

Matthias Wagenmann

Java-Middleware zur Anbindung dedizierter
mobiler Endgeräte an multimediale
Informationssysteme

Diplomarbeit

BEI GRIN MACHT SICH IHR WISSEN BEZAHLT



- Wir veröffentlichen Ihre Hausarbeit, Bachelor- und Masterarbeit
- Ihr eigenes eBook und Buch - weltweit in allen wichtigen Shops
- Verdienen Sie an jedem Verkauf

Jetzt bei www.GRIN.com hochladen
und kostenlos publizieren



Universität Karlsruhe (TH)
Institut für Telematik, Telecooperation Office (TecO)

Java-Middleware zur Anbindung dedizierter mobiler Endgeräte an multimediale Informationssysteme

Diplomarbeit

von
Matthias Wagenmann



Tag der Anmeldung: 15. Januar 1998
Tag der Abgabe: 14. Juli 1998

Hiermit erkläre ich, daß ich diese Arbeit selbständig und ohne Hilfsmittel angefertigt habe. Die verwendeten Quellen sind vollständig im Literaturverzeichnis angegeben.

Karlsruhe im Juli 1998

Zusammenfassung

Sowohl die Mobilkommunikation per Funktelefon als auch Informationssysteme wie das Internet erleben in den letzten Jahren ein enormes Wachstum. Eine befriedigende Lösung, multimediale Informationssysteme auch mobil nutzbar zu machen, existiert aber bisher noch nicht. Dies liegt unter anderem daran, daß teils sehr komplexen Diensten, Dokumenttypen und Protokollen dedizierte Endgeräte gegenüberstehen. Faktoren wie mangelnde Display- oder Speichergröße, schwache Prozessorleistung sowie die in der Regel geringe Bandbreite der Funkverbindung erzwingen eine Adaption bestehender Informationssysteme auf die individuellen Fähigkeiten der unterschiedlichen mobilen Endgeräte.

Aufbauend auf der Untersuchung der Randbedingungen wie bestehende Dienste und Endgeräte und der Betrachtung verschiedener Ansätze zur Adaption wird der Systementwurf für einen Adapterer spezifiziert. Da eine Adaption weder alleine auf Endgeräte- noch auf Serverseite adäquat verwirklicht werden kann, wird dazu eine Middleware entworfen und implementiert. Die Adaptionseinheit sowie die Schnittstellen zu den unterschiedlichen Diensten und Endgeräten werden modular und objektorientiert gehalten, um eine maximale Erweiterbarkeit und Flexibilität zu gewährleisten. Weitere Endgeräte, Dienste oder komplette Adaptionstrategien sollen einfach einzubetten sein. Die eigentliche Adaption wird durch eine Kombination von Transformations-, Reduktions-, Substitutions- und Selektionsmodulen verwirklicht. Durch exemplarische Evaluationen der Extrempositionen des Anwendungsraums des Adapterers wird darauf dessen Adaptionfähigkeit im Gesamtanwendungsraum untersucht.

Abstract

Despite the tremendous growth in mobile computing systems and information systems like the Internet in the recent years there is still a lack of solutions for enabling multimedia systems via mobile communication systems. One reason for this is the fact that the dedicated mobile terminal equipment doesn't match the needs of the highly complex services, data types and protocols that are usual in current non-mobile systems. Insufficient display size, insufficient memory, low processor clock speed and a small bandwidth for data exchange enforce an adaption of the mobile information systems to the individual abilities of the different mobile terminal equipment.

Based upon the analysis of the marginal conditions e.g. existing services and terminal equipment and considering other comparable work, a solution for an adaption system for mobile terminal equipment is provided. Due to the fact that an adaption can not adequately be implemented neither solely on terminal equipment nor on server side, a middleware is developed and implemented. The adaption unit and the interfaces for the numerous services and terminal equipment is kept in a modular and object-oriented way in order to achieve the maximum extendability and flexibility. Further terminal equipment, services or complete adaption strategies should be easy to embed. The actual adaption is achieved by a combination of transformational, reductional, substitutional and selectional modules. The functionality of this adaption process is shown by an exemplary evaluation of the occurring extremes for such applications.

Keywords

- H.5.1 Multimedia Information Systems
- H.3.5 Online Information Systems
- H.5.2 User Interfaces
- H.2.4 Distributed Systems

Inhaltsverzeichnis

KAPITEL 1: EINFÜHRUNG & AUFGABENSTELLUNG.....	1
1.1 Einleitung.....	1
1.2 Motivation.....	1
1.3 Aufgabenstellung.....	2
1.4 Gliederung der Arbeit.....	2
KAPITEL 2: GRUNDLAGEN.....	3
2.1 Randbedingungen	3
2.1.1 Bestehende Formate & Dienste.....	3
2.1.2 Mobile Endgeräte.....	8
2.1.3 Funknetze	12
2.2 Begriffsklärung.....	13
2.2.1 Mobilkommunikation.....	13
2.2.2 Middleware	13
2.2.3 Metadaten.....	14
2.2.4 Adaptionmodell für Dokumente	16
2.2.5 Adaptionmodell für Protokolle	20
2.2.6 Adaptionkriterien.....	21
2.3 Mögliche Ansätze	22
2.3.1 Anpassung auf Server-Seite	22
2.3.2 Anpassung auf Client-Seite.....	23
2.3.3 Erstellung eigener Dokumente	24
2.3.4 Anpassung durch einen Mittler	25
KAPITEL 3: STAND DER TECHNIK.....	27
3.1 Anbindungen	27
3.1.1 EGate.....	27
3.1.2 Smart Messaging.....	29
3.1.3 HDTP/HDML	30
3.1.4 Nokia 9000(i).....	33
3.2 Adaptionstrategien/-ansätze	34
3.2.1 HTML-Spracherweiterungen	34
3.2.2 AltaVista's Sprachadaption.....	35
3.2.3 Realaudio/-video	36
3.2.4 Konverter	36
3.3 Integration	37
3.3.1 Narrow-Band Socket.....	37
3.3.2 WAP.....	39
3.3.3 Personal/Embedded Java.....	41
3.4 Forschungsprojekte	43

3.4.1 Projekt „Mobile Visualisierung“	43
3.4.2 Projekt „MARVIN“	45
3.5 Ergebnisse.....	46
KAPITEL 4: SYSTEMENTWURF.....	49
4.1 Zielsetzung.....	49
4.2 Aufteilung der Adaptionpipeline.....	49
4.3 Entwurf der Pipelinestufen	50
4.3.1 Verbindung von Endgerät zum Mittler	50
4.3.2 Geräteidentifizierer	53
4.3.3 Anfrageadapterer.....	54
4.3.4 Verbindung zwischen Mittler und Informationsdienst.....	55
4.3.5 Fragmentierer.....	56
4.3.6 Bewerter.....	58
4.3.7 Adapterer.....	59
4.3.8 Komposer.....	81
4.3.9 Verbindung vom Mittler zum Endgerät	82
4.4 Zusammenfassung.....	82
KAPITEL 5: REALISIERUNG.....	85
5.1 Was ist Java?	85
5.2 Zielsetzung.....	87
5.3 Klassen-Hierarchie.....	87
5.4 Das Layout.....	88
5.5 Das Plugin-Konzept	90
5.5.1 Funknetz-Gateway	90
5.5.2 Internet-Gateway.....	92
5.5.3 Reduzierer.....	92
5.5.4 Transformator.....	94
5.5.5 Adaptionstrategie.....	94
5.6 Beispieldurchläufe.....	95
5.6.1 Adaption einer einfachen Email für einen PDA.....	96
5.6.2 Adaption einer einfachen Email für ein Smartphone	98
5.6.3 Adaption einer HTML-Seite für einen PDA	101
5.6.4 Adaption einer HTML-Seite für ein Handy	102
5.7 Erweiterungsmöglichkeiten.....	103
5.7.1 Ausbaumöglichkeiten.....	103
5.7.2 Verfügbarkeits- und Geschwindigkeitsoptimierung.....	103
5.7.3 Einbettung	105
KAPITEL 6: ZUSAMMENFASSUNG & AUSBLICK.....	107
ANHANG A: PROGRAMMBESCHREIBUNG.....	109
A.1 Nötige Programmumgebung.....	109

A.2 Benötigte Dateien	109
A.3 Die Helpers-Klasse	111
A.4 Die TreeObject-Klasse	112
A.5 DefaultMutableTreeNode-Klasse.....	113
A.6 Hilfsmethoden für Adaptionstrategien	115
ANHANG B: SPEZIFIKATIONEN	119
B.1 Email-Grammatik	119
B.2 MIME-Grammatik.....	121
B.3 HTML3.2.....	123
ABKÜRZUNGEN	131
ABBILDUNGSVERZEICHNIS	133
TABELLENVERZEICHNIS	135
QUELLEN	137

Kapitel 1: Einführung & Aufgabenstellung

1.1 Einleitung

Weltweit ist der Trend zur viel zitierten Kommunikationsgesellschaft zu beobachten. So nahmen 1997 die privaten Internet-Anschlüsse in Deutschland um über 500 Prozent zu. Dies zeigt, wie groß das Interesse in der Bevölkerung ist, an die unterschiedlichsten Informationen zu gelangen und weltweit kommunizieren zu können. Eine weitere Branche, die momentan hohe Zuwachsraten erzielt, ist die des Mobilfunks. Sowohl das stetig steigende Bedürfnis, immer und überall mobil kommunizieren zu können, als auch die immer vielfältigeren Möglichkeiten der zugrundeliegenden Technik, treiben auch in diesem Marktsegment die Verkaufszahlen in die Höhe. Eine weitere Folge dieser Entwicklung ist, daß die Anbieter mobiler Endgeräte versuchen, ihre Produktpalette möglichst breit zu fächern, um den individuellen Anforderungen potentieller Kunden möglichst gut entsprechen zu können.

Diese Fakten lassen nur erahnen, wie in den kommenden Jahren die Begriffe Kommunikation und Information die Entwicklung unserer Gesellschaft bestimmen werden. Ein weiterer Schritt zur Vernetzung des täglichen Lebens wäre beispielsweise, multimediale Informationssysteme wie das Internet auch mobil erreichbar zu machen.

1.2 Motivation

Im Bereich der mobilen Endgeräte gibt es eine kaum überschaubare Anzahl von Produkten. Von Notebooks über Handheld PCs, PDAs¹, Mobilfunktelefonen bis hin zu Two- und Oneway-Pagern unterscheiden sich diese in ihren Leistungsfähigkeiten enorm. Und auch innerhalb einer Produktkategorie gibt es teilweise erhebliche Funktionsunterschiede.

Des weiteren existieren im Internet eine Vielzahl von verschiedenartigen Dokumenten und Übertragungsprotokollen. So werden beispielsweise im WWW schon standardmäßig einige Grafik- und Soundformate unterstützt. Allerdings werden auch dort immer mehr zusätzliche Formate angeboten, die von den gängigen Browsern dargestellt werden können. Dazu kommen noch die verschiedensten Protokolle zum Austausch von Emails, Übertragung von Dateien, dem Fernsteuern von Computern und anderen Diensten.

Sollen Inhalte aus dem Internet mobil zugänglich gemacht werden, muß gewährleistet sein, daß die in Frage kommenden Endgeräte diese Dokumente auch darstellen können. Mobile Endgeräte sind in der Regel in bezug auf Speicherkapazität, Prozessorleistung, Größe des Displays, Übertragungsgeschwindigkeit und der unterstützten Dokumenttypen in ihren Möglichkeiten beschränkt. Dadurch wird es nötig, eine Adaption der im Internet verbreiteten Informationsinhalte auf die Fähigkeiten des jeweiligen Endgerätes vorzunehmen. Nun stellt sich die Frage, wo eine derartige Anpassung vorzunehmen ist.

Eine Möglichkeit wäre, dies direkt auf der Server-Seite zu tun. Dazu müßte das Endgerät, welches die Anfrage gestellt hat, vom Server identifiziert und das gewünschte Dokument vor der Auslieferung an dessen Fähigkeiten angepaßt werden. Das würde jedoch bedeuten, daß jeder Internet-Server jedes Endgerät unterstützen müßte.

¹ Personal Digital Assistant

Eine weitere Vorgehensweise wäre, die Adaption innerhalb des Endgerätes vorzunehmen. Dazu müssen jedoch alle Dokumente zuerst einmal komplett übertragen werden, unabhängig davon, ob diese im vollen Umfang dargestellt werden können oder nicht. Dies würde die im allgemeinen ohnehin schmalbandige Funkverbindung zum Endgerät unnötigerweise belasten. Außerdem wird dabei dem Endgerät abverlangt, eine Vielzahl von Dokumenten interpretieren und darstellen zu können. Dazu sind ein leistungsstarker Prozessor und ein großer Speicher vonnöten, was eine größere Leistungsaufnahme und mehr Platzverbrauch bedeutet. Dies steht im Widerspruch zur gewollten Mobilität des Endgerätes.

Als dritte Variante bleibt, die Adaption „*im Netz*“ selbst, das heißt zwischen Server und Endgerät, vorzunehmen. Bei diesem Ansatz bleibt es einer weiteren Instanz überlassen, zwischen den vom Server angebotenen Dokumenten und den Fähigkeiten des Endgerätes einen Konsens zu finden. Da ein derartiger „*Mittler*“ beide Seiten, das heißt sowohl Endgerät als auch den Server, entlastet, ist diese Möglichkeit den anderen vorzuziehen.

1.3 Aufgabenstellung

Im Rahmen dieser Diplomarbeit soll eine „*Middleware*“ spezifiziert und implementiert werden, welche die Aufgabe übernimmt, die verschiedensten Dienste den unterschiedlichen Endgeräten anzupassen. Dabei werden verschiedene Adaptionenmechanismen - wie Transformation, Substitution und Reduktion - untersucht und klassifiziert. Ein weiterer Schwerpunkt liegt auf der Untersuchung verschiedener Strategien, durch die Kombination dieser Mechanismen ein optimales Ergebnis zu erreichen. Dabei soll eine möglichst offene Architektur gewählt werden, um auch künftige Endgeräte oder Dienste unterstützen und gegebenenfalls auch andere Adaptionstrategien integrieren zu können.

1.4 Gliederung der Arbeit

Im nachfolgenden Kapitel wird zuerst auf die bestehenden Randbedingungen, das heißt auf die existierenden Endgeräte und Dokumententypen, eingegangen. Außerdem werden die verschiedenen Ansätze, Dokumente den Ausprägungen unterschiedlicher Endgeräten anzupassen, untersucht und bewertet. Dabei werden die grundlegenden Begriffe eingeführt und erklärt. Im dritten Kapitel werden bereits bestehende Adaptionssysteme vorgestellt und auf ihre Fähigkeiten untersucht. Auf den Ergebnissen dieser beiden Kapitel aufbauend wird im vierten Kapitel der Systementwurf für den zu realisierenden Adapter vorgestellt und im Detail spezifiziert. Das folgende Kapitel beschäftigt sich dann mit der Realisierung des zuvor theoretisch hergeleiteten Systems. Abschließend folgt im sechsten Kapitel eine Zusammenfassung der Arbeit und ein kurzer Ausblick.

Kapitel 2: Grundlagen

2.1 Randbedingungen

Um geeignete Adaptionenmechanismen zu finden, ist es unumgänglich, die gegebenen Randbedingungen zu analysieren. Da sind auf einer Seite die im Internet verbreiteten Dokumente, die gegebenenfalls einer Anpassung unterzogen werden müssen. Auf der anderen Seite sind die heute zur Verfügung stehenden Endgeräte mit ihren verschiedenen Möglichkeiten und Restriktionen einer genaueren Untersuchung zu unterziehen.

2.1.1 Bestehende Formate & Dienste

Das Internet wird seit seinem Bestehen immer wieder durch neue Formate und Dienste erweitert, um dem aktuellen technischen Stand und der zur Verfügung stehenden Bandbreite Rechnung zu tragen. Einer der ersten verwirklichten Dienste im Internet war die Möglichkeit, Textnachrichten via Email zwischen einzelnen Teilnehmern auszutauschen. Weitere frühe Dienste waren FTP² und Telnet, deren Aufgabe die Fernsteuerung von Computern ist. Mit der Einführung des WWW³-Dienstes begann dann der kommerzielle Siegeszug des Internets außerhalb der Forschung und Lehre. Es bleibt abzuwarten, welche Dienste das Internet in der Zukunft noch zu bieten haben wird.

Doch auch innerhalb eines bestehenden Dienstes werden immer neue Spezifikationen und Formate verabschiedet, um Neuentwicklungen unterstützen und auf geänderte Bedürfnisse der Benutzer und Anbieter reagieren zu können. So konnten innerhalb des WWW-Dienstes ursprünglich lediglich formatierte Textseiten, die untereinander mit sogenannten Hyperlinks verbunden waren, dargestellt werden. Heute können beliebige Medien wie Grafiken, Animationen, Videos, Töne oder Musik in den verschiedensten Formaten in die Seiten eingebettet werden. Im folgenden sollen die beiden erfolgreichsten Dienste des Internets, Email und WWW, genauer betrachtet werden.

2.1.1.1 Der Email-Dienst

Seit 1982 gibt es mit dem RFC⁴ 822 [Crocker82] einen Standard für die Übertragung von textuellen Nachrichten über das Internet. So ist es heute ohne Probleme möglich, elektronische Nachrichten unter den Internet-Benutzern auszutauschen. Allerdings ist dieser Standard derart beschränkt, daß ein Austausch von Nachrichten, die mehr als nur Textinformationen enthalten, fast unmöglich ist. Das wurde mit dem Aufkommen von Multimediaanwendungen zu einem immer größeren Manko für dieses Verfahren. Für den deutschen Anwender war außerdem störend, daß RFC 822 nur Texte zuläßt, die Zeichen aus einem 7 Bit ASCII⁵ Code und damit keine deutschen Umlaute benutzen.

Mit dem im RFC 1521 [Borenstein93] und im RFC 1522 [Morre93] festgelegten MIME⁶-Standard wurde eine Erweiterung des bisherigen Email-Dienstes geschaffen, um nichttextuelle Dokumente wie Grafik, Audio, Video, usw. zu verschicken. Der herkömmliche Internet-Maildienst wurde dadurch

² File Transfer Protocol

³ Word Wide Web

⁴ Request for Comments

⁵ American Standard Code for Information Interchange

⁶ Multipurpose Internet Mail Extensions

multimediafähig, da mit MIME mehrere Objekte unterschiedlichster Modalität in einer Nachricht versendet werden können. Des weiteren können verschiedene Zeichensätze verarbeitet werden.

Eine Email besteht aus zwei Teilen, dem Kopf (*Header*) und dem Rumpf (*Body*). Der Header enthält Informationen über die Nachricht und entspricht in etwa dem Briefumschlag bei einem herkömmlichen Brief. Im Body steht der eigentliche Inhalt. Die komplette MIME-Spezifikation ist ab Seite 121 im Anhang B aufgelistet.

Der Email-Header besteht aus verschiedenen Feldern, die jeweils eine besondere Bedeutungen haben. So gibt es zum Beispiel die *To*-Zeile mit der Adresse des Empfängers, die *From*-Zeile mit der Absenderadresse oder die *Subject*-Zeile zur Angabe eines kurzen Betreffs. Da nach dem MIME-Standard die verschiedensten Dokumente übermittelt werden können, müssen bei derartigen Emails im Header weitere Angaben zu den im Body stehenden Daten gemacht werden. Die wichtigsten Felder sind hier der Inhaltstyp (*Content-Type*), die verwandte Inhaltskodierung (*Content-Transfer-Encoding*) oder die MIME-Version. Das Inhaltstyp-Feld besteht aus dem Typ und Subtyp des Inhalts der Mail. Es gibt derzeit acht standardisierte Basistypen und für jeden Basistyp eine Reihe von Subtypen. Ein sogenannter Medientyp wird dabei immer durch ein Tupel Basistyp/Subtyp beschrieben. Zu diesem Tupel können noch verschiedene Parameter, wie zum Beispiel der Name der Datei, hinzukommen. Die verschiedenen Basistypen mit den dazugehörigen, wichtigsten Subtypen werden im folgenden vorgestellt:

Text

Bei dem Basistyp „*Text*“ werden standardmäßig zwei Subtypen unterstützt: „*plain*“ und „*richtext*“. Während beim erst genannten Subtyp nur reiner Text im Body erwartet wird, können beim zweiten zusätzlich Formatierungszeichen übermittelt werden. Als Parameter wird üblicherweise der benutzte Zeichensatz angegeben, so zum Beispiel „*charset=us-ascii*“. Dadurch wird es möglich, auch länderspezifische Sonderzeichen zu übertragen und diese beim Empfänger wieder korrekt darzustellen.

Multipart

Wie bereits erwähnt, ist es beim MIME-Format möglich, mit einer Email mehrere verschiedene Dokumente zu übermitteln. Sollen beispielsweise Bilder als Anhang zu einer Textdatei hinzugefügt werden, wird der Subtyp „*mixed*“ benutzt. Soll jedoch eine grafische und als Alternative eine textuelle Wegbeschreibung dem Leser der Email zur Verfügung gestellt werden, ist der Subtyp „*alternative*“ zu benutzen, wie es in Abbildung 1 dargestellt ist.

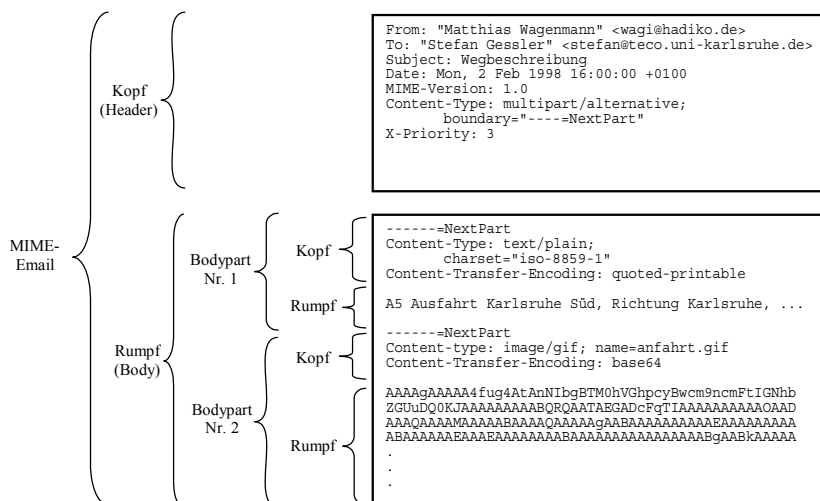


Abbildung 1: Aufbau einer MIME-Email

Ein weiteres Beispiel ist die Möglichkeit, mit diesem Subtyp eine Nachricht als Plaintext und als Rich-text oder HTML zu übertragen. Je nachdem, welches Format der Email-Client unterstützt, kann der Text passend dargestellt werden. Sind die verschiedenen Teile der Email gleichzeitig darzustellen, wird „parallel“ als Subtyp verwendet. Als Parameter muß in jedem Fall eine sogenannte „Boundary“ angegeben werden, anhand derer festgestellt werden kann, wo im Body die verschiedenen Dokumente beginnen und enden.

Image

Unter diesem Basistyp werden die verschiedenen Grafikformate zusammengefaßt. Als Subtypen wurden ursprünglich nur die beiden Grafikformate „JPG⁷“ und „GIF⁸“ unterstützt. Später kam noch das Format „TIFF⁹“ dazu.

Video

Hier werden alle unterstützten Videoformate zusammengefaßt. „MPEG“ und „Quicktime“ sind die von Anfang an unterstützten Formate.

Audio

Beim Audio-Basistyp wurde ursprünglich nur der Subtyp „basic“ unterstützt, welcher vor allem auf Sun-Workstations benutzt wird. Allerdings haben sich Formate wie „WAV¹⁰“, „MPEG¹¹“ oder „MI-DI¹²“ als Quasi-Standards durchgesetzt.

Message

Soll innerhalb einer Email eine weitere Email verschickt werden, wird dieser Typ benutzt. Dies wird oft benutzt, wenn in einer Email Bezug auf eine andere genommen wird, und diese als Anhang mitgeschickt wird. Als Subtyp wird die Kodierung der eingebetteten Email, beispielsweise „RFC822“, angegeben.

Model

Dieser Basistyp wurde erst später zur MIME-Spezifikation dazugenommen. Hier können multidimensionale, abstrakte Objekte, wie zum Beispiel „VRML¹³“, abgelegt werden.

Application

Unter diesem Basistyp werden Programmdateien und Dateien, die sich in keine der obigen Kategorien einteilen lassen, übertragen.

Zusätzlich zu den Basis- und Subtypen können sogenannte X-Typen angegeben werden. Diese müssen mit „X-“ beginnen und können beliebige neue Typen beinhalten. Allerdings müssen Sender und Empfänger selbst dafür sorgen, daß dieser Typ entsprechend interpretiert werden kann. Vor allen Dingen bei den Subtypen haben sich hier viele Quasi-Standard etabliert. Bei kommenden MIME-Spezifikationen werden wahrscheinlich einige als Standard übernommen.

Außer dem Inhaltstyp-Feld muß auch das Inhaltskodierungs-Feld angegeben sein, innerhalb dessen angegeben wird, mit welcher Methode die nachfolgenden Daten kodiert sind. Beispiele wären base64, quoted printable, 8bit, 7bit, binary oder selbstdefinierte Kodierungen (x-foo). Die Felder Inhalts-Beschreibung und -ID sind optional und dienen der Beschreibung der Daten im Nachrichtenrumpf.

⁷ Joint Photographic Experts Group

⁸ Graphics Interchange Format

⁹ Tag(ged) Image File Format

¹⁰ Waveform

¹¹ Moving Picture Experts Group

¹² Musical Instrument Digital Interface

¹³ Virtual Reality Modeling Language

2.1.1.2 Das World Wide Web

Das WWW [Steinov97] hat als weltweites, multimediales Informationssystem die Etablierung des Internets als zentrale Kommunikationsplattform sehr vorangetrieben, wenn nicht gar erst ermöglicht. Es wurde am europäischen Kernforschungszentrum CERN in Genf entwickelt und wird vom W3-Consortium [W3C98] weiter entwickelt. Die Beliebtheit des WWW liegt vor allem an der Tatsache, daß es die hierarchische Struktur des Internets verschleiert und sinnverwandte Informationen auf verschiedenen Servern über Hyperlinks verbindet. Zum Transport der verschiedenen Dokumente zwischen Servern und Clients wird das HTTP¹⁴-Protokoll benutzt. Die Dokumente selbst sind in der Dokumentensprache HTML¹⁵ verfaßt.

2.1.1.2.1 Das HTTP-Protokoll

HTTP ist das Protokoll, über welches Server und Clients miteinander kommunizieren. Eine Konversation zwischen Client und Server besteht aus Client-Anfragen an den Server („request“) und Server-Antworten („response“). Im Regelfall verlangt der Client, ein bestimmtes Objekt zu lesen oder zu schreiben, und der Server antwortet, indem er das Objekt an den Client schickt. Kommt es dabei zu Fehlern, so schickt der Server eine Fehlermeldung anstelle des Objekts. Die momentan aktuelle Version ist HTTP/1.1 [Franks97, Klute95].

Der Request-Block

Beim Request, also der Anfrage des Clients beim Server, unterscheidet man zwischen einem *SimpleRequest* und einem *FullRequest*. Der SimpleRequest entspricht der Funktionalität von HTTP/V0.9 und besteht aus der Zeile

```
GET URL <CR><LF>
```

Der FullRequest FullRequest hat einen etwas komplizierteren Aufbau:

```
Method URL ProtocolVersion <CR><LF> [*<HTRQ Header>] [<CR><LF> <DATA>]
```

Der FullRequest unterscheidet sich durch die Angabe einer Protokollversion vom SimpleRequest, über die das Aussehen der gesamten Anfrage spezifiziert wird. Die Protokollversion HTTP/V1.0 erlaubt eine beliebige Anzahl von HTRQ¹⁶-Headern, über die der Client dem Server bestimmte Mitteilungen machen kann (siehe Tabelle 1), und neben GET weitere Methoden (siehe Tabelle 2).

From	Die Mailadresse des Anwenders
Accept	Feld, über das der Client anzeigt, welche MIME Typen er akzeptiert.
Authorisation	Wird zur Autorisierung verwendet

Tabelle 1: HTRQ-Headers

Get	weist den Server an, das Objekt unabhängig vom Format zu schicken
Head	weist den Server an, nur den HTTP-Header zu schicken
Put	Die im Datenteil vorhandenen Daten sollen vom Server unter der angegebenen URL abgespeichert werden. Dies setzt geeignete Berechtigungen voraus.
Post	Erzeugt ein neues Objekt, welches über einen Link zum spezifizierten Objekt in Beziehung gesetzt wird und diesem untergeordnet ist. POST wird zum Beispiel verwendet, um die Eingaben eines Formulars einem externen Programm zu übergeben.

Tabelle 2: Die verschiedenen Methoden eines FullRequests

¹⁴ Hypertext Transfer Protokoll

¹⁵ HyperText Markup Language

¹⁶ Hypertext Transfer Request

Der Response-Block

Der Response-Block, der als Antwort auf einen Request-Block vom Server zurück an den Client geschickt wird, hat folgenden Syntax:

```
<http version><status code><reason line><CR><LF>
<response header><CR><LF>
<data><CR><LF>
```

Im HTTP-Versions-Feld wird, wie der Name schon sagt, die Versionsnummer des verwendeten HTTP-Protokolls übergeben. Der Status-Code gibt den Status des Servers wieder. So bedeutet der Status-Code 200, daß die Anfrage ordnungsgemäß, das heißt ohne Fehler, ausgeführt werden konnte. In der Reason-Line wird eine kurze Beschreibung des Status-Codes geliefert, beim Code 200 beispielsweise ein einfaches „OK“.

Im Response-Header werden diverse Informationen über das gesendete Objekt geliefert (siehe Tabelle 3). Im Data-Feld werden schließlich die eigentlichen Daten übertragen, also etwa ein HTML-Dokument oder ein GIF-Bild.

Allowed	Erlaubte Methoden auf diesem Objekt (unter der Kennung des Anfragenden)
Public	Erlaubte Methoden für Jedermann
Content-Length	Dokumentlänge
Content-Type	Inhalt des Dokuments nach MIME
Content-Transfer-Encoding	Transfer-Kodierung
Content-Encoding	Dokumentdarstellung
Date	Erstellungsdatum des Objekts
Expires	Verfallsdatum des Objekts
Last-Modified	Letzte Änderungen
Message-Id	Dokumentidentifikation (ähnlich wie bei Newsartikeln)
URI	Adresse des Objekts
Version	unspezifiziert, soll die Version des Objekts angeben
Derived-From	Zeigt bei Objektänderungen auf die ursprüngliche Version
Content-Language	Sprache nach ISO 3316
Cost	Kosten des Objekts (unspezifiziert)
Link	Verweise auf andere Objekte
Title	Dokumenttitel

Tabelle 3: Die Felder des Response-Headers

Das für diese Arbeit wirklich Interessante an dem HTTP-Protokoll ist die Tatsache, daß im Response-Header weitestgehend die selben Felder zur Spezifikation des Dokuments gesendet werden wie bei einer MIME-Email im Email-Header. Über diese Angaben können die Dokumente unabhängig vom zugrundeliegenden Dienst klassifiziert werden. In anderen Diensten, in denen der MIME-Typ des übermittelten Dokuments nicht durch das benutzte Protokoll explizit zur Verfügung steht, kann beispielsweise die Endung des Dateinamens zur Ermittlung des Typs herangezogen werden.