



Technische Universität Ilmenau  
Institut für Theoretische und Technische Informatik  
Fachgebiet Prozeßinformatik

**STUDIENJAHRESARBEIT**

# **Be- und Verarbeitung von Service Informationen im DVB-Stream**

von **Sven Schaepe**  
2003

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>2</b>
<b>2</b>	<b>DVB-Stream</b>	<b>3</b>
2.1	DVB Standards . . . . .	3
2.1.1	DVB-S Standard . . . . .	3
2.1.2	DVB-C Standard . . . . .	3
2.1.3	DVB-T Standard . . . . .	4
2.1.4	Weitere Standards . . . . .	5
2.2	DVB Multiplexer . . . . .	6
2.2.1	Transport-Stream . . . . .	7
<b>3</b>	<b>Service Informationen</b>	<b>9</b>
<b>4</b>	<b>EPG-Daten</b>	<b>12</b>
4.1	EIT . . . . .	12
4.2	EIT-Deskriptoren . . . . .	13
4.3	Datenübertragung . . . . .	13
4.4	Programmzeitung . . . . .	17
4.5	Probleme . . . . .	18
<b>5</b>	<b>Implementierung</b>	<b>20</b>
5.1	Syntax - Beschreibung . . . . .	20
5.2	Aufbau . . . . .	20
5.2.1	Datenerhebung . . . . .	20
5.2.2	libdvbsi . . . . .	20
5.2.3	Datenspeicherung . . . . .	21
5.3	Design . . . . .	21
5.4	Austausch- und Speicherformat . . . . .	22
<b>6</b>	<b>Fazit</b>	<b>25</b>
<b>7</b>	<b>Abkürzungen</b>	<b>26</b>

## Abbildungsverzeichnis

1	Programm- und Transportmultiplex . . . . .	6
2	Transport-Stream Syntax Diagramm . . . . .	8
3	EIT-Aufbau . . . . .	15
4	Epg-Scheduleraufbau . . . . .	23

<i>TABELLENVERZEICHNIS</i>	1
----------------------------	---

5	Aufbau der XML-Datei . . . . .	24
---	--------------------------------	----

## **Tabellenverzeichnis**

1	PID-Werte . . . . .	9
2	EIT Tabellen IDs . . . . .	13
3	Event Information Table . . . . .	14
4	EIT-Deskriptoren . . . . .	16
5	EIT-Durchsatz der Sendergruppen . . . . .	17

## 1 Einleitung

Ein großer Vorteil der digitalen Übertragungstechnik ist neben der verbesserten Bild- und Tonqualität auch das Senden von Zusatzinformationen, mit denen es möglich ist die Bedienung und Programmierung eines digitalen Videorecorders (DVR(4)) zu vereinfachen. Um einen Teil der mit gesendeten Zusatzinformationen innerhalb des digitalen Übertragungssignals soll es in dieser Arbeit gehen, den EPG-Daten(6). Es soll geklärt werden wie die Zusatzinformationen beschafft und aufbereitet werden können, um sie später für Applikationen verfügbar zu machen. Da es sich bei den Daten die aufgezeichnet werden sollen, um DVB-Daten(3) handelt wird im folgenden Abschnitt der Aufbau des DVB-Stream erläutert, der das digitale Übertragungssignal mit den entsprechenden Daten repräsentiert. In den daran anschließenden Abschnitten wird untersucht, wie und welche Daten aufgezeichnet werden können. Danach muss noch geklärt werden, wie die aufgezeichneten Daten gespeichert werden, ob in einer Datei, Datenbank oder ob sie nur im Speicher gepuffert werden sollen. Als letztes ist dann noch zu klären, wie man die gespeicherten Daten anderen Applikationen zur Verfügung stellt.

## 2 DVB-Stream

Digitales Fernsehen (DVB) ist die Ausstrahlung von Bild-, Ton- und Zusatzinformationen als digitale Datensignale. Trotzdem findet die eigentliche Übertragung analog statt, da die digitalen Datensignale zur Kanalanpassung analogen Trägern aufmoduliert werden. Es gibt verschiedene Kanäle auf denen das digitale Datensignal ausgestrahlt werden kann. Zu jedem dieser Übertragungskanäle gibt es einen entsprechenden Standard, siehe 2.1 S. 3.

Neben der Codierung der Bild- und Toninformationen definiert der MPEG-2 Standard [3] ebenfalls das Multiplexen der Bild-, Ton- und Zusatzinformationen zu einem einzelnen Bit-Strom, dem DVB-Stream, siehe 2.2 S. 6.

### 2.1 DVB Standards

#### 2.1.1 DVB-S Standard

Als erstes sei der DVB-S Standard genannt, der seit Januar 1995 gültig ist. Der Standard ist seit Dezember 1994 unter der Bezeichnung ETS 300 421 bei der ETSI(7) normiert. Er beschreibt die Übertragung des digitalen Datensignals via Satellit und die damit verbundenen Anforderungen an die Übertragungsstrecke, wie z. B.

- Technische Anforderungen
- Benutzeranforderungen
  - Eine hohe Übertragungsrate für möglichst viele und hochwertige Dienste.
  - Verschiedene Dienste mit unterschiedlichen Datenraten.
  - Den Erfordernissen entsprechend angepaßter Fehlerschutz.
  - Das Empfangssystem muss günstig und die Antenne möglichst klein und unauffällig sein.

#### 2.1.2 DVB-C Standard

Der DVB-C Standard ist seit Dezember 1994 unter der Bezeichnung ETS 300 429 bei der ETSI normiert. Er beschreibt die Übertragung des digitalen Datensignals via Breitbandkabel und die damit verbundenen Anforderungen an die Übertragungsstrecke. Beim DVB-C Standard wurden die folgenden Nutzeranforderungen besonders berücksichtigt:

- Die vorhandenen Breitbandkabelnetze müssen in ihrer bisherigen Struktur weiterzuverwenden sein, d. h. die Netzwerktopologie müssen unangetastet bleiben.
- Die Einspeisung der digitalen Signale darf bei den analogen Kanälen keine Störungen hervorrufen.
- Es sollen möglichst viele Daten in einem Kabelkanal übertragen werden können, damit die Kompatibilität auch mit den breitbandigen Satellitenkanälen gewährleistet ist.
- Die Kosten für den Nutzer sollen gering sein. Kabel- und Satellitendecoder sollen gleichzeitig auf dem Markt eingeführt werden. Somit ist eine weitgehende Baugleichheit der Kabeldecoder mit den Satellitendecodern gefordert.

### 2.1.3 DVB-T Standard

Der DVB-T Standard ist seit März 1997 unter der Bezeichnung ETS 300 744 bei der ETSI normiert. Er beschreibt die Übertragung des digitalen Datensignals via terrestrischer Übertragung und die damit verbundenen Anforderungen an die Übertragungsstrecke. Wie beim DVB-C Standard wurden auch beim DVB-T Standard die folgenden Nutzeranforderungen besonders berücksichtigt:

- Das System für terrestrisches Digitalfernsehen soll möglichst ähnlich zum Kabel- und Satellitensystem sein.
- Die Übertragung soll mittels Datencontainern mit möglichst großer Kapazität erfolgen. Die Kanalbandbreite soll so gewählt werden, dass ein Nachbarkanalabstand von 8 MHz unterstützt wird.
- Das System ist auf bestmögliche Flächendeckung bei stationärem Betrieb ausgelegt.
- Es sollen nationale Gleichwellennetze möglich sein.
- Der Heimempfänger soll kostengünstig sein.
- Als Option soll eine hierarchische Modulation möglich sein.

### 2.1.4 Weitere Standards

Neben den Standards zum Versenden des DVB-Streams gibt es weitere Standards, die nur erwähnt werden sollen aber ebenfalls wichtig sind.

- DVB-CI Standard:

Dieser Standard beschreibt das Common Interface (CI), das die Verbindung zwischen dem Conditional Access (CA) Modul und dem Decoder Modul darstellt. Dieses Interface ermöglicht den Austausch des CA-Moduls gegen ein anderes. Somit ist der Empfang von verschlüsselten Programmen möglich, die in verschiedenen Verschlüsselungssystemen (z. B. VIACCESS und IRDETO) ausgestrahlt werden. Diese Module erlaubt eine Abstraktion der Verschlüsselung vom Empfangsgerät, was dazu führt, dass die Modulhersteller die CI implementieren müssen ebenso die Gerätehersteller. Dies hat zur Folge das europaweit Empfangsgeräte durch Anschluss eines Dekodermoduls zum jeweiligen Verschlüsselungssystem kompatibel gemacht werden können.

- DVB-TXT Standard:

Da bei analogen Übertragungen der Teletext in den nicht sichtbaren Bildzeilen übertragen wird, der Teletext bei digitalem Fernsehen nach DVB jedoch in einem gemultiplexten Datenstrom enthalten ist, beschreibt dieser Standard die Umsetzung des im gemultiplexten Datenstrom enthaltenen Teletextes in eine analoge Übertragung, so dass es jedes Teletext Endgerät versteht. Der DVB-TXT Standard wird im ETS 300 472 beschrieben.

- DVB-Subtitling Standard:

Diese Spezifikation ist eng verbunden zu der des DVB-TXT. Der Unterschied besteht darin, dass Zusatzdaten als wählbare Untertitel im Bild eingeblendet werden können. In diesem Fall ist jedoch die Set-Top-Box für die Darstellung zuständig, da eine Untertiteleinblendung getrennt vom Teletext bei Fernsehgeräten nicht vorhanden ist.

- DVB-SI Standard:

Der Service-Informationen Standard (DVB-SI, ETS 300 468) spezifiziert die Verarbeitung von Zusatzdaten, die jeden Transportstrom begleiten. Dies sind zum Beispiel der Programmname, Informationen zur laufenden Sendung, Startzeiten des nächsten Films bei Near Video On Demand und systemeigene Informationen (PMT(14), NIT(10) etc.). Diese Daten werden von der Set-Top-Box verarbeitet und entweder automatisch angezeigt oder sie können vom Zuschauer in verschiedenen Menüseiten abgerufen werden.

## 2.2 DVB Multiplexer

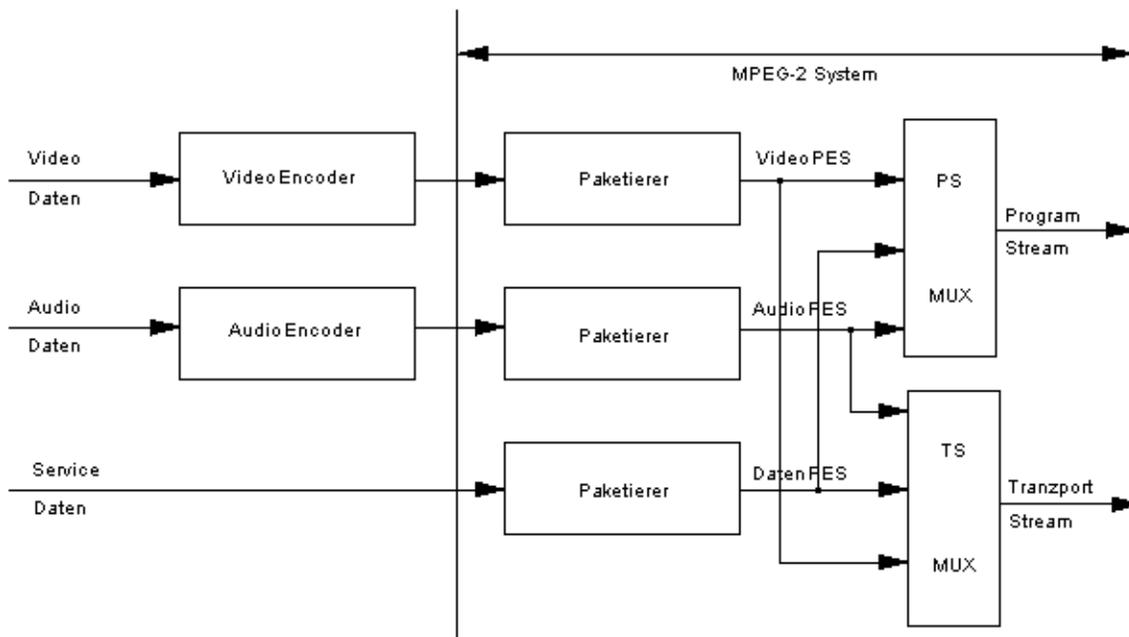


Abbildung 1: Programm- und Transportmultiplex

Das Multiplexen des DVB-Streams ist in Abb. 1 zu sehen. Als Erstes werden die Bild- und Tondaten nach dem MPEG-2 Standard codiert, um anschließend paketierr zu werden. Ebenso die Zusatzinformationen, d. h. die einzelnen Daten-Streams werden in kleine Pakete aufgeteilt und mit Kontrollinformationen versehen. Die resultierenden Datenströme, PES(12), werden anschließend gemulplexiert. Es kann zum einen in einen Programm-Stream, PS(15), und zum anderen in einen Transport-Stream, TS(24) gemulplexiert werden. Die Unterschiede der Mul-

tiplxverfahren sind:

Programm-Multiplex:

- Alle PES haben eine einheitliche Zeitbasis.
- Sind angepasst an einen „relativ“ fehlerfreien Übertragungskanal. (Aufzeichnen auf Festplatte)
- Pakete haben eine variable Länge.

Transport-Multiplex:

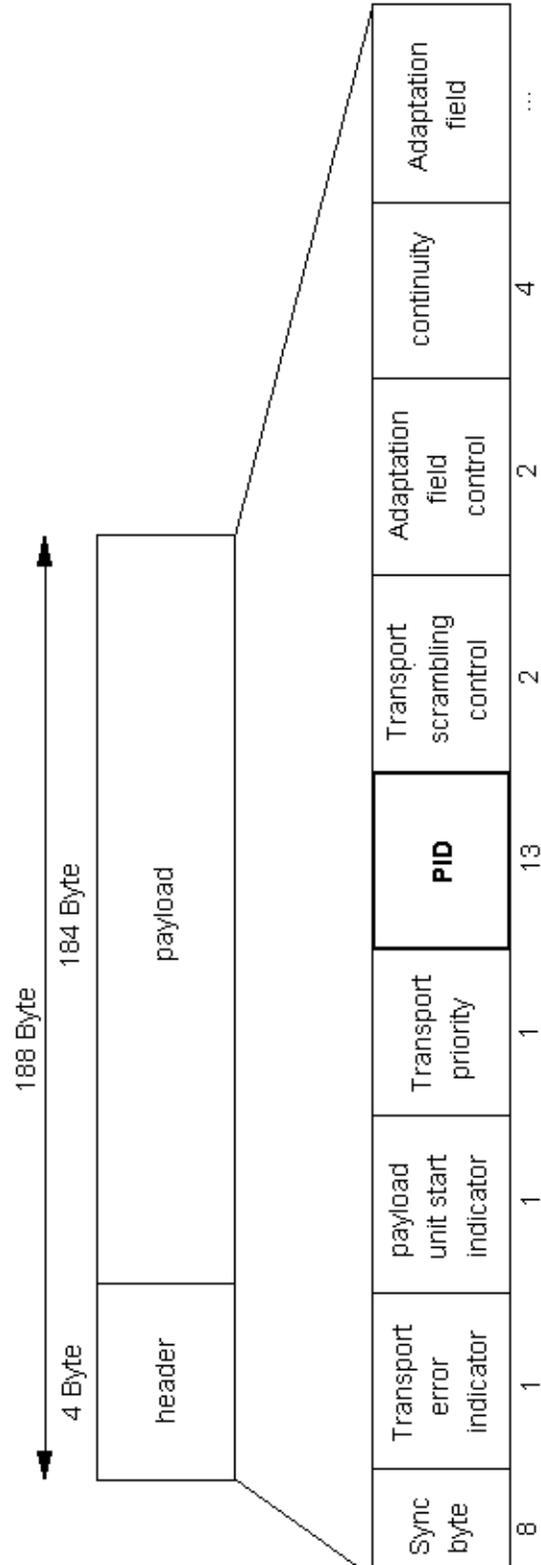
- Mehrere verschiedene Zeitbasen sind möglich.
- Sind angepaßt an einen „relativ“ fehleranfälligen Übertragungskanal. (Satellitenübertragung)
- Pakete haben eine feste Länge von 188 Byte.

Da im DVB-Stream der TS zum Einsatz kommt wird dem PS, in dieser Arbeit, keine weitere Aufmerksamkeit geschenkt und im anschließenden Abschnitt der TS näher betrachtet.

### 2.2.1 Transport-Stream (TS)

Wie in Abb. 2 zu sehen, besteht ein Transport-Stream Paket aus einem Kopf- und einem Datenteil. Das 13-Bit große PID-Feld(13) innerhalb des Kopfes gibt an welche Art von Daten gesendet werden, siehe dazu Tabelle 1.

Die jeweiligen Tabellen sind ebenfalls im DVB-SI Standard [2] beschrieben.



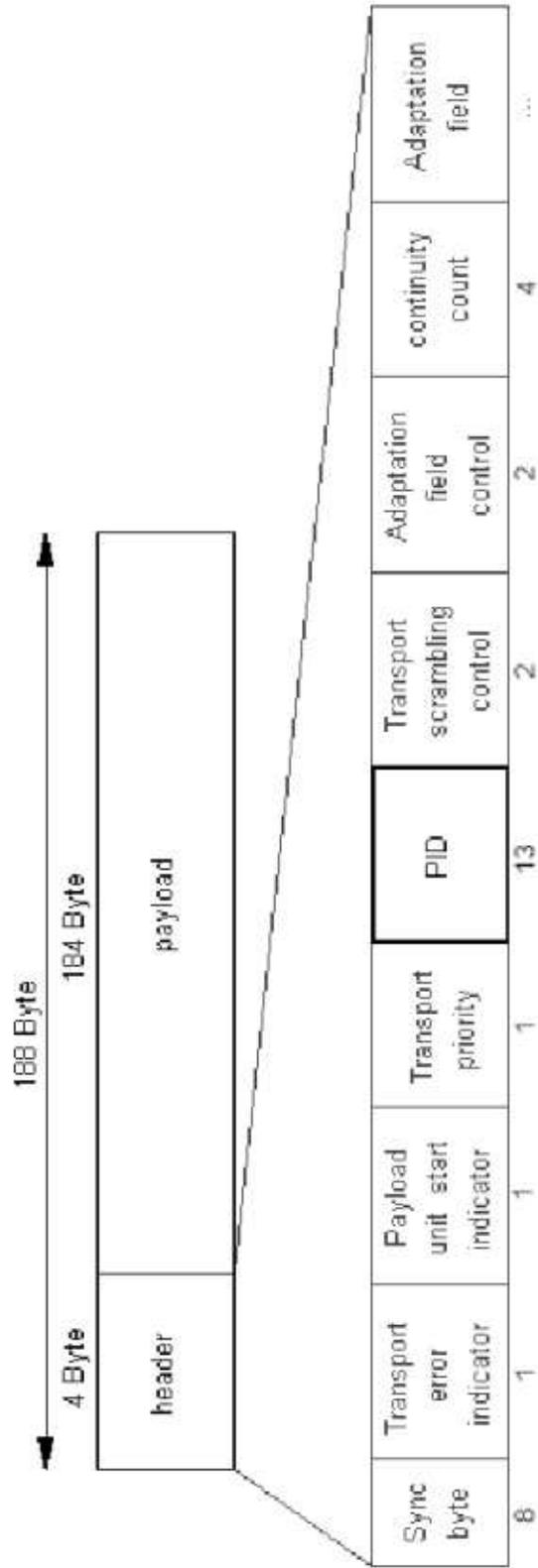


Abbildung 2: Transport-Stream Syntax Diagramm

PID	Table
0x0	PAT(11)
0x1	CAT(2)
0x2-0xF	reserved
0x10	NIT(10), ST(21)
0x11	SDT(18), BAT(1), ST(21)
0x12	EIT(5), ST(21)
0x13	RST(17), ST(21)
0x14	TDT(22), TOT(23), ST(21)
0x15	weiter PIDS unter anderem
...	Audio- und Video-PIDs
0x1FFE	sowie PMT- und AC3-PIDs
0x1FFF	Null Paket

Tabelle 1: PID-Werte

### 3 Service Informationen

Die Daten innerhalb eines SI(20)-PIDs werden in Tabellenform gesendet. Eine solche Tabelle beschreibt den Aufbau der gesendeten Information. Der MPEG-2 Standard definiert vier Tabellen, die PSI(16) enthalten. Darüber hinaus ist die Möglichkeit vorgesehen eigene Tabellen zu definieren. Der DVB-Standard hat von dieser Möglichkeit Gebrauch gemacht und eigene Tabellen definieren, um zum einen eine automatische Abstimmung des Empfangsgerätes vorzunehmen und zum anderen eine komfortable Benutzerführung zu ermöglichen. Diese durch den DVB-Standard zusätzlich definierten Tabellen sind im DVB-SI Standard, siehe 2.1.4, zusammen gefasst und umfasst fünf weitere Tabellen. Von den insgesamt neun Tabellen für „Service Informationen“ sind vier Tabellen im MPEG-2 Standard definiert.

- Program Association Table (PAT)
  - Enthält eine Liste aller Programme im Transport Multiplex und verweist auf die PIDs der zugehörigen „Program Map Table (PMT)“.
- Program Map Table (PMT)
  - Verweist auf die einzelnen PIDs des betreffenden Programms.
  - Enthält Verschlüsselungs-Informationen.

- Conditional Access Table (CAT)
  - Enthält „Private Data“ für Conditional Access.
- Network Information Table (NIT)
  - Enthält „Private Data“, z. B. Orbitalposition, Transpondernummer

Alle diese Tabellen beinhalten Angaben, die zur technischen Realisierung der Datenübertragung benötigt werden. So ergänzt die PMT die PAT mit für die Dekodierung benötigten Informationen der einzelnen Programme. Die NIT enthält von den Netzbetreibern bereitgestellte Informationen für das Abstimmen des Empfängers. Die CAT liefert Informationen, die zur Entschlüsselung von Programmen oder Programmteilen benötigt werden.

Die fünf Tabellen die im DVB-SI Standard definiert sind:

- Bouquet Association Table (BAT)
  - Enthält Informationen über das Bukett der Programme eines Anbieters, auch wenn diese über verschiedene Ausstrahlungswege verbreitet werden.
- Service Description Table (SDT)
  - Enthält die angebotenen Programme.
  - Enthält z. B. Hinweise auf die Sendeanstalten.
- Event Information Table (EIT)
  - Enthält Programmtafeln ähnlich wie in Programmzeitschriften.
  - Enthält die Kennung der jeweiligen Programmart.
  - Enthält eine Klassifizierung der Programme mit Bezug auf die Eignung für Zuschauer bestimmter Altersgruppen.
- Time and Date Table (TDT)
  - Enthält die augenblickliche Uhrzeit.
- Running Status Table (RST)
  - Gibt an, ob eine bestimmte Sendung gerade läuft, noch nicht läuft oder in unmittelbarer Zukunft beginnen wird und ermöglicht so unter anderem die Steuerung von Videorecordern.

Diese Service Informationen sind ein wichtiger Teil der gesendeten Daten, da sie dem Benutzer die Möglichkeit geben, sich in einfacher Form durch das gegenüber den Zeiten der analogen Ausstrahlung wesentlich vergrößerte Programmangebot hindurchzufinden. Des Weiteren ist es möglich, den herkömmlichen Teletext auch im DVB anzubieten, siehe 2.1.4. Es können separate Untertitel ausgestrahlt werden, die dann im Empfänger in das Bild eingeblendet werden können, siehe 2.1.4. Die Benutzerführung ist wesentlich umfangreicher und weitergehender als man dies bisher vom Teletext in der analogen Ausstrahlung gewöhnt war, was auch erforderlich ist, da die Programmanzahl um ein Vielfaches höher ist, als beim analogen TV. So könnten beispielsweise die Sehgewohnheiten durch das Empfangsgerät in Form der bevorzugten Programmarten (Sport, Krimi, ...) erfasst und beim abendlichen Einschalten auf die gerade laufenden oder auf die in naher Zukunft startenden Sendungen hinweisen und gegebenenfalls diese aufzeichnen, ohne vorher auf komplizierte Art und Weise den Recorder zu programmieren.

Der folgende Abschnitt beschäftigt sich mit der EIT(5), da diese die EPG-Daten enthält.

## 4 EPG-Daten

Der Electronic-Program-Guide, EPG(6), ist ein einfach zu bedienendes Navigationssystem und Informationssystem, das eine komplette Übersicht über das gesamte Programm eines Anbieters liefert, die Suche nach bestimmten Sendungen erleichtert und die Programmierung zur Aufzeichnung einer Sendung vereinfacht.

### 4.1 EIT

Die Event-Information-Table, EIT(5), beinhaltet in chronologischer Reihenfolge die für ein Programm gesendeten Event Informationen, Tabelle 3 zeigt den Aufbau der EIT. Dabei unterscheidet der MPEG2-Standard vier Tabellen-IDs, die innerhalb der EIT auftreten können, Tabelle 2 gibt einen Überblick. Digitale Empfänger zeigen bei jedem Umschalten auf einen neuen Sender das laufende Programm, die Startzeit, die Programmdauer und eine Beschreibung der Sendung an. Für diese Informationen wird in der EIT die Tabelle mit der ID=0x4E ausgelesen. Aus folgenden Gründen werden die Event Informationen auf mehrere Tabellen aufgeteilt. Für Informationen, die schnell zur Verfügung stehen müssen wurden die Tabellen mit der ID=0x4E und ID=0x4F eingeführt, die ausschließlich Event Informationen für die aktuelle und chronologisch folgende Sendung bereitstellen. Diese Tabellen innerhalb der EIT enthalten wenig Informationen, die häufig gesendet werden. Dabei bezieht sich die Tabelle mit der ID=0x4E auf den aktuellen TS(24) und die Tabelle mit der ID=0x4F auf irgend einen anderen TS. Da es mit Hilfe der digitalen Technik möglich ist innerhalb der Bandbreite eines analogen Kanals mehrere digitale Kanäle unterzubringen und diese zu einem Datenstrom gemultiplext werden, Abschnitt 2.2, werden für alle diese einzelnen Sender innerhalb eines Kanals, diese Zusatzinformationen gesendet. Es gibt aber Anbieter die in ihrer Anzahl an Sendern mehr anbieten als an Bandbreite pro analogem Kanal zur Verfügung steht, diese müssen dann auf einen anderen Kanal ausweichen. Jetzt wäre es unmöglich von diesen Sendern die Event Informationen zu erhalten, da sie auf einem anderen Kanal liegen. Um dies zu vermeiden ist der Empfang von Event Informationen anderer Kanäle ebenfalls möglich, indem diese Event Informationen innerhalb einer weiteren Tabelle gesendet werden, die Tabelle mit der ID=0x4F. Nun ist es möglich eine Übersicht der laufenden und folgenden Sendungen der Sender eines Anbieters zu erstellen. Die restlichen Tabellen, die innerhalb der EIT gesendet werden, beinhalten Event Informationen für den gesamten Tag bzw. mehrere Tage der einzelnen Sender. Innerhalb dieser Tabellen werden sehr viel mehr Informationen gesendet, die sich nicht so häufig wiederholen, wie die in den Tabellen für die laufende und folgende Sendung. Mit Hilfe dieser Informationen ist es möglich sich eine elektronische Programmzeitung aufzubauen. Es hängt natürlich von den jeweiligen Anbietern ab, welche Informationen gesendet

werden und in welchem Umfang. Siehe Abbildung ??, die den Datenstrom auf dem EIT-PID darstellt.

TableID	Inhalt	TS
0x4E	aktuelle/anschließende Event Informationen	aktueller
0x4F	aktuelle/anschließende Event Informationen	anderer
0x50 - 0x5F	Event Zeitplan (Vorschau)	aktueller
0x60 - 0x6F	Event Zeitplan (Vorschau)	anderer

Tabelle 2: EIT Tabellen IDs

## 4.2 EIT-Deskriptoren

Innerhalb der EIT werden Deskriptoren gesendet. Diese Deskriptoren sind Untertabellen und werden anhand eines Descriptor-TAGs identifiziert. Deskriptoren treten immer nur innerhalb einer eigenen Tabelle auf, d. h. ein Descriptor wird nicht direkt gesendet sondern immer als Untertabelle. Laut dem DVB-SI Standard [2] können die in Tabelle 4 aufgelisteten Deskriptoren innerhalb der EIT gesendet werden. Alle diese Deskriptoren werden in ihrem Aufbau und ihrer Bedeutung im DVB-SI Standard beschrieben.

Abbildung 3 zeigt den Zusammenhang zwischen den EIT-Tabellen und den Deskriptoren.

## 4.3 Datenübertragung

Es wurde bereits erwähnt wie die Daten innerhalb des EIT-PIDs gesendet werden. Da es sich bei einem EIT-Datenstrom um einen gemultiplexten Datenstrom von mehreren Sendern eines Anbieters handelt, müssen die Tabellen, die innerhalb der EIT gesendet werden einem Sender zugeordnet werden können. Dazu dient der Eintrag `service_id` innerhalb des EIT-Tabellenkopfes, siehe Tabelle 3. Die `service_id` ist eine eindeutige Nummer, d. h. sie kommt nur einmal pro Satellit oder nur einmal pro Kabelnetzbetreiber vor und repräsentiert einen Sender. Die Wiederholungen von Event Informationen innerhalb einer Event Information Table, variieren von einer Wiederholung bis zu n Wiederholungen. Das heißt, dass sich eine Tagesübersicht aus mehreren Tabellen und den entsprechend Untertabellen zusammensetzen kann. Aus diesem Grund werden innerhalb des Kopfes jeder Tabelle, siehe Tabelle 3, entsprechende Informationen gesendet, die es ermöglichen die Anzahl der Event Information Tabellen zu spezifizieren, die zu einem

Syntax	Bits
table_id	8
section_syntax_indicator	1
reserved_future_use	1
reserved	2
section_length	12
service_id	16
reserved	2
version_number	5
current_next_indicator	1
section_number	8
last_section_number	8
transport_stream_id	16
original_network_id	16
segment_last_section_number	8
last_table_id	8
for(i=0;i<N;i++){	
event_id	16
start_time	40
duration	24
running_status	3
free_CA_mode	1
descriptors_loop_length	12
for(j=0;j<M;j++){	
descriptor()	
}	
}	
CRC_32	32

Tabelle 3: Event Information Table

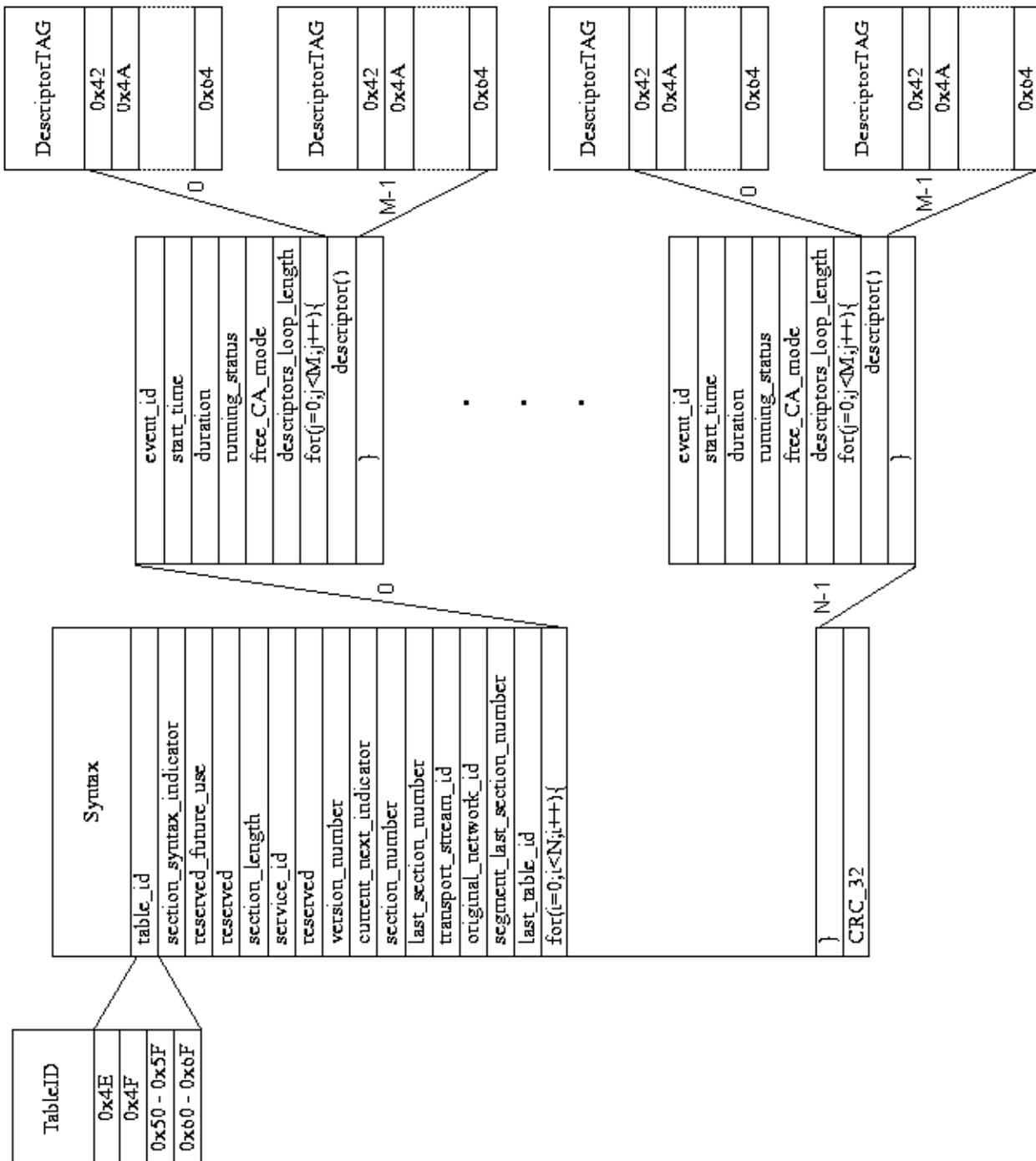


Abbildung 3: EIT-Aufbau

DescriptorTAG	Descriptor
0x42	stuffing_descriptor
0x4A	linkage_descriptor
0x4D	short_event_descriptor
0x4E	extended_event_descriptor
0x4F	time_shifted_event_descriptor
0x50	component_descriptor
0x53	CA_identifier_descriptor
0x54	content_descriptor
0x55	parental_rating_descriptor
0x57	telephone_descriptor
0x5E	multilingual_component_descriptor
0x5F	private_data_specifier_descriptor
0x60	PDC_descriptor
0x61	short_smoothing_buffer_descriptor
0x64	data_broadcast_descriptor

Tabelle 4: EIT-Deskriptoren

Sender gehören. So wird durch die `service_id`, wie bereits erwähnt, innerhalb der EIT der Sender identifiziert, ob RTL oder RTL2, durch die `section_number` der Tabellenindex spezifiziert und durch die `last_section_number` der letzte Tabellenindex spezifiziert. Mit Hilfe dieser Informationen ist es möglich alle Tabellen, die innerhalb der EIT-PIDs gesendet werden einem Sender zuzuordnen und so auf Vollständigkeit zu prüfen.

Die Bandbreite, die für die EPG-Daten zur Verfügung steht kann von der jeweiligen Sendergruppe festgelegt werden. Dadurch ist die Geschwindigkeit mit der alle EPG-Daten eingelesen werden können abhängig vom Anbieter. So wurden im nachfolgenden Beispiel die fünf großen Sendergruppen Deutschlands auf die Geschwindigkeit der EPG-Daten hin untersucht, siehe Tabelle 5. Es wurden dazu 2000 Tabellen eingelesen und gemessen wieviel Byte/sec gelesen wurden, was den Umfang der EPG-Daten repräsentiert und wieviele Tabellen/sec gelesen wurden, was die Geschwindigkeit repräsentiert.

Sendergruppe	[bytes/sec]	[table/sec]
ARD-Sendergruppe	18510	30
ZDF-Sendergruppe	19819	18
RTL-Group	4177	10
Sat1ProSieben	39125	200
Premiere	63362	200

Tabelle 5: EIT-Durchsatz der Sendergruppen

#### 4.4 Programmzeitung

Anhand der bereitgestellten Informationen innerhalb der EIT ist es möglich eine elektronische Programmzeitung zu erstellen, die die folgenden Informationen beinhaltet:

- Eventname mit kurzer und ausführlicher Beschreibung des Events
- eine Beschreibung der Streams in mehreren Sprachen
  - Bildformat: z. B. 4:3 Aspekt Ratio mit 25 Hz
  - Tonformat: Stereo oder Surround Sound (AC3)
- ob und mit welchem Verfahren ein solches Event verschlüsselt ist
- Klassifikation des Events, ob Spielfilm, Sportevent, Kinderprogramm usw.
- Altersbeschränkung für das entsprechende Event, ab 12, 16, 18 Jahren
- Länderkennung für das entsprechende Event
- Kennzeichnung einer Interaktiven Sendung
- Bit-Raten Angaben zu einem Event

Abgesehen von zwei Informationen würden diese Daten reichen, um mit einer herkömmlichen Programmzeitschrift zu konkurrieren. Die fehlenden Informationen wären zum einen Bilddaten und zum anderen eine Wertung des Events. Das Problem mit den Bilddaten läßt sich lösen in dem entweder ein Drittanbieter als Informationsquelle genutzt wird, z. B. Internet Movie Database oder man sich eine eigene Bilderdatenbank aufbaut, beispielsweise durch anfertigen eines Snapshot während ein Event läuft, was den Vorteil einer gewissen Unabhängigkeit gegenüber einem Drittanbieter hat, aber den Nachteil von Ressourcenverbrauch bringt,

wie z. B. Speicherplatz oder Administrationsbedarf der Bilddaten. Für die Wertung eines Events macht es keinen Sinn eine eigene Datenbank zu pflegen, da es zu aufwändig wäre entsprechend viele Stimmen einzuholen um ein repräsentatives Ergebnis zu erhalten, somit bleibt nur noch sich diese Informationen von einem Drittanbieter zu beschaffen.

## 4.5 Probleme

Eigentlich ist dies nur eine theoretische Möglichkeit eine elektronische Programmzeitung zu erstellen, da die Daten, die innerhalb der EIT gesendet werden können nicht gesendet werden müssen. Das hat zur Folge, dass jedem Anbieter selber überlassen wird, was an zusätzlichen Daten gesendet wird. Somit ist es nicht möglich eine konsistente elektronische Programmzeitung zu erstellen.

Ein weiteres Problem besteht darin, dass es nicht möglich ist von einem Sender aus eine vollständige elektronische Programmzeitung aufzubauen, d. h. wenn man sich auf einem Sendepfad der ARD befindet und man gerade die EPG-Daten einliest ist es unmöglich die EPG-Daten der RTL-Group ebenfalls einzulesen, was dazu führt, dass auf einen Sender der RTL-Group geschaltet werden muss, um die EPG-Daten der RTL-Group einlesen zu können. Dieses Problem hat keinen Einfluss auf die EPG-Daten selbst aber auf die Geschwindigkeit mit der die elektronischen Programmzeitung erfasst werden kann.

Ein weiteres Problem besteht darin, dass die gesendeten Beschreibungsdaten zu einem Event nicht nochmal separat unterteilt sind in Beschreibungsdaten, beteiligte Schauspieler, den Regisseur oder das Erscheinungsjahr. Stattdessen werden sie mit den Beschreibungsdaten zusammen gesendet. Daher ist es nötig diese Informationen von den Beschreibungsdaten zu trennen, um sie später weiter nutzen zu können. Beispielsweise könnten diese Informationen benutzt werden, die bereits bestehenden Informationen mit weiteren Informationen anzureichern, z. B. mit Informationen aus der Internet Movie Database. Eine eigene Lösung zur Erfassung dieser Informationen ist mit relative hohem Aufwand verbunden, aber zwingend erforderlich um möglichst viele Informationen zu einem Event zusammenstellen zu können.

Ein Problem welches auf Grund des Aufwandes noch nicht gelöst werden konnte ist die „Intelligenz“ der Datenerfassung. Momentan ist die Datenerfassung noch statisch, d. h. es werden 2000 Tabellen eingelesen und dann das Ergebnis präsentiert. Wie Abschnitt 4.3 beschreibt, ist es jedem Sender überlassen wieviel Tage im voraus er EPG-Daten sendet, was dazu führen kann, dass 2000 Tabellen nicht reichen, d. h. sie nur die Hälfte der EPG-Daten repräsentieren oder zu viel

Daten gelesen werden, d. h. dass die EPG-Daten doppelt gelesen werden und so unnötig Zeit verbraucht wird. Somit sind die erfassten Daten nicht zwangsläufig ein Abbild der gesendeten Daten. Aber die 2000 Tabellen repräsentieren einen guten Mittelwert, d. h. es werden mindestens der heutige und der folgende Tag vollständig repräsentiert.

Dieses Problem kann gelöst werden, wenn weitere SI-Daten ausgewertet werden, die Informationen darüber enthalten, welche Sender auf diesem Kanal senden und wieviel EPG-Tabellen für jeden dieser Sender ausgestrahlt werden, siehe Abschnitt 4.3.

## 5 Implementierung

Bevor die Implementierung beschrieben wird, sollte die Syntax erläutert werden zum, besseren Lesen der Quellen.

### 5.1 Syntax - Beschreibung

Bei den Quellen handelt es sich um C++-Quellen. Namen werden in Unterworte aufgeteilt, die jeweils mit einem Großbuchstaben beginnen und dann mit Kleinbuchstaben fortfahren, z. B. class EpgXml. Diese Syntax gilt für Klassennamen, für Methoden und Attribute. Zusätzlich gilt bei Methoden und den Attributen einer Klasse, dass die Namen immer klein beginnen, z. B. dumpXml(), int serviceId. Übergabeparameter haben als Präfix einen Unterstrich, z. B. \_serviceId. Jede Klasse wird in einer separaten Header-Datei definiert und in einer separaten C++-Datei deklariert, z. B. epgXml.h und epgXml.cpp. Ausnahme bilden gekapselte Klassen, sie werden in der Deklarationsdatei der Kapselklasse deklariert, betrifft die alle List Klassen. Die Dateinamen lauten wie die entsprechende Klasse aber beginnen mit einem kleinen Buchstaben.

### 5.2 Aufbau

Die Implementierung teilt sich in zwei Aufgaben auf, in die Datenerhebung und die Datenspeicherung. Die Abbildung 4 zeigt den Aufbau der Implementierung.

#### 5.2.1 Datenerhebung

Die Datenerhebung wird durch die beiden Module eit.o und dvbSiStream.o implementiert. Die Klasse DvbSiStream kapselt den Zugriff auf die Hardware und den Zugriff auf die libdvbsi (Abschnitt 5.2.2). Das eit.o Modul füllt das Objekt, welches die erfassten Daten von der Datenerhebung zu Speicherung transportiert. Dieses Objekt ist eine Instanz der Klasse EpgEventContents. In dem eit.o Modul wird beschrieben welche Daten erfasst werden sollen und wo sie gespeichert werden sollen.

#### 5.2.2 libdvbsi

Die Bibliothek libdvbsi erlaubt es komfortabel auf den SI-Datenstrom, der von der DVB-Karte kommt, zuzugreifen. Dabei zerlegt die Bibliothek den SI-Stream in seine Tabellen und Descriptoren und speichert diejenigen Tabellen- und Descriptorenwerte die benötigt werden in einer Struktur ab. Auf diese Struktur können andere Programmteile zugreifen und bekommen so die erforderlichen Daten. Mit

Hilfe dieser Bibliothek wird der Zugriff auf den SI-Stream abstrahiert und somit die Arbeit erleichtert.

### 5.2.3 Datenspeicherung

Die Datenspeicherung teilt sich in Adressdaten und Inhaltsdaten auf. Die Adressdaten werden durch die Epg-Klassen beschrieben, die Inhaltsdaten werden durch die Event-Klassen beschrieben. Wie Abbildung 4 zeigt, besteht die Adresse aus drei Teilen: dem Startdatum, der Programm-Id und der Startzeit. Implementiert wird diese Adressierung durch drei Klassen. Die Klasse EpgSchedule, legt eine Map an, eine Hashtabelle, welche als Schlüsselwert das Startdatum benutzt und als Wert einen Zeiger auf ein Objekt der Klasse EpgScheduleDay besitzt. Die zweite Stufe der Adressierung wird in der Klasse EpgScheduleDay implementiert. Innerhalb dieser Klasse wird wiederum eine Map erzeugt, welche als Schlüsselwert die Programm-Id, im Quellcode Service-Id, benutzt und als Wert einen Zeiger auf ein Objekt der Klasse EpgEventMap besitzt, welches die dritte Stufe der Adressierung darstellt. Innerhalb dieser Klasse wird wiederum eine Map erzeugt, dass als Schlüsselwert die Startzeit benutzt und als Wert einen Zeiger auf ein Objekt der Klasse EpgEvent besitzt. Somit ist die Adressierung abgeschlossen und es kann direkt auf die Inhaltsdaten zugegriffen werden.

Die Inhaltsdaten werden durch Instanzen der Klasse EpgEvent repräsentiert. Alle Event-Klassen repräsentieren Inhaltsdaten innerhalb der Klasse EpgEvent. Die Inhaltsdaten, die eigentlichen Epg-Daten, setzen sich aus der Beschreibung des Events, aus der Bestimmung der Kategorie, den Komponenten des Events, der Altersbeschränkung zusammen und ein paar weiteren Informationen wie die Event-Identifikation, die Event-Dauer und ob das Event verschlüsselt ist oder nicht.

Wie bereits erwähnt, dient die Klasse EpgEventContents als Datenaustauschcontainer zwischen Datenerhebung und Datenspeicherung. Das heißt ein Objekt dieser Klasse wird erzeugt und mit Daten gefüllt. Dann durch die einzelnen Adressierungstufen gereicht, wobei sich jedes Adressierungsobjekt die Daten aus dem Container entnimmt, die es benötigt und es dann weiter reicht. Dies wird innerhalb der Inhaltsobjekte so beibehalten bis alle erhobenen Daten gespeichert sind. Der resultierende Baum sieht dann wie in Abbildung 4 dargestellt aus.

## 5.3 Design

Das im Abschnitt 5.2 beschriebene Design ist relativ simpel und hat bisher keinerlei Optimierung erfahren. Das Ziel der ersten Implementierung war es alle Epg-

Daten, die gesendet werden zu erfassen, sie zu sortieren und anderen Applikationen zur Verfügung zu stellen. Aus diesem Grund werden alle Informationen, die gesendet werden erfasst und gespeichert. Somit sind auf den ersten Blick nicht alle erfassten Informationen relevant für eine elektronische Programmzeitung, beispielsweise die Bitrate des Videostreams. Es wären denkbar Optimierungen auf Speicherplatz bzw. auf Zugriffszeit bei Suchanfragen durchzuführen.

## 5.4 Austausch- und Speicherformat

Da die Epg-Daten anderen Applikationen zur Verfügung gestellt werden sollen, z. B. einem Palm-Organizier, ist es erforderlich ein Austausch- und Speicherformat zu spezifizieren. An dieses Format sind einige Anforderungen gestellt. Es soll selbsterklärend sein, die Daten sollen manipulierbar sein, z. B. um sie mit weiteren Informationen zu versehen, die Daten sollen mit Standardwerkzeugen betrachtet werden können und sie sollen in andere Formate transformiert werden können. All diese Eigenschaften erfüllt XML(25). XML wurde neben anderen Anwendungsgebieten auch als Datenaustauschformat eingesetzt, beispielsweise im Gnome Desktop Projekt. XML ist eine Markup Language, die von SGML(19) abstammt und sehr flexibel ist. Es ist möglich sich eine eigene Sprache zu definieren und sie mit Hilfe von Stylesheets in eine andere Sprache zu überführen, z. B. nach HTML(8). Somit ist es möglich, die erfassten Epg-Daten auszudrucken, sie auf einem Web-Browser, auf einem Palm oder auf dem Bildschirm des Fernsehers zu visualisieren.

Neben der Bereitstellung der Epg-Daten für andere Applikationen wären die gespeicherten Daten auch für den Epg-Scheduler selbst sinnvoll. Die Daten können beim erneuten Start des Epg-Schedulers eingelesen werden, um den Zustand der vor der letzten Beendigung bestand wieder herzustellen. Dadurch werden hohe Latenzen verhindert, die entstehen, wenn die Epg-Daten wieder komplett neu eingelesen werden müssen.

Die in der XML-Datei erfassten Daten sind Rohdaten, d. h. sie sind unbearbeitet. Zum Beispiel wird das Datum im MJD(9) Format angegeben, was unlesbar für den Mensch ist und daher in ein lesbares Format konvertiert werden sollte. Eine solche Konvertierung muss immer im Ausgabegerät durchgeführt werden, da nur dieses die Zeitzone und das entsprechende Ausgabeformat, M:T:J oder T.M.J usw. kennt.

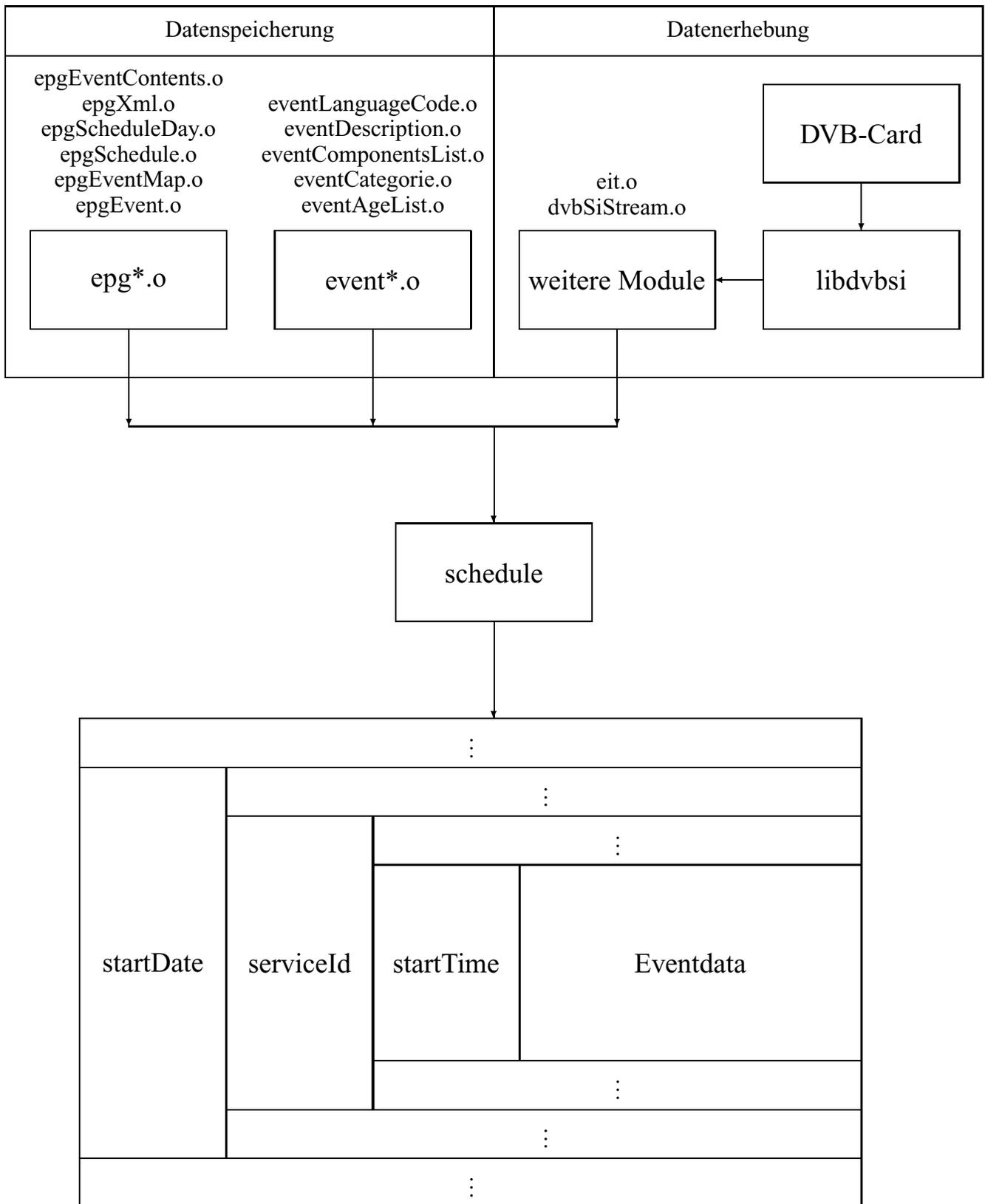


Abbildung 4: Epg-Scheduleraufbau

```

<?xml version = "1.0" encoding="ISO-8859-15" ?>
<EPG>XMLDUMP>
  <DAY date = "0x0e0b">
    <PROGRAM serviceId = "0x8">
      <TIME startTime = "0x181000">
        <EVENT id = "0x4bd9" duration = "0x20000" freeCatMode = "0x1">
          </EVENT>
          <DESCRIPTION name = "13TH STREET Ein Köder für die Bestie"
          </DESCRIPTION>
          <CATEGORIE nibble1 = "0x1" nibble2 = "0x1">
            </CATEGORIE>
          <COMPONENTS streamContent = "0x1" componentType = "0x1"
          </COMPONENTS>
          <AGE value = "0xd">
            <LANGUAGECODE ICode = "DEU">
              </LANGUAGECODE>
            </AGE>
          <AGE value = "0xd">
            <LANGUAGECODE ICode = "GBR">
              </LANGUAGECODE>
            </AGE>
          <AGE value = "0xd">
            <LANGUAGECODE ICode = "ITA">
              </LANGUAGECODE>
            </AGE>
          </TIME>
        </PROGRAM>
      <PROGRAM serviceId = "0x9">
        <TIME startTime = "0x31500">
          <EVENT id = "0xae0e" duration = "0x13000" freeCatMode = "0x1">
            </EVENT>
            <DESCRIPTION name = "Jeepers Creepers" text = "Horrorfilm"
            </DESCRIPTION>
            <CATEGORIE nibble1 = "0x1" nibble2 = "0x3">
              </CATEGORIE>
            <COMPONENTS streamContent = "0x1" componentType = "0x1"
            </COMPONENTS>
            <COMPONENTS streamContent = "0x2" componentType = "0x3"
            </COMPONENTS>
          </TIME>
        </PROGRAM>
      </DAY>
    <DAY date = "0x0e0c">
      <PROGRAM serviceId = "0x8">
        <TIME startTime = "0x0">
          <EVENT id = "0x4be0" duration = "0x4500" freeCatMode = "0x1">
            </EVENT>
            <DESCRIPTION name = "PREMIERE SERIE MacGyver"
            </DESCRIPTION>
            <CATEGORIE nibble1 = "0x1" nibble2 = "0x5">
              </CATEGORIE>
            <COMPONENTS streamContent = "0x1" componentType = "0x1"
            </COMPONENTS>
          </TIME>
        </PROGRAM>
      </DAY>}
    </EPG>XMLDUMP>

```

Abbildung 5: Aufbau der XML-Datei

## 6 Fazit

Die Aufgabe bestand darin EPG-Daten zu erfassen, sie aufzubereiten und anderen Applikationen zur Verfügung zu stellen. Es wurde geklärt wie die EPG-Daten zu beschaffen sind, wie sie aufbereitet werden können, wie sie gespeichert werden können, wie sie anderen Applikationen zur Verfügung gestellt werden können, welche Probleme entstanden sind und wie sie gelöst werden können. Dies wurde nicht nur theoretisch sondern auch praktisch umgesetzt. Dazu wurde ein Programm implementiert, welches die EPG-Daten erfasst, sie aufbereitet, sortiert und anschließend serialisiert, d. h. speichert. Dabei wurde auf hohe Abstraktion geachtet, um unabhängig zu sein, z. B. von der Plattform. Natürlich muss die Datenerhebung dabei auf das entsprechende System angepaßt werden, in diesem Fall Linux. Es sind noch einige Probleme offen, die angesprochen und für die auch Lösungsvorschläge gemacht wurden. Der momentane Implementierungsstand funktioniert und kann eingesetzt werden. Mit Hilfe des beiliegenden Stylesheets kann die entstandene XML-Datei, welche die aufbereiteten EPG-Daten enthält, in eine HTML-Datei konvertiert werden und in jedem herkömmlichen Web-Browser betrachtet werden.

## 7 Abkürzungen

- (1) BAT *BouquetAssociationTable*
- (2) CAT *ConditionalAccessTable*
- (3) DVB *DigitalVideoBroadcasting*
- (4) DVR *DigitalerVideoRecorder*
- (5) EIT *EventInformationTable*
- (6) EPG *ElectronicProgramGuide*
- (7) ETSI *EuropeanTelecommunicationsStandardsInstitute*
- (8) HTML *HyperTextMarkupLanguage*
- (9) MJD *ModifiedJulianDate*
- (10) NIT *NetworkInformationTable*
- (11) PAT *ProgramAssociationTable*
- (12) PES *PacketisedElementaryStream*
- (13) PID *PacketIDentifier*
- (14) PMT *ProgramMapTable*
- (15) PS *ProgramStream*
- (16) PSI *ProgramSpecificInformation*
- (17) RST *RunningStatusTable*
- (18) SDT *ServiceDescriptionTable*
- (19) SGML *StructuredGeneralizedMarkupLanguage*
- (20) SI *ServiceInformation*
- (21) ST *StuffingTable*
- (22) TDT *TimeandDateTable*
- (23) TOT *TimeOffsetTable*
- (24) TS *TransportStream*
- (25) XML *ExtensibleMarkupLanguage*

## Literatur

- [1] ETSI. Guidelines on implementation and usage of Service Information (SI). *ETSI ETR 211*, August 1997.
- [2] ETSI. Specification for Service Information (SI) in DVB systems. *ETSI EN 300 468*, January 2003.
- [3] ISO/IEC. MPEG-2 generic coding of moving pictures and associated audio information. *ISO/IEC 13818-1*, January 1994.
- [4] Bjarne Stroustrup. *Die C++ Programmiersprache*. Addison-Wesley, Reading, 3rd edition, 1997.