

Création de présentation Multimédia



Comment représenter du MM ?

- **Un groupe spécialisé**
 - MPEG (Moving Picture Experts Group)
 - a été chargé de la normalisation des techniques de représentation numérique du multimédia
 - Ces techniques concernent notamment :
 - Le codage des médias (audio, vidéo, graphique)
 - ❖ **aspect le plus connu de MPEG**
 - Le codage des scènes composées
 - La description du système de transport des médias...



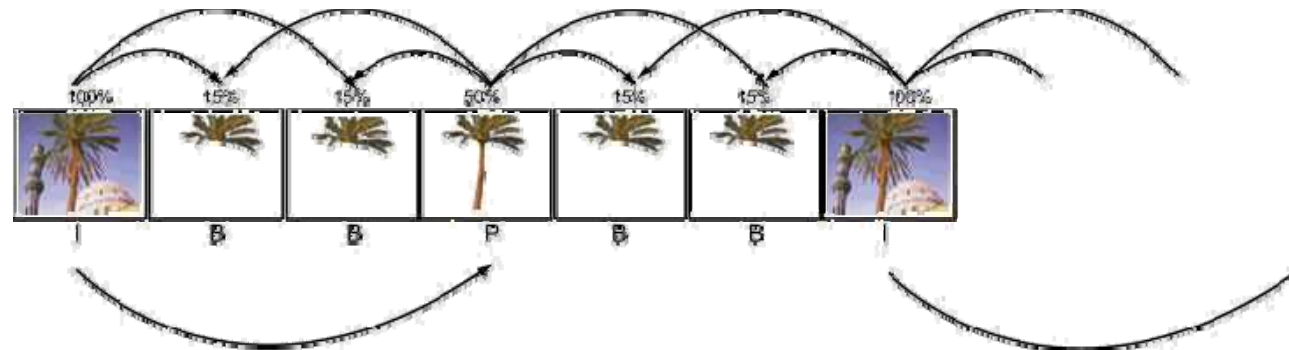
MPEG

- Les normes importantes de ce groupe
 - MPEG-1 :
 - Première norme audio et vidéo chargée du codage, de la compression...
 - ❖ Comprend le populaire format audio MPEG-1 partie 3 (MP3)
 - MPEG-2 :
 - Amélioration et adaptation de MPEG1 pour la télévision numérique
 - MPEG-4 :
 - Norme pour la description de scènes
 - MPEG-7 :
 - Norme pour la recherche du contenu multimédia

MPEG-1

■ Principe

- Façon de représenter de manière économique la vidéo et la **compresser**
- La vidéo est décomposée en images (GOP : Group of Pictures)
 - Images clés, images déduites
 - ❖ Seules les images clé sont transmises avec quelques informations sur le mouvement
 - ❖ Les images clés sont compressées par JPEG



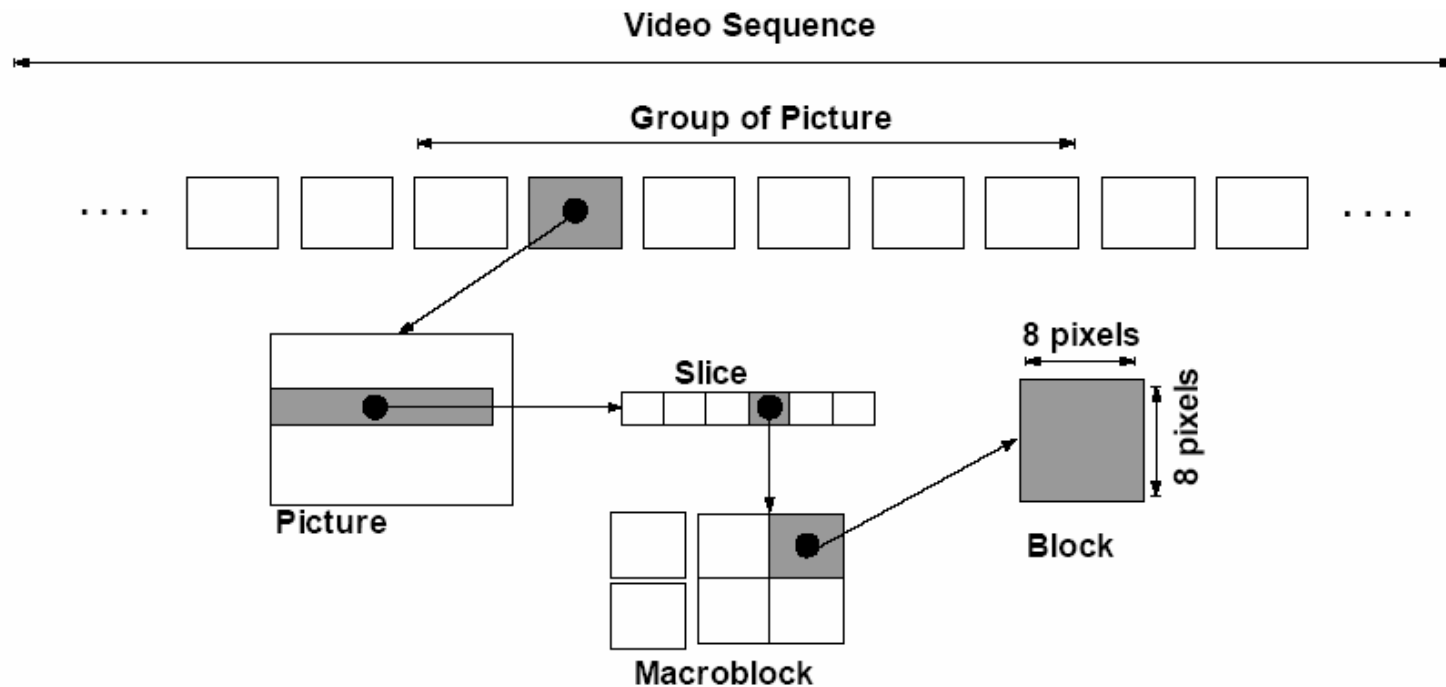


MPEG-1

- **GOP** : utilise trois types d'images : I, P et B
 - I : Intra
 - Image "clé" (*Intra picture*) comprimée indépendamment de son contexte (c'est une image JPEG)
 - P : Prédite
 - Codée à partir de l'image précédente
 - ❖ Utilise la compensation de mouvement pour un meilleur taux de compression
 - B : Bi-directionnelle
 - Utilise l'image précédente et l'image suivante comme références

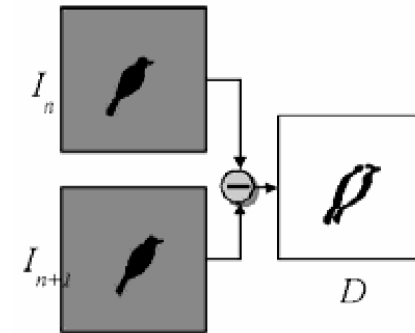
MPEG-1

- La compression JPEG sur les blocs de l'image clé



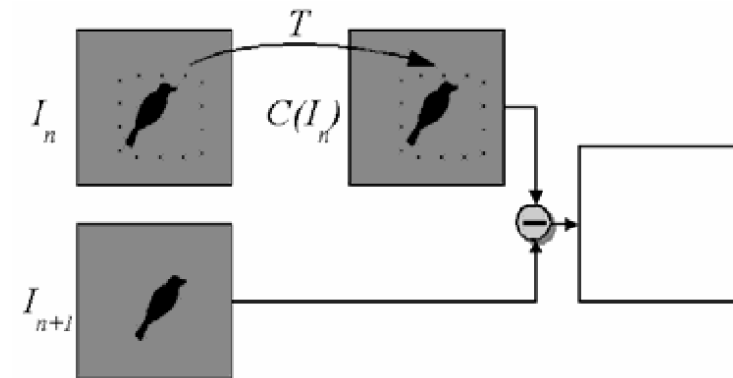
- MPEG-1 utilise un codage différentiel

- transmet uniquement I_n et $D = I_{n+1} - I_n$



- MPEG-1 compense le mouvement

- code l'image I_{n+1} comme étant la juxtaposition de translations de l'image I_n sur des blocs (macro-blocs)





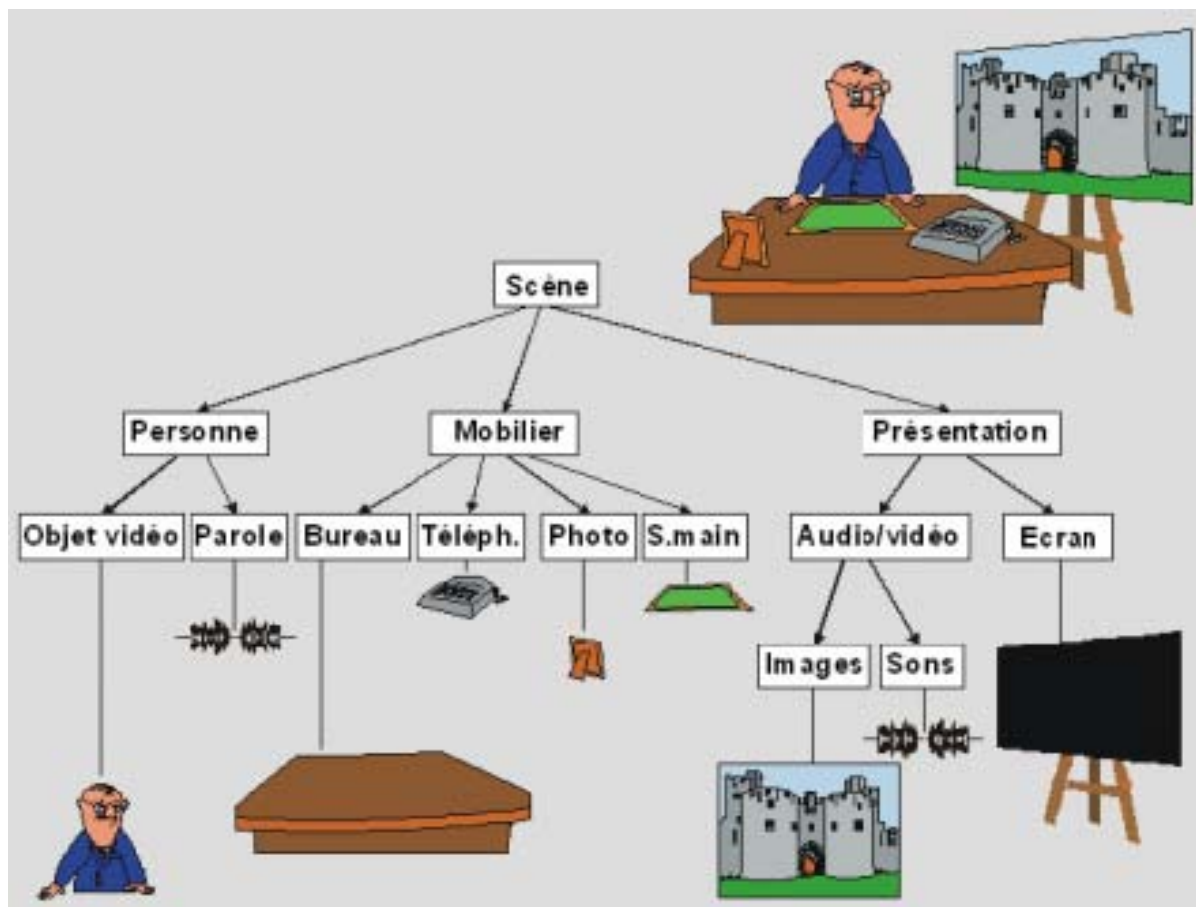
MPEG-4

- Une approche orientée "objets"
 - Pour une description autonome du contenu :
 - l'image n'est plus codée comme une surface de pixels
 - mais comme une composition d'"objets" audiovisuels
 - Les objets peuvent être de différentes natures :
 - image fixe
 - ❖ par exemple, le décor du fond, un tableau,
 - objets vidéos
 - ❖ le personnage sans le décor,
 - objets audios
 - ❖ la voix de la personne, le fond musical



MPEG-4

- Chacun de ces objets est
 - codé indépendamment des autres (autonomie), et isolé du fond ou de l'environnement
- Une hiérarchisation des contenus est également établie :
 - les objets “basiques” correspondant à une même entité
 - ❖ par exemple l'image d'un personnage et sa voix
 - peuvent être réunis et créer un nouvel objet composite comprenant l'ensemble des attributs
- Cette hiérarchisation permet de
 - faciliter le travail de conception
 - aider à la manipulation des contenus par les utilisateurs



<http://www.pfast.fr/?Le-MPEG-4&artpage=2-7>

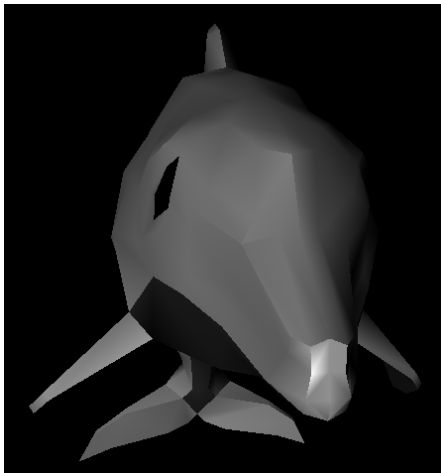


MPEG-4

- Un langage de description de scènes
 - MPEG-4 utilise BIFS (Binary Format for Scenes)
 - issu du VRML (Virtual Reality Modeling Language) pour la description et l'interactivité des objets 3D, auquel il ajoute quelques fonctionnalités
 - Exemple : Exemples-XMT-O/Dauphin.wrl
 - Il faut un plug-in pour le lire (c'est du 3D), par ex. CosmoPlayer.exe pour Opera
 - Une description textuelle de ce format binaire est
 - XMT (Extensible MPEG-4 Textual)
 - ❖ Deux langages textuels fondés sur le format XMT
 - le XMT-A (représentation basique de la scène)
 - et XMT-O (représentation avancée de la scène)

MPEG-4

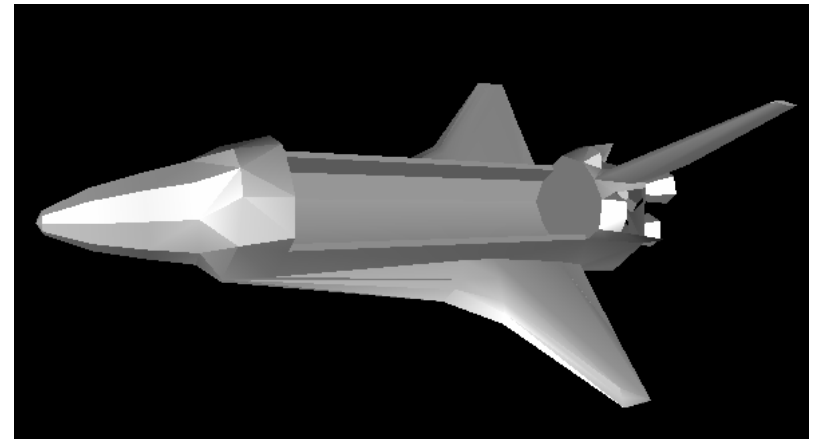
- Exemples



dauphin.wrl



Cheval.wrl



shuttle_wrl.wrl



MPEG-4

■ Dans ces formats et langages

- Le découpage des scènes en objets se fait au niveau des flux de données :
 - à chacun des objets va correspondre un flux élémentaire
- Avantages
 - Au niveau du codage
 - ❖ Choix de la technique de compression, tolérance à la perte la plus adaptées à la nature du média, et à son débit...
 - ❖ Meilleure synchronisation des flux venant de différents serveurs : avec la qualité requise pour chaque média
 - Au niveau de l'usage
 - ❖ L'autonomie des objets permet plusieurs formes d'interactivité



MPEG-4

■ Construction de fichiers XMT-O

1. Par annotation manuelle

- **VIA** : annotation par des régions → XML
- **ELAN** : annotation par des commentaires sur des lignes de temps → .eaf ~ SMIL ~ XMT-O

2. Par programmation (déclaration)

- Fichier XML avec des balises dédiées à la gestion de l'espace et du temps
 - ❖ **Attributs de synchronisation, animation**
 - ❖ **Éléments d'interactions**

MPEG-4 : annotation manuelle

■ Annotation par des régions

- Logiciel VIA : <http://mklab.iti.gr/via/>
- Interfaces-annotation/VIA
- On crée des régions (cadres) par frame ou pour une suite de frames
- Chaque région aura une position et un label
- Pour une suite de frames, on peut faire évoluer la position de la région pour suivre l'objet pendant le déroulement de la vidéo
- On obtient un fichier XML qui décrit ces régions :
Exemples-annotations/annotation-VIA.XML





MPEG-4 : annotation manuelle

■ Étapes d'annotation

1. Lancer l'application
 - via.exe
2. Créer un nouveau projet
 - File>>New Project >>Video Annotation
 - Donnez un nom au projet, et entrer un fichier vidéo
 - La vidéo s'affiche avec un nombre de frames allant de 0 à n
 - On peut faire avancer les frames (Ctrl F10) ou les faire reculer (Ctrl F9)
 - On peut sélectionner un frame (F7), en donnant son numéro



MPEG-4 : annotation manuelle

2. Annotation

- On peut annoter des régions, des shots et de la vidéo entière à partir de l'interface

3. Annotation par des régions

- Une région est un cadre que l'on positionne sur un Frame à un endroit avec une taille
- Pour créer une région, cliquer sur '+' dans la zone des régions
- Les régions peuvent être déplacées, redimensionnées avec la souris
 - Pour cela double cliquer sur les boîtes d'édition X/Y/Width/Height pour les changer, ainsi que sur les numéros des frames



MPEG-4 : annotation manuelle

3. Annotation par des régions (suite 1)

- Dans Edit
 - Le mode 'Record' permet le déplacement des cadres des régions sélectionnées, sur un frame fixe
 - Alors que le mode 'Play' permet de manipuler les régions pendant que la vidéo est en mouvement, permettant de faire évoluer la région avec ce mouvement
- Une région peut être définie sur une série de Frames
- On peut les fixer en double cliquant dessus et en changeant leur numéros dans une fenêtre de dialogue
- La modification ne s'exécute pas si on modifie directement la valeur



MPEG-4 : annotation manuelle

3. Annotation par des régions (suite 2)

- Essayer de définir une région qui suit la main :
VideoProject.XML
- Définir les frames de départ et de fin
- Se placer sur le frame de départ
- Lancer le mode Play et déplacer le cadre sur la main
- On peut importer des descripteurs
 - **File >> Import descripteurs >> descriptors.txt**

4. Enregistrement

- File >> Save Project
- Dans la fenêtre d'enregistrement, choisir le format XML

■ ELAN : annotation par des tiers : lignes

- Installation : <http://www.lat-mpi.eu/tools/elan/download>
- A = vidéo, B = annotations, C = contrôleur, D = zone de l'audio , E = zone de création des annotations
 - On crée à chaque fois une ligne en pointant la colonne de début de la vidéo
 - On lui donne un nom (acteur), puis on crée des plages dans lesquelles on inscrit les annotations. La création : Maj + clic gauche permanent, CTRL +entrée, pour enregistrer l'annotation

The screenshot displays the ELAN software interface with several key components highlighted by red circles:

- A**: Video window showing a dark frame.
- B**: Annotations table with columns for 'Nr', 'Annotation', 'Temps de d.', 'temps de fin', and 'Durée'. The table contains 7 rows of annotations.
- C**: Control panel with playback buttons and a timeline.
- D**: Audio waveform display showing amplitude over time.
- E**: Annotation creation area with a grid for actors (JEB, JER, JEG, JEP, SOP, reg JEB) and time slots for annotations.

Nr	Annotation	Temps de d.	temps de fin	Durée
1	ouais! (0.2) deux même	00:00:00.895	00:00:02.063	00:00:01.086
2	c'était un [x]	00:00:02.462	00:00:03.167	00:00:00.705
3	[ça fait] rien! j'aime bien quand c'est bien sucré!	00:00:03.536	00:00:06.428	00:00:01.892
4	mais c'était euh, rien tout ça! j'connais bien son sujet hein! ça!	00:00:06.263	00:00:08.466	00:00:03.203
5	[tous]! il avait dit qu'on pouvait faire réintervenir hein!	00:00:10.271	00:00:12.527	00:00:02.256
6	h ah ouais!	00:00:13.896	00:00:14.877	00:00:00.882
7	d'accord.	00:00:22.868	00:00:23.507	00:00:00.639



MPEG-4 : annotation manuelle

■ Principe

- Une annotation peut être une phrase, un mot ou un commentaire
- Les annotations peuvent être créées sur de multiples lignes de temps, appelées des **tiers**
- Les tiers peuvent être hiérarchiquement interconnectés
- L'annotation est conservée dans un format **XML**
- ELAN accepte en format d'entrée les *fichiers audio* .wav, les *fichiers vidéo* .mpg1, .mpg2, .mov, .avi, .wmv



MPEG-4 : annotation manuelle

■ Utilisation

- Suivre le tutoriel : [Tutorial_ELAN_v2.0.pdf](#)
- Création d'un nouveau document
 - Fichier >> nouveau >> nom du fichier >> Exemples-annotations >> coffee.mpeg
- Entrer des annotations
 - Lors de la création, un tier default est créé (1ère ligne)
 - Pour supprimer un tier, en ajouter, modifier → passer par le menu : onglet : Acteur >> modifier les attributs...
 - Les champs à renseigner :
 - ❖ Nom de l'acteur = Pseudo affiché au début de la ligne
 - ❖ Participant = Nom détaillé de l'acteur (non affiché)
 - ❖ Annotateur = champ facultatif à renseigner.
 - Ensuite, laisser par défaut les autres champs
 - Valider en cliquant sur **Ajouter**



MPEG-4 : annotation manuelle

■ Utilisation (suite 1)

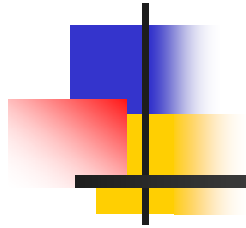
- Ajout d'une annotation
 - Se mettre sur une ligne, à l'endroit désiré (clic souris) → un trait vertical se dessine
 - Maintenir le bouton gauche appuyé et tirer vers la gauche → une zone mauve se dessine
 - Aller dans Annotation >> Nouvelle annotation ici
 - Inscrire le texte de l'annotation
 - Faire Ctrl Entrée pour valider l'annotation
- Modification d'une annotation
 - On peut agrandir, déplacer, modifier le contenu ...



MPEG-4 : annotation manuelle

■ Utilisation (suite 2)

- Enregistrement
 - Le format natif d'ELAN est un fichier **.eaf**
 - **C'est du XML**
- Réouverture
 - Fichier >> ouvrir >> nom.eaf
- Import
 - Plusieurs formats sont possibles
- Export
 - Là aussi, plusieurs formats sont possibles, HTML...
 - Très intéressant : **on peut exporter en SMIL**
 - Essayer de rejouer le fichier SMIL en utilisant Real Player
- Alignement de sources vidéo
 - Voir paragraphe 9 (dans le manuel)



Utilisation de XMT-O



Création d'une présentation MPEG-4

■ XMT-O

- Langage qui décrit le comportement dans le temps et les relations entre les différents éléments
- XMT-O est basée sur la syntaxe et la sémantique de SMIL 2.0
- SMIL utilise un scénario ou un modèle temporel
 - Tous les objets sont placés dans un espace physique composé de régions
 - Les objets évoluent dans le temps suivant une logique de synchronisation temporelle
 - Une interaction est possible avec les objets

SMIL

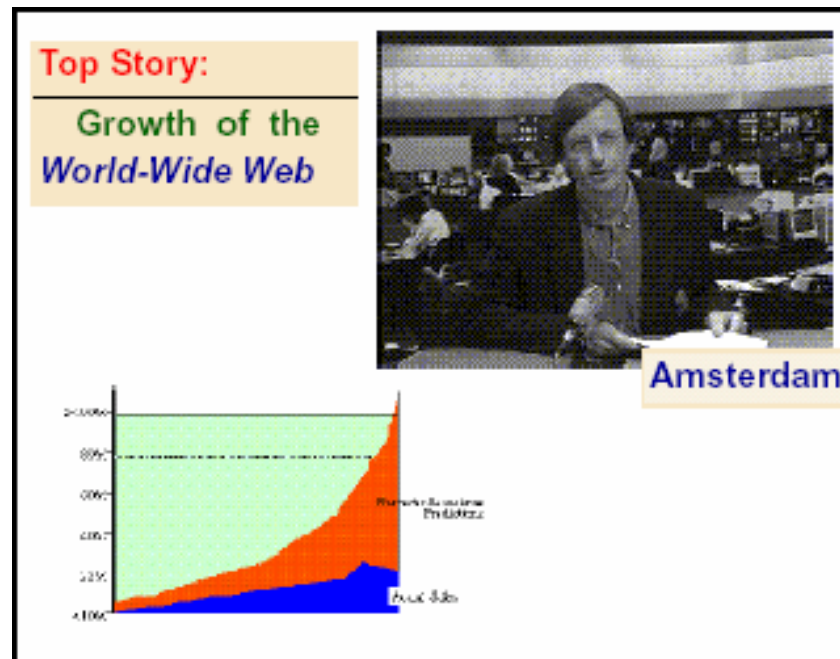
■ Exemple de scénario : simulation du JT

- Simuler ou décrire une transmission d'un journal télévisé d'une grande chaîne avec des interventions de journalistes différents, localisés à des endroits différents, avec des commentaires textuels, de l'audio et de l'image



Exemple

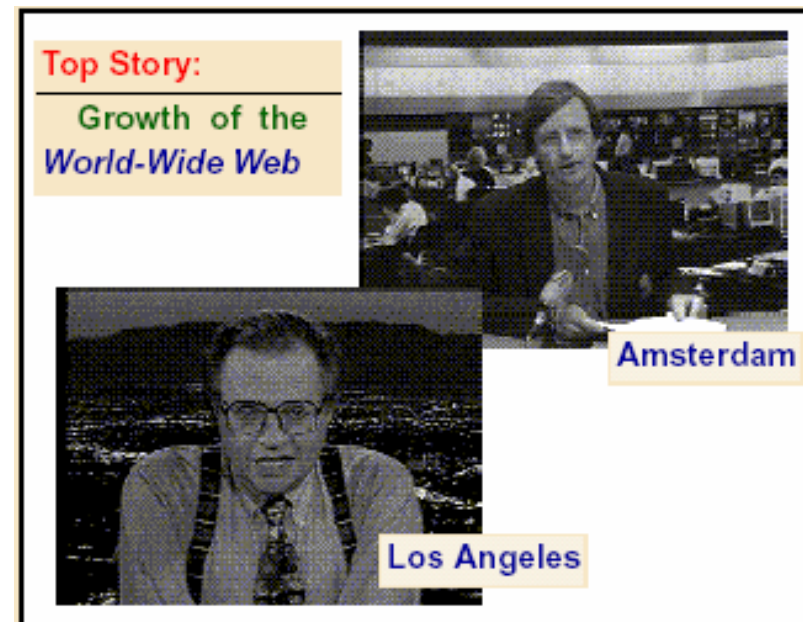
Scénario : ... On commence par mettre une image, avec de l'audio, de la vidéo, puis d'un seul coup on va avoir besoin d'illustrer ce qui est dit, donc on va mettre très rapidement un graphique qui va illustrer par ex. l'aspect vitesse avec le Web



le graphique apparaît durant le commentaire parlé

Exemple

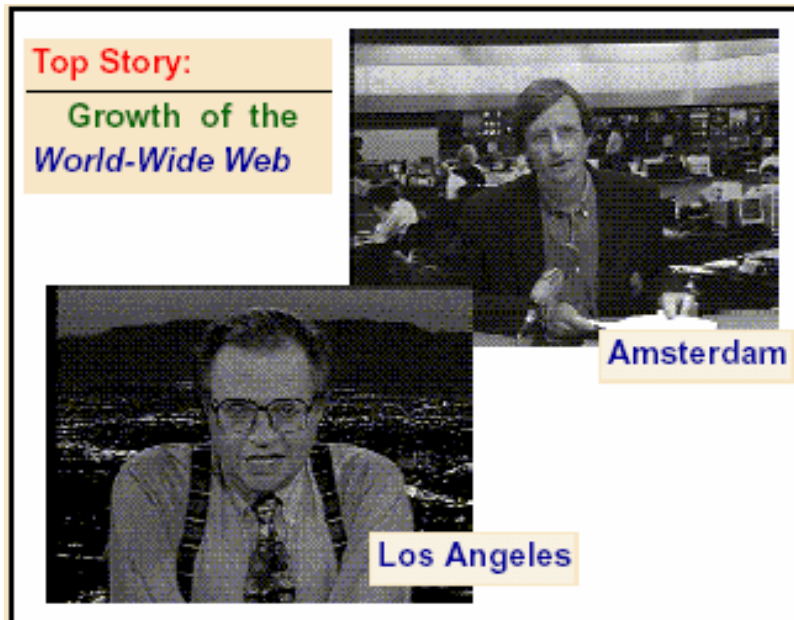
- Imaginons maintenant qu'un journaliste va appeler un autre journaliste à l'autre bout du monde
- Il va lui demander un commentaire sur ce qui se passe
- On va alors faire apparaître une autre vidéo qui va dire autre chose... qui se passe ailleurs
- C'est ce qui se passe réellement à la télé, il faut qu'on puisse le faire sur le Web, avec une fenêtre qui surgit



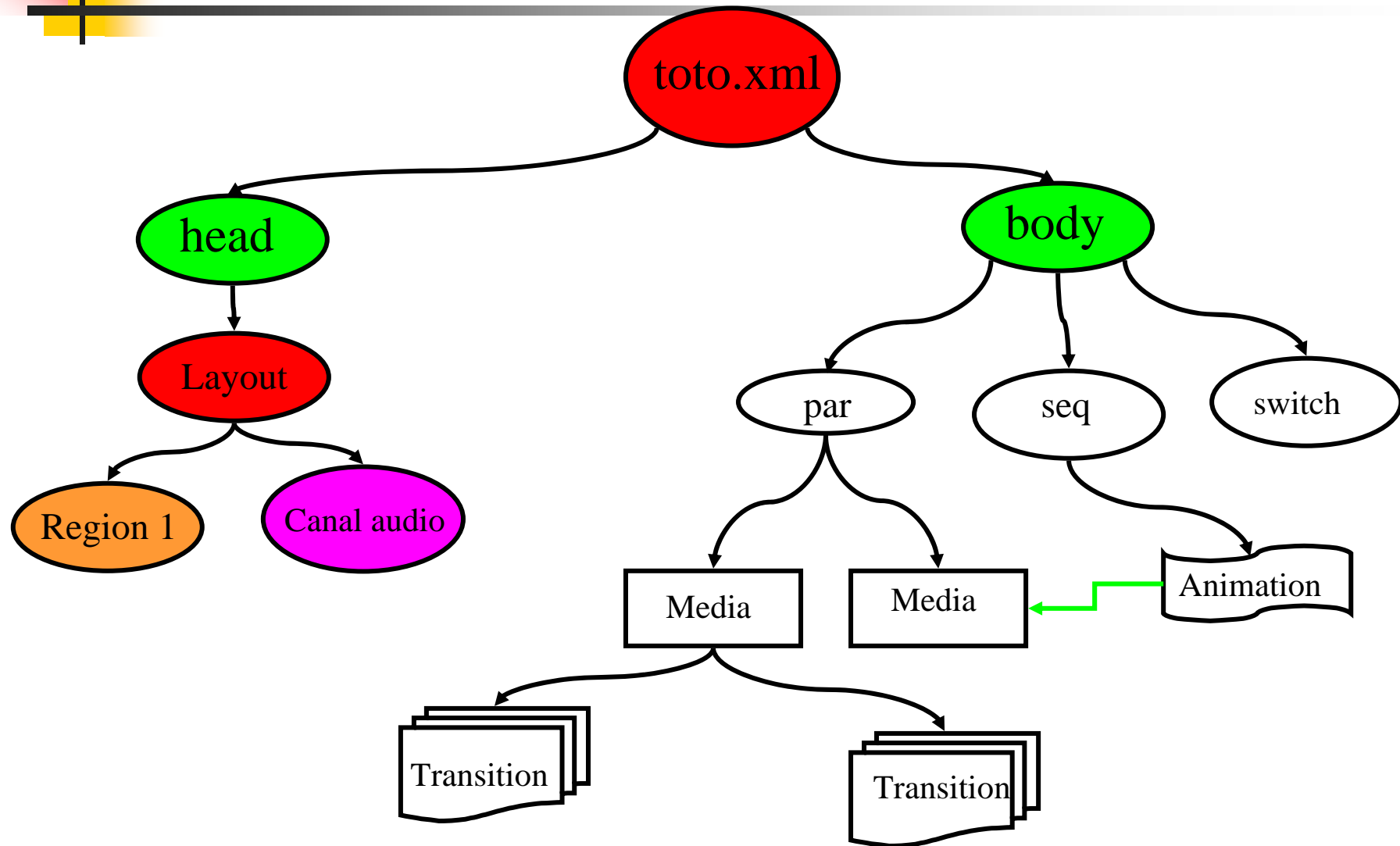
La première vidéo se termine, la deuxième démarre

Exemple

- Si le correspondant est distant :
 - Il faut alors mettre en place un lien vers le correspondant
 - A tout moment pendant la vidéo, le journaliste doit pouvoir demander au correspondant de l'information supplémentaire



Structure du fichier XMT





Médias traités

■ Graphique

- `<rectangle size="100 20">`
- `<circle radius="90">`
- `<curve points="0 -.18;0 .18;.12 .18">`
- `<lines coord="0 -.18;0 .18;.12 .18;.12 .06 ">`
- `<points coord="-250 50; -120 80; -50 50; 30 10; 50 -80/>`
- `<polygons coord="175 112; 175 150; -175 150; -175 112"/>`

■ Texte

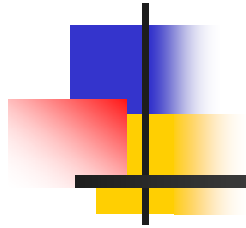
- `<string textLines="Just click to goto back">`

■ Visuel

- `<video src=".." />`
- ``

■ Audio

- `<audio src=".." />`

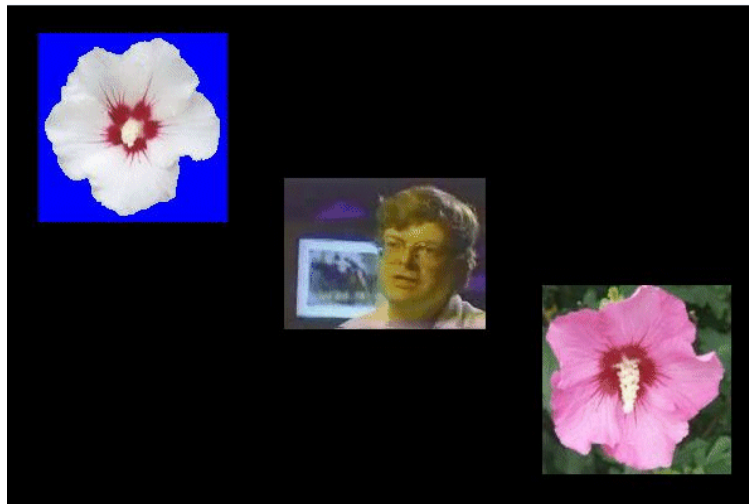


Positionnement des médias dans l'espace

Les éléments <layout> et <region>

Layout

Exemple : IBMMp4Media.xml



```
<layout type="xmt/xmt-basic-layout" metrics="pixel" >
  <topLayout height="400" width="600" >
    <region id="region00" translation="-200 100"/>
    <region id="region01" translation="0 0"/>
    <region id="region02" translation="200 -100"/>
  </topLayout >
</layout >
```



Layout

Exemple : IBMSamplesForXmt/IBMMp4Media.xml

```
<body>
  <par>
    
    <!-- first image track -->
    <video dur="indefinite" region="region01"
      src="../multimedia/IBMMp4Media.mp4#video" />
    <!-- first video track -->
    
    <!-- third media track -->
    <audio dur="indefinite"
      src="../multimedia/IBMMp4Media.mp4#audio(1)"/>
    <!-- first audio track -->
  </par>
</body>
```



Layout

■ Le paramètre "z-index"

- détermine le niveau de l'empilement de la région dans son contexte d'empilement (l'indice le plus élevé sur le dessus)

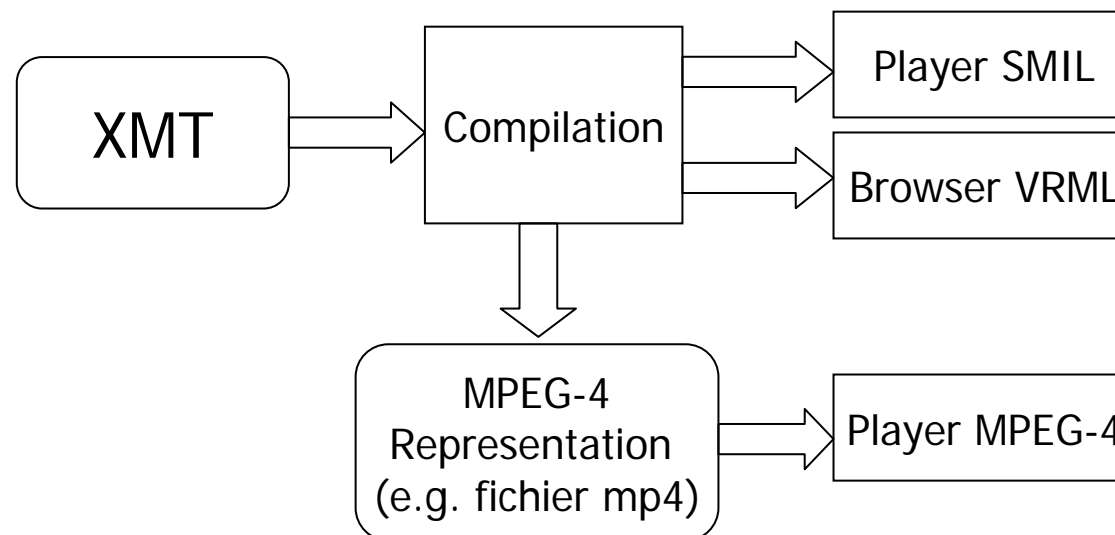
Si deux éléments A et B ont la même profondeur **alors**:

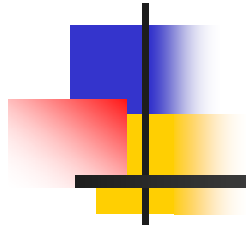
- Si B débute après A **alors** B est placé au-dessus de A
- Si B et A débute en même temps et si l'élément B est après l'élément A dans l'arbre XML **alors** B est placé au-dessus de A

XMT-O

■ Démarche pour la création d'une animation MPEG-4 (mp4)

- On écrit des fichiers XMT-O
- On utilise le toolkit d'IBM : IBMToolkitForMpeg4/XmtBatch.bat
- On donne le fichier XMT-O et le fichier de sortie suffixé par mp4
➔ on obtient un fichier mp4
- Pour lire le mp4, utiliser Osmo4 de GPAC (IBM)





Timing et synchronisation

Les éléments `<par>`, `<seq>`, `<dur>`,
`<begin>`...

Timing et synchronisation

- L'élément *par* groupe les éléments qui sont joués en parallèle
 - Les enfants d'un élément *par* démarrent en même temps

<par>

```

```

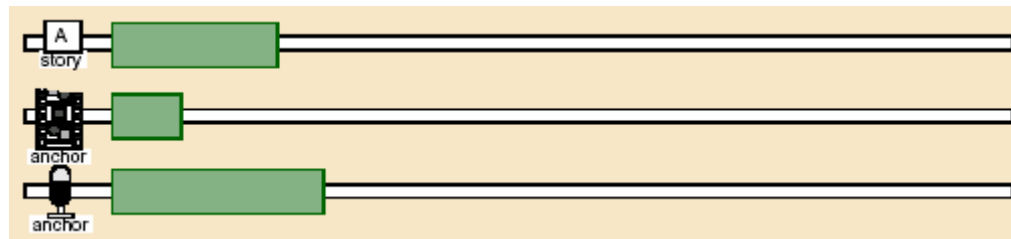
```
<video dur="indefinite" region="region01"
    src="../multimedia/IBMMp4Media.mp4#video" />
```

```

```

```
<audio dur="indefinite"
    src="../multimedia/IBMMp4Media.mp4#audio(1)"/>
```

</par>





Timing et synchronisation

- L'élément *par*
 - Les éléments <par> peuvent s'imbriquer
 - Ex : IBMTimeContainersPar/IBMTimeContainersPar.xml

Timing et synchronisation

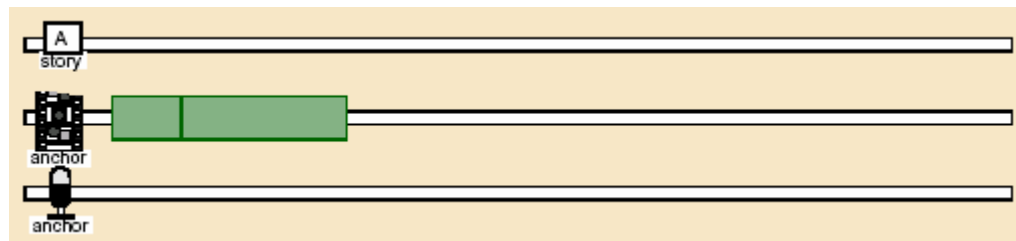
- L'élément `seq` groupe les éléments qui sont joués en séquence
 - Les enfants sont joués les uns après les autres dans l'ordre où ils apparaissent dans le fichier

```
<seq>
```

```
  <video src="logo.mpv" region="V-main" />
```

```
  <video src="anchor.mpv" region="V-main" />
```

```
</seq>
```

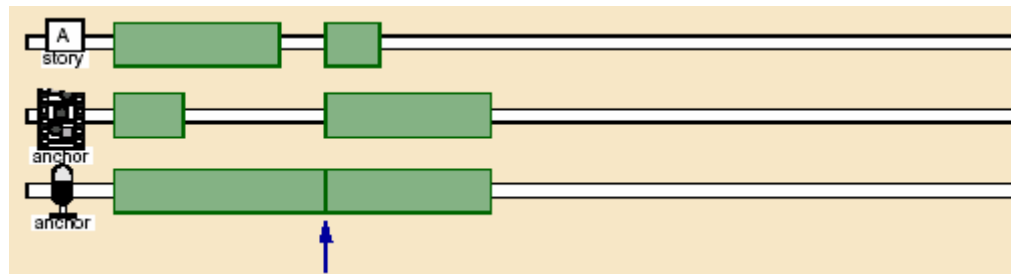


- Le temps de début d'un fils de l'élément `seq` est le temps de début de l'élément `seq` lui-même
- Le temps de début du prochain fils est le temps de fin du fils précédent

Timing et synchronisation

- Les structures `par` et `seq` peuvent s'imbriquer les unes dans les autres

```
<seq>
  <par>
    <text src="leader_title.html" region="m_title" dur="5s" />
    <video src="cnn.mpv" region="V-main" />
    <audio src="cnn.aiff" region="music" />
  </par>
  <par>
    <text src="story_title.html" region="m_title" dur="2s" />
    <video src="anchor.mpv" region="V-main" />
    <audio src="anchor.aiff" region="music" />
  </par>
</seq>
```



Timing et synchronisation

- Gestion du temps : durée, retard, synchronisation
 - L'objet multimédia a une durée intrinsèque
 - Ex : l'audio (ou la vidéo) dure 5,3 secondes
 - On peut fixer cette durée : attribut <dur>
 - la raccourcir ou la rallonger
 - la valeur *indefinite* (pour une durée indéfinie)

```
<video src="zoomin.mpv" region="V-main" dur="4s" />
```



l'objet multimédia s'arrête après 4 secondes

```
<video src="zoomin.mpv" region="V-main" dur="6.5s" />
```



l'objet multimédia s'arrête au bout de 6,5 secondes

Timing et synchronisation

- La durée d'un objet peut être étendue par répétition du contenu

– Attributs *repeatcount*, *repeatDur*

```
<video src="zoomin.mpv" region="V-main" repeatcount="3" />
```



L'objet s'arrête après 15,9 secondes

```
<video src="zoomin.mpv" region="V-main" repeat="3"
dur="11s" />
```



L'objet s'arrête après 11 secondes

```
<video src="zoomin.mpv" region="V-main"
repeatcount="indefinite" />
```



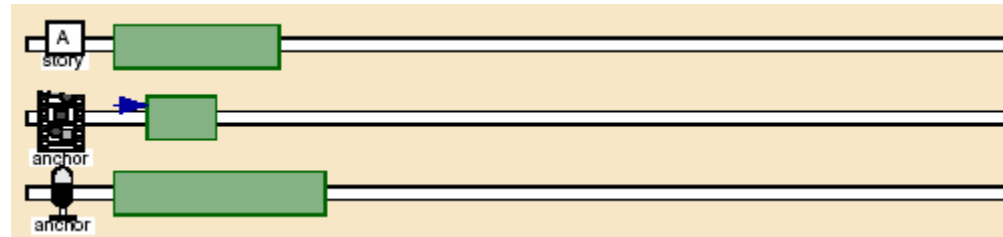
L'objet s'arrête quand son parent s'arrête

- La valeur de l'attribut *repeatcount* est un entier ou indéfinie

Timing et synchronisation

- **Attribut begin** : valeur de retard : modifie la valeur de départ

```
<par>  
  <text src="leader_title.html" region="m_title" dur="5s" />  
  <video src="cnn.mpv" region="V-main" begin="1.4s" />  
  <audio src="cnn.aiff" region="music" />  
</par>
```



- La vidéo est retardée de 1,4 seconde par rapport au début de l'élément **par**
- La syntaxe de la valeur du retard est une valeur d'horloge
- Remarque
 - On peut afficher des pages HTML dans le player

Timing et synchronisation

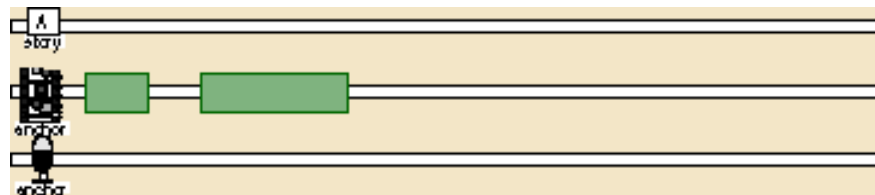
- **Attribut begin** : valeur de retard dans une seq

```
<seq>
```

```
<video src="logo.mpv" region="V-main" />
```

```
<video src="anchor.mpv" region="V-main" begin="1.4s" />
```

```
</seq>
```

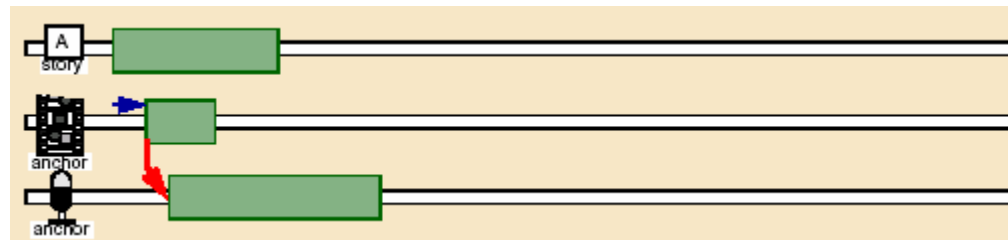


- La vidéo est retardée de 1,4 seconde par rapport à la fin de l'élément précédent

Timing et synchronisation

- **Attribut begin** : valeur d'événement

```
<par>  
  <text src="leader_title.html" region="m_title"  
    dur="5s" />  
  <video id="v1" src="cnn.mpv" region="V-main"  
    begin="1.4s" />  
  <audio src="cnn.aiff" region="music"  
    begin="id(v1) (0.5s)" />  
</par>
```

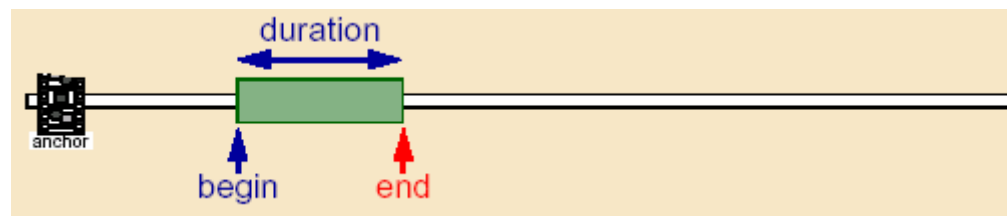


- L'audio est retardée de 0,5 seconde par rapport au début de l'élément vidéo "v1"
- La syntaxe d'une valeur d'événement est : "id(id-value)(begin|end|clock-value)"

Timing et synchronisation

- Un élément multimédia possédant une durée implicite ou explicite et un temps de début, a un temps de fin = temps de début + durée

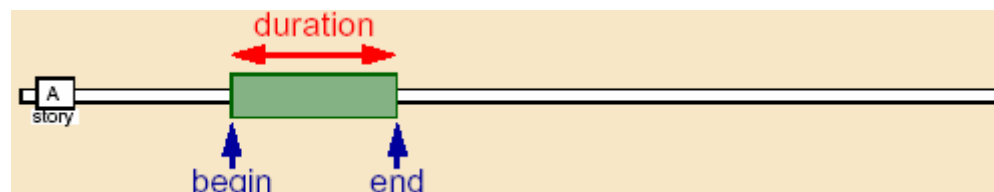
```
<video src="cnn.mpv" region="V-main" begin="4s" />
```



Attribut **end** : même syntaxe que l'attribut **begin**

- Un élément multimédia possédant un temps de début explicite et un temps de fin explicite a une durée = temps de fin - temps de début

```
<text src="title.html" region="m_title" begin="4s" end="8s" />
```



Timing et synchronisation

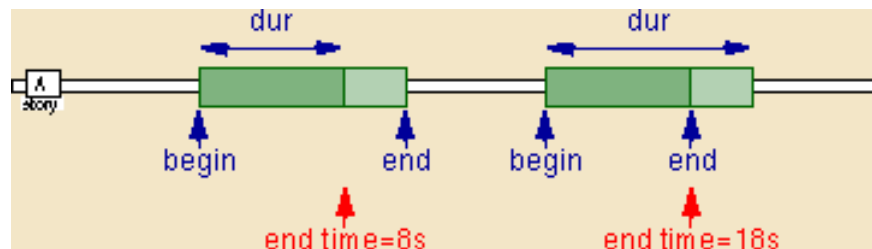
- Un élément multimédia possédant un temps de début explicite, une durée et un temps de fin explicite, va avoir un temps de fin égal au minimum entre le temps de fin explicite et la somme temps de début + durée

```
<seq>
```

```
  <text scr="title.html" region="m_title" begin="4s"  
    dur="4s" end="10s" />
```

```
  <text scr="title.html" region="m_title" begin="14s"  
    dur="6s"end="18s" />
```

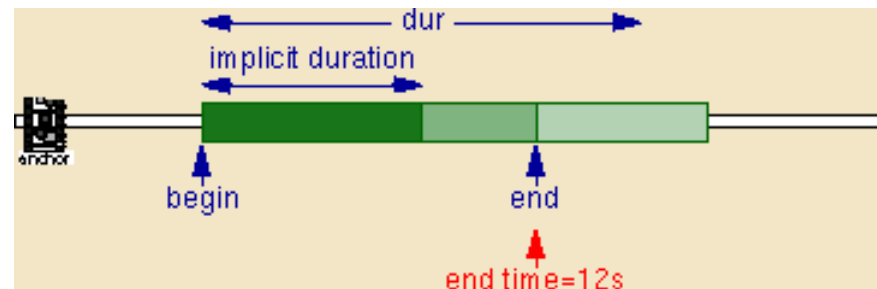
```
</seq>
```



Timing et synchronisation

- Un élément multimédia peut aussi avoir une durée intrinsèque

```
<video src="cnn.mpv" region="V-main" begin="3s" dur="14s"  
end="12s" />
```



- Si le temps de fin est antérieur au temps de début alors l'élément n'est pas joué
- Le temps de fin d'un élément multimédia possédant une durée indéfinie est déterminé par le temps de fin de son parent
- Si le parent n'a pas de temps de fin défini (ou n'existe pas) alors l'élément multimédia sera joué indéfiniment

Timing et synchronisation

- Un élément `seq` se termine quand son dernier fils se termine
- A noter que si son dernier fils a une fin indéterminée alors l'élément `seq` a une fin indéterminée

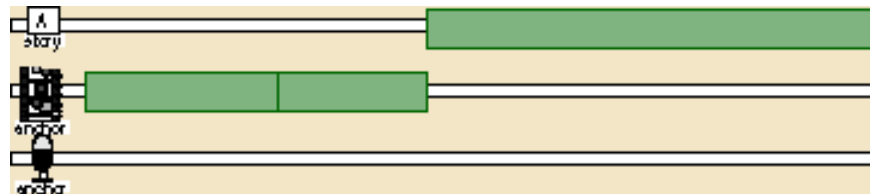
```
<seq>
```

```
  <video src="logo.mpv" region="V-main" />
```

```
  <video src="anchor.mpv" region="V-main" />
```

```
  <text src="title.html" region="m_title"  
    dur="indefinite" />
```

```
</seq>
```

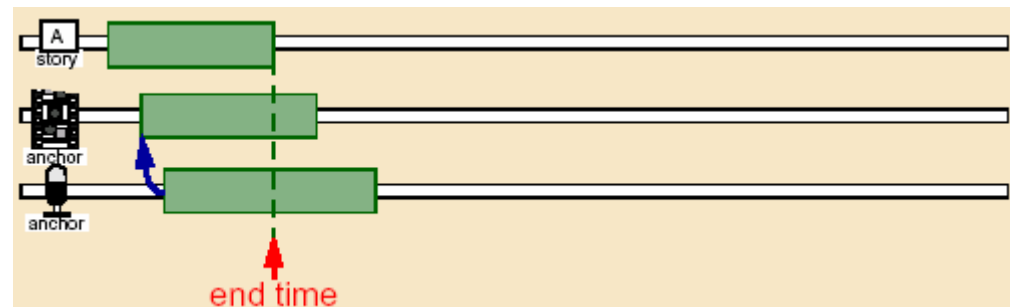


Timing et synchronisation

■ Attribut *endsync*

- *par* peut se terminer quand le premier élément multimédia se termine

```
<par endsync="first">  
  <text src="leader_title.html" region="m_title"  
    dur="5s" />  
  <video id="v1" src="cnn.mpv" region="V-main"  
    begin="1.4s" />  
  <audio src="cnn.aiff" region="music"  
    begin="id(v1) (0.5s)" />  
</par>
```

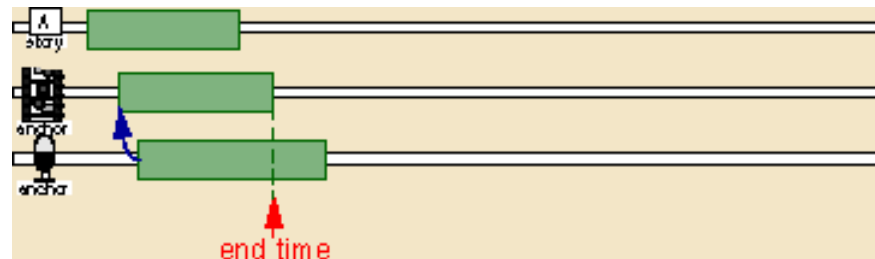


Timing et synchronisation

■ Attribut *endsync*

- *par* peut se terminer quand l'élément multimédia référencé se termine : `id(id-value)`

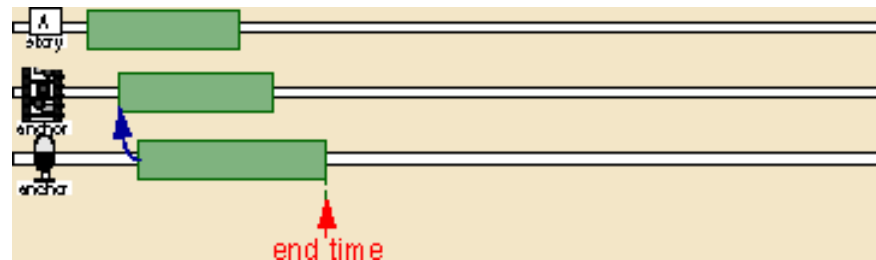
```
<par endsync="id(v1)">  
  <text src="leader_title.html" region="m_title"  
    dur="5s" />  
  <video id="v1" src="cnn.mpv" region="V-main"  
    begin="1.4s" />  
  <audio src="cnn.aiff" region="music"  
    begin="id(v1) (0.5s)" />  
</par>
```



Timing et synchronisation

- **Attribut *endsync* :**
 - *par* peut se terminer quand le dernier élément multimédia se termine

```
<par endsync="last">  
  <text src="leader_title.html" region="m_title"  
    dur="5s" />  
  <video id="v1" src="cnn.mpv" region="V-main"  
    begin="1.4s" />  
  <audio src="cnn.aiff" region="music"  
    begin="id(v1) (0.5s)" />  
</par>
```

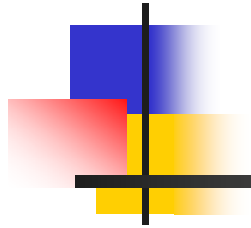




Timing et synchronisation

■ Exemples

- IBMTimeContainersParBeginSync.xml
- IBMTimeContainersParEndSync.xml
- IBMTimeContainersParSeqEndSync.xml



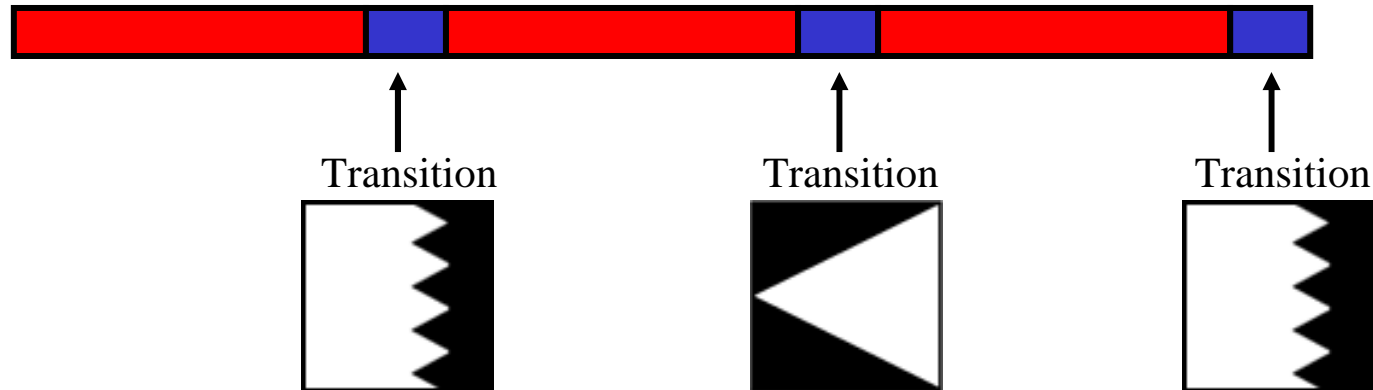
Transitions

`<transitionFilter>`

Transitions

■ L'élément `<transitionFilter>`

- On peut adoucir les effets de transition du clip courant au clip suivant
- Au lieu que le clip apparaisse ou disparaît brutalement, un effet de fondu apparaît :





Transitions

- Exemples : IBMTransitionDemo.xml, IBMTransitionsGroup.xml

```

  <transitionFilter type="slideWipe" subtype="fromTop" dur="2s"
    mode="in"/>
</img>
```







- Il y a plus de 100 effets de transitions

- <transitionFilter begin="30s" dur="2s" mode="in" type="boxWipe" />
- <transitionFilter begin="35s" dur="2s" type="fourBoxWipe" subtype="cornersOut" />
- <transitionFilter begin="40s" dur="2s" type="waterfallWipe" subtype="verticalRight"/>
- <transitionFilter begin="45s" dur="2s" type="spiralWipe" />
- <transitionFilter begin="50s" dur="2s" mode="in" type="zoom"



Transitions







■ Transitions de type contour

Type	Subtype	SMPTE	Transition Appearance
barWipe	leftToRight	1	 A bar moves from left to right.
	topToBottom	2	 A bar moves from top to bottom.
boxWipe	topLeft	3	 A box expands from the upper-left corner to the lower-right corner.
	topRight	4	 A box expands from the upper-right corner to the lower-left corner.
	bottomRight	5	 A box expands from the lower-right corner to the upper-left corner.
	bottomLeft	6	 A box expands from the lower-left corner to the upper-right corner.



Transitions






■ Transitions de type Iris

Type	Subtype	SMPTE	Transition Appearance
irisWipe	rectangle	101	 A rectangle expands from the center.
	diamond	102	 A four-sided diamond expands from the center.
triangleWipe	up	103	 A triangle pointed toward the top edge expands from the center.
	right	104	 A triangle pointed toward the right edge expands from the center.
	down	105	 A triangle pointed toward the bottom edge expands from the center.
	left	106	 A triangle pointed toward the left edge expands from the center.



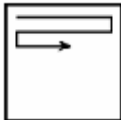
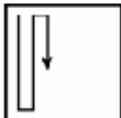




Transitions

■ Transitions de type Clock

Type	Subtype	SMPTE	Transition Appearance
clockWipe	clockwiseTwelve	201	 <p>A radial hand sweeps clockwise from the twelve o'clock position.</p>
	clockwiseThree	202	 <p>A radial hand sweeps clockwise from the three o'clock position.</p>
	clockwiseSix	203	 <p>A radial hand sweeps clockwise from the six o'clock position.</p>
	clockwiseNine	204	 <p>A radial hand sweeps clockwise from the nine o'clock position.</p>
pinWheelWipe	twoBladeVertical	205	 <p>Two radial hands sweep clockwise from the twelve and six o'clock positions.</p>

Transitions

■ Transitions de type Matrix

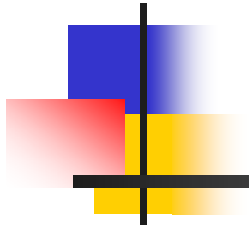
Type	Subtype	SMPTE	Transition Appearance
snakeWipe	topLeftHorizontal	301	 <p>Tiles move in a horizontal zigzag from the upper-left corner.</p>
	topLeftVertical	302	 <p>Tiles move in a vertical zigzag from the upper-left corner.</p>
	topLeftDiagonal	303	 <p>Tiles move in a diagonal zigzag from the upper-left corner.</p>
	topRightDiagonal	304	 <p>Tiles move in a diagonal zigzag from the upper-right corner.</p>
	bottomRightDiagonal	305	 <p>Tiles move in a diagonal zigzag from the lower-right corner.</p>
	bottomLeftDiagonal	306	 <p>Tiles move in a diagonal zigzag from the lower-left corner.</p>



Transitions

■ Transitions de type Push, Fade...

Type	Subtype	Transition Appearance
fade	crossfade	The clip fades into the clip that follows it.
	fadeFromColor	The clip fades in from a solid color.
	fadeToColor	The clip fades out into a solid color.
pushWipe	fromLeft	The clip pushes out the preceding clip from left to right.
	fromRight	The clip pushes out the preceding clip from right to left.
	fromTop	The clip pushes out the previous clip from top to bottom.
	fromBottom	The clip pushes out the previous clip from bottom to top.
slideWipe	fromLeft	The clip slides over the preceding clip from left to right.
	fromRight	The clip slides over the preceding clip from right to left.
	fromTop	The clip slides over the previous clip from top to bottom.
	fromBottom	The clip slides over the previous clip from bottom to top.



Animation

`<animate>`



Animations

■ L'élément `<animate>`

- Les utilisations communes de l'animation incluent :
 - agrandissement ou rétrécissement ,
 - déplacement sur l'écran,
 - changement de la couleur du fond d'une région,
 - amplification ou diminution du niveau sonore
 - changement de la transparence d'une agrafe pour le faire davantage, ou moins, opaque
- L'élément `<animate>` ou `<animateColor>` ou `<animateMotion>` est utilisé pour animer un attribut ou une propriété dans le temps



Animations

■ Les attributs de <animate>

Attribute	Value	Function
accumulate	none sum	Makes a repeating animation build with each iteration.
additive	replace sum	Adds the animation value to the existing value.
attributeName	<i>attribute_name</i>	Selects the attribute to animate.
by	<i>pixels percentage color_value</i>	Animates the element by a certain amount. Do not use with to.
calcMode	discrete linear paced	Controls the flow of an animation.
from	<i>pixels percentage color_value</i>	Sets a starting point for the animation. Use with to or by.
targetElement	<i>ID</i>	Identifies the tag that contains the animated attribute.
to	<i>pixels percentage color_value</i>	Sets an end point for the animation. Do not use with by.
values	<i>pixels percentage color_value</i>	Defines a list of animation values. Not used with from, to, or by.



Animations

■ Examples

- IBMAnimation.xml
- IBMAnimationColor.xml
- IBMAnimationCoordinates.xml
- IBMAnimationDemo.xml
- IBMAnimationMotion.xml
- IBMAnimationRotation.xml



Groupement d'éléments

■ Possibilités

- XMT-O possède plusieurs constructions pour regrouper des objets dans des blocs
- Liste des éléments les plus importants :
 - Groupement d'éléments : `<g>`
 - Groupement de définitions : `<defs>`
 - Utilisation d'éléments : `<use>`
- Exemple
 - `IBMDefs.xml`



Groupement d'éléments

■ Groupement d'éléments : <g>

- L'élément <g> sert à regrouper des éléments qui vont ensemble, qui se partagent des attributs : couleur, style, ...
- Les fils de <g> héritent les propriétés

■ Exemple

```
<group id="defGroup3">  
  <rectangle id="defRect3" size="100 100">  
    <material id="defMat3" color="lime" filled="true">  
      <animateColor id="defAnim3" dur="2s" repeatDur="indefinite"  
        calcMode="linear" attributeName="color" to="aqua"/>  
    </material>  
  </rectangle>  
</group>
```



Groupement de définitions

- **Groupement de définitions : <defs>**
 - autorise la définition d'objets référencés plus tard dans le même fichier
- **Exemple :**

```
<defs>
```

 - On définit un élément transition

```
<a:transition id="t1" type="barWipe" subType="horizontal"
direction="forward" dur="2">
  <a:param id="param" name="cellSize" value="5"/>
</a:transition>
```
 - On définit également deux images :

```
<image id="im1" width="400" height="400"
xlink:href="fondu1.jpg"/>
<image id="im2" width="400" height="400"
xlink:href="fondu2.jpg"/>
```

```
</defs>
```



Utilisation des éléments

- **<use> permet de réutiliser les objets**

- On peut préciser les paramètres d'usage comme le temps de début, de fin, de durée...
- Exemple

```
<group translation="-50 0150">  
  <use id="use5" begin="00s" dur="12s" xlink:href="#defRect2"/>  
</group>  
<group translation="050 0150">  
  <use id="use6" begin="03s" dur="12s" xlink:href="#defRect2"/>  
</group>  
<group translation="-50 0050">  
  <use id="use7" begin="06s" dur="12s" xlink:href="#defRect3"/>  
</group>
```

Les liens

Lien d'un élément à une présentation

- Indiqué par : `<a>`

- Plusieurs formes

- Éléments cliquables : IBMAnchor.xml

```
<a href="IBMAnchor1.mp4">  
  <rectangle dur="indefinite" size="180 120">  
    <material color="lime" filled="true">  
      <set begin="8.0" dur="indefinite"  
        attributeName="color" to="blue"/>  
    </material>  
  </rectangle>  
  <string dur="indefinite...>  
</a>
```





Les liens

Lien d'un élément à une présentation

- Éléments cliquables : IBMAnchorActivate.xml
- Ici , le lien se fait directement au chargement. Pas besoin de cliquer

```
<a href="IBMAnchorActivate1.mp4" actuate="onLoad">  
  <rectangle begin="5s" dur="indefinite" size="1 1">  
    <transformation visibility="false"/>  
  </rectangle>  
</a>
```



Éléments d'interaction

■ Interaction

- Ici, en cliquant sur le rectangle d'Id=greenRect1, on fait apparaître un cercle rouge pendant 5s

```
<circle begin="greenRect1.click" dur="5s" radius="50">
```

```
<transformation translation="150 -75"/>
```

```
<material color="red" filled="true"/>
```

```
</circle>
```

- D'autres exemples
 - [IBMEventMedia.xml](#)
 - [IBMEventxMediaChildren.xml](#)