

Vorlesung / Übungen

Multimedia Technologie II

Prof. Dr. Klaus Hering

SS 2004

HTWK Leipzig, FB IMN

Lösungen zum Übungsblatt 2

1. das Vorkommen des Elements **DDD**:

/AAA/DDD

2. alle Elemente **CCC**:

//CCC

Welche Dokumentteile werden durch **//CCC/DDD** beschrieben?

alle Elemente DDD, die Kinder eines Elements CCC sind

3. alle Elemente, die genau 2 Vorfahren (*ancestors*) haben:

/*/*/*

alle Elemente, die **BBB** als *parent* besitzen:

//BBB/*

4. alle Elemente, die an Position 1 einer Elementliste (Liste von Elementen mit gleichem *parent*-Element oder Liste, die nur das *root*-Element enthält) stehen:

/*[1]

5. alle Attribute:

//@*

Lösungen zum Übungsblatt 2

6. alle Elemente, die einen von 'bbb' verschiedenen (!=) Wert für das Attribut *name* besitzen:
`//*[@name != 'bbb']`
7. alle Elemente, die genau einen Vorfahren (*ancestor*) und genau zwei Kinder (*children*) haben:
`//*[@count(*)=2]`
8. alle Elemente mit genau einem Vorfahren, deren Name ein C enthält:
`//*[@contains(name(),'C')]`
9. alle Elemente, deren Namenslänge größer als 3 ist oder deren Name mit B beginnt (logische Operatoren: **or**, **and**):
`//*[@ string-length(name()) > 3 or starts-with(name(),'B')]`
10. alle Kinder (*children*) von DDD und alle Elemente EEE:
`//DDD/* | //EEE`
12. alle Elemente, die Abkömmlinge (*descendants*) von BBB oder EEE sind
`//BBB/descendant::* | //EEE/descendant::*`

4.2 Beispiele

Die folgenden Beispiele entstammen dem *RealNetworks Production Guide with RealOne Player*. Pfadangaben in Bezug auf den Quellcode erfolgen relativ zu dem beim Entpacken von *pgd.zip* angelegten Directory *samples*.

Beispiel 1: `cliptags/plaintext.smil`

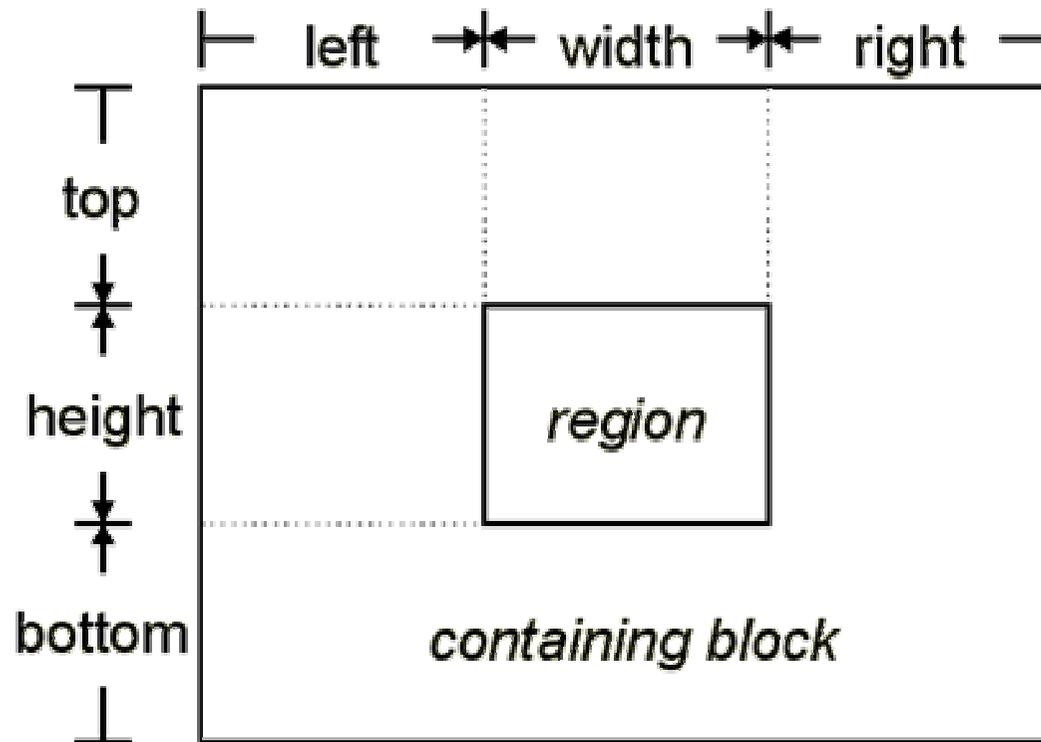
Der einzige einbezogene Clip verkörpert ein einfaches Text-File. Dieses wird angezeigt und die Anzeige „eingefroren“.

Beispiel 2: `cliptags/brushobject.smil`

Brushes verkörpern farbige Rechtecke, welche anstelle eines Clips in einer Region angezeigt werden können. Es werden 5 Regionen definiert (eine entsprechend dem *root-layout* und 4 Quadranten).

Im Rahmen einer <seq>-Gruppe werden die Regionen mit farbigen Brushes belegt. Die Anzeige bleibt bis zum Überschreiben erhalten.

Attribute zur Positionierung von Regionen:



[L. Rutledge]

Beispiel 3: layout/fit{hidden|fill|meet|scroll|slice}.smil

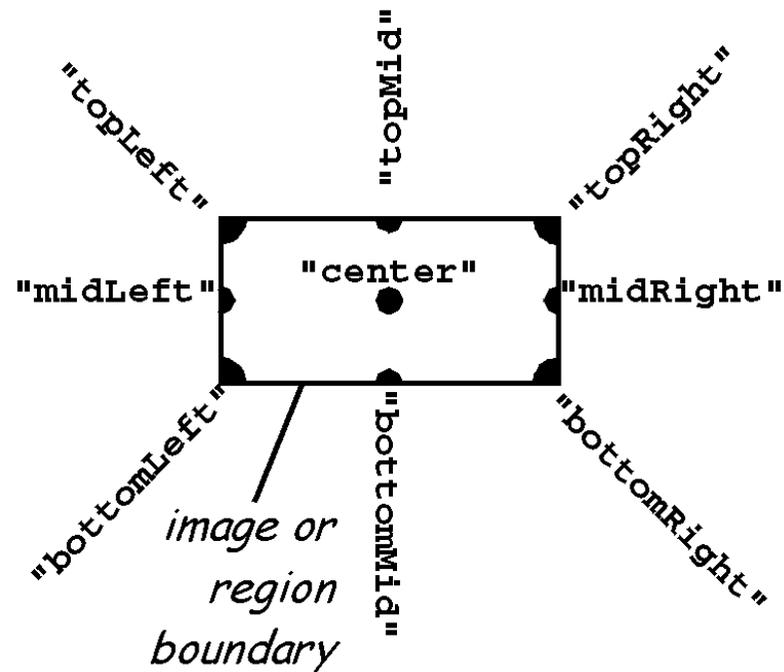
Es wird das „Einpassen“ von Clips in Regionen mit einer abweichenden Größe nach verschiedenen Strategien betrachtet. Dabei wird zunächst standardmäßig die linke obere Clipsecke mit der linken oberen Regionsecke in Übereinstimmung gebracht (sofern kein *registration point* vereinbart ist).

- **hidden:** Abschneiden überstehender Clipbereiche, Füllen nicht überdeckter Regionsbereiche mit Hintergrundfarbe
- **fill:** Skalieren des Clips derart, daß dieser die Regionsgröße annimmt
- **meet:** Skalieren des Clips bei Wahrung des Seitenverhältnisses, bis Größe in einer Dimension mit Region übereinstimmt und in der zweiten Dimension in den Regionsgrenzen liegt
- **scroll:** Anzeige des Clips in Normalgröße, Einsatz von scroll bars bei Überschreitung von Regionsgrenzen
- **slice:** Skalieren des Clips bei Wahrung des Seitenverhältnisses, bis in einer Dimension die Regionsgröße erreicht und in der anderen überschritten wird, Abschneiden des überstehenden Clipbereichs

Beispiel 4: layout/regalign.smil

Es wird das „Einpassen“ von Clips in Regionen unter Verwendung von *registration points* betrachtet. Durch RegPoint-Elemente wird ein Punkt der betreffenden Region definiert und über das *regAlign-Attribut* wird festgelegt, welcher Teil des Clips mit diesem Punkt zusammenfallen soll.

mögliche *alignment values*:



[L. Rutledge]

Zeitaspekte

- Video- und Audioclips besitzen eigene (interne) Zeitachse
⇒ clip-immanente Laufzeit
- für diskrete Medien ist die clip-immanente Laufzeit 0
- clip-immanenter Laufzeit steht explizite Clip-Präsentationszeit gegenüber
⇒ diese kann über *dur-*, *repeatCount-*, *repeatDur-*, *clipBegin-* und *clipEnd-Attribut* explizit beeinflußt werden
- <seq>- und <par>-Elemente mit spezieller Wirkung auf Start- und Endzeit von darin enthaltenen Clips
⇒ innerhalb eines <par>-Elements kann Startzeit für einen Clip explizit oder auch relativ zu einem anderen Element festgelegt werden (*begin-Attribut*)
⇒ <par>-Element erlaubt verschiedene Formen der Ende-Synchronisation (<par>-Ende wenn erstes Element / letztes Element / ein bezeichnetes Element aus dem <par>-Element zum Ende kommt)

Beispiel 5: timing/dursample.smil

Ein Videoclip wird zunächst in voller Länge und danach eingeschränkt durch das *dur-Attribut* gespielt.

Beispiel 6: timing/clipbeginsample.smil

Ausschnitte eines Videos werden durch *clipBegin* und *clipEnd* spezifiziert.

Beispiel 7: timing/repeat1.smil

Ein Videoclip wird dreimal wiederholt. Eine bestimmte Zeit nach dem Start der *i-ten* Iteration wird ein Bildclip b_i eingeblendet, welcher eine festgelegte Zeit vor dem Ende der *i-ten* Iteration wieder ausgeblendet wird. Die Bindung der Start- und End-Zeiten des Bildclips an die Iterationen des Videoclips werden z.B. durch folgende Attributbelegungen im ``-Element realisiert:

`begin="videoclip.repeat(1)+1.5s" end="videoclip.repeat(2)-1s"`

Beispiel 8: groups/unsynched.smil

Innerhalb eines <par>-Elements wird ein Clip völlig aus der Synchronisation (bezüglich des zweiten im <par>-Element eingebundenen Clips) herausgenommen. Innerhalb des betreffenden Elements wird das Attribut *syncBehavior*=“*independent*“ gesetzt.

Beispiel 9: linking/selffragment.smil

Es wird eine Sequenz von 3 Videoclips abgespielt. Zu jedem Zeitpunkt (auch nach dem Ende der Sequenz) kann über einen speziellen Button jeder der 3 Clips neu gestartet werden. Die Realisierung dieses Szenarios erfolgt über den Einsatz von <area>-Elementen innerhalb der -Elemente, welche die Buttons verkörpern. Jeder Button ist über sein <area>-Element mit dem entsprechenden Videoclip “verlinkt“. Die Links sind solange aktiv, wie die Buttons sichtbar bleiben. Die Aktivitätseigenschaft von Links kann an Zeitbedingungen geknüpft werden.

Beispiel 10: switching/languageswitch.smil

Zu einem englischsprachigen Video werden spanische oder französische Untertitel eingeblendet, sofern eine dieser Sprachen als bevorzugte Sprache für den RealOne Player eingestellt ist.

Die Verzeichnisse *animation* und *transitions* enthalten eine Reihe von Beispielen zu Animationen und Übergangseffekten.

Desweiteren kann eine Vielzahl von Beispielen unter

<http://www.w3.org/2001/SMIL20/testsuite/>

gefunden werden.

4.3 Timing und Synchronisation

Hauptelemente (*Time Container*) in SMIL zur Realisierung des zeitlichen Zusammenspiels von Medienobjekten:

<seq>

- Präsentation enthaltener Elemente in der gegebenen Reihenfolge
- Synchronisationsbezugspunkt eines Kindelements ist das Ende der aktiven Phase des in der Sequenz vor ihm liegenden Kind-Elements (für das erste Kind-Element ist es der begin-Zeitpunkt des gesamten <seq>-Elements)
- für Kind-Elemente eines <seq>-Elements darf das *begin-Attribut* lediglich nicht-negative Offsets als Werte haben (und damit zum Beispiel auch keine *event-values* !)
- <sec>-Elemente haben kein *endsync-Attribut*

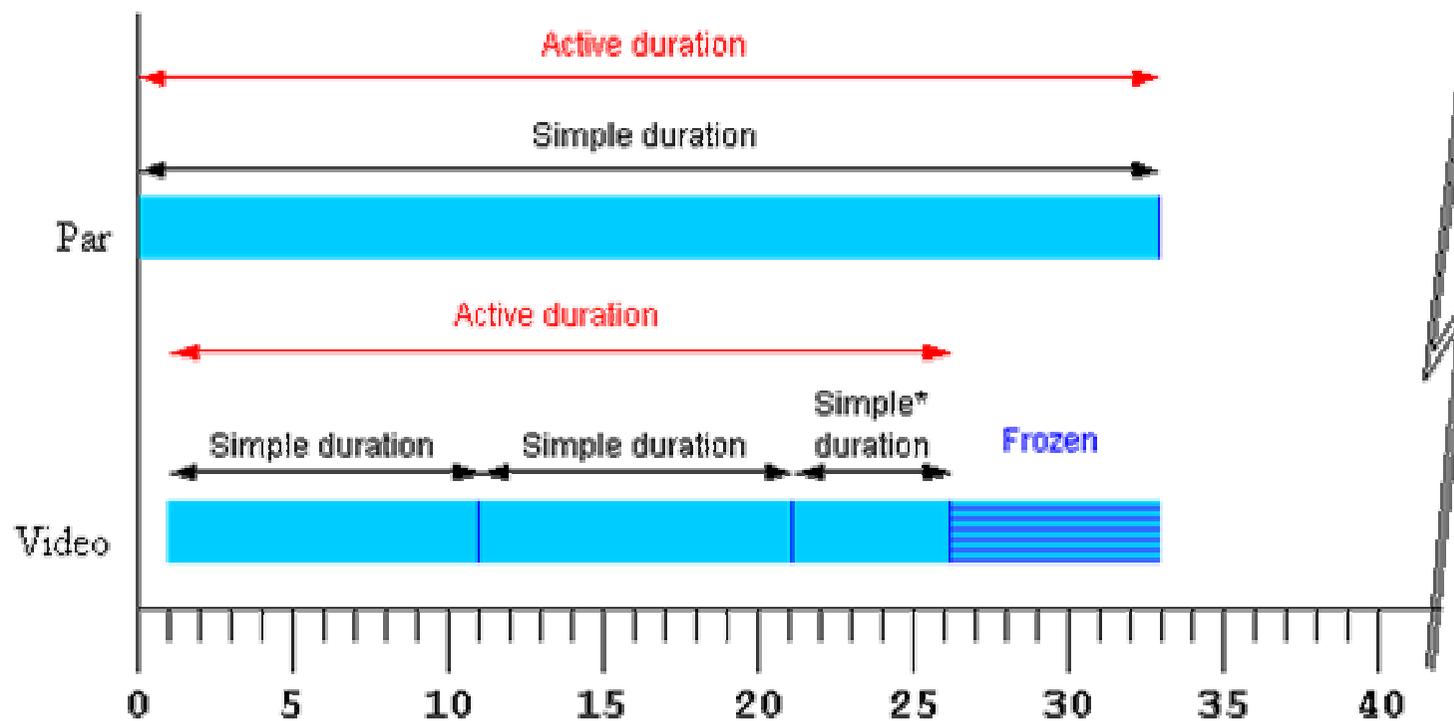
<par>

- simultane Präsentation enthaltener Elemente
- Synchronisationsbezugspunkt für alle Kind-Elemente eines <par>-Elements ist der begin-Zeitpunkt des <par>-Elements

- <par>-Elemente erlauben alle für Elemente vorgesehenen Timing-Mechanismen
- Präsentationsdauer wird implizit über *endsync-Attribut* gesteuert

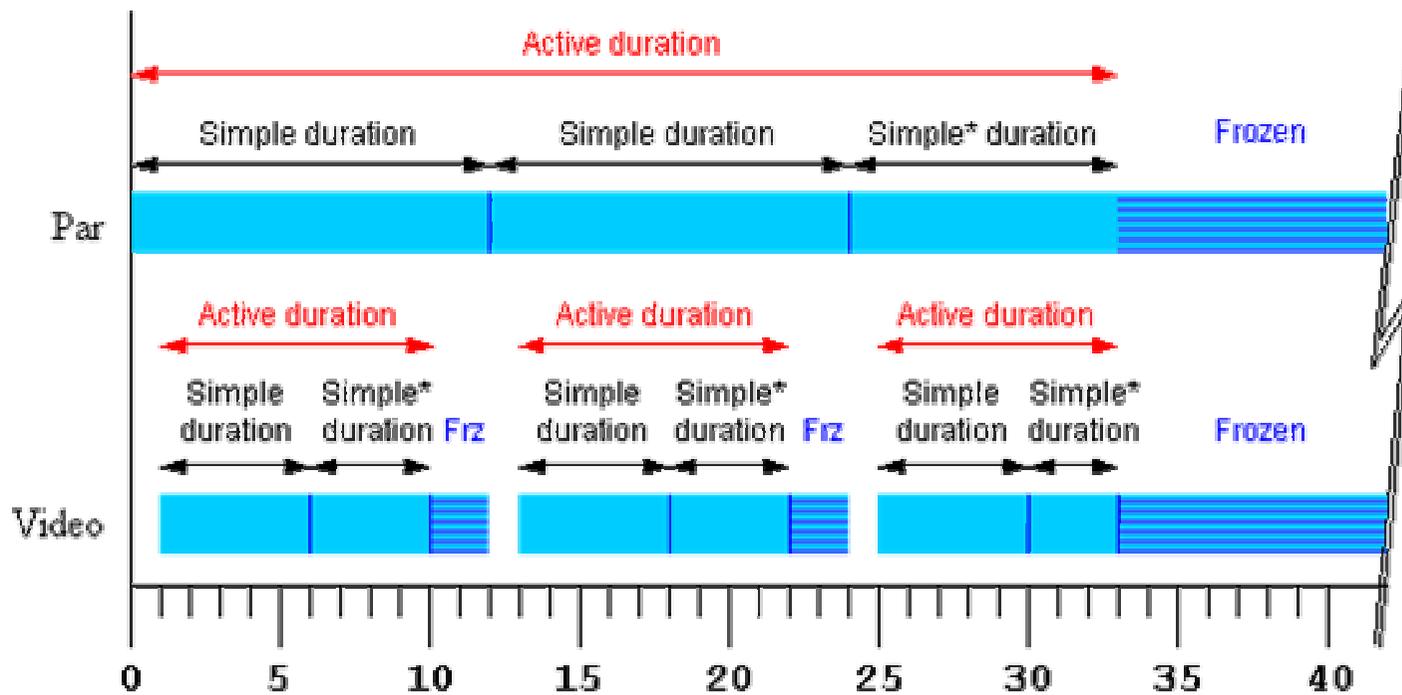
<excl>

- Semantik basiert auf der von <par>-Elementen, zu einem gegebenen Zeitpunkt kann allerdings höchstens eines der Kind-Elemente präsentiert werden
- Startet die Präsentation eines Kindelements eines <excl>-Elements, während ein anderes Kind dieses Elements gerade präsentiert wird, wird letzteres in eine Pause überführt oder gestoppt
- Die Kind-Elemente von <excl>-Elementen können innerhalb von <priorityClass>-Elementen gruppiert werden, welche das Verhalten in Bezug auf Pausieren/Stoppen steuern
- Der default-Wert des *begin-Attributs* für Kind-Elemente von <excl>-Elementen ist *“indefinite“*
- <excl>-Elemente erlauben alle für Elemente vorgesehenen Timing-Mechanismen
- Applikationshintergrund: *interactive playlists, audio descriptions, ...*



[W3C]

```
<par begin="0s" dur="33s">
  <video begin="1s" dur="10s" repeatCount="2.5" fill="freeze" ... />
</par>
```



[W3C]

```
<par begin="0s" dur="12s" repeatDur="33s" fill="freeze" >
  <video begin="1s" dur="5s" repeatCount="1.8" fill="freeze" .../>
</par>
```

Audio Description

```
<excl dur="indefinite">
  <priorityClass peers="pause">
    <video id="main_video" .../>
    <audio id="scene1_description" begin="20s" dur="30s".../>
    <audio id="scene2_description" begin="2min" dur="30s" .../>
    ...
    <audio id="sceneN_description" ... />
  </priorityClass>
</excl>
```

[W3C]

- nur eine Prioritätsklasse
- Der erste Audioclip unterbricht den Videoclip nach 20 Sekunden für 30 Sekunden. Danach wird der Videoclip fortgesetzt bis zum Einsetzen des nächsten Audioclips (u.s.w.).
- Pausierende Elemente werden in einer Queue verwaltet.

Zwei Prioritätsklassen

```
<excl dur="indefinite">
  <priorityClass id="ads" peers="defer">
    <video id="advert1" ... />
    <video id="advert2" ... />
  </priorityClass>
  <priorityClass id="program" peers="stop" higher="pause">
    <video id="program1" ... />
    <video id="program2" ... />
    <video id="program3" ... />
  </priorityClass>
</excl>
```

**Höhere
Priorität**

**Niedrigere
Priorität**

[W3C]

- Werbevideos der Prioritätsklasse *ads* können einander nicht unterbrechen oder stoppen (*peers* = “*defer*“). Sie unterbrechen Videoclips der Prioritätsklasse *program*.
- Videoclips der Prioritätsklasse *program* unterbrechen einander.