

Multimedia Home Platform

— Standard 1.0.1

Seitdem der „MHP-Standard 1.0“ verabschiedet wurde, gibt es schon wieder Erweiterungen zu 1.0.1 und 1.1, die auf 1.0 aufbauen, sowie Überlegungen für einen neuen „MHP 2.0“-Standard. Der Beitrag liefert aus erster Hand neueste Erkenntnisse, Implementierungsvorschläge und erläutert den Standard unter verschiedenen Gesichtspunkten.

1. Einleitung

In einer Welt der konvergierenden Medien kann es nicht genügen nur die Übertragung von Audio- und Videosignalen in digitaler Form zu spezifizieren, um damit die Zukunft des digitalen Fernsehens zu sichern. Auf Grund der Erfahrungen aus dem Internet erwarten die Nutzer auch im Fernsehen Informationen und Services in interaktiver und multimedialer Form. Die Realisierung interaktiver Services basiert dabei immer auf dem so genannten „Application Programming Interface“ (API). Das stellt ein Set von Funktionen zur Verfügung, wie zum Beispiel Elementen zur Darstellung von Schriften und Graphiken auf dem Bildschirm, auf denen der Programmierer von interaktiven Services bei der Erstellung so genannter „Applikationen“ (Anwendungen) aufbauen kann. Diese Applikationen sind Programme, die auf einem Rechner ablaufen, der inzwischen unbemerkt neben den bekannten Strukturelementen wie Empfangsteil, digitalem Video- und Audiodecoder „in die Empfangsgeräte gewandert“ ist.

Da innerhalb der internationalen Normungsgremien lange Zeit keine Standardisierung für ein solches API erfolgte, entstanden zunächst mehrere proprietäre API-Systeme. Ausgelöst durch die zueinander nicht kompatiblen Systeme, wie OpenTV, Media Highway oder Betanova entstand eine Fragmentierung der Märkte. Fernsehzuschauer, die an den neuen Ser-

Dipl.-Ing. Robert Sedlmeyer ist verantwortlich für die Gebiete „Audio und Video Processing“ im IRT und aktiv in den MHP-Gremien zur Definierung des Standards tätig

vices verschiedener Anbieter interessiert waren, mussten sich mehrere Set-Top-Boxen neben ihr Fernsehgerät stellen, um die Services unterschiedlicher Anbieter angesichts dieser babylonischen Sprachverwirrung nutzen zu können.

Um diesen Zersplitterungseffekt zu vermeiden und die Entwicklung eines „horizontalen Marktes“ zu fördern, der es Herstellern erlaubt, mit konkurrierenden aber kompatiblen Produkten nebeneinander am Markt aufzutreten, beschloss man schließlich bei DVB, einen europäischen Standard für eine „Multimedia Home Platform“ (MHP) zu erstellen. Deren Kernstück ist das API. Die Spezifikation sollte den ganzen Bereich möglicher Geräteimplementationen abdecken, von der Set-Top-Box über das integrierte Fernsehgerät, das keine externe Set-Top-Box mehr benötigt, bis hin zum Multimediacomputer, der über ein „Local Home Network“ mit verschiedenen anderen Geräten verbunden ist.

Zum Erhalt einer marktgerechten MHP-Spezifikation erarbeitete man vor der technischen Spezifikation einen Katalog kommerzieller Anforderungen, der bei der Ausgestaltung der technischen Spezifikation zugrunde gelegt wurde.

Schlüsselemente dieser Anforderungen waren

- offener und frei zugänglicher Standard,
- Trennung von API und „Conditional Access“-System (CA) bei gleichzeitiger Unterstützung verschiedener CA-Systeme, sowie eines „Common Interfaces“,

- Interoperabilität von Plattformen unterschiedlicher Hersteller,
- Darstellbarkeit von Services (Dienst-) verschiedenster Inhalte- (Content-) und Netzeranbieter,
- Skalierbarkeit von Low-End- bis zu High-End-Lösungen,
- Modularität bis hin zur Vernetzung,
- optionale Fähigkeit zur Darstellung von HTML,
- Sicherheit,
 - Schutz der MHP-Plattform gegen Korruption und Viren,
 - Schutz von Inhalten gegen Veränderung und Angriffe,
 - Schutz von Urheberrechten,
 - Schutz des (mitunter kostenpflichtigen) Rückkanals gegen unerlaubte Benutzung durch Services,
 - Schutz der persönlichen Daten des Konsumenten,
- kontrollierter Weg für Erweiterungen des Standards entsprechend neuer technischer Entwicklungen,
- „Upgrade“-Fähigkeit auch über Rundfunk- und Datennetze,
- einfache, benutzerfreundliche Bedienung und
- geringe Kosten.

2. Technische Spezifikation

Zur Spezifikation wurden auf einen „Request for Detailed Specifications“ hin sehr unterschiedliche Vorschläge eingereicht: angefangen von einem HTML-basierten System, über das bereits von DAVIC spezifizierte MHEG bis hin zu einem System mit JAVA als Ausgangspunkt. Der Auswahlprozess führte schließlich zu einer Entscheidung für die Programmiersprache JAVA als Basis für das API, mit zusätzlichen fernsehspezifischen Erweiterungen. Um auch Services auf der Basis von HTML anbieten zu können, wurde eine Option zur Darstellung von HTML-Inhalten vorgesehen. Sie wurde aber in der MHP 1.0.1 noch nicht spezifiziert.

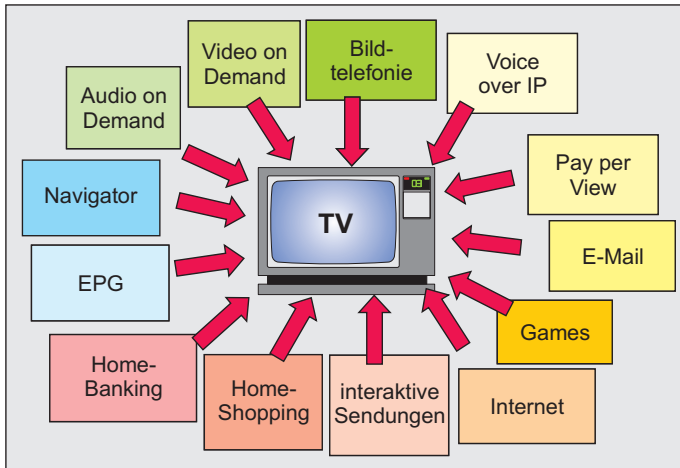


Bild 1. Anvisierte Services auf MHP

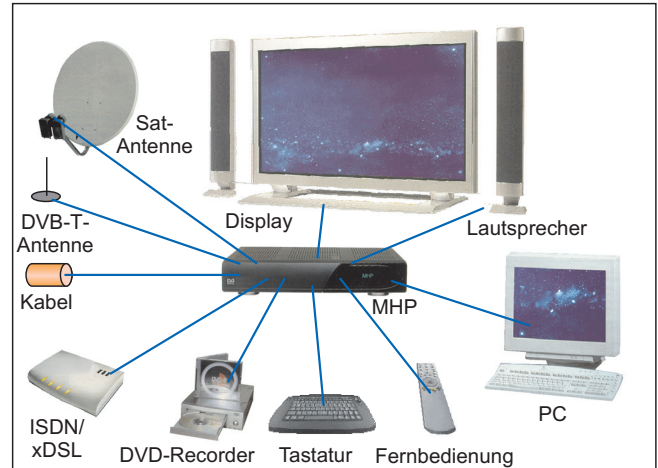


Bild 2. MHP und ihre Verbindungen nach außen

3. Services und Applikationen

Kernstück der Innovation beim Entwurf einer MHP war stets die Nutzung von Interaktivität — sowohl lokaler Interaktivität als auch Interaktivität mit einem Informationsanbieter. Für Letzteres ist ein Rückkanal erforderlich.

Interaktivität stellt das ausschlaggebende Element beim Aufbau neuer Services dar, die dem Fernsehzuschauer eine Menge neuer Benutzungsmöglichkeiten eröffnen. Interaktive Spielshows, an denen er sich aktiv beteiligen kann, erlauben eine weit intensivere Einbindung des Zuschauers in das Geschehen als Sendungen, bei denen er nur zuschauen kann. Darüber hinaus können innerhalb von Fernsehsendungen Hinweise und zusätzliche Inhalte und Services am Bildschirm erscheinen, auf die er per Tastendruck zugreifen kann. Damit lassen sich Informationen „On Demand“ aufrufen — wie aus Inhalten im Internet bekannt. Durch die Möglichkeit, Hinweise zu weiterführenden Inhalten nicht nur anzeigen zu können, sondern auf Wunsch des Zu-

schauers hin auch direkt aufrufen zu können, sind weitere Inhalte für den Kunden viel leichter erreichbar. Er muss nicht mehr die angezeigte Internetadresse mühsam auf Papier aufschreiben, gegebenenfalls in ein anderes Zimmer gehen, um dort den PC zu booten und ähnliches. Das alles dauert mitunter mehrere Minuten, und der Zuschauer muss von der Sendung „weggehen“, um an die Informationen zu gelangen. Das erreicht er in einer MHP auf Tastendruck auf der Fernbedienung — entweder während oder nach der Sendung. Die MHP bietet aber darüber hinaus noch eine Menge weiterer Neuerungen.

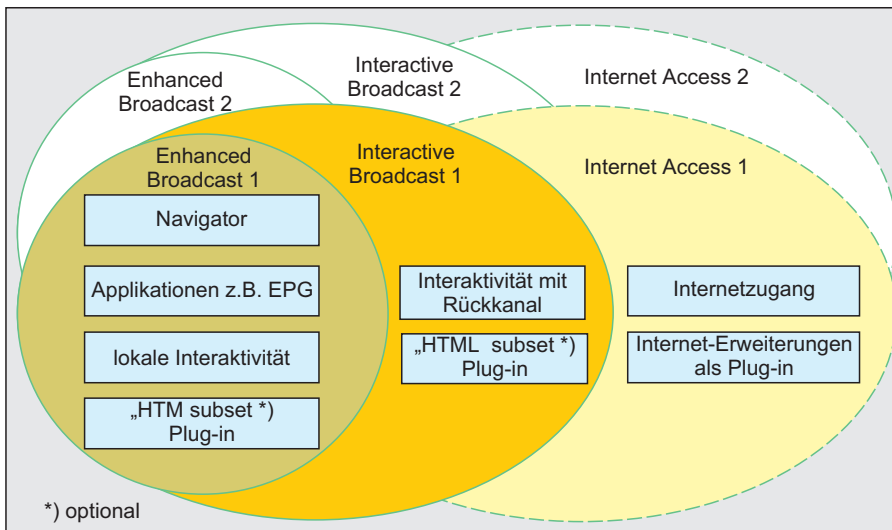
Mit der Digitalisierung von Audio und Video ist eine Übertragung von immer mehr Fernsehprogrammen möglich geworden. Innerhalb dieser Vielzahl kann der Zuschauer inzwischen nur noch mit Hilfe geeigneter Instrumente die ihn interessierenden Inhalte effektiv aussuchen. Ein Navigator, der vom Hersteller in der MHP resident mitgeliefert wird, soll alle verfügbaren Fernsehprogramme und Inhalte in einer Übersicht anzeigen. Darüber

hinaus bringt ein „Electronic Programme Guide“ (EPG), der von einem „Broadcaster“ übertragen wird, und auch Bilder aus Filmen und andere Informationen über die Inhalte darstellen kann, wesentliche Verbesserungen (Bild 1). Bereits zu Beginn der Arbeiten an der MHP dachte man aber auch an interaktive Applikationen, wie zum Beispiel:

- Home Shopping,
- Home Banking,
- Interaktive Fernsehsendungen (Gewinnspiele, Ratespiele, usw.),
- „Pay-per-View“,
- VoD (Video on Demand),
- AoD (Audio on Demand),
- E-Mail,
- Internetzugang,
- VoIP (Voice over IP) und
- Computerspiele.

Die zuletzt genannten Applikationen befinden sich aber bereits relativ weit weg von digitalem Fernsehen und wurden deshalb zwar für weitere Entwicklungen vorgesehen, lagen aber für eine erste Spezifikation noch nicht im Zentrum der Aufmerksamkeit.

Die „Multimedia Home Platform“ war zunächst vorwiegend für Services im Umfeld des digitalen Fernsehens gedacht. Um dem Fernsehzuschauer jedoch die neuen Services anbieten zu können, benötigt eine MHP eine ganze Reihe von Schnittstellen (Interfaces) zur Außenwelt (Bild 2). Ihre Inhalte kann sie über Satellit, Kabel oder terrestrisch genauso empfangen wie über ISDN, xDSL oder ein Modem. Das Feedback zum Inhalteanbieter in interaktiven Applikationen kann ebenfalls über einen dieser Kommunikationswege erfolgen, sofern er rückkanalfähig ist. Die Interaktion zwischen der „Multimedia Home Platform“ und dem Zuschauer geschieht normalerweise über eine Fernbedienung oder gegebenenfalls eine Tastatur. Die Präsentation von Video und Applikationen erfolgt



*) optional

Bild 3. Profile für MHP-Implementierungen

über das Fernsehgerät oder ein Display, von Audio über Lautsprecher. Inhalte sollen natürlich nicht nur aus den genannten Übertragungsmedien abgerufen werden können, sondern auch von Speichermedien wie beispielsweise DVD-Playern. Als zukünftiges Szenario ist an eine Vernetzung mit PCs und anderen Geräten zur audiovisuellen Verarbeitung gedacht.

Zur Abbildung der geplanten Services wurden drei „Profiles“ vorgesehen:

- „Enhanced Broadcast Profile“,
- „Interactive Broadcast Profile“ und
- „Internet Access Profile“.

Von diesen wurden die ersten beiden in der „MHP-Spezifikation 1.0“ im Februar 2000 umgesetzt und im Mai 2000 bei dem europäischen Standardisierungsgremium ETSI eingereicht (**Bild 3**). Die Version 1.0 wurde in der Zwischenzeit mit großem Aufwand und durch Einfügen einer Menge Korrekturen in die „MHP-Spezifikation 1.0.1“ übergeführt, die als Grundlage der im Beitrag durchgeführten Beschreibung dient. Eine detaillierte Auflistung wichtiger obligatorischer und optionaler Funktionalitäten und unterstützter Formate der einzelnen Profiles ist in **Tabelle I** dargestellt.

4. Architektur

Um für die MHP-Implementierungen einen horizontalen Markt zu schaffen, wurde die Möglichkeit der Informationsverarbeitung in einem Mehrschichtenmodell vorgesehen. Festgelegt wurde in der technischen Spezifikation in erster Linie die Grenzschicht zwischen Systemsoftware und Applikation. Es wird explizit darauf hingewiesen, dass die Spezifikation nur die Schnittstellen (Interfaces) zu den Netzwerken beschreibt, sowie zwischen Applikations- und Systemsoftware der MHP — nicht aber die Implementation dieser Elemente.

Bild 4 zeigt eine mögliche Implementation der MHP. In der obersten Schicht befinden sich eine oder mehrere, auch parallel ablaufende MHP-Applikationen. Die Systemsoftware ist hier wieder in mehrere Schichten aufgeteilt, die „JAVA VM“ (Virtual Machine), das Betriebssystem sowie die Treiber, die den Zugriff auf die Hardware ermöglichen. MHP-Hersteller können somit einzelne Schichten der Systemsoftware wie zum Beispiel die „JAVA-VM“ von verschiedenen Softwareprovidern verwenden. Sie können sich aber auch für ein hochintegriertes „embedded“ System entscheiden. In der mittleren Schicht arbeiten neben den Elementen der Systemsoftware auch

Tabelle I. Detaillierte Auflistung von Funktionalitäten und Formaten der einzelnen Profile

| Bereich | Spezifikationen | Enhanced Broadcast Profile 1 | Interactive Broadcast Profile 1 | Internet Access profile 1 |
|---|---|------------------------------|---------------------------------|---------------------------|
| Statische Formate | | | | |
| Bitmap-Bilder | PNG with PNG restrictions | M | M | |
| | PNG without restrictions | — | — | |
| | GIF | — | — | |
| | MPEG-2 I-Frames | M | M | |
| | JPEG with JPEG restrictions | | — | — |
| | JPEG without restrictions | — | M | |
| Audio clips | Monomedia format for audio clips | M | M | |
| Video „drips“ | MPEG-2 Video „drips“ | M | M | |
| Text | Monomedia format for text | M | M | |
| „Broadcast Streaming“-Formate | | | | |
| Video | Video | M | M | |
| Audio | Audio | M | M | |
| Subtitles | Subtitles | M | M | |
| „Broadcast channel“-Protokolle | | | | |
| | MPEG-2 sections | M | M | |
| | DSM-CC User-to-User Object Carousel | M | M | |
| | IP Multicast stack based on: DVB Multiprotocol Encapsulation Internet Protocol (IP) User Datagram Protocol (UDP) | O | Ro | M |
| „Interaction channel“-Protokolle | | | | |
| TCP/IP | Transmission Control Protocol (TCP) Internet Protocol (IP) | — | M | M |
| UDP/IP | User Datagram Protocol (UDP) Internet Protocol (IP) | — | M | M |
| DSM-CC U-U RPG | DSM-CC User-to-User Object Carousel UNO-RPC UNO-CDR | — | O | |
| HTTP | HTTP 1.1 | — | O | M |

M = mandatory, O = optional, Ro = recommended optional

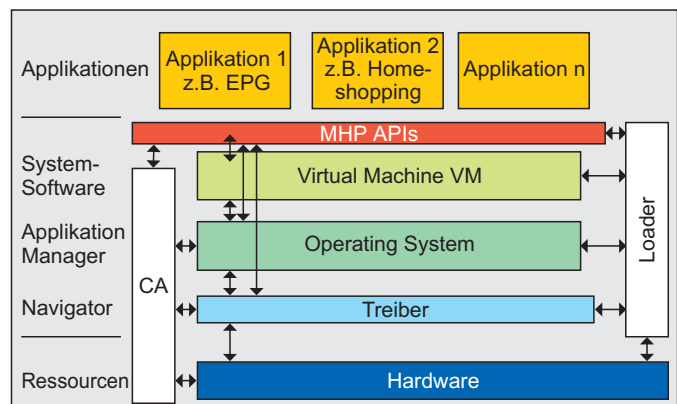
der „Application Manager“. Er startet und stoppt Applikationen und wacht über deren Aktivitäten. Außerdem findet man hier einen je nach Implementation einen unterschiedlich ausgestalteten Navigator.

Ein „Loader“ zum Nachladen von „Updates“ für die Systemsoftware wird typischerweise vorgesehen, um die Zukunftssicherheit der Implementation zu gewährleisten. Er ist aber nicht Bestandteil der MHP-Spezifikation, sondern kann von jedem Hersteller anders implementiert werden. In der untersten Schicht befinden sich schließlich die Ressourcen der MHP mit Elementen wie dem Tuner, MPEG-Decoder usw.

Wie im **Bild 4** gezeigt, sind alle Elemente unterhalb der Applikationen auf der MHP resident und müssen zum Ablauf nicht erst übertragen werden. Das spart Übertragungskapazität auf dem Übertragungsweg, da nur die darüberliegenden Applikationen übertragen werden müssen. Es erfordert aber einen größeren Speicher in der MHP für das API bzw. der einzelnen APIs. Im **Bild 5** wird noch detaillierter das Zusammenwirken der einzelnen Elemente der MHP gezeigt, die durch die Applikationen über die APIs angesprochen werden.

Um die Migration von existierenden, nicht offenen API-Systemen wie „Open-

Bild 4. Allgemeine MHP-Architektur



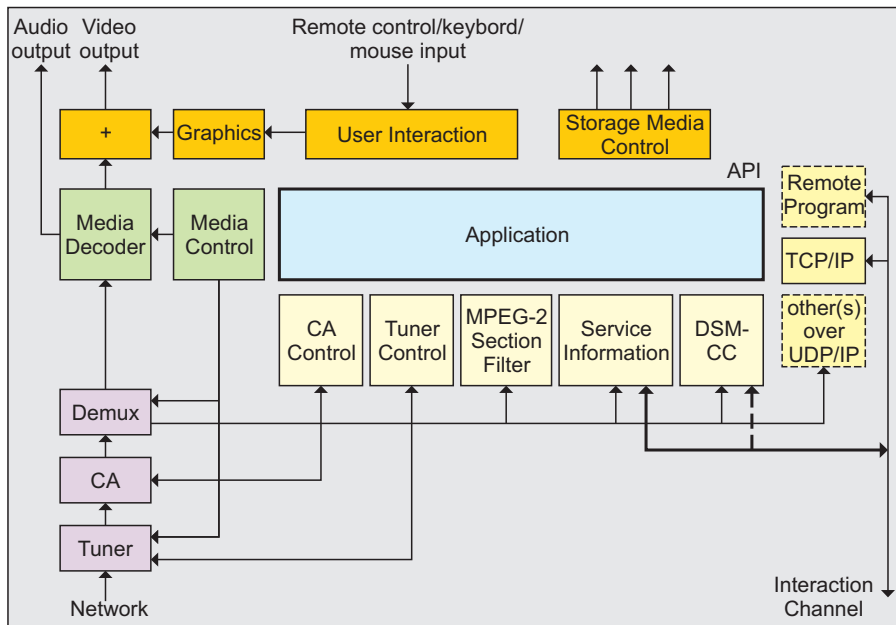


Bild 5. Applikationen und ihr Zusammenwirken mit den einzelnen Elementen der MHP

TV“ oder „Media-Highway“ zu MHP zu erleichtern, aber auch um weitere Content-Formate (zum Beispiel Flash) darstellen zu können (werden von MHP-Spezifikation nicht abgedeckt), beschreibt die „MHP-Spezifikation 1.0“ im Kapitel „Architektur“ bereits Strukturen zur Einbindung so genannter Plug-Ins in die Implementation einer MHP. Es erfolgt dann aber keine detailliertere Spezifikation der Schnittstellen zu Plug-Ins — das geschieht erst in „MHP 1.1“.

5. Transportprotokolle

Da die MHP von der digitalen Rundfunkwelt ausgehend konzipiert wurde, stellen Rundfunkübertragungsprotokolle, wie bei DVB spezifiziert, einen wesentlichen Baustein für die Kommunikation nach außen dar.

Sie unterteilen sich in Protokolle für „Streamed Video und Audio“ und in Protokolle für Daten und Applikationen. Um bei

Datenservices und Applikationen das Vorhandensein aller notwendigen Informationen sicher zu stellen, die für die Funktion notwendig sind, müssen sie zyklisch wiederholt werden, da nicht bekannt ist, wann der Benutzer die MHP einschaltet. Dieses Prinzip wird heute schon beim Teletext verwendet.

Bei der üblichen Rundfunkübertragung in Form von „Streamed Video“ und „Streamed Audio“ wird eine sehr gute Nutzung der Übertragungskapazität erreicht, da eine einzige Übertragung von vielen Teilnehmern genutzt wird (Multicast-Prinzip). Bei Datenservices und Applikationen gilt das natürlich auch, wenn ein Interesse von vielen Zuschauern vorhanden ist. Die Effektivität wird zwar durch die notwendigen Wiederholungen eingeschränkt, man erzielt aber bei einer Übertragung von einer Quelle an sehr viele Senken immer noch ein weit effektiveres Ergebnis als beim Einsatz von „Point-to-Point“-Verbindungen wie im Internet.

Für Applikationen mit geringen Zuschauerzahlen, die über den Rundfunkweg nicht effektiv bedient werden können, ist die Verwendung von „Point-to-Point“-Verbindungen aber sehr gut geeignet. Dem Zuschauer muss dann die Möglichkeit gegeben werden, Informationen auf Anforderung zu bekommen. Die Anforderung geschieht entweder über einen von Hause aus bidirektionalen Informationskanal, in dem auch die Information zum Teilnehmer übertragen wird, oder über einen Rückkanal, der zusammen mit dem unidirektionalen Rundfunkweg einen bidirektionalen Informationskanal bildet. Das ergänzt somit ideal den Rundfunkübertragungsweg, der seine Vorteile in der Massenkommunikation ausspielt. Als Übertragungsprotokolle verwendet man „Interaction Channel“-Protokolle, wie zum Beispiel das „Internet Protocol“ (IP) (Bilder 6 und 7).

6. Formate von Inhalten (Content)

Natürlich können MPEG-2-codierte Audio- und Videostreams, wie sie zur Übertragung von Fernsehprogrammen in digitaler Form seit längerer Zeit verwendet werden, von einer MHP dargestellt werden. Um interaktive Services interessant gestalten zu können, benötigt man aber darüber hinaus die Möglichkeit zur Darstellung von Graphiken, Bildern und Videosequenzen, die nicht als kontinuierliche Streams angeliefert werden, sondern als Datenpakete mit endlicher Länge. Elektronische Programmführer (EPGs) beispielsweise werden erst wirklich attraktiv, wenn sie bebildert sind, oder noch besser, die Ansicht kurzer Filmausschnitte erlauben.

Für die Übertragung und Darstellung von statischen Bitmap-Bildern hat man in „MHP 1.0“ das JPEG- und PNG-Format vorgesehen, auf Grund der Lizenzpflicht aber keine GIF-Bilder. Auch „Flash“-Animationen sind

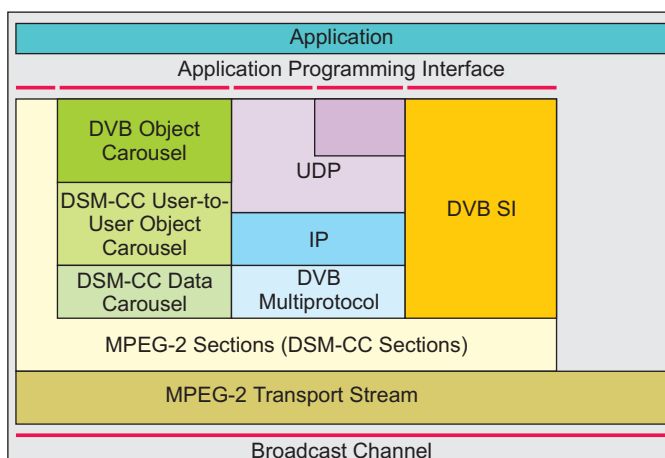


Bild 6 (links). Rundfunk-Übertragungs-Protokolle

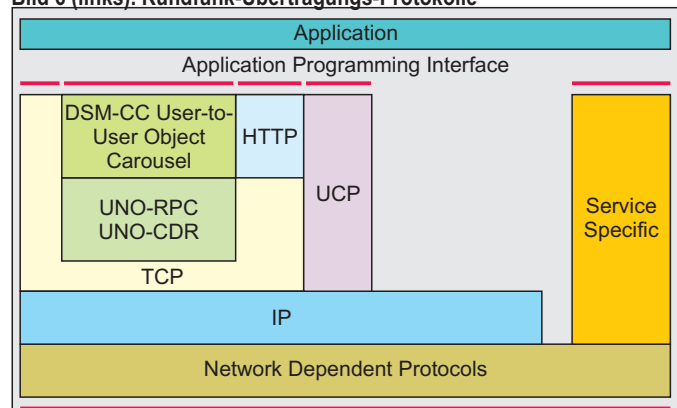


Bild 7. Interaction-Channel-Protokolle

nicht Bestandteil des „MHP 1.0“-Standards. Für die Darstellung von Audio können MPEG-codierte Audioclips verwendet werden. Für Animationen und Videosequenzen, die als Datensatz zum Beispiel innerhalb von MPEG-2-Sektionen verpackt übertragen werden können, hat man I-Frames und Video-„drips“ in MPEG-2 vorgesehen. MPEG-2-I-Frames sind zeitlich aufeinander folgende intra-codierte Bilder, während MPEG-2-Video-„drips“ aus I-Frames und P-Frames bestehen. Zwar kann man damit Einzelbilder nacheinander auf dem Display anzeigen, es lassen sich aber keine Sequenzen in Form von fortlaufenden Videos zeigen. Das würde eine Bildwechselfrequenz von 40 ms erfordern. In der Spezifikation wird aber eine minimale Bildwechsel-frequenz von bis zu 500 ms erlaubt.

Die Einbeziehung von Video-„drips“ hat folgenden Hintergrund: Es ist für die Verwendbarkeit von Animationen in der Praxis von großer Bedeutung, dass die Datenrate so weit wie möglich verringert werden kann, ohne den Aufwand bei der Decodierung allzusehr zu steigern. Das wird schnell ersichtlich, wenn man sich vor Augen führt, dass ein EPG, wie er zum Beispiel von der ARD ausgestrahlt wird, eine Übertragungsrate von etwa 4,5 Mbit/s benötigt. Das ist mehr als 10% der Bandbreitenkapazität eines einzelnen Satelliten-transponders und mehr als üblicherweise für eine Video- und Audiokomponente vorgesehen ist. Die Relevanz der gewählten Bildübertragungsformate wird noch größer, wenn man bedenkt, dass etwa 90% der Übertragungsrate für das Bildmaterial innerhalb eines derartigen EPG verbraucht wird, und nur 10% für den Programmcode der Applikation.

Neben der reichlichen Garnierung von Applikationen mit statischem Bildmaterial wäre eine Darstellung von Filmausschnitten in EPGs wünschenswert. Will man für die Übertragung das MPEG-Format verwenden, so könnte man daran denken, den ohnehin vorhandenen MPEG-2-Decoder nicht nur für die Decodierung von digitalen Fernsehprogrammen, sondern auch für MPEG-2-Datenfiles einzusetzen. Der Decoder müsste dann allerdings auch MPEG-2-Datenfiles verarbeiten können. Das kann je nach Hardwareimplementierung bei einigen zurzeit auf dem Markt verfügbaren Decoderchips aber zu Problemen führen. Alternativ ließe sich ein entsprechender Softwaredecoder einsetzen, der aber eine sehr große Rechenkapazität innerhalb der CPU erfordern würde. Um die Videos nicht permanent online übertragen zu müssen — sondern in kleinen Portionen mit geringerer Datenrate

—, ist darüber hinaus in der MHP ein relativ großer Datenspeicher notwendig.

Aus diesen Gründen hat die Darstellung von Videosequenzen aus Datenfiles bis jetzt keinen Eingang in die Spezifikation gefunden. Da sie aber einen großen Nutzwert, sowohl für den Zuschauer als auch für den „Broadcaster“ bieten würde, weil man damit zum Beispiel Filmausschnitte innerhalb eines EPGs zeigen könnte, wäre es erfreulich, wenn sie vielleicht in einer späteren Version der MHP-Spezifikation doch noch Platz finden würde.

Neben der Darstellung von Bildern und Graphiken ist auch die Anzeige von Texten in Applikationen notwendig. Die abschließliche Darstellung von Text (wie beim Teletext) ist aber für den Zuschauer nicht mehr attraktiv genug. So unterstützt MHP zwar DVB-„Subtitles“, und es ist den Geräteherstellern freigestellt Teletext zum Beispiel mittels eines Teletextdecoders darzustellen, MHP liefert aber keine APIs, die einen Zugriff auf Teletextdaten ermöglichen/erlauben.

Texte und ihre Erscheinungsformen sind aber bei der Präsentation von Inhalten ein wichtiges Gestaltungsmittel. Inhaltehersteller wollen sogar auf eine möglichst große Auswahl von Textschrifftypen zurückgreifen, um den Inhalten eine besondere Note bzw. ein „Branding“ zu verleihen. Um Ressourcen zu sparen, wurde in „MHP 1.0“ aber nur eine Schrifttype, nämlich der „Tiresias“-Font ausgewählt, der in jeder MHP in vier verschiedenen Größen immer resident vorhanden ist. Er wird per „Default“ zur Darstellung von Texten verwendet. Sollen andere Schrifttypen zur Anzeige gebracht werden, so müssen sie in „MHP 1.0“ mit der jeweiligen Applikation übertragen werden. Das gilt genauso für „Buttons“ und andere graphische Darstellungselemente. Damit ist sichergestellt, dass der Inhalte-Designer immer in der Lage ist, das „Look-and-Feel“ selbst zu bestimmen.

7. Starten und Stoppen von Applikationen

Multimediale Services werden in einer MHP durch Applikationen dargestellt. Ausgehend von Rundfunk als zentralem Medium wurde das Starten und Stoppen von Applikationen zunächst mit der Auswahl von Rundfunkservices verknüpft, wie das bereits vom Teletext her bekannt ist. Ein Service besteht jedoch aus verschiedenen Teilen (Audio, Video, Daten und Applikationen), die eine gewisse Zusammengehörigkeit haben und dem Benutzer

vielleicht auch zusammen präsentiert werden sollen.

Bisher werden Rundfunkservices am Fernsehgerät durch „Zapping“ ausgewählt, mittels Up- und Downkeys oder durch die Nummerntasten der Fernbedienung. Springt der Zuschauer so von einem Fernsehprogramm zum nächsten, so wählt er dadurch implizit auch einen neuen „Service“ an.

Bei einer MHP soll „Zapping“ auch möglich sein. Darüber hinaus gibt es die Auswahlmöglichkeit durch einen Navigator oder durch EPG-artige Applikationen. Wie die Auswahl mit einem Navigator im Einzelnen auszusehen hat, darüber sagt die MHP-Spezifikation nichts. Die Ausgestaltung des Navigators ist vollständig implementations-abhängig.

Damit ist die Forderung der „MHP Commercial Requirements“, alle Services im Navigator anzuzeigen und zur Auswahl anzubieten, genau genommen nicht Bestandteil der technischen MHP-Spezifikation. Die Anzeige und Auswahl in einem Navigator kann zum Beispiel „Programm für Programm“ erfolgen oder durch Auswahl eines Bouquets. Bouquets sind in der Regel Ansammlungen von mehreren TV-/Hörfunkprogrammen, MHP-Applikationen und Datenservices eines oder mehrerer Anbieter (wobei die Sache dadurch kompliziert wird, dass unterschiedliche Bouquets auch gemeinsame Services enthalten können).

Wird ein Service angewählt, so geschieht die Präsentation der unterschiedlichen Serviceelemente (Audio, Video, Daten und Applikationen) in einem so genannten Servicecontext. Ein Servicecontext stellt das Umfeld dar, in dem eine Applikation agiert. Kontrolliert wird dieses Agieren von einem „Application Manager“, der darüber wacht, dass verschiedene Regeln eingehalten werden. Innerhalb eines „Servicecontexts“ kann zwar nur jeweils ein Service zu einer Zeit dargestellt werden, es können aber mehrere Applikationen, die sich innerhalb desselben Services befinden, gleichzeitig ausgeführt und dargestellt werden. Der „Broadcaster“ muss jedoch dafür Sorge tragen, dass sich die einzelnen Applikationen nicht gegenseitig stören und dass das gesamte System für den Zuschauer benutzbar bleibt.

Es gibt auch die Möglichkeit eine Applikation nach einem Servicewechsel automatisch zu starten, ohne dass es der Benutzer gesondert veranlassen müsste. Das geschieht dadurch, dass der „Application Control Code“ innerhalb der AIT (Application Information Table) als „Autostart“-Applikation deklariert wird.

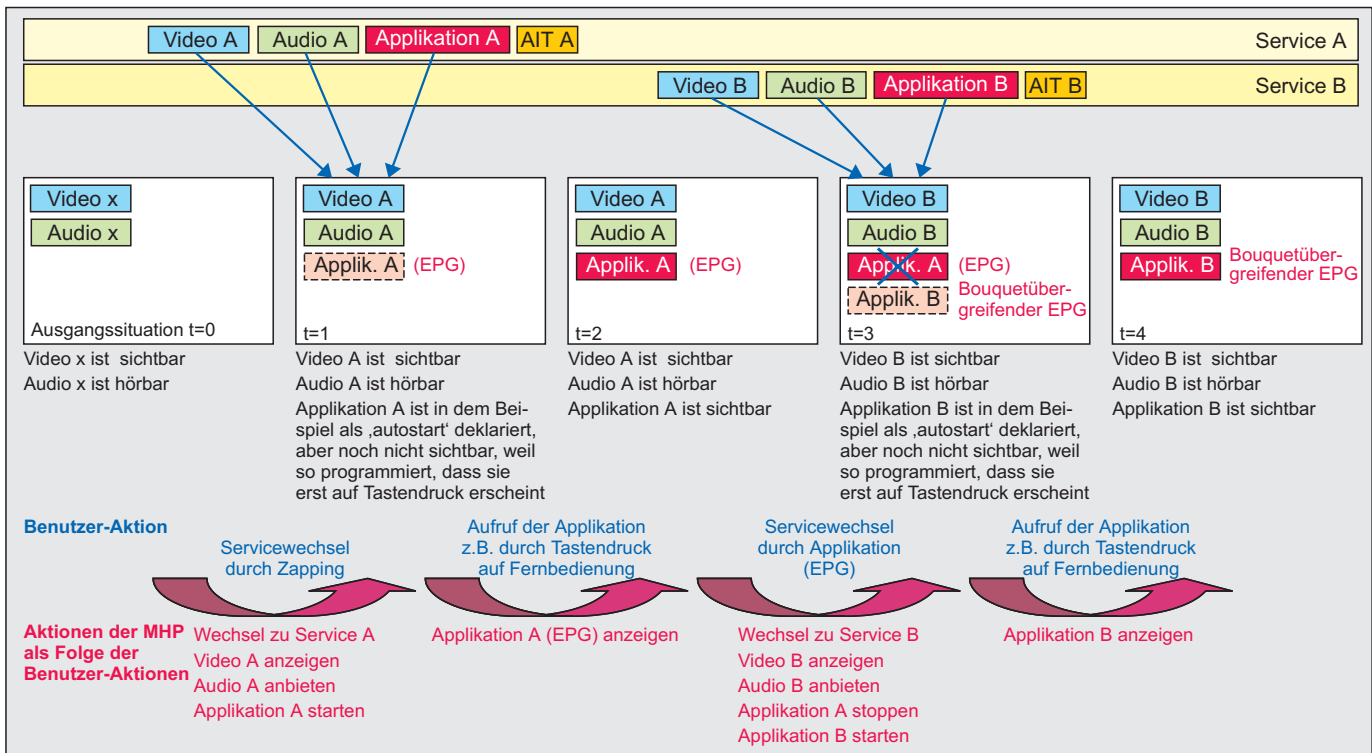


Bild 8. Beispiel zu Start und Stopp von Applikation

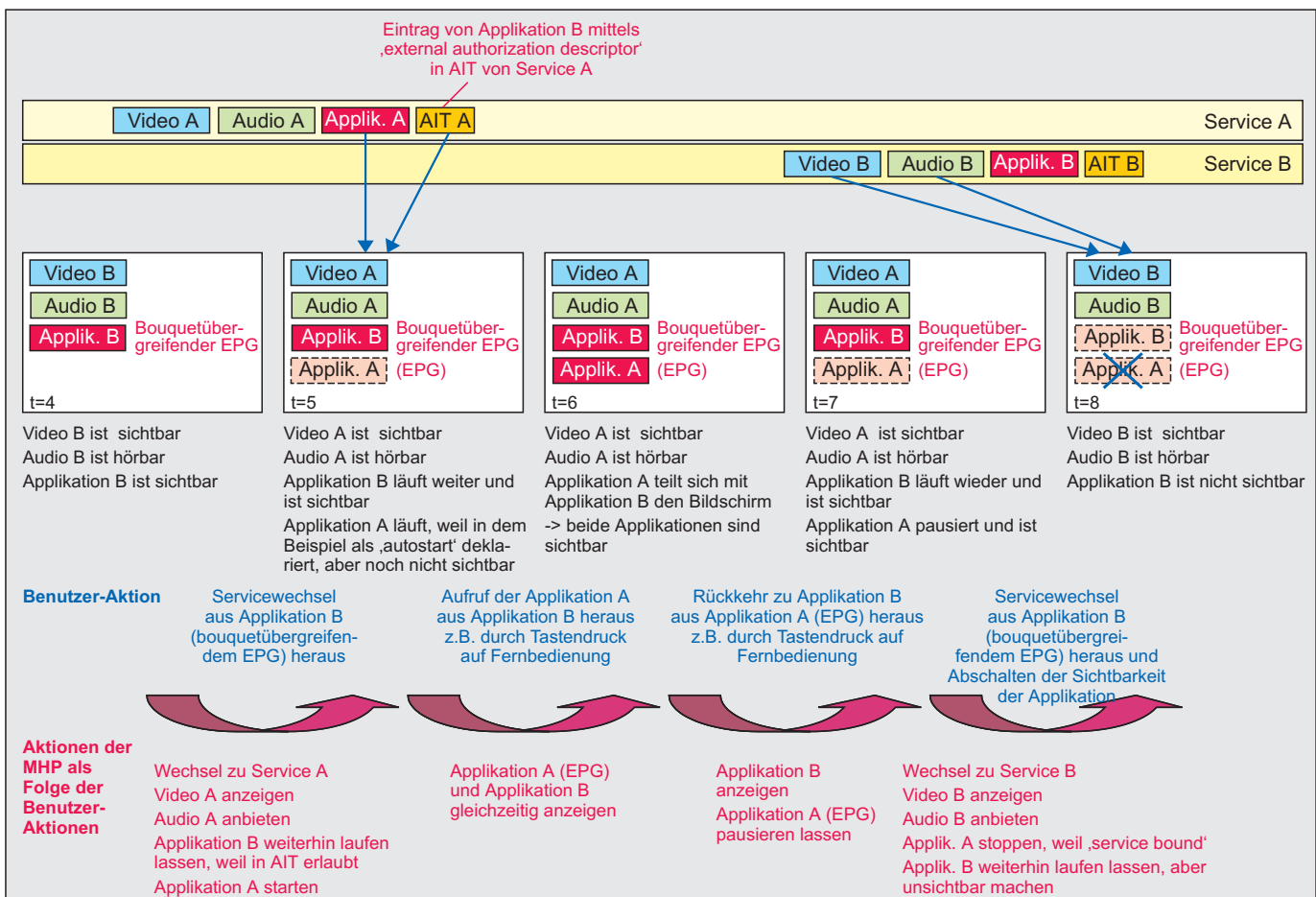


Bild 9. Beispiel zu Start und Stopp eines bouquetübergreifenden EPG, sowie organisiertes Zusammenwirken mit einem 'co-operierenden' Service-gebundenen EPG

Angenommen, es wird ein neuer Service durch „Zapping“ angewählt (wie im **Bild 8** gezeigt) und es gibt einen EPG, der sich neben dem TV-Programm innerhalb dieses Services befindet. Ist der EPG als „Autostart“-Applikation in der „Application Information Table“ (AIT) der „Service Informationen“ eingetragen, so startet die EPG-Applikation. Der Sender kann entscheiden, ob die Applikation sofort sichtbar werden soll oder zum Beispiel erst durch Drücken einer vorher festgelegten Taste auf der Fernbedienung. Aus der EPG-Applikation heraus kann nun agiert werden; es können Texte, Bilder und Video-„drips“ dargestellt werden. Es kann zu anderen „Services“, wie zum Beispiel einem anderen TV-Programm gewechselt werden. Es kann aber auch eine andere Applikation aufgerufen werden, die sich entweder innerhalb desselben „Services“ befindet oder aber in einem anderen „Service“.

Der EPG wird im Beispiel (s. Bild 8) gestoppt und die neue Applikation gestartet. Sie könnte aber auch weiter laufen (**Bild 9**), wie anhand zweier „befeundeter“ EPG-Applikationen gezeigt ist. Wird ein neuer Service selektiert und dadurch der aktuelle ersetzt, so sollen Applikationen des aktuellen „Services“ nur dann weiter ausgeführt werden können, wenn sie auch in dem neuen Service durch den „External Application Authorisation Descriptor“ signalisiert werden.

Diese Regelung hat den Hintergrund, dass eine Applikation nur auf „Services und Applikationen“ verzweigen darf, wenn diese „befeundet“ sind und sie „weiter leben“ soll. Damit kann ein beliebiger Applikationsanbieter zum Beispiel nicht einen EPG anbieten, der auf alle möglichen existierenden Services verzweigt, ohne sich mit den Anbietern dieser Services zu arrangieren. Doch damit nicht genug. Ein Serviceanbieter kann seine Applikationen durch ein „Service Bound Flag“ als „Service“-gebunden deklarieren. Dann wird bei einem Wechsel zu einem anderen Service die Applikation immer gestoppt.

Zum Schutz des Zuschauers vor Fehlbedienung wurde darüber hinaus vereinbart, dass während der Zeit, in der eine Applikation die Nummerntasten oder Up- und Down-tasten der Fernbedienung benutzt, die Funktion des Navigators zum Programmwechsel nicht ausgeführt wird.

Um die Regeln für den Lebenszyklus (Lifecycle) einer „Applikation in JAVA“ beschreiben zu können, hat man in Anlehnung an „Applets“, die sich aufbauend auf der Programmiersprache JAVA im Internet zur Abbildung ablauffähiger Programme etabliert haben, so genannte „Xlets“ definiert. Diese können einen Zyklus mit den Zuständen „loaded“, „paused“, „active“ und „destroyed“ durchlaufen (**Bild 10**).

8. HTML

APIs zur Darstellung von DVB-HTML werden erst in der „MHP-Spezifikation 1.1“ definiert. Das Gleiche gilt für eine standardisierte Schnittstelle zu einem Webbrowser oder einem E-mail-Client.

9. Graphikreferenzmodell

Die MHP stellt dem Applikationsprogrammierer viele Werkzeuge zur Verfügung, um Videobilder, Graphiken, „Buttons“, Texte und Listen auf dem Bildschirm darzustellen. Er kann sie so positionieren, wie er es will. Das Bild, das der Benutzer auf dem Display zu sehen bekommt, ist dabei immer aus mehreren hintereinanderliegenden virtuellen Schichten aufgebaut, die sich zu einem gemeinsamen Bild ergänzen. Als unterste Schicht dient die Hintergrundebene, gefolgt von

der Videoebene und — dem Zuschauer am nächsten liegend — der Graphikebene. Schematisch ist das im **Bild 11** dargestellt.

Wie die einzelnen Schichten zu einem gemeinsamen Bild zusammengefügt werden zeigt **Bild 12**. Gut zu erkennen sind hier auch die Möglichkeiten, die sich durch die Transparenz einzelner Schichten ergeben. Graphische Elemente und Video-komponenten, die nur Teile des Bildschirms ausfüllen, lassen ein dahinter liegendes Videobild an den transparenten Stellen sichtbar bleiben. Neben vollständiger Transparenz (wie hier gezeigt) erlaubt „MHP 1.0“ auch die Definition semi-transparenter Bereiche, die in Applikationen ein dahinterliegendes Videobild zwar noch sichtbar belassen, es aber entsprechend abtönen. Den davor liegenden Schriften kann auf diese Weise beispielsweise eine bessere Lesbarkeit verliehen werden.

Nachdem die MHP auf der Programmiersprache JAVA aufbaut, liegt es nahe, zur Realisierung der Darstellung von Graphiken, Texten und „Buttons“ auf die in JAVA existierenden Sprachelemente als Werkzeuge zur Darstellung zurückzugreifen. Das ist im Wesentlichen die „JAVA-AWT“, die ihrerseits aus so genannten „Lightweight Components“ und „Heavyweight Components“ besteht. Die „JAVA-AWT“ bietet dem Programmierer bereits eine Menge Werkzeuge, um zum Beispiel Linien, Flächen, Texte und „Buttons“ darzustellen.

Sie hat aber auch Nachteile. Der Gebrauch von „Heavyweight Components“ greift auf Funktionen des unterhalb der „Virtual Machine“ liegenden Betriebssystems für das Zeichnen von Graphiken zu. Das verursacht, dass das Aussehen von graphischen Elementen auf unterschiedlichen Plattformen voneinander abweichen kann, wenn diese ein unterschiedliches Betriebssystem benutzen.

Darüber hinaus fehlen innerhalb der „JAVA.awt“ geeignete Möglichkeiten zur Darstellung der im Fernsehbereich am häufigsten vorkommenden Inhalte — den Videos. Dafür wird das „JAVA Media Fra-

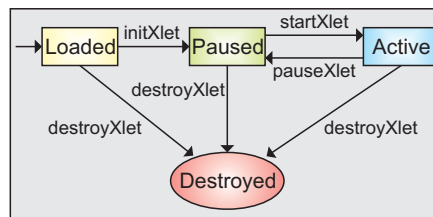


Bild 10. Lebenszyklen von Xlets

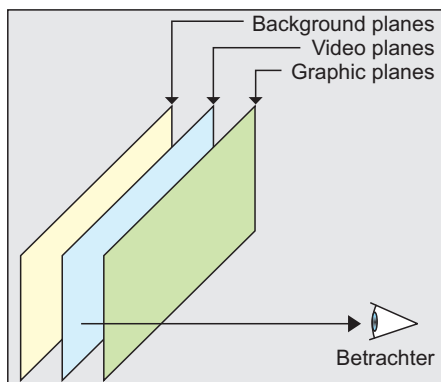
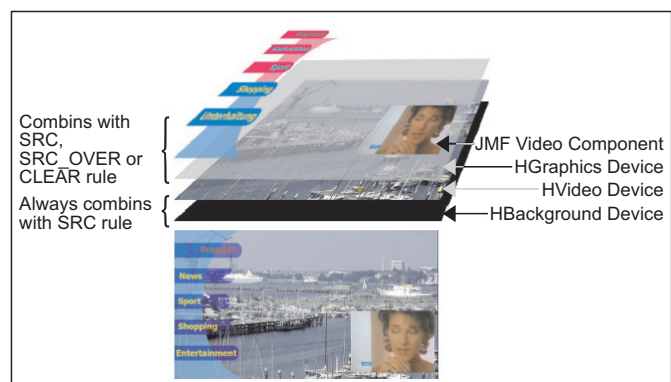


Bild 11 (links). Schematische Darstellung der unterschiedlichen Schichten, die für ein Bild nach dem MHP-Graphik-Referenz-Modell verantwortlich sind
Bild 12 (rechts). Kombination der verschiedenen Schichten zu einem einzigen Bild



mework“ (JMF) eingesetzt. Neben der „JAVA-AWT“ und „JMF“ wurde zusätzlich das durch das HAVI-Consortium standardisierte „User Interface“ (Havi.ui) in „MHP 1.0“ aufgenommen. Es enthält in „H-Look“ Methoden zur Skalierung, Positionierung und Darstellung von graphischen Inhalten, die nur auf „Lightweight Components“ der „JAVA.awt“ zurückgreifen. Der Programmierer von Applikationen hat durch die Übertragung seiner „H-Look“-Informationen die Möglichkeit, das „Look-and-Feel“ seiner Applikationen eindeutig festzulegen, da keine Betriebssystem-abhängigen „Heavyweight Components“ aus „JAVA-AWT“ verwendet werden müssen.

Zur besseren Anpassung an das Benutzungsumfeld einer MHP-Plattform, die durch den Zuschauer in der Regel nicht über eine Computermaus gesteuert wird, sondern über eine Fernbedienung kommuniziert, wurde HAVI anstelle des „User Interfaces“ von „JAVA-AWT“ verwendet.

10. Sicherheitsaspekte

Der Fernsehzuschauer ist beim Konsum von Fernsehprogrammen und Inhalten einen sehr hohen Grad an Sicherheit gewöhnt, auch wenn ihm das normalerweise nicht bewusst wird. In die Aufmerksamkeit rückt die Bedeutung von Sicherheit (der englische Begriff Security ist etwas weiter gefasst als der deutsche) meist erst dann, wenn sie *nicht* vorhanden ist. Gute Beispiele sind Virenattacken und Hackerangriffe, wie sie im Bereich des Internets zur täglichen Praxis gehören. Schaltet der Fernsehzuschauer sein Fernsehgerät ein, so läuft er bisher wenig Gefahr, dass die von ihm betrachteten Inhalte und Nachrichten von Dritten manipuliert sind. Die Rundfunkanstalten achten bei der Wahl der Übertragungswege sehr genau darauf, dass etwaige Manipulationen nach Kräften verhindert werden. Und es kommt in der Regel auch nicht vor, dass das Fernsehgerät einen schwarzen Bildschirm zeigt, weil die Systemsoftware durch einen Virus außer Kraft gesetzt wurde.

Mit der Nutzung von ablauffähigen Programmen als Serviceinhalte sowie mit der Integration eines Rechners in eine „Multimedia Home Platform“, stiegen natürlich die Gefahren für derartige Szenarien drastisch an, wenn man keine Gegenmaßnahmen ergreifen würde. Durch den Ablauf von Programmen auf einer MHP, die vorher übertragen wurden, entsteht ein Gefahrenpotenzial, wie es von der Computerwelt bekannt ist. Für das klassische Fernsehen werden relativ sichere

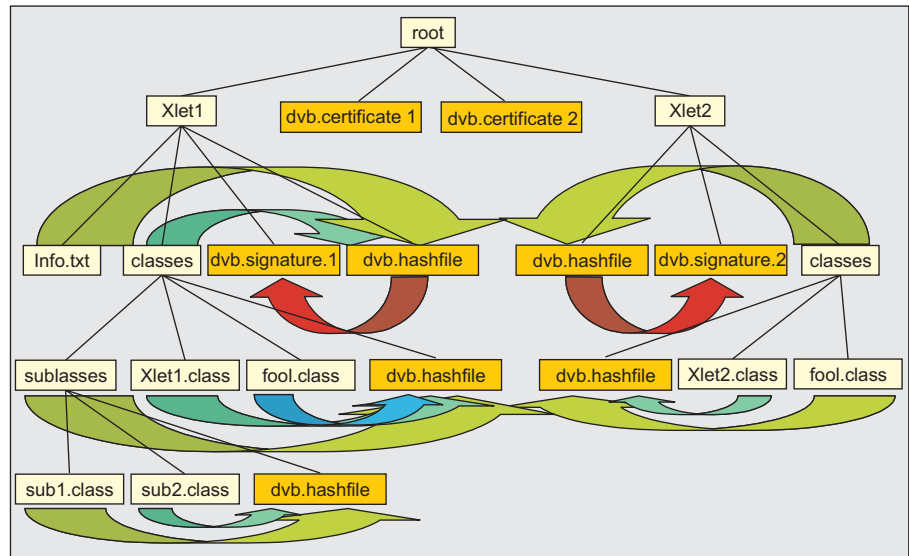


Bild 13. Authentifizierung von Applikationen

Übertragungswege wie Satellit, Kabel oder terrestrischer Ausstrahlung genutzt, die eigentlich geschlossene Übertragungssysteme darstellen. Sollen nun aber auch noch Inhalte oder Applikationen aus dem Internet abgerufen werden, so wird ein effektiver Schutz von Inhalten und Benutzerplattform umso dringlicher.

Deshalb hat man in der MHP-Spezifikation mehrere Vorkehrungen getroffen, um die Sicherheit zu gewährleisten:

Alle Applikationen laufen in einer so genannten „Sandbox“ ab.

Die Aktionen von Applikationen und der Zugriff auf Ressourcen werden durch einen „Security Manager“ überwacht. Er verhindert alle gefährlichen Zugriffe und Aktionen. Ausgangspunkt war hier das JAVA-„Sandbox“-Modell und die Überlegung, dass eine Applikation, die nur in einer „Sandbox“ läuft, keinen großen Schaden anrichten kann. Es ist ihr dann nur erlaubt, Inhalte auf dem Display darzustellen, nicht aber zum Beispiel auf Kosten des Benutzers Telefonnummern anzurufen, seine in der MHP gespeicherten Zuschauerereinstellungen und Präferenzen oder gar seine Kreditkartennummer nach außen weiterzugeben. Diese Sicherheitskriterien haben sich im Internet seit Jahren bewährt, sind jedoch für viele Applikationen im Fernsbereich zu restriktiv. Um die Rechte von Applikationen erweitern zu können, hat man die Möglichkeit des Signierens von Applikationen geschaffen. Während unsignierte Applikationen in einer maximal gesicherten „Sandbox“ laufen, werden bei signierten Applikationen über ein „Permission File“ zusätzlich zugewiesene Rechte genau definiert. Das kann zum Beispiel das Recht zur Nutzung des Modems, des nichtflüchtigen Speichers oder ähnlichem sein.

Unsignierte Applikationen können

- keinen Zugang zu Inhalten des nichtflüchtigen Speichers erlangen,
- nur die Benutzereinstellungen Sprache, Kinderschutzeinstellung, „Default“-Schriftgröße und Ländercode lesen. Sonst sind für unsignierte Applikationen keine Präferenzen des Benutzers lesbar,
- den „Return Channel“ nicht benutzen,
- nicht „tunen“ oder einen anderen Service selektieren,
- nicht mit signierten Applikationen kommunizieren und
- keine Video-„drips“ darstellen.

Signierte Applikationen können:

- Zugang zu Inhalten des nichtflüchtigen Speichers und des Dateisystems erlangen, wenn das im „Permission Request File“ angefordert wurde. Eine Applikation enthält die Dateien, die sie erzeugt hat, und kann für diese Lese- und Schreibrechte für andere Applikationen beliebig vergeben;
- für alle Benutzereinstellungen und Präferenzen Lesezugriff oder Schreib-/Lesezugriff erlangen,
- nur den „Return Channel“ benutzen, wenn die Telefonnummer — die angerufen wird — in einem „Permission Request File“ spezifiziert wurde,
- nur „tunen“ oder einen anderen Service selektieren, wenn es im „Permission Request File“ angefordert wurde,
- mit signierten Applikationen im gleichen Service kommunizieren und
- Video-„drips“ darstellen, wenn es angefordert wurde.

Damit die „Multimedia Home Platform“ eine Entscheidung treffen kann, ob sie diese Rechte einer signierten Applikation gewährt, muss sichergestellt sein, dass

diese auf dem Übertragungsweg nicht verändert wurde. Das könnte durch einen Virus oder bewusste Manipulation geschehen sein. Dazu werden über alle Teile die zu einer Applikation gehören, einschließlich der Daten, so genannte „Hash Files“ gebildet (**Bild 13**). Sie lassen schon die Veränderung eines einzigen Bits in einer Applikation und an Daten erkennen. Damit auch das „Hash File“ selbst sicher ist, wird es mit Hilfe des „Signature Files“ gesichert. Über die mitgesendete „X.509“-Datei kann die MHP die Authentizität der Signatur feststellen. Das „X.509“-Zertifikat identifiziert eindeutig den „Broadcaster“. Hierfür wird das bewährte „RSA Public Key“-Verfahren verwendet, wobei der „Broadcaster“ einen geheimen privaten Schlüssel zur Erstellung der Signatur verwendet. Diese Signatur kann nur mit Hilfe des in der „X.509“-Datei mitgelieferten „Public Keys“ entschlüsselt werden.

Den im „X.509“-Zertifikat gespeicherten RSA-Schlüssel (Public Key) bekommt der „Broadcaster“ von einer nationalen Organisation. Diese wiederum erhält ihn von einer übergeordneten Organisation, wie zum Beispiel in Europa ETSI. Die Hersteller von MHP-Geräten bekommen andererseits so genannte „Root“-Zertifikat zugewiesen.

Eine Anzahl von „Root“-Zertifikaten, von denen alle gesendeten Zertifikate abgeleitet werden müssen, wird bei der Herstellung der MHP in ihren permanenten Speicher eingetragen. Es könnte aber notwendig werden, den Satz von „Root“-Zertifikaten später einmal zu ersetzen, weil sie „gehackt“ wurden, weil eine größere Schlüssellänge benutzt werden soll oder weil sie schon zu alt sind. Ein solche Aufrüstung (Update) geschieht durch die Benutzung von „Root Certificate Management Messages“ (RCMM), die von der MHP nur ausgeführt werden, wenn sie durch mehrere Signaturen authentifiziert wurden.

Damit versucht man Sicherheitsprobleme, wie sie bei DVDs aufgetreten sind, zu vermeiden.

11. Ausblick

Die ursprünglich im Mai 2000 an die Standardisierungsorganisation ETSI weitergereichte „MHP-Spezifikation 1.0“ wurde in die „MHP-Spezifikation 1.0.1“ übergeführt. Grundlage hierfür bildeten die bisher durchgeführten Arbeiten zur Implementation des Standards. Praktisch alle großen Firmen, die auf dem Feld der Unterhaltungselektronik operieren, haben

sich der Umsetzung des über tausendseitigen Standards gewidmet. Ebenso wie die „Broadcaster“ und „Service“-Anbieter, die zwischenzeitlich eine Menge von MHP-Applikationen erzeugt haben, die auf dem Standard basieren.

Um aber nach vorne in Richtung „Internet Access“ und dem zugehörigen „Internet Access Profile“ zu schreiten, wurde in Rekordgeschwindigkeit die „MHP-Spezifikation 1.1“ erarbeitet und bereits bei DVB verabschiedet.

Die Entwicklung im Medienbereich steht jedoch nicht still. Es geht im Gegenteil immer rascher voran. Schon spricht man vom „MHP-Standard 2.0“, der verschiedene Erweiterungen gegenüber den bestehenden Spezifikationen erfahren soll. Zu nennen ist hier das „Local Home Network“, das eine Verbindung verschiedenster Geräte mit einer MHP zum Ziel hat. Die Kommunikation soll hierbei auch drahtlose Verbindungen im Haus mit einschließen. Darüber hinaus wird die Speicherung von Inhalten auf lokalen Festplattenspeichern in späteren Versionen von MHP thematisiert. Letzteres hat bereits unter dem Namen „TV-Anytime“ Aufmerksamkeit erregt.

Vergleich von MHP 1.0 mit MHP 1.1

MHP1.1 definiert die nächste Generation von MHP-Empfangsgeräten. Neben einigen neuen Features wie SmartCard-Unterstützung, Speichern von Applikationen, Plugins und DVB-HTML, verfeinert und verbessert es vor allem bestehende APIs aus MHP1.0. Es implementiert nun endlich auch das bereits in MHP 1.0 in Ansätzen vorhandene Internet-Access-Profile. MHP 1.1 baut dabei auf der ersten Corrigenda-Version von MHP1.0.1 auf. Die wichtigsten Design-Ziele waren eine maximale Rückwärtskompatibilität, eine einfache Integration in bestehende MHP-1.0.1-Systeme, die Eignung für „Embedded“-Systeme sowie die Erweiterbarkeit für zukünftige Technologien wie beispielsweise TV-Anytime.

1. Einleitung

Nur etwas über ein Jahr benötigte die DVB-Gruppe zur Fertigstellung von MHP 1.1 und präsentiert damit eine neue Version, noch bevor die ersten Decoder für MHP 1.0 auf dem Markt sind. Da die Erweiterungen für das „Interaktive & Enhanced-Broadcasting-Profile“ jedoch sehr leicht zu implementieren sind und einfach über ein „Firmware-Update“ auf die Empfangsgeräte übertragen werden können, ist die Hoffnung groß, dass MHP 1.1 bald auf allen MHP-Decodern zu finden sein wird. Neben der umfangreichen für die Hersteller optionalen Erweiterung DVB-HTML und dem nun spezifizierten „Internet-Access-Profile“ bietet es vor allem eine verbesserte Integration des „Interaction-Channels“ (Interaktivität) sowie des „Persistent Storage“, also der Speicherung auf einem vorhandenen Datenträger.

2. Laden von Applikationen über den Rückkanal

Applikationen können in MHP 1.0 nur über den Broadcast-Weg geladen werden, Daten für Applikationen, wie zum Beispiel Bilder, Texte- und Audiofiles, ließen sich darüber hinaus noch zusätzlich über den im „Enhanced Broadcasting Profile“ vorhandenen Rückkanal auf das Empfangsgerät laden. Diese Einschränkung für Applikationen ist in MHP 1.1 nun gefallen. Damit wird es möglich, dem Zuschauer wesentlich umfangreichere Applikationen zu präsentieren oder die Startzeiten von Applikationen wesentlich zu reduzieren. Die für eine Applikation zum Starten wesentlichen Daten und Klassen werden wie bisher gesendet, „exotische“ bzw. seltener genutzte Teile der Applikation können dagegen über den Rückkanal („interaction channel“ oder Interaktionskanal) nachgeladen werden. Selbstverständlich lassen sich auch ganze Applikationen über diesen Kanal laden. Damit könnte ein über den Broadcast-Kanal gesendeter EPG wesentlich umfangreichere Dienste und Inhalte anbieten — auch solche für die ein Broadcasting zu aufwendig erscheint.

3. Speichern von Applikationen

Bereits MHP 1.0 sieht die Möglichkeiten vor, Daten auf einem optional vorhandenen „persistenten“ Speicher auf dem Empfangsgerät zu speichern. Für JAVA-Classfiles war dies bisher analog zum Interaktionskanal — nicht möglich. MHP 1.1 bietet nun Applikationen die Möglichkeit, ihre Klassen über die „org.dvb.application.storage-API“ auf dem MHP-Terminal zu speichern und zu einem späteren Zeitpunkt wieder von dort zu laden. Die Kontrolle über diesen Vorgang behält der Broadcaster über einen in der AIT (Application Information Table) gesendeten „Application Storage Descriptor“. Er enthält eine Versionsnummer, ein Prioritätsfeld, sowie ein Flag mit dem das Empfangsgerät dazu aufgefordert werden kann, auf jeden Fall die gespeicherte Version der Applikation zu laden. Damit kann — entsprechend kraftvoll ausgestattete Empfangsgeräte vorausgesetzt — das Starten von interaktiven Diensten beim wiederholten auswählen eines Services erheblich beschleunigt werden.

4. „Stored Services“

Die Möglichkeit, Applikationen sowohl vom Rückkanal als auch vom persistenten Speicher zu starten, ermöglicht eine für das Fernsehen ganz neue Klasse von interaktiven Diensten — den „Standalone Applications“. Um dabei nicht das Service-Modell in MHP zu durchbrechen, werden sie in so genannten „Stored Services“ gekapselt und treten damit als eigenständige Services, gleichberechtigt zu den „Broadcast Services“ auf. Ein gespeicherter Service kann durch den Broadcaster angelegt werden indem er das „Storage Property“ in dem „Application Storage Descriptor“ auf „stand alone“ setzt. Setzt er es dagegen auf „broadcast related“ wird die zu speichernde Applikation — analog zur MHP 1.0-Applikation — allein über den Broadcast-Service gesteuert. Wie die klassischen Broadcast-Services können auch gespeicherte Services mehrere Applikationen enthalten. Applikationen aus anderen Services erlauben innerhalb ihres Services weiter aktiv zu bleiben bzw.

Dipl.-Ing. Marc Sieburg ist Geschäftsführer der ms² GmbH, Schliersee
Er arbeitet ferner für das IRT in den DVB-Gruppen zur Spezifikation von MHP: „DVB-TAM“, „DVB-Java“, „DVB-Application Lifecycle, Signalling & Security“, MHP-Corrigenda, zeitweise auch in der DVB-HTML und MHP-Gruppe

können selbst in anderen Services aktiv bleiben, falls es diese erlauben.

Möglich werden damit wesentlich komplexere Multi-Bouquet-EPGs und Applikationen, die in MHP 1.0 nur dem Navigator des MHP-Terminal-Herstellers vorbehalten waren.

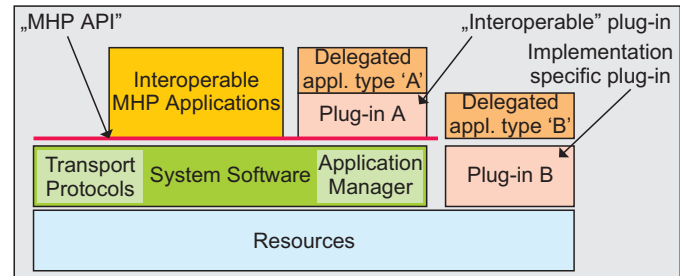
Nicht verwechselt werden dürfen „Stored Services“ mit dem Speichern von Audio- und Video-Diensten, wie es zum Beispiel TV-Anytime definiert. Das bleibt zukünftigen Versionen von MHP vorbehalten.

5. Plugins

Plugins sollen es der MHP ermöglichen, zusätzliche, nicht im Standard enthaltene Datenformate zu decodieren. Typische Kandidaten für Plugins wären zum Beispiel das aus dem Internet bekannte Flash, OpenTV, MHEG und Programme zum Ausführen beliebiger anderer proprietärer Inhaltsformate. Um jedoch die Decodierungsprogramme für diese Formate nicht jedesmal neu auf das Empfangsgerät laden zu müssen, benötigt man eine API, die es erlaubt, diese Programme fest auf der Box zu speichern und für ein bestimmtes Datenformat zu registrieren (**Bild 1**).

Von gespeicherten Applikationen zu Plugins ist es also nur noch ein kleiner Schritt. Und so definiert MHP 1.1 nun auch die in MHP 1.0 bereits im Architektur-Kapitel angekündigten Plugins. Allerdings beschränkte man sich dort auf die Definition der so genannten „interoperablen“ Plugins. Das sind Plugins, die selbst wieder in DVB-J geschrieben wurden. MHP definiert zwei Wege Plugins zu signalisieren, die auch miteinander kombiniert werden können. Zum einen besteht die Möglichkeit, Plugins im „Plugin Descriptor“ zu definieren und sie für bestimmte Applikationstypen zu registrieren, und zwar so, wie sie im „Application Descriptor“ in der AIT signalisiert werden. Allerdings müssen neue Applikationstypen zuvor bei DVB registriert werden. Dies würde die Definition für Plugins in vielen Fällen unnötig komplizieren. Daher besteht zusätzlich die Möglichkeit für Applikationen, die innerhalb eines Plugins ausgeführt werden sollen („Plug-in Applications“), mittels des „Delegated Application Descriptors“ die „Applications ID“ des Plugins, in dem sie ausgeführt werden sollen, anzugeben. Damit kann eine MHP-Applikation leicht zu einem Plugin erweitert werden. Die vorherige zeitaufwendige Registrierung eines neuen Applikationstyps bei DVB kann entfallen.

Bild 1. Architektur der MHP mit Plugins



Zusätzlich zur Möglichkeit Plugins zu signalisieren wurde eine API definiert, um Applikationen, die innerhalb eines Plugins ausgeführt werden, von anderen MHP-Applikationen aus zu adressieren und mit diesen Nachrichten auszutauschen (org.dvb.application.inner-package). Die org.dvb.application.plugin-API definiert die notwendigen Schnittstellen vom Plugin zum MHP-System.

Die Plugin-API wurde dabei so offen gehalten, dass selbst komplexeste Formate, wie das in MHP 1.1 neu definierte DVB-HTML, als Plugin realisiert werden können.

6. DVB-HTML

Die bei weitem umfangreichste Erweiterung in MHP 1.1 ist die Definition DVB-HTML. Es ist für alle drei MHP-Profiles (Enhanced-, Interactive- und Internet-Access-Profile) vorgesehen, ihre Implementierung jedoch für die MHP-Hersteller optional. Damit bieten sich die Kapitel über DVB-HTML in MHP 1.1 auf den ersten Blick zum Überblättern an — was die Einarbeitungszeit in dieses Papier erheblich reduzieren würde. Da es jedoch möglich ist, DVB-HTML als Plugin auf jede beliebige MHP-1.1-Box herunterzuladen und — vorausgesetzt sie hat genug Speicher und CPU-Leistung — auch zu speichern und auszuführen, ist eine weite Verbreitung auf zukünftigen MHP-Systemen durchaus wahrscheinlich. Zusätzlich erscheint durch diese Lösung auch weniger kräftig ausgestatteten MHP-Systemen, die mit einer Implementierung von DVB-HTML noch überfordert wären, die Migration zu MHP 1.1 wesentlich attraktiver. Für den Inhalteanbieter ergibt sich durch DVB-HTML theoretisch die Möglichkeit, bestehende HTML-Autorentools auch für DVB zu verwenden. War die Einführung von DVB-J schon eine wesentliche Vereinfachung der Entwicklung von interaktiven Inhalten, so wird es durch DVB-HTML noch einfacher und damit günstiger, interaktive Zusatzangebote zu erstellen. Im Idealfall können sie sogar für das Internet und DVB gleichzeitig erstellt werden. Durch Verwendung von Stylesheets können die gleichen Inhalte für das Internet

und das Fernsehen jeweils optimal formatiert werden. Praktisch gibt es jedoch einige Einschränkungen. So ist das Internet auf die Bedienung mit der Computer-Maus ausgerichtet, DVB-HTML muss jedoch auch mit der konventionellen Fernsteuerung bedienbar bleiben. Wer dies nicht beim Design der HTML-Seite berücksichtigt, wird unter Umständen nur wenig Benutzer für seine interaktiven Dienste finden.

Auch der verfügbare Platz zur Präsentation der Inhalte unterscheidet sich erheblich zwischen Computer und Fernseher. Sind auf dem Computer flimmerfreie 1024×768 Pixel Auflösung Standard, so stehen auf dem Fernseher nur 720×576 Pixel zur Verfügung. Von denen schneiden allerdings viele Fernseher noch einen Rand weg, und wegen der 50-Hz-Interlaced-Darstellung der Fernseher muss in der Regel ein größerer Zeichensatz verwendet werden, um ein die Augen stark ermüdendes Flimmern zu vermeiden. Bedenkt man noch die beim Fernsehen erheblich größere Distanz des Betrachters zum Bildschirm, bekommt man schon ein Gefühl für die unterschiedlichen Anforderungen, die Fernsehen und Computer an den Web-Designer stellen. Unter diesen Gesichtspunkten erwiesen sich die bestehenden Inhalteformate des Internets als nicht ausreichend für das Fernsehen. Darüber hinaus galt es, die im Internet vorhandenen Inkompatibilitäten verschiedener Browser und die dem Benutzer am PC abverlangte Zustimmung, permanent neue Software zu installieren, zu vermeiden.

Als Basis für DVB-HTML wurde XHTML1.0 gewählt. Das ist der derzeit neueste, auf XML basierende Standard des W3C-Konsortiums und bietet die notwendige technische Basis und Flexibilität für DVB. CSS (Cascading Stylesheets) erlaubt die dynamische Anpassung des Layouts an das jeweilige Medium. So kann im Idealfall der gleiche Inhalt für das Internet und das Fernsehen erstellt werden, das Layout bestimmen für das jeweilige Medium definierte Stylesheets. Da der vom W3C-Konsortium für das Fernsehen vorgesehene Medientyp „Screen“ einige Features von MHP nur unzureichend definiert, wurde er vor allem um die die gra-

phischen Fähigkeiten betreffenden Parameter zum Medientyp „dvb-tv“ erweitert. Ein detailliertes auf der Box bereits installiertes DVB-HTML-Stylesheet soll für ein einheitliches „Default“-Layout der Empfangsgeräte sorgen. Durch das Mitsenden eigener Stylesheets kann der Broadcaster das Layout seines Services dann noch beliebig an seine Bedürfnisse anpassen.

Als Scriptsprache wurde ECMA-Script verwendet. Das ist eine standardisierte und frei verfügbare Form der bekannten Script-Dialekte (zum Beispiel JAVA-Script) des Internets. Mittels DOM (Document Object Model) können solche Scripte auf einzelne Komponenten der DVB-HTML-Seite zugreifen und diese zum Beispiel dynamisch verändern. ECMA-Scripte haben Zugriff auf Cookies mit denen DVB-HTML-Seiten kleinere Informationen temporär auf dem Empfangsgerät speichern können sowie auf die CSS und über eine spezielle Schnittstelle auf alle in DVB-J definierten APIs. Da in ein DVB-HTML-Dokument zusätzlich noch Xlets also MHP-JAVA-Applikationen eingebettet werden können und über die Schnittstelle zwischen ECMA-Script und DVB-J aus ECMA-Script heraus angesprochen werden können, bietet das für den Inhalteanbieter eine ungeahnte Flexibilität. Es wird möglich, Inhalte und Funktionalität bei der Entwicklung zu trennen. Web-Designer entwerfen mit XHTML-Autorentools, die Inhalte und das Layout und greifen auf in eingebetteten Xlets gekapselte komplexere Funktionen zu. Dies können zum Beispiel komplexe Berechnungen, Animationen, oder Funktionen zur Kommunikation mit einem Server sein. JAVA-Entwickler werden auf diese Weise vom Web-Design entkoppelt. Sowohl JAVA-Entwickler als auch Web-Designer können sich der Aufgabe widmen, auf die sie spezialisiert sind. Viele einfachere Funktionen lassen

sich auch direkt in ECMA-Script entwickeln. Eine zusätzliche Anbindung von DVB-J (Xlets) könnte daher überflüssig erscheinen. Bei komplexeren Funktionen wird ECMA-Script jedoch schnell sehr unhandlich und die für JAVA verfügbaren Entwicklungsumgebungen erlauben in diesem Fall eine wesentlich bessere und effizientere Applikationsentwicklung.

Gerade durch diese Integration von DVB-J in DVB-HTML entsteht ein Werkzeug, mit dem sich hochkomplexe interaktive Inhalte auf extrem einfache und kompakte Weise in Module zerlegen lassen und damit besser beherrschbar werden. Die damit mögliche Trennung von Inhalten und Funktionen vereinfacht den Produktionsprozess bei der Content-Erstellung und ist damit ein wichtiger Schritt für die Bereitstellung hoch aktueller interaktiver Angebote.

7. Internet-Access-Profile

Bereits in MHP 1.0 wird ein „Internet-Access-Profile“ definiert, eine detaillierte Spezifikation dieses Profils sucht man jedoch vergeblich. MHP 1.1 holt dies nun nach durch die Definition von Schnittstellen von DVB-J zu „Internet-Client-Applications“. Das org.dvb.internet-Package definiert ein flexibles und erweiterbares Framework, um verschiedene Internet-Clients aus MHP-Applikationen heraus anzusprechen. Die Schnittstellen zu einem WWW-Browser, einem E-Mail-Client sowie einem Usenet-Client wurden bereits definiert. So können MHP-Applikationen einen resident auf der Box installierten WWW-Browser mit einer bestimmten URL starten, Bookmarks setzen, Informationen über den WWW-Browser abfragen, einen E-Mail-Client mit vordefinierten Feldern starten, E-Mails senden sowie einen Usenet-Client an definierter Adresse star-

ten und Newsgruppen registrieren. Die Details der Clients wurden dagegen bewusst nicht definiert.

Kaum jemand wagt eine Prognose über das Internet in fünf oder zehn Jahren, und eine Entscheidung für einen konkreten Internet-Client (zum Beispiel Mozilla) kann morgen schon veraltet sein. Der Standard hat daher nur eine API definiert, die es erlaubt, resident auf der Box installierte Internet-Clients anzusprechen. Er lässt es jedoch bewusst der Wahl des Herstellers und des Marktes überlassen, um welchen konkreten Client es sich tatsächlich handelt und wie der einzelne Hersteller die permanenten neuen Versionen im Internet auf seinem Empfangsgerät umsetzt (zum Beispiel durch regelmäßigen Download, CDs in Fernsehmagazinen, Wartung durch den Fachhändler usw.).

8. Open Card Framework (OCF)

Schon MHP 1.0 erlaubt grundsätzlich eine Anbindung von Smartcards über die CA-API (Conditional Access) über das auch ein „Common Interface“ angesteuert werden kann. Da diese API jedoch mit der Zielrichtung „Conditional Access“ definiert wurde, ist sie für allgemeine Smartcards nicht unbedingt geeignet. Hier entschied sich die DVB-Gruppe für das in der JAVA-Welt etablierte „Open Card Framework“, allerdings in einer abgespeckten Version (Embedded Open Card Framework). Es erlaubt die Ansteuerung beliebiger Smartcards. Möglich wird dadurch zum Beispiel die einfache Anbindung von Karten für Homebanking, Geldkarten oder Speicherkarten. Die Implementierung der Smartcard-Leser-Hardware bleibt allerdings für MHP-Hersteller optional und damit den Kräften des freien Marktes überlassen.