaufruf	37
2.14	UPnP Control: Funktionsantwort	38
2.15	UPnP Eventing: Subscribe-Nachricht	39
2.16	UPnP Eventing: Benachrichtigung	40
2.17	Dienst und Dienst-Proxy	41
2.18	Discovery-Protokolle	42
2.19	Join und Lookup	44
2.20	Transaktionen	46
3.1	Dreischichtige Gerätearchitektur	54
3.2	Control und Notification	59
3.3	Multihomed Non Transforming Device	69
3.4	Multihomed Non Transforming Device - Beispiel	69
3.5	System Gateway	70
3.6	Entfernte Ausführung von HAVi-Anwendungen auf IP-Geräten	71
3.7	Service Gateway	72
3.8	User Interface Gateway	75
3.9	Benutzerschnittstellen - Merkmale	76
3.10	Kopplungsarten - Symbole	78
3.11	Vergleich: Gerätearchitektur vs. OSI-Referenzmodell	87
4.1	HAVi-Stack	90
4.2	CE-HA Service-Gateway	91

4.3	Unterteilung des Heim-Automatisierungs-Netzwerks	92
4.4	HAVi-LON Service-Gateway	96
4.5	HAVi-EIB Service-Gateway	97
4.6	HAVi-HAVi Kopplung via TCP/IP-Tunnel	100
4.7	HAVi-HAVi Kopplung über System-Gateways	101
4.8	HAVi-HAVi Kopplung: Komponenten	102
4.9	HAVi-HAVi Kopplung: Kommunikationsablauf	103
4.10	HAVi-AV/C Service-Gateway	106
4.11	UPnP-HAVi Kopplung	109
4.12	UPnP-HAVi Service-Gateway	110
4.13	UPnP-HAVi/DDI UI-Gateway	111
4.14	UPnP-HAVi/App UI-Gateway	113

Tabellenverzeichnis

1.1	Typische Bitraten von Audio/Video-Daten [Zie03b]	3
2.1	Vordefinierte Attribute der Registry (Auszug)	25
3.1	Local Operating Network (LON)	63
3.2	European Installation Bus (EIB)	64
3.3	Audio Video Control (AV/C)	65
3.4	Home Audio/Video interoperability (HAVi)	66
3.5	Universal Plug and Play (UPnP)	67
3.6	Jini	68
3.7	Beispiel für Rückimport und Mehrfachexport	79
4.1	Abbildung des HA-Netzwerks auf HAVi-DCM (D) und -FCMs (F)	92
4.2	HAVi-Dienste der HA-Erweiterung	94
4.3	Gateway-Kommunikation: Nachrichtenformat	105
4.4	AvDisc-API-Umsetzung: HAVi-AV/C (Auszug)	108
4.5	DDI-zu-HTML-Umsetzung (Übersicht)	112
5.1	Schablone zur Beschreibung von Gerätenetzwerken	117
5.2	Analyse von Kopplungen	119

Literaturverzeichnis

- [13998a] 1394 TRADE ASSOCIATION: *AV/C Tuner Broadcast System Specification - Digital Video Broadcast (DVB), Version 1.0*. Austin, Texas, USA: 1394 Trade Association, 1998
- [13998b] 1394 TRADE ASSOCIATION: *AV/C Tuner Model and Command Set, Version 1.0*. Austin, Texas, USA: 1394 Trade Association, 1998
- [13999a] 1394 TRADE ASSOCIATION: *AV/C Monitor Subunit Model and Command Set, Version 1.0*. Austin, Texas, USA: 1394 Trade Association, 1999. – TA Document 1999028
- [13999b] 1394 TRADE ASSOCIATION: *Configuration ROM for AV/C Devices, Version 1.0*. Austin, Texas, USA: 1394 Trade Association, 1999. – TA Document 1999027
- [13999c] 1394 TRADE ASSOCIATION: *Enhancements to the AV/C Tuner Broadcast System Specification - Digital Video Broadcast (DVB), Version 1.0*. Austin, Texas, USA: 1394 Trade Association, 1999
- [13901a] 1394 TRADE ASSOCIATION: *AV/C Digital Interface Command Set General Specification, Version 4.1*. Austin, Texas, USA: 1394 Trade Association, 2001. – TA Document 2001012
- [13901b] 1394 TRADE ASSOCIATION: *AV/C Tape Recorder/Player Subunit Specification, Version 2.2*. Austin, Texas, USA: 1394 Trade Association, 2001. – TA Document 2001017
- [13902a] 1394 TRADE ASSOCIATION: *AV/C Descriptor Mechanism Specification, Version 1.1*. Austin, Texas, USA: 1394 Trade Association, 2002. – TA Document 2001021
- [13902b] 1394 TRADE ASSOCIATION: *AV/C Disc Subunit - Hard Disk Drive Device Type Specification, Version 1.1*. Austin, Texas, USA: 1394 Trade Association, 2002. – TA Document 2001023
- [13902c] 1394 TRADE ASSOCIATION: *AV/C Disc Subunit General Specification, Version 1.1*. Austin, Texas, USA: 1394 Trade Association, 2002. – TA Document 2001022

- [ACGI03] ALLARD, J. ; CHINTA, V. ; GUNDALA, S. ; III, G. G. R.: Jini meets UPnP: An Architecture for Jini/UPnP Interoperability. In: *Proceedings of the 2003 Symposium on Applications and the Internet (SAINT 03)*, 2003, S. 268– 275
- [AHLZ02] ALBRIGHT, Shivaun ; HASTINGS, Tom ; LEWIS, Harry ; ZEHLER, Peter: *Printer Device Template v1.01 for UPnP v1.0*. UPnP Forum, 2002
- [And99] ANDERSON, Don: *FireWire System Architecture: IEEE 1394a*. Second. Reading, MA, USA : Addison-Wesley, 1999. – 544 S. – ISBN 0–201–48535–4
- [Anh02] ANHÄUPL, Bernd: Übertragung von IEEE1394-Datenverkehr über IP-Verbindungen / HomeNet2Run consortium partners. 2002. – Forschungsbericht
- [As03] VAN AS, Harmen. *Vorlesung Datenkommunikation - Teil 1.7 (Schichtenmodelle und Protokolle)*. Vorlesungsskript - Institut für Kommunikationsnetze, TU Wien. Februar 2003
- [BGSS03] BAIER, R. ; GRAN, C. ; SCHELLER, A. ; STOLP, R.: Control of CE Devices through a HAVi/IP Gateway / FhG FOKUS. 2003. – Forschungsbericht
- [BHP02] BROY, Manfred ; HEGERING, Heinz-Gerd ; PICOT, Arnold: *Integrierte Gebäudesysteme - Technologien, Sicherheit und Märkte / Bundesamt für Sicherheit in der Informationstechnik*. SecuMedia-Verlag, 2002. – ISBN 3–922746–35–7
- [BLFM98] BERNERS-LEE, T. ; FIELDING, R. ; MASINTER, L.: Uniform Resource Identifiers (URI): Generic Syntax / Internet Engineering Task Force. 1998 (2396). – RFC
- [BPS+02] BOUFFIOUX, Alain ; PERROT, Sebastien ; SPALINK, Gerd ; MISCHLER, Denis ; PHILIPS, Norbert: Transparent Bridges for IEEE1394 Networks with Hiperlan2 Interconnection between IEEE1394 Portals. In: *Proceedings of International Symposium on Consumer Electronics (ISCE 2002)*, Technische Universität Illmenau, Deutschland, 2002, S. F 163–166
- [Bra96] BRADNER, S.: The Internet Standards Process – Revision 3 / Internet Engineering Task Force. 1996 (2026). – RFC
- [Bra02] BRANECI, Mohamed: Overview of IEEE 1394 serial bus bridging solutions / HomeNet2Run consortium partners. 2002. – Forschungsbericht
- [Chi02] CHIASSERINI, Carla F.: Special-Use IPv4 Addresses / Internet Engineering Task Force. 2002 (3330). – RFC
- [Cho02] CHO, Song Y.: Framework for the composition and interoperation of the home appliances based on heterogeneous middleware in residential networks. In: *IEEE Transactions on Consumer Electronics* 48 (2002), S. 484– 489

- [Con03] CONTRIBUTING MEMBERS OF THE UPnP FORUM: *Universal Plug and Play Device Architecture, Version 1.0.1*. UPnP Forum, Mai 2003. – WWW: <http://www.upnp.org/>
- [CW98] COHEN, Aaron ; WOODRING, Mike: *Win32 Multithreaded Programming*. first Edition. O Reilly, 1998. – ISBN 1-56592-296-4
- [DKS00] DIETRICH ; KASTNER ; SAUTER: *EIB Gebäudebussystem*. Hüthig Verlag Heidelberg, 2000. – ISBN 3-7785-2795-9
- [DLS97] DIETRICH, Dietmar ; LOY, Dietmar ; SCHWEINZER, Hans-Jörg: *LON-Technologie. Verteilte Systeme in der Anwendung*. Hüthig, 1997
- [dmn02a] DMN: *AL1394 - Abstraction Layer 1394, Version 1.2*. dmn Software-Entwicklung GmbH, Mai 2002
- [dmn02b] DMN: *AL1394Iec - Abstraction Layer 1394, Version 1.0*. dmn Software-Entwicklung GmbH, 2002
- [dmn02c] DMN: *AL1394Rom - Abstraction Layer 1394, Version 1.2*. dmn Software-Entwicklung GmbH, 2002
- [dmn02d] DMN: *HAVi SDK Professional, Programmers Guide Vers. 1.2*. dmn Software-Entwicklung GmbH, Oktober 2002
- [dmn02e] DMN: *HAVi SDK Professional, Users Guide Vers. 1.2*. dmn Software-Entwicklung GmbH, Oktober 2002
- [dmn03] DMN: *ALOS - Abstraction Layer for Operating System, Version 1.2*. dmn Software-Entwicklung GmbH, November 2003
- [DR02] DAIMLERCHRYSLER AG ; ROBERT BOSCH GMBH: *HAVi over MOST Specification*. 2002. – Draft 0.9.4
- [Dro97] DROMS, R. E.: *Dynamic Host Configuration Protocol / Internet Engineering Task Force*. 1997 (2131). – RFC
- [Ech00] ECHELON: *LNS for Windows Programmers Guide, Vers. 3.0*. Echelon Corporation, 2000. – WWW: <http://www.echelon.com>
- [Ech01] ECHELON: *PCLTA-20 PCI Interface Users Guide*. Echelon Corp., 2001. – WWW: <http://www.echelon.com>
- [Ech02a] ECHELON: *LonMark SCPT Master List*. Echelon Corp., Mai 2002. – WWW: <http://www.echelon.com>
- [Ech02b] ECHELON: *LonMark SNVT Master List*. Echelon Corp., Mai 2002. – WWW: <http://www.echelon.com>

- [Ech03a] ECHELON: *Neuron C Programmer's Guide*. Palo Alto, CA, USA: Echelon Corp., 2003. – Document No. 29300
- [Ech03b] ECHELON: *Neuron C Reference Guide*. Palo Alto, CA, USA: Echelon Corp., 2003. – Document No. 29350
- [Edw00] EDWARDS, W. K.: *Core Jini*. Prentice Hall, 2000. – ISBN 3-8272-9592-0
- [EIA98] EIA: Control Network Protocol Specification / Electronic Industries Alliance. 1998 (EIA-709.1). – Standard
- [EIA02] EIA: Control Network Protocol Specification (Revision) / Electronic Industries Alliance. 2002 (EIA-709.1-B-2002). – Standard
- [EIB99] EIBA: Handbook Series Release 3.0 / EIB Association. 1999. – Specification
- [Els91] ELSING, Jürgen: *Das OSI-Schichtenmodell - Grundlagen und Anwendungen der X.200*. 1. IWT Verlag GmbH, 1991. – ISBN 3-88322-317-4
- [ETS00] ETSI: Broadband Radio Access Networks (BRAN) - HIPERLAN Type 2 - System Overview / European Telecommunications Standards Institute. F-06921 Sophia Antipolis Cedex - FRANCE, Februar 2000 (ETSI TR 101 683). – Standard
- [EUR99] EURESCOM PARTICIPANTS IN PROJECT P915-PF: HINE - Heterogeneous Inhouse Networking Environment: Deliverable 1 - Analysis of Inhouse System Technologies. 1999. – Forschungsbericht
- [FGM⁺99] FIELDING, R. ; GETTYS, J. ; MOGUL, J. C. ; FRYSTYK, H. ; MASINTER, L. ; LEACH, P. J. ; BERNERS-LEE, T.: Hypertext Transfer Protocol – HTTP/1.1 / Internet Engineering Task Force. 1999 (2616). – RFC
- [Foc03] FOCUS. *Der Markt der Computer-Hard- und Software: Daten - Fakten - Trends*. WWW: <http://www.medialine.de>. 2003
- [Fre91] Free Software Foundation, Inc.: *GNU General Public License*. Juni 1991. – WWW: <http://www.gnu.org/copyleft/gpl.html>
- [Fro02] FROUIN, Laurent. *HAVi Network Adapter - IEEE 1394.1 / HAVi-Bridge - Canon Research Centre France S.A.S. Prototype Presentation*. WWW: <http://www.homenet2run.org>. 2002
- [Geh03] GEHLSSEN, John: *UPnP Technology in Windows*. UPnP Forum Summit Presentation, 2003. – WWW: <http://www.upnp.org/>
- [GJSB00] GOSLING, James ; JOY, Bill ; STEELE, Guy ; BRACHA, Gilad: *The Java Language Specification*. Second Edition. ADDISON-WESLEY, 2000 (The Java Series). – ISBN 0-201-31008-2

- [Hau03] HAUSBAU AUGSBURG GMBH. *Das Intelligente Haus - Der zukunftsweisende Trend in der Gebäudesystemtechnik*. WWW: <http://www.augusta-ziegelbau.de/bus/bus01.html>. 2003
- [HAV01] HAVI INC: *Specification of the Home Audio/Video Interoperability Architecture, Version 1.1*. HAVi, Inc, Mai 2001
- [Hec03] HECKER, Rainer. *Referat anlässlich der Eröffnungsveranstaltung zur IFA am 28.8.2003*. 2003
- [Her99] HEROLD, Helmut: *Linux-Unix-Systemprogrammierung*. 2nd Edition. Addison-Wesley, 1999. – ISBN 3-8273-1512-3
- [Hom03] HOMENET2RUN: *Enabling the Interconnected Home - a European vision / HomeNet2Run consortium partners*. 2003. – Whitepaper
- [ICT04] ICT. *IST Project HomePlanet - Hardware and Software Specification of the HAVi Demo Network*. Deliverable HomePlanet-ICT-D9a, TU-Wien, Institut für Computertechnik. Januar 2004
- [IEC98] IEC: *Consumer audio/video equipment – Digital Interface – Part 1: General / IEC*. Geneva, Switzerland, Februar 1998 (IEC 61883-1). – Standard
- [IEE95] IEEE COMPUTER SOCIETY: *IEEE Standard for a High Performance Serial Bus / IEEE Computer Society*. New York, USA, 1995 (IEEE 1394-1995). – Standard. – ISBN 1-55937-583-3
- [IEE98] IEEE COMPUTER SOCIETY: *IEEE Std. 802.1D - Media Access Control (MAC) Bridges / IEEE Computer Society*. 1998 (IEEE 802.1D). – Standard
- [IEE99a] IEEE COMPUTER SOCIETY: *IEEE Std. 802.11 - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications / IEEE Computer Society*. 1999 (IEEE 802.11). – Standard
- [IEE99b] IEEE COMPUTER SOCIETY: *IEEE Std. 802.11a - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment 1: High-speed Physical Layer in the 5 GHz band / IEEE Computer Society*. 1999 (IEEE 802.11a). – Standard
- [IEE99c] IEEE COMPUTER SOCIETY: *IEEE Std. 802.11b - Wireless LAN MAC and PHY specifications: Higher speed Physical Layer (PHY) extension in the 2.4 GHz band / IEEE Computer Society*. 1999 (IEEE 802.11b). – Standard
- [IEE01] IEEE COMPUTER SOCIETY: *IEEE Std. 802.11b - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment 2: Higher-speed Physical Layer (PHY) extension in the 2.4 GHz band - Corrigendum1 / IEEE Computer Society*. 2001 (IEEE 802.11b). – Standard

- [IEE02a] IEEE COMPUTER SOCIETY: IEEE Standard for a Control and Status Registers (CSR) Architecture for Microcomputer Buses / IEEE Computer Society. New York, USA, September 2002 (IEEE 1212-2001). – Forschungsbericht. – ISBN 0-7381-3100-8
- [IEE02b] IEEE COMPUTER SOCIETY: IEEE Std. 802.3 - Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications / IEEE Computer Society. 2002 (IEEE 802.3). – Standard
- [IEE03] IEEE COMPUTER SOCIETY: IEEE Std. 802.11g - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band / IEEE Computer Society. 2003 (IEEE 802.11g). – Standard
- [ISO99] ISO/IEC: Programming languages - C / ISO/IEC. Geneva, Switzerland, 1999 (ISO/IEC 9899:1999). – International Standard
- [IT94] ITU-T: *ITU-T Recommendation X.200 - Open Systems Interconnection - Basic Reference Model*. International Telecommunication Union, 1994. – also: ISO/IEC International Standard 7498-1
- [IW01] IYER, Prakash ; WARRIER, Ulhas: *InternetGatewayDevice Device Template v1.01 for UPnP v1.0*. UPnP Forum, 2001
- [Joh99] JOHANSSON, Martin. *HiperLAN/2 - The Broadband Radio Transmission Technology Operating in the 5 GHz Frequency Band*. WWW: <http://www.hiperlan2.com/>. 1999
- [Joh02] JOHANSSON, Peter: P1394.1 Draft Standard for High Performance Serial Bus Bridges / IEEE Computer Society. 2002 (1.04). – P1394.1 Draft
- [KCK+02] KIM, D. S. ; CHO, G. Y. ; KWON, W. H. ; KWAN, Y. I. ; KIM, Y. H.: Home network message specification for white goods and its applications. In: *IEEE Transactions on Consumer Electronics* 48 (2002), Februar, Nr. 1, S. 1-9. – ISSN 0098-3063
- [KDP02] KABITZSCH, Klaus ; DIETRICH, Dietmar ; PRATL, Gerhard: *LonWorks - Gewerkeübergreifende Systeme*. VDE Verlag GmbH, 2002. – ISBN 3-8007-2669-6
- [KK99] KASTNER, Wolfgang ; KRÜGEL, Christopher: Jini Connectivity for EIB Home and Building Networks: From Design to Implementation. In: *EIB Scientific Conference, Germany, 1999*
- [KL01] KASTNER, Wolfgang ; LEUPOLD, Markus: Discovering Internet Services: Integrating Intelligent Home Automation Systems to Plug and Play Networks. In: *Innovative Internet Computing Systems, International Workshop, IICS 2001, Ilmenau, Germany, 2001*, S. 67-78

- [KON02] KONNEX ASSOCIATION. *KNX Specifications, Volume 3, Part 7: Interworking*. 2002
- [lib] *The XML C parser and toolkit of Gnome*. – WWW: <http://xmlsoft.org>
- [Lin01] LINDNER, Manfred. *Vorlesung Datenkommunikation und Netzwerke*. Vorlesungsskript - Institut für Computertechnik, TU Wien. 2001
- [Lip98] LIPINSKI, Klaus: *Lexikon Lokale Netze*. 2. MITP-Verlag GmbH, 1998. – ISBN 3-8266-4049-7
- [Loe03] LOEWE AG. *Loewe Wireless Home Automation Starter Set*. WWW: <http://www.loewe.de>. 2003
- [Lon96a] LONMARK: *Functional Profile: Switch*. LonMark Interoperability Association, 1996. – WWW: <http://www.lonmark.org>
- [Lon96b] LONMARK: *Functional Profile: Thermostat*. LonMark Interoperability Association, 1996. – WWW: <http://www.lonmark.org>
- [Lon02a] LONMARK: *Application Layer Interoperability Guidelines*. LonMark Interoperability Association, 2002. – WWW: <http://www.lonmark.org>
- [Lon02b] LONMARK: *Layer 1-6 Interoperability Guidelines*. LonMark Interoperability Association, 2002. – WWW: <http://www.lonmark.org>
- [Lon03] LONMARK: *Device Interface File Reference Guide*. LonMark Interoperability Association, 2003. – WWW: <http://www.lonmark.org/products/guides.htm>
- [Mat02a] MATSUSHITA: *Command Specifications For AVHDD-G3YZ*. Matsushita Electric Industrial Co., Ltd, 2002
- [Mat02b] MATSUSHITA: *Descriptor Specifications For AVHDD-G3YZ*. Matsushita Electric Industrial Co., Ltd, 2002
- [Mes03] MESSE BERLIN GMBH. *IFA Highlights - Interessante Trends und Exponate zur Internationalen Funkausstellung in Berlin 2003*. 2003
- [Mic00] MICROSOFT: *Universal Plug and Play Device Architecture, Version 1.0*. Microsoft Corporation, Juni 2000. – WWW: <http://www.upnp.org/>
- [Mil01] MILLER, Franz: Vernetzt und smart: vom Kühlschrank bis zur Badewanne - Innovationszentrum Intelligentes Haus (inHaus Duisburg). In: *Frauenhofer Magazin* (2001), Nr. 2, S. 36-38
- [Moa97] MOATS, R.: URN Syntax / Internet Engineering Task Force. 1997 (2141). – RFC

- [Moc87a] MOCKAPETRIS, P. V.: Domain names - concepts and facilities / Internet Engineering Task Force. 1987 (1034). – RFC
- [Moc87b] MOCKAPETRIS, P. V.: Domain names - implementation and specification / Internet Engineering Task Force. 1987 (1035). – RFC
- [Nix03] NIXON, Toby: *UPnP Forum - State of the Union*. UPnP Forum Summit Presentation, 2003. – WWW: <http://www.upnp.org/>
- [NMV02] NIKOLOVA, M. ; MEIJS, F. ; VOORWINDEN, P.: Remote mobile control of home appliances. In: *International Conference on Consumer Electronics, 2002. ICCE. Digest of Technical Papers.*, 2002, S. 100–101
- [NMV03] NIKOLOVA, Mariana ; MEIJS, Frans ; VOORWINDEN, Peter: Remote mobile control of home appliances. In: *IEEE Transactions on Consumer Electronics* 49 (2003), Nr. 1, S. 123– 127
- [NSA02] NAKAJIMA, T. ; SATOH, I. ; AIZU, H.: A Virtual Overlay Network for Integrating Home Appliances. In: *Symposium on Applications and the Internet (SAINT 2002)*, 2002, S. 246–253
- [Obj98] OBJECT MANAGEMENT GROUP: *Common Data Representation (CDR), CORBA specification 2.0*. Object Management Group (OMG), 1998
- [O'D01] O'DRISCOLL, Gerard: *The Essential Guide to Home Networking Technologies*. Prentice Hall PTR, 2001. – ISBN 0–13–019846–3
- [Pfe99] PFEIFER, Tom: Internet - Intranet - Infranet: A Modular Integrating Architecture. In: *7th IEEE Workshop on Future Trends of Distributed Computer Systems (FTDCS)*, 1999, S. 81–88
- [Plu82] PLUMMER, D. C.: Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware / Internet Engineering Task Force. 1982 (826). – RFC
- [Pos80] POSTEL, J. B.: User Datagram Protocol / Internet Engineering Task Force. 1980 (768). – RFC
- [Pos81a] POSTEL, J. B.: Internet Protocol / Internet Engineering Task Force. 1981 (791). – RFC
- [Pos81b] POSTEL, J. B.: Transmission Control Protocol / Internet Engineering Task Force. 1981 (793). – RFC
- [Pro02] PROCOMP: *User Guide for FireTV (DVB-S, AV/C Tuner)*. Procomp Group Hungarian Office Ltd, 2002
- [Rah01] RAHM, Helmut: *Jini - HAVi Bridge*, Vienna University of Technology, Master Thesis, 2001

- [RHB02] RIGOLE, Peter ; HOLVOET, Tom ; BERBERS, Yolande: Using Jini to integrate home automation in a distributed software-system. In: *Fourth International Conference on Distributed Communities on the Web - DCW 2002 (Sydney)*, Springer-Verlag Berlin Heidelberg, 2002, S. 291–304
- [Rie02] RIEMANN, Uwe. *Streaming Set-Top-Box with HAVi-UPnP Gateway Functionality - Deutsche Thomson-Brandt GmbH - Prototype Presentation*. WWW: <http://www.homenet2run.org>. 2002
- [RMK⁺96] REKHTER, Y. ; MOSKOWITZ, B. ; KARREBERG, D. ; DE, G. J. ; LEAR, E.: Address Allocation for Private Internets / Internet Engineering Task Force. 1996 (1918). – RFC
- [Ros02a] ROSE, Bill: Consumer Requirements for Home Networks. In: *4th IEEE International Workshop On Network Appliances*, 2002
- [Ros02b] ROSE, Bill. *Wireless Home Entertainment - Stuck In A Traffic Jam?* WWW: http://www.broadbandhomecentral.com/report/backissues/Report0212_3.html. Dezember 2002
- [Ros03] ROSE, Bill: *Connected Home Entertainment - Myths, Hype and Reality*. UPnP Forum Summit Presentation, 2003
- [Sch03] SCHLIEPKORTE, Hans-Jürgen: Das intelligente Heim: Einbindung von Haushaltsgeräten. In: *BUS SYSTEME* 10 (2003), Nr. 3, S. 146–147
- [SH03] SAINT-HILAIRE, Ylian: *Intel Tools for UPnP Technology*. Intel Corporation, Februar 2003. – WWW: <http://www.intel.com/technology/UPnP>
- [Sie01] SIEMENS: *BCU Help File*. Siemens AG, 2001
- [Ski02] SKICZUK, Peter: *Network Protocol Architecture for Home Access Points*, Vienna University of Technology, Ph.D. dissertation, 2002
- [Sta98] STALLINGS, William: *Operating Systems - Internals and Design Principles*. 3rd Edition. Prentice Hall Inc., 1998. – ISBN 0–13–917998–4
- [Sun98] SUN MICROSYSTEMS: *Java Object Serialization Specification*. Sun Microsystems, Inc., 1998
- [Sun02] SUN MICROSYSTEMS: *Java Remote Method Invocation Specification*. Sun Microsystems, Inc., 2002
- [Sun03a] SUN MICROSYSTEMS: *A Collection of Jini Technology Helper Utilities and Services Specifications Version 2.0Beta*. Sun Microsystems, Inc., 2003
- [Sun03b] SUN MICROSYSTEMS: *Jini Architecture Specification Version 2.0Beta*. Sun Microsystems, Inc., 2003

- [Sun03c] SUN MICROSYSTEMS: *Jini Device Architecture Specification Version 2.0Beta*. Sun Microsystems, Inc., 2003
- [Tan00] TANENBAUM, Andrew S.: *Computernetzwerke*. 3. Pearson Studium, 2000. – ISBN 3-8273-7011-6
- [Tel01] TELKAMP, Gerrit. *Technologien für das intelligente Haus - Konferenz: Das vernetzte Haus; Wiesbaden 08/01*. 2001
- [Tha01] THALLNER, Bernd: *Ein Linuxtreiber für den Europäischen Installationsbus EIB*, Diplomarbeit an der TU-Wien, 2001
- [TW02] TEGER, Sandy ; WAKS, David J.: End-User Perspectives on Home Networking. In: *IEEE Communications Magazine* (2002), S. 114–119
- [UPn03] UPNP FORUM. *Standardized Device Control Protocols (DCP)*. WWW: <http://www.upnp.org/standardizeddcp/default.asp>. Mai 2003
- [W3C99] W3C: *HTML 4.01 Specification, W3C Recommendation*. World Wide Web Consortium, Dezember 1999. – WWW: www.w3c.org
- [W3C00a] W3C: *Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation*. World Wide Web Consortium, 2000. – WWW: www.w3c.org
- [W3C00b] W3C: *Simple Object Access Protocol (SOAP) 1.1, W3C Note 08 May 2000*. World Wide Web Consortium, Mai 2000. – WWW: www.w3c.org
- [W3C01a] W3C: *XML Schema Part 1: Structures, W3C Recommendation*. World Wide Web Consortium, Mai 2001. – WWW: www.w3c.org
- [W3C01b] W3C: *XML Schema Part 2: Datatypes, W3C Recommendation*. World Wide Web Consortium, Mai 2001. – WWW: www.w3c.org
- [WG00] WISCHY, Markus A. ; GITSELS, Martin: *EIB-Universal Plug & Play Gateway*. Siemens, ZT SE2, Munich, Oktober 2000
- [Wis01] WISCHY, Markus A.: *Siemens UPnP Stack (Java)*. Siemens AG, Oktober 2001. – WWW: <http://www.plugin-play-technologies.com/downloads.html>
- [WT04] WEIS, Peter ; TOLLOSCHKE, Thomas. *HAVi-Rule Engine*. Projektdokumentation Computertechniklabor (Betreuer: Michael Ziehensack), TU-Wien, Institut für Computertechnik. 2004
- [WUB01] WENDORFT, Roli G. ; UDINK, Rob T. ; BODLAENDER, Maarten P.: Remote execution of HAVi applications on Internet-enabled devices. In: *IEEE Transactions on Consumer Electronics* 47 (2001), August, Nr. 3, S. 485–495. – ISSN 0098-3063

- [Wür03] WÜRTEMBERG, Jens. *Das Netzwerk im eigenen Heim*. Elektronik 17/2003, WEKA Fachzeitschriften-Verlag GmbH. 2003
- [WZ03] WEIHS, Manfred ; ZIEHENSACK, Michael: Convergence between IEEE 1394 and IP for Real-time A/V Transmission. In: *Fieldbus Systems and their Applications 2003* IFAC, Elsevier, 2003
- [Zie00] ZIEHENSACK, Michael: *Echtzeit Audio/Video Datenübertragung gemäß HAVi Spezifikation 1.0*, Vienna University of Technology, Master Thesis, 2000
- [Zie02] ZIEHENSACK, Michael: *HAVi HA Service Specification - Light, Version 0.8*. Institute of Computer Technology, Technical University of Vienna, November 2002
- [Zie03a] ZIEHENSACK, Michael: *Home Automation Extension for HAVi, Version 0.3.6*. Institute of Computer Technology, Technical University of Vienna, 2003
- [Zie03b] ZIEHENSACK, Michael. *Multimedia-Heim-Netzwerke - Vortrag bei der OVE/OGMA-Fachtagung: Intelligente Gebäude und Wohnungen - Vernetzt Denken*. Mai 2003
- [Zim80] ZIMMERMANN, Hubert: OSI Referenz Model - The ISO Model of Architecture for Open Systems Interconnection. In: *IEEE Transactions on Communications* COM-28 (1980), Nr. 4
- [ZP02] ZAHARIADIS, Th. ; PRAMATARIS, K.: Multimedia home networks: standards and interfaces. In: *Computer Standards and Interfaces* 24 (2002), S. 425-435
- [ZW03] ZIEHENSACK, Michael ; WEIHS, Manfred: Dynamic Interconnection of Consumer Electronics and Home Automation Networks. In: *Fieldbus Systems and their Applications 2003* IFAC, Elsevier, 2003