Notion of Synchronization

Sync in correspondence to

- Content relation
- Spatial relation
- Temporal relation

Content Relation

- Define dependency of media objects for some data
- Example: dependency between spreadsheet and graphics that represent data listed in spreadsheet

Slides courtesy Prof. Nahrstedt
Spatial Relation

- **Layout relation**
  - Defines space used for presentation of media object on output device at certain point of multimedia presentation
  - Example: desktop publishing

- **Layout frames**
  - Placed on output device and content assigned to frame
  - Positioning of layout frames:
    - Fixed to position of document
    - Fixed to position on page
    - Relative to position of other frame
  - Example: in window-based system, layout frames correspond to windows and video can be positioned in window
Temporal Relation (Our focus!!!)

- Defines temporal dependencies between media objects
- Example: lip synchronization
- Time-dependent object
  - Media stream since there exist temporal relations between consecutive units of the stream
- Time-independent object
  - Traditional medium such as text or images
- Temporal synchronization
  - Relation between time-dependent and time-independent objects
  - Example: audio/video sync with slide show
Temporal Relations

- Synchronization considered at several levels of Multimedia Systems

- Level 1: OS and lower level communication layers
  - CPU scheduling, semaphores during IPC, traffic shaping network scheduling
  - Objective: avoid jitter at presentation time of one stream

- Level 2: Middleware/Session layer (Run-time)
  - Synchronization of multimedia streams (schedulers)
  - Objective: bounded skews between various streams

- Level 3: Application layer (Run-time)
  - Support for synchronization between time-dependent and time-independent media together with handling of user interaction
  - Objective: bounded skews between time-dependent and time-independent media
Synchronization Specification

- Implicit
  - Temporal relation specified implicitly during capturing of media objects
  - Goal: use this temporal relation to present media in the same way as they were originally captured
  - Example: Audio and Video recording and playback

- Explicit
  - Temporal relation specified explicitly to define dependency in case media objects were created independently
  - Example: creation of slide show
    - Presentation designer
      - selects slides,
      - creates audio objects,
      - defines units of audio presentation stream,
      - defines units of audio presentation stream where slides have to be presented
Logical Data Units and their Classification

- Time-dependent presentation units are called **logical data units (LDUs)**.
- LDU classification
  - Open
  - Closed
- LDUs important
  - In specification of system synchronization

<table>
<thead>
<tr>
<th>Samples or Pixels</th>
<th>Notes or frames</th>
<th>Movements or Scenes</th>
<th>Symphony or Movie</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDU spec during capturing</td>
<td>LDU spec defined by user</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fixed LDU</td>
<td>Audio/Video</td>
<td>Animation/Timer</td>
<td></td>
</tr>
<tr>
<td>Variable LDU</td>
<td>Recorded Interactions</td>
<td>User Interactions</td>
<td></td>
</tr>
</tbody>
</table>
Synchronization Classification

- **Intra-object Synchronization**
  - Time relation between various presentation units of one time-dependent media stream

- **Inter-object Synchronization**
  - Time relation between media objects belonging to two time-dependent media streams
Synchronization Classification

- **Live Synchronization**
  - **Goal:** exactly reproduce at presentation temporal relations as they existed during capturing process
  - **Requirement:** must capture temporal relation information during media capturing
  - **Example:** video conference, phone service
  - **Example:** recording and retrieval services – presentations with delay
Synchronization Classification

- Synthetic Synchronization
  - Goal: arrange stored data objects to provide new combined multimedia objects via artificial temporal relations
  - Requirements: support flexible synchronization relations between media
  - Example: authoring, tutoring systems

- Two phases:
  - Specification phase – define temporal relations
  - Presentation phase – present data in sync mode
Synchronization Requirements during media presentations

- For intra-object synchronization
  - Need accuracy concerning jitter and delays in presentation of LDUs

- For inter-object synchronization
  - Need accuracy in parallel presentation of media objects

- Implication of blocking:
  - O.K. for time-independent media
  - Problem for time-dependent media – gap problem
Gap Problem in Synchronization

- What does blocking of stream mean for output device?
  - Should we repeat previous music, speech, picture?
  - How long should such gap exist?

- Solution 1: restricted blocking method
  - Switch output device to last picture as still picture
  - Switch output device to alternative presentation if gap between late video and audio exceeds predefined threshold

- Solution 2: resample stream
  - Speed up or slow down streams
  - Off-line re-sampling – used after capturing of media streams with independent streams
    - Example: concert which is captured with two independent audio/video devices
  - Online re-sampling – used during presentation in case gap between media streams occurs
Lip Synchronization

- Temporal relation between audio and video
- **Synchronization skew**
  - Time difference between related audio and video LDUs
- **Streams in sync** iff skew = 0 or skew ≤ bound
- **Negative skew**: video before audio
- **Positive skew**: Audio before video

![Graph showing error rate (%) vs skew [ms]](image)

Legend:
- **head view**
- **shoulder view**
- **body view**
Lip Synchronization

Perception of Synchronization Errors
Skew Level found to be annoying
Lip Synchronization Requirements

- **In sync:**
  - $-80\text{ms} \leq \text{skew} \leq 80\text{ms}$

- **Out of sync:**
  - Skew $< -160\text{ms}$
  - Skew $> 160\text{ms}$

- **Transient:**
  - $-160\text{ms} \leq \text{skew} < -80\text{ms}$
  - $80\text{ms} < \text{skew} \leq 160\text{ms}$
Pointer Synchronization

Pointer Sync based on technical drawing

Pointer Sync based on map
Pointer Synchronization

Negative skew: pointer before audio
Positive skew: pointer after audio
Pointer Synchronization Requirements

- In sync:
  - \(-500\text{ms} \leq \text{skew} \leq 750\text{ms}\)

- Out of sync:
  - Skew < -1000ms
  - Skew > 1250ms

- Transient sync situation:
  - \(-1000\text{ms} \leq \text{skew} < -500\text{ms}\)
  - \(750\text{ms} < \text{skew} \leq 1250\text{ms}\)
Other Sync Requirements

- Jitter delay of digital audio
  - Max. allowable jitter:
    - 5-10 ns (perception experiments)
    - 2 ms (other experiments)

- Combination of audio and animation
  - Not stringent as lip sync
  - Max allowable skew: +/- 80ms

- Stereo audio
  - Tightly coupled
  - Max allowable skew: 20 ms
    - Due to listening errors, suggestion even +/- 11ms

- Loosely coupled audio channels (speaker and background music)
  - Max allowable skew: 500ms
Conclusion

- Carefully analyze what kind of synchronization is needed in your multimedia system and application
- Determine at which level you need synchronization
- Determine what the synchronization requirements should be based on prior experiments
Reference Models

- We need reference models to
  - Understand various requirements for multimedia sync
  - Identify and structure run-time mechanisms to support execution of sync
  - Identify interface between run-time mechanisms
  - Compare system solutions for multimedia sync
Sync multimedia objects are classified according to:

- Media level
- Stream level
- Object level
- Specification level
Media Level (1)

- Each application operates single continuous media streams composed of sequence of LDUs
- Assumption at this level: device independence
- Supported operations at this level:
  - read(devicehandle, LDU)
  - write(devicehandle, LDU)
window = open("videodevice");
movie = open("file");
while (not EOF (movie) ) {
    read(movie, &LDU);
    if (LDU.time == 20)
        printf("Subtitle 1");
    else if (LDU.time == 26)
        printf("Subtitle2");
    write(window, LDU);  }
close(window);
close(movie);
Stream Level (1)

- Operates on continuous media streams and groups of streams
- Models inter-stream synchronization for need of parallel presentation
- Offers abstractions:
  - notion of streams,
  - timing parameters concerning QoS for intra-stream and inter-stream synchronization
Stream Level (2)

- Supports operations:
  - Start(stream), stop(stream), create-group(list-of-streams);
  - Start(group), stop(group);
  - Setcuepoint(stream/group, at, event);

- Classifies implementation according to
  - Support for distribution (end-to-end, local)
  - Support of type of guarantees (best effort, deterministic)
  - Support of types of supported streams (analog, digital)
Object Level (1)

- Operates on all types of media and hides differences between discrete and continuous media
- Offers abstractions:
  - Complete sync presentation
- Computes and executes complete presentation schedules that include presentation of non-continuous media objects and calls to stream level
- Does not handle intra-stream and inter-stream synchronization
  - (relies on media and stream levels)
Object Level (2) - Example

- MHEG – Multimedia Hypermedia Experts Group of ISO
  - Defines representation and encoding of multimedia and hypermedia objects
  - Provides abstractions suited to real-time presentations
    - implemented via multimedia synchronization functionalities
  - Provides abstracts for real-time exchange
    - implemented with minimal buffering
  - Evaluates status of objects and performs actions (e.g., prepare, run, stop, destroy)
    - For time-dependent streams – access to stream level
    - For time-independent streams – direct access the object to present it

- Classification of this level according to (a) distribution capabilities, (b) type of presentation schedule, (c) schedule calculation
Specification Level

- Open layer included in tools which allow to create sync specifications

- Examples:
  - Synchronization editors, document editors, authoring systems, conversion tools
  - Examples of such tools: multimedia document formatter that produces MHEG specifications

- Classification:
  - Interval-based spec
  - Time-axes based spec
  - Control flow-based spec
  - Event-based spec
Synchronization in Distributed Environments

- Information of synchronization must be transmitted with audio and video streams, so that receiver(s) can synchronize streams.

- Sync information can be delivered before start of presentation (used by synthetic synchronization)
  - Advantage: simple implementation
  - Disadvantage: presentation delay

- Sync information can be delivered using separate sync channel - out-band (used by live synchronization)
  - Advantage: no additional presentation delay
  - Disadvantage: additional channel needed
Sync in Distributed Environments

- Sync information can be delivered using multiplexed data streams - in-band sync
  - Advantage: related sync information is delivered together with media units
  - Disadvantage: difficult to use for multiple sources
Location of Sync Operation

- Sync media objects by combining objects into new media object
- Sync operation placed at sink
  - Demand on bandwidth is larger because additional sync operations must be transported
- Sync operation placed at source
  - Demand on bandwidth smaller because streams are multiplexed according to sync requirements
Clock Synchronization

- Sync accuracy depends on clocks at source and sink nodes
  - \( T_a = T_{av} - N_{la} - O_a \)
  - \( T_v = T_{av} - N_{lv} - O_v \)

- End-to-end delay
  - \( N_{la} = EED_a = T_{av} - T_a - O_a \)
  - \( N_{lv} = EED_v = T_{av} - T_v - O_v \)
  - \( EED_a = (T_{a1} - T_{a2})/2 \)

- NTP (Network Time Protocol)

---

**Important for resource coordination**

```plaintext
Source Audio
Ta, Oa

Sink Audio/Video
Tv, Ov

Oa, Ov = clock offset
Nla, Nlv = network delay
```
Other Sync Issues

- Sync must be considered during object acquisition
- Sync must be considered during retrieval
  - Sync access to frames of stored video
- Sync must be considered during transport
  - If possible use isochronous protocols
- Sync must be considered at sink
  - Sync delivery to output devices
- Sync must consider support of functions such as pause, forward, rewind with different speeds, direct access, stop or repeat
Sync Specification Methods - Requirements

- Object consistency and maintenance of sync specifications
  - Media objects should be kept as one LDU in spec
- Temporal relations must be specify-able
- Easy Description of Sync Relations
- Definition of QoS requirements
- Integration of time-dependent and independent media
- Hierarchical levels of synchronization
Models

- Interval
- Timeline
- Hierarchical
- Reference points
- Petri net
- Event-based

Common threads
- provide language to express relationships
- runtime system to monitor relationships
- policies to enforce relationships
Interval-based Specification (1)

- Presentation duration of an object is specified as interval
- Types of temporal relations:
  - A before B, A overlaps B, A starts B, A equals B, A meets B, A finishes B, A during B
- Enhanced interval-based model includes 29 interval relations, 10 operators handle temporal relations (e.g., before(δ₁),…)}
Interval Model (2)

- 13 relationships between two intervals

### Before
- A
- B

### Meets
- A
- B

### Equal
- A
- B

### Overlaps
- A
- B

### Starts
- A
- B

### Ends
- A
- B

### During
- A
- B
Example (3)

Audio1 while(0,0) Video
Audio1 before(0)
RecordedInteraction
RecordedInteraction before(0) P1
P1 before(0) P2
P2 before(0) P3
P3 before(0) Interaction
P3 before(0) Animation
Animation while(2,5) Audio2
Interaction before(0) P4
Interval-based Specification (4)

Advantages:
- Easy to handle open LDUs (i.e., user interactions)
- Possible to specify additional non-deterministic temporal relations by defining intervals for durations and delays
- Flexible model that allows specification of presentations with many run-time presentation variations
Interval-based Specification (5)

- Disadvantages:
  - Does not include skew spec
  - Does not allow specification of temporal relations directly between sub-units of objects
  - Flexible spec leads to inconsistencies
    - Example:
      - A NOT in parallel with B
      - A while(2,3) I
      - I before(0) B
Timeline Axis-based Specification

- Presentation events like start and end of presentation are mapped to axes that are shared by presentation objects
- All single medium objects are attached to time axis that represents abstraction of real-time
- This sync specification is very good for closed LDUs
Timeline Model (2)

- Uses a single global timeline
- Actions triggered when the time marker reaches a specific point along timeline
Example (3)

- Define a timed sequence of images, each image has a caption that goes with it.

![Diagram showing time sequence with images and captions]
Example (4)

- Rule language
  - At (t1), show (I1, C1)
  - At (t2), show (I2, C2)
  - At (t3), show (I3, C3)

- Visual environment

![Visual environment](image-url)
Time-Axis-based Spec (based on Virtual Axis)

- Introduction of virtual axis – generalization of global time axis approach
- Possible to create coordinate system with user-defined measurement units
- Mapping of virtual axes to real axes done during run-time
Control Flow-based Spec - Hierarchical Model (1)

- Possibility to specify concurrent presentation threads at predefined points of presentation

- Basic hierarchical spec types:
  - Serial synchronization
  - Parallel synchronization of actions

- Actions: atomic or compound
  - Atomic action handles presentation of single media object, user input, delay
  - Compound actions are combinations of sync operators and atomic actions
  - Delay is atomic action – allows modeling of delays in serial presentations
Example (3)

- Narrated slide show
  - image, text, audio on each slide
  - select link to move to the next slide

```
S1 A1 T1 I1 S2 A2 T2 I2 ...
```
Example (4) (and Comparison with Interval-based Spec)

Audio1 while(0,0) Video
Audio1 before(0)
RecordedInteraction
RecordedInteraction before(0) P1
P1 before(0) P2
P2 before(0) P3
P3 before(0) Interaction
P3 before(0) Animation
Animation while(2,5) Audio2
Interaction before(0) P4
Advantages
- Easy to understand
- Natural support for hierarchies
- Integration of interactive object easy

Disadvantage
- Need additional descriptions of skews and QoS
- No duration description

Some synchronization scenarios cannot be described
Control Flow-based Spec – Reference Points (1)

- Time-dependent single medium objects are regarded as sequences of closed LDUs
- Start/stop times of object presentation are reference points
- Connected reference point is synchronization points
- Temporal relations specified between objects without explicit reference to time
Control Flow-based Spec – Reference Points (3)

- **Advantages:**
  - Sync at any time during presentation of objects
  - Easily integrated object presentation with unpredictable duration
  - Intuitive type of synchronization spec

- **Disadvantages:**
  - Not easy way to detect inconsistencies
  - Cannot specify delays in presentation
Event-based Specification

- Presentation actions initiated by synchronization events
  - Example:
    - Start presentation
    - Stop presentation
    - Prepare presentation
  - Events initiating presentation
    - External or internal
Event-based Spec

- **Advantage:**
  - Easily extended to new sync types
  - Easy integration of interactive objects

- **Disadvantage:**
  - Difficult to handle in case of realistic scenarios
  - Too complex specification
  - Need separate description of skew/QoS
  - Difficult use of hierarchies
Event Model (Nsync)

- Associate *actions* with *expressions*
- Expressions may contain scalars, clocks, variables, relations, and connectives
- When the expression becomes TRUE, invoke associated action

```
When “Time > Q.end + 5 && !Response” Answer=WRONG
```

Background and Time Model

- Each media object attached to a clock
- Clock implements logical time
  - Media-time = Speed * Real-Time + Offset
- Speed (S) – ratio of media-time progression to that of real-time
  - E.g., a speed of 2.0 for cont. media indicates that the media is being played at twice its normal playout rate
- Express temporal behavior as relationships among clocks
- Interactive events tied to variables
Model Specification

When “Narration >= Overview && !MoreInfo”       NextSlide
When “Narration >= Overview && MoreInfo”       PlayDetails
When “Narration >= Overview + Details”       NextSlide

Narration: narration’s logical timeline
Overview: normal transition point
Details: additional narrative details
MoreInfo: records kitchen info status
Reactive Interface
Model Specification

When “Video >= 0 && Video < T1”
Select Kitchen

When “Video >= T1 && Video < T2”
Select Deck

When “Video >= T2 && Video <= T3”
Select Yard