Multimedia-Systems:
Synchronization

Prof. Dr.-Ing. Ralf Steinmetz
Dr. L.Wolf, Dr. S.Fischer

TU Darmstadt - Darmstadt University of Technology,
Dept. of Electrical Engineering and Information Technology, Dept. of Computer Science
KOM - Industrial Process and System Communications, Tel.+49 6151 166151,
Merckstr. 25, D-64283 Darmstadt, Germany, Ralf.Steinmetz@KOM.tu-darmstadt.de Fax. +49 6151 166152

GMD - German National Research Center for Information Technology
IPSI - Integrated Publication and Information Systems Institute, Tel.+49 6151 869869
Dolivostr. 15, D-64293 Darmstadt, Germany, Ralf.Steinmetz@darmstadt.gmd.de Fax. +49 6151 869870
Scope

Usage
- Learning & Teaching
- Design
- User Interfaces

Services
- Content Processing
- Documents
- Security
- ... Synchro-nization
- Group Communications

Systems
- Databases
- Programming
- Media-Server
- Operating Systems
- Communications
- Opt. Memories
- Quality of Service
- Networks

Basics
- Computer Architectures
- Compression
  - Image & Graphics
  - Animation
  - Video
  - Audio
Contents

1. Motivation - What is (Multimedia) Synchronization?
2. Synchronization and Multimedia
3. Reference Model for Multimedia - Synchronization
4. Synchronization in a Distributed Environment
5. Synchronization Techniques
6. Case Studies
1. Motivation - What is (Multimedia) Synchronization?

We concentrate on:

- **Temporal relationship**
  - New flavor in the context of multimedia
  - Essential for communications
Content Relations

Dependence of media objects on data

Examples:
- A graphic that visualizes data from a spreadsheet
- Two graphics showing different views of the same data

Explicit definition of dependences for automated update
- Only the data are edited
- All views of the data are
  - generated automatically
  - cannot be edited directly
- An update of the data triggers an update of the related views.

Implementation of content relations is based on the use of
- common data structures or
- common object interfaces
Spatial Relations

Usually known as layout relationships
- Define the space which is used for the presentation of a media object
  - on an output device
  - at a certain point of time in a multimedia presentation

Typically expressed in layout frames:
- A layout frame is placed and a content is assigned to this frame
- The positioning of a layout frame in a document may be
  - fixed to a position in a document
  - fixed to a position on a page
  - relative to the positioning of other frames

Spatial relations for time-dependent media objects:
- A frame or a group of frames may be presented in a window
- An audio presentation can be positioned on a stereo output device
Temporal Relations

Importance for time-dependent media objects
• new flavor in context of Multimedia
• essential for communications

Example for temporal relations:
• Temporal relation between
  • video object
  • audio object
recorded during a concert
At presentation time:
• temporal relation of the two media objects must correspond to the temporal relation at the recording moment

Relations:
• at the same time
• independently
• after another
Intra-object Synchronization

Intra-object Synchronization

*Intraobject synchronization defines the time relation between various presentation units of one time-dependent media object*

Example:

- Time relation between the single frames of a video sequence. For a video with a rate of 25 frames per second each of the frames has to be displayed for 40 ms

Frames of a video sequence that shows a jumping ball
Inter-object Synchronization

Inter-object synchronization defines the synchronization between media objects

Example:
- Time relations of a multimedia synchronization that starts with an audio/video sequence, followed by several pictures and an animation that is commented by an audio sequence.

```
Audio1
  P1  P2  P3
Video
    Animation
  Audio2
```

```t```
Logical Data Units

Logical Data Unit

An information unit - usually part of a sequence of information units - of a time-dependent media object

Granularity levels of LDUs

- Application dependent
- Imply a hierarchical decomposition of media objects
- Example:

<table>
<thead>
<tr>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notes</td>
</tr>
<tr>
<td>Movements</td>
</tr>
<tr>
<td>Symphony</td>
</tr>
</tbody>
</table>

The LDU hierarchy of a Symphony
Classification of LDUs

Closed LDUs
• have a predictable duration

Example:
• LDUs that are parts of stored media objects of continuous media like
  • audio and video or
  • stored media objects with a fixed duration.

Open LDUs
• duration is not predictable before the presentation execution

Example:
• Open LDUs typically represent
  • input from a life source, for example a camera or a microphone, or
  • media objects that include an user interaction
Classification of LDUs (cont.)

Physical units as LDUs:

<table>
<thead>
<tr>
<th>Pic. 1</th>
<th>Pic. 2</th>
<th>Pic. 3</th>
<th>Pic. 4</th>
<th>Pic.n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/30 s</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Physical units of a video sequence as LDUs

Blocks of samples as LDUs:

physical frame duration = 1/sample frequency (e.g., 1/8000 s)

Audio 1 Audio 2 Audio 3 Audio 4 Audio n

duration of a Logical Data Unit of 512 Bytes (e.g., = 0.064 s)

Blocked physical units of an audio sequence as LDUs
Classification of LDUs (cont.)

User selected LDU duration
• typically in the case of computer generated media objects.

LDUs of an animation sequence of user selected size

Varying durations of LDUs
• typically in the case of recorded user interactions.

LDUs of a recorded user interaction

LDUs of an user interaction in a window system with varying durations
Classification of LDUs (cont.)

Open LDUs with unpredictable duration
• Typically a user interaction with in advance unknown duration

An open LDU representing an user interaction

Timers represented as stream of empty LDUs

LDUs of a timer
## Classification of LDUs - Overview

<table>
<thead>
<tr>
<th>Duration defined by</th>
<th>Fixed Duration of LDU</th>
<th>Variable or unknown Duration of LDU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Capturing</td>
<td>Recorded interaction</td>
</tr>
<tr>
<td></td>
<td>Audio, Video</td>
<td>User interaction</td>
</tr>
<tr>
<td></td>
<td>Animation, Timer</td>
<td></td>
</tr>
</tbody>
</table>
Synchronization Examples

Lip synchronization
- demands for a tight coupling of audio and video streams with
- a limited skew between the two media streams

LDU view of lip synchronization

Slide show with audio comment:

LDU view of a slide show
Application Example

Example used further in the course:

A lip synchronized audio video sequence (Audio1 and Video) is followed by a replay of a recorded user interaction (RI), a slide sequence (P1 - P3) and an animation (Animation) which is partially commented using an audio sequence (Audio2). Starting the animation presentation, a multiple choice question is presented to the user (Interaction). If the user has made a selection, a final picture (P4) is shown.
2. Synchronization and Multimedia

Typical Requirements:
- **hard vs. soft**

**Strong Interaction with QoS**
- **acceptable, too bad ...**

**Variety of Mechanisms**
- Live Synchronisation
- Live Synchronisation with Storage
- Synthetical Synchronisation
Why Multimedia-Synchronization?

<table>
<thead>
<tr>
<th>Media</th>
<th>Influenced by</th>
<th>Skew (QoS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video</td>
<td>Animation</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>Audio</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>Image</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>Text</td>
<td>?</td>
</tr>
<tr>
<td>Audio</td>
<td>Animation</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>Audio</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>Image</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>Text</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>Pointer</td>
<td>?</td>
</tr>
</tbody>
</table>
Multimedia Synchronization - Experiments

Example:
• at IBM ENC Heidelberg to quantify synchronization requirements for:
  • Audio/video synchronization
  • Audio/pointer synchronization

Selection of material:
• Duration
  • 30s in experiments
  • 5s would have been sufficient
• Reuse of same material for all tests

Introduction of artificial skew:
• By media composition with professional video equipment
• With frame based granularity

Test conditions:
• Huge set of test candidates
  • Professional: cutter at TV studios
  • Casual: every day “user”
• Awareness of the synchronization issues
• Set of tests with different skews lasted 45 min
### Feedback - Questionnaire

1. While watching this video clip, did you **detect any artifact** or **strange effect**? If so, please try to describe it in a few words. (-)

   ....

2. Are you able to **identify** if audio was ahead of or behind the moving pictures? (.)

   a) Yes, I identify **audio** to be played **ahead of video**
   
   b) Yes, I identify **audio** to be played **behind video**
   
   c) No, I notice that audio is out of sync with respect to video but, I am **not sure** if audio is played ahead of or behind video.

3. You noticed a synchronization error. How would you **qualify this error** if you have to watch all your TV programs with such an error? (.)

   a) I would not mind, the error is **acceptable**
   
   b) I dislike it, the error is **annoying**
   
   c) I am **not sure** if I would accept such an error or if I would really dislike it
Lip Synchronization: Major Influencing Factors

Video:
- **Content**
  - Continuous (talking head) vs. discrete events (hammer and nails)
  - Background (no distraction)
- **Resolution and quality**
- **View mode** (head view, shoulder view, body view)

Audio:
- **Content**
- **Background noise or music**
- **Language and articulation**
Lip Synchronization: Level of Detection

Areas:
- In sync QoS: +/- 80 ms
- Transient
- Out of sync
Lip Synch.: Level of Accuracy/Annoyance

Some observations:

- **Asymmetry**
- **Additional tests with long movie**
  - +/- 80 ms: no distraction
  - -240 ms, +160 ms: disturbing
Pointer Synchronization

Fundamental CSCW shared workspace issue

Analysis of CSCW scenarios:
- Discrete pointer movement (e.g. “technical sketch”)
- Continuous pointer movements (e.g. “route on map”)

Most challenging probes:
- Short audio
- Fast pointer movement
Observations:
- **Difficult to detect “out of sync”:**
  - i.e., other magnitude than lip sync
- **Asymmetry:**
  - According to every day experience
Pointer Synchronization: Level of Annoyance

Areas:
- In sync: QoS -500 ms, +750 ms
- Transient
- Out of sync
# Synchronization QoS: Skew Values

<table>
<thead>
<tr>
<th>Media</th>
<th>Mode, Application</th>
<th>QoS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video</td>
<td>Animation correlated</td>
<td>+/- 120 ms</td>
</tr>
<tr>
<td>Video</td>
<td>Audio lip synchronization</td>
<td>+/- 80 ms</td>
</tr>
<tr>
<td>Image</td>
<td>overlay</td>
<td>+/- 240 ms</td>
</tr>
<tr>
<td>Image</td>
<td>non-overlay</td>
<td>+/- 500 ms</td>
</tr>
<tr>
<td>Text</td>
<td>overlay</td>
<td>+/- 240 ms</td>
</tr>
<tr>
<td>Text</td>
<td>non-overlay</td>
<td>+/- 500 ms</td>
</tr>
<tr>
<td>Audio</td>
<td>Animation event correlation (e.g., dancing)</td>
<td>+/- 80 ms</td>
</tr>
<tr>
<td>Audio</td>
<td>tightly coupled (stereo)</td>
<td>+/- 11 µs</td>
</tr>
<tr>
<td>Audio</td>
<td>loosely coupled (dialogue mode with various participants)</td>
<td>+/- 120 ms</td>
</tr>
<tr>
<td>Audio</td>
<td>loosely coupled (e.g., background music)</td>
<td>+/- 500 ms</td>
</tr>
<tr>
<td>Image</td>
<td>tightly coupled (e.g., music with notes)</td>
<td>+/- 5 ms</td>
</tr>
<tr>
<td>Image</td>
<td>loosely coupled (e.g., slide show)</td>
<td>+/- 500 ms</td>
</tr>
<tr>
<td>Text</td>
<td>Anmerkungen zu Text</td>
<td>+/- 240 ms</td>
</tr>
<tr>
<td>Pointer</td>
<td>Audio Related to the Item</td>
<td>-500ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+750 ms</td>
</tr>
</tbody>
</table>
3. Reference Model for Multimedia - Synchronization

A reference model is needed

• to understand the various requirements for multimedia synchronization

• to identify and structure runtime mechanisms

• to identify interfaces between runtime mechanisms

• to compare system solutions for multimedia synchronization systems
Existing Synchronization Models

Classification methods:
- Physical, system, and human level
- Intra-stream (fine-grain) synchronization and inter-stream (coarse-grain) synchronization
- Live and synthetical synchronization
- Mapping

Uninterpreted Media Objects

Uninterpreted Bytestream

Multimedia Object

Derived Media Objects

Time composition

Derivation

Interpretation

Media Objects
Classification of Synchronization Techniques

Inter-object synchronization techniques:
• used to control jitter between media streams
• classification according to the type and location of the synchronization control

Distributed synchronization control:
• distributed control based on protocols
• distribution based on servers
• distribution on nodes without server structure

Local synchronization control:
• control on several layers
• use of local servers
Four Layer Model

Multimedia application

Specification layer

Object layer

Stream layer

Media layer

Abstraction

low

high
Synchronization Specification

Synchronization specification

• describes all temporal dependencies of the included objects in a multimedia object
• is part of the description of a multimedia object
• should comprise:
  • Intra-object synchronization specifications for the media objects of the presentation
  • Quality of service descriptions for the intra-object synchronizations
  • Inter-object synchronization specifications between the media objects of the presentation
  • Quality of service descriptions for the inter-object synchronizations
• may comprise:
  • alternative presentation forms
• must be explicitly described in a synthetical synchronization
• is implicitly defined in a live synchronization
# Layered Model - Summary

<table>
<thead>
<tr>
<th>Layer</th>
<th>Interface Abstraction</th>
<th>Tasks</th>
</tr>
</thead>
</table>
| Specification | The tools performing the tasks of this layer have interfaces, the layer itself has no upper interface | to edit  
to format  
to map the quality of service values to the object layer |
| Object    | Objects that hide types of enclosed media  
Synchronization Specification  
Media-oriented quality of service (In terms like allowed skew, allowed jitter) | Plan and coordinate presentation execution  
Initiate presentation of time-dependent media objects  
Initiate presentation of time-independent media objects |
| Stream    | Streams and groups of streams  
Guarantees for intra-stream synchronization  
Guarantees for inter-stream synchronization of streams in a group | Resource Reservation and scheduling of LDU processing |
| Media     | Device independent access to LDUs.  
Guarantees for single LDU processing | File and device access |
4. Synchronization in a Distributed Environment

More complex than in a local environment:

- Distributed storage of synchronization information
- Distributed storage of media objects
- Communication delays and jitter
- Multi-party communication patterns
Transport of Synchronization Specification

Delivery of the complete synchronization information before the start of the presentation

Use of an additional synchronization channel:

Use of a multiplexed channel:
Location of Synchronization Operations

Synchronization at the sink node

Source

Bandwidth demand

Sink

Synchronization at the source node

Source

Bandwidth demand

Sink
Clock Synchronization

Problem: Skew between the clocks

- Clock skew has to be considered in the planning of the presentation:
  - \( T_{av} \): Start of presentation
  - \( N_{la} \): Known net delay
  - \( S_a, S_v \) as Skews
  - Start time of audio transmission of Source A: \( T_a = T_{av} - N_{la} - S_a \)
  - Start time of video transmission of Source V: \( T_v = T_{av} - N_{lb} - S_v \)

- Restricted buffer size demands for restricted skew
Multiple Communication Relations

Sources : Sinks

1:1

n:1

1:n

n:m

Sources  Sinks
Multi Step Synchronization

• Synchronization during object acquirement

• Synchronization of retrieval

• Synchronization during delivery of the LDUs to the network

• Synchronization during the transport

• Synchronization at the sink

• Synchronization within the output device
5. Synchronization Techniques

Specification of great importance

- Simple specification method:

Types of temporal relations between two objects:

- A before B
- A overlaps B
- A starts B
- A equals B
- A meets B
- A during B
- A finishes B
Specification Methods - Requirements

Complex specification comprise:
- multiple object synchronization
- user interactions

Requirements for a specification method:
- Support of object consistency and for maintenance of specifications
- Media objects should be kept as one logical unit in the specification
- All types of synchronization relations should be easily describable
- Support for the integration of time-dependent and time-independent media objects
- Quality of service requirements must be describable
- Support of hierarchical levels of a synchronization
Basic Hierarchical Specification

Two main synchronization operations:
• Serial synchronization of actions
• Parallel synchronization of actions

Action:
• an atomic action handles the presentation of either a single-media object or a user input or a delay.
• a compound action is a combination of synchronization operators and atomic actions.
Basic Hierarchical Specification

Delay as a possible action allows for
• delays in serial presentations and
• delayed presentations of objects in a parallel synchronization.

Application example:

Hierarchical specification of the application example (RI = Recorded Interaction, Pic. = Picture, Aud. = Audio, Ani. = Animation, UI = User Interaction)
Basic Hierarchical Specification

Not all relations can be described.

Example of a synchronization that can not be described

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to understand</td>
<td>Additional description of skew quality of service necessary</td>
</tr>
<tr>
<td>Natural support of hierarchies</td>
<td>For the presentation of time-independent media objects presentation durations must be added</td>
</tr>
<tr>
<td>Integration of interactive objects is easy</td>
<td>Need for the splitting of media objects for synchronization purposes</td>
</tr>
<tr>
<td></td>
<td>No adequate abstractions for media object contents</td>
</tr>
<tr>
<td></td>
<td>Some synchronization scenarios can not be described</td>
</tr>
</tbody>
</table>
Synchronization Based on a Global Timer

All single-media objects are attached to a time axis

Application example

Time axis synchronization specification of the application example
## Synchronization Based on a Global Timer

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to understand</td>
<td>Objects of unknown duration cannot be integrated, extensions to the model are required</td>
</tr>
<tr>
<td>Support of hierarchies easy to realize</td>
<td>Skew quality of service has to be specified indirectly by using the common time axis or additional quality of service specifications have to be given</td>
</tr>
<tr>
<td>Easy to maintain because of the mutual independence of objects</td>
<td>In a distributed environment the notion of a global unique time has to be kept</td>
</tr>
<tr>
<td>Good abstraction for media contents</td>
<td></td>
</tr>
<tr>
<td>Integration of time-independent objects is easy</td>
<td></td>
</tr>
</tbody>
</table>
Synchronization Based on Virtual Axes

Specification using
- coordinate systems with user defined measurement units
- several virtual axes to create a virtual coordinate space

Example:

Virtual axes with measurement unit pitch

Virtual axis with measurement unit beat

Music notes as example for virtual axes

At runtime the virtual axes are mapped to real axes
Synchronization Based on Virtual Axes

Application example:

Virtual time axis specification of the application example
Synchronization Based on Virtual Axes

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to understand</td>
<td>Skew quality of service only indirectly or by additional specifications definable</td>
</tr>
<tr>
<td>Often specification according to the problem space possible</td>
<td>Specification may become complex by many axes</td>
</tr>
<tr>
<td>Good possibility for building hierarchies</td>
<td>Mapping of axes at runtime may be complex and time consuming</td>
</tr>
<tr>
<td>Easy maintainable, because objects are kept as units and mutual independent objects</td>
<td></td>
</tr>
<tr>
<td>Good abstraction for media content</td>
<td></td>
</tr>
<tr>
<td>Easy integration of time independent media objects</td>
<td></td>
</tr>
<tr>
<td>Interactive objects can be included using specialized axes</td>
<td></td>
</tr>
</tbody>
</table>
Synchronization based on Reference Points

Synchronization via reference points:
• Time-dependent single-media objects are regarded as sequences of LDUs
• Reference points:
  • the start time and the stop time of the presentation of a media object
  • the start-times of the subunits of time-dependent media objects
• Synchronization point:
  • A set of connected reference points
  • Defines the synchronization between objects
  • LDUs have to be started resp. stopped when it is reached
• Description of temporal relations without explicit reference to time.
Synchronization based on Reference Points

Lip synchronization example

Lip synchronization in the reference point synchronization model

Slide show example.

Example of a slide show with audio sequence in the reference point model
Synchronization based on Reference Points

Application example

Example of the integration of time-dependent and time-independent media objects as well as closed and open LDUs in a reference point synchronization specification.
# Synchronization based on Reference Points

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to understand</td>
<td>More difficult to change because of the direct description of dependencies</td>
</tr>
<tr>
<td>Natural description of temporal relations</td>
<td>Sometimes complex hierarchies</td>
</tr>
<tr>
<td>Easy integration of open LDUs</td>
<td></td>
</tr>
<tr>
<td>Integrated description of skew quality of service</td>
<td></td>
</tr>
<tr>
<td>Easy integration of time-independent objects</td>
<td></td>
</tr>
<tr>
<td>Time axis are special case of the reference point synchronization model</td>
<td></td>
</tr>
</tbody>
</table>
Event-based Synchronization

Event-based synchronization presentation actions are initiated by the synchronization events

Typical presentation actions are:
• to start a presentation
• to stop a presentation
• to prepare a presentation

Events
• may be external, e.g. generated by a timer, or
• internal to the presentation generated by a time-dependent media object that reaches a specific LDU
Event-based Synchronization

Application example:

<table>
<thead>
<tr>
<th>Action</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Start</td>
</tr>
<tr>
<td>Audio1</td>
<td>start</td>
</tr>
<tr>
<td>Video</td>
<td>start</td>
</tr>
<tr>
<td>Pic.1</td>
<td></td>
</tr>
<tr>
<td>Timer1</td>
<td></td>
</tr>
<tr>
<td>Pic.2</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
# Event-based Synchronization

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy integration of interactive objects</td>
<td>Not easy to handle</td>
</tr>
<tr>
<td>Easy extensible by new events</td>
<td>Complex specification</td>
</tr>
<tr>
<td>Flexible because any event can be specified</td>
<td>Hard to maintain</td>
</tr>
<tr>
<td>Integration of time-independent objects by using additional timers</td>
<td>Separate descriptions of quality of service necessary</td>
</tr>
<tr>
<td></td>
<td>Difficult use of hierarchies</td>
</tr>
</tbody>
</table>
Petri net-based synchronization

The rules for a timed petri net are:
• A transition fires if all input places contain a non blocking token.
• If a transition fires, on each input place a token is removed and to each output place a token is added
• After firing, a token is blocked for the duration that is assigned to its new place

Lip synchronization example.

Petri net based synchronization of a lip synchronization
Petri-Net based synchronization

Example for a hierarchy:
Petri-Net based synchronization

Application example:

Petri net specification of the application example
## Petri-Net based synchronization

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchies can be created</td>
<td>Not easy to handle</td>
</tr>
<tr>
<td>Easy integration of time-independent objects</td>
<td>Complex specification</td>
</tr>
<tr>
<td>Easy integration of interactive objects</td>
<td>Splitting of media objects</td>
</tr>
<tr>
<td>Integrated skew quality of service</td>
<td>Insufficient abstraction for the content of the media objects</td>
</tr>
</tbody>
</table>
Scripts

Script

- Textual description of a synchronization scenario
- Elements of scripts are activities and subscripts
- Three main operations in a script are
  - serial presentation
  - parallel presentation
  - repeated presentation of a media object
- Often full programming languages extended by the timing operations
activity DigAudioAudio("video.au");
activity SMPVideo("video.smp");
activity XRecorderRecorder("window.rec");
activity PicturePicture1("bild1.jpeg");
activity PicturePicture2("bild2.jpeg");
activity PicturePicture3("bild3.jpeg");
activity PicturePicture4("bild4.jpeg");
activity StartInteractionSelection;
activity DigAudioAniAudio("animation.au");
activity RTAnimaAnimation("animation.ani");

script Picture_sequence 3Pictures=
    Picture1.Duration(5) >>
    Picture2.Duration(5) >>
    Picture3.Duration(5);

script Lipsynch AV = Audio & Video;
script AniComment AA = Animation &
    AniAudio.Translate(2);
script Multimedia Application_example {
    AV >> Record. UI >> 3Pictures >>
    ( (Selection >> Picture4) & AA )}
## Scripts

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy hierarchies</td>
<td>Not easy to handle</td>
</tr>
<tr>
<td>Logical structure of objects can be kept</td>
<td>Complex specification</td>
</tr>
<tr>
<td>Easy integration of time-independent objects</td>
<td>Implicit usage of common timers necessary</td>
</tr>
<tr>
<td>Easy integration of interactive objects</td>
<td>Special constructs for skew quality of service necessary</td>
</tr>
<tr>
<td>Skew quality of service can be specified</td>
<td></td>
</tr>
<tr>
<td>Easy extensible by new synchronization constructs</td>
<td></td>
</tr>
<tr>
<td>Flexible because programmable</td>
<td></td>
</tr>
</tbody>
</table>
6. Case Studies

MHEG

Coded Representation of Multimedia and Hypermedia Objects
- representation and encoding of
- final form multimedia and hypermedia information objects to be interchanged within or across applications and services

MHEG objects
- *Content objects* encapsulate media objects that are part of the presentation
- *Composite objects* act as a ‘container’ to group a set of MHEG objects
- *Action objects* are used to exchange sets of actions to be performed between objects
- *Link objects* are used to specify spatial, temporal and conditional relations between MHEG objects
Synchronization in MHEG

Generic space:
- **provides a virtual coordinate system**
- **used to specify the layout and relation of content objects**
  - one time axis of infinite length measured in generic units of time
  - three spatial axes (X=latitude, Y=longitude, Z=altitude) of finite length in an interval of [-32768,+32767]
- **MHEG maps from the virtual to the real coordinate space**

Presentation of content objects:
- **Based on the exchange of action objects that are sent to an object**
- **Examples for actions are:**
  - *prepare* for setting the object in a state that enables the object to present itself
  - *run* to start the presentation
  - *stop* to stop the presentation
MHEG action lists

Action list:
- Combination of actions to action lists
- Parallel action lists are executed in parallel
- Each list is composed of a delay followed by delayed sequential actions.

Links:
- used to create event-based synchronizations
- can trigger events
MHEG Engine

Application

MHEG Engine
- Object Manager
- Link Processor
- Interpreter

Generic Presentation Services

AV-Sub-system

User Interface Services

Operating System
MHEG Relation to Reference Model

- Specification layer
- Object layer interface
- Object layer
- Stream layer interface
- Stream layer
- Media layer interface
- Media layer

---

MHEG coded multimedia object

MHEG Engine

Interpreter

Object Manager

Link Processor

User Interface Service

AV Subsystem
Case Studies

Other systems to be mentioned:

• HyTime
• FireFly
• MODE (Multimedia Objects in a Distributed Environment)
• Multimedia Tele-Orchestra
• ACME (Abstractions for Continuous MEdia)
• Apple QuickTime
• ...