

Festlegung der einheitlichen Schnittstelle für
Zugriffsbeschränkung und andere digitale
Fernsehrundfunkdecoder-Anwendungen
Deutsche Fassung EN 50221:1997 + Corrigendum:2000

DIN
EN 50221

ICS 33.160.25

Ersatz für
DIN EN 50221:1997-08

Common interface specification for conditional access and other digital
video broadcasting decoder applications;

German version EN 50221:1997 + Corrigendum:2000

Spécification d'une interface commune pour l'accès conditionnel et
d'autres applications dans un décodeur de télévision numérique;

Version allemande EN 50221:1997 + Corrigendum:2000

**Die Europäische Norm EN 50221:1997, zusammen mit dem eingearbeiteten
Corrigendum:2000, hat den Status einer Deutschen Norm.**

Beginn der Gültigkeit

Die EN 50221 wurde am 1997-02-15 angenommen.

Das Corrigendum wurde im Februar 2000 herausgegeben.

Nationales Vorwort

Für die vorliegende Norm ist das nationale Arbeitsgremium K 742 „Audio-, Video- und Multimediasysteme, -geräte und -komponenten“ der Deutschen Elektrotechnischen Kommission im DIN und VDE (DKE) zuständig.

Norm-Inhalt war veröffentlicht als E DIN EN 50221/A1:2000-06.

Änderungen

Gegenüber der DIN EN 50221:1997-08 wurden folgende Änderungen vorgenommen:

Die Abschnitte der eingearbeiteten Änderung korrigieren einen Fehler in der Information über das PC-Karten-Metaformat und passen den Text von DIN EN 50221:1997 so an, dass er mit der allgemeinen Industriepraxis der Verwendung von Interrupts für Modul-I/O übereinstimmt.

Frühere Ausgaben

DIN EN 50221: 1997-08

Das Corrigendum:2000 zu EN 50221:1997 wurde in den Text eingearbeitet und durch eine senkrechte Linie am linken Seitenrand im Text gekennzeichnet.

Fortsetzung Seite 2
und 79 Seiten EN

Deutsche Elektrotechnische Kommission im DIN und VDE (DKE)

Nationaler Anhang NA (informativ)

Zusammenhang mit Europäischen und Internationalen Normen

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe der Norm.

Der Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ist nachstehend wiedergegeben. Zum Zeitpunkt der Veröffentlichung dieser Norm waren die angegebenen Ausgaben gültig.

IEC hat 1997 die Benummerung der IEC-Publikationen geändert. Zu den bisher verwendeten Normnummern wird jeweils 60000 addiert. So ist zum Beispiel aus IEC 68 nun IEC 60068 geworden.

Tabelle NA.1

Europäische Norm	Internationale Norm	Deutsche Norm
EN ISO/IEC 13818-1:1997	ISO/IEC 13818-1:1996	DIN EN ISO/IEC 13818-1:1997-06
–	ISO/IEC 13818-9:1996	–
–	Normen der Reihe ISO/IEC 8824	–
–	ISO/IEC 8825:1990	DIN 66333:1993-09
–	ETS 300468:1995 zurückgezogen Ersatz EN 300468:1998	DIN ETS 300468:1996-05
–	EN 300468:1998	DIN EN 300468:1998-10
–	ETR 162:1995	–
–	PC Card Standard Volume 2	–
–	PC Card Standard Volume 3	–
–	PC Card Standard Volume 4	–
–	ETS 300743:1997	DIN ETS 300743:1998-06

Nationaler Anhang NB (informativ)

Literaturhinweise

DIN 66333, *Informationstechnik – Kommunikation Offener Systeme; Basis-Codierungsregeln zur Notation Eins für darstellungsunabhängige Syntax; Identisch mit ISO/IEC 8825:1990.*

DIN EN ISO/IEC 13818-1, *Informationstechnik – Codierung von bewegten Bildern und damit verbundenen Toninformationen – Teil 1: Systeme (ISO/IEC 13818-1:1996); Englische Fassung EN ISO/IEC 13818-1:1997.*

DIN EN 300468, *Digitaler Fernsehrundfunk (DVB) – Festlegung der Serviceinformation (SI) für DVB-Systeme; Englische Fassung EN 300468 V 1.3.1 (1998-02).*

DIN ETS 300468, *Digitale Rundfunk-Systeme für Fernsehen, Ton und Datendienste – Festlegungen der Serviceinformation (SI) für digitale Fernsehrundfunk-Systeme (DVB); Englische Fassung ETS 300468:1995.*

DIN ETS 300743, *Digitaler Fernsehrundfunk (DVB) – Untertitelungssysteme; Englische Fassung ETS 300743:1997.*

Deutsche Fassung

**Festlegung der einheitlichen Schnittstelle für
Zugriffsbeschränkung und andere digitale
Fernsehrundfunkdecoder-Anwendungen**

Common interface specification for conditional
access and other digital video broadcasting de-
coder applications

Spécification d'une interface commune pour l'accès
conditionnel et d'autres applications dans un déco-
deur de télévision numérique

Diese Europäische Norm wurde von CENELEC am 1997-02-15 angenommen und das Corrigendum im Februar 2000 herausgegeben.

Die CENELEC-Mitglieder sind gehalten, die CEN/CENELEC-Geschäftsordnung zu erfüllen, in der die Bedingungen festgelegt sind, unter denen dieser Europäischen Norm ohne jede Änderung der Status einer nationalen Norm zu geben ist.

Auf dem letzten Stand befindliche Listen dieser nationalen Normen mit ihren bibliographischen Angaben sind beim Zentralsekretariat oder bei jedem CENELEC-Mitglied auf Anfrage erhältlich.

Diese Europäische Norm besteht in drei offiziellen Fassungen (Deutsch, Englisch, Französisch). Eine Fassung in einer anderen Sprache, die von einem CENELEC-Mitglied in eigener Verantwortung durch Übersetzung in seine Landessprache gemacht und dem Zentralsekretariat mitgeteilt worden ist, hat den gleichen Status wie die offiziellen Fassungen.

CENELEC-Mitglieder sind die nationalen elektrotechnischen Komitees von Belgien, Dänemark, Deutschland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Luxemburg, Niederlande, Norwegen, Österreich, Portugal, Schweden, Schweiz, Spanien, der Tschechischen Republik und dem Vereinigten Königreich.

CENELEC

EUROPÄISCHES KOMITEE FÜR ELEKTROTECHNISCHE NORMUNG
European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique

Zentralsekretariat: rue de Stassart 35, B-1050 Brüssel

Vorwort

Diese Europäische Norm wurde von dem Technischen Komitee CENELEC TC 206 „Rundfunk-Empfangseinrichtungen“ ausgearbeitet.

Der Text des Entwurfs wurde dem Einstufigen Annahmeverfahren unterworfen und von CENELEC am 1997-02-15 als EN 50221 angenommen.

Nachstehende Daten wurden festgelegt:

- spätestes Datum, zu dem die EN auf nationaler Ebene durch Veröffentlichung einer identischen nationalen Norm oder durch Anerkennung übernommen werden muss (dop): 1997-10-01
- spätestes Datum, zu dem nationale Normen, die der EN entgegenstehen, zurückgezogen werden müssen (dow): 1997-10-01

Vorwort zu Corrigendum:2000

ANMERKUNG Die Ersatzabschnitte in diesem Corrigendum korrigieren einen Fehler in der Information über das PC-Karten-Metaformat und passen den Text von EN 50221:1997 so an, dass er mit der allgemeinen Industriepraxis der Verwendung von Interrupts für Modul-I/O übereinstimmt.

Inhalt

	Seite		Seite
Vorwort.	2	8.1 Einführung	20
1 Einführung und Anwendungsbereich	3	8.2 Hilfsmittel	21
2 Normative Verweisungen	4	8.3 Anwendungsprotokoll-Dateneinheit (APDU)	22
3 Definitionen	4	8.4 Systemmanagement-Hilfsmittel	23
4 Designphilosophie	5	8.5 Hilfsmittel Hauptgerätesteuerung und Information	30
4.1 Schichtung	5	8.6 Hilfsmittel Mensch-Maschine-Schnittstelle (MMI)	32
4.2 Physikalische Implementierung	5	8.7 Kommunikationshilfsmittel	46
4.3 Client-Server	5	8.8 Hilfsmittelkennungen und Anwendungsobjekt-Tags	50
4.4 Codierung der Daten	5		
4.5 Erweiterbarkeit	5	Anhang A (normativ) Auf der PC-Karte basierende physikalische Schicht	53
4.6 Einbringen bestehender Normen	6	A.1 Allgemeine Beschreibung	53
5 Beschreibung und Architektur	6	A.2 Elektrische Schnittstelle	54
5.1 Überblick	6	A.3 Verbindungsschicht	56
5.2 Transportstromschnittstelle	6	A.4 Implementierungsspezifische Transport-Unterschicht über PC-Karten-Schnittstelle	57
5.3 Befehlsschnittstelle	6	A.5 Teilspezifikation für PC-Karten, die bei konformen Hauptgeräten und Modulen benutzt werden	64
5.4 Physikalische Anforderungen	7		
5.5 Betriebsbeispiel	9	Anhang B (informativ) Zusätzliche Objekte	71
6 Transportstromschnittstelle (TSI)	9	B.1 Authentisierung	71
6.1 TSI – physikalisch, Verbindungsschichten	9	B.2 Hilfsmittel EBU-Teletextanzeige	72
6.2 TSI – Transportschicht	9	B.3 Hilfsmittelklasse Chipkartenleser	73
6.3 TSI – obere Schichten	9	B.4 Klasse für die Unterstützung zukünftiger DVB-EPG-Ereignisse	76
7 Befehlsschnittstelle – Transport- und Sitzungsschichten	10		
7.1 Generische Transportschicht	10		
7.2 Sitzungsschicht	14		
8 Befehlsschnittstelle – Anwendungsschicht	20		

1 Einführung und Anwendungsbereich

Für den Digitalen Fernseh Rundfunk wurde eine Reihe von Normen geschaffen. Diese Normen enthalten Quellencodierung, Kanalcodierung, Informationen über Dienste und Decoderschnittstellen. Wenn die Notwendigkeit besteht, den Zugriff auf einen Rundfunkdienst zu steuern, wird zusätzlich ein System zur Zugriffsbeschränkung (CA) benutzt. Es wurde entschieden, dass es nicht erforderlich ist, dieses System zu normen, obwohl ein einheitlicher Scrambling-Algorithmus zur Verfügung steht. Den Rundfunkbetreibern bleibt dann die Möglichkeit, Decoder mit unterschiedlichen Systemen zur Zugriffsbeschränkung zu nehmen und sicherzustellen, dass es bei der Beschaffung solcher Systeme eine entsprechende Auswahl gibt. Eine Lösung besteht darin, den einheitlichen Scrambling-Algorithmus zu verwenden und für den Zugriff Lösungen zu benutzen, die auf kommerziellen Abmachungen zwischen den Betreibern beruhen. Diese Lösung kann mit einzelnen CA-Systemen betrieben werden, die in die Decoder eingebaut sind.

Eine zweite Lösung beruht auf einer genormten Schnittstelle zwischen einem Modul und einem Hauptgerät, wobei in dem Modul Funktionen zur Zugriffsbeschränkung und mehr allgemein definierte, systemgebundene Funktionen implementiert sein können. Diese Lösung erlaubt auch den Rundfunkbetreibern Module zu benutzen, die in demselben Rundfunksystem Lösungen von verschiedenen Lieferanten enthalten, wodurch sich die Auswahl und die Möglichkeiten gegen Piraterie verbessern. Zweck dieses Schriftstückes ist es, diese Einheitliche Schnittstelle zu beschreiben.

Der Decoder, in dieser Spezifikation als Hauptgerät bezeichnet, enthält die erforderlichen Funktionen, um MPEG-2-Bild, -Ton und -Daten in Klartext zu empfangen. Diese Spezifikation legt die Schnittstelle zwischen dem Hauptgerät und den Scrambling- und CA-Anwendungen fest, die mit einem externen Modul betrieben werden.

Es werden zwei logische Schnittstellen definiert, die in derselben physikalischen Schnittstelle enthalten sind. Die erste Schnittstelle ist die Schnittstelle für den MPEG-2-Transportstrom. Die Verbindungsschicht und die physikalische Schicht werden in dieser Spezifikation, die höheren Schichten in den MPEG-2-Spezifikationen definiert. Die zweite Schnittstelle, die Befehlsschnittstelle, überträgt die Befehle zwischen dem Hauptgerät und dem Modul. Für diese Schnittstelle werden sechs Ebenen definiert. Ein Beispiel eines einzelnen Moduls in Verbindung mit einem Hauptgerät wird in Bild 1 gezeigt.

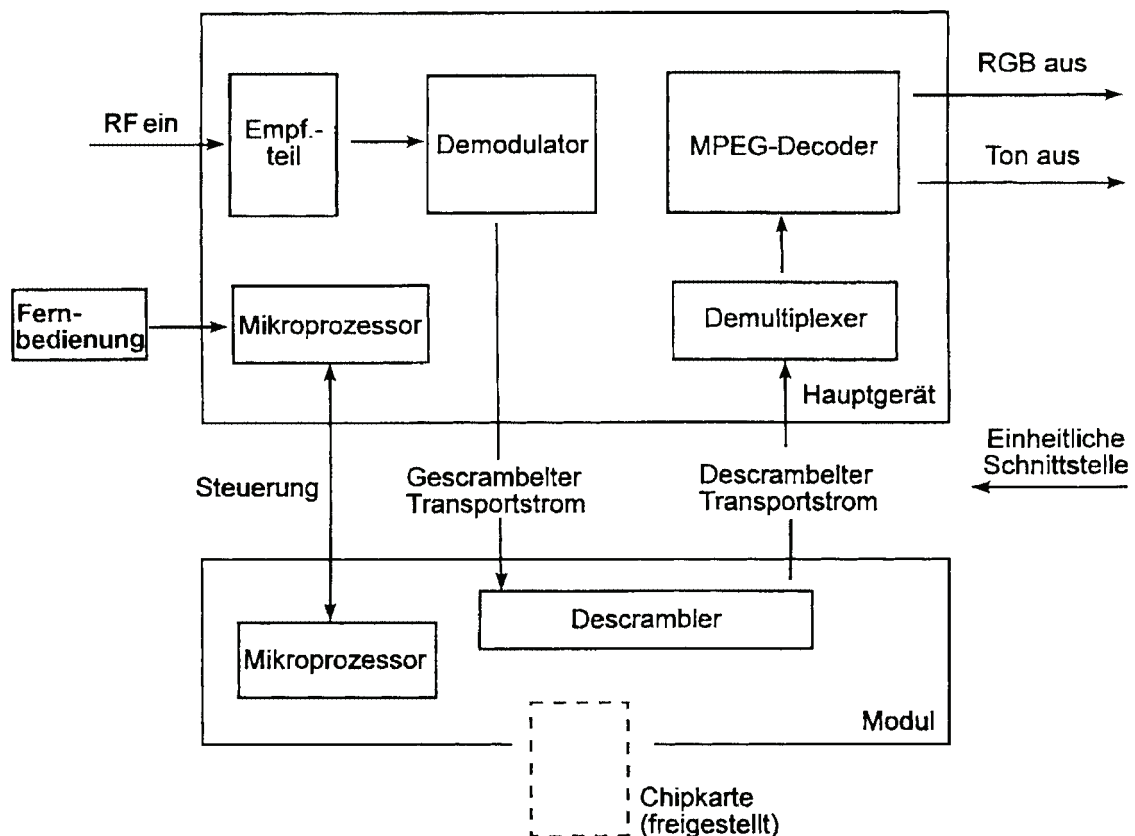


Bild 1 – Beispiel eines einzelnen Moduls in Verbindung mit dem Hauptgerät

Diese Spezifikation definiert nur solche Aspekte des Hauptgerätes, die erforderlich sind, um die Dialoge über diese Schnittstelle vollständig festzulegen. Die Spezifikation setzt nichts über die Konstruktion des Hauptgerätes voraus, mit der Ausnahme, dass eine Reihe von Hilfsmittel definiert wird, die das Hauptgerät benötigt, um das Modul betreiben zu können.

Die Spezifikation definiert nicht den Betrieb oder funktionelle Fähigkeiten einer CA-System-Anwendung im Modul. Die Anwendungen, die von einem Modul durchgeführt werden können, das über die Schnittstelle mit dem Hauptgerät in Verbindung steht, sind nicht auf Zugriffsbeschränkung oder auf Anwendungen begrenzt, die in dieser Spezifikation beschrieben sind. Es dürfen mehr als ein Modul gleichzeitig unterstützt werden.

2 Normative Verweisungen

Diese Europäische Norm enthält durch datierte oder undatierte Verweise Angaben aus anderen Publikationen. Diese normativen Verweisungen werden an den entsprechenden Stellen im Text zitiert und die Publikationen nachstehend aufgelistet. Für datierte Verweisungen gelten spätere Änderungen oder Überarbeitungen dieser Publikationen für die vorliegende Europäische Norm nur dann, wenn sie durch Änderung oder Überarbeitung aufgenommen sind. Für undatierte Verweisungen gilt die letzte Ausgabe der Publikation, auf die verwiesen wurde (einschließlich Änderungen).

- [1] ISO/IEC 13818-1, *Generic Coding of Moving Pictures and Associated Audio: Systems*.¹⁾
- [2] ISO 8824:1987, *Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN. 1)*.
- [3] ISO 8825:1987, *Open Systems Interconnection – Specification of basic encoding rules for Abstract Syntax Notation One (ASN. 1)*.
- [4] ETS 300468, *Specification for Service Information (SI) in Digital Video Broadcasting (DVB) systems*.
- [5] ETR 162, *Allocation of Service Information (SI) codes for Digital Video Broadcasting (DVB) Systems*.
- [6] PC Card Standard, *Volume 2 – Electrical Specification, February 1995, Personal Computer Memory Card International Association, Sunnyvale, California*.
- [7] PC Card Standard, *Volume 3 – Physical Specification, February 1995, Personal Computer Memory Card International Association, Sunnyvale, California*.
- [8] PC Card Standard, *Volume 4 – Metaformat Specification, February 1995, Personal Computer Memory Card International Association, Sunnyvale, California*.
- [9] prETS 300743, *DVB Subtitling Specification*.

3 Definitionen

Für die Anwendung dieser Norm gelten die folgenden Definitionen:

Anwendung:

Eine Anwendung läuft in einem Modul, das mit dem Hauptgerät in Verbindung steht. Sie bietet dem Benutzer darüber hinaus solche Möglichkeiten, die direkt vom Hauptgerät zur Verfügung gestellt werden. Eine Anwendung kann den Transportstrom verarbeiten.

Hauptgerät:

Gerät, an das ein oder mehrere Module angeschlossen werden können, z. B. IRD, VCR, PC.

Modul:

Kleines Gerät, das nicht für sich alleine betrieben werden kann, hergestellt um zusammen mit dem Hauptgerät spezialisierte Aufgaben durchzuführen, zum Beispiel: ein Untersystem für Zugriffsbeschränkung, ein Anwendungsmodul für einen elektronischen Programmführer oder um Hilfsmittel bereitzustellen, die von einer Anwendung angefordert werden, aber von dem Hauptgerät nicht direkt geliefert werden können.

Hilfsmittel:

Funktionelle Einheit, die vom Hauptgerät zur Benutzung in einem Modul zur Verfügung gestellt wird. Ein Hilfsmittel definiert eine Reihe von Objekten, die zwischen einem Modul und dem Hauptgerät ausgetauscht werden, durch das das Modul das Hilfsmittel benutzt.

¹⁾ Nationale Fußnote: Zu ergänzen ist ISO/IEC 13818-9; *Information technology – Generic Coding of Moving Pictures and Associated Audio Information – Part 9: Extension for real time interface for real time decoders*

Dienst:

Reihe von Elementarströmen, die dem Benutzer als Programm angeboten werden. Sie sind durch eine gemeinsame Synchronisierung verbunden. Sie können aus verschiedenen Daten bestehen, d. h. Bild, Ton, Untertitel und andere Daten.

Transportstrom:

MPEG-2-Transportstrom.

4 Designphilosophie

4.1 Schichtung

Die Spezifikation wird in Schichten beschrieben, um zukünftige Varianten beim Implementieren aufnehmen zu können. Die Anwendungs- und Sitzungsschicht werden für alle Anwendungen der einheitlichen Schnittstelle definiert. Die Transport- und die Verbindungsschicht können von der physikalischen Schicht abhängig sein, die bei einer speziellen Implementierung benutzt wird. Die physikalische Schnittstelle wird innerhalb dieser Spezifikation definiert und umfasst die vollständige physikalische Spezifikation des Moduls.

Die Schichtung der Spezifikation ermöglicht Flexibilität beim Einsatz der Schnittstelle für eine Reihe von Anwendungen über CA hinaus. Die Schichtung lässt auch zu, dass es mehrere CA-Verfahren für dasselbe Hauptgerät gibt.

Die grundlegende Schichtung an der Befehlsschnittstelle ist in Bild 2 dargestellt. Das Hauptgerät darf Transportverbindungen mit mehr als einem Modul einrichten, die direkt oder indirekt mit dem Hauptgerät verbunden sind. Jede Verbindung bleibt so lange erhalten, wie das Modul vorhanden ist. Jedes Modul darf eine Anzahl unterschiedlicher Sitzungen mit dem Hauptgerät durchführen.

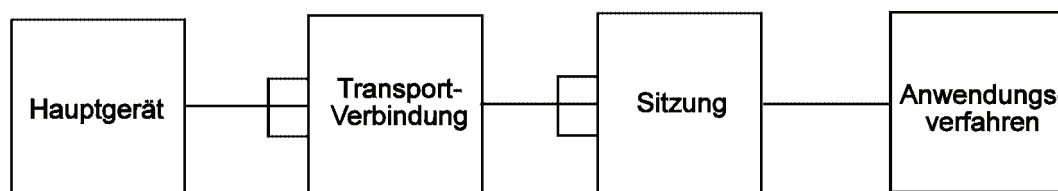


Bild 2 – Schichtung an der Befehlsschnittstelle

4.2 Physikalische Implementierung

Die Basisspezifikation enthält die Implementierung an einer physikalischen Schnittstelle, die mit der PC-Kartennorm kompatibel ist, die in der PC-Industrie benutzt wird. Andere physikalische Implementierungen sind in der Zukunft möglich.

4.3 Client-Server

Die Schnittstelle ist nach dem Prinzip konzipiert, dass Anwendungen, wie Clients, die von einem Server bereitgestellten Hilfsmittel benutzen. Die Anwendungen befinden sich in einem Modul und die Hilfsmittel können entweder durch das Hauptgerät oder ein anderes Modul auf eine von dem Hauptgerät bestimmte Art gehandhabt werden. Im englischen Text wurde für Hilfsmittel anstelle der Benennung 'services' vorzugsweise die Benennung 'resources' benutzt, weil 'services' beim Rundfunk allgemein für 'TV and radio services' verwendet wird und dort Unklarheiten vermieden werden sollten.

4.4 Codierung der Daten

Die Übertragung der Daten über die Befehlsschnittstelle wird in Form von Objekten definiert. Die Objekte werden mit Hilfe einer allgemeinen Tag-Längenwert(TLV)-Codierung codiert, die von der Syntax abgeleitet ist, die für das Codieren von ASN.1-Syntax benutzt wird (siehe [2] und [3]). Sie ist im Allgemeinen erweiterbar. Es gibt eine spezielle Transportschicht-Codierung für die PC-Karten-Implementierung, sie kann aber bei anderen physikalischen Implementierungen anders sein. Die Semantik ist jedoch identisch.

4.5 Erweiterbarkeit

Die höheren Schichten wurden so konzipiert, dass sie erweiterbar sind. Wie oben gezeigt, ist die benutzte TLV-Codierung erweiterbar, so dass neue Objekte hinzugefügt und bestehende Objekte erweitert werden können. Es gibt bezüglich des Platzbedarfs für die Tag-Codierung oder einer Längenbeschränkung des Wertes keine Probleme. Das Hilfsmittel „Hilfsmittelmanager“ bietet einen Mechanismus zur Erweiterung des Bereiches der Hilfsmittel, die von den Hauptgeräten sowohl für CA-Zwecke als auch für andere auf Modulen basierende Anwendungen bereitgestellt werden.

4.6 Einbringen bestehender Normen

Bestehende Normen wurden, wo möglich und geeignet, als Bausteine für diese Spezifikation benutzt. Dies ergibt wichtige zeitliche Marktvorteile, da die gesamte Entwicklungsarbeit für diese Normen schon geleistet wurde. Es bringt auch Vorteile bei der Implementierung, weil Software und Hardware, die schon für bestehende Normen entwickelt wurden, hier mit möglichen Kostenvorteilen wiederverwendet werden können.

5 Beschreibung und Architektur

5.1 Überblick

Um die Stelle im Hauptgerät zu definieren, an der die Einheitliche Schnittstelle logischerweise angeordnet sein kann, wird für ein Hauptgerät eine teilweise logische Architektur vorausgesetzt. In anderer Hinsicht wurde die Freiheit für den Entwickler des Hauptgerätes so wenig wie möglich eingeschränkt. Bild 1 zeigt ein vereinfachtes Bild einer typischen Hauptgerätearchitektur und die Lage der Schnittstelle. Dabei ist zu beachten, dass es in einem Hauptgerät mehr als eine Schnittstelle geben kann.

Die Einheitliche Schnittstelle besteht aus zwei Teilen, der Transportstromschnittstelle und der Befehlsschnittstelle. Beide sind geschichtet, um die Gesamtkonstruktion der Schnittstelle und die Implementierung zu erleichtern. Die höheren Schichten sind für alle Implementierungen einheitlich. Für die unteren Schichten sind alternative Implementierungen möglich. Diese Spezifikation enthält eine, die auf der PC-Kartennorm basiert. In zukünftigen Versionen können aber andere enthalten sein.

5.2 Transportstromschnittstelle

Die Transportstromschnittstelle überträgt MPEG-2-Transportpakete in beide Richtungen. Wenn das Modul den Zugriff auf bestimmte Dienste im Transportstrom freigibt und diese Dienste vom Hauptgerät ausgewählt wurden, dann werden die Pakete, die diese Dienste übertragen, descrambelt zurückgegeben. Die anderen Pakete werden nicht verändert. An der Transportstromschnittstelle wird durch das Modul und eine zugeordnete Bedingungslogik der physikalischen Schicht unter den meisten Bedingungen eine konstante Verzögerung aufrechterhalten (siehe 5.4.2). Die Schichten der Transportstromschnittstelle werden in Bild 3 gezeigt. Die Transportschicht und alle höheren Schichten werden in der MPEG-2-Spezifikation ISO 13818 definiert.

Höhere Schichten
Transportschicht
PC-Karten-Verbindungsschicht
PC-Karten-physikalische Schicht

Bild 3 – Schichten der Transportstromschnittstelle

5.3 Befehlsschnittstelle

Die Befehlsschnittstelle überträgt jede Art von Kommunikation zwischen den in dem Modul laufenden Anwendungen und dem Hauptgerät. Um für die notwendige Funktionsfähigkeit zu sorgen, werden die Kommunikationsprotokolle an dieser Schnittstelle in verschiedenen Schichten definiert. Diese Funktionsfähigkeit enthält die Fähigkeit, an einem Hauptgerät mehrere Module zu unterstützen, die Fähigkeit komplexe Kombinationen von Transaktionen zwischen Modul und Hauptgerät zu unterstützen und eine erweiterbare Reihe von funktionellen Grundformen (Objekte), die dem Hauptgerät erlauben, Hilfsmittel an die Module zu liefern. Die Schichtung wird nachstehend in Bild 4 gezeigt.

Die in dieser Spezifikation beschriebene PC-Karten-Implementierung hat ihre eigene physikalische und Verbindungsschicht und auch ihre eigene Untere Transport-Unterschicht. Eine zukünftige unterschiedliche Implementierung unterscheidet sich wahrscheinlich in diesen Schichten und der Unterschied wird auf diese Schichten beschränkt sein. Die implementierungsspezifischen Merkmale der Unteren Transport-Unterschicht sind auf das Codieren beschränkt und auf spezielle Einzelheiten des Nachrichtenaustauschprotokolls. Die gemeinsame Obere Unterschicht definiert Kennzeichnung, Beginnen und Beenden von Transportschichtverbindungen. Die Sitzungs-, Hilfsmittel- und Anwendungsschichten sind für alle physikalischen Implementierungen gemeinsam.

So weit wie möglich wurde die Anwendungsschicht der Schnittstelle so ausgeführt, dass sie frei von spezieller Anwendungssemantik ist. Die Kommunikation findet in Form von Hilfsmitteln, wie Benutzer-Schnittstellen-Dialog, und als Kommunikation mit niedriger Datenrate statt, die das Hauptgerät den in einem Modul laufenden Anwendungen zur Verfügung stellt. Diese Strategie macht es sehr viel leichter, Module für andere Aufgaben als nur die Zugriffsbeschränkung zu beschaffen.

Anwendung			
Hilfsmittel:			
Benutzerschnittstelle	Kommunikation niedriger Datenrate	System	Freigestellte Erweiterungen
Sitzungsschicht			
Generische Transport-Unterschicht			
PC-Karten-Transport-Unterschicht			
PC-Karten-Verbindungsschicht			
PC-Karten-physikalische Schicht			

Bild 4 – Schichtung der Befehlsschnittstelle

5.4 Physikalische Anforderungen

5.4.1 Einführung

Dieser Abschnitt definiert die Anforderungen, die die physikalische Schicht erfüllen muss, um alle benötigten Funktionen durchführen zu können. Die folgenden Kennwerte der physikalischen Schicht werden hier nicht eingeschränkt, obwohl in der Spezifikation für jede benutzte physikalische Schicht die mechanische und elektrische Verbindung zwischen dem Hauptgerät und dem Modul, d. h. Typ und Größe der Buchse, Anzahl der Kontaktstifte, Spannungen, Impedanzen und Leistungsgrenzwerte festgelegt werden.

Anforderungen und Grenzwerte der folgenden Kennwerte der physikalischen Schicht werden hier definiert:

- logische Verbindungen für Transportstrom und Befehle;
- Datenraten;
- Verhalten beim Verbinden und Trennen;
- Initialisierung auf unterer Ebene;
- Verwendung von mehreren Modulen.

5.4.2 Logische Verbindungen für Daten und Befehle

Die physikalische Schicht muss unabhängige logische Verbindungen in beiden Richtungen für den Transportstrom und für die Befehle unterstützen.

Die Transportstromschnittstelle muss einen MPEG-2-Transportstrom annehmen, der aus einer Folge von Transportpaketen besteht, entweder zusammenhängend oder durch Nulldaten getrennt. Der zurückgesendete Transportstrom kann einige der eingehenden Transportpakete in descrambelter Form enthalten. Die Transportstromschnittstelle ist Gegenstand der folgenden Beschränkungen:

- 1 Wenn das Modul die Quelle eines Transportstromes ist, muss sein Ausgangssignal ISO/IEC 13818-9 entsprechen.
- 2 Jedes Ausgangspaket muss zusammenhängend sein, wenn die Quelle des Paketes oder des Eingangspaketes zusammenhängend ist.
- 3 Ein Modul muss eine konstante Verzögerung einführen, wenn es ein Eingangs-Transportpaket mit maximaler Abweichung der Verzögerung (t_{mdv}) verarbeitet, die nach folgender Gleichung für jedes Byte gilt:

$$t_{mdv_{max}} = (n * TMCLKI) + (2 * TMCLKO)$$

und

$$t_{mdv_{max}} \leq 1 \text{ Mikrosekunde, wenn } n = 0.$$

Dabei ist:

t_{mdv} Abweichung der Verzögerung des Moduls;

n Anzahl der Lücken in dem entsprechenden Eingangs-Transportpaket;

$TMCLKI$ Eingangsdaten-Taktperiode;

$TMCLKO$ Ausgangsdaten-Taktperiode.

* Eine 'Lücke' wird durch eine MCLKI-Anstiegflanke definiert, für die das MIVAL-Signal inaktiv ist. Eine 'Lücke' ist immer ein Byte breit. Mehrere 'Lücken' dürfen aufeinander folgen.

- * Alle Hauptgeräte sollten zusammenhängende Transportpakete ausgeben.
 - * Hauptgeräte dürfen nicht zusammenhängende Transportpakete nur dann abgeben, wenn sie weniger als 3 Buchsen für die einheitliche Schnittstelle implementieren.
 - * Die Anzahl der 'Lücken innerhalb eines Paketes' können beträchtlich voneinander abweichen.
- 4 Ein der Befehlsschnittstelle entsprechendes Hauptgerät sollte konzipiert sein, Nm Module zu unterstützen. Nm ist die größere Zahl der durch das Hauptgerät implementierten Buchsen der Befehlsschnittstelle oder 16. Es sollte den Jitter, der sich von den Nm Modulen ergibt, plus dem Jitter in dem Eingangs-Transportstrom tolerieren. Der Worst-Case-Jitter kann entweder von dem eigenen Eingang des Hauptgerätes, dem Nm Module folgen oder einem Eingangsmodul mit ISO/IEC 13818-9 entsprechendem Ausgang, auf den $(Nm - 1)$ Module folgen, herrühren.
- 5 Alle Schnittstellen müssen Datenraten von mindestens 58 Mbit/s unterstützen, gemittelt über den Zeitabschnitt zwischen den Synchronisationsbytes aufeinander folgender Transportpakete.
- 6 Alle Schnittstellen müssen eine Mindest-Byteübertragungs-Taktperiode von 111 ns unterstützen.

Die Befehlsschnittstelle muss Befehle, wie sie in dem entsprechenden Teil dieser Spezifikation über die Transportschicht definiert sind, in beiden Richtungen übertragen. Die unterstützte Datenrate muss in jeder Richtung mindestens 3,5 Mbit/s betragen.

5.4.3 Verhalten beim Verbinden und Trennen

Die physikalische Schicht muss Verbinden und Trennen des Moduls jederzeit unterstützen, unabhängig davon ob das Hauptgerät eingeschaltet ist oder nicht. Verbinden oder Trennen darf keinen elektrischen Schaden, weder am Modul noch am Hauptgerät, verursachen und darf keine unerwünschte Änderung von gespeicherten nicht-flüchtigen Daten in dem Modul verursachen. Wenn ein Modul nicht angeschlossen ist, muss die Transportstromschnittstelle das Modul überbrücken, und die Befehlsschnittstelle zu diesem Modul muss inaktiv sein. Zum Verbinden des Moduls muss das Hauptgerät eine Initialisierungssequenz auf unterer Ebene mit dem Modul einleiten. Dies wird durchgeführt, unabhängig davon, welches Verfahren zum Einrichten der Verbindung auf unterer Ebene von der speziellen physikalischen Schicht benutzt wird, und dann so eingerichtet, dass das Modul ein DVB-konformes Modul ist. Nach erfolgreichem Abschluss muss das Hauptgerät die Transportstromverbindung durch Einfügen des Moduls in den Transportstrompfad des Hauptgeräts einrichten. Es wird akzeptiert, dass während dieses Vorgangs einige Transportstromdaten verloren gehen. Gleichzeitig muss eine Transportschichtverbindung an der Befehlsschnittstelle eingerichtet werden, um zu ermöglichen, dass die Initialisierung der Anwendungsschicht stattfindet und die übliche Anwendungsschicht-Kommunikation fortgeführt wird.

Wenn die physikalische Schicht in anderen Anwendungen als in einer DVB-konformen Modulverbindung benutzt wird und wenn ein nichtkonformes Modul an das Hauptgerät angeschlossen ist, darf keine Beschädigung am Modul oder Hauptgerät verursacht werden. Das Hauptgerät darf keine Initialisierung wie bei einem DVB-konformen Modul durchführen. Wahlweise kann das Hauptgerät dem Benutzer anzeigen, dass ein nicht erkanntes Modul angeschlossen wurde.

Beim Trennen des Moduls muss das Hauptgerät das Modul aus dem Transportstromdatenpfad wegnehmen. Es ist akzeptabel, dass einige Transportstromdaten während dieses Vorgangs verloren gehen. Auch die Befehlsschnittstellenverbindung muss durch das Hauptgerät abgeschlossen werden.

5.4.4 Mehrere Module

Die Anwendungsschicht setzt in der Anzahl der Module, die gleichzeitig an das Hauptgerät angeschlossen werden können, keine Grenzen. Bei speziellen physikalischen Schichten und speziellen Hauptgerätekonstruktionen kann dies jedoch vorkommen. Die Spezifikation der physikalischen Schicht muss zulassen, dass dort mehrere Module gleichzeitig angeschlossen werden, wenn auch bei einem Hauptgeräte-Minimalkonzept nur eine Verbindung vorhanden sein kann. Idealerweise sollte die Spezifikation der physikalischen Schicht bei der Anzahl der Module keine strengen Grenzen setzen, außer wenn eine Grenze vorgeschrieben wird. Die muss dann bei nicht weniger als 15 Modulen festgelegt werden.

Wenn es Einrichtungen dafür gibt, mehr als ein Modul anzuschließen, muss, wie in Bild 5 gezeigt, die Verbindung der Transportstromschnittstellen nacheinander durch jedes einzelne Modul durchgeschleift werden. Das Hauptgerät muss davon getrennte und gleichzeitige Befehlsschnittstellen-Verbindungen zu jedem Modul aufrechterhalten, so dass Transaktionen zwischen Hauptgerät und Modul für jedes Modul unabhängig durchgeführt werden können. Wenn ein Modul ausgestöpselt wird, darf die Befehlsschnittstellen-Transportschichtverbindung zu keinem anderen Modul gestört oder beendet werden.

Wenn mehrere Module an einem Hauptgerät angeschlossen sind, sollte das Hauptgerät in der Lage sein, die für das Descrambeln von ausgewählten Diensten verwendbare Module auszuwählen.

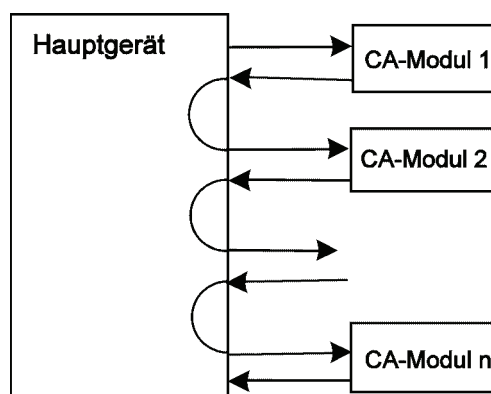


Bild 5 – Verkettung der Transportstromschnittstellen zwischen den Modulen

5.5 Betriebsbeispiel

Um einige der oben beschriebenen Merkmale zu verdeutlichen, behandelt dieses Beispiel die Vorgänge, die dann ablaufen, wenn ein PC-Kartenmodul in ein Hauptgerät gesteckt wird. Die Initialisierung der PC-Karte beginnt mit dem Abtasten des Moduls, das mit Abtaststiften an die Schnittstelle gesteckt wurde. Das Hauptgerät liest dann die Struktur der Karteninformation, die sich in dem Attributspeicher des Moduls befindet. Dieser enthält Konfigurationsinformationen auf unterer Ebene für das Modul, wie PC-Karten-Lesen und -Schreiben von Adressen, die von dem Modul benutzt werden, und zeigt dem Hauptgerät an, dass dies ein DVB-konformes Modul ist. Das Hauptgerät schaltet nun die Überbrückungsleitung der Transportstromschnittstelle ab und ermöglicht den Transportpaketen durch das Modul zu fließen. Dies führt zu einer Verzögerung und folglich zu einer kurzen Lücke in den Transportstromdaten, die jedoch unvermeidbar ist. Gleichzeitig findet der Initialisierungsprozess der physikalischen Schnittstelle statt, um die für die Kommunikation zu benutzende Puffergröße abzustimmen. An dieser Stelle ist der Initialisierungsvorgang der physikalischen Schicht abgeschlossen, und einheitlich für alle physikalischen Implementierungen beginnt der Initialisierungsvorgang der höheren Schicht damit, dass das Hauptgerät eine Transportschichtverbindung zu dem Modul herstellt. Dieser Vorgang und der übrige Initialisierungsvorgang der höheren Schicht werden an anderer Stelle in diesem Schriftstück beschrieben.

Der Initialisierungsvorgang wird logischerweise für die anderen physikalischen Implementierungen ähnlich sein, obwohl sich die Einzelheiten unterscheiden werden.

6 Transportstromschnittstelle (TSI)

6.1 TSI – physikalisch, Verbindungsschichten

Diese Schichten sind von der physikalischen Implementierung des Moduls abhängig.

6.2 TSI – Transportschicht

Die verwendete Transportschicht ist dieselbe wie die Transportschicht des MPEG-2-Systems. Die Datenbewegung über die Transportstromschnittstelle wird mit MPEG-2-Transportpaketen abgewickelt. Das gesamte MPEG-2-Multiplexsignal wird über diese Transportstromschnittstelle übertragen und voll oder teilweise descrambelt wieder empfangen. Wenn das Paket nicht gescrambelt ist, gibt das Modul es so wie es ist zurück. Wenn es gescrambelt ist und das Paket gehört zu dem ausgewählten Dienst und das Modul kann den Zugriff auf diesen Dienst freigeben, dann gibt das Modul das entsprechend descrambelte Paket mit auf '00' gesetztem transport_scrambling_control-Flag zurück.

Wenn das Scrambeln auf der Ebene des Paketierten Elementarstromes (PES) durchgeführt wird, dann reagiert das Modul in derselben Weise und unter gleichen Bedingungen wie oben, und gibt den entsprechend descrambelten PES mit dem auf '00' gesetztem PES_scrambling_control-Flag zurück.

Das Transportpaket und das PES-Paket werden in der MPEG-2-Systemspezifikation [1] vollständig definiert.

6.3 TSI – obere Schichten

Bis auf den Paketierten Elementarstrom ist jede Schichtung oder Struktur der MPEG-2-Daten oberhalb der Transportschicht für diese Spezifikation nicht relevant. Die Spezifikation setzt jedoch voraus, dass das Modul bestimmte Daten, die für seinen Betrieb benötigt werden, wie ECMs (Berechtigungs-Steuerungsmeldungen) und EMMs (Berechtigungs-Verwaltungsmeldungen), findet und direkt aus dem Transportstrom entnimmt.

7 Befehlsschnittstelle – Transport- und Sitzungsschichten

Die Kommunikation der Daten über die Befehlsschnittstelle wird in Form von Objekten definiert. Die Objekte werden mit Hilfe einer allgemeinen Tag-Längenwert-Codierung codiert, die von der Codierung der ASN.1-Syntax abgeleitet ist.

Tabelle 1 – Längenfeld, das bei allen Protokolldateneinheiten bei Transport-, Sitzungs- und Anwendungsschichten benutzt wird

Syntax	Anzahl der Bits	Mnemonic
length_field() { size-indicator if (size_indicator == 0) length_value else if (size_indicator == 1) { length_field_size for (i=0; i<length_field_size; i++){ length_value_byte } } }	1 7 7 8	bslbf uimsbf uimsbf bslbf

Dieser Abschnitt beschreibt die ASN.1-Objekte für die Transport- und die Sitzungsschichten, die über die Befehlsschnittstelle laufen. Bei all diesen Objekten und den Objekten der Anwendungsschicht in Abschnitt 8 gilt für das Längenfeld, das die Anzahl der Bytes in dem darauf folgenden Wertfeld anzeigt, die Codierung nach Tabelle 1.

Das erste Bit des length_field ist der size_indicator. Wenn der size_indicator = 0, wird die Länge des Datenfeldes in den nachfolgenden 7 Bits codiert. Somit kann jede Länge von 0 bis 127 in einem Byte codiert werden. Wenn die Länge 127 übersteigt, wird der size_indicator auf 1 gesetzt. Dann werden in den nachfolgenden 7 Bits die Anzahl der in dem Längenfeld nachfolgenden Bytes codiert. Um einen ganzzahligen Wert zu codieren, müssen diese nachfolgenden Bytes verkettet werden, das erste Byte mit dem höchstwertigen Ende. So kann jede Wertfeldlänge bis zu 65535 mit drei Bytes codiert werden.

Das unbestimmte Längenformat, das durch die Basiscodierregeln von ASN.1 (siehe [3]) festgelegt ist, wird nicht benutzt.

7.1 Generische Transportschicht

7.1.1 Einführung

Die Transportschicht der Befehlsschnittstelle arbeitet oberhalb einer Verbindungsschicht, die durch die benutzte spezielle physikalische Implementierung zur Verfügung gestellt wird. Für die physikalische Implementierung der Basis-PC-Karte wird die Verbindungsschicht in Anhang A beschrieben. Das Transportprotokoll setzt voraus, dass die Verbindungsschicht zuverlässig ist, d. h. dass die Daten in richtiger Reihenfolge und ohne Löschen und Wiederholen transportiert werden.

Das Transportprotokoll ist ein Befehl-Antwort-Protokoll, wobei das Hauptgerät an das Modul einen Befehl sendet, und dabei eine Befehls-Transportprotokoll-Dateneinheit (C_TPDU) benutzt und auf eine Antwort von dem Modul mit einer Antwort-Transportprotokoll-Dateneinheit (R_TPDU) wartet. Das Modul kann die Kommunikation nicht einleiten: es muss auf das Hauptgerät warten, das es abfragt oder zuerst Daten sendet. Das Protokoll wird durch elf Transportschichtobjekte unterstützt. Einige davon treten nur in C_TPDUs vom Hauptgerät auf, einige nur in R_TPDUs vom Modul und einige können in beiden auftreten. Create_T_C und C_T_C_Reply erstellen eine neue Transportverbindungen. Delete_T_C und D_T_C_Reply löschen sie. Mit Request_T_C und New_T_C kann ein Modul das Hauptgerät auffordern, eine neue Transportverbindung herzustellen. Mit T_C_Error können Fehlerbedingungen signalisiert werden. T_SB überträgt Statusinformation vom Modul zum Hauptgerät. T_RCV fordert von einem Modul abrufbereite Daten an und T_Data_More und T_Data_Last befördern Daten von höheren Schichten zwischen Hauptgerät und Modul. T_Data_Last mit einem leeren Datenfeld wird vom Hauptgerät benutzt, um vom Modul regelmäßig Daten abzufragen, wenn es selbst keine Daten zu senden hat.

Ein C_TPDU vom Hauptgerät enthält nur ein Transportprotokollobjekt. Ein R_TPDU vom Modul kann ein oder zwei Transportprotokollobjekte übertragen. Das einzige oder das zweite Objekt eines Paares in einer R_TPDU ist immer ein Objekt T_SB.

7.1.2 Transportprotokollobjekte

Alle Transportschichtobjekte enthalten eine Transportverbindungskennung. Diese ist ein Oktett, das ermöglicht, dass bis zu 255 Transportschichtverbindungen an dem Hauptgerät gleichzeitig aktiv sind. Transportverbindungskennungswert 0 ist reserviert. Der Kennungswert wird immer von dem Hauptgerät zugewiesen. Das Protokoll wird hier in Einzelheiten beschrieben, da es für alle physikalischen Implementierungen einheitlich ist. Die Objekte werden aber nur allgemein beschrieben. Die detaillierte Codierung der Objekte hängt von der benutzten speziellen physikalischen Schicht ab. Die Codierung für die physikalische Implementierung der PC-Karte wird in dem normativen Anhang A beschrieben.

Am Hauptgerät muss es möglich sein, mindestens 16 Transportverbindungen je unterstützter Modulbuchse zu erstellen, wobei aber vorzugsweise alle 255 Verbindungen unter den Modulbuchsen verteilt sein sollten.

- 1 Create_T_C erstellt die Transportverbindung und wird nur vom Hauptgerät ausgegeben und überträgt den Transportverbindungskennungswert für die einzurichtende Verbindung.
- 2 C_T_C_Reply ist die Antwort vom Zielmodul auf Create_T_C und überträgt die Transportverbindungskennung für die erstellte Verbindung.
- 3 Delete_T_C löscht eine bestehende Transportverbindung und hat als Parameter die Transportverbindungskennung für die zu löschende Verbindung. Das Objekt kann sowohl vom Hauptgerät als auch vom Modul ausgegeben werden. Wenn es vom Modul ausgegeben wird, ist es als Antwort auf Abfragen oder auf Daten des Hauptgerätes.
- 4 D_T_C_Reply ist die Antwort auf das Löschen. Unter manchen Umständen kann diese Antwort ihr Ziel nicht erreichen, da das Objekt Delete_T_C eine ihm zugeordnete Zeitbegrenzung für die Antwort hat. Wenn diese Zeitgrenze erreicht ist, bevor die Antwort empfangen wird, dann können alle Aktionen, die beim Empfang der Antwort durchgeführt werden sollten, zu dieser Zeitgrenze durchgeführt werden.
- 5 Request_T_C fordert das Hauptgerät auf, eine neue Transportverbindung zu erstellen. Das Objekt wird über eine bestehende Transportverbindung vom Modul gesendet. Es wird als Antwort auf Abfragen oder auf Daten des Hauptgerätes gesendet.
- 6 New_T_C ist die Antwort auf Request_T_C. Sie wird auf derselben Transportverbindung gesendet wie das Objekt Request_T_C und überträgt die Transportverbindungskennung für die neue Verbindung. Auf New_T_C folgt unmittelbar ein Objekt Create_T_C für die neue Verbindung, das die Transportverbindung entsprechend einrichtet.
- 7 T_C_Error wird gesendet, um einen Fehler zu signalisieren und überträgt einen 1-Byte-Fehlercode, der den Fehler beschreibt. In dieser Version wird T_C_Error nur als Antwort auf Request_T_C gesendet, um zu signalisieren, dass keine weiteren Transportverbindungen zur Verfügung stehen.
- 8 T_SB wird als Antwort auf alle vom Hauptgerät empfangenen Objekte gesendet, entweder an andere Protokollobjekte angehängt oder gegebenenfalls selbständig. T_SB überträgt ein Byte, das anzeigt, wenn beim Modul Daten zum Absenden vorhanden sind.
- 9 T_RCV wird vom Hauptgerät gesendet, um anzufordern, dass die Daten, die im Modul zum Senden vorliegen (in einer vorangegangenen T_SB vom Modul signalisiert), zum Hauptgerät zurückgesendet werden.
- 10 T_Data_More und T_Data_Last befördern Daten zwischen Hauptgerät und Modul und können entweder in einer C_T_PDU oder einer R_T_PDU sein. Sie werden von dem Modul immer nur als Antwort auf eine ausdrückliche Anforderung des Hauptgeräts durch ein T_RCV gesendet. T_Data_More wird benutzt, wenn eine Protokolldateneinheit (PDU) von einer höheren Schicht in Teilstücke aufgespalten werden muss, um sie bei einer externen Beschränkung der Größe der Datenübertragung zu senden. Das Objekt zeigt an, dass mindestens noch ein Teilstück der PDU der oberen Schicht nach diesem einen gesendet wird. T_Data_Last zeigt das letzte oder einzige Teilstück einer PDU der oberen Schicht an.

7.1.3 Transportprotokoll

Die Bilder 6 und 7 zeigen die Zustandsübergangsdiagramme für das Einrichten und Löschen einer Verbindung auf der Seite des Hauptgeräts bzw. des Moduls. Jeder Zustandsübergangsbogen wird mit dem Ereignis gekennzeichnet, das diesen Übergang verursacht. Wenn der Übergang auch ein zu sendendes Objekt verursacht, dann wird dies mit einem eingerahmten Text angegeben.

Wenn das Hauptgerät eine Transportverbindung zu einem Modul einrichten will, sendet es das Objekt Create_T_C und geht in den Zustand 'im Aufbau'. Das Modul muss unmittelbar mit einem C_T_C_Reply-Objekt antworten. Wenn nach einer begrenzten Zeit das Modul nicht antwortet, kehrt das Hauptgerät in den Zustand 'nicht in Betrieb' – über den Bogen 'Zeitbegrenzung' – zurück. Bei dieser speziellen Transportverbindung wird

das Hauptgerät nicht noch einmal senden oder abfragen. Ein zu spätes C_T_C_Reply wird ignoriert. Wenn das Hauptgerät dieselbe Transportverbindungskennung wieder benutzt, wird das Modul wieder Create_T_C empfangen und daraus muss es ableiten, dass die alte Transportverbindung außer Betrieb ist und eine neue eingerichtet werden muss.

Wenn das Modul mit C_T_C_Reply antwortet, geht das Hauptgerät in den Zustand 'Aktiv' der Verbindung. Wenn das Hauptgerät Daten zu senden hat, kann es sie nun senden, sonst gibt es aber eine Abfrage aus und fragt dann danach regelmäßig die Verbindung ab.

Wenn das Hauptgerät die Transportverbindung beenden will, sendet es ein Objekt Delete_T_C und geht in den Zustand 'in Löschung'. Es geht dann bis zum Empfang eines Objektes D_T_C_Reply oder nach einer Zeitbegrenzung, während der nichts empfangen wird, in den Zustand 'nicht in Betrieb' zurück. Wenn das Hauptgerät ein Objekt Delete_T_C vom Modul empfängt, gibt es ein Objekt D_T_C_Reply aus und geht direkt in den Zustand 'nicht in Betrieb'. Außer in dem Zustand 'Aktiv' wird jedes Objekt ignoriert, das in einem Zustand empfangen wird, der nicht erwartet wurde.

In dem Zustand 'Aktiv' sendet das Hauptgerät periodisch Abfragen, oder es sendet Daten, wenn es eine PDU der oberen Schicht senden muss. Als Antwort empfängt es ein Objekt T_SB, eingeleitet durch ein Objekt Request_T_C bzw. Delete_T_C.

In dem Zustand 'Aktiv' können jederzeit vom Hauptgerät Daten gesendet werden. Das Modul kann erst Daten senden, wenn es eine Nachricht vom Hauptgerät empfangen hat – üblicherweise Daten oder eine Abfrage – und dann in der Antwort T_SB anzeigen, dass Daten vorhanden sind. Das Hauptgerät wird dann zu einem Zeitpunkt – nicht notwendigerweise sofort – eine Anforderung T_RCV an das Modul senden, auf die das Modul durch Senden eines Objektes T_Data mit den abrufbereiten Daten antwortet. Wenn T_Data_More benutzt wird, muss jedes darauf folgende Teilstück auf eine weitere Anforderung T_RCV vom Hauptgerät warten, bevor es gesendet werden kann.

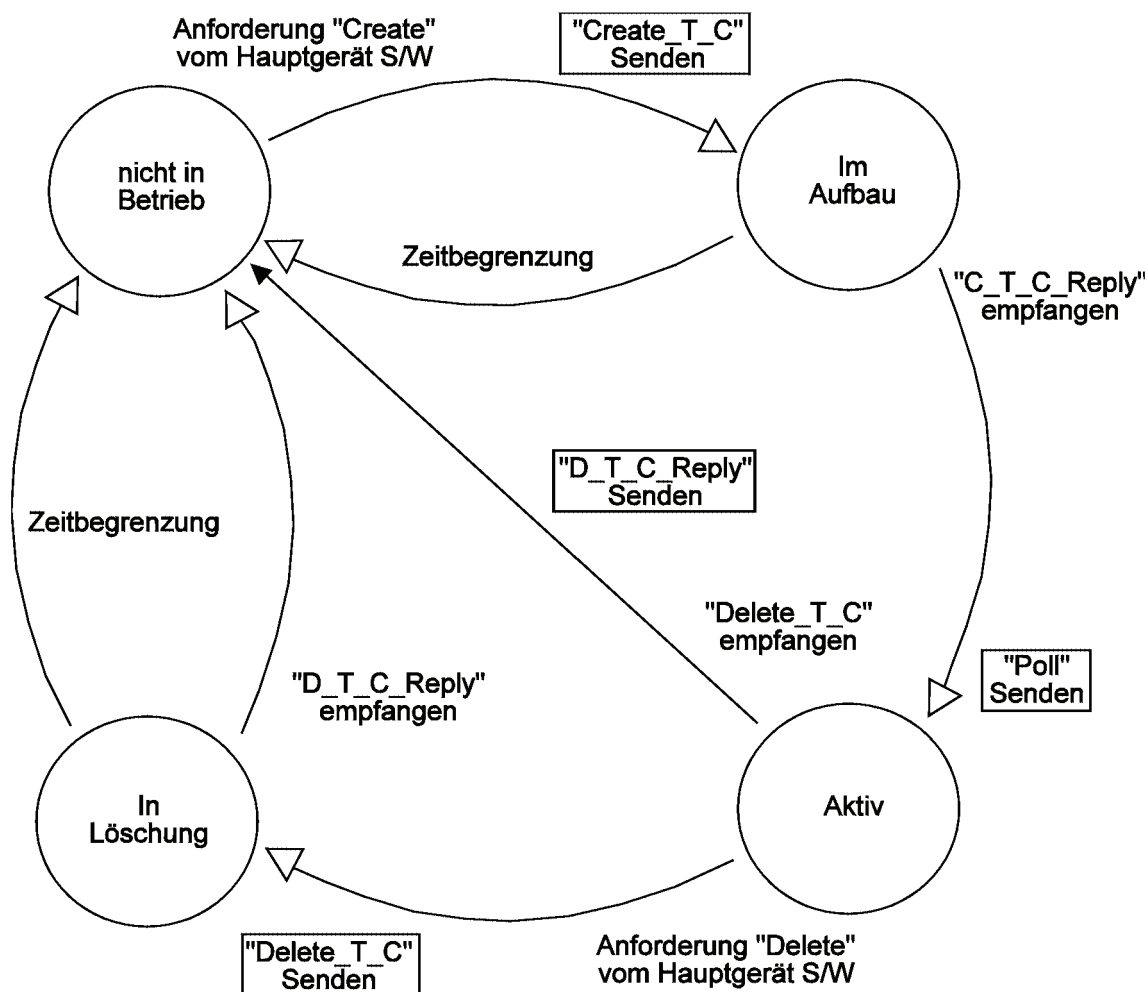


Bild 6 – Diagramm der Zustandsübergänge für die Hauptgerätseite des Transportprotokolls

Tabelle 2 – Objekte, deren Empfang bei den Zuständen einer Transportverbindung am Hauptgerät erwartet wird

Zustand	Erwartete Objekte – Hauptgerät
Nicht in Betrieb	Keine
Im Aufbau	C_T_C_Reply (+ T_SB)
Aktiv	T_Data_More, T_Data_Last, Request_T_C, Delete_T_C, T_SB
In Löschung	D_T_C_Reply (+ T_SB)

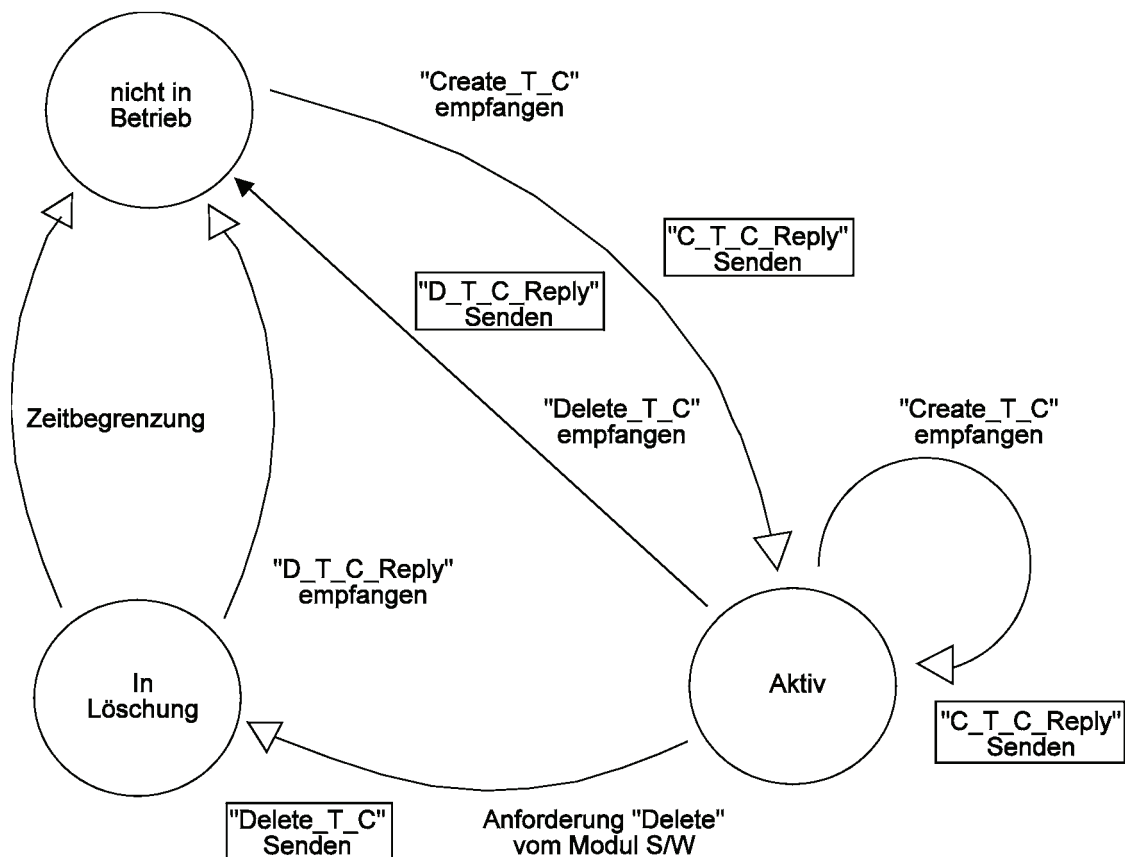


Bild 7 – Diagramm der Zustandsübergänge für die Modulseite des Transportprotokolls

Tabelle 3 – Objekte, deren Empfang bei den Zuständen einer Transportverbindung am Modul erwartet wird

Zustand	Erwartete Objekte – Modul
Nicht in Betrieb	Create_T_C_
Aktiv	Create_T_C, T_Data_More, T_Data_Last, New_T_C, Delete_T_C, T_RCV, T_C_Error
In Löschung	D_T_C_Reply

Zum Einrichten einer weiteren Transportverbindung muss das Modul das Objekt Request_T_C entweder als Antwort auf eine Abfrage oder auf Daten senden. Wenn das Hauptgerät die Anforderung erfüllen kann, wird es mit einer New_T_C antworten, die die Transportverbindungskennung für die neue Verbindung enthält, unmittelbar gefolgt von einem Objekt Create_T_C, um die Verbindung zu erstellen. Wenn das Hauptgerät die Anforderung nicht erfüllen kann, weil alle Transportverbindungskennungen in Benutzung sind, dann muss es mit T_C_Error antworten, das den entsprechenden Fehlercode enthält.

Ein Beispiel einer Objektübertragungsfolge für das Erstellen und Benutzen einer Transportverbindung wird in Bild 8 gezeigt.

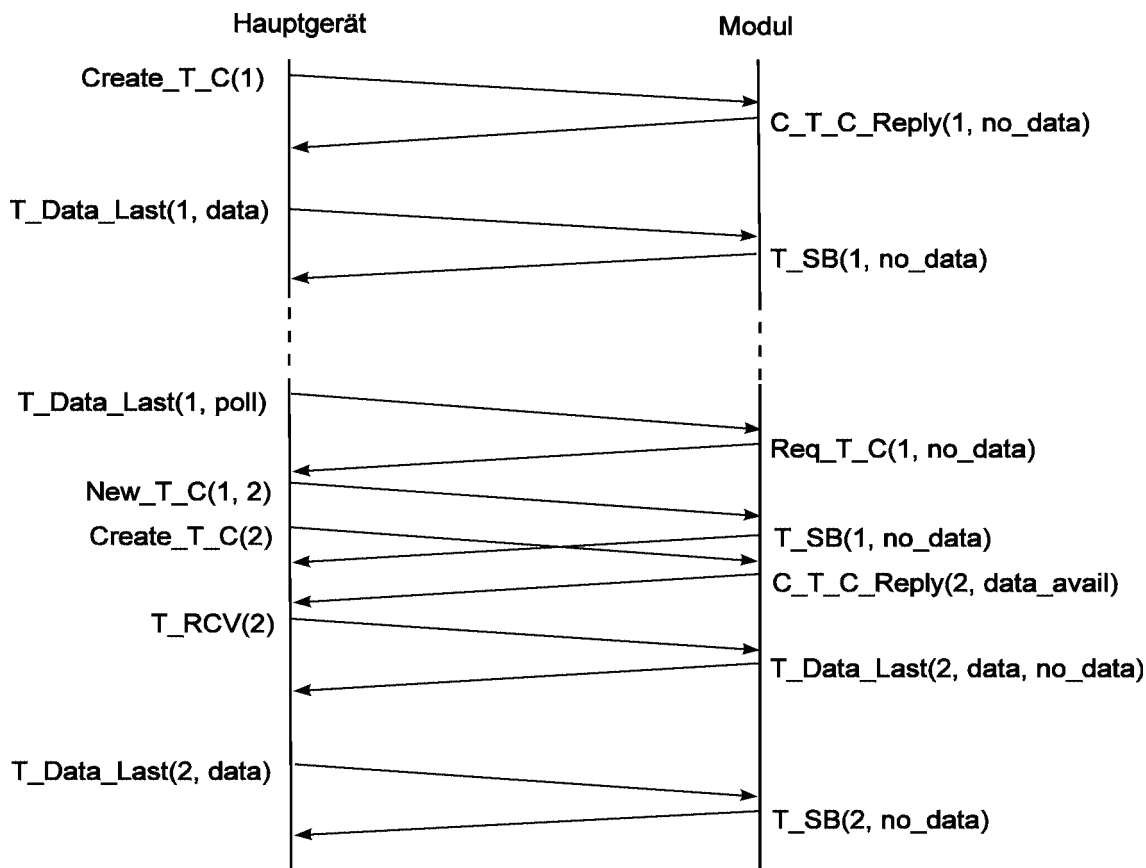


Bild 8 – Beispiel einer Objektübertragungsfolge für das Transportprotokoll

In diesem Beispiel wird angenommen, dass das Modul gerade eingesteckt und eine physikalische Verbindung (PC-Karten-Initialisierung usw.) eingerichtet wurde. Das Hauptgerät gibt nun ein Create_T_C für die Transportverbindung Nummer 1 aus (wenn schon andere Module eingesteckt sind, ist dies eine höhere Nummer). Das Modul antwortet unmittelbar mit C_T_C_Reply für die Transportverbindung 1 und zeigt auch an, dass es keine Daten zu senden hat. Das Hauptgerät sendet einige Daten mit einem T_Data_Last und das Modul antwortet nur mit einem T_SB und zeigt damit an, dass es keine Daten zu senden hat. Einige Zeit später fragt das Hauptgerät das Modul mit einem leeren T_Data_Last ab. Das Modul antwortet mit einem Request_T_C was bedeutet, dass es eine neue Transportverbindung einrichten muss, und zeigt auch an (in dem angehängten T_SB), dass es keine Daten hat, um sie über Verbindung 1 zu senden. Das Hauptgerät antwortet mit New_T_C und zeigt damit an, dass die Transportverbindung 2 eingerichtet wird. Unmittelbar darauf folgt Create_T_C für Transportverbindung 2. Das Modul antwortet mit einem T_SB auf die erste und einem C_T_C_Reply auf die zweite Verbindung und zeigt damit auch an, dass es über diese Verbindung Daten zu senden hat. Das Hauptgerät antwortet mit T_RCV, um die Daten zu empfangen, und das Modul antwortet mit T_Data_Last, das die Daten enthält. Das Hauptgerät antwortet mit seinen eigenen Daten und das Modul antwortet darauf, dass er keine weiteren Daten zu senden hat. Beide Transportverbindungen bleiben nun bestehen, bis entweder das Modul oder das Hauptgerät eine davon löscht. Das Hauptgerät fragt regelmäßig auf beiden Verbindungen mit einem leeren T_Data_Last ab.

7.2 Sitzungsschicht

7.2.1 Einführung

Die Sitzungsschicht stellt den Mechanismus zur Verfügung, durch den die Anwendungen mit den Hilfsmitteln kommunizieren und von den Hilfsmitteln Gebrauch machen. Das Hilfsmittel ist ein Mechanismus zum Zusammenfassen von funktionellen Fähigkeiten in der Anwendungsschicht und wird in 8.2 vollständig beschrieben.

Hilfsmittel unterscheiden sich in der Anzahl der gleichzeitigen Sitzungen, die sie unterstützen können. Manche Hilfsmittel unterstützen nur eine Sitzung. Wenn eine zweite Anwendung versucht, eine Sitzung mit einem Hilfsmittel anzufordern, das schon in Gebrauch ist, wird sie eine Antwort 'Hilfsmittel belegt' empfangen. Andere Hilfsmittel können mehr als eine Sitzung gleichzeitig unterstützen, wobei dann Hilfsmittelanforderungen bis zu einer von dem Hilfsmittel festgelegten Grenze berücksichtigt werden. Ein Beispiel für diesen Fall ist das Hilfsmittel 'Anzeigen', das bei einigen Hauptgeräteimplementierungen in der Lage ist, gleichzeitige Anzeigen in verschiedenen Fenstern zu unterstützen.

7.2.2 Sitzungsprotokollobjekte

Die Sitzungsobjekte werden hier so allgemein ausgedrückt beschrieben, wie es dem Protokoll entspricht. Die ausführliche Codierung der Objekte wird in späteren Abschnitten beschrieben.

- `open_session_request` wird von einer Anwendung über ihre Transportverbindung an das Hauptgerät ausgegeben, das die Benutzung eines Hilfsmittels anfordert. Das Hauptgerät kann das Hilfsmittel direkt unterstützen oder es kann von einem anderen Modul (über eine zweite Transportverbindung) unterstützt werden, wobei das Hauptgerät das Objekt `create_session` benutzt, um die Sitzung über die geeignete Transportverbindung zu erweitern.
- `open_session_response` wird von dem Hauptgerät an eine Anwendung geantwortet, die ein Hilfsmittel angefordert hat, um eine Sitzungsnummer zuzuteilen, oder um dem Modul mitzuteilen, dass seine Anforderung nicht erfüllt werden kann.
- `create_session` wird von dem Hauptgerät an einen Hilfsmittelversorger in einem Modul ausgegeben, um eine Sitzungsanforderung von einer Anwendung auf eine andere Transportverbindung zu erweitern.
- `create_session_response` wird von dem Hilfsmittelversorger in einem Modul an das Hauptgerät geantwortet, so dass das Hauptgerät dem Ausgangsmodul mitteilen kann, ob die Sitzung eröffnet werden kann.
- `close_session_request` wird von einem Modul oder dem Hauptgerät ausgegeben, um eine Sitzung zu schließen.
- `close_session_response` wird von einem Modul oder dem Hauptgerät ausgegeben, um das Schließen der Sitzung zu bestätigen.
- `session_number` leitet immer den Inhalt der SPDU ein, die APDU(s) enthält.

7.2.3 Sitzungsprotokoll

Der Dialog in der Sitzungsschicht wird durch ein Modul oder das Hauptgerät eingeleitet. Nachstehend werden zwei Beispiele erläutert. Im ersten Beispiel (Bild 9) benutzt ein Modul A ein Hilfsmittel, das vom Hauptgerät zur Verfügung gestellt wird. In dem zweiten Beispiel (Bild 10) fordert ein Modul A an, ein Hilfsmittel zu benutzen, das vom Modul B zur Verfügung gestellt wird.

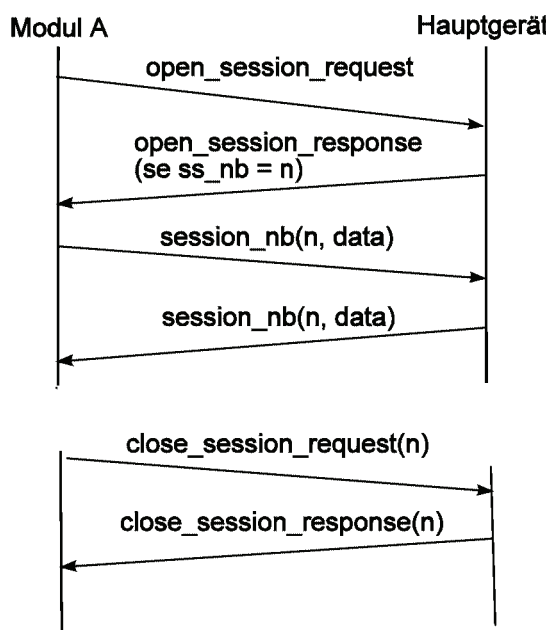


Bild 9 – Sitzung mit einem vom Hauptgerät gelieferten Hilfsmittel

Modul A fordert für ein Hilfsmittel auf seiner Transportverbindung eine Sitzung zu eröffnen. Da das Hauptgerät selbst das Hilfsmittel liefert, antwortet es in seiner 'Sitzung-Eröffnen-Antwort' direkt mit einer Sitzungsnummer. Die Kommunikation wird nun mit Anwendungsschichtdaten fortgeführt, die durch Objekte `session_number` eingeleitet werden. Schließlich wird die Sitzung geschlossen, in diesem Beispiel durch das Modul A. Sie könnte aber auch durch das Hauptgerät geschlossen werden, z. B. wenn das Hilfsmittel aus irgendeinem Grund nicht mehr verfügbar ist.

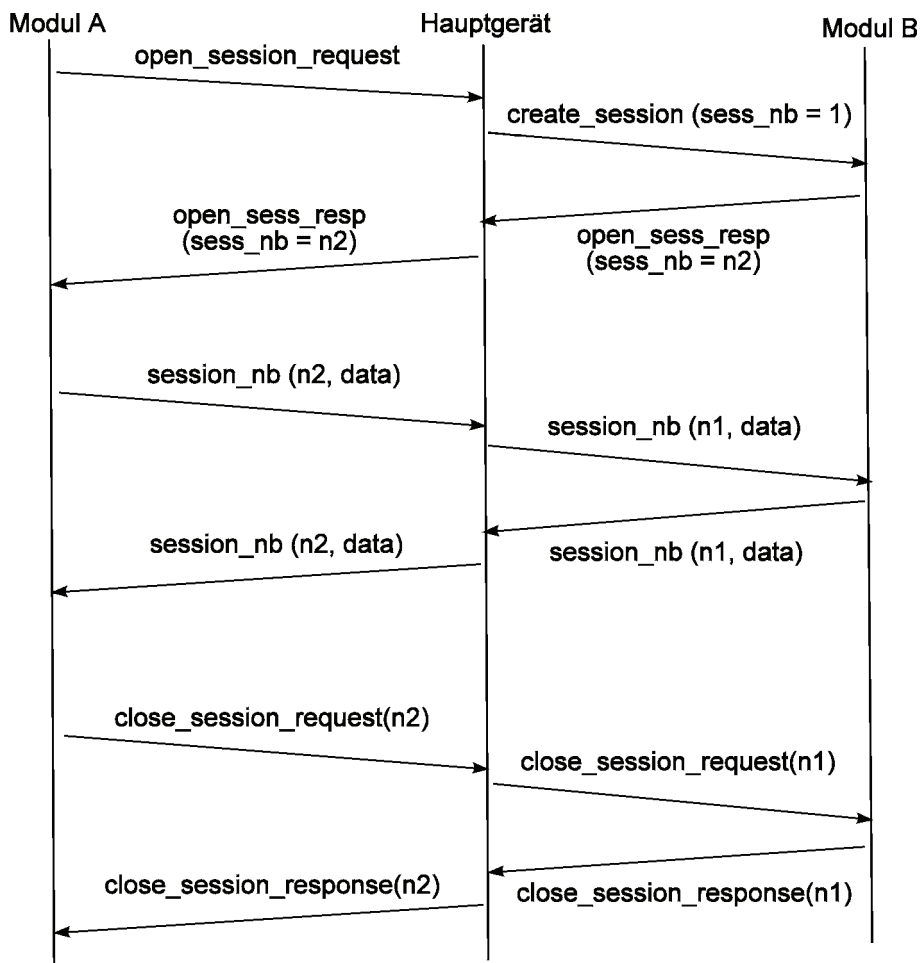


Bild 10 – Sitzung mit einem vom Modul gelieferten Hilfsmittel

Modul A fordert für ein Hilfsmittel, auf seiner Transportverbindung eine Sitzung zu öffnen. Das Hauptgerät kann dieses Hilfsmittel von dem Modul B erhalten, so dass das Hauptgerät mit einem Objekt `create_session` die Sitzung auf die Transportverbindung des Hilfsmittelversorgers erweitert. Der Hilfsmittelversorger antwortet darauf, und das Hauptgerät antwortet dann wieder dem ursprünglich anfordernden Modul A. Die Kommunikation der Anwendungsschichtdaten wird fortgeführt und das Hauptgerät führt die notwendige Lenkung zwischen eingehenden und ausgehenden Transportverbindungen durch. Zuletzt wird die Sitzung geschlossen, in diesem Beispiel durch Modul A. Sie könnte aber auch durch das Hauptgerät oder das Modul B geschlossen werden. Die Sitzungsnummern `n1` und `n2` werden vom Hauptgerät zugeteilt. In diesem Beispiel werden sie als unterschiedlich angegeben. Bei der Implementierung des Hauptgerätes kann gewählt werden, ob es bei den einzelnen Transportverbindungen dieselben oder unterschiedliche Sitzungsnummern benutzt. Die Anwendungsschicht kann keine Voraussetzungen dafür schaffen, dass am Ende der Sitzung die Sitzungsnummern dieselben sind oder nicht.

7.2.4 SPDU-Struktur

Die Sitzungsschicht benutzt zum Austausch der Daten auf Sitzungsebene entweder vom Hauptgerät zum Modul oder vom Modul zum Hauptgerät die Struktur der Sitzungsprotokoll-Dateneinheit (SPDU).

Die SPDU-Struktur wird nachstehend in Bild 11 und Tabelle 4 gezeigt:

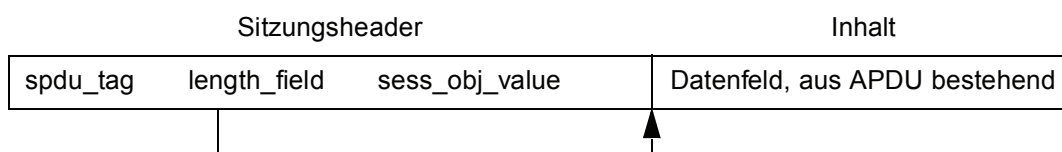


Bild 11 – SPDU-Struktur

Tabelle 4 – Codierung der SPDU

Syntax	Anzahl der Bits	Mnemonik
SPDU() { spdu_tag length_field() for (i=0;i<length_value;i++) session_object_value byte } for (i=0;i<N;i++) { apdu() } }	8 8	uimsbf uimsbf

Die SPDU besteht aus folgenden zwei Teilen:

- einem **obligatorischen** Sitzungsheader, bestehend aus einem Tag-Wert spdu_tag, einem length_field, das die Länge des Sitzungsobjekt-Wertfeldes codiert, und einem Sitzungsobjektwert. Diese Sitzungsobjekte werden nachstehend in 7.2.6 beschrieben. Hinweis: Das Längenfeld enthält nicht die Länge von gegebenenfalls nachfolgenden APDUs;
- einem bedingten Hauptteil variabler Länge, der eine ganzzahlige Anzahl von APDUs enthält, die zu derselben Sitzung gehören (siehe Anwendungsschicht). Das Vorhandensein des Hauptteils hängt von dem Sitzungsheader ab.

7.2.5 Transport der SPDU

Eine SPDU wird in dem Datenfeld von einer oder mehreren TPDU transportiert. Für weitere Information siehe die Beschreibung der TPDU der einzelnen physikalischen Modulimplementierung.

7.2.6 Beschreibung der Sitzungsheader

Die nachstehend aufgelisteten Objekte sind Sitzungsheaderobjekte. Auf nur einen SPDU-Header folgt ein Datenfeld – das Objekt 'session_number' – dem immer ein SPDU-Hauptteil folgt, der eine oder mehrere APDUs enthält.

7.2.6.1 Anforderung Sitzung Öffnen (open_session_request)

Dieses Objekt wird von dem Modul an das Hauptgerät ausgegeben, um das Öffnen einer Sitzung zwischen dem Modul und einem Hilfsmittel anzufordern, das entweder vom Hauptgerät oder vom Modul zur Verfügung gestellt wird. Die resource_identifier muss sowohl nach Klasse als auch nach Typ zu einem Hilfsmittel passen, das in der Liste der verfügbaren Hilfsmittel des Hauptgerätes vorkommt. Wenn das Versionsfeld der gelieferten Hilfsmittelkennung 0 ist, dann benutzt das Hauptgerät die aktuelle Version in seiner Liste. Wenn die Versionsnummer in der Anforderung niedriger oder gleich der Versionsnummer in der Liste des Hauptgerätes ist, dann wird die aktuelle Version benutzt. Wenn die angeforderte Versionsnummer höher als die Version in der Liste des Hauptgerätes ist, dann lehnt das Hauptgerät die Anforderung mit dem entsprechenden Antwortcode ab.

Tabelle 5 – Codierung der Anforderung Sitzung Öffnen

Syntax	Anzahl der Bits	Mnemonik
open_session_request () { open_session_request_tag length_field() = 4 resource_identifier() /* siehe 8.2.2. */ }	8	uimsbf

7.2.6.2 Sitzung-Öffnen-Antwort (open_session_response)

Dieses Objekt wird vom Hauptgerät an das Modul ausgegeben, um eine Sitzungsnummer zuzuteilen oder dem Modul mitzuteilen, dass seine Anforderung nicht erfüllt werden kann.

Tabelle 6 – Codierung der Sitzung-Öffnen-Antwort

Syntax	Anzahl der Bits	Mnemonic
open_session_response() {		
open_session_response_tag	8	uimsbf
length_field() = 7		
session_status	8	uimsbf
resource_identifier()		
session_nb	16	uimsbf
}		

session_status values (Sitzungsstatuswerte)

Tabelle 7 – Statuswerte von Sitzung Öffnen

session status	Sitzungsstatus	Sitzungsstatuswert (hex)
session is opened	Sitzung ist eröffnet	00
session not opened, resource non-existent	Sitzung nicht eröffnet, Hilfsmittel nicht vorhanden	F0
session not opened, resource exists but unavailable	Sitzung nicht eröffnet, Hilfsmittel vorhanden, jedoch nicht verfügbar	F1
session not opened, resource exists but version lower then requested	Sitzung nicht eröffnet, Hilfsmittel vorhanden, jedoch niedrigere Version als angefordert	F2
session not opened, resource busy	Sitzung nicht eröffnet, Hilfsmittel belegt	F3
other	Andere	Reserviert

resource_identifier (Hilfsmittelkennung)

Das Hauptgerät antwortet mit der aktuellen resource_identifier des angeforderten Hilfsmittels zusammen mit der aktuellen Versionsnummer. Wenn die Antwort 'resource non-existent' ist, dann wird das resource_identifier-Feld mit dem Feld identisch sein, das in der Anforderung Sitzung Öffnen geliefert wurde.

session_nb (Sitzungsnummer)

Vom Hauptgerät zugeteilte Nummer für die angeforderte Sitzung. Wert 0 ist reserviert. Die session_nb wird für den gesamten nachfolgenden Datenaustausch von APDUs zwischen dem Modul und dem Hilfsmittel (über das Hauptgerät) benutzt, bis die Sitzung geschlossen wird. Wenn die Sitzung nicht geöffnet werden kann (session_status ungleich 0), hat der session_value keine Bedeutung.

7.2.6.3 Sitzung Erstellen (create_session)

Dieses Objekt wird vom Hauptgerät an ein Modul ausgegeben, das ein Hilfsmittel liefert, um das Öffnen einer Sitzung anzufordern.

Tabelle 8 – Codierung von Sitzung Erstellen

Syntax	Anzahl der Bits	Mnemonic
create_session () {		
create_session_tag	8	uimsbf
length_field() = 6		
resource_identifier()		
session_nb	16	uimsbf
}		

resource_identifier (Hilfsmittelkennung)

Ist der Hilfsmittelkennungswert des angeforderten Hilfsmittels, vollständig mit der aktuellen Versionsnummer.

session_nb (Sitzungsnummer)

Ist die Sitzungsnummer, die vom Hauptgerät für die neue Sitzung zugeteilt wurde.

7.2.6.4 Sitzung-Erstellen-Antwort (create_session_response)

Dieses Objekt wird von dem Modul ausgegeben, das dem Hauptgerät ein Hilfsmittel liefert, um dem Modul mitzuteilen, ob die Sitzung eröffnet werden kann.

Tabelle 9 – Codierung der Sitzung-Erstellen-Antwort

Syntax	Anzahl der Bits	Mnemonik
create_session_response () {		
create_session_response_tag	8	uimsbf
length_field() = 7		
session_status	8	uimsbf
resource_identifier()		
session_nb	16	uimsbf
}		

session_status values (Sitzungsstatuswerte)

Es wird derselbe Statuswert benutzt wie für open_session_response – siehe Tabelle 7.

resource_identifier (Hilfsmittelkennung)

Das Modul setzt in die create_session_response die Klasse und den Typ des angeforderten Hilfsmittels ein, jedoch mit der Versionsnummer, die das Modul zur Zeit unterstützt.

session_nb (Sitzungsnummer)

Hat denselben Wert wie das session_nb-Feld in dem Objekt create_session, für das dies eine Antwort ist.

7.2.6.5 Anforderung Sitzung Schließen (close_session_request)

Dieses Objekt wird vom Modul oder vom Hauptgerät ausgegeben, um eine Sitzung zu schließen.

Tabelle 10 – Codierung der Anforderung Sitzung Schließen

Syntax	Anzahl der Bits	Mnemonik
close_session_request () {		
close_session_request_tag	8	uimsbf
length_field() = 2		
session_nb	16	uimsbf
}		

7.2.6.6 Sitzung-Schließen-Antwort (close_session_response)

Dieses Objekt wird vom Modul oder vom Hauptgerät ausgegeben, um das Schließen der Sitzung zu bestätigen.

Tabelle 11 – Codierung der Sitzung-Schließen-Antwort

Syntax	Anzahl der Bits	Mnemonik
create_session_response () {		
close_session_response_tag	8	uimsbf
length_field() = 3		
session_status	8	uimsbf
session_nb	16	uimsbf
}		

session_status values (Sitzungsstatuswerte)

Tabelle 12 – Statuswerte für Sitzung Schließen

Sitzungsstatus	Sitzungsstatuswert (hex)
Sitzung ist wie gefordert geschlossen	00
Sitzungsnummer in der Anforderung ist nicht zugeteilt	F0
Andere	Reserviert

7.2.6.7 Sitzungsnummer (session_number)

Dieses Objekt geht immer einem Hauptteil der SPDU voraus, die APDU(s) enthält.

Tabelle 13 – Codierung der Sitzungsnummer

Syntax	Anzahl der Bits	Mnemonic
session_number () { session_number_tag length_field() = 2 session_nb }	8 16	uimsbf uimsbf

7.2.7 Codierung der Sitzungs-Tags

Die nachstehende Tabelle 14 enthält die Namen der Objekte, die von der Sitzungsschicht an der Befehlsschnittstelle benutzt werden. Die Codierung des spdu_tag folgt den ASN.1-Regeln. Jedes spdu_tag wird in einem Byte codiert.

Tabelle 14 – Sitzungs-Tag-Werte

spdu-Tag	Tag-Wert (hex)	Einfach (P) oder zusammengesetzt	Richtung Hauptgerät ↔ Modul
T _{open_session_request}	'91'	P	←
T _{open_session_response}	'92'	P	→
T _{create_session}	'93'	P	→
T _{create_session_response}	'94'	P	←
T _{close_session_request}	'95'	P	↔
T _{close_session_response}	'96'	P	↔
T _{session_number}	'90'	P	↔

Weitere Werte sind in den Bereichen 80..8F, 90..9F, A0..AF, B0..BF reserviert.

8 Befehlsschnittstelle – Anwendungsschicht

8.1 Einführung

Die Anwendungsschicht implementiert eine Reihe von Protokollen, die auf dem Konzept eines Hilfsmittels basieren. Ein Hilfsmittel definiert eine funktionelle Einheit, die den in einem Modul laufenden Anwendungen zur Verfügung steht. Jedes Hilfsmittel unterstützt eine Reihe von Objekten und ein Protokoll, um zur Benutzung des Hilfsmittels die Objekte auszutauschen. Kommunikation mit einem Hilfsmittel wird mit einer Sitzung durchgeführt, die von diesem speziellen Hilfsmittel erstellt wurde. Dieser Abschnitt beschreibt den Mindestsatz von Hilfsmitteln, die alle mit dieser Spezifikation konformen Hauptgeräte bereitstellen müssen. Es ist zu beachten, dass einige Hilfsmittel in Abschnitt 8 als Vorsilbe in ihrem Namen „DVB“ haben. Diese Hilfsmittel machen Gebrauch von DVB-spezifischen Merkmalen, die nicht in allen möglichen Situationen vorhanden sein können, in denen diese Schnittstelle benutzt werden kann. Die Hilfsmittel mit der Vorsilbe DVB müssen in allen DVB-konformen Hauptgeräten zur Verfügung stehen und dürfen wahlweise von anderen Hauptgeräten geliefert werden. Weitere wahlfreie Hilfsmittel werden in Anhang B zu dieser Spezifikation beschrieben und dürfen in weiteren Anhängen beschrieben werden.

8.2 Hilfsmittel

8.2.1 Einführung

Ein Hilfsmittel darf von dem Hauptgerät direkt zur Verfügung gestellt werden oder es ist in einem anderen Modul gespeichert. Ein Hilfsmittel wird durch die Hilfsmittelkennung gekennzeichnet, die drei Komponenten enthält: Hilfsmittelklasse, Hilfsmitteltyp und Hilfsmittelversion. Die Hilfsmittelklasse definiert eine Reihe von Objekten und ein Protokoll zu deren Benutzung. Der Hilfsmitteltyp definiert verschiedene Hilfsmiteleinheiten innerhalb einer Klasse. Alle Hilfsmitteltypen innerhalb einer Klasse benutzen dieselben Objekte und dasselbe Protokoll, bieten jedoch unterschiedliche Funktion oder sind unterschiedliche Fälle derselben Funktion. Die Hilfsmittelversion ermöglicht dem Hauptgerät, die letzte Version eines Hilfsmittels zu erkennen (höchste Versionsnummer), wenn mehr als ein Hilfsmittel derselben Klasse und Typ vorhanden sind. Dies ermöglicht aktualisierte oder erweiterte Hilfsmittel, die an ein Modul geliefert werden müssen, um im Hauptgerät oder einem anderen Modul vorhandene Hilfsmittel zu ersetzen. Hilfsmittel mit höherer Versionsnummer müssen zu vorhergehenden Versionen rückwärtskompatibel sein, so dass Anwendungen, die eine vorhergehende Version anfordern, ein Hilfsmittel mit dem erwarteten Verhalten bekommen.

Hilfsmittel werden von einer Anwendung benutzt, die eine Sitzung mit einem Hilfsmittel (siehe 7.2) erstellt. Durch einen vom Hilfsmittel-Manager durchgeführten Initialisierungsprozess hat das Hauptgerät alle verfügbaren Hilfsmittel identifiziert, unabhängig davon, ob das Hauptgerät sie selbst oder ob sie ein anderes Modul bereitstellt, und es kann die Sitzung an der entsprechenden Stelle abschließen. Wenn die Sitzung erstellt wurde, kann die Anwendung das Hilfsmittel durch Austausch von Objekten entsprechend dem definierten Protokoll benutzen.

Ein Beispiel für die Anwendung des Hilfsmittelkonzeptes ist die Hilfsmittelklasse Kommunikation niedriger Datenrate (8.7.1). Diese stellt einen allgemeinen Mechanismus zur Benutzung einer seriellen Kommunikationsverbindung zur Verfügung, typischerweise ein Modem oder ein Rückkanal in einem Kabelsystem. Für verschiedene Übertragungsgeschwindigkeiten des Modems werden Hilfsmitteltypen definiert, die durch die Nummern der ITU-T-Empfehlungen z. B. V.21, V.22, V.32bis gekennzeichnet sind. Die meisten modernen Modems bieten einen Bereich von Geschwindigkeiten, so dass ein Modem mehrere Hilfsmitteltypen enthalten kann. Die Wahl der Übertragungsgeschwindigkeit wird durch Erstellen einer Sitzung mit dem speziellen Hilfsmitteltyp durchgeführt, den das Modem anbietet.

8.2.2 Hilfsmittelkennung (resource_identifier)

Eine Hilfsmittelkennung besteht aus vier Achttbitzeichen. Die beiden höchstwertigen Bits des ersten Achttbitzeichens zeigen an, ob das Hilfsmittel öffentlich oder privat ist, und somit die Struktur des übrigen Feldes. Werte von 0,1 und 2 zeigen ein öffentliches Hilfsmittel an, ein Wert 3 ein privates Hilfsmittel.

Öffentliche Hilfsmittelklassen haben Werte im Bereich von 1 bis 49 150, die das resource_id_type-Feld als den höchstwertigen Teil der resource_class behandeln. Der Wert 0 ist reserviert. Der Maximalwert (ein und derselbe) aller Felder ist reserviert. Private Hilfsmittel werden durch einen privaten Hilfsmitteldefinierer festgelegt, das ist die Organisation, die ein privates Hilfsmittel definiert. Jeder private Hilfsmitteldefinierer kann die Struktur und den Inhalt des private_resource_identify-Feldes auf jede von ihm gewählte Art definieren, mit der Ausnahme, dass der Maximalwert (ein und derselbe) reserviert ist.

Tabelle 15 – Codierung der Hilfsmittelkennung

Syntax	Anzahl der Bits	Mnemonik
resource_identifier() {		
resource_id_type	2	uimsbf
if (resource_id_type != 3) {		
resource_class	14	uimsbf
resource_type	10	uimsbf
resource_version	6	uimsbf
}		
else {		
private_resource_definer	10	uimsbf
private_resource_identity	20	
}		uimsbf
}		

8.2.3 Anwendungen und Hilfsmittelversorger

Dies sind die beiden Funktionsgruppen der Anwendungsschicht, die in einem Modul vorkommen können. Anwendungen machen von Hilfsmitteln Gebrauch, um für den Benutzer des Hauptgerätes Aufgaben zu lösen. Hilfsmittelversorger liefern Hilfsmittel zusätzlich zu denen, die im Hauptgerät direkt verfügbar sind, oder eine neuere Version eines Hilfsmittels, die eine vorher vom Hauptgerät oder einem anderen Modul gelieferte Version ersetzt. Um Probleme mit Blockierungen und Schwierigkeiten bei der Initialisierung zu vermeiden, dürfen die Hilfsmittelversorger nicht von dem Vorhandensein irgendwelcher anderer Hilfsmittel abhängig sein, mit Ausnahme des Hilfsmittelmanagers, der die Hilfsmittel liefert, die sie anbieten.

8.3 Anwendungsprotokoll-Dateneinheit (APDU)

8.3.1 Einführung

Alle Protokolle in der Anwendungsschicht benutzen zum Senden der Anwendungsdaten zwischen Modul und Hauptgerät oder zwischen den Modulen eine einheitliche Struktur für die Anwendungsprotokoll-Dateneinheit (APDU).

Die APDU-Struktur wird nachstehend in Bild 12 und Tabelle 16 dargestellt:



Bild 12 – APDU-Struktur

Tabelle 16 – Codierung der APDU

Syntax	Anzahl der Bits	Mnemonik
APDU() {		
apdu_tag	24	uimsbf
length_field()		
for (i=0;i<length_value;i++) {		
data_byte	8	uimsbf
}		
}		

Die APDU besteht aus zwei Teilen:

- einem vorgeschriebenen Header, der den Tag-Wert `apdu_tag` enthält, der anzeigt, welcher Parameter innerhalb des Datenfeldes gesendet wird, und ein Längenwert, der mit der Länge des folgenden Datenfeldes codiert ist. `apdu_tag` und `length_field` sind entsprechend den Basiscodierungsregeln von ASN.1 codiert;
- einem bedingten Hauptteil variabler Länge gleich der durch `length_field` codierten Länge.

Die APDU-Hauptteile werden in den folgenden Abschnitten beschrieben. Diese Liste kann in zukünftigen Versionen erweitert werden.

8.3.2 Verkettung von APDU-Datenfeldern

Das in der APDU befindliche Datenfeld darf, wenn es wegen der Größe der Übertragungs- und Empfangspuffer des Hauptgerätes und des Moduls erforderlich ist, in mehrere Blöcke geringerer Größe aufgeteilt werden. Dieser Mechanismus steht nur einigen APDUs zur Verfügung. Die Verkettung wird durch die Verwendung von zwei unterschiedlichen `apdu_tag`-Werten (`M_apdu_tag` und `L_apdu_tag`) durchgeführt. Alle Blöcke des Datenfeldes, außer dem letzten, werden innerhalb einer APDU mit einem `M_apdu_tag`-Wert gesendet. Dieser Tag-Wert zeigt an, dass noch mehr Daten in einer weiteren APDU gesendet werden. Der letzte Block des Datenfeldes wird innerhalb einer APDU mit einem `L_apdu_tag`-Wert gesendet. Wenn der letzte Block empfangen ist, verkettet die Empfangseinrichtung alle empfangenen Datenfelder.

Dieser Mechanismus ist für alle APDUs mit zwei definierten Werten (`M_apdu_tag` und `L_apdu_tag`) gültig. Er ist in nachstehendem Bild 13 dargestellt.

8.3.3 Transport von APDUs innerhalb von SPDU

Eine ganzzahlige Anzahl von APDUs, die zu derselben Sitzung gehört, kann in dem Inhalt einer SPDU befördert werden.

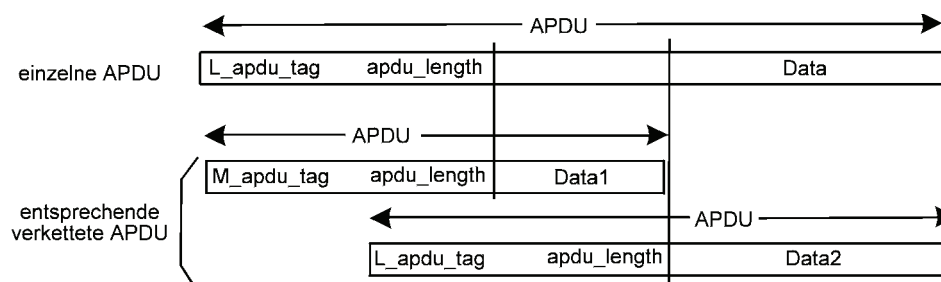


Bild 13 – Verkettungsmechanismus

8.4 Systemmanagement-Hilfsmittel

8.4.1 Hilfsmittelmanager

Der Hilfsmittelmanager ist ein vom Hauptgerät zur Verfügung gestelltes Hilfsmittel. Es gibt in der Klasse nur einen Typ und er kann jede Anzahl von Sitzungen unterstützen. Er steuert die Erfassung und Bereitstellung von Hilfsmitteln für alle Anwendungen. Zwischen Modul und Hauptgerät wird ein symmetrisches Kommunikationsprotokoll definiert, um die Hilfsmittel zu bestimmen, die jeder liefern kann. Das Protokoll wird zuerst vom Hauptgerät benutzt, um dann wieder jede Transportverbindung abzufragen und festzustellen, welche Hilfsmittel gegebenenfalls für die Benutzung in dieser Transportverbindung vorhanden sind. Dann wird es von Anwendungen benutzt, um alle verfügbaren Hilfsmittel zu ermitteln. Es wird dann periodisch benutzt, wenn Hilfsmittel sich ändern, um die allgemeine Übersicht der verfügbaren Hilfsmittel zu aktualisieren. Der Hilfsmittelmanager wird vom Hauptgerät zur Verfügung gestellt und kann nicht durch ein Hilfsmittel in einem Modul ersetzt werden. Jeder Versuch, ein Hilfsmittel 'Hilfsmittelmanager' durch ein Modul zu liefern, muss vom Hauptgerät ignoriert werden.

8.4.1.1 Hilfsmittelmanagerprotokoll

Wenn ein Modul eingesteckt oder das Hauptgerät eingeschaltet wird, werden ein oder vielleicht zwei Transportverbindungen zu dem Modul erstellt, die für eine Anwendung und/oder einen Hilfsmittelversorger bestimmt sind. Das erste, was eine Anwendung oder ein Hilfsmittelversorger ausführt, ist eine Sitzung für das Hilfsmittel 'Hilfsmittelmanager' anzufordern, die immer erstellt wird, weil der Hilfsmittelmanager keine Sitzungsbegrenzung hat. Der Hilfsmittelmanager sendet dann eine Profilanfrage an die Anwendung oder den Hilfsmittelversorger, die mit einer Profilantwort antworten, in der gegebenenfalls die Hilfsmittel aufgelistet sind, die sie bereitstellen. Die Anwendung oder der Hilfsmittelversorger müssen nun auf ein Objekt Profiländerung warten. Während des Wartens auf die Profiländerung kann sie (er) weder Sitzungen mit anderen Hilfsmitteln erstellen, noch kann sie (er) Sitzungen von anderen Anwendungen akzeptieren, die mit 'Hilfsmittel existiert nicht' bzw. 'Hilfsmittel existiert aber nicht verfügbar' antworten.

Wenn der Hilfsmittelmanager auf allen Transportverbindungen nach Profilen gefragt und Profilantworten empfangen hat, stellt das Hauptgerät eine Liste der verfügbaren Hilfsmittel zusammen. Wenn zwei oder mehr Hilfsmittel in beiden Klassen und Typen übereinstimmen, bleibt das Hauptgerät bei dem mit der höchsten Versionsnummer in seiner Liste. Wo die Versionsnummern auch übereinstimmen, behält das Hauptgerät alle Hilfsmittel und wählt eines davon zufällig, wenn eine Anforderung 'Sitzung Erstellen' dafür empfangen wird. Wenn das Hauptgerät seine Hilfsmittelliste aufgestellt hat, sendet es ein Objekt Profiländerung an alle augenblicklichen Hilfsmittelmanager-Sitzungen, und die Anwendungen können dann unter Benutzung des Objektes Profiländerung beim Hauptgerät nach dessen Hilfsmittelliste anfragen.

Wenn sie (er) die Profiländerungs-Benachrichtigung das erste Mal empfängt, kann die Anwendung oder der Hilfsmittelversorger mit einer Profiländerung beim Hauptgerät anfragen und eine Profilantwort mit der Liste der verfügbaren Hilfsmittel des Hauptgerätes empfangen. Nach diesem ersten Einsatz des Profiländerungs-Protokolls ist die Anwendung oder der Hilfsmittelversorger frei, um weitere Sitzungen zu erstellen oder zu akzeptieren. Ihre (seine) Sitzung mit dem Hilfsmittelmanager besteht weiter, um von Zeit zu Zeit weitere Profiländerungs-Benachrichtigungen durch das Hauptgerät zu ermöglichen.

Wenn ein Hilfsmittelversorger eine Änderung im Profil der Hilfsmittel, die er liefert, mitteilen will, gibt er eine Profiländerung an das Hauptgerät. Das Hauptgerät antwortet mit einer Profilanfrage, auf die der Hilfsmittelversorger dann wieder mit ihrer aktualisierten Hilfsmittelliste antwortet. Das Hauptgerät verarbeitet diese Antworten. Wenn sich daraus irgendeine Änderung für die eigene Hilfsmittelliste des Hauptgerätes ergibt, gibt das Hauptgerät eine Profiländerung an alle aktiven Hilfsmittelmanager-Sitzungen. Die Anwendungen können dann eine aktualisierte Hilfsmittelliste anfordern und empfangen.