



# Fachprüfung: Hyper- und Multimedia

## Hypermedia- und Multimediasysteme (1,2)

04.10.2001

<b>Name</b> (bitte in Blockschrift)	
<b>Matrikelnummer</b>	
<b>Unterschrift</b>	

**Hinweise:**

- Überprüfen Sie Ihr Klausur-Exemplar bitte vor Beginn der Klausur auf Vollständigkeit.
- Bitte halten Sie Ihren Lichtbildausweis sowie den Studentenausweis zur Kontrolle bereit.
- Dauer der Klausur: 120 min.
- maximal erreichbare Punktzahl: 200
- Bitte füllen Sie das Deckblatt vollständig aus, beschriften jedes Blatt mit Ihrer Matrikelnummer und unterschreiben Sie dieses Klausur-Exemplar.
- Jedes Verlassen des Prüfungsraums muss ausdrücklich mit der Aufsicht vereinbart werden.
- Zugelassene Hilfsmittel sind ausschließlich Schreibutensilien, nicht-programmierbare Taschenrechner und das eigene(!) Gedächtnis.
- Bitte vermeiden Sie die Verwendung von roter Farbe.
- Die nach jeder Frage eingeklammerte Zahl ist die bei dieser Frage maximal erreichbare Punktzahl.
- Beachten Sie die in vielen Fragen enthaltenen Teilfragen!
- Falls der Platz für die Beantwortung einer Frage nicht ausreichen sollte, verwenden Sie bitte die Rückseite.
- Nutzen Sie im Falle von Unklarheiten hinsichtlich der Fragestellung die Möglichkeit zu Rückfragen!

**Viel Erfolg!**

Punkte Teil 1	
Punkte Teil 2	
Punkte gesamt	

1. Prüfer .....	
2. Prüfer .....	..... Note



**- I. Prüfungsfragen zur Veranstaltung im SS 2000 (HMS-1) -**

1. Mit welchen Zielsetzungen wurden die unterschiedlichen MPEG-Standards entwickelt? Mit welchen neuen Konzepten wurde dies jeweils technisch umgesetzt? (16)

- MPEG-1:  
Ziel: Speicherung und Wiedergabe von bewegten Bildern und Audio-Speichermedien (CD-I, DAB, ISDN=64 kbps) (++)  
Konzept: optimierte Kompression von Audio und Video(++)
- MPEG-2:  
Ziel: digitales Fernsehen (DVD, DVB) (++)  
Konzept: verschiedene Qualitäts- und Kompressionsstufen, Interlace (++)
- MPEG-4:  
Ziel: Multimedia-Anwendungen (++)  
Konzept: Interaktivität, unabhängige Medienobjekte, Kombination realer und synthetischer Objekte (++)
- MPEG-7:  
Ziel: Inhalts-basierte Informations-Suche, Archiv-Funktionalität (++)  
Konzept: Meta-Daten zur Inhaltsbeschreibung (++)

2. Wozu dient MP3 und wie ist es in die MPEG-Standards einzuordnen? (2)

- MP3: Audio-Codierung mit hoher Kompression (+)
- MPEG-1 Audio Layer 3 (+)



3. Erläutern Sie das Konzept von *Profiles* und *Levels* in MPEG-2. Was bedeutet MP@HL? Wo findet sich HDTV in diesem Konzept wieder? (12)

- MPEG-2 beinhaltet verschiedene AV-Protokolle unterschiedlicher Qualität, die sich durch Kombination je eines Profiles und eines Levels definieren. (++)
- Level gibt die Auflösung in Bildpunkten (+) und Frames/s (+) an.
- Profile gibt die Bandbreite bzw. Kompressionsrate an, sowie Farbsampling (4:2:2, etc.) und B-Frame-Rate. (++)
- MP@HL = Main Profile (<15 Mbps) (++) bei High Level (1920\*1152\*60) bzw. (1440\*1152\*60) (++)
- HDTV: High Level (1920\*1152\*60) (++)

4. Vergleichen Sie natürliche und synthetische Video-Codierung in MPEG-4. (10)

- natural Video Coding:
  - Realaufnahmen (Kamera) (+)
  - 5 kbps - 5 Mbps (++)
  - Basis: MPEG-1 und H263 (++)
  - [SP: robust gegen Störungen (+)]
- synthetic Video Coding:
  - Computer-generierte Videoströme (+)
  - 2D-Gitterstrukturen als Basis für Animation (++)
  - spezialisiert für Animation von Gesichtern und menschlichen Körpern (++)



5. Welche Merkmale nutzt MPEG-7 für die Beschreibung von Audio- und Video-Inhalten?  
(8)

- Audio:
  - Audio Objects, Harmony, Frequency Profile, Amplitude Envelope, Temporal Structure (Rhythm etc.), Textual content (++++)
- Video:
  - Visual Objects, Color, Texture, Shape, Volume, Spatial Relations (++++)

6. Beschreiben Sie das 3-Schichten-Modell für MPEG-4 Audio. (12)

- Demux/Decoder-Schicht → AudioBIFS-Schicht → BIFS-Schicht (+++)
- Demux/Decoder-Schicht:
  - Decoder (Para-Decoder, CELP-Decoder, T/F-Decoder) nehmen Multiplex-Ströme von Servern auf. (+++)
- AudioBIFS-Schicht:
  - Audio-Subgraph (Szene Graph): Audioquellen werden aufgeteilt (Switch), gemischt (Audio Mix) und/oder mit Effekten überlagert (Audio FX) (+++)
- BIFS-Schicht:
  - logische Verschaltung (Kombination) der Audio-Elemente mit weiteren BIFS-Elementen (z.B. Video) (+++)



7. Wie erlaubt MPEG-4 Interaktivität im Audio-Bereich? (4)

- Modifikation von Elementen (Feldern) des Szene-Graphen zur Laufzeit (++) über exposed fields. (++)

8. Beschreiben Sie das Prinzip der *Cut-Out Animation*. Worin bestehen hier die besonderen Vorteile der Digitalisierung? (6)

- *Cut-Out Animation*: Veränderung von Position und Orientierung bzw. Austausch von statischen 2D-Bildelementen (+++)
- Digitalisierung erleichtert Zwischenphasen-Automatisierung, Skalierung, Farbwechsel, ... durch Ersparnis manueller Erzeugung einzelner Bilder zwischen Keyframes. (+++)

9. Beschreiben Sie drei Typen der Anbindung von Datendiensten an Audioströme in DAB-Anwendungen? (6)

- Fast Information Channel (FIC): allgemeine Informationen über den Datenstrom (Basissignale, Multiplex-Konfiguration, Service-Information, etc.) (++)
- Programme Associated Data (PAD): programmbezogene Zusatz-Informationen, an Audio-Frames angebunden (begrenzte Kapazität: 8-16 kbit/s) (++)
- unabhängige Datendienste über Main Service Channel (MSC): z.B. Webseiten, Electronic Program Guide (EPG), etc. über Nutzung bis zur Gesamt-Kapazität des Übertragungskanals; auch Media Object Transfer (MOT) zur geordneten Kombination verschiedener multimedialer Daten, ggf. Datenkarussell (++)



10. Worin besteht das grundsätzliche Prinzip der Lauflängenkodierung? Auf welche Weise (und unter welchen Voraussetzungen) trägt diese zur Datenreduktion bei der JPEG-Bildkompression bei? Wie sähe eine solche Codierung konkret für die Zahlenfolge [ 0 2 2 2 2 5 1 1 ] aus? (9)

- Lauflängenkodierung: Angabe der Wiederholungshäufigkeit eines Zahlenwerts in Folge. (++)
- Bei häufiger Wiederholung einzelner Werte in Folge (+) werden jeweils nur 2 statt vieler Bytes pro Folge benötigt (+). Durch vorausgegangene Umstrukturierung der Bilddaten (DCT, Quantisierung, Differenzkodierung) enthalten JPG-Datenströme zum Zeitpunkt der Lauflängenkodierung lange Folgen von 0-Werten. (+++)
- [ 0 2 2 2 2 5 1 1 ] → [ 1 0 4 2 1 5 2 1 ] (++)

11. Welche Randbedingungen bzw. Sonderfälle (nennen Sie mindestens drei) müssen bei der JPEG-ähnlichen Bildkompressions-Implementierung in einem (C/C++ od. Java- od. Perl- etc.) Programm besonders berücksichtigt werden (unabhängig von der jeweiligen Programmiersprache)? Machen Sie jeweils Lösungsvorschläge. (15)

- Rundungsfehler (++) → Verwendung von double precision bei Zwischenrechnungen od. int zu float zu int bei RGB-Umwandlung in YUV (+++)
- Bildränder passen nicht zur 8x8-Matrix (DCT u.a.) (++) → Verwendung kleinerer Matritzen oder Auffüllen mit Nullwerten oder Anschluss von Bildkopien an den Rändern (+++)
- Chrominanz-Reduktion: Wahl der kombinierten Werte (++) → Mittelwert oder Auswahl eines der zusammen gezogenen Werte (+++)
- ...

– II. Prüfungsfragen zur Veranstaltung im SS 2001 (HMS-2) –

1. Beschreiben Sie den Grundaufbau von WAP-Anwendungen (mit Hilfe einer schematisch aufgebauten Darstellung einer WAP-Seite). (15)

WAP-Anwendungen werden mit Hilfe von WML, einem XML-Dialekt, erstellt. Der Grundaufbau einer WML-Seite sieht vereinfacht wie folgt aus:

```
<?xml ... ?>
<!DOCTYPE ... >
<wml>
  <head>...
    <meta>...</meta>
    ...
  </head>
  <card>
    <p> ... </p>
    <p> ... </p>
    ...
    <onevent> ... </onevent>
    ...
  </card>
  <card>
    ...
  </card>
  ...
</wml>
```

Im Abschnitt „<head>“ sind analog zu HTML Metainformationen zu der geladenen WML-Seite enthalten. Eine WML-Seite kann mehrere Anzeigeseiten („<card>“) haben, innerhalb derer das Element Absatz „<p>“ Text, Hyperlinks, Tabellen oder Eingabefelder enthält. Mit Hilfe von Eventhandlern „<onevent>“ lässt sich auf Benutzereingaben, z.B. vom Vorwärts- oder Rückwärtsblättern, reagieren. Die Syntax für Tabellen, Texte und andere Elemente ist HTML sehr ähnlich, aber im Befehlsumfang deutlich eingeschränkt.

2. Wie lassen sich innerhalb von JavaScript eigene Objekte erstellen? Erstellen Sie ein Beispiel. (10)

Beispiel:

```
function Farbe(Farbwert_R, Farbwert_G, Farbwert_B)
{
  this.Farbwert_R = Farbwert_R;
  this.Farbwert_G = Farbwert_G;
  this.Farbwert_B = Farbwert_B;
}

function TestObjekt()
{
  Test = new Farbe("33","99","C0");
  alert("Der Rotwert meiner Farbe ist Hexadezimal " +
  Test.Farbwert_R);
}
```

3. Wie lassen sich innerhalb von JavaScript globale Variablen deklarieren? Wo sind deren Grenzen? (10)

Globale Variablen werden außerhalb von Funktionen deklariert. Sie gelten nur innerhalb der Seite, in der sie in einem Scriptabschnitt deklariert sind, d.h. z.B. für benachbarte Frameseiten sind sie nicht unmittelbar gültig.

4. Was ist der Unterschied zwischen „innerText“ und „innerHTML“? (5)

Mit „innerText“ werden nur rein textuelle Elementinhalte ermittelt oder gesetzt, über „innerHTML“ zusätzlich alle innerhalb des betreffenden Tags ggf. weiter enthaltenen Elemente und deren Attribute.





5. Was verstehen Sie im Kontext der Webtechnologie unter „DOM“? Welche Bestandteile gehören zum DOM? (5)

DOM ist die Abkürzung für Dokumenten-Objekt-Modell, das eine HTML-Seite als in Objekthierarchien angeordnete Elemente versteht. Es umfasst alle herkömmlichen Tags sowie Mengen und Aufzählungstypen, sogenannte Collections.

6. Nennen Sie 5 Beispiele für Collections in DHTML. (5)

all; anchors; applets; areas; cells; children; elements; embeds; filters; forms; frames; images; imports; links; options; plugins; rows; rules; scripts; styleSheets; tbody

7. Nennen Sie typische Bestandteile von DHTML und deren Funktion. (10)

Das Dokumentenobjektmodell (DOM) beschreibt den hierarchischen Aufbau der Elemente einer HTML-Seite, auf die (und deren Eigenschaften und Methoden) mit Hilfe von Scriptsprachen (dynamic content) wie JavaScript und Cascading Style Sheets (CSS) (dynamic styles) zur Laufzeit lesend und schreibend zugegriffen werden kann. Events schließlich dienen der Möglichkeit, auf Benutzeraktionen (button click, scrolling, ...) mit Hilfe des Scriptings reagieren zu können.

8. Wann setzen Sie Elemente und Attribute in XML ein? (5)

Metainformationen werden gewöhnlich als Attribute, Inhalte als Elemente eingesetzt. Grundsätzlich sind jedoch beide Varianten mit kleinen Einschränkungen möglich (Textlänge und Sonderzeichenbehandlung bei Attributen...).



9. Nennen Sie die Standardobjekte in ASP und deren Funktion. (10)

- Application : Verwaltung von Zuständen in einer Multiuserumgebung
- ObjectContext : Zusammenarbeit mit dem MTS
- Request : Auswertung vom Client übermittelter Angaben
- Response : Senden von Ausgaben an den Client
- Session : Verwaltung von Zuständen im Rahmen einer Benutzersitzung
- Server : Verwaltung des Webservers

10. Was ist der Unterschied zwischen einer synchronen und asynchronen Kommunikation, wo liegen jeweils typische Einsatzgebiete? (5)

- Bei einer synchronen Kommunikation wartet der Sender auf ein Ergebnis vom Empfänger zurück übermitteltes Ergebnis, bevor er eine Weiterverarbeitung von Daten vornimmt.
- Bei der asynchronen Kommunikation wartet der Sender nicht auf das Ergebnis.
- Anwendungsbeispiele für die synchrone Kommunikation sind z. B. Plausibilisierung von Benutzereingaben, alle Vorgänge, bei denen der Anwender eine unmittelbare Reaktion des angestoßenen Kommunikationsprozesse erwartet. Asynchrone Prozesse werden z. B. dort initiiert, wo größere Datenmengen im Hintergrund kommuniziert werden sollen, der Anwender aber dennoch flüssig weiterarbeiten können soll.

11. Wie lassen sich XML-Dokumente weiterverarbeiten? (5)

Über direkte Methoden und Eigenschaften des Parsers oder via Transformation mittels XSLT in modifizierte XML-Dokumente, HTML-Seiten, Textdateien oder Datenbankinhalte.



12. Was sind „DataIslands“, wie und in welchem Kontext werden sie verwendet? Wie kann auf „DataIslands“ zugegriffen werden? (15)

Mit Hilfe von DataIslands können XML-Daten direkt innerhalb einer HTML-Seite enthalten sein. Der Vorteil ist dabei, dass die Daten dem Client zur ständigen Verfügung stehen, sobald die betreffende Seite im Browser geladen ist. Eine mögliche Anwendung ist zum Beispiel das Verwalten von Tabelleninhalten, die auf Anforderung hin umsortiert werden sollen ohne dass die Seite oder die Informationen in der Tabelle erneut vom Server geladen werden sollen.

Eingebunden werden können DataIslands u.a. wie folgt:

```
<XML ID="XMLID">
  <XMLDATA>
    <DATA>TEXT</DATA>
  </XMLDATA>
</XML>
```

oder

```
<XML SRC="http://localhost/xmlFile.xml"></XML>
```

oder

```
<SCRIPT LANGUAGE="XML" SRC="http://localhost/xmlFile.xml"></SCRIPT>
```

oder

```
<SCRIPT ID="XMLID" LANGUAGE="XML">
  <XMLDATA>
    <DATA>TEXT</DATA>
  </XMLDATA>
</SCRIPT>
```

Der Zugriff geschieht dann ganz normal mit JavaScript, z.B.:

```
function returnXMLData()
{ return document.all("XMLID").XMLDocument.nodeValue; }
```

```
function returnXMLData()
{ return XMLID.documentElement.text; }
```

Punkte

**Viel Erfolg!**