



## Descriptions de scènes multimédia

Cyril Concolato

SI350

01/06/2010



## Principes généraux

Comment fonctionnent les descriptions de scènes existantes ?

page 1

SI350

Descriptions de scènes multimédia



## Exemples de scènes multimédia



DVD



Télévision interactive



Diaporama musical



Portail de vidéo à la demande



Dessins animés



Cartographie interactive

page 2

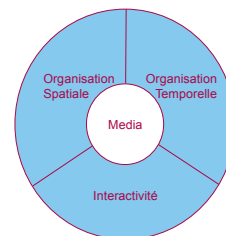
SI350

Descriptions de scènes multimédia



## Scène et description de scène multimédia

- **Scène**
  - Ensemble de données audiovisuelles synthétiques ou naturelles
  - Organisées et présentées selon les choix d'un auteur
- **Description de scène**
  - Information principalement déclarative
  - Utilisation des médias
  - Organisation selon 3 axes
  - Exemples: SVG, BIFS, Flash, LAsER, XHTML, VRML, ...



page 3

SI350

Descriptions de scènes multimédia



## Traitements (côté client) d'une description de scène

- **Chargement de la description**
  - À partir d'un document textuel (e.g. XML) ou
  - À partir d'un flux binaire
- **Créer une représentation interne de la scène**
- **Composer la scène**
  - Synchroniser/contrôler les éléments média (audio, vidéo)
  - Traiter les interactions
  - Effectuer les animations (interpolations)
- **Produire une image 'raster'**
  - En effectuant le tracé des primitives graphiques
  - En effectuant le tracé des textes
  - En positionnant et en intégrant/fusionnant les média visuels naturels (image, vidéo)
- **Produire le son final**
  - Créer les sons synthétiques
  - Mixer les sons décodés

page 4

SI350

Descriptions de scènes multimédia



## Structuration des descriptions (1/2)

- **Représentation hiérarchique de l'information**
  - Arbre (ou Graphe)
    - Structuration selon le positionnement spatial (SVG, BIFS, Flash)
    - Structuration selon le positionnement temporel (SMIL, BIFS)
  - Représentations arborescentes (DOM, VRML/BIFS, Flash)

page 5

SI350

Descriptions de scènes multimédia



## Structuration des descriptions (2/2)

### ■ Représentations temporelles

- Tous les états de la scène sont donnés sous la forme d'un seul arbre
  - Notion de document, format XML
  - Avantages: connaissance à l'avance de tous les états
- Flux de scène composé d'un arbre initial et de mises à jour
  - Avantages: méthodes de distribution (ex: streaming)

page 6

SI350

Descriptions de scènes multimédia



## Descriptions de scènes et interactivité

### ■ Modèle événementiel en 3 parties

#### ■ Capture des événements applicatifs

- Dépendant du terminal et des périphériques d'entrée/sortie
- Génération d'événements internes à la scène
  - Par les écouteurs (listener)
  - Événements simples (booléen, entier) ou complexes (interfaces manipulable par script)

#### ■ Transmission et propagation des événements

- Mécanismes existants
  - ROUTE selon VRML
  - Propagation selon DOM Events: capture, ciblage, bouillonnement et action par défaut

#### ■ Modification de la scène et génération d'événements secondaires

- Par des gestionnaires d'événements (handler)



page 7

SI350

Descriptions de scènes multimédia



## Modifications de la scène

### ■ 2 modes d'interactivité

- Sans script / par mise à jour
  - Conditional (BIFS et LASeR)
- Avec (ECMA)Script : différentes API
  - Accès à l'arbre de scène (navigation, lecture et modifications)
    - VRMLScript (VRML et BIFS)
    - JavaScript: DOM Core, DOM Events
    - ActionScript

page 8

SI350

Descriptions de scènes multimédia



## Scénarios d'interactivité

### ■ Scénarios orientés client

- Traitement intelligent de l'interaction côté client
  - Nécessite une puissance de calcul +/- importante
- Indépendance entre serveur de données et serveur de scène
  - Réduit la charge du serveur de scène
  - Peut nécessiter plusieurs voies de retour
- Exemples dans le monde de l'internet : Applications « AJAX » (Asynchronous Javascript And XML)
  - = Description de scène + Ensemble de code ECMAScript
  - + Téléchargement (asynchrone) de données XML
  - + Interprétation des données XML
  - + Manipulation/Modifications de la scène
  - SVG + DOM + XMLHttpRequest / Flash + ActionScript + LoadVars + XML

### ■ Scénarios orientés serveur

- Traitement et agrégation des données côté serveur
  - Charge serveur importante
- Préparation des données au format de la scène
  - Un service différent par type de scène
- Transmission des données en mode « push » par mises à jour
  - Faible puissance de calcul nécessaire côté client
- Exemple dans le monde de la télévision numérique : MPEG-4 BIFS (noeud Anchor/ServerCommand + Updates)

page 9

SI350

Descriptions de scènes multimédia



## Organisation Temporelle

### ■ Notion à la base de l'animation

#### ■ 2 modèles:

- Hierarchie temporelle des média:
  - SMIL (par, seq) and BIFS (flextime)
- Positionnement d'évènements temporels sur une ligne temporelle
  - VRML (start-time, stop-time), BIFS, Flash, SVG

#### ■ Synchronisation avec les média

- Références et utilisation de temps dans la base de temps des média: BIFS (mediaStartTime), SMIL (clipBegin)

page 10

SI350

Descriptions de scènes multimédia



## Descriptions de scènes et animations

### ■ But

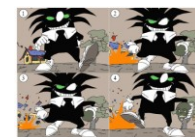
- Représenter les évolutions temporelles de la scène

### ■ Applications

- Dessins animés, bandeaux publicitaires animés
- Sous-titrage, suivi d'objets

### ■ Représentations possibles

- Animations autonomes
  - Interpolations pilotées par le lecteur
- Animations asservies
  - Par mises à jour émises par un serveur



page 11

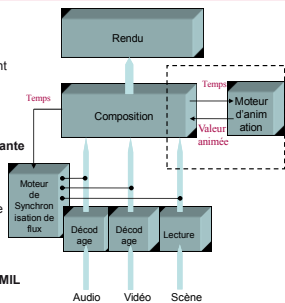
SI350

Descriptions de scènes multimédia



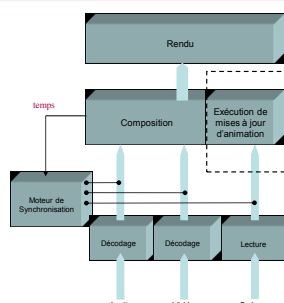
## Animations autonomes

- **Gestion du temps piloté par le lecteur**
  - Fréquence de rafraîchissement variable
- **Animation par interpolation**
  - Simple: discrète ou linéaire
  - Complexe: spline
- **Modification de la scène résultante simple**
  - Modification d'une valeur
- **Fonctionnement**
  - Animation entièrement connue
  - Simple (interpolation et copie)
  - Complexe
    - Animation de groupe
    - Combinaison d'animations
- **Exemples: BIFS/VRML, SVG, SMIL**



## Animations asservies

- **Réception de commandes**
  - Issues d'un serveur
  - Lues dans un fichier
- **Exécution à instants déterminés**
  - Temps de composition
  - Synchronisation avec les autres média
- **Fonctionnement simple**
- **Modifications de la scène**
  - Simples
    - Remplacement de valeurs (équivalence interpolation)
  - Complexes
    - Insertion, suppression d'objets
    - Modification de valeurs
- **Exemples: BIFS, LAsEr, Flash, REX**



## Organisation Spatiale

- **Dépend principalement du type de scène**
- **Scène 2D uniquement:**
  - SMIL (root-layout element), SVG (svg element)
  - Flash
- **Scène 3D uniquement:**
  - VRML
- **2D et/ou 3D**
  - BIFS (sous-arbres 2D ou 3D)

## Exemple d'un personnage (BIFS, SVG)

- **Ensemble d'éléments**
  - <Shape> en BIFS
  - <path> en SVG
- **Groupés**
  - Dans un <Transform2D> en BIFS
  - Dans un <g> en SVG



## Références Media

- **Références directes:**
  - SMIL, SVG, VRML et BIFS en utilisant les URLs traditionnelles (http://, file://, ...)
  - Dépendance média / scène
- **Références indirectes:**
  - Utilisation
    - en BIFS de MPEG-4 OD
    - en LAsEr de SAF
  - Permet une certaine indépendance entre la scène et les média

## Comparatifs des langages existants

Langage	Organisation spatiale	Organisation temporelle	Interactivité	Média
HTML	2D / ligne / régions hiérarchiques	Notion de temps non définie / Utilisation de Script = DHTML	Navigation simple / Interactivité complexe par script	Tout type de média (sans contrôle a priori - plugins) Pas de primitives graphiques
SMIL	2D / régions hiérarchiques	Multiples bases de temps hiérarchiques	Navigation simple / Interactivité simple à base d'animations	Tout type de média Pas de primitives graphiques
SVG	2D / positionnement matriciel	Une base de temps par document, une base de temps (potentielle) par média	Navigation simple / Interactivité complexe par script / Manipulation simplifiée de la scène (MicroDOM)	Tout type de média (sans contrôle de base de temps) Primitives graphiques 2D (simple ou complexe)
LAsEr	SVG + positionnement implicite	SVG + liens entre bases de temps média possible	SVG + déclenchement interactif de mises à jour	Tout type de média (avec contrôle de base de temps) Primitives graphiques SVG + extensions clipping
BIFS	2D et/ou 3D / positionnement matriciel ou implicite	Une base de temps par flux / Possibilité d'intégrer des flux desynchronisés	VRML + déclenchement interactif de mises à jour + interactions clavier ...	Tout type de média (tout contrôle) / Primitives 2D et 3D
VRML	3D / positionnement matriciel	Une base de temps par document	Navigation simple / Interactivité par script	Tout type de média (sans contrôle) / Primitives 3D
Flash	2D / positionnement matriciel	Une base de temps par sprite / Intégration de plusieurs sprites	Interactivité par script	Tout type de média (tout contrôle) / Primitives 2D

## Descriptions de scènes et compression

### ■ Buts

- Réduire le débit des scènes
  - Ex: dessins animés > 1 Mb/s
- Augmenter la vitesse de lecture
  - Réduction du temps de chargement

### ■ Problématique

- Différencier structure et données
  - Sélection des outils appropriés
- Différencier les types de description
  - Structurées ou non
  - Volumineuses ou non

### ■ VRML, SMIL and SVG:

- Formats textuels, potentiellement zippé

### ■ BIFS, LAsER, Flash

- Formats binaires

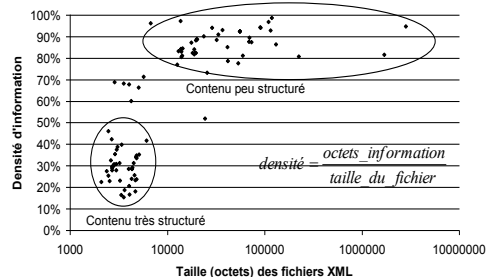
page 18

SI350

Descriptions de scènes multimédia



## Densité d'information



page 19

SI350

Descriptions de scènes multimédia



## Codages des structures

### ■ Mécanismes génériques

- Réutilisation de sous-arbres (dictionnaire évolué)
  - Simple d'utilisation mais peu flexible
- Styles et héritage de propriétés
  - Simple mais peu flexible et potentiellement complexe

### ■ Mécanismes spécifiques par langage

- Codage contextuel BIFS ad-hoc pour les scènes de type VRML
- Codage MPEG-7 BIM basé sur les schémas XML

### ■ Mécanismes spécifiques à un contenu particulier ou à une classe de contenus

- Utilisation de sous-arbres paramétrables
  - Prototypes VRML/BIFS
  - Transformations XSLT
  - Composants XBL/sXBL/XBL2

page 20

SI350

Descriptions de scènes multimédia



## Compression de la structure BIFS

### ■ Forte redondance de la structure des nœuds dans les scènes

- Nombre de Shape
  - ~# Nombre de Appearance
  - ~# Nombre de Material2D
  - ~# Nombre de IFS2D + ILS2D
  - ~# Nombre de Coordinate2D

### ■ Définition de prototypes de nœuds

- Complémentaire avec le mécanisme DEF/USE
- Méthode transposable à d'autres technologies comme XBL

### ■ Gain

- (gain d'un prototype) x (nombre d'utilisations) - (coût de la définition)
- observé : 10% de la structure

```
<Shape>
  <appearance DEF="a1">
    <Material2D materialColor="0.78 0.78 0.72">
      <linePropag>
        <lineProperties lineColor="0 0">
          <linePropag>
            <Material2D>
              <appearance>
                <appearance>
                  <appearance>
                    <Coordinate2D colorPerVertex="false">
                      <Coordinate2D point="1147 0 -_"/>
                    </Coordinate2D>
                  </Coordinate2D>
                </Material2D>
              </appearance>
            </linePropag>
          </linePropag>
        </Material2D>
      </appearance>
    </Shape>
```

page 21

SI350

Descriptions de scènes multimédia



## Compression de la structure SVG

### ■ Structure de scène différente de BIFS

- Moins d'éléments, plus d'attributs
  - Arbre plus plat
    - 1 objet graphique SVG : profondeur = 1
    - 1 objet graphique BIFS : profondeur = 4
  - Utilisation de prototypes
    - Inefficace au niveau de l'objet graphique

### ■ Statistiques intéressantes

- Faible nombre d'éléments différents
  - Il existe des éléments très probables (path, g, text, ...)
  - et des éléments peu probables (video, ...)
- Faible nombre d'attributs différents
  - Tous éléments confondus
  - Il existe des attributs très probables (transform, fill, ...)
  - et des attributs peu probables (audio-level, ...)

page 22

SI350

Descriptions de scènes multimédia



## Codage des données

### ■ Types de données majoritaires dans les contenus volumineux

- Listes de points, couleurs, chaînes de caractères, matrices ...

### ■ Mécanismes sans perte

- Algorithmes de codage non typés (LZW)
- Codage sur un nombre fixe de bits (IEEE 754 32-bits)
- Codage sur un nombre de bits variable (BIFS « Efficient Float Coding »)

### ■ Quantification des données vectorielles

- Résolution fixe (Flash, 20° pixel) ou variable (BIFS, LAsER)
- Résolution globale (Flash, LAsER) ou par objet (BIFS)
- Opérateurs à virgule fixe (LAsER, Flash) ou flottante (BIFS)

page 23

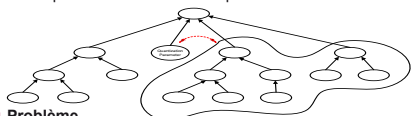
SI350

Descriptions de scènes multimédia



## Signalisation de la quantification

- **Signalisation globale (Flash et LAsER)**
  - Tous les points sont quantifiés de la même façon
- **Signalisation par objet (BIFS)**
  - Une partie de l'arbre de scène peut être quantifiée indépendamment des autres parties



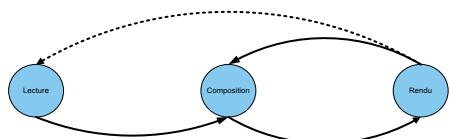
- **Problème**
  - Coût de signalisation
    - 100~200 bits par nœud QP (min, max, nb bits, flags)

## Problèmes d'artefacts de quantification

- Si deux objets graphiques partageant un même tracé sont quantifiés avec des paramètres différents, des trous apparaissent
- **Principe**
  - Une seule grille de quantification par objet 'sémantique'
  - Grouper tous les points d'un objet 'sémantique' dans une liste commune de points
  - Utilisation des index sur cette liste
- **Problème**
  - Coût de l'indexation vs. coût du codage des points



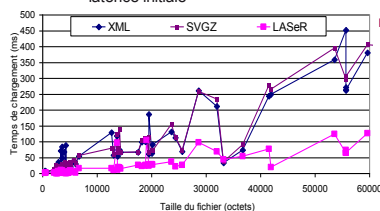
## Visualisation d'une scène



- **Problèmes**
  - Rapidité du cycle de présentation (frame rate)
    - Minimiser l'utilisation CPU pour cibler des machines peu puissantes
  - Minimisation de la consommation mémoire
    - Pour pouvoir lire des contenus plus volumineux

## Lecture optimisée de scène

- **Rapidité de lecture des données sources**
  - La lecture rapide d'unité d'accès permet de laisser du temps aux autres étapes du processus de visualisation
  - La lecture rapide d'un document complet permet de réduire la latence initiale



- **Recommandations**
  - Lecture binaire
  - Lecture progressive
  - Fragmentation

## Consommation mémoire

- **Buts**
  - Réduire la consommation mémoire des contenus volumineux
  - Permettre la création rapide de l'arbre de scène
- **Possibilité d'améliorations limitées**
  - Nécessité d'un accès rapide aux données pendant la phase de composition (lecture et écriture)
    - Impossible de conserver les objets compressés en mémoire
  - Nécessité de conserver une structuration cohérente avec les interfaces programmatiques de manipulation de l'arbre
    - Structuration des objets mémoire dépendante du langage de scène

## Structuration BIFS vs. structuration SVG

- **Scènes BIFS**
  - Beaucoup de nœuds très différents (> 170)
  - Peu de propriétés par nœuds (~4)
- **Allocation possible d'un nœud en un bloc**
  - Occupation mémoire des propriétés non spécifiées négligeable
- **Scène SVG**
  - Peu d'éléments dans le langage (~50)
  - Beaucoup d'attributs potentiels par éléments (~65 pour l'élément <rect>)
- **Allocation par bloc**
  - Surconsommation des attributs non spécifiés
- **Contribution**
  - Allocation générique des éléments



## Méthode d'allocations

### ■ Structuration BIFS et SVG 1 ■ Structuration SVG 2

```
typedef struct {
  type1 propriété_1;
  type2 propriété_2;
  type3 propriété_3;
} Record_1;
```

```
struct AttributSVG {
  int identifiant_Attribut;
  void *valeur;
};
struct ElementSVG {
  int type;
  struct AttributSVG *attributs;
};
```

Séquence SVG	Taille du fichier (Ko)	Nombre d'éléments	Nombre d'attributs	Consommation Mémoire (Ko)	
				Structuration 1	Structuration 2
Plan.svg	38 395	287 059	246 296	259 120	56 233
GareDuNord.svg	1 647	13 263	50 891	13 903	5 801
Plan_layer_1.svg	7 130	12 896	27 659	13 845	5 206
Face_frame_1.svg	94	490	980	555	228
Cee.svg	115	81	99	407	306
Butterfly.svg	31	4	14	92	90

### ■ Gain consommation mémoire important sur les contenus volumineux

page 30

SI350

Descriptions de scènes multimédia



## Composition de scènes

### ■ Fonctionnement

- Composition des données issues de plusieurs sources
  - Application des mise à jours lues/reçues à des fréquences différentes
- Gestion des animations par interpolations
  - Gestion du temps, calcul d'interpolation et application
- Exécution des mécanismes d'interactivité
  - Génération d'évènements, propagation et modification(s)

### ■ Problématiques

- Plus la scène est volumineuse, plus l'arbre est gros, plus la composition est lente
- L'utilisation des normes CSS et SMIL Animation implique le maintien de 3 arbres: arbre DOM, arbre CSS, arbre de présentation

page 31

SI350

Descriptions de scènes multimédia



## Rendu visuel de scènes

### ■ But: Produire l'affichage d'une scène composée

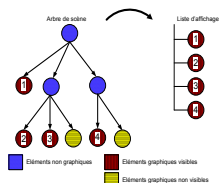
- Évènements, mises à jour et animations appliqués

### ■ Principes généraux du rendu 2D

- Parcours des objets graphiques à afficher
- Création d'une représentation pixel à partir d'une représentation vectorielle des objets graphiques
- Fusion des objets graphiques avec les objets naturels (vidéo et image)

### ■ Problèmes

- Limiter l'accès à la mémoire vidéo d'une trame à l'autre
- Déterminer les zones de la fenêtre à retracer
- Déterminer les objets nouveaux, supprimés ou modifiés
- Utilisation de marqueurs de changement



page 32

SI350

Descriptions de scènes multimédia



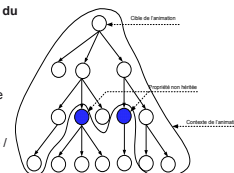
## Rendu optimisé de scènes avec héritage

### ■ Une animation peut modifier un sous-arbre (héritage simple) ou une fraction du sous-arbre (semi-héritage)

- Nécessité de traverser tout l'arbre à nouveau

### ■ Contribution

- Héritage simultané de marqueurs et de propriétés
- Grouper les propriétés par type de marqueurs de changement (géométrie / apparence / textuel ...)



### ■ Résultats

- Fortement dépendant du type de contenu
- Vitesse de rendu multipliée par 9 dans des contenus utilisant l'héritage

### ■ Limitations

- Quantité limitée de marqueurs / groupement des marqueurs

page 33

SI350

Descriptions de scènes multimédia



## Aspects économiques

Pourquoi et où utiliser une description de scène ?

### ■ Télévision (fixe et mobile)

- DVB (MHP)
- ATSC
- ISDB (BML)

### ■ Téléphonie mobile

- 3GPP DIMS
- OMA RME

### ■ Internet

- HTML
- Flash

### ■ Radio numérique

- BIFS

### ■ Home Network

- UPnP, DLNA et Web4CE

### ■ Besoins

- Solution standard
- Édition facile
- Tout mode de transport

### ■ Applications

- Interfaces graphiques
- Guide de programme
- Moteurs de recherche

page 34

SI350

Descriptions de scènes multimédia



page 35

SI350

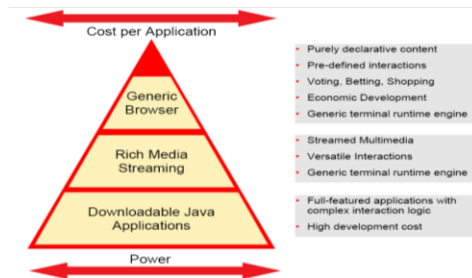
Descriptions de scènes multimédia



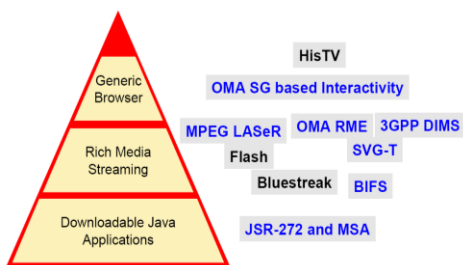
## Convergence des domaines

- **VRML, SML, SVG:**
  - Langages dédiés au Web
  - Intégration native avec les autres outils du Web (CSS, XML...)
  - Limitation: streaming
- **BIFS, LAsEr, Flash:**
  - Multi-environnements: Internet, TV, Mobile
  - Outils supplémentaires: streaming, carrouselling

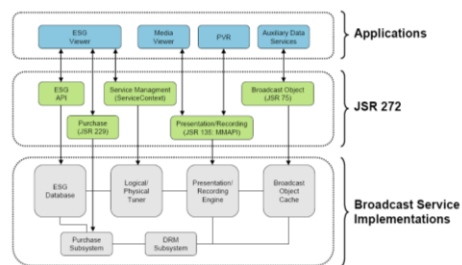
## Différence entre Scène et Programme (selon Vodafone)



## Le paysage actuel



## JSR 272: Mobile Broadcast Service API for Handheld Terminals



## Le langage BIFS

### Le langage BIFS

### BIFS – Binary Format for Scene

- **Premier format historiquement (1999)**
- **Basé sur VRML 1997 (langage pour la 3D)**
- **Ajouts du langage**
  - Primitives 2D, amélioration du texte ...
  - Primitives de contrôle de média (play, pause, stop)
  - Commandes de mise à jour = BIFS Update
  - Capacités graphiques à la Flash
- **Format binaire**
  - Codage contextuel de l'arbre de scène
  - Utilisation de techniques de codage : quantification, codage arithmétique
- **Formats d'échange XML: XMT**
- **Interactivité avancée: Java (MPEG-J) ou ECMAScript**
- **Utilisation actuelle:**
  - Standard de télévision mobile en Corée (DMB)
  - Standard de radio numérique en France

## Principes du langage BIFS

- **Scène BIFS = Arbre BIFS = ensemble de nœuds**
- **Nœud BIFS = Primitive de base**
  - Chaque nœud à un type dérivant d'un type de base (SFNode)
  - Types de nœuds
    - Nœuds pour déclarer des objets visuels
    - Nœuds pour déclarer des objets sonores
    - Nœuds pour gérer le positionnement
    - Nœuds pour gérer l'interactivité
    - Nœuds pour gérer l'animation
  - Chaque type de nœud possède un ensemble de propriétés
    - Propriétés à une valeur (SF)
    - Propriétés à plusieurs valeurs (liste ou tableau) (MF)
  - Un nœud peut avoir un identifiant numérique ou littéral

page 42

SI350

Descriptions de scènes multimédia



## Principes du langage BIFS (2)

- **Valeurs BIFS = données typées**
- **Types de données simples**
  - Booléens (SFBool/MFBool)
  - Entiers (32 bits) (SFInt32/MFInt32)
  - Nombres Réels (32 bits) (SFFloat/MFFloat)
  - Vecteur réel 2D (SFVec2f, MFVec2f)
  - Vecteur réel 3D (SFVec3f, MFVec3f)
  - Couleur (R,G,B)  $\in [0, 1]^3$  sur 32 bits (SFColor, MFColor)
  - Temps en secondes, nombre réel sur 64 bits (SFTime, MFTime)
  - Chaîne de caractères (UTF-8) (SFString, MFString)
- **Types de données complexes**
  - Nœuds (SFNode, MFNode)
  - Commandes BIFS (SFCommandBuffer)
  - Code ECMAScript (SFScript)

page 43

SI350

Descriptions de scènes multimédia



## Principes du langage BIFS (3)

- **Mécanisme évènementiel**
  - Évènement + transmission
- **Type évènementiels des propriétés**
  - eventIn – peut recevoir des évènements (write only)
  - eventOut – peut émettre des évènements (read-only)
  - field – ne peut ni émettre, ni recevoir (valeur fixe)
  - exposedField – peut émettre et recevoir (read/write)
- **Mécanisme de transmission = ROUTE**
  - Nœud émetteur + propriété émettrice (eventOut)
  - Nœud récepteur + propriété réceptrice (eventIn)
  - !! Propriétés de même type de données
    - Sfx -> Sfx et pas Sfx -> SFy

page 44

SI350

Descriptions de scènes multimédia



## Principes du langage BIFS (Résumé)

- **1 scène BIFS =**
  - 1 arbre de nœuds +
  - 1 ensemble de routes
- **1 nœud =**
  - 1 identifiant (optionnel)
  - 1 type de nœud
  - 1 ensemble de propriétés
- **1 propriété =**
  - 1 nom
  - 1 type de données
  - 1 type évènementiel
  - 1 valeur par défaut
- **1 route =**
  - 1 identifiant (optionnel)
    - Émetteur
      - Identifiant du nœud
      - Propriété eventIn
    - Récepteur
      - Identifiant du nœud
      - Propriété eventOut

page 45

SI350

Descriptions de scènes multimédia



## Nœuds BIFS visuels

- **Un objet visuel = un nœud « Shape » =**
  - 2 propriétés :
    - exposedField SFNode appearance NULL
    - exposedField SFNode geometry NULL
- **Géométrie d'une forme**
  - Nœuds 2D (Rectangle, Ellipse, Curve2D ...)
  - Nœuds 3D (Box, Sphere, IndexedFaceSet ...)
- **Apparence d'une forme = nœud « Appearance »**
  - 3 propriétés:
    - exposedField SFNode material NULL
    - exposedField SFNode texture NULL
    - exposedField SFNode textureTransform NULL

page 46

SI350

Descriptions de scènes multimédia



## Propriétés d'apparence BIFS

- **Propriétés « material » = apparence synthétique =**
  - Nœud « Material » = apparence uniforme 3D
    - Résultat de l'éclairage ombragé
  - Nœud « Material2D » = apparence uniforme 2D
    - Rempli ou non / Couleur de remplissage / Transparence
    - Propriété de tracé de contour = « LineProps »
      - Nœud « LineProperties » = Couleur/Épaisseur/style du tracé
        - Nœud « XLInProperties » = LineProperties++ (centrage, texture, join, terminaisons...)
- **Propriété « texture » = apparence raster =**
  - Nœud « ImageTexture » = utilisation d'une image
  - Nœud « MovieTexture » = utilisation d'une vidéo
  - Nœud « CompositeTexture(2D/3D) » = résultat de la composition d'un sous arbre
  - Nœud « PixelTexture » = utilisation d'un motif simple
  - Nœud « RadialGradient » ou « LinearGradient »
- **Propriété « textureTransform » = transformation sur la texture**
  - Nœud « TextureTransform » ou « TransformMatrix2D »

page 47

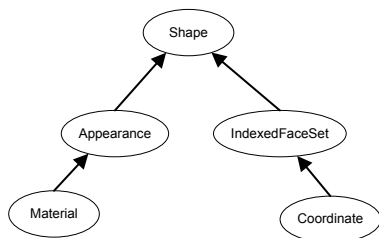
SI350

Descriptions de scènes multimédia





## Exemple: Partie d'arbre BIFS



page 48

SI350

Descriptions de scènes multimédia



## Positionnement BIFS

- **Scène = fenêtre rectangulaire + système de coordonnées global**
  - Largeur / Hauteur exprimées en
    - Pixel
    - Mètre (convention largeur scène = 2 mètres)
  - Repère (Ox, Oy) avec O au centre de la fenêtre
    - [Ox] horizontal, positif vers la droite
    - [Oy] vertical, positif vers le haut
- **forme géométrique =**
  - système de coordonnées local (propre à la forme)
    - Rectangle, Ellipse, Box ... = système centré
    - Texte = système centré en fonction de l'alignement

page 49

SI350

Descriptions de scènes multimédia



## Positionnement BIFS (2)

- **Nœuds de transformation explicite (2D/3D)**
  - Transformation du repère local par rapport au repère parent
  - 2 types de nœuds:
    - « Transform(2D/3D) »
      - exposedField SFVec2f center 0 0
      - exposedField SFFloat rotationAngle 0
      - exposedField SFVec2f scale 1 1
      - exposedField SFFloat scaleOrientation 0
      - exposedField SFVec2f translation 0 0
    - « TransformMatrix(2D/3D) »
      - exposedField SFFloat mxx 1
      - exposedField SFFloat mxy 0
      - exposedField SFFloat tx 0
      - exposedField SFFloat myx 0
      - exposedField SFFloat myy 1
      - exposedField SFFloat ty 0

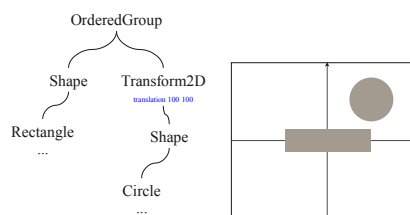
page 50

SI350

Descriptions de scènes multimédia



## Scène Simple



page 51

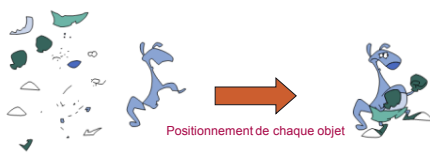
SI350

Descriptions de scènes multimédia



## Représentations graphiques BIFS avancées

- **Utilisations de primitives spéciales**
  - Polygones, poly-lignes, courbes de Bézier
  - Pour représenter des contours ouverts et fermés
    - Avec les mêmes propriétés de tracé
    - Utilisation de primitives de positionnement (différence avec Flash)



page 52

SI350

Descriptions de scènes multimédia



## Positionnement BIFS avancé

- **Positionnement implicite par contraintes**
  - Exemples:
    - le bord gauche de l'objet A est aligné avec le bord droit de l'objet B
      - Nœud « Form »
    - Les objets A,B,C sont disposés le long d'un chemin
      - Nœud « PathLayout »
    - Les objets A,B,C sont disposés de gauche à droite et de haut en bas dans une zone rectangulaire
      - Nœud « Layout »
- **Positionnement en profondeur**
  - Nœud « Group » = algorithme du peintre en 2D
  - Nœud « OrderedGroup » = ordre indiqué

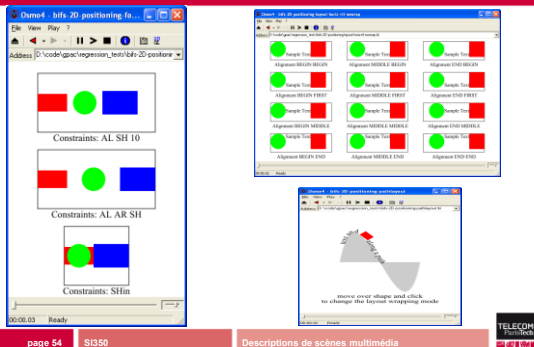
page 53

SI350

Descriptions de scènes multimédia



## Positionnement par contraintes



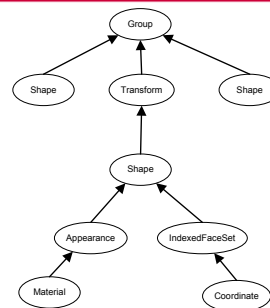
page 54

SI350

Descriptions de scènes multimédia



## Exemple: Partie d'arbre BIFS - suite



page 55

SI350

Descriptions de scènes multimédia



## Positionnement BIFS (4)

- **Nœuds de clipping simple**
  - Layer3D: fenêtre de visualisation 3D
  - Layer2D: fenêtre de visualisation 2D
- **Nœuds de visualisation avancés**
  - 3D: nœud « Viewpoint »
  - 2D: nœud « Viewport »

page 56

SI350

Descriptions de scènes multimédia



## Mise à jour de la scène BIFS

- **Possibilité de modifier la scène par commande**
  - Adressant un nœud par identifiant
  - Une propriété par son nom
  - Une index dans le cas de propriétés de type MF\*
- **Types de commandes**
  - Insertion d'une valeur
  - Remplacement d'une valeur
  - Destruction d'une valeur
  - Remplacement de la scène complète
- **Exemple**
  - REPLACE M emissiveColor BY 1 0 0
  - DELETE ROUTE R1
  - INSERT AT MonPolygone.point[0]-50 -50
- **Démonstration: Convertisseur Flash => BIFS**

page 57

SI350

Descriptions de scènes multimédia



## 2 types de flux BIFS

- **Flux BIFS-Command**
  - Toutes les commandes BIFS
  - Configuration =
    - taille de la scène et unité de mesure
    - Version de codage (v1 (sans proto), v2 (avec proto))
    - Nombres max d'identifiants (nœuds, route, proto)
- **Flux BIFS-Anim**
  - Equivalent à des commandes REPLACE
  - Mises à jour continues de données numériques
    - Nœuds et propriétés cibles indiquées dans la configuration du décodeur
  - Quantification / Prédiction / Codage Arithmétique
  - Contexte de codage sur plusieurs Access Units

page 58

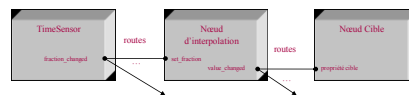
SI350

Descriptions de scènes multimédia



## Animation en BIFS

- **Utilisation de 3 nœuds + 2 routes**
  - **Timer = nœud « TimeSensor »**
    - exposedField SFTime cycleInterval 1
    - exposedField SFBool enabled TRUE
    - exposedField SFBool loop FALSE
    - exposedField SFTime startTime 0
    - exposedField SFTime stopTime 0
    - eventOut SFTime cycleTime 0
    - eventOut SFFloat fraction\_changed 0
    - eventOut SFBool inactive FALSE
    - eventOut SFTime time 0
  - Nœud d'interpolation en fonction du type de données
    - ScalarInterpolator, ColorInterpolator, CoordinateInterpolator ...
  - Nœud cible de l'animation



page 59

SI350

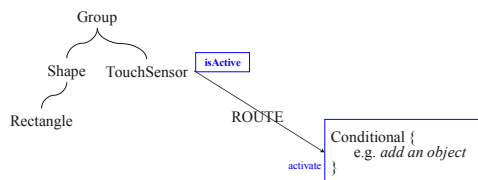
Descriptions de scènes multimédia



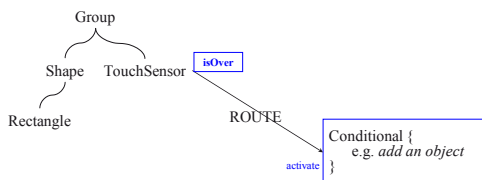
## Interactivité en BIFS (simple)

- **Utilisation de nœuds capteurs des événements de l'utilisateur**
  - TouchSensor, InputSensor, PlaneSensor, ProximitySensor ...
- **Routage des événements vers**
  - La cible
  - Un nœud « Conditional » pour exécuter une liste de mise à jour BIFS
- **Possibilité de faire du 'type-casting' d'évènements = nœud « Valuator »**

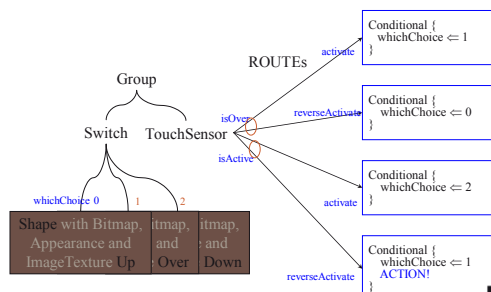
## Scène interactive avec bouton



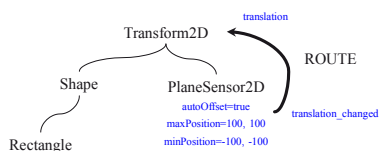
## Rollover



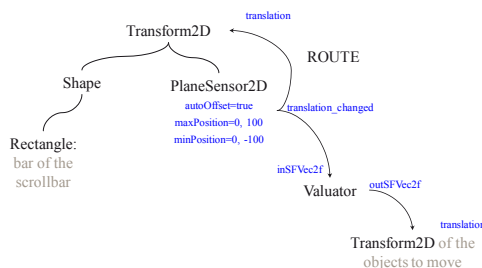
## Bouton 3-états



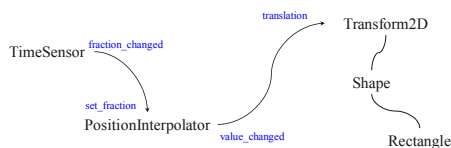
## Objet "draggable"



## Ascenseur



## Mouvement



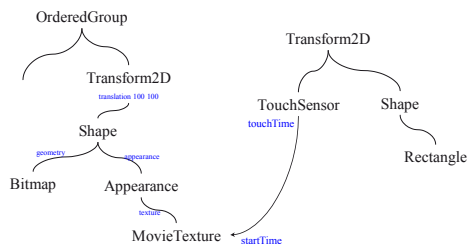
page 66

SI350

Descriptions de scènes multimédia



## Media Start/Stop



page 67

SI350

Descriptions de scènes multimédia



## Interactivité avancée BIFS

### ■ Utilisation du nœud « Script »

- Définition des événements en entrée à la création du nœud
  - eventIn SFCOLOR monEvenementCouleur
- Déclenchement d'une fonction ECMAScript à la réception d'un événement
  - fonction monEvenement(value, timestamp) { ... }
- Utilisation d'API classiques + d'API propres à VRML/BIFS
  - Création de nœuds/de valeurs ...

page 68

SI350

Descriptions de scènes multimédia



## Utilisation de médias

### ■ Utilisation des médias dans éléments temporels

- Nœud « MovieTexture » = vidéo
- Nœud « AudioSource » = son
- Nœud « ImageTexture » = image fixe
- Nœud « AnimationStream » = flux BIFS

### ■ Identification de la source

- Propriété url
  - Possibilité d'utiliser une URL: http://, rtsp:// ...
  - Possibilité d'utiliser un objet MPEG-4 (od://)

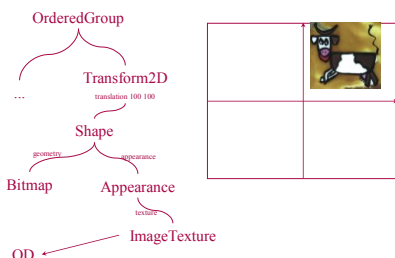
page 69

SI350

Descriptions de scènes multimédia



## Scène avec Image



page 70

SI350

Descriptions de scènes multimédia



## Contrôle des médias

### ■ Contrôle du temps d'affichage par les propriétés

- exposedField SFBool loop FALSE
- exposedField SFFloat speed 1
- exposedField SFTIME startTime 0
- exposedField SFTIME stopTime 0

### ■ Synchronisation définie dans les descripteurs d'objets

#### ■ Monitoring du temps média = nœud « MediaSensor »

- eventOut SFTIME mediaCurrentTime 0
- eventOut SFTIME streamObjectStartTime 0
- eventOut SFTIME mediaDuration 0
- eventOut SFBool isActive FALSE
- eventOut MFString info []

#### ■ Contrôle du temps média = nœud « MediaControl »

- exposedField SFTIME mediaStartTime -1
- exposedField SFTIME mediaStopTime 3.40282e+038

page 71

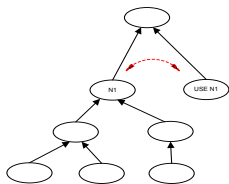
SI350

Descriptions de scènes multimédia



## Réutilisation d'éléments de scènes

- **Utilisation de la primitive USE**
  - Permet de faire des liens symboliques entre nœuds d'une même scène
- **Nœud « Inline »**
  - Réutilisation d'une autre scène
- **Nœud « Anchor »**
  - Lien vers une autre scène



## Définition de nœuds ad-hoc - PROTO

- **Mécanisme de PROTO**
  - Déclaration
    - Permet de définir un nouveau nœud
    - MonProto
  - De lui définir son interface
    - eventIn SFBool monChamp1 FALSE
    - eventOut SFColor monChamp2 0 0 0
    - ...
  - De définir le comportement de ce nœud
    - Sous la forme d'une sous scène
    - En connectant des propriétés de cette sous scène avec les propriétés de l'interface
      - » Ex: emissiveColor IS monChamp2
  - Utilisation – comme un nœud régulier
- **Permet de faire de concevoir des blocks de scènes paramétrables et indépendants**
  - Ex: éléments d'interface graphique

## Techniques de codages

- **Codage des nœuds**
  - Utilisation de tables de codage
  - pour identifier un type de nœud (Shape, MovieTexture ...)
  - La table change en fonction du type de propriété contenant le nœud (appearance, geometry, texture ...)
- **Codage des propriétés**
  - Utilisation de tables indiquant les propriétés possibles
  - par type de nœud
  - par contexte d'encodage (ROUTE, ...)
- **Codage des valeurs**
  - Valeurs réelles par défaut sur 32 bits
  - Possibilité de codage sans perte IEEE 754
  - Possibilité d'utilisation de quantification
  - Possibilité de codage Arithmétique sur les listes de valeurs quantifiées

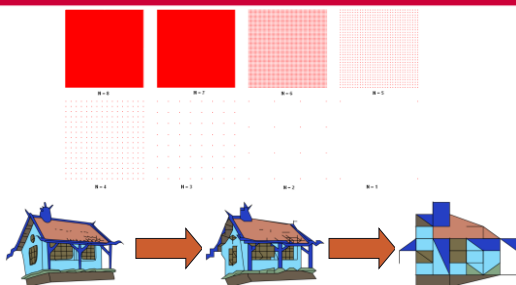
## Quantification BIFS

### Quantification linéaire

$$v_q = \text{int} \left( \frac{v - v_{\min}}{v_{\max} - v_{\min}} (2^N - 1) \right)$$

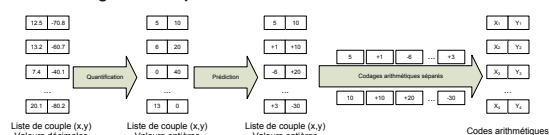
- **Par type de donnée**
  - N, Vmin, Vmax pour les couleurs
  - N, Vmin, Vmax pour les coordonnées 2D...
- **Par contexte de quantification**
  - Contexte signalé par un nœud « QuantizationParameter »

## Artefacts de codage



## Codage arithmétique BIFS

- **3 étapes de codage des listes**
  - Quantification
  - Prédiction
  - Codage arithmétique
- **Signalisation par le nœud « QuantizationParameter »**



## MPEG-4 BIFS 3D

- BIFS V1
  - VRML + Animation Faciale + Compression de Polyèdre
- BIFS V2
  - Animations Corporelles
- AFX = Animation Framework Extension
  - Subdivision de Surface, Nurbs, Compression ondelette, ...
  - Compression d'interpolation, animation de corps basée sur un squelette
  - Gestion améliorée de textures
- Lien: <http://www.mpeg-3dgc.com/>



page 78

SI350

Descriptions de scènes multimédia



## Intégration 2D / 3D

- Déclaration d'un environnement 3D
  - Nœuds « Layer3D », « CompositeTexture »
- Utilisation possible de sous-arbre 2D



page 79

SI350

Descriptions de scènes multimédia



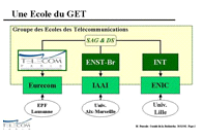
## Exemples de contenu BIFS 'pur'



Dessins animés 2D



Mondes virtuels 3D



Transparents à la PowerPoint



Cartographie

page 80

SI350

Descriptions de scènes multimédia



- Carousel
- Scalabilité

page 81

SI350

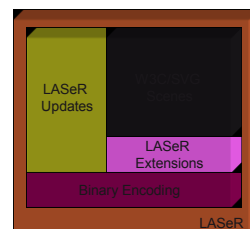
Descriptions de scènes multimédia



## Les langages SVG et LAsER

## La spécification LAsER

- Une sélection d'outils de la norme SVG
- des extensions spécifiques LAsER compatible SVG
- un mécanisme de mise à jour
- un format binaire pour la transmission et le stockage



page 82

SI350

Descriptions de scènes multimédia



page 83

SI350

Descriptions de scènes multimédia



## Scalable Vector Graphics

### ■ Langage XML

- Défini par le W3C
  - V1.0 – 2001
  - V1.1 – 2003
  - V1.2 – 2008?
- But:
  - Décrire des scènes graphiques vectorielles 2D

### ■ Caractéristiques

- Éléments visuels et sonores
- Organisation spatiale
- Gestion du temps
- Capacité d'animation
- Capacité d'interactivité

page 84

SI350

Descriptions de scènes multimédia



## SVG – Éléments visuels et sonores

### ■ Formes graphiques

- Formes primitives
  - <rect>, <ellipse>, <circle>, <line>
- Formes libres (ouvertes ou fermées)
  - <polygon>, <polyline>
  - Tracé complexe <path>: segments, courbes de Bézier cubique, quadratique, ...

### ■ Texte: <text>, <tspan>, <tref>

- Définition d'un format de police de caractère: <font>

### ■ Média

- Images fixes (JPG, PNG) : <image>
- Audio, vidéo (dans la version 1.2 en cours)
  - <audio>, <video>
- Graphiques animés (nouveau 1.2) : <animation>

page 85

SI350

Descriptions de scènes multimédia



## SVG – Propriétés visuelles

### ■ Remplissage des formes

- Uniforme ou Gradients (SVG Tiny 1.2)
- Avec transparence

### ■ Tracé des contours

- Uniforme ou gradient
- Joint et terminaison différentes
- Épaisseur et style de trait (pointillé ...)

page 86

SI350

Descriptions de scènes multimédia



## SVG – Organisation spatiale

### ■ Repère global (Ox,Oy) défini par l'élément <svg>

- Centré en haut à gauche, [Ox] orienté à droite, [Oy] orienté vers le bas
- Fenêtre de visualisation: « width », « height », « viewBox », « preserveAspectRatio »

### ■ Repère local

- Propre à chaque forme géométrique
  - Cercle, ellipse: centré
  - Rectangle: coin haut-gauche
  - Polygone, lignes, tracés: Point (0,0)
  - Texte: en fonction de l'alignement
- Possibilité de transformation matricielle
  - Attribut « transform »

### ■ Pas de positionnement implicite par contrainte (sauf <textPath>)

page 87

SI350

Descriptions de scènes multimédia



## SVG – Organisation temporelle

### ■ Hérité du modèle de temps SMIL

- Éléments temporels + Conteneurs temporels

### ■ Restrictions SVG (actuelles)

- 1 seul conteneur temporel: élément <svg>
- Éléments temporels:
  - Éléments d'animations
  - Éléments média

### ■ Contrôle du temps d'affichage:

- Attributs « begin », « end », « dur », « repeatDur » ...

### ■ Contrôle du temps média

- Impossible en SVG (possible en SMIL)
- Attributs « clipBegin », « clipEnd »

### ■ Possibilité de désynchroniser un élément temporel

- Attributs « syncBehavior », « syncTolerance »

page 88

SI350

Descriptions de scènes multimédia



## SVG – Animation

### ■ Réutilisation du modèle d'animation SMIL

- Capacités importantes et flexibles
- Imbrication d'animations possibles
- Intérfaçage avec CSS

### ■ Éléments

- <animate>, <animateColor>, <animateMotion>, <animateTransform>, <set>

### ■ Attributs

- « xlink:href », « attributeName »: identification cible
- « from », « to », « values »: valeurs d'interpolation
- « calcMode », « keyTimes », « keySplines »: type d'interpolation
- « additive », « accumulate »: comportement d'animations multiples

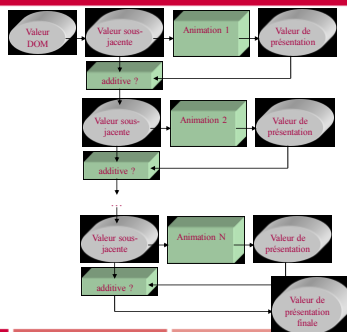
page 89

SI350

Descriptions de scènes multimédia



## SVG – Animation – Cascade



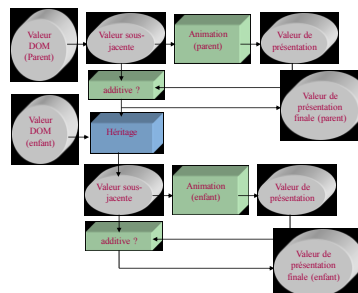
page 90

SI350

Descriptions de scènes multimédia



## SVG – Animation – Cascade et Héritage



page 91

SI350

Descriptions de scènes multimédia



## SVG – Interactivité

### ■ En cours de modification

- V1.1: utilisation de « onclick » ... à la HTML
- V1.2: utilisation du modèle:
  - DOM 3 Events + XML Events

### ■ Principes

- Évènement = nom + espace de nommage
- Listener:
  - Écoute un évènement
  - Déclenche un « handler »
- Handlers SVG = animation, lien ou script

page 92

SI350

Descriptions de scènes multimédia



## SVG – Interactivité et Script

### ■ Possibilité d'utiliser un langage de script

- Élément <script>
- Code ECMAScript, Java ...
- Utilisation d'interface
  - SVG 1.1: DOM
  - SVG 1.2 Tiny: MicroDOM

### ■ Interactivité limitée sans script

- Déclenchement et arrêt animation
- Suivi de lien

page 93

SI350

Descriptions de scènes multimédia



## SVG – Composition avancée

### ■ Uniquement en SVG 1.1 Full

- Clipping, Masking
- Effets Photoshop
- Opacité de groupe
- Utilisation de feuilles de style: CSS ou XSLT
- Textures synthétiques: <pattern>
- Symboles
- Marqueurs en bout de ligne

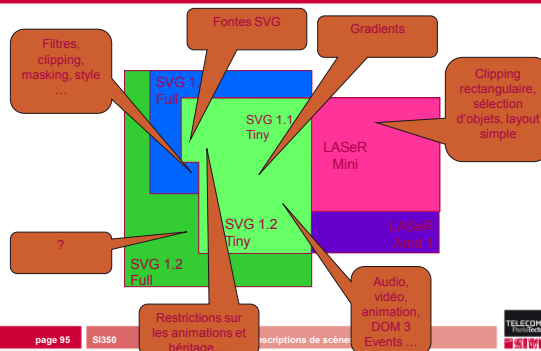
page 94

SI350

Descriptions de scènes multimédia



## Relations entre LAsER et SVG



page 95

SI350

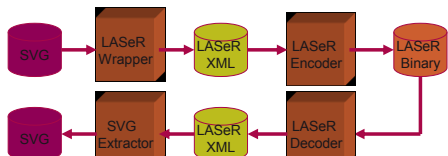
Descriptions de scènes multimédia





## Relation entre LASeR et SVG (suite)

- Les scènes LASeR sont basées sur SVG Tiny 1.1
  - Les scènes LASeR peuvent être créées en SVG
  - Le contenu LASeR encodé à partir de SVG peut être décodé en SVG
- Les extensions LASeR ne peuvent pas être exprimées en SVG
  - Le contenu utilisant ces extensions ne peut être décodé en SVG
  - Mais à tout moment, le résultat d'exécution des extensions peut-être représenté en SVG
- LASeR n'est pas un format XML mais
  - Un langage XML existe pour représenter un flux LASeR (conformité)
  - Le contenu SVG peut être encapsulé dans du LASeR XML puis encodé



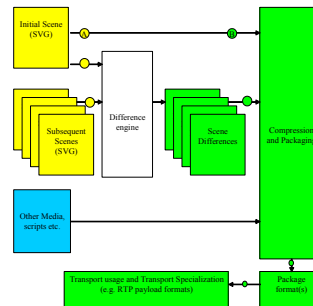
page 96

SI350

Descriptions de scènes multimédia



## Relation entre LASeR et SVG (encodeur)



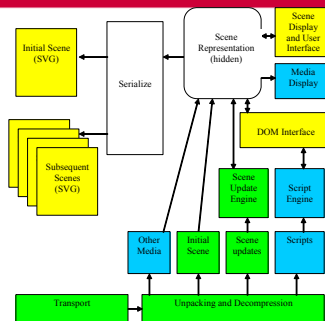
page 97

SI350

Descriptions de scènes multimédia



## Relation entre LASeR et SVG (client)



page 98

SI350

Descriptions de scènes multimédia



## LASeR – Mise à jour de scène

- Similaire à BIFS
  - Insertion, suppression, remplacement
- Nouveautés
  - Opérations (addition, soustraction)
  - Concaténations
  - Gestion d'un cache de scène
  - Déclenchement d'événements
- Ajout par rapport à SVG
  - Utilisation de commandes en alternative à un langage de script

page 99

SI350

Descriptions de scènes multimédia



## LASeR – Compression

- Codage compatible MPEG-7 BiM
- Utilisation du codage de structure BiM
  - Extensible aux évolutions de LASeR et SVG
  - Possibilité de mélanger du XML (XHTML ...)
- Utilisation de codage de données spécifiques
  - Quantification des coordonnées, longueur ...
  - Codage Exp-Golomb sur les listes de points
  - Utilisation d'une palette (dynamique) de couleur

page 100

SI350

Descriptions de scènes multimédia



## Comparaisons – Fonctionnalités

- SVG ~ BIFS ~ LASeR
- Principales différences
  - Intégration 3D dans BIFS
  - Utilisation indispensable de Script en SVG pour une interactivité riche
  - Mode de création
    - BIFS = primitives simples, mais nombreuses
    - SVG = primitives complexes, mais peu nombreuses
  - Modèles événementiels
  - Moins de tables de codage en LASeR par rapport à BIFS
  - Pas encore de PROTO en SVG (cf. sXBL)

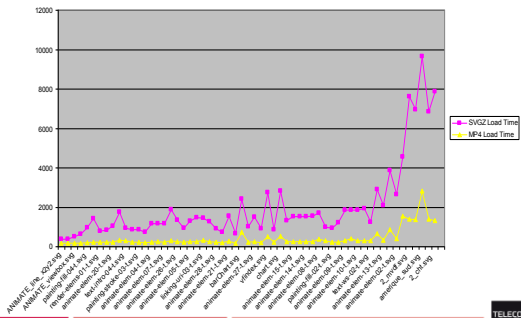
page 101

SI350

Descriptions de scènes multimédia



### Comparaisons – Temps de chargement



### Comparaisons – Compression

