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

Synchronization Reference Model

- 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)

Media Level (2) - Example

```
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
  - 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
    - More later (few slides)

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

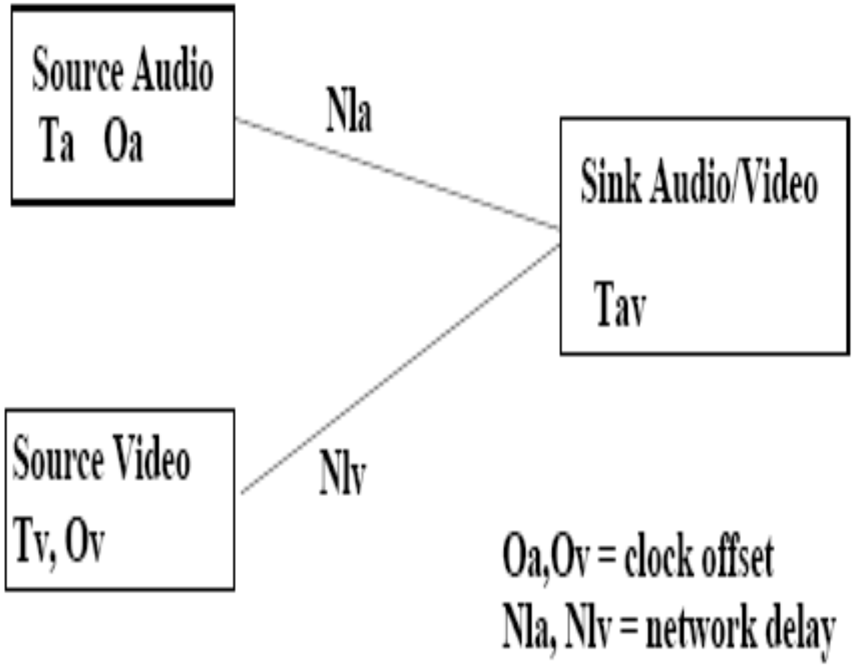
Location of Sync Operation

- 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
  - $T_a = T_{av} – NI_a – O_a$
  - $T_v = T_{av} – NI_v – O_v$
- End-to-end delay
  - $NI_a = EED_a = T_{av}\text{-}T_a\text{-}O_a$
  - $NI_v\text{=}EED_v = T_{av}\text{-}T_v\text{-}O_v$
  - $EED_a = (T_{a1}\text{-}T_{a2})/2$
- NTP (Network Time Protocol)

important for resource coordination



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

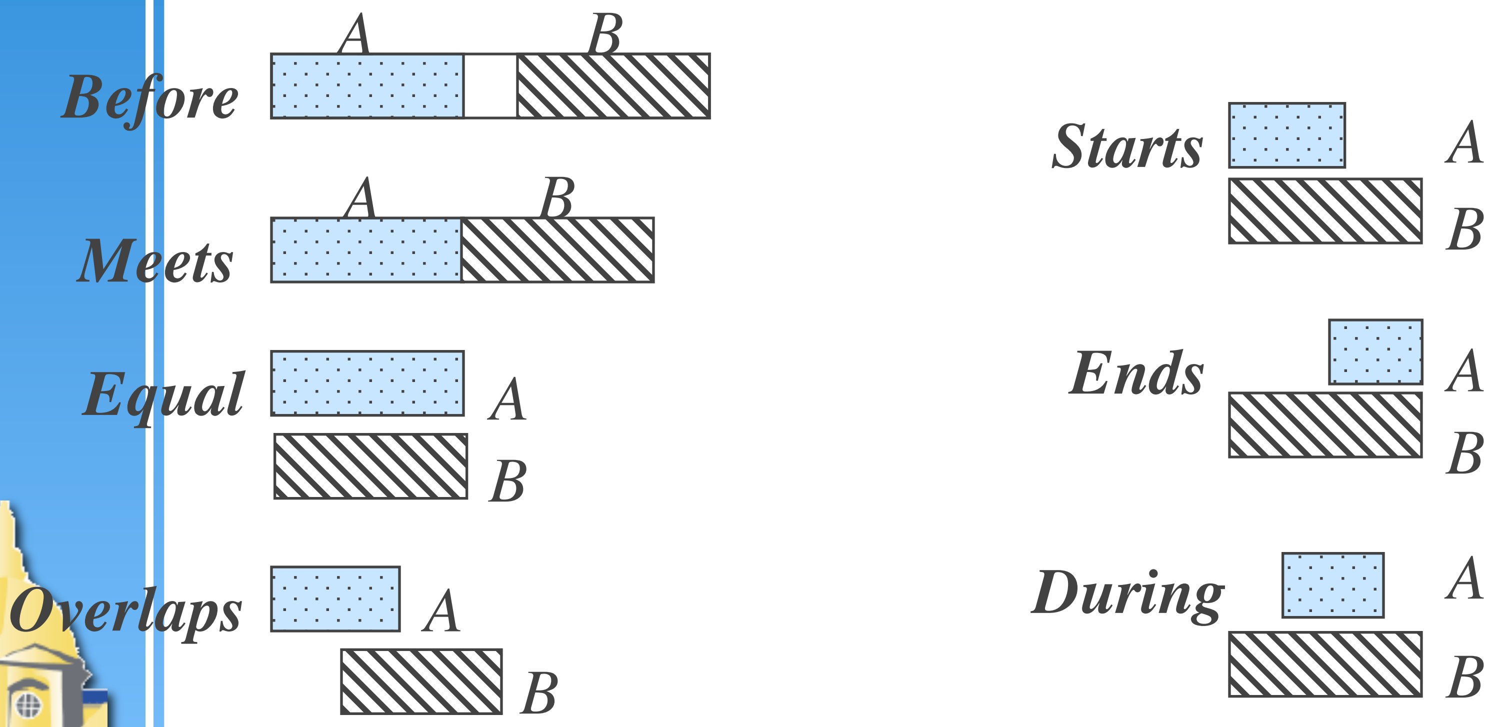
- Common threads
  - provide language to express relationships
  - runtime system to monitor relationships
  - policies to enforce relationships

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



Example (3)

Audio1 while(0,0) Video

Audio1 before(0) RecordedInteraction

RecordedInteraction before(0) P1

P1 before(0) P2

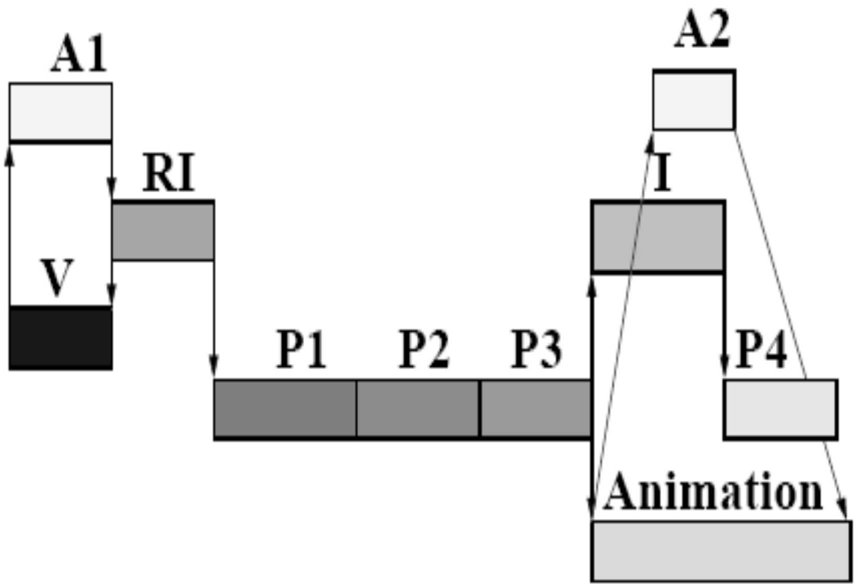
P2 before(0) P3

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

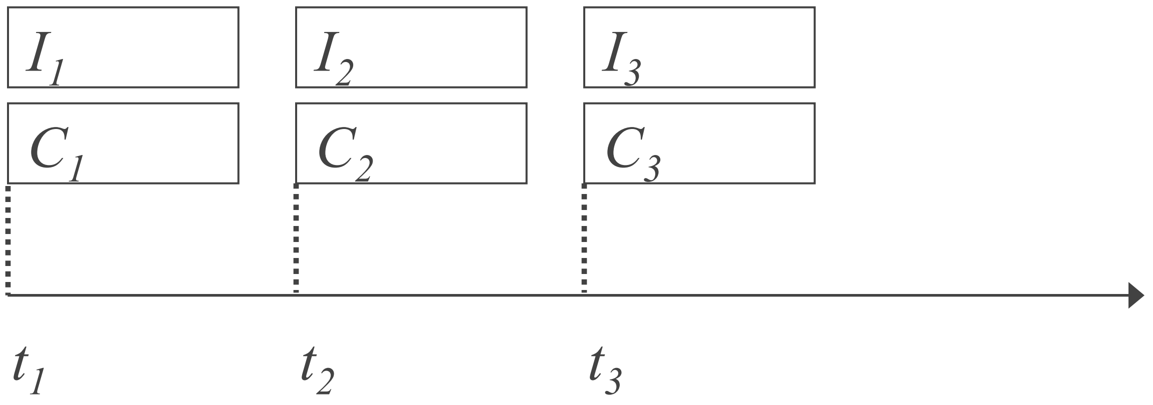
- Presentation events like start and end of presentation are mapped to axes that are shared by presentation objects
- All single medium objects are attached to time axis that represents abstraction of real-time
- This sync specification is very good for closed LDUs

Timeline Model (2)

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

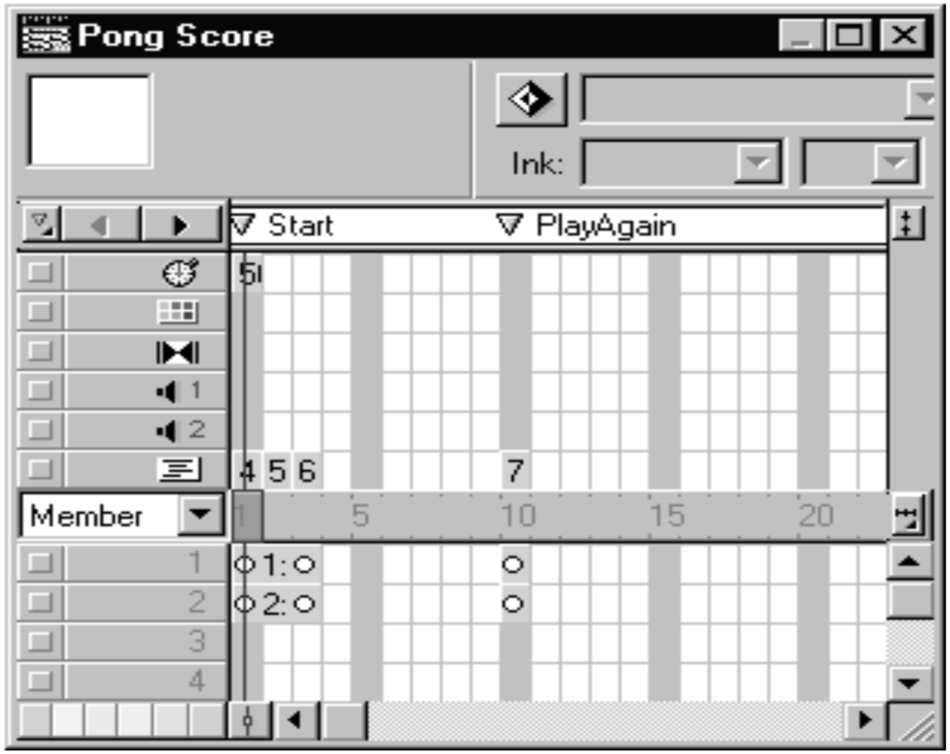
Example (3)

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



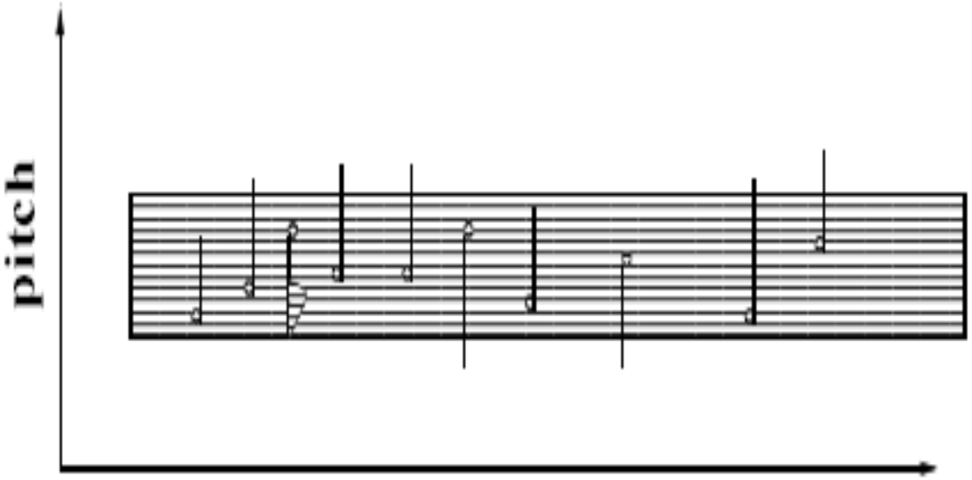
Example (4)

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



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



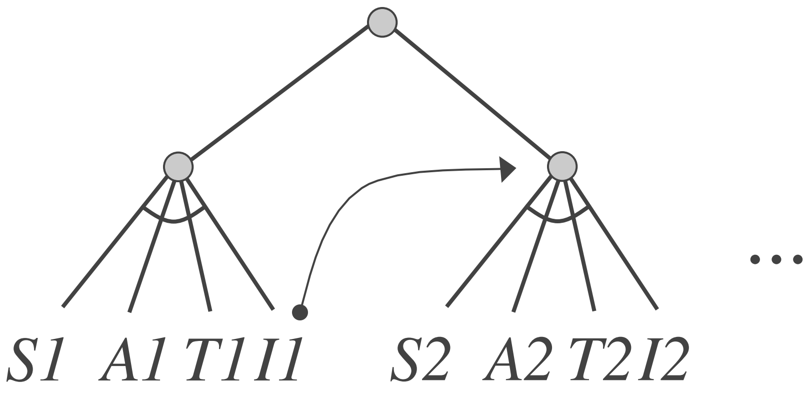
Virtual axis with measurment unit beat

Control Flow-based Spec - Hierarchical Model (1)

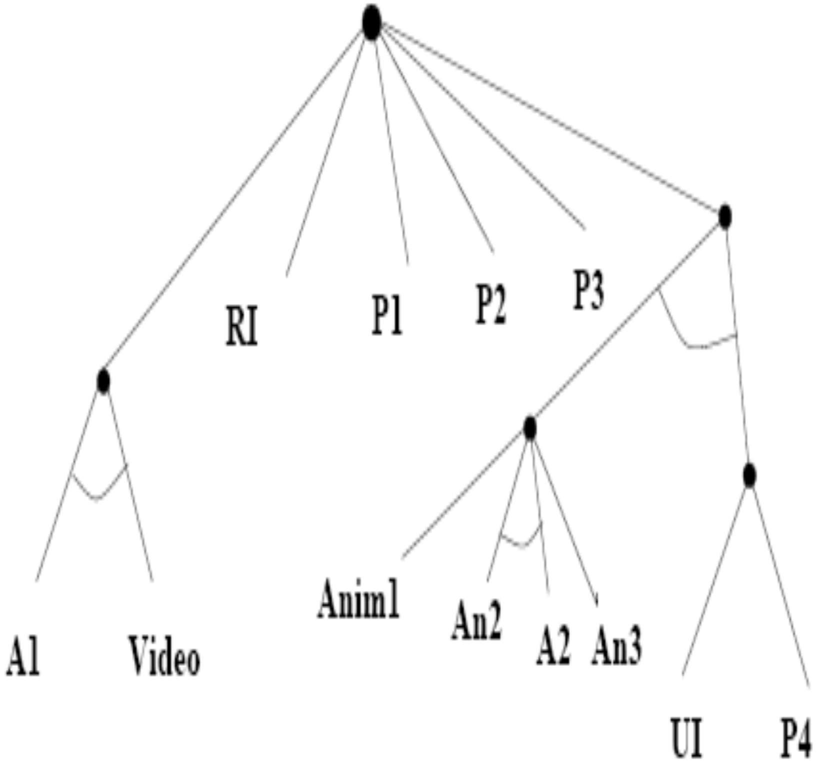
- Possibility to specify concurrent presentation threads at predefined points of presentation
- Basic hierarchical spec types:
  - Serial synchronization
  - Parallel synchronization of actions
- Actions: atomic or compound
  - Atomic action handles presentation of single media object, user input, delay
  - 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



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

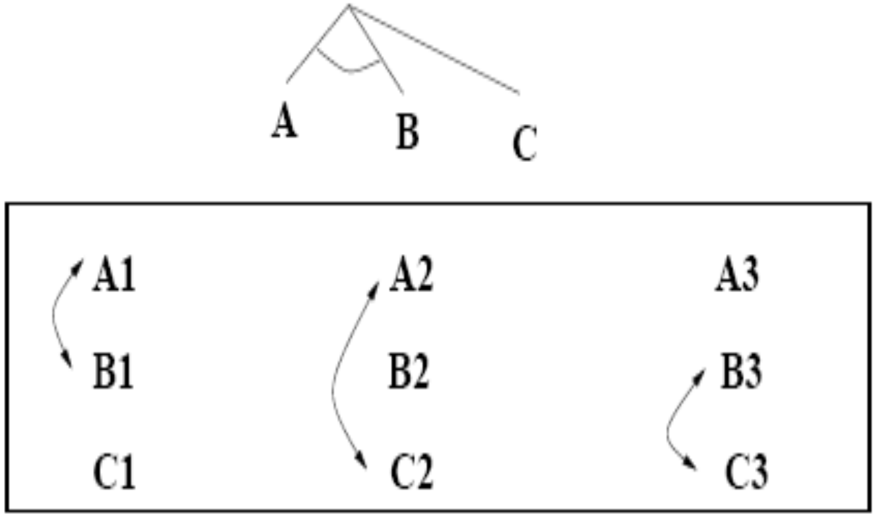
Animation while(2,5) Audio2

Interaction before(0) P4

Control Flow-based Spec – Hierarchy (5)

- Advantages
  - Easy to understand
  - Natural support for hierarchies
  - Integration of interactive object easy
- Disadvantage
  - Need additional descriptions of skews and QoS
  - No duration description

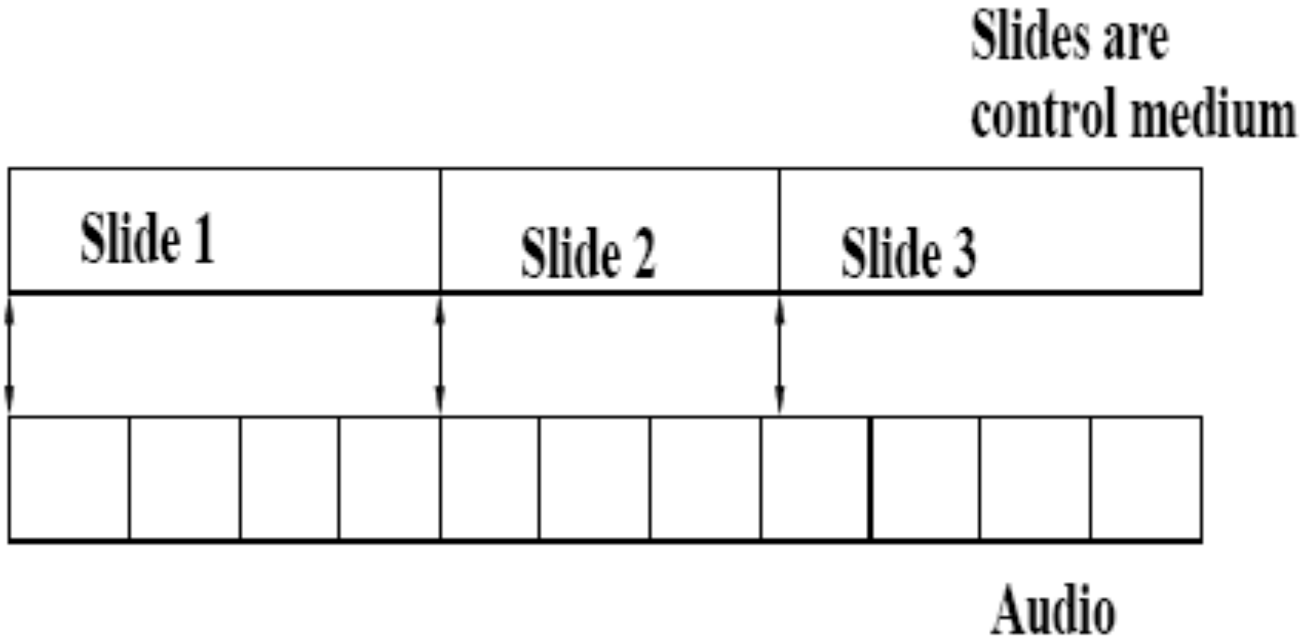
Some synchronization scenarios cannot be described



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

Example (2)



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

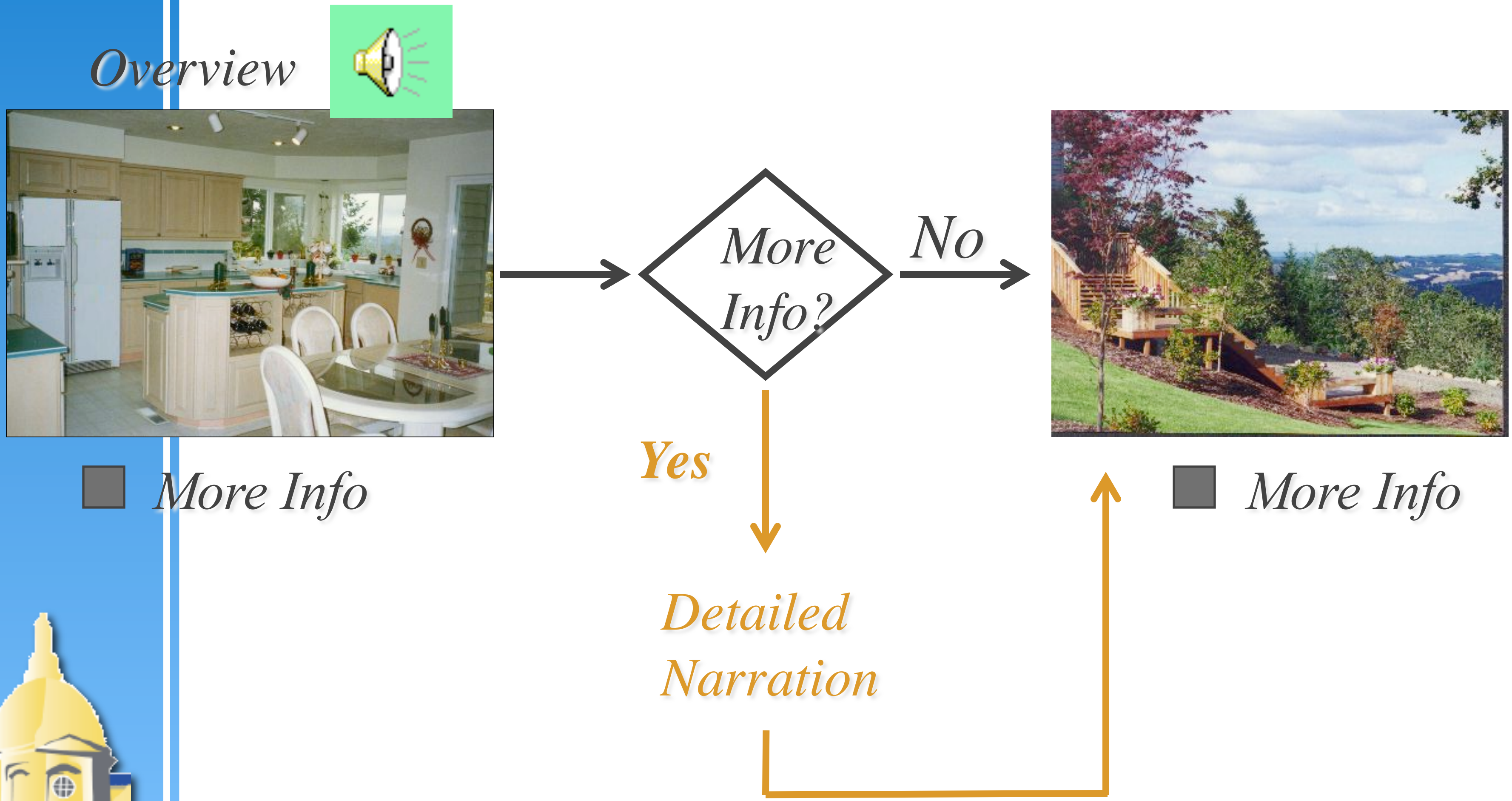
```
When “Time > Q.end + 5 &&
      !Response” Answer=WRONG
```

Source: B. Bailey et al. “Nsync- A Toolkit for Building Interactive Multimedia Presentations”, ACM Multimedia 1998

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

Example: Delayed Transition



Overview

□ *More Info*

More Info?

No

Yes

Detailed Narration

□ *More Info*

Model Specification

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

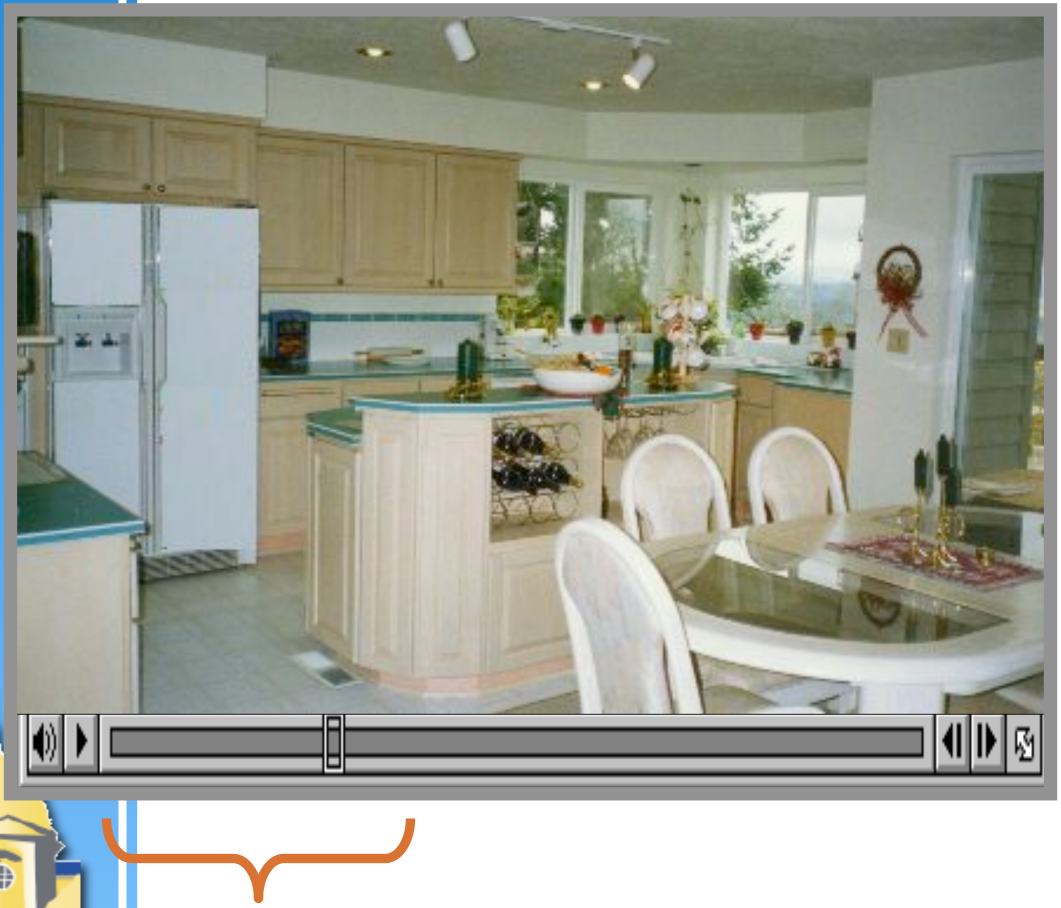
`Narration`: narration’s logical timeline

`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
```