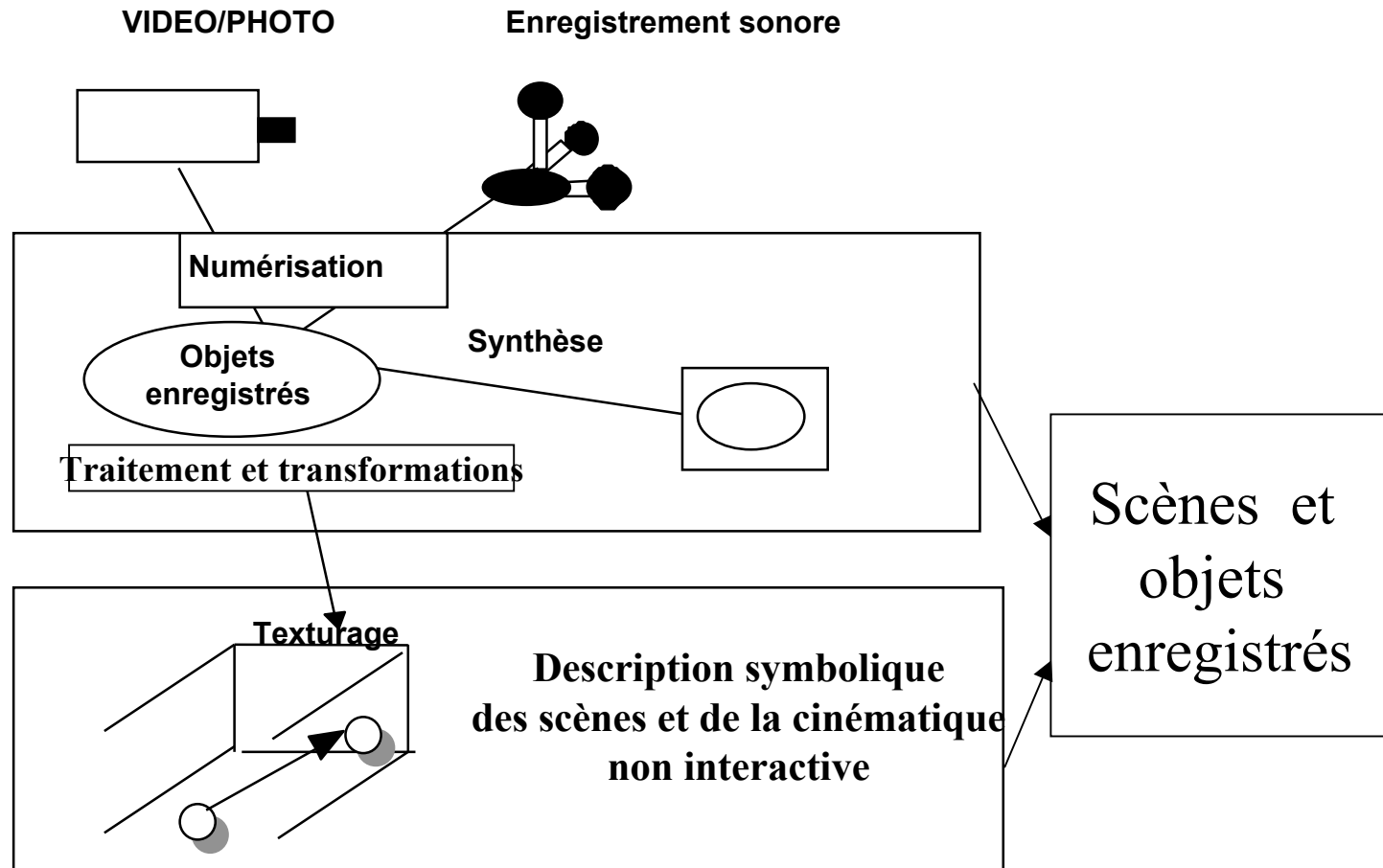


Représentation symbolique des objets multimédia Notion de scène

S. Natkin

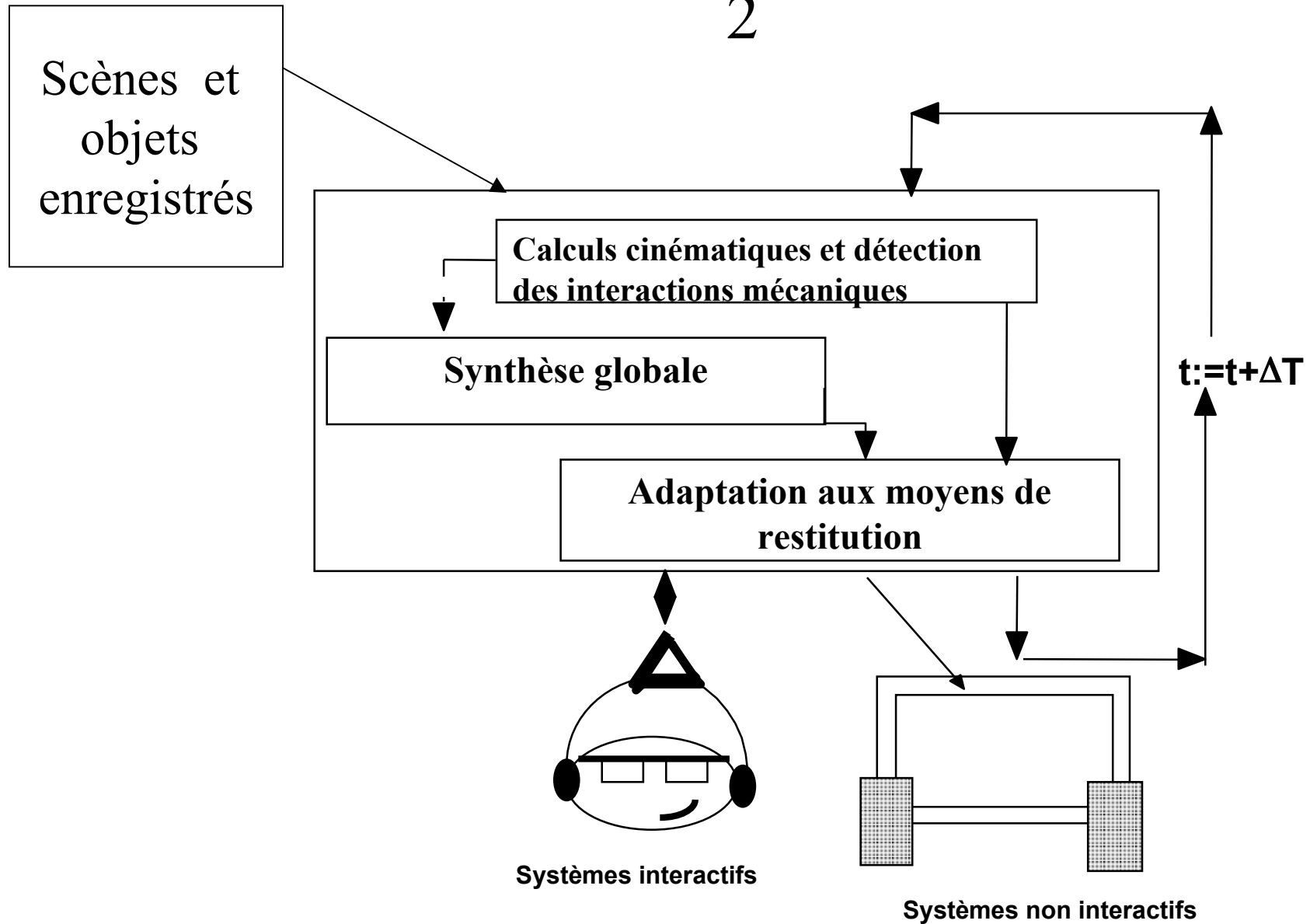
26/11/2001

Processus de création d'un objet multimédia 1



Processus de création d'un objet multimédia

2



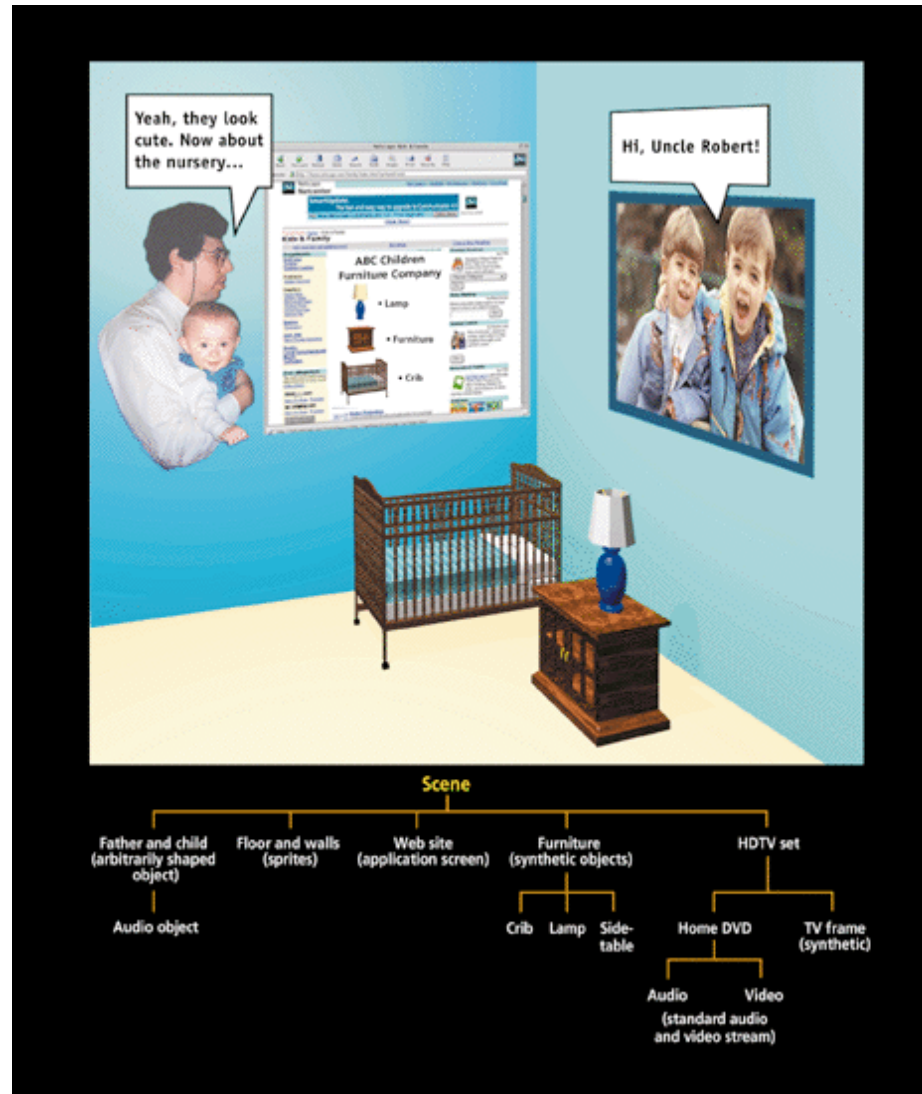
La scène

C'est une description symbolique d'un univers virtuel

Interprétable par des outils de synthèse et de visualisation

Qui se présente comme un document structuré

La scène MPEG 4

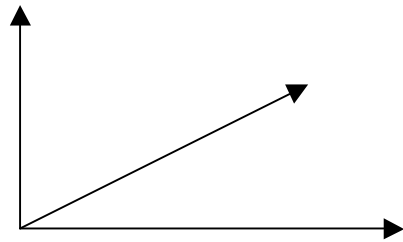


Que faut il représenter ?

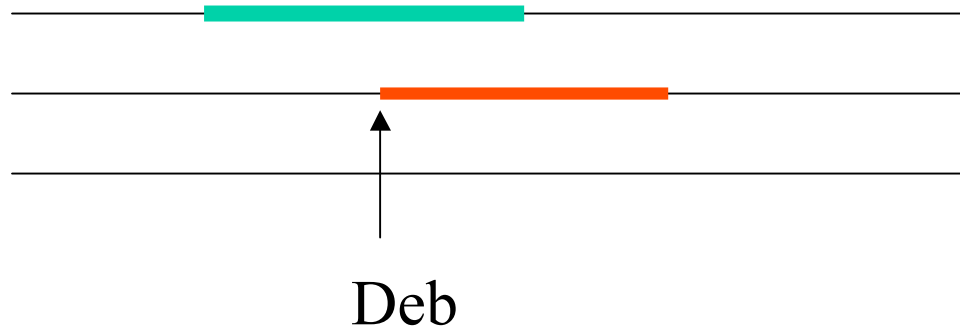
- Un référentiel spatial et temporel
- Du texte
- Des objets virtuels
- Des textures
- Des comportements dynamiques symboliques
- Des indications pour les outils de synthèse et de visualisation

Référentiel

- Un système de coordonnées 3D



- Une base de temps absolue et des évènements



Des objets virtuels

- Une forme
- Des caractéristiques lumineuses: source ou réflecteur, couleur, transparence...
- Des caractéristiques acoustiques: source ou réflecteur, caractéristique d'absorption ou de réflexion...
- Des caractéristiques physiques si l'objet est amené à régir dans le cadre d'un modèle physique: masse, élasticité...
- Une position initiale dans un repère physique

Des textures

Pointeurs vers des éléments enregistrés utilisés comme texture:

- image fixe,
- son,
- vidéo numérique

Des éléments dynamiques

- Des caractéristiques invariantes entre objets (positions relatives)
- Description comportementales propres:
 - trajectoires (équation, point de passage obligés...)
 - élément de vie propre
 - Description symbolique des signaux émis par les sources sonores ou lumineuses
- Des réaction au comportement des utilisateurs (interactivité)
- Des mécanismes de synchronisation entre les évolutions dynamiques synchrones (texture son et vidéo) et asynchrones (événements)

Des indications pour les outils de synthèse et de visualisation

Très variable:

Notion de fonte

Nature des algorithmes à utiliser

Qualité perceptives imposées (définition des
images, fréquences d'échantillonnage...)

....

Langages utilisés

- Des méta langage de type pour définir des document structurés: par exemple XML
 - Des langages de types pour décrire les données statiques: graphes de scènes VRML, MPEG4...
 - Des langages procéduraux pour les éléments dynamiques : Java(script), SMIL,...ou tout langage informatique. Nécessité d 'une API graphique et sonore de haut niveau.
 - Des langages de types pour les indicateurs aux visualisateurs/synthétiseur (séparation forme contenu)

Outils de production

Outils graphiques et sonore permettant de générer les éléments statiques de façon intuitive: modeleurs 3D, synthétiseurs midi... qui produisent du VRML ou du midi à partir d'une image ou d'une musique

Les systèmes de coordonnées

Conforme : Quatre coordonnées pour que toutes les transformations de bases soient linéaires $(x, y, z, 1)$

Les coordonnées d'un fils sont relatives à celle du père.

Les transformations géométriques de base

Translation, Rotation, Homotétie sont des opérations linéaires dans le plan conforme.

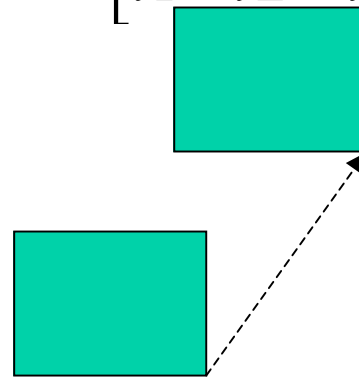
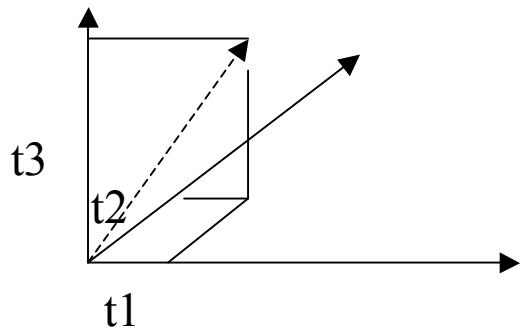
Elles peuvent être composée pour former des transformations plus complexes.

Les compositions se traduisent par des produits de matrices

Translation

Point $(x,y,z,1)$ translaté en $(x+t1,y+t2,z+t3,1)$

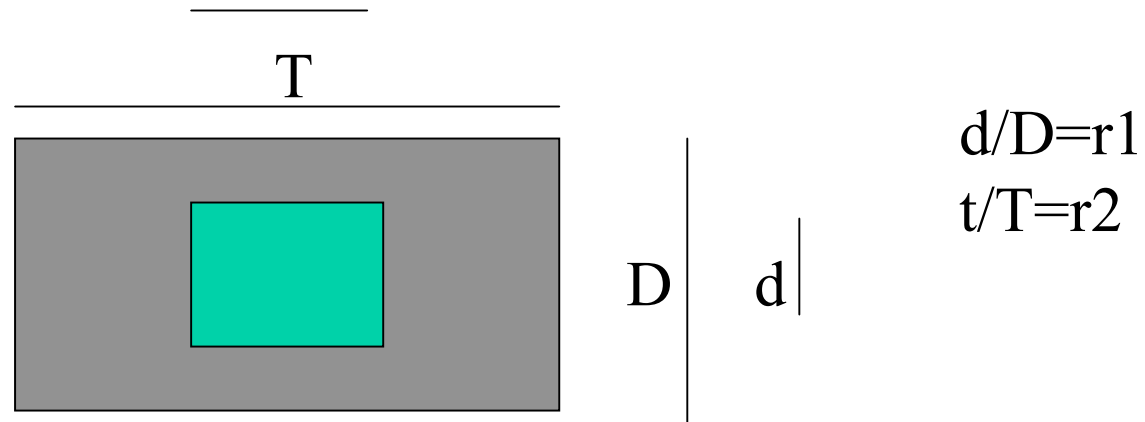
$$(x' \quad y' \quad z' \quad 1) = (x \quad y \quad z \quad 1) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t1 & t2 & t3 & 1 \end{bmatrix}$$



Homotétie

Point $(x,y,z,1)$ translaté en $(r1.x,r2.y,r3.z,1)$

$$(x' \quad y' \quad z' \quad 1) = (x \quad y \quad z \quad 1) \begin{bmatrix} r1 & 0 & 0 & 0 \\ 0 & r2 & 0 & 0 \\ 0 & 0 & r3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

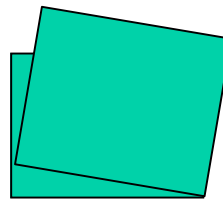
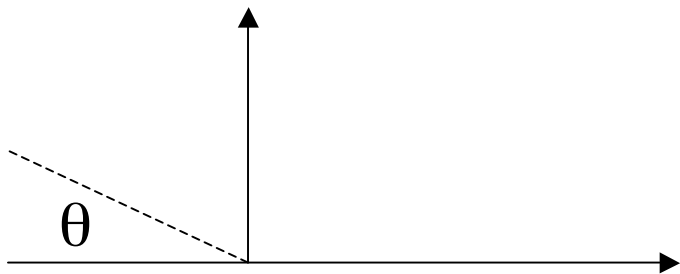


Rotation dans le plan

Point $(x,y,1)$ translaté en

$$(x.\cos(\theta) + y.\sin(\theta), x \sin(\theta) - y \cos(\theta), 1)$$

$$(x' \quad y' \quad 1) = (x \quad y \quad 1) \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ \sin(\theta) & -\cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

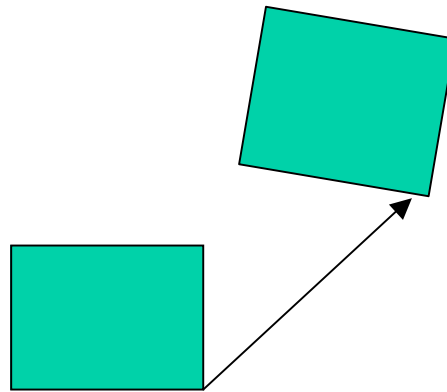
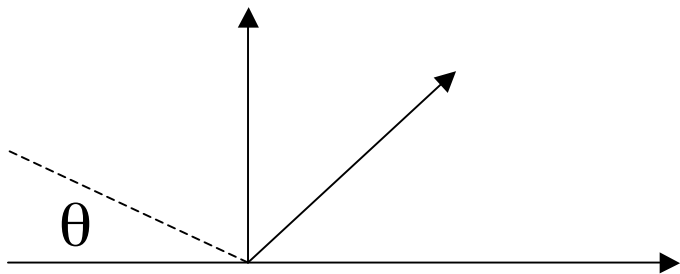


Composition d'une rotation et d'une translation

Point $(x,y,1)$ translaté en

$$(x.\cos(\theta) + y.\sin(\theta) + t1, x \sin(\theta) - y \cos(\theta) + t2,1)$$

$$(x' \quad y' \quad 1) = (x \quad y \quad 1) \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ \sin(\theta) & -\cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t1 & t2 & 1 \end{bmatrix}$$



Graphe de scène

- Représentation hiérarchisée de la scène sous forme d 'arbre
- Nœuds constructeurs
- Nœuds feuilles écrivant les objets finaux

Exemple VRML et MP4

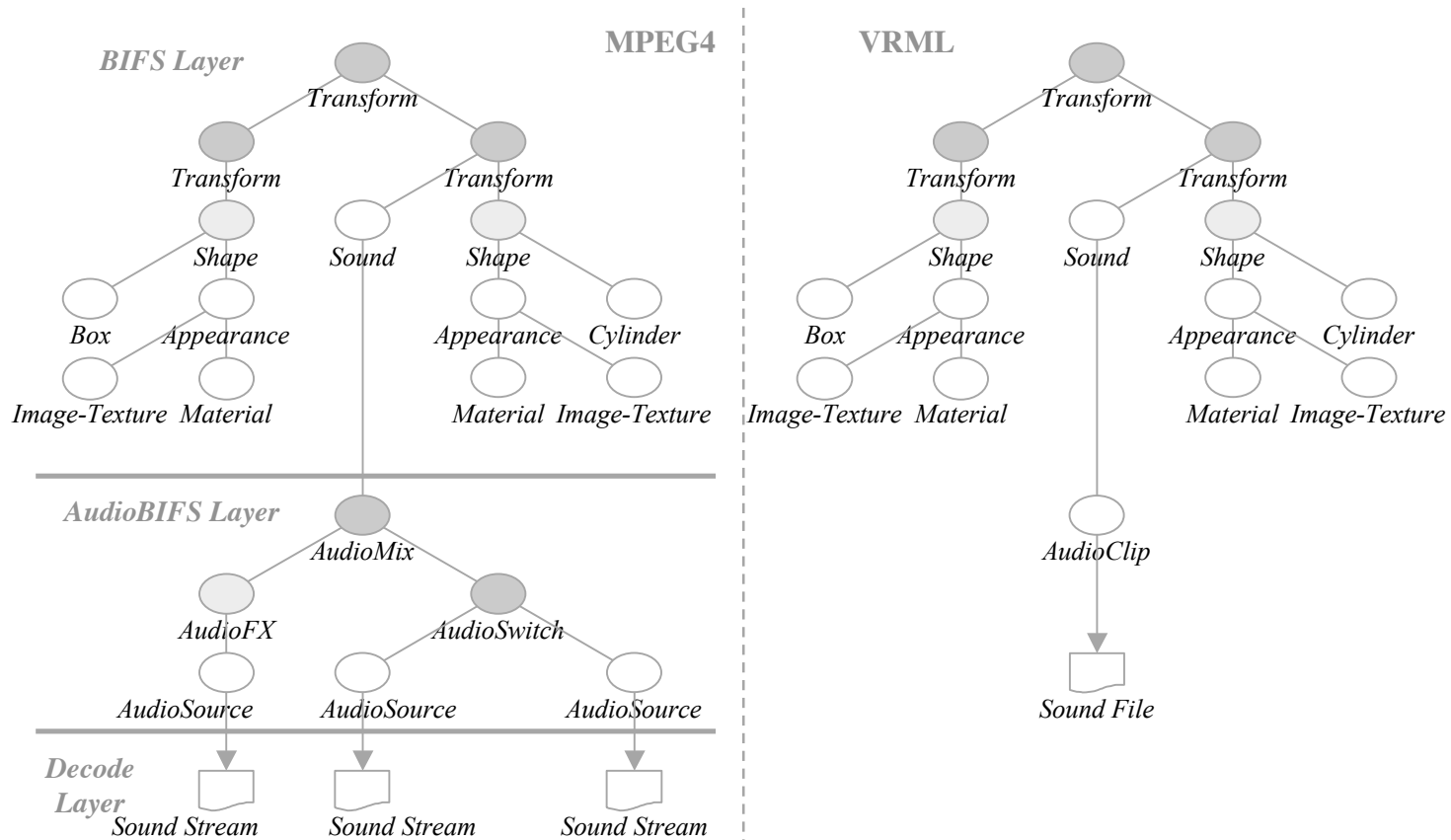


Figure 1 : MPEG4 and VRML scene graphs with sound nodes

Exemple de types de nœuds VRML

- ∇ • Background
- ∇ • Billboard
- ∇ • Collision
- ∇ • ColorInterpolator
- ∇ • CoordinateInterpolator
- ∇ • CylinderSensor
- ∇ • DirectionalLight
- ∇ • Fog
- ∇ • Group
- ∇ • Inline
- ∇ • LOD
- ∇ • NavigationInfo
- ∇ • NormalInterpolator
- PlaneSensor
- PointLight
- PositionInterpolator
- ProximitySensor
- ScalarInterpolator
- Script
- Shape
- Sound
- SpotLight
- SphereSensor
- Switch
- TimeSensor
- TouchSensor
- Transform
- Viewpoint
- VisibilitySensor
- WorldInfo

Nœud constructeurs VRML

- *Anchor* defines the nodes that act as a link to another web content,
- *Billboard* groups nodes that must be automatically oriented toward the user's viewpoint,
- *Collision* specifies the collision properties of its children,
- ***Group*** contains children nodes without introducing some specific operations,
- ***Inline*** reads its children nodes in an external url,
- *LOD* defines various level of detail for an object and gives hints to the browser for choosing the appropriate appearance,
- *Switch* traverses zero or one of the nodes specified in the *choice* field,
- *Transform* defines a local coordinate system for its children.

Exemple (1)

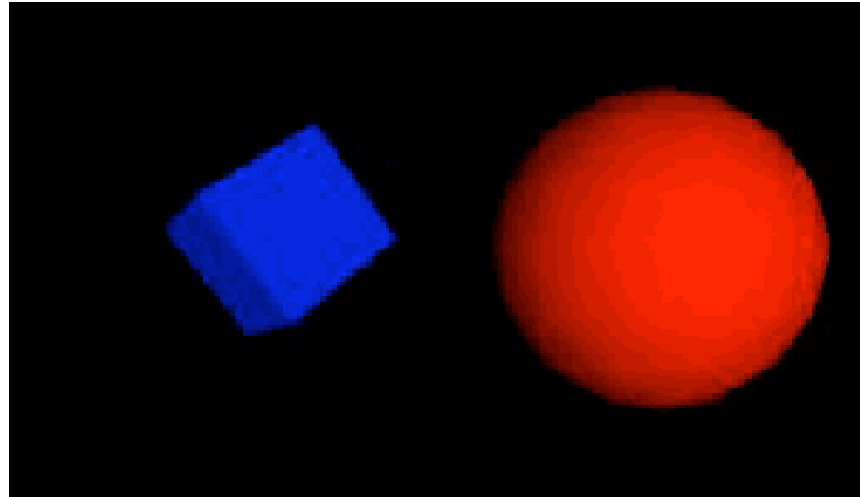
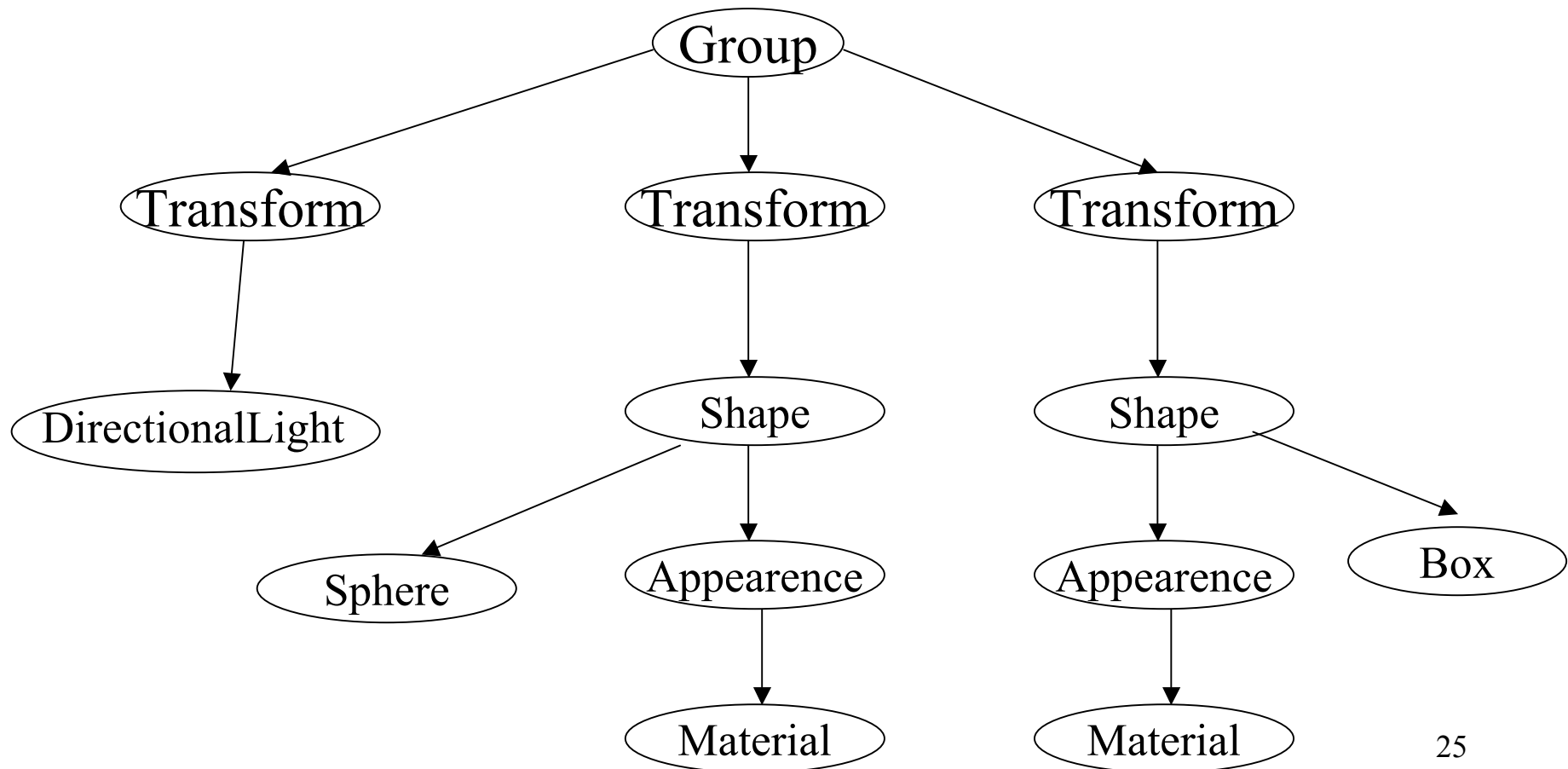


Figure D.1: Red sphere meets blue box

Exemple (2)



Exemple (3)

```
#VRML V2.0 utf8
Transform {
  children [
    NavigationInfo { headlight FALSE } # We'll add our own light

    DirectionalLight {           # First child
      direction 0 0 -1          # Light illuminating the scene
    }

    Transform {                 # Second child - a red sphere
      translation 3 0 1
      children [
        Shape {
          geometry Sphere { radius 2.3 }
          appearance Appearance {
            material Material { diffuseColor 1 0 0 } # Red
          }
        }
      ]
    }
  ]
}
```

Exemple (4)

```
Transform {                                # Third child - a blue box
  translation -2.4 .2 1
  rotation    0 1 1 .9
  children [
    Shape {
      geometry Box {}
      appearance Appearance {
        material Material { diffuseColor 0 0 1 } # Blue
      }
    }
  ]
}

] # end of children for world
}
```

Les évènements

Certains objets ont la possibilité d'émettre ou de recevoir des messages appelés évènements vers d'autres objets de la scène (`eventIn`, `eventOut`)

Les seuls évènements externe récupérables sont les commandes souris captés via le nœud `TouchSensor`

Certains attributs d'un nœud (`exposedField`) peuvent être envoyés comme évènements (`attr_changed`) ou modifiés par un évènement (`set_attr`)

Le routage des évènements

La relation entre évènements émis et reçus est définie par une définition de canaux de messages:

ROUTE..TO

Exemple son

```
#VRML V2.0 utf8DEF MyForm Transform { children [
```

Exemple son (2)

Nœuds:

Transform (MyForm)

TouchSensor (Mouse)

TimeSensor (Timer)

Shape

Sphere

Sound

AudioClip (MySound)

Exemple son (3)

Affiche un objet (Shape) ayant une forme de sphère (Sphere) de rayon une unité (par défaut) et positionné à l'origine de la scène puisque aucune transformation n'est donnée dans le noeud Transform.

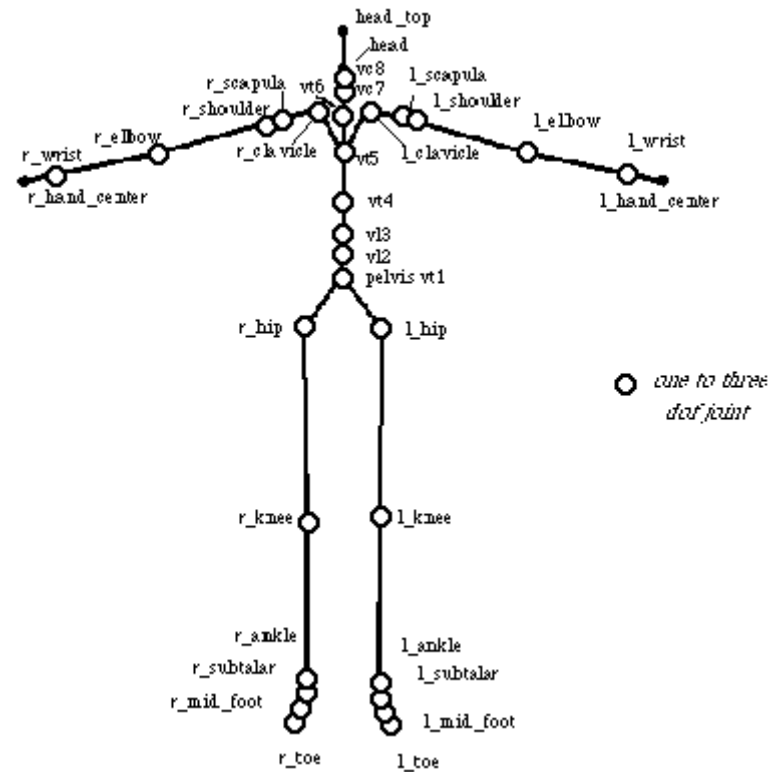
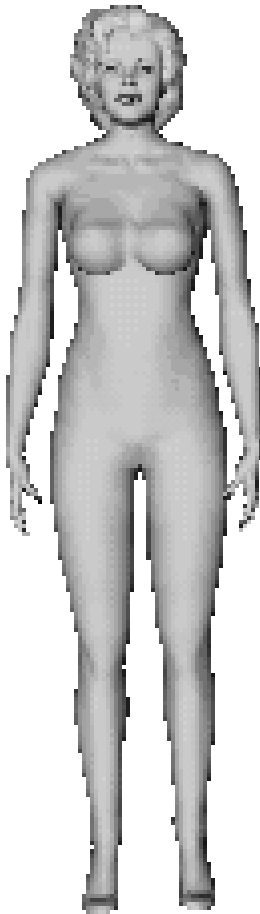
La sphère est coloriée en blanc car aucune apparence n'est donnée à la forme géométrique (absence de noeud Appearance attaché au champ appearance du noeud Shape).

Le noeud TouchSensor rend sensible aux événements souris tous ses noeuds frères.

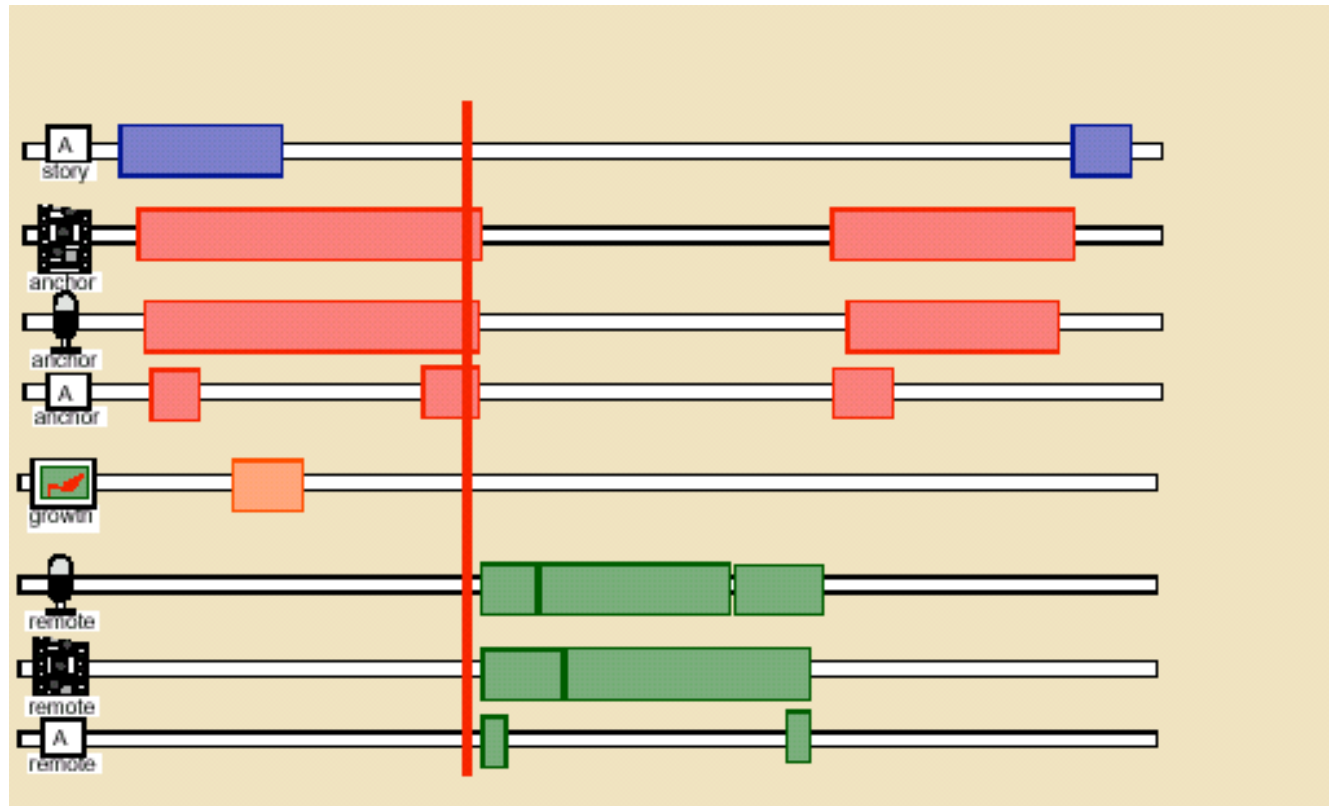
Ainsi, quand l'utilisateur clique sur la sphère, cela démarre le timer ayant un cycle de 3 secondes (par le premier routage d'événement).

Lorsque le timer démarre un nouveau cycle après que le cycle précédent ait été terminé, le son spécifié dans le noeud Sound est joué (deuxième routage). Ainsi, il n'est pas possible de rejouer le son à moins de 3 secondes d'intervalle.

La représentation des humains dans MPEG 4



Exemple la gestion du temps dans SMIL



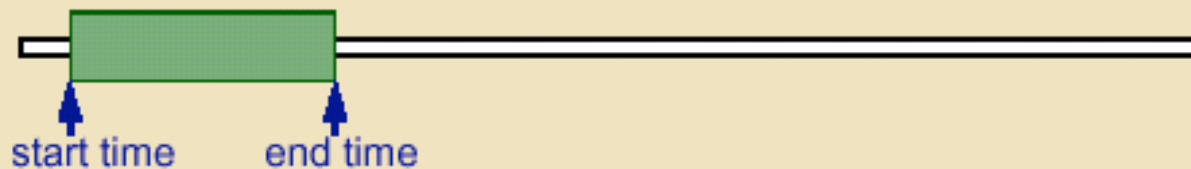
Le temps dans SMIL

Types of time:

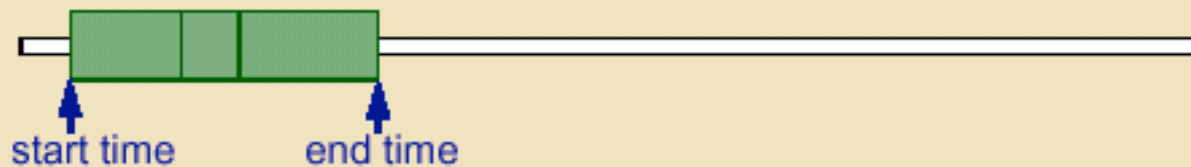
- media item time axis
 - video divided in frames, audio sampled at 44kHz



- document time
 - image starts at certain time and ends at a later time

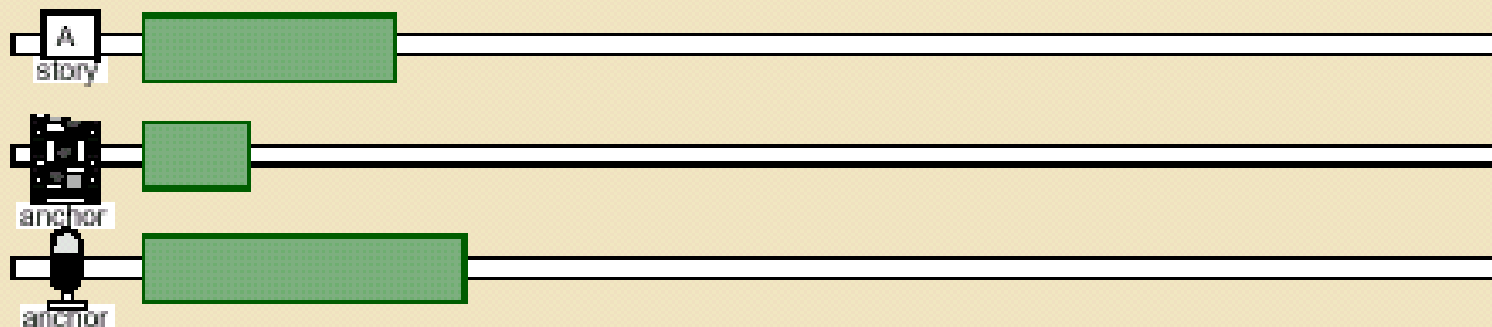


- run-time presentation
 - video data bits get caught up in network, so end time is delayed



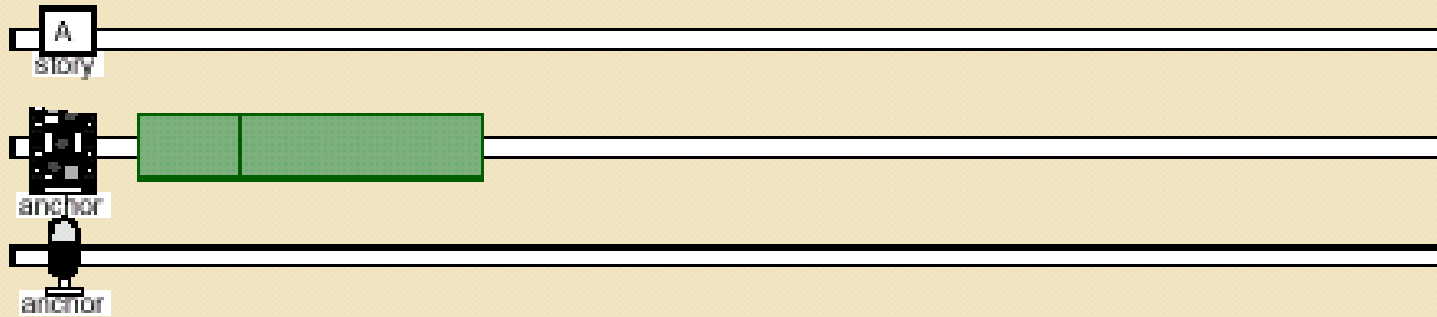
Constructeur PAR

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



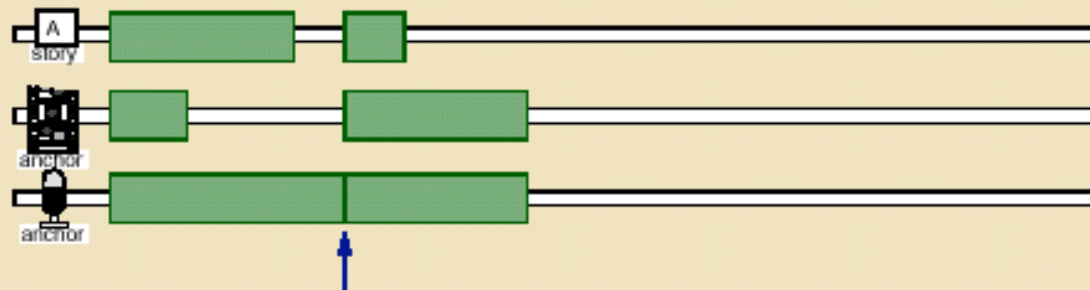
Constructeur SEQ

```
<seq>  
  <video src="logo.mpg" region="V-main" />  
  <video src="anchor.mpg" region="V-main" />  
</seq>
```



Combinaison des deux

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



La norme MIDI (1)

La norme midi est en fait une norme d'architecture de Réseaux de communication a trois couches

- Physique
- Liaison
- Présentation

Physique: norme d'interconnections et de branchement, signalisation en Bande de base

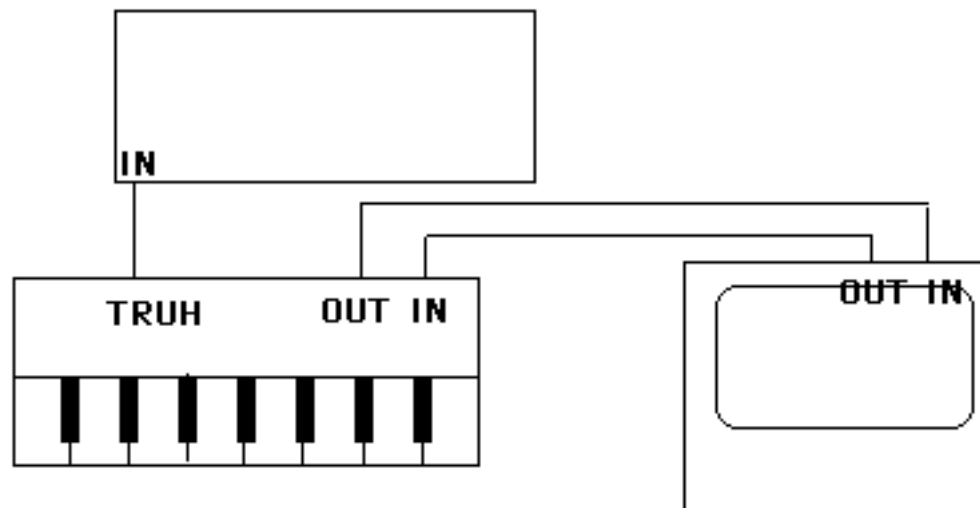
Liaison: protocole multipoint en mode caractères centralise

Adressage: notion de canal midi (16)

Présentation: représentation de commandes pour Instruments midi.

C'est un mode de représentation symbolique de la musique comparable aux Objets d'une scène 3d en synthèse d'image

La norme MIDI (2)



Les messages Midi

TROIS CLASSES

CHANNEL MESSAGES

ENVOYES A UN DESTINATAIRE

EXEMPLE

MIDI MODE

NOTE-ON, NOTE -OFF

MONO, POLY PRESSURE, CONTINUOUS CONTROLLER, PITCH

BEND

PROGRAM CHANGE

SYSTEM MESSAGES

ENVOYE A TOUS LES DESTINATAIRES PAR LE MAITRE

EXEMPLES

MIDI CLOCK

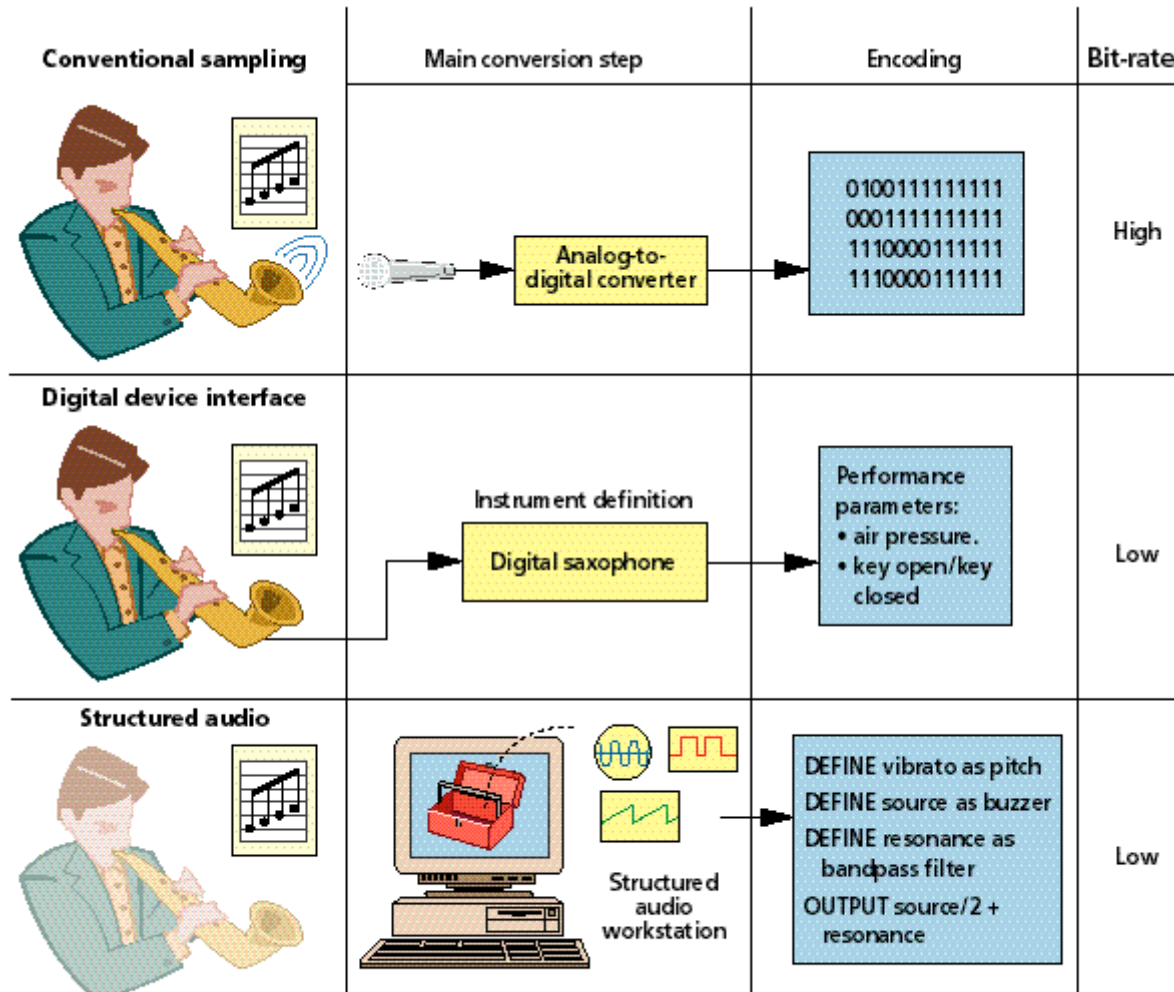
START, STOP, CONTINUE

SONG POSITION POINTER

SYSTEM EXCLUSIVE

SPECIFIQUE A UNE MARQUE D'INSTRUMENT

Le son dans MPEG 4



MPEG 4 Architecture

