

## **Seminar Multimediale Informationssysteme**

# **Hypermedia: Modelle und Anwendungen**

Andreas Kämpel

## Gliederung

- 1 Begriffserklärungen
- 2 Hypermedia-Modelle
  - 2.1 Das Dexter-Hypertext-Modell
  - 2.2 Das Amsterdam-Hypermedia-Modell
  - 2.3 HyTime
  - 2.4 MHEG
  - 2.5 ZYX
- Hypermedia-Anwendungen
  - 3.1 Cardio-OP
  - 3.2 GLASS

### Begriffserklärungen

#### ***Diskrete Medien:***

- zeitunabhängig
- Text
- Grafiken

#### ***Kontinuierliche Medien:***

- zeitabhängig
- Audio
- Video

#### ***Hypertext:***

- Präsentation von diskreten Informationen als lineares Netzwerk
- freie, nicht lineare Navigation mittels Hyperlinks auf andere Dokumente

#### ***Hypermedia:***

- Erweiterung des Hypertext
- Präsentation von Kontinuierlichen Medien

# Hypermedia-Modelle

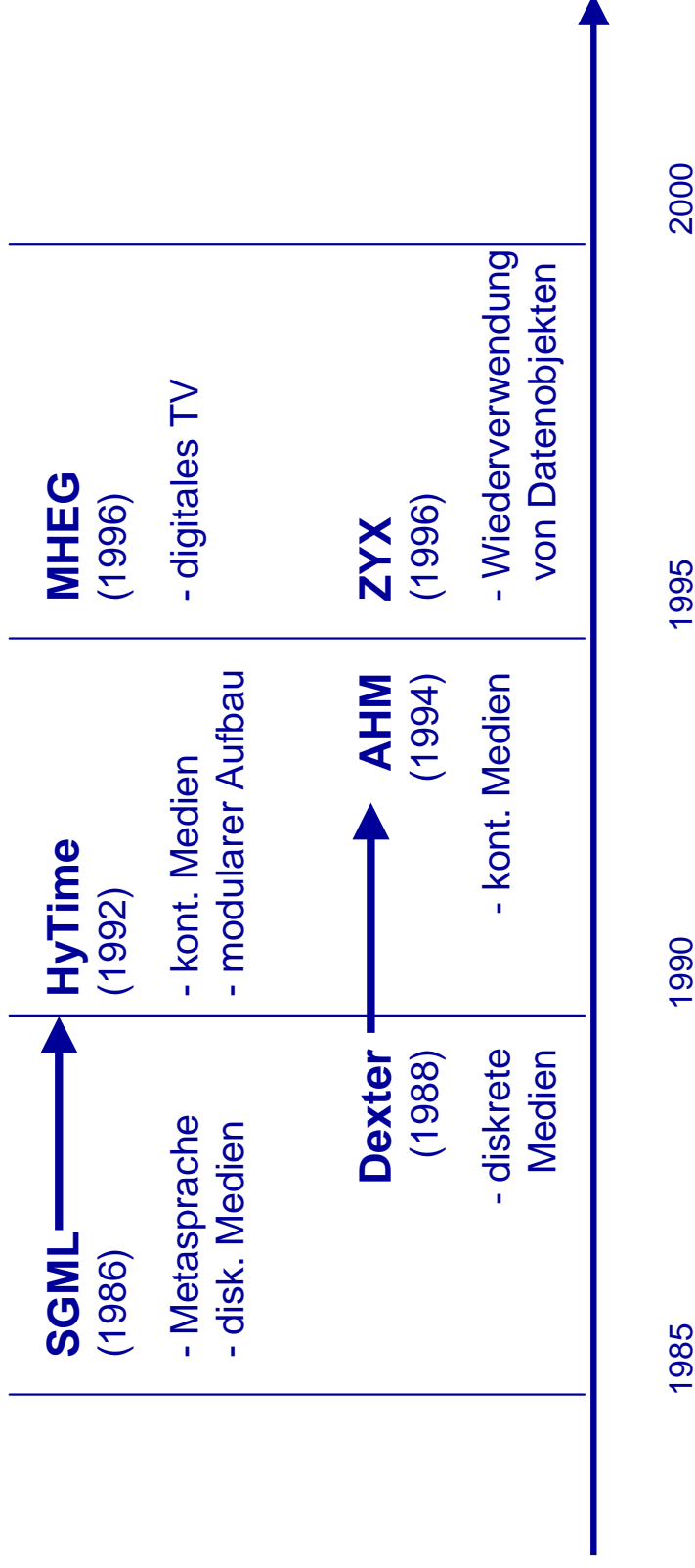
### Was sind Hypertext bzw. Hypermedia-Modelle?

- Modell für **Struktur der Daten** und der Anwendung
- *Kein* Modell für Entwicklungsprozess (SE)

#### **Ziele:**

- **Trennung von Informationsgehalt und Darstellung**
- **Standardisierung**
- **einfache Portierung** von Hypermedia-Anwendungen

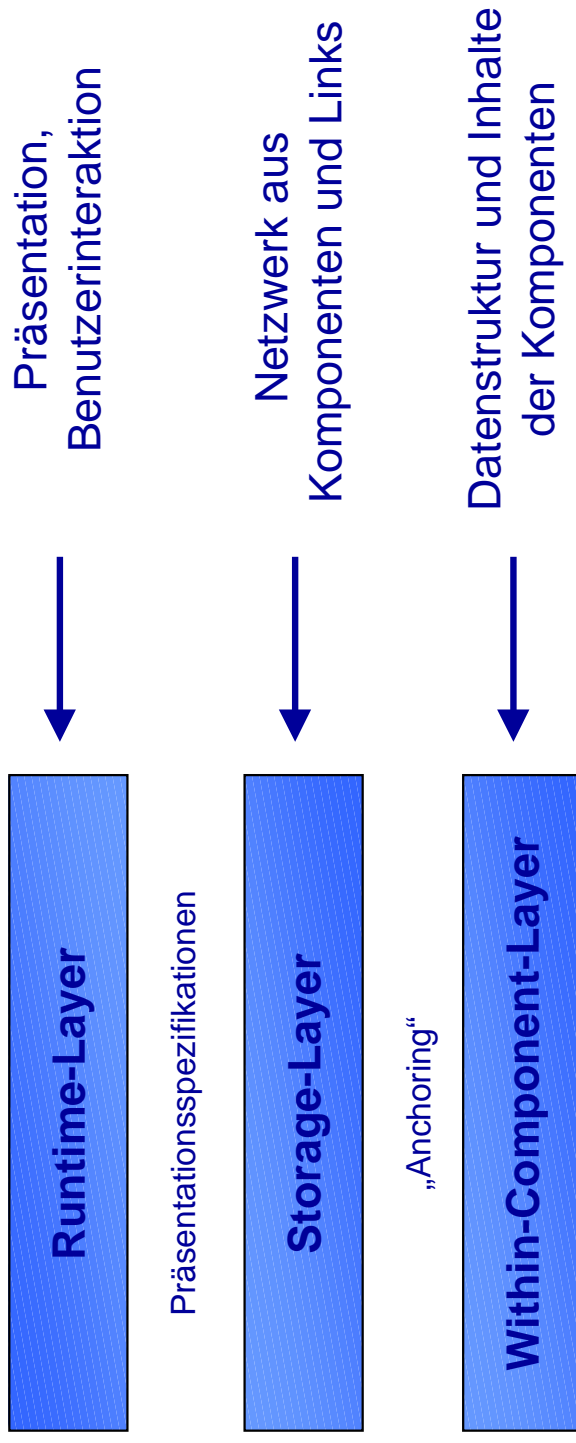
## Entwicklung von Hypertext- und Hypermedia-Modellen



### Dexter-Hypertext-Modell

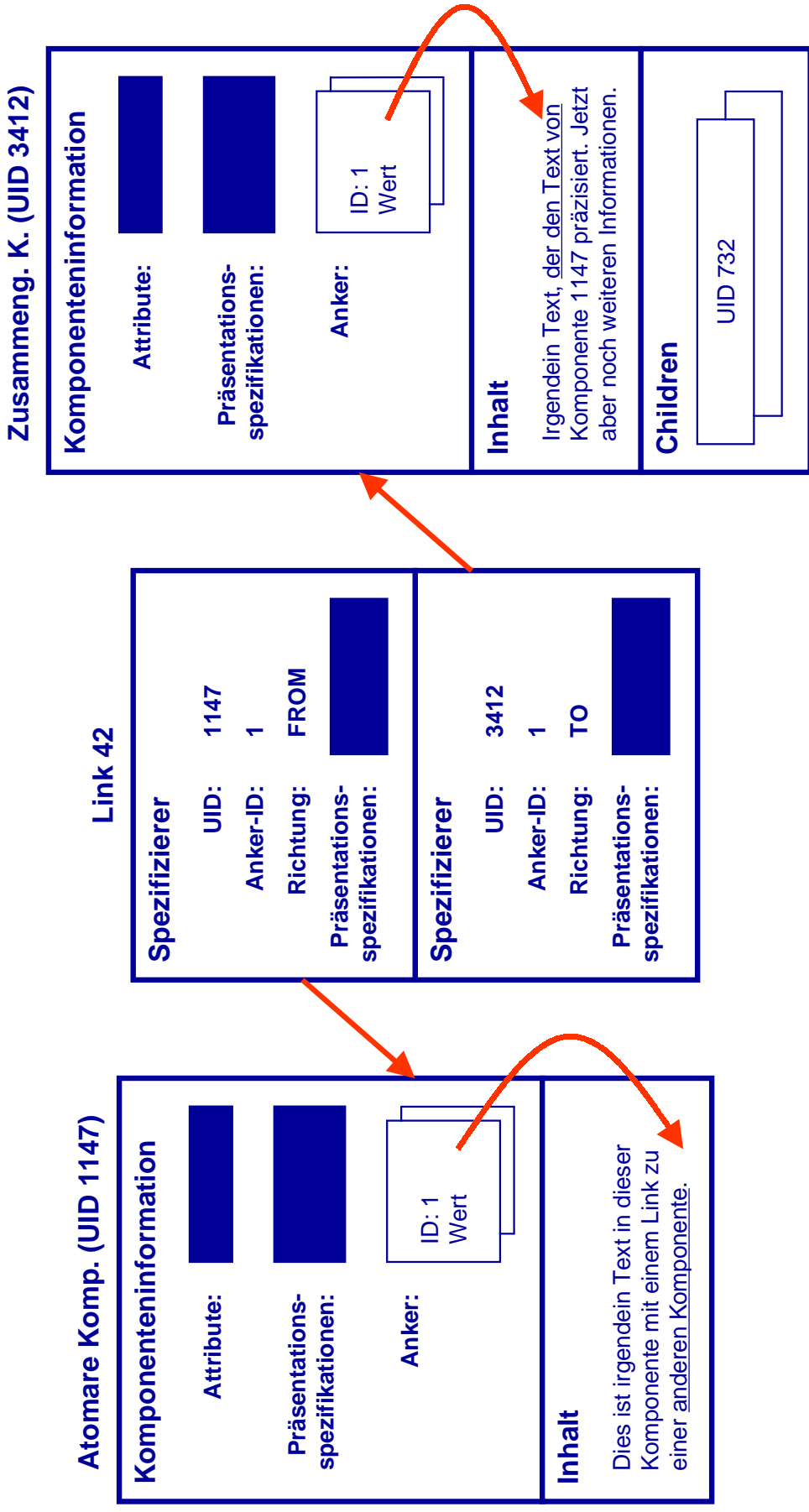
- 1988 von einer Gruppe von Hypertextsystem-Entwicklern geschaffen unter der Leitung von John Legget und Jan Walker
- Ziel: **Abstrakte Terminologien und Definitionen** anwendbar auf alle bestehenden Hypertextsysteme
- Zweck: Vergleich von bestehenden **Hypertextsystemen**
- dient als **Referenz-Modell**
- unterstützt nur **diskrete Medien**

## Dexter-Hypertext-Modell - Schichten





## Dexter-Hypertext-Modell – „Storage Layer“



## Dexter-Hypertext-Modell runtime- und within-component Layer

### runtime Layer

- **Darstellung** der gespeicherten Daten
- Arbeiten mit Instanzen (Kopien) der Daten
- **Interaktion** mit dem Benutzer
- Interaktionsmöglichkeiten nicht genauer spezifiziert

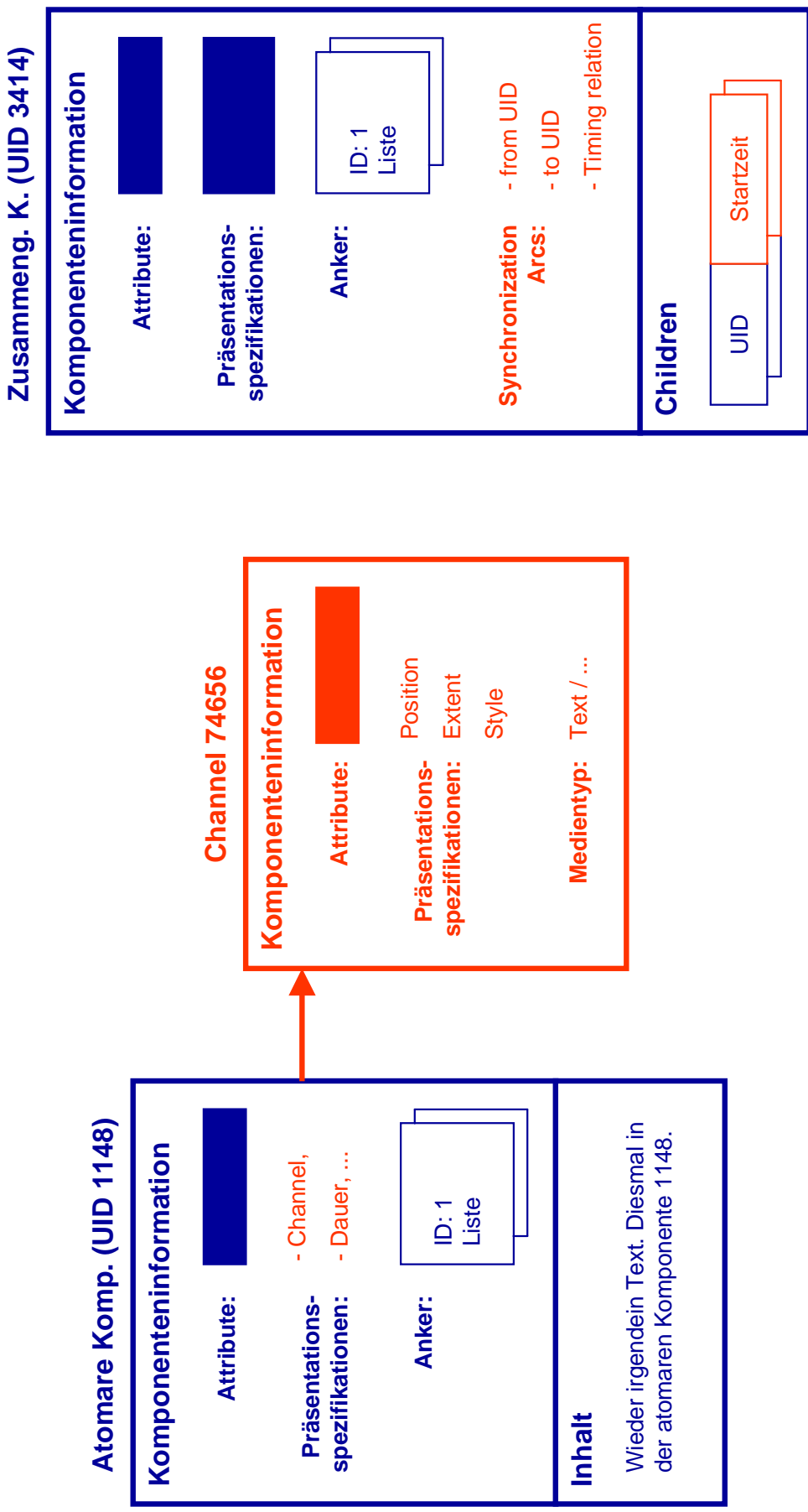
### within-component Layer

- definiert die **Struktur der Datenobjekte** des *storage Layers*
- beinhaltet einzelne Datenobjekte
- Schicht wird nur rudimentär beschrieben

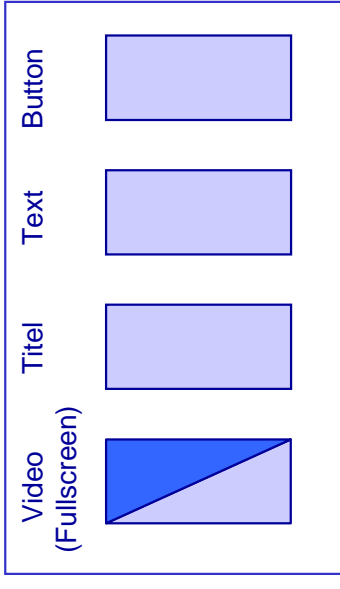
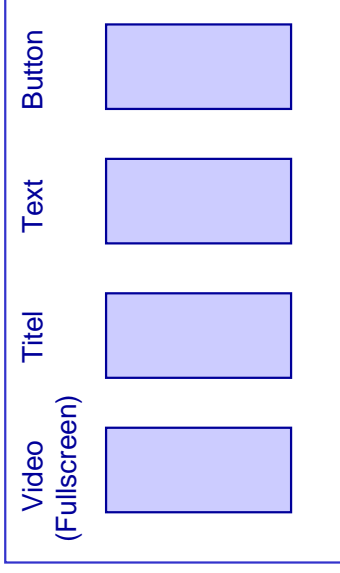
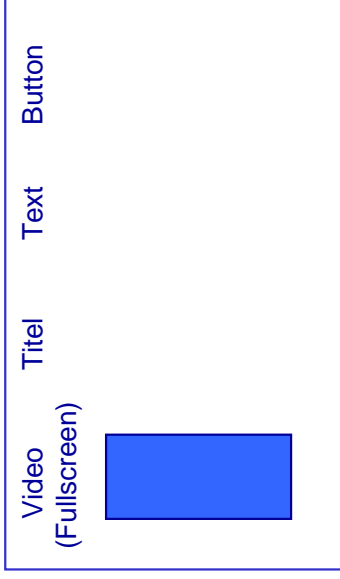
### Amsterdam-Hypermedia-Modell

- Erweiterung des Dexter-Modells um **kontinuierliche Medien** (Audio, Video) → „**Hypermedia-Modell**“
- mögliche Einbindung **zeitlicher Abhängigkeiten**
- **Kontextinformationen** für Anker möglich
- Entfernung des Dateninhalts aus zusammengesetzten Komponenten zwecks besserer Strukturierung
- gemeinsame Präsentationsattribute für Gruppen von Komponenten → **Abstrakte Ausgabegeräte „Channels“**

## Amsterdam-Hypermedia-Modell (2)



## Amsterdam-Hypermedia-Modell – Ressourcen-Konflikte



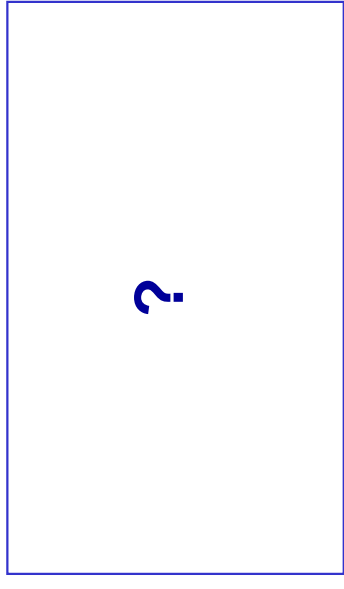
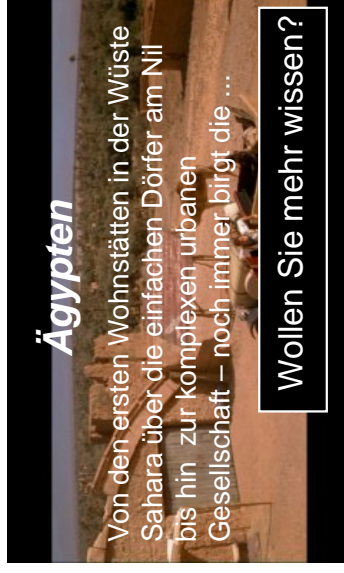
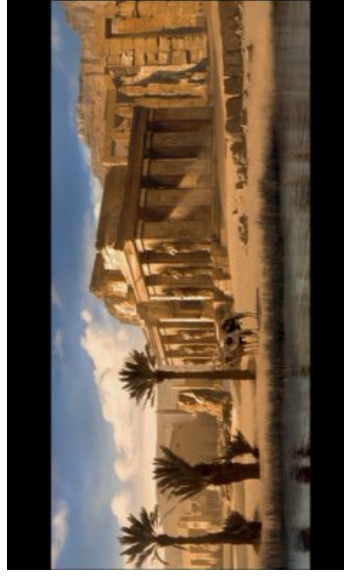
Komponente 1

+

Komponente 2



Neue Komponente



## HyTime – Rückblick: SGML

- Structured Generalized Markup Language
- Trennung von Inhalt, Struktur und Layout
- Kennzeichnung von Abschnitten durch Tags
- Portable Form der Definition und des Austauschs von Dokumenten

### Beispiel DTD / SGML-Dokument:

```
<!DOCTYPE seminar [
<!ELEMENT seminar          - -      (kopf, text+)
<!ELEMENT kopf             - -      (thema, person)
<!ELEMENT thema           - -      (#PCDATA)
<!ELEMENT person          - -      (#PCDATA)
<!ELEMENT text            - o      (#PCDATA)
]>
<!DOCTYPE seminar SYSTEM "seminar.dtd" >
<SEMINAR>
<KOPF><THEMA>Mein erstes Seminar</THEMA><PERSON>Ich selber</PERSON></KOPF>
<TEXT>Der erste Satz.
<TEXT>Der zweite Satz.
</SEMINAR>
```

### HyTime

- Hypermedia / Time-based Structuring Language
- ISO/IEC International Standard 10744 (1994) zur Strukturierung von Multimedia-Dokumenten
  - baut auf **SGML** auf
  - ging aus der SGML-Erweiterung SMDL (*Standard Music Description Language*) hervor, einer Sprache zur Beschreibung zeitlicher und synchroner Abläufe
  - neue Elementtypen: „**Architectural Forms**“ (HyTime-Element)
  - modularer Aufbau der HyTime-Elemente
- Zeitabhängigkeiten durch Positionierung von Objekten im mehrdimensionalen Raum

### HyTime - Module

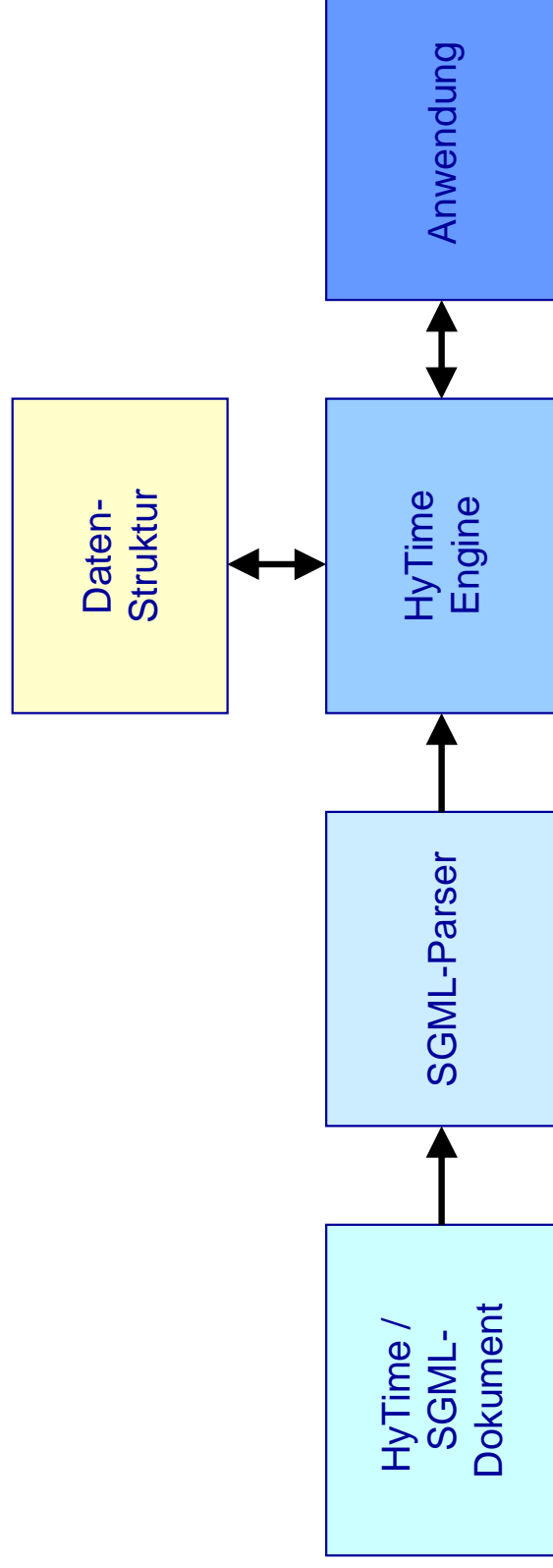
- Base Module*
- bietet Grundfunktionalität:
  - Verwaltung von Ids
  - Activity-Reporting-Facilities
- (Logging bei verschiedenen Aktionen auf Dokumente)
- Location Address Module*
- bietet drei verschiedene Adressierungsarten von Objekten (für Verlinken auf Dokumente):
1. Namensraumadressierung  
(übernommen aus SGML)
  2. Koordinatenadressierung  
(mehrdimensionale Position)
  3. Semantische Adressierung  
(HyTime Query Language)



### HyTime – Module (2)

- Hyperlink Module*
- zur Adressierung wird das Location Address Module verwendet
  - stellt mehrere verschiedene Arten von Hyperlinks zur Verfügung
- Measurement Module*
- zuständig für die Ablaufsteuerung der Darstellung der Objekte (beinhaltet Scheduling- und Rendition Module)

## HyTime – Präsentation von Dokumenten



## MHEG

- Multimedia and Hypermedia information coding expert group (1996) (ISO 13522)

- Beschreibung von statischen, zeitlichen und räumlichen Beziehungen zwischen Medienobjekten

- Ziel: Erstellung portabler und hardware-unabhängiger **Multimedia-Anwendungen**

- Klassenhierarchie, Erstellung von Objekten durch Autorensysteme

- Eventssteuerung durch Timer, Benutzereingaben, ...

- MHEG-5 entwickelt für „*minimal resource systems*“ (Set-Top Boxen) → digitales Fernsehen

- „Final Form“ von MHEG-Objekten



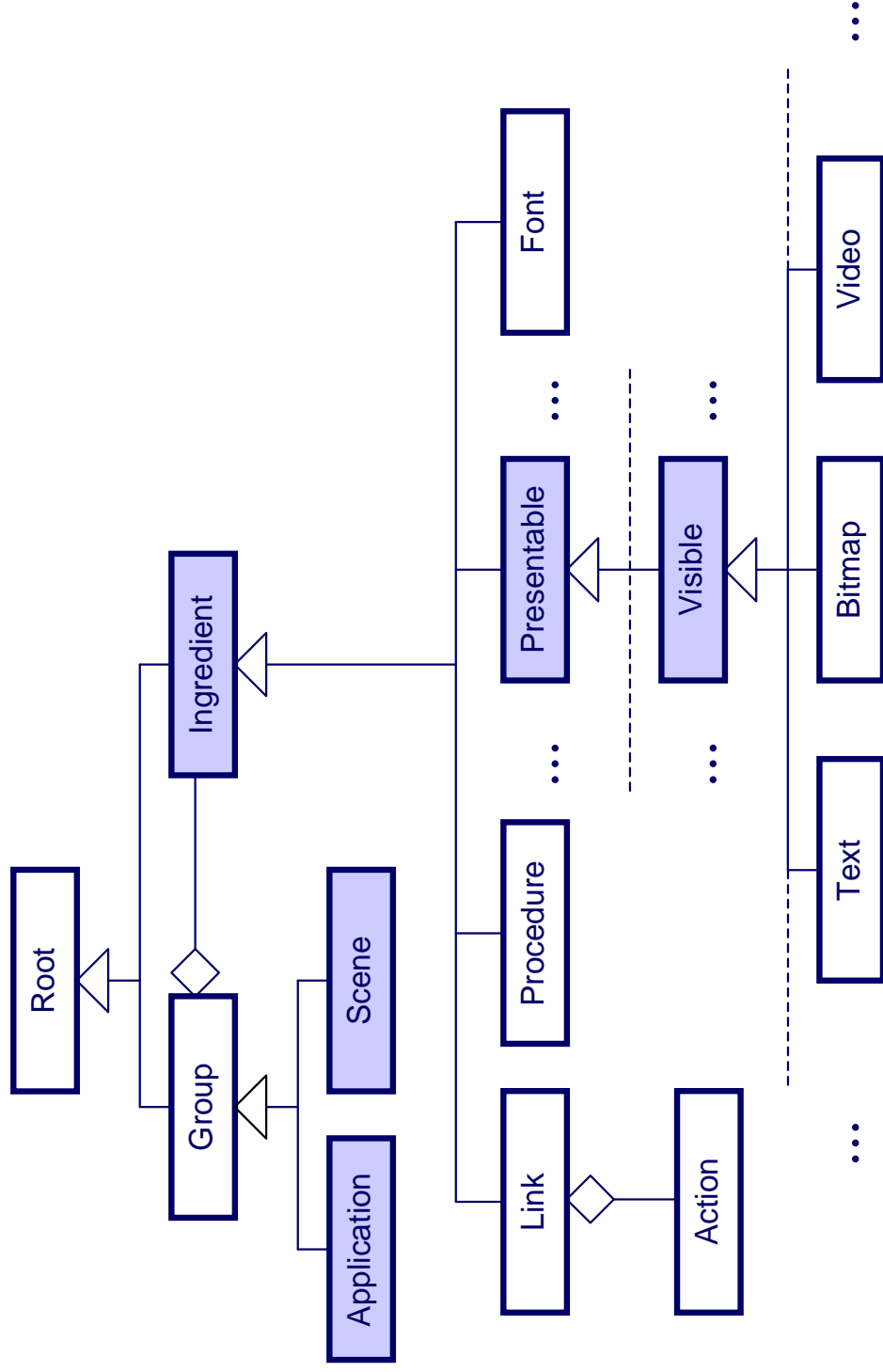
## MHEG – Stufen in MHEG

- MHEG-1 - Definiert Klassenhierarchie und Kodierung (ASN.1) von MHEG-Klassen
- MHEG-2 - geplant: alternative Kodierung der Objekte mit SGML (*wurde nicht fertiggestellt*)
- MHEG-3 - Unterstützung einer Scriptsprache:  
Definition einer „Virtual Machine“ mit Möglichkeit zur Definition von Variablen und arithmetischen / logischen Operationen

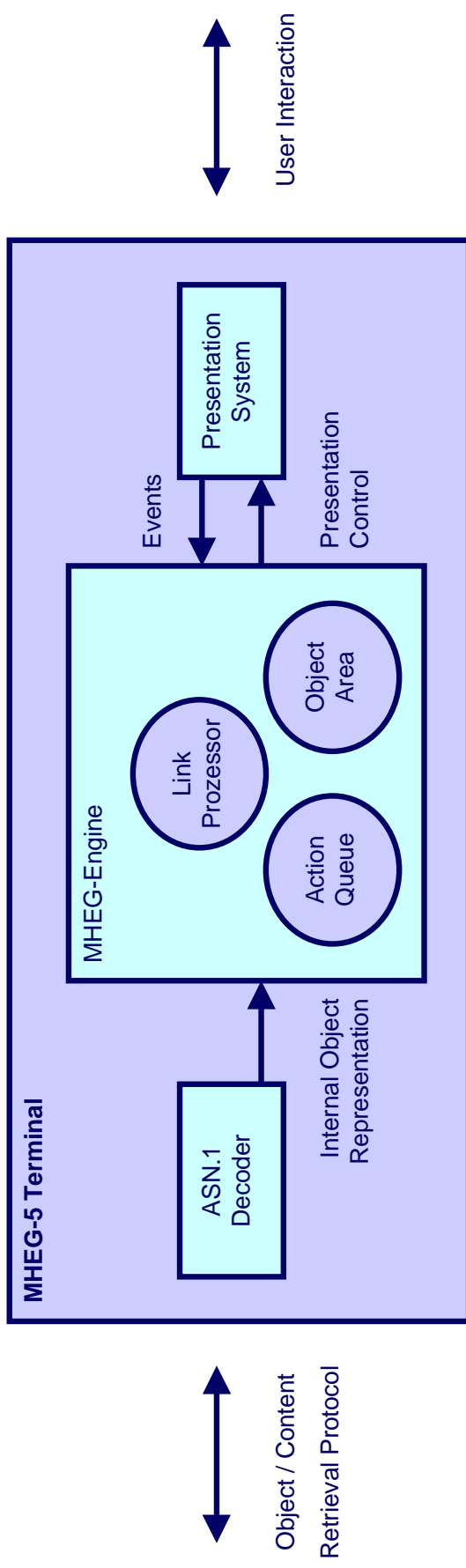
### MHEG – Stufen in MHEG (2)

- MHEG-5
  - Überarbeitung der Klassenstruktur, zugeschnitten auf „minimal resource systems“, Einführung plattformabhängiger Interface-Komponenten
  
- MHEG-6
  - Einbettung von Java-Programmcodes möglich,
  - Java Virtual Machine
  - Class-Mapper

## MHEG – Klassenstruktur MHEG-5 (Ausschnitt)



## MHEG-5 – Terminal-Struktur



### ZYX

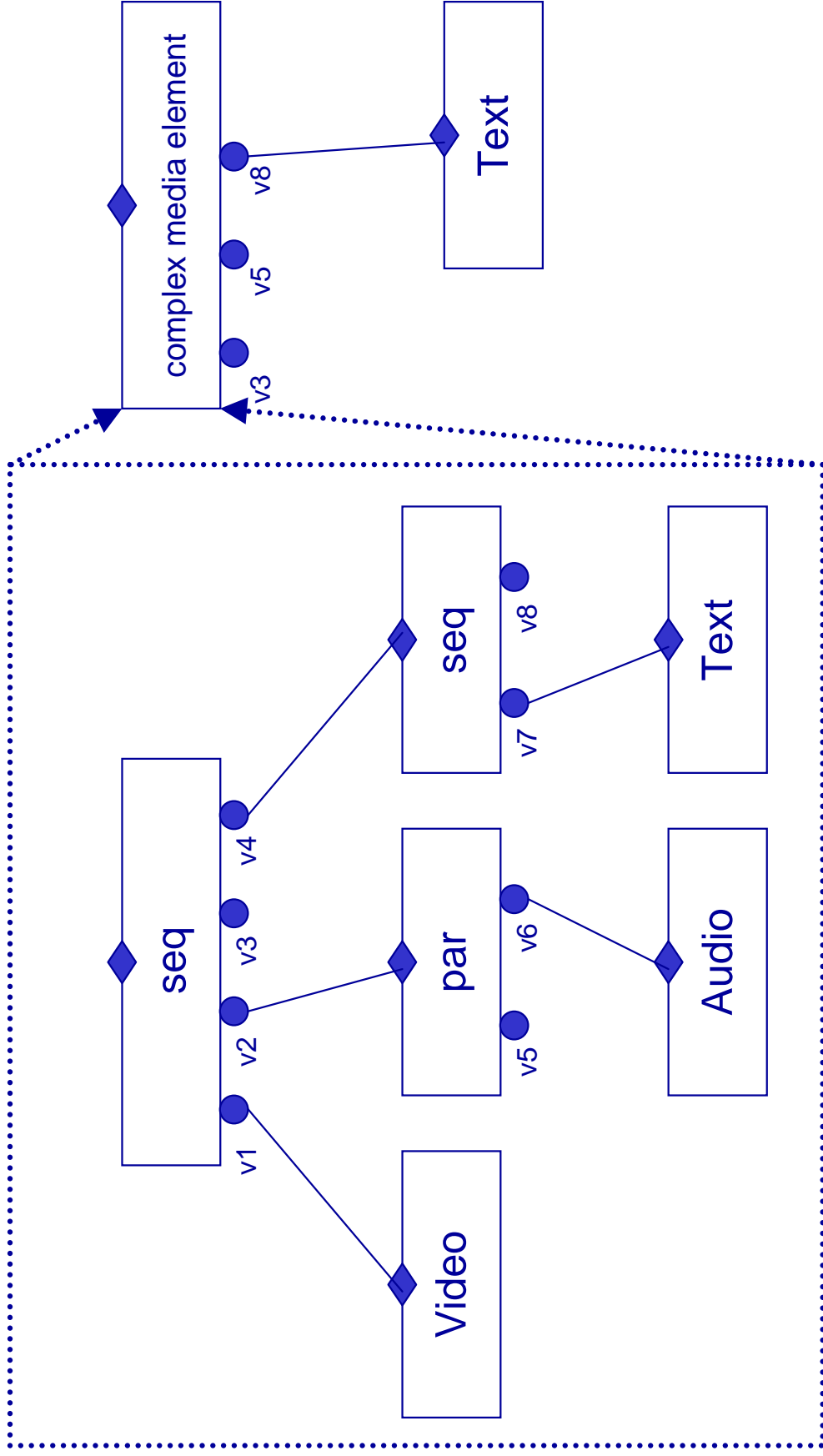
- speziell entwickelt für computer-basiertes Ausbildungssystem  
in der Medizin (1996)

### Spezielle Anforderungen:

- multiple Wiederverwendung von Multimedia-Daten
- Wiederverwendung von zeitlichen Abhängigkeiten und Interaktionsmöglichkeiten
- Benutzerspezifische Charakteristika erforderlich
- Darstellung auf heterogenen Software- und Hardwareumgebungen



## ZYX - Elemente

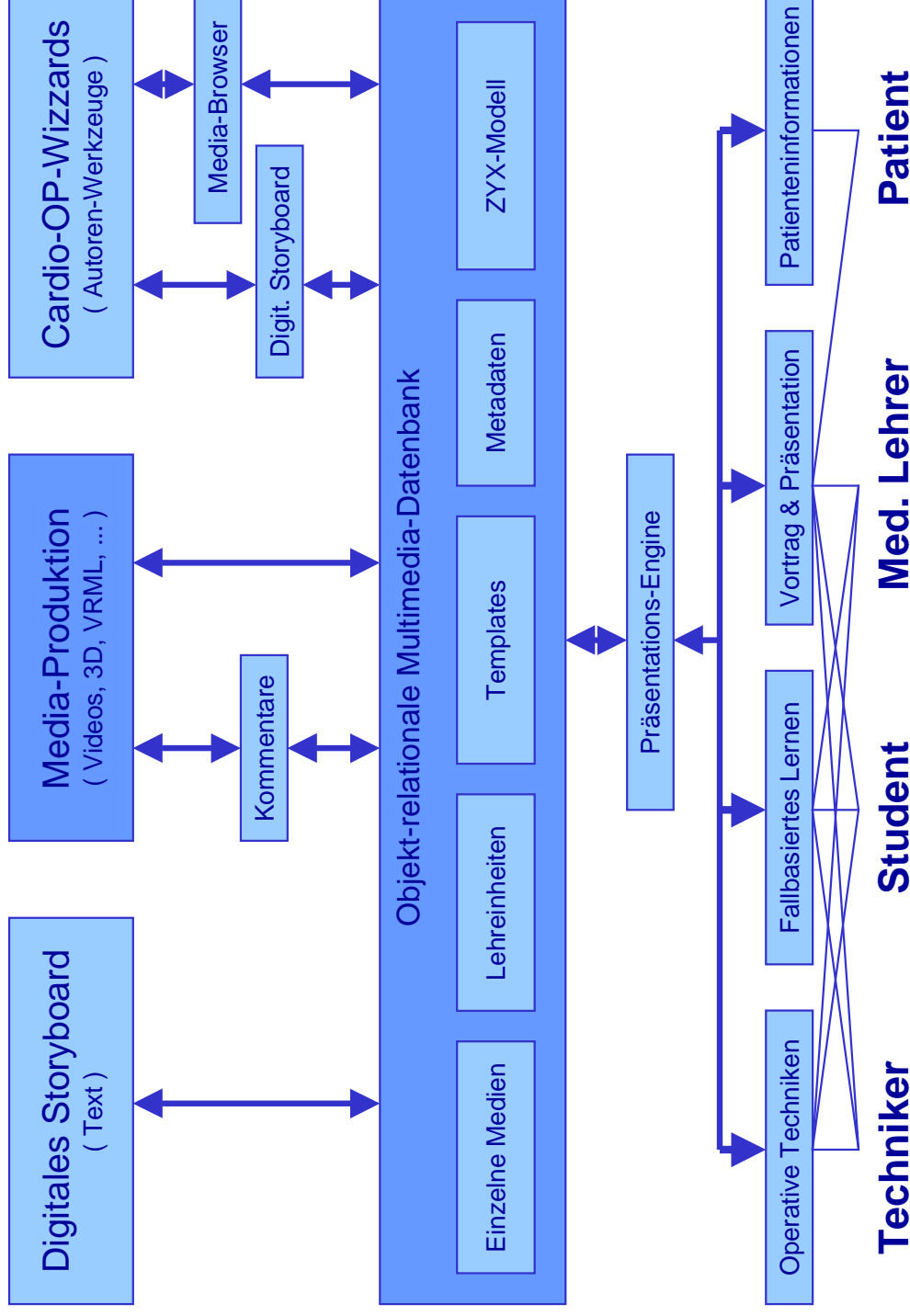


# Hypermedia-Anwendungen

### Cardio-OP-Projekt

- entwickelt von Mitarbeitern der Uni Ulm und Heidelberg
- Entwicklung eines **computer-basierten Ausbildungssystems** in der Herzchirurgie
- **multiple Wiederverwendung von Multimedia-Daten** in unterschiedlichen Ausbildungsszenarien
- **flexible Komposition** von Inhalten für verschiedene Benutzergruppen
- Entwicklung eines **Media Integration (MIB) DataBlade Module** für Informix zum Verwalten der Unterschiedlichen Medientypen als **Zwischenschicht** zwischen Anwendung und existierenden Modulen

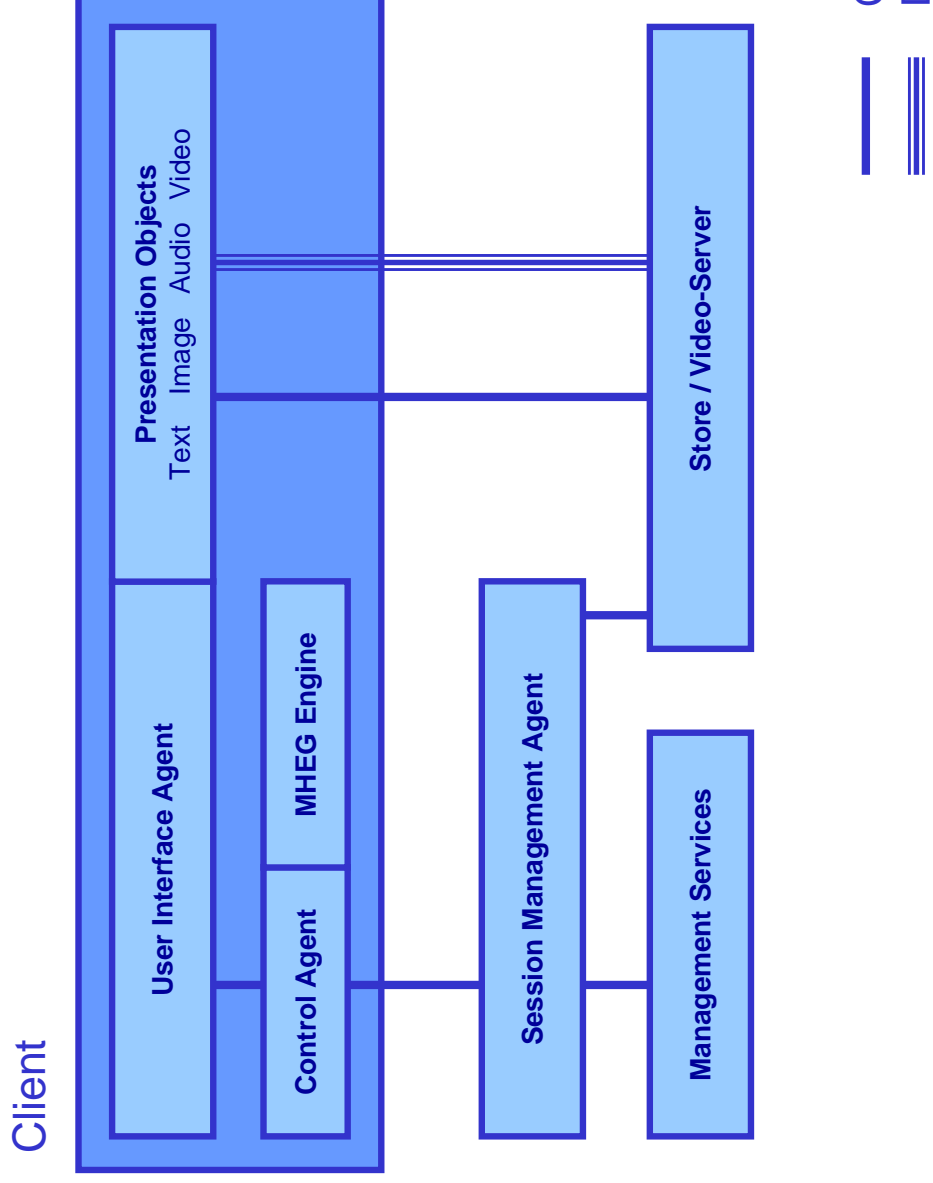
## Cardio-OP-Projekt - Produktionsprozess



### GLASS

- Globally Accessible Services System
- Prototyp für das digitale Fernsehen basierend auf dem MHEG-Standard
- Komponenten: Clients, Application Server, Video-Server, System Management, Gateways zu WWW, eMail, Fax, ...
- Lauffähig auf unterschiedlichen Hardware- und Softwaresystemen (DEC Alpha, Intel 80x86, Sun Sparc; Linux, MacOS, Windows, Solaris, OS2)

## GLASS - Aufbau



**Ende**