CS 414 – Multimedia Systems Design

Lecture 37 – Synchronization (Part 4)

Klara Nahrstedt

Spring 2008

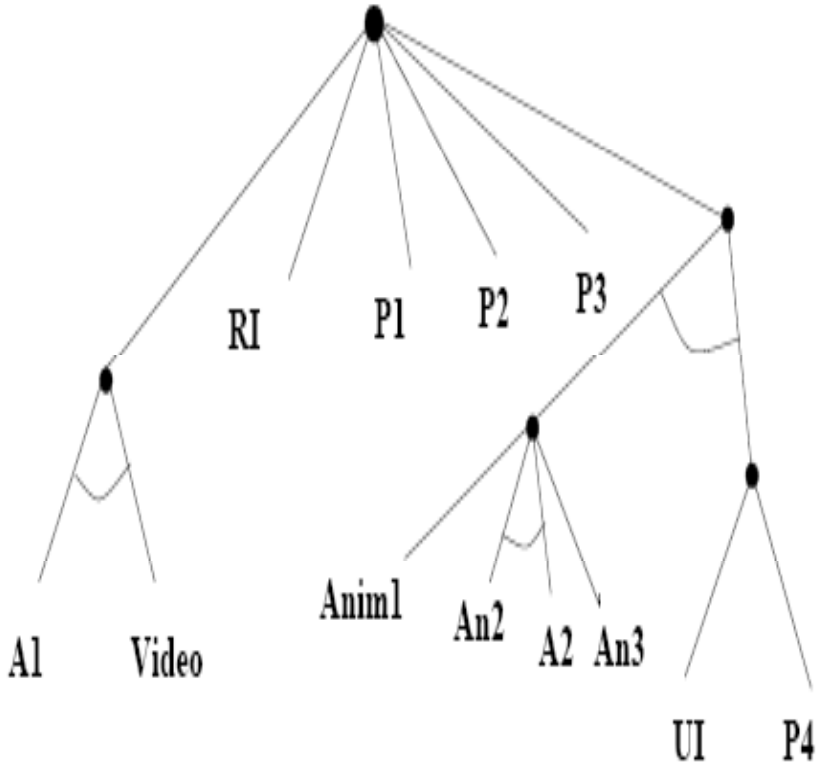
Administrative

- 4/18 - HW2 posted
- 4/25 - Deadline for HW2 (midnight deadline)
  - Delivery:
    - Slide paper forms of your under the door of 3104
    - Give the homework solutions to the secretary in 3120 SC between 10am and 4:30pm
    - Email in pdf format to klara@cs.uiuc.edu and wconner@cs.uiuc.edu
- 4/28 – review session, preparation for final exam

Outline

- Synchronization Specification Methods
  - Control Flow-based Specification
    - hierarchical approach
    - Timed Petri nets
  - Event-based Specification
    - Nsync Presentation System

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

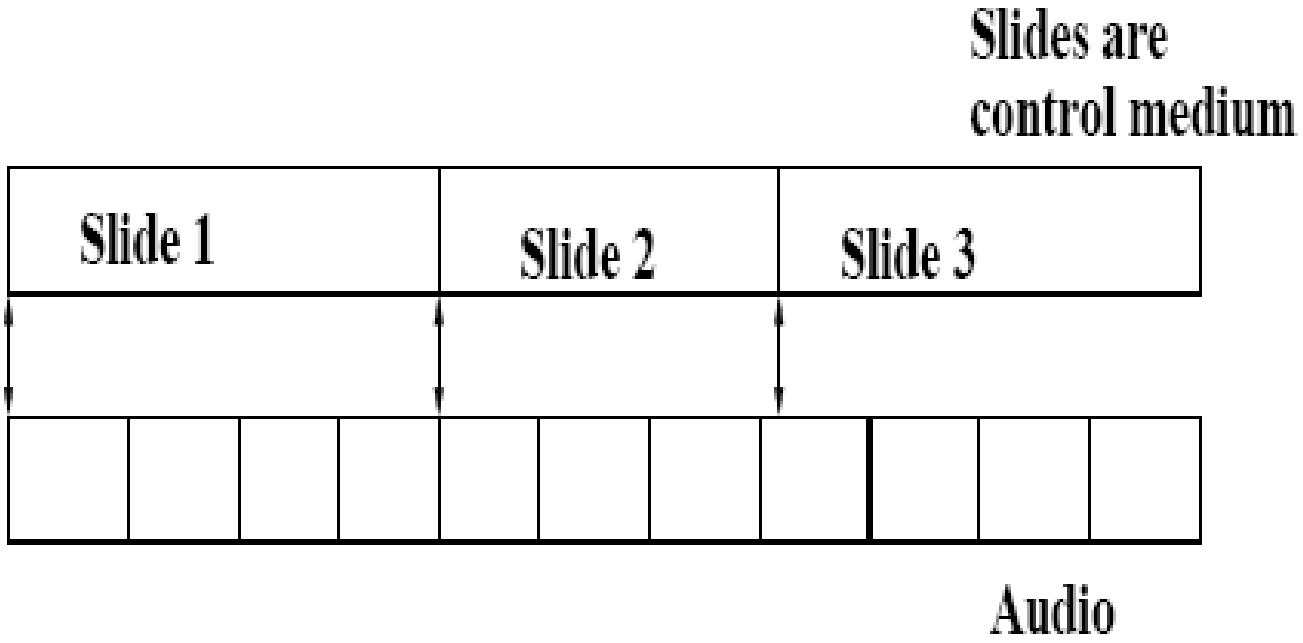
Animation while(2,5) Audio2

Interaction before(0) P4

Control Flow-based Spec – 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)

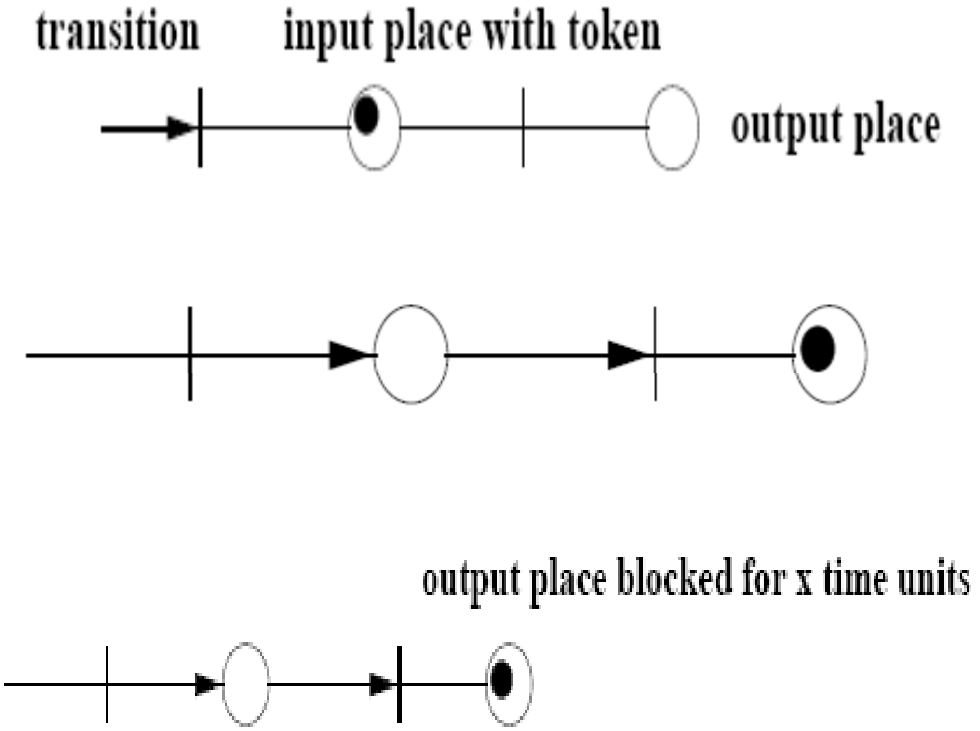


Control Flow-based Spec – Reference Points (3)

- 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

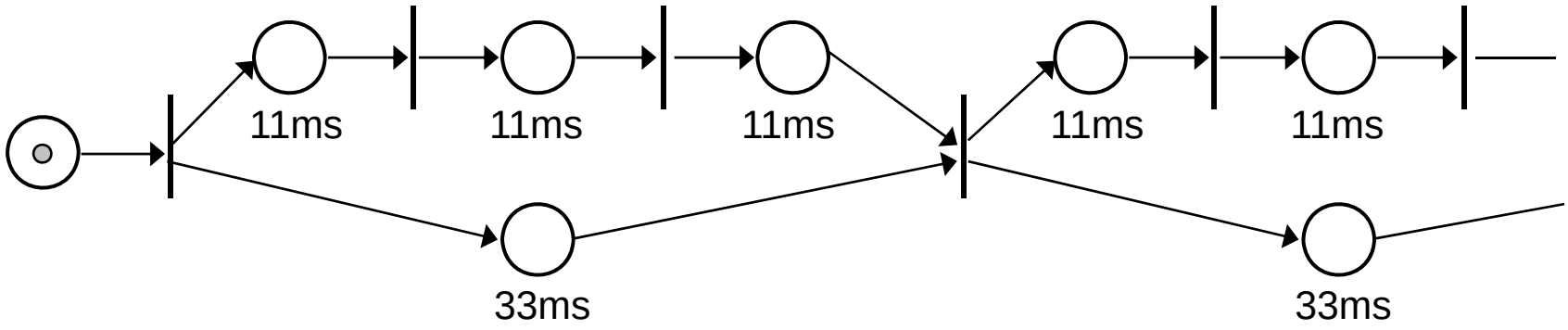
Control Flow-based Specification – Timed Petri Nets

- Petri Nets obey Rules
  - Transition fires, if all input places contain non-blocking token
  - If transition fires, token is removed from each input place and token is added to each output place
  - Token that is added to new place is blocked for duration assigned to this place



Example

Specify audio video synchronization



Timed Petri Nets

- Advantage:
  - Allow all kinds of sync specification
- Disadvantage:
  - Very complex specifications since media object must be split into sub-objects

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

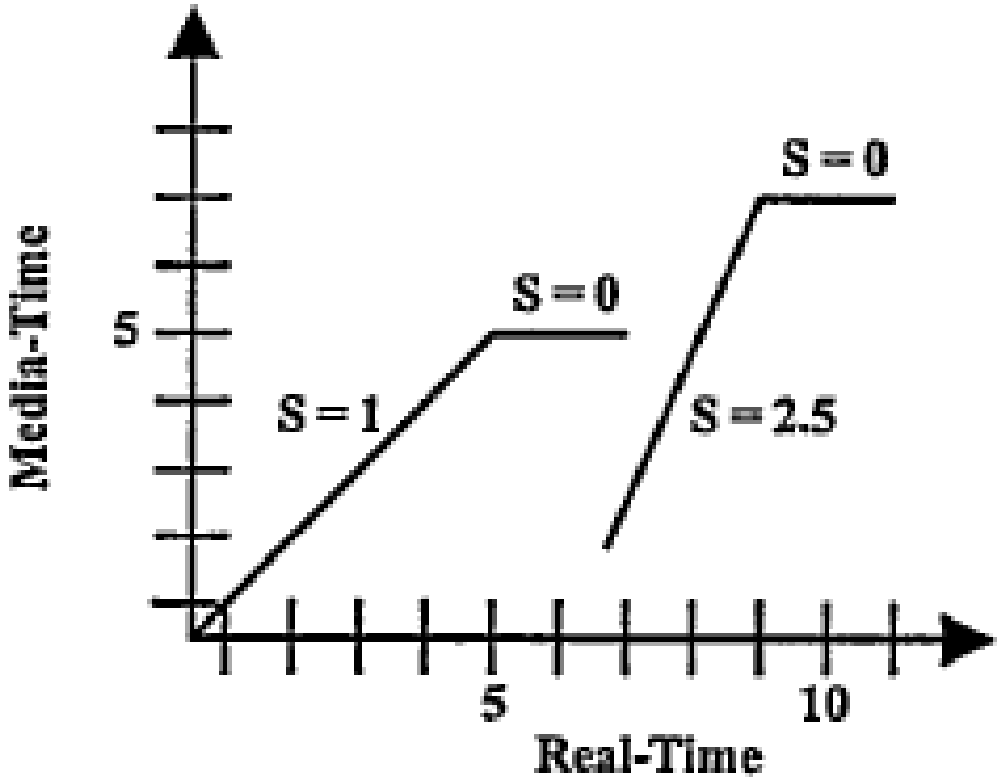
```
When “Time > Q.end + 5 &&
      !Response” Answer=WRONG
```

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

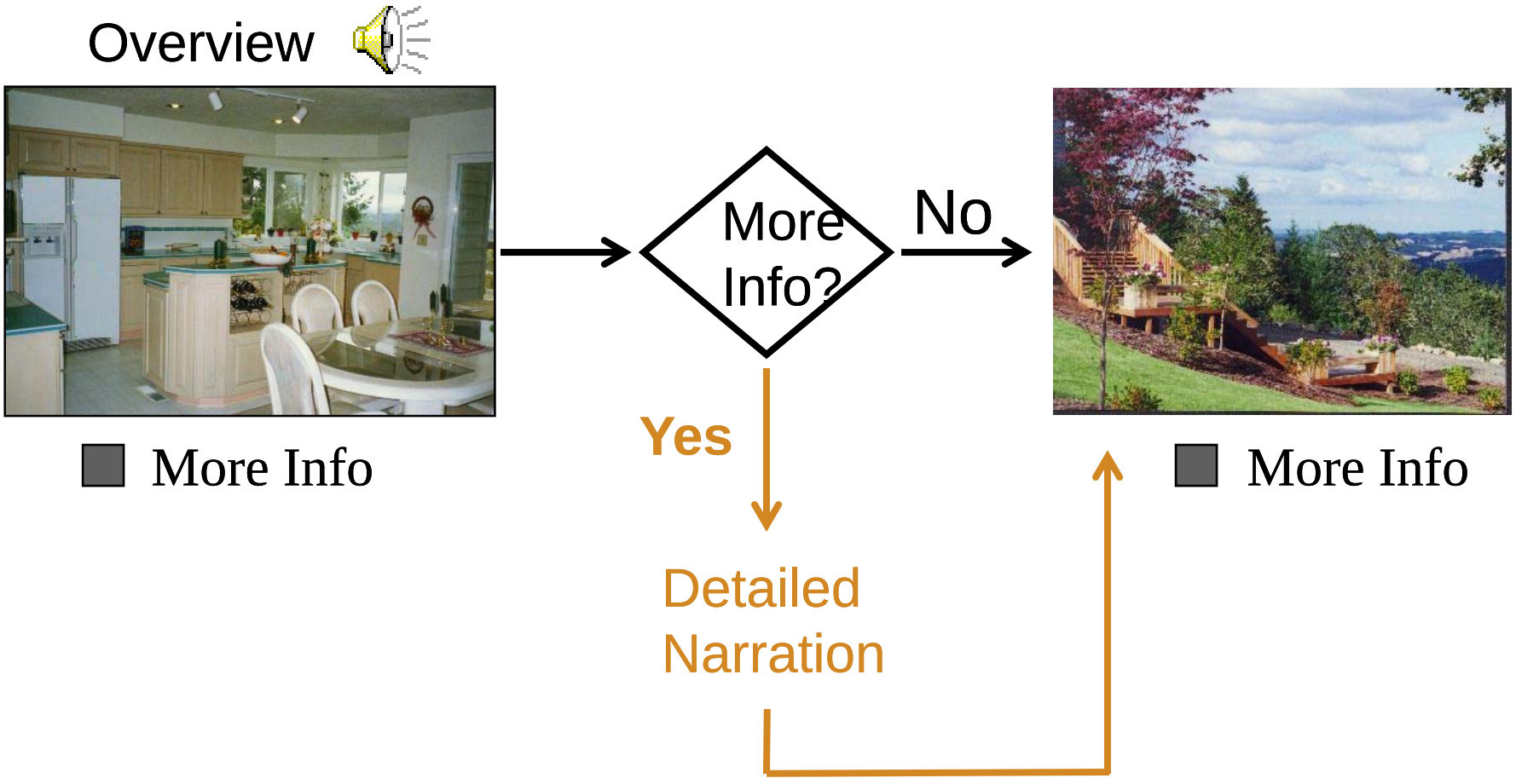
Background and Time Model

- 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

Graph Demonstrating Media-time Mapping



Example: Delayed Transition



Model Specification

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

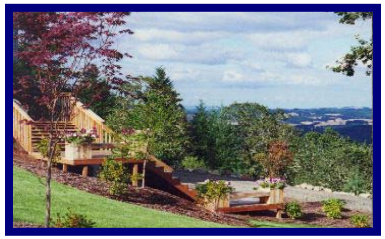
`Narration`: narration’s logical timeline

`Overview`: normal transition point

`Details`: additional narrative details

`MoreInfo`: records kitchen info status

Reactive Interface



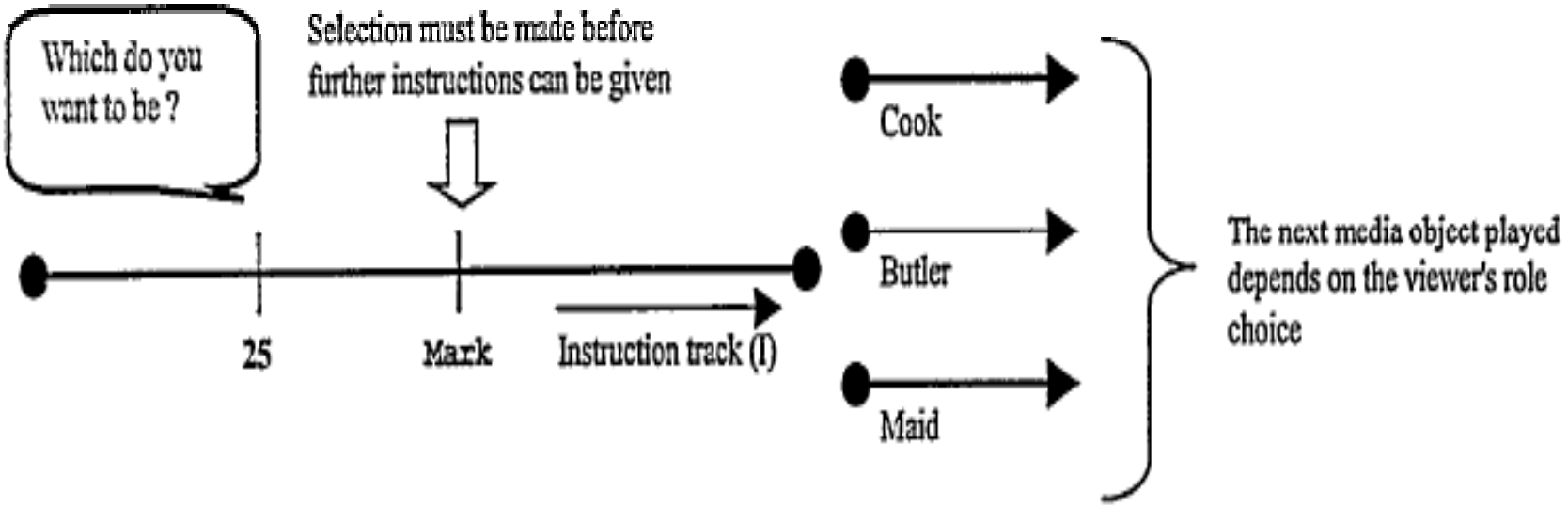
Model Specification

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

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

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

Example (Virtual Murder Mystery)




```
When {I.value >= 25}                    { display_choose_role_dialog   }
When {I.value < 25}                     { undisplay_choose_role_dialog }
When {I.value >= Mark && role_chosen}   { I.speed = 1; undisplay_choose_role_dialog }
When {I.value >= Mark && !role_chosen}  { I.value = Mark; I.speed = 0; display_choose_role_dialog }
When {I.value == I.end}                 { $role_chosen.speed = 1 }
```

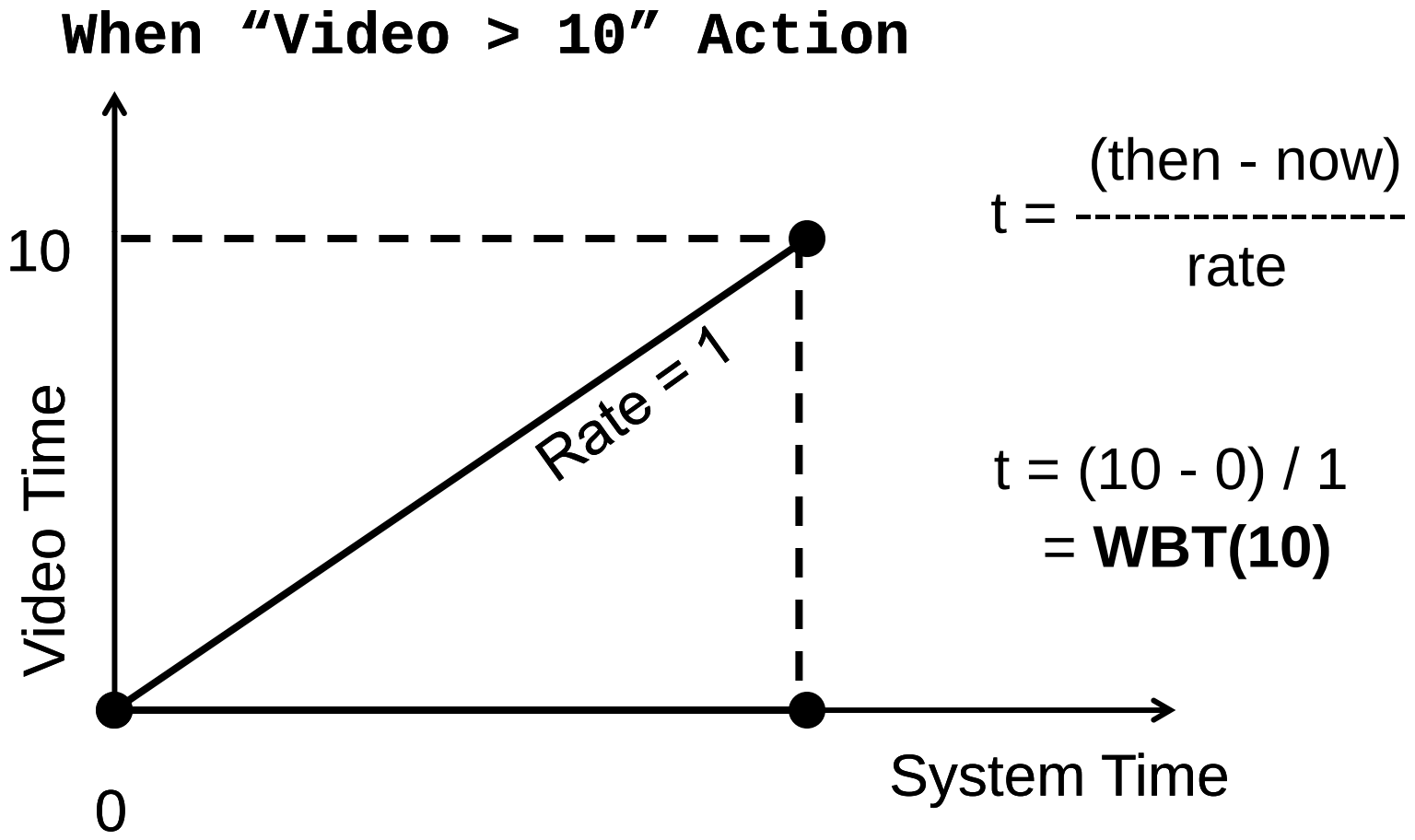
Expression Evaluation

- Propositional logic breaks down
  - returns logic value only at *present* time
  - requires polling to catch future transitions
- *Predictive* logic
  - returns logic value at present time along with a prediction of any future transition
  - eliminates need for intermittent polling/timers

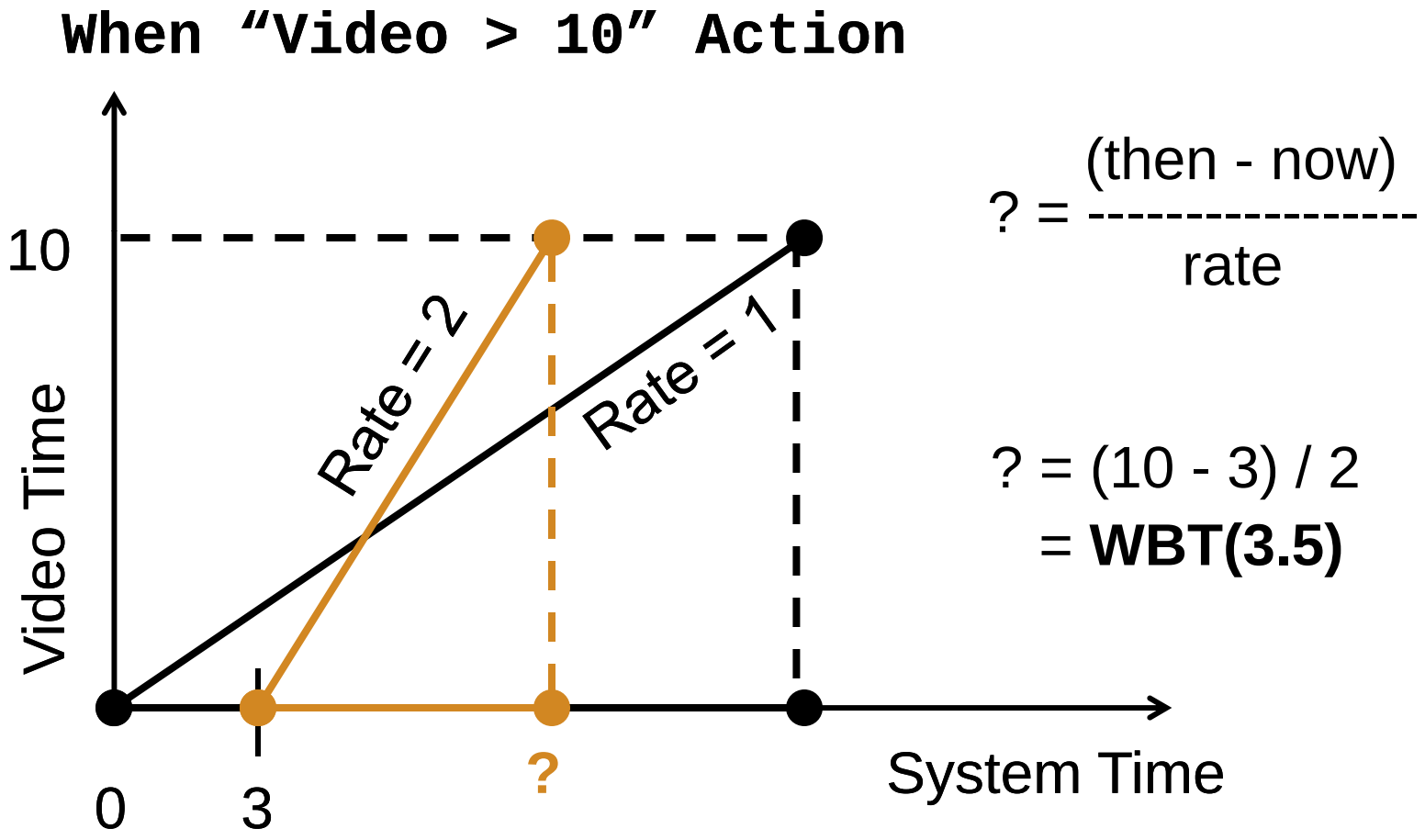
Predictive Logic States

- WBT(t) False now, but *Will Become True* at future time t
- WBF(t) True now, but *Will Become False* at future time t
- t represents an offset from the current system time (real-time), not media-time

Prediction Example



Prediction Example



Evaluation Rules for “AND”

WBT(x) && WBT(y) = WBT(max(x, y))

WBF(x) && WBF(y) = WBF(min(x, y))

WBF(x) && WBT(y) =

FALSE if (x < y)

WBT(y) then WBF(x) otherwise

Pros

- Complements current languages
  - adds ability to express combinations of interactive and temporal behavior
  - syntax can easily be translated into mark up
- Predictive logic useful in run-time engines
  - eliminates need for polling/timers
  - enables look-ahead pre-fetching

Cons

- Difficult to visualize rule propagation
  - makes system difficult to debug
- Rules are not groups into hierarchies
  - enable divide and conquer strategy
- Lack of scope
  - all rules always active
  - guard actions with complex expressions

Take Home Exercise

- Be able to model relationships within relatively simple applications
- Weigh tradeoffs between models