

**Seminar der Arbeitsgruppe  
Datenbanken und Informationssysteme**

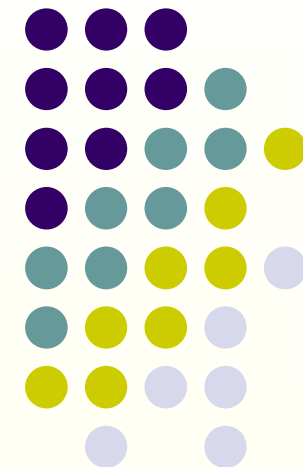


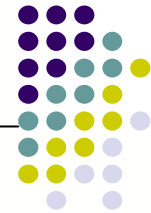
**Thema: Multimediale Informationssysteme**

---

# **Multimedia Authoring: Autorensysteme**

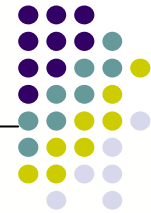
**Steffen Apfel**





## Gliederung des Vortrages

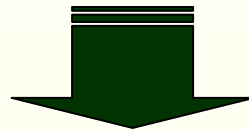
- 1. Einführung in das Themengebiet
- 2. Multimedia
- 3. Anwendungsgebiete von Autorensystemen
- 4. Autorensysteme
- 5. Zusammenfassung & Ausblick



Multimediale Dokumente (Anwendungen)  
bekommen eine immer wichtiger werdende Rolle

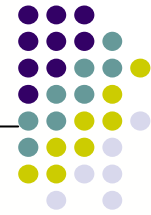
Multimediales Dokument:

- Interaktive Programme
- Linkstrukturen
- Integration von Audio, Video, Texte, Bilder und Animationen
- Übersichtliche Informationsdarbietung



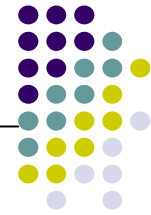
Wie kann man solche multimediale  
Dokumente erstellen?

**AUTORENSYSTEME**



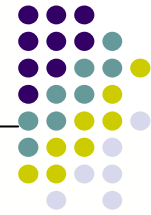
## 2. Multimedia

- 2.1 Geschichtlicher Abriss
- 2.2 Medientypen
- 2.3 Definition



### Geschichtlicher Abriss

- 1945 Vannevar Bush: MEMEX (Datenverwaltungssystem)
- 1963 Doug Engelbart: Erfindung der Maus
- 1973 Alan Key: GUI ( graphical user interface)
- ~1973 der erste PC ( Alto) wurde ans LAN angeschlossen
- 1989 Tim Berners-Lee: Geburtsstunde des Internets (Woodstock of the Web)

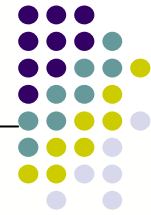


## Medientypen

Im multimedialen Bereich spielen 2 Medientypen eine dominierende Rolle:

- Diskrete Medien:
  - Zeitunabhängig, statisch
  - Repräsentation erfolgt einmalig und ist konstant
  - z. B.: Grafik, Text
  
- Kontinuierliche Medien
  - Zeitabhängig, dynamisch
  - Repräsentation über eine Zeitspanne hinweg
  - z. B.: Audio, Video, Animation

Nur durch die kombinierte Einbindung beider Medientypen erhält man ein Multimedia Dokument



### Definition

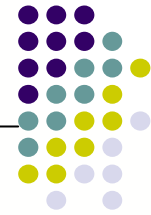
„Multi“ = mehrere, viele

„Medium“ = Überbringer, „eine Sache in der Mitte“

„Ein Multimedia System ist charakterisiert durch die integrierte, Computer-kontrollierte Behandlung von unabhängigen diskreten und kontinuierlichen Medien.“

Beispiele:

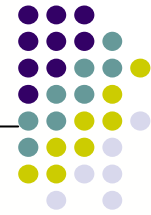
- ☹ Ein Schreibprogramm mit Texten und Grafiken (nur diskrete Medien)
- ☺ Ein Schreibprogramm zur Textverarbeitung mit verbaler Hilfefunktion (beide Medien und unabhängig)
- ☹ Eine Animation könnte man als eine Aneinanderreihung einzelner Bilder (diskret) und Ton (kontinuierlich) auffassen. (beide Medien, abhängig)



### 3. Anwendungsgebiete von AS

- 3.1 Kiosksysteme
- 3.2 CBT
- 3.3 WBT
- 3.4 Virtuelle Hochschule



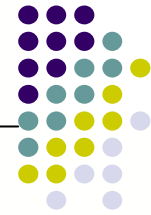


### Kiosksysteme

Kiosksysteme versuchen dem Benutzer in kurzer Zeit viel Information zu bieten.

- Analog zu einem Einkauf an einem Kioskstand
- Beispiele:
  - Informationsterminal in öffentlichen Gebäuden
  - Touristeninformation bei Sehenswürdigkeiten
  - E-Commerce Anwendungen im Internet
  - Lernprogramme für Kinder

„Listening is easier than Reading“



### Computer Based Training - CBT

Form des Computer-basierten Lernens für Schule, Arbeit oder privat

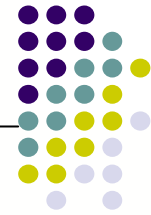
- Fast 30 Jahre alte Lernmethode
- Lernmaterial: CD, Diskette

+

- Zeit- und ortunabhängige Lernmethode
- Lernunterstützung durch Video, Audio, Animation und Simulation
- Kein Arbeitsausfall und hohe Kosten für Schulungen

-

- isolierte Arbeitstationen
- fehlende Kommunikation
- Keine individuelle Anpassung
- schlechte Updatemöglichkeiten



## Web Based Training - WBT

Form des Web-basierten Lernens für Schule, Arbeit oder privat

- Weiterentwicklung von CBT
- Lernmaterial im Internet erhältlich, evtl. zusätzlich CD

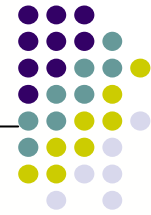
### Vorteile von CBT



- Leicht aktualisierbar, Daten zentral auf Server
- Individuelles Lernen durch Navigation
- Kommunikation:
  - Synchron ( Chat, Whiteboard, Videokonferenz)
  - Asynchron ( Email, Newsletter, Diskussionsforen)

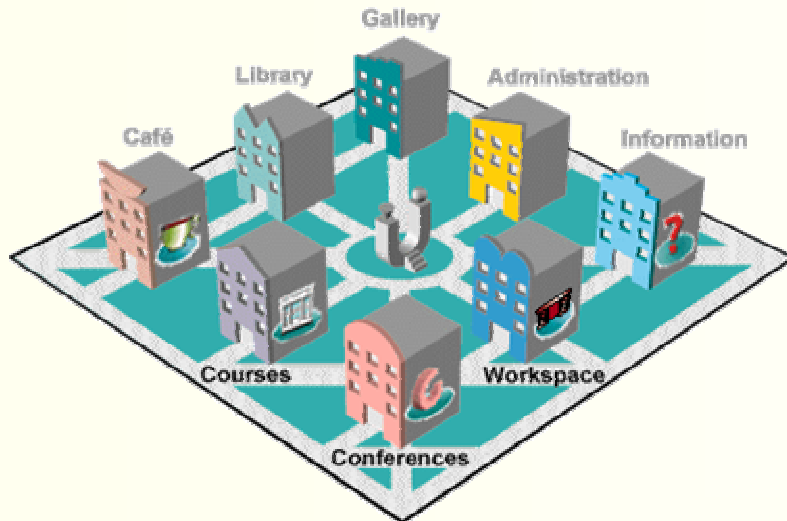
Computer Supported Collaborative Learning (CSCL)

-Kommunikation steht im Vordergrund



## Virtuelle Hochschule

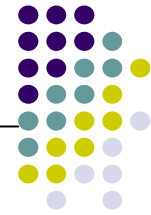
Einstiegseite einer virtuellen Uni:



### Probleme:

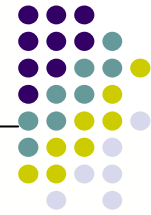
- Cafe: soziale Kontakte
- Bücherei: Urheberrecht von digitalen Büchern
- Klausuren: Sicherheitsaspekte

- (zeit-) und ortunabhängige Lernmethode
- Onlineübertragung der Vorlesungen
- Bildung von Kernkompetenzen
- Universtität übergreifende Lehre
- Vorteil für Schüler mit familiären, zeitlichen, finanziellen oder körperlichen Problemen
- Parallel zu normaler Lehre einsetzbar



## 4. Autorensysteme

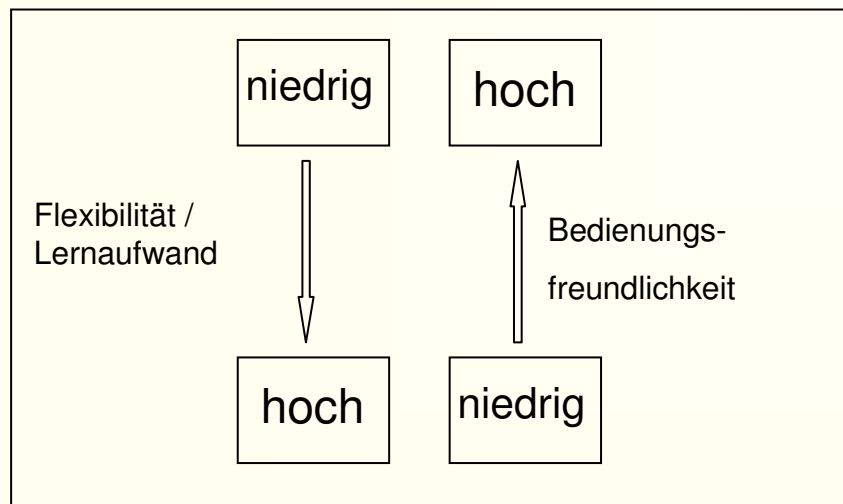
- 4.1 Definition
- 4.2 Autorensprachen
- 4.3 Klassifizierung der AS
  - 4.3.1 Frame-basierte AS
  - 4.3.2 Timeline-basierte AS
  - 4.3.3 Flowchart-basierte AS



## Definition

- AS bieten ein vorgefertigtes Gerüst zu Erstellung von MM-Anwendungen
- Scriptsprachen ergänzen Mächtigkeit
- Autoren sind keine Programmierer

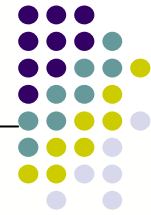
„AUTHORING“ here refers to „programming by non-programmers.“



Trade-Off zwischen Flexibilität und  
Bedienungsfreundlichkeit

### Def: Autorensysteme

- grafisch-interaktive Werkzeuge
- unterstützen den technischen Entwicklungsprozess MM-Dok
- benutzen visuelle Programmierung
- platzieren Medienobjekte und spezifizieren deren Beziehungen

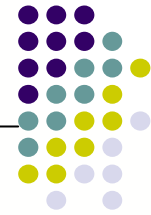


## Autorensprachen

### Codefragment von MTEACH:

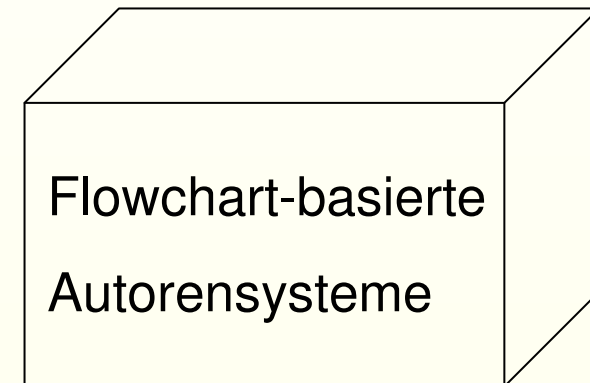
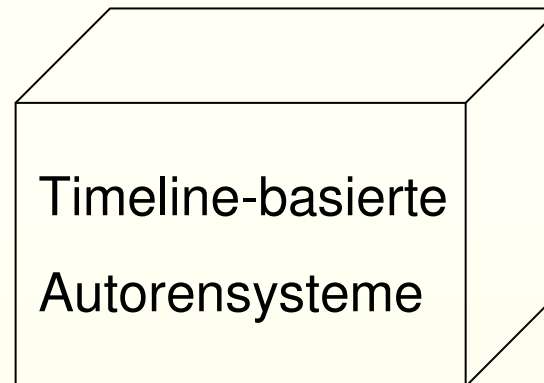
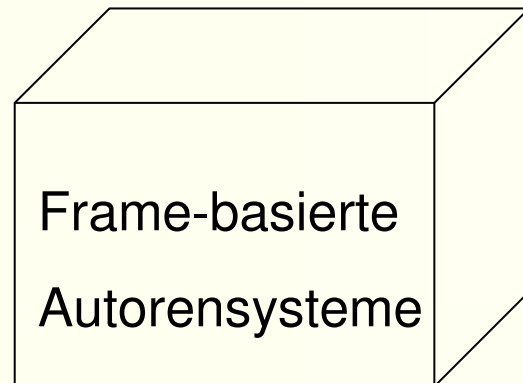
```
lesson "Ulnar nerve exposure at the elbow"
  ("Ulnar", "Exposure")
{
  argument "Sites of entrapment"
    ("Entrapment", "Elbow")
  {
    clip "Sites of entrapment" "sites.rm" ()
    {
      slide "sites.gif" () at 01
      slide "sites.gif"
        ("Anatomy of elbow") at 10
    }
  }
  argument "Surgical steps"
    ("Operating techniques")
  {
    clip "Skin incision" "incision.rm" ()
    {
      slide "skin_incision.gif" () at 01
    }
  }
  ...
}
```

- Zeitlich vor Autorensystemen
- 1961: Coursewriter
- Neuere Sprachen:  
MTEACH, SMIL
- Bessere Performance als bei AS:
  - Medien bleiben im Originalzustand
  - portables textuelles Dok.
- Mächtiger Sprachkonstrukte als normale Programmiersprachen
- Für erfahrene Programmierer

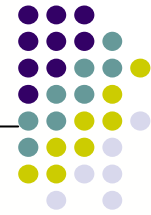


## Klassifizierung von Autorensystemen

- Aufgrund der unterschiedlichen Arbeitsweise werden Autorensysteme in 3 Klassen unterteilt
- Unterteilung erfolgt durch Metaphern, die die Arbeitsweise verdeutlichen
- Grenzen sind fließend
- Eindeutige Zuordnung nicht möglich



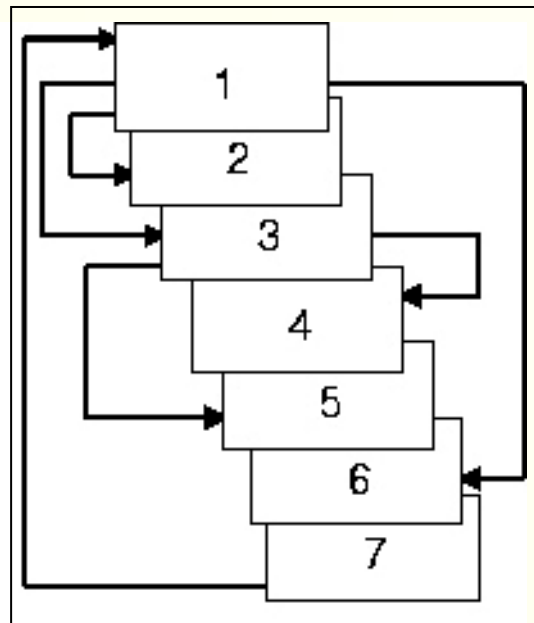




## Frame-basierte Autorensysteme

Zugrunde liegende Metapher:  
einzelne Frames werden spezifiziert und anschließend miteinander vernetzt.

Vernetzung von Frames:

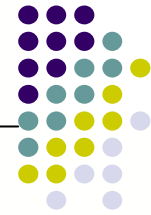


- Medienobjekte werden auf Frames (o.a. „Seiten“, „Karten“, „Fenster“) entsprechend ihrer späteren Lage platziert
- Jedes Frame repräsentiert einen Bildschirmdialog
- WYSIWYG-Prinzip
- Seitenwechsel durch Navigation
- Beispiele:

MetaCard, ToolBook, PictureBook

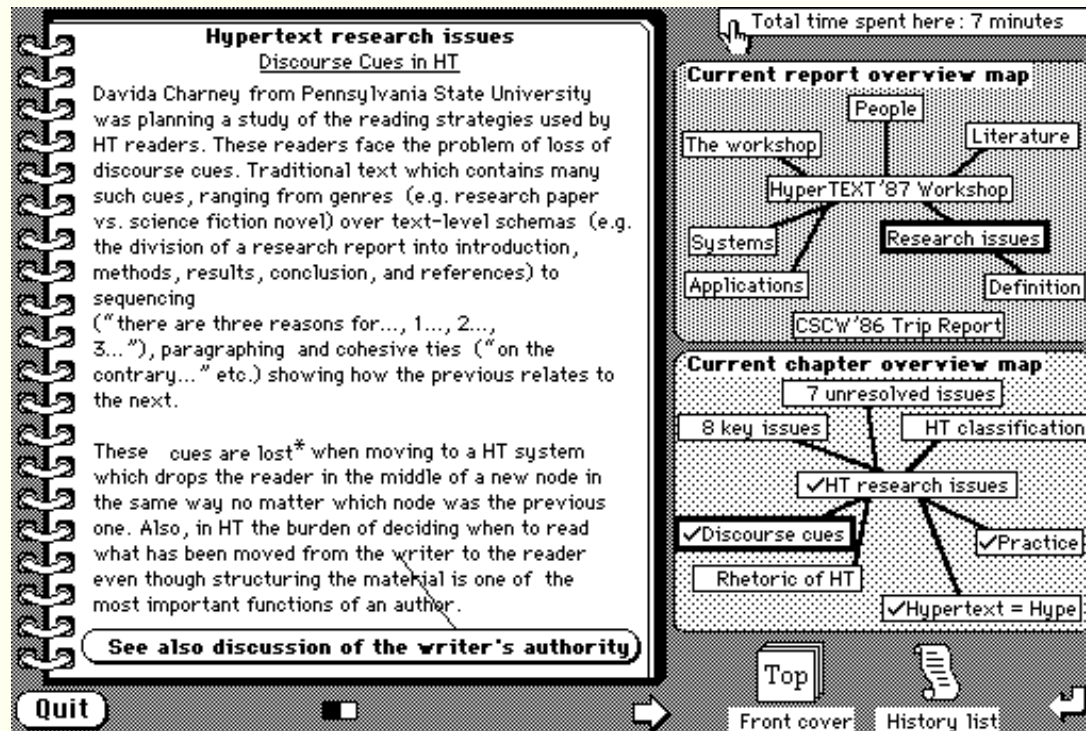
Createlt!, SuperCard, HyperCard

# Multimedia Authoring: Autorensysteme



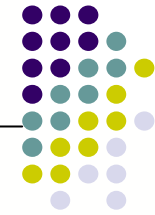
## Frame-basierte Autorensysteme: HyperCard

- 1987 von Apple eingeführt, Plattformabhängig
- Interne Scriptsprache für Erweiterungen: HyperTalk
- Screenshot:



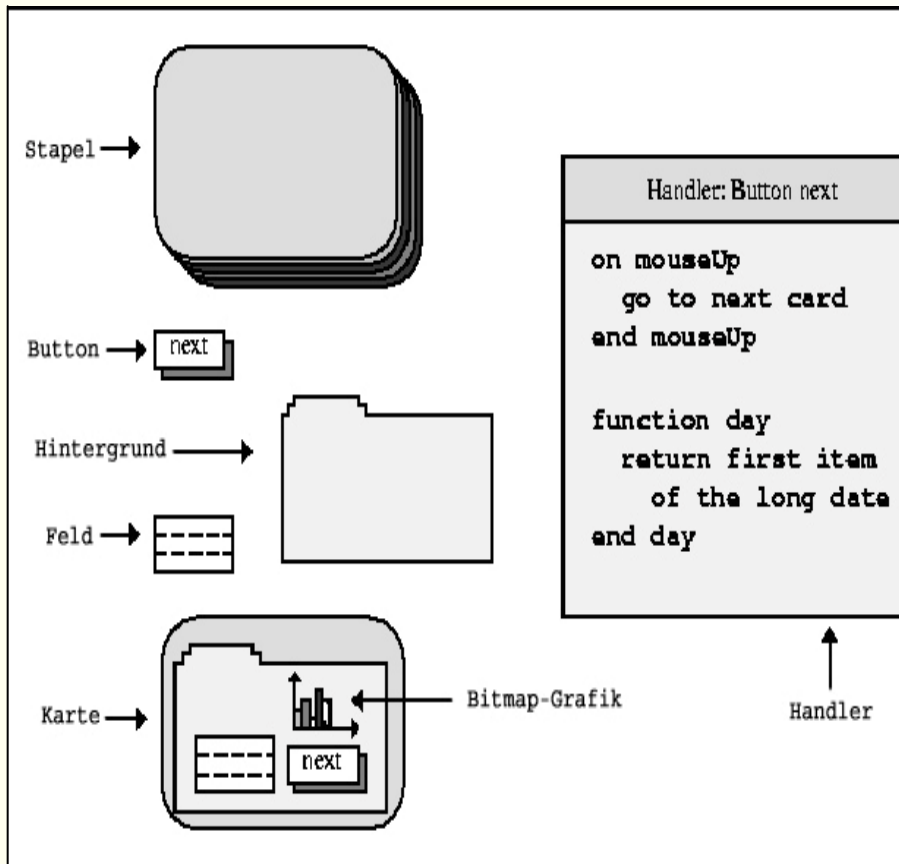
Links: grafische  
Präsentation

Rechts: interne Struktur



## Frame-basierte Autorensysteme: HyperCard

### Konstrukte von HyperCard:



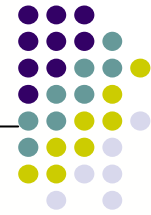
Karte: symbolisiert einen Bildschirm-ausschnitt

Hintergrund: Informationen, die für mehrere Karten gelten

Stapel: Menge von vernetzten Karten

Handler: Scriptsprache Hypertalk

Feld/Button: Navigation, Verhalten



## Timeline-basierte Autorensysteme

Zugrunde liegende Metapher:

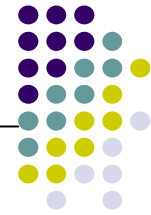
Die Medienobjekte werden entlang einer Zeitachse angebracht. Das MM-Dokument läuft intern diese Zeitachse ab und präsentiert alle Medienobjekte.

- ❖ Mehrere Medienobjekte können parallel auf einem Zeitpunkt liegen
- ❖ Sprünge und Schleifen können leicht realisiert werden
- ❖ Beispiele:

Action!, M3Integrator, Passport Producer Pro

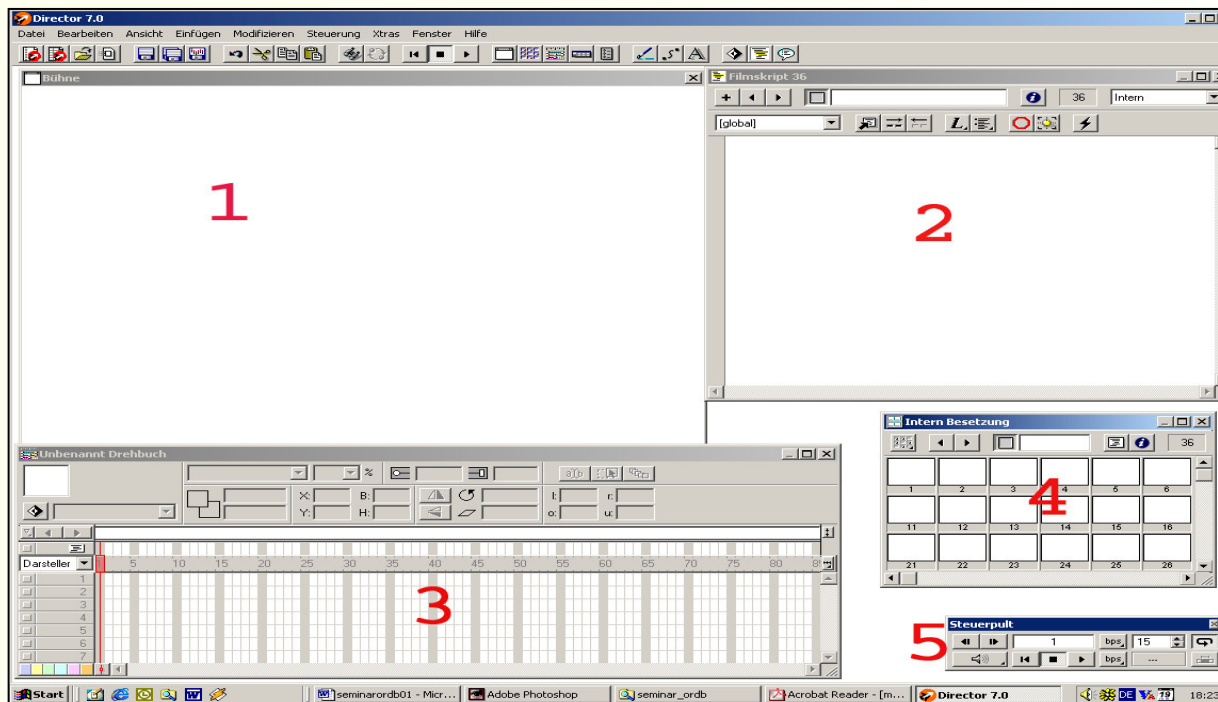
Macromedia Director und Authoring on the Fly (AoF)

werden nun näher vorgestellt und ihre Unterschiede diskutiert



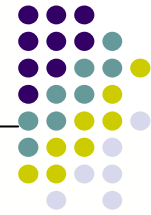
## Timeline-basierte Autorensysteme: DIRECTOR

- Übersichtliche Benutzeroberfläche
- Gut für Laien & unerfahrene Programmierer geeignet
- Schnelle Einarbeitung
- Arbeitsweise ähnlich zum Erstellen eines Filmes mit einem Drehbuch (Zeitachse!)
- Interne Scriptsprache: LINGO



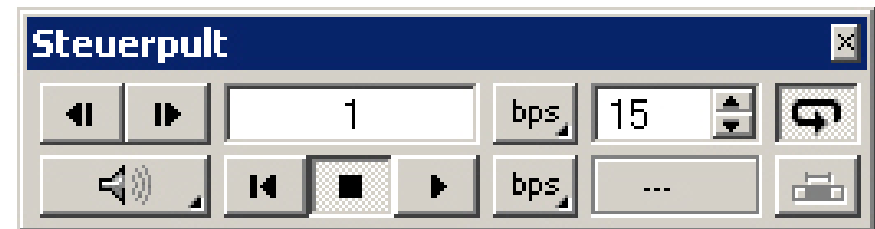
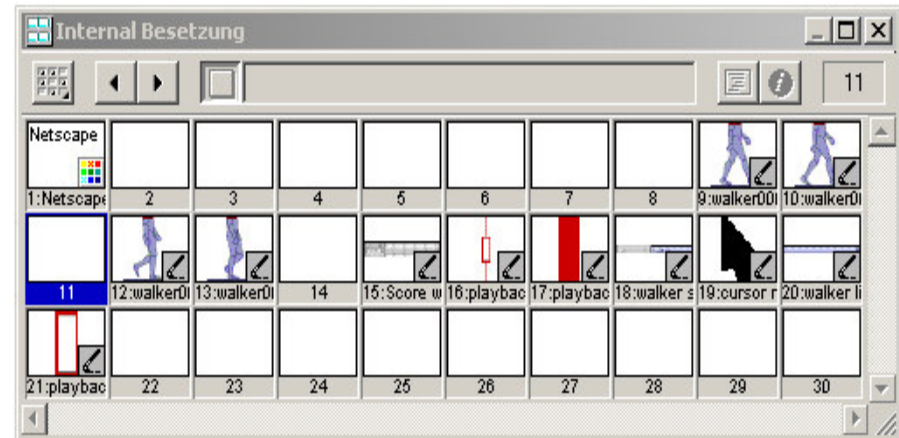
### Konstrukte von Director

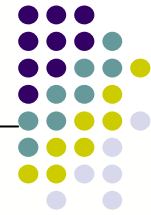
- 1: Bühne
- 2: Scriptfenster
- 3: Drehbuch
- 4: Besetzung
- 5: Steuerpult



## Timeline-basierte Autorensysteme: DIRECTOR

- Besetzung:  
interne Datenbank, um alle Darsteller (Medienobjekte) für die spätere Anwendung zu speichern
- Steuerpult:  
Analog zur Bedienung eines Videorecorders  
Zwischenresultate können direkt angeschaut werden

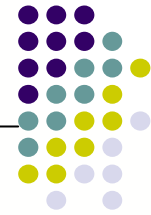




## Timeline-basierte Autorensysteme: DIRECTOR

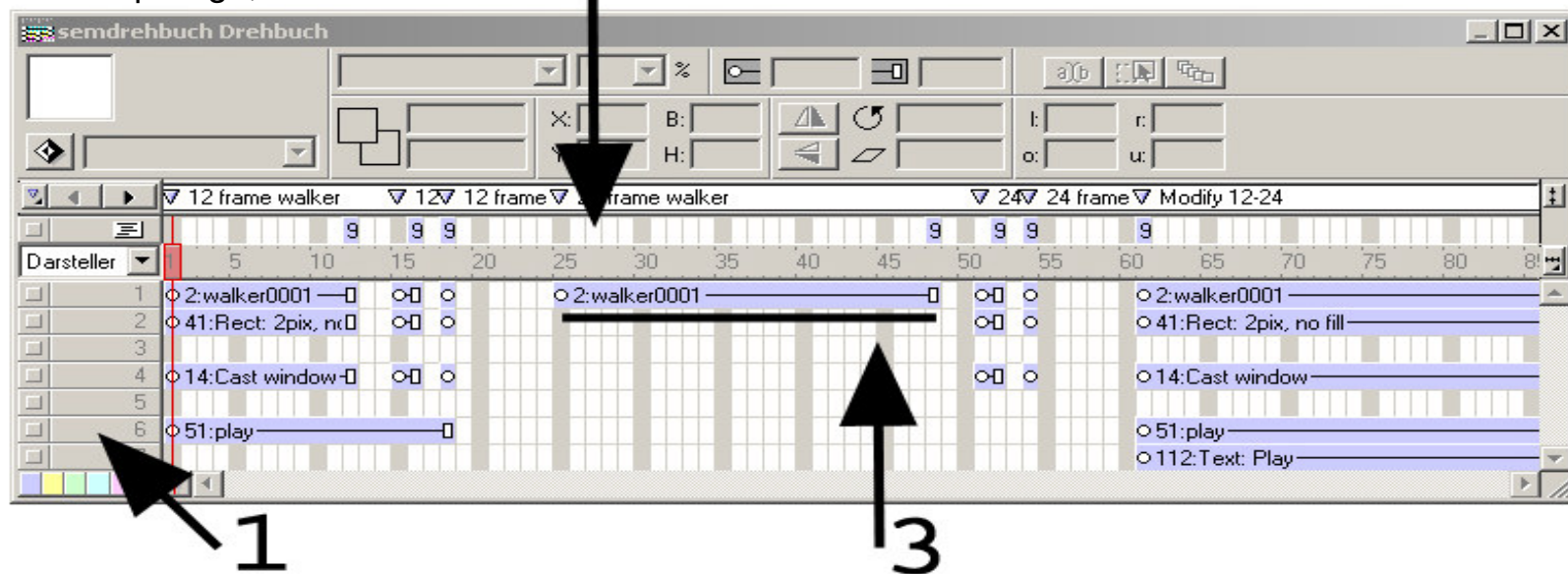
- Bühne:
  - WYSIWYG-Prinzip
  - Repräsentiert das spätere Bildschirmdesign
  - Darsteller aus der Besetzung werden auf der Bühne platziert
  - Anordnung auf der Bühne entspricht der Anordnung im späteren Programm



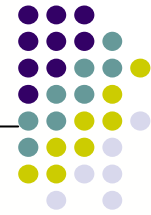


## Timeline-basierte Autorensysteme: DIRECTOR

- Drehbuch:
  - Zentrales Arbeitsfenster
  - Die einzelnen Bühnenausschnitte werden entlang einer Zeitachse gelegt
  - Besteht aus einer Matrix: (Marke1) Kanäle für die Darsteller (horizontal)  
(Marke2) Zeitpunkten/Einzelframes (vertikal)
  - Marke3: Sichtbarkeit eines Darstellers
  - Darsteller werden entlang 2 der Zeitachse im Drehbuch platziert
  - Sprünge, Schleifen







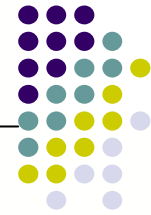
## Timeline-basierte Autorensysteme: DIRECTOR

- Scriptsprache LINGO:
  - Syntax ähnlich der englischen Sprache
  - Um nicht vorgefertigte Funktionalitäten zu verwirklichen
  - Darstellerskripte werden Darstellern zugeordnet
  - Bildskripte werden Frames zugeordnet

```
Score Script 5
+ < > [ ] [i] 5 Internal
mouseUp
on mouseUp
  global gInventoryUp
  if gInventoryUp = 0 then
    invisible
    CloseFolder
    go to "Map"
  end if
end

on mouseEnter
  global gInventoryUp
  if gInventoryUp = 0 then
    set the text of member "TextStatus" to "Game Map"
    netStatus "Show the Game Map"
  end if
end mouseEnter

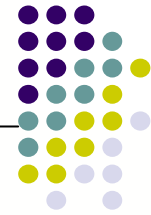
on mouseLeave
  global gInventoryUp
  if gInventoryUp = 0 then
    set the text of member "TextStatus" to " "
    netStatus " "
  end if
end mouseLeave
```



## Timeline-basierte Autorensysteme: DIRECTOR

Ein mit Director erstelltes MM-Dokument:

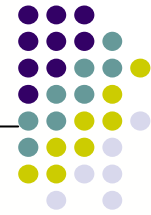
[Starte Beispiel!](#)



### Timeline-basierte Autorensysteme: Authoring on the Fly

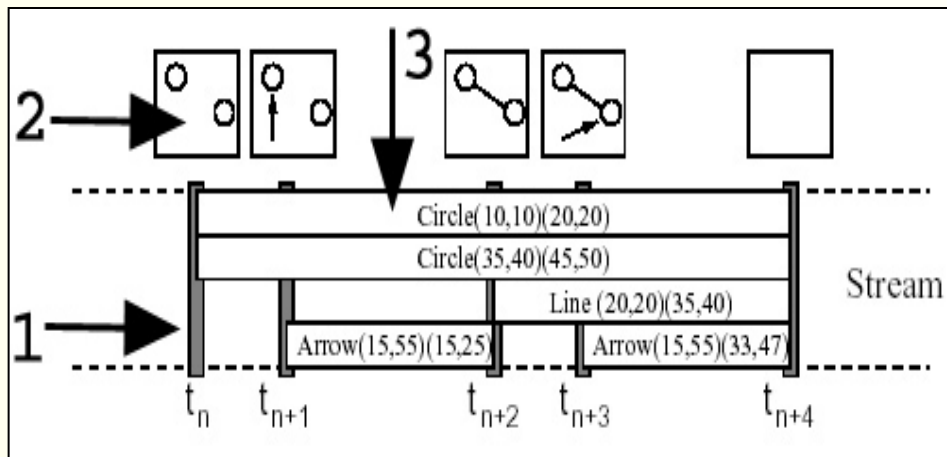
- AS arbeitet im Hintergrund um Vorträge live im Internet zu veröffentlichen
- Nachbearbeitung und Archivierung des Vortrags im Netz
- Vortragsmaterialien werden in Aof eingebunden
- Alle eingesetzten Medien müssen synchronisiert werden
- Automatische Vernetzung mit relevanter Literatur im Netz
- Verwendung bei WBT, CSCL und virtuellen Hochschulen

Die Arbeitsweise von Aof wird anhand der Synchronisation der Medienobjekte und Durchführung eines Vortrags näher erläutert



## Timeline-basierte Autorensysteme: AoF

### Synchronisation(1):



Presentation Element (PE):

eingesetzte Medienobjekte (Marke 3)

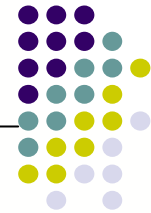
Presentation Channel (Kanal):

Kanal wird mit PE besetzt, solange dieser sichtbar ist während des Vortrags

Time Stamps (Zeitschranken):

unterteilt alle Kanäle für Synchronisation

- (M3) Kanal mit einem PE
- (M2) zeigt Bildschirmausschnitt
- (M1) Zeitschranke: alle PE's (die in Kanälen sind) werden angezeigt



## Timeline-basierte Autorensysteme: AoF

### Synchronisation(2):

ID	Object (PE)
15	Circle(10,10)(20,20)
16	Circle(35,40)(45,50)
17	Arrow(15,55)(15,25)
18	Line (20,20)(35,40)
19	Arrow(15,55)(33,47)

Object List

Time Stamp	Nr. of Objects	Object IDs
45500	2	15 16
46350	3	15 16 17
50670	3	15 16 18
51480	4	15 16 19 18
54020	0	

Event Queue

Object List:

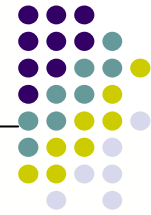
sequentiell aufsteigende Nummernliste  
mit allen im Vortrag enthaltenden PE's

Event Queue:

verwaltet alle Zeitschranken

Anzahl der PE's (→Kanalreservierung)

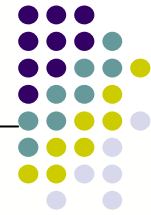
aktive PE's mit Identifikationsnummer



### Timeline-basierte Autorensysteme: AoF

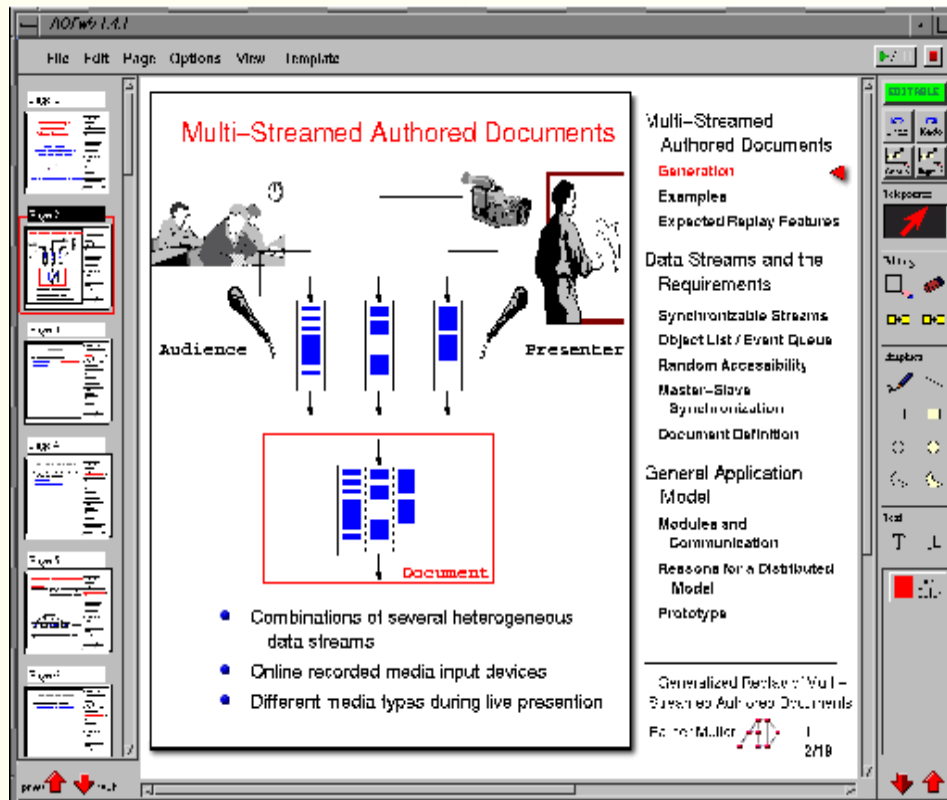
#### Master-Slave-Synchronisation:

- Synchronisation erfolgt auf Anwender-PC
- Ein PE wird zum **Master** (unkomprimiert + kontinuierlich)
- Alle anderen PE's sind automatisch **Slaves**
- Der Masterstrom (Medium, das am zeitkritischsten ist) wird in gleichgroße Blöcke zerlegt
- Blöcke werden durch Zeitschranken begrenzt
- In Event Queue stehen alle Slave-PE's, die zum Zeitpunkt aktiv sind
- Zu jedem Zeitpunkt werden aktive PE's abgespielt
- Durch Master erfolgt : *Inter-Stream-Synchronisation*



## Timeline-basierte Autorensysteme: AoF

### Durchführung eines Vortrages:



AofWhiteBoard: Arbeitsfläche des Dozenten

### **Vorbereitung:**

#### AofWhiteBoard:

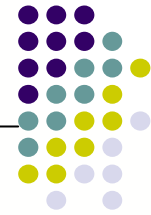
- integrierter Text- und Grafikeditor
- verwaltet alle Vortragsmaterialien (Bilder, Vortragskript, Video, Animationen, etc.)

### **Präsentation:**

- arbeitet mit Whiteboard
- Telepointer, Thumbnail-Übersicht
- neben Whiteboard wird auch Dozent gefilmt
- und übers Netz gemeinsam übertragen

### **Nachbereitung:**

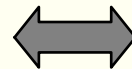
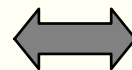
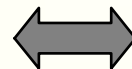
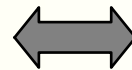
- Korrekturen beim Material vornehmen
- mit AofHTML im Netz archivieren und mit Literatur vernetzen



### Vergleich von Director und AoF

#### ● Director:

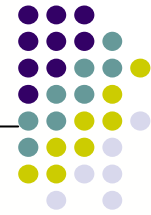
- Dominierende Rolle des AS
- Wird in Designphase eingesetzt
- Modelliert die Oberfläche des MM-Dokuments
- Arbeit nach Erstellung beendet



#### ● Authoring on the Fly:

- Arbeitet im Hintergrund, synchronisiert
- Designphase nur bei Nachbearbeitung
- Vortragsmaterial wird fertig integriert
- Nach der Liveübertragung archiviert aufHTML das MM-Dok.



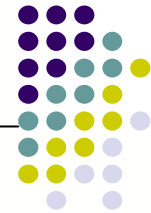


### Flowchart-basierte Autorensysteme

Zugrunde liegende Metapher:

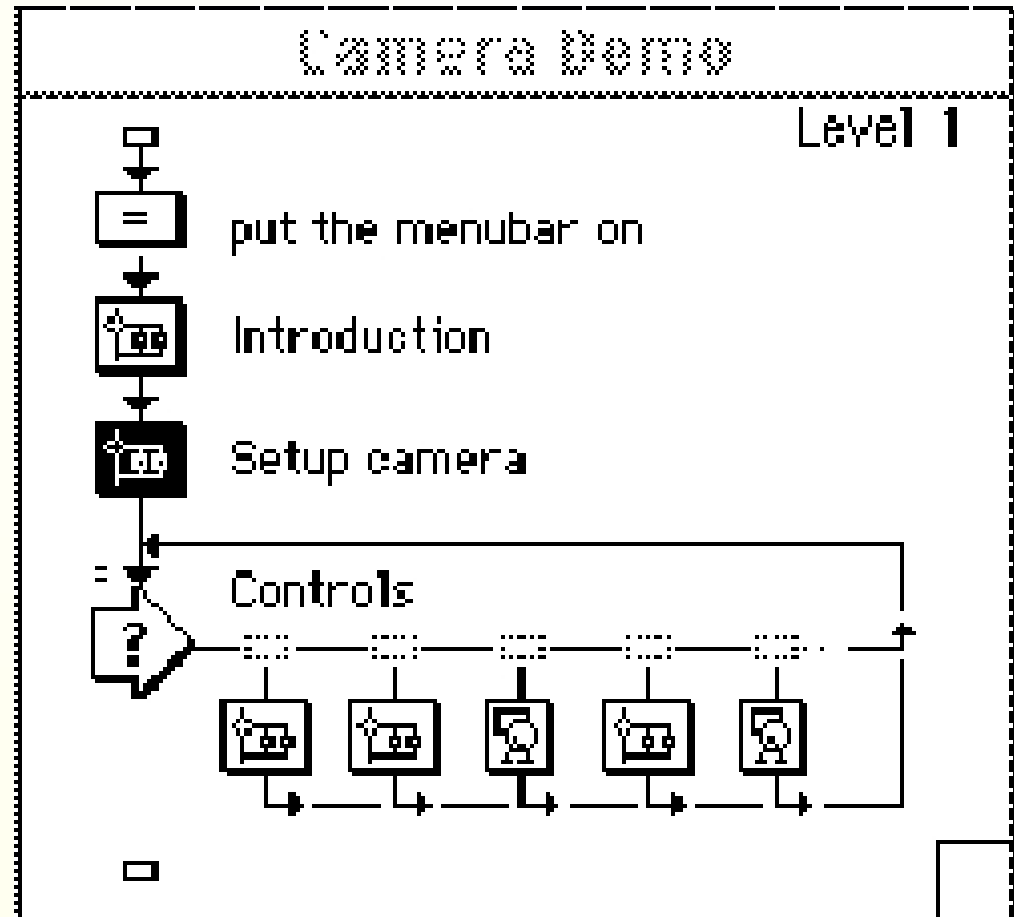
Die Medienobjekte werden durch Kanten miteinander verbunden, die den Navigationsverlauf wieder spiegeln. Die Kantenverläufe gleichen einem Flussdiagramm. Auch als *Icon-basierten AS* bekannt.

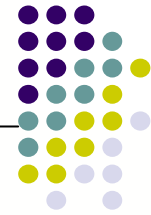
- Navigation wird durch ausgehende Kanten symbolisiert
- Verzweigung, Iteration leicht darstellbar
- Hierarchische Strukturierung möglich:
  - Module können ausgelagert werden
- Vernetzung erfolgt über Medienobjekte
- Die Erstellung einzelner Seiten erfolgt grafisch (ähnlich der Bühne von Director)



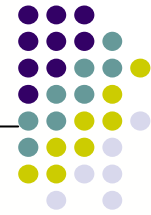
## Flowchart-basierte Autorensysteme

- Beispiele:
  - Authorware Prof.
  - Course Builder
  - C.A.M.E
  - IconAuthor
  - InterActive
  - MediaTool





- Die Entwicklung von Autorensystemen ist noch lange nicht abgeschlossen
- Ein ultimatives Autorensystem hat sich noch nicht heraus kristallisiert.
- Heutzutage gibt es viele ad-hoc Entwicklungen
- Prinzipien des Software-Engineering (Analyse- und Entwurfsmodell) müssen noch in die Autorensysteme integriert werden.  
Wartbarkeit und Wiederverwendung kann so gesteigert werden.



Vielen Dank für Ihre  
Aufmerksamkeit!

Für weitere Fragen stehe ich  
Ihnen nun gerne zur Verfügung.