

# Le langage SMIL au service des Sciences Humaines et Sociales

Vincent Kober

INRIA Rhône-Alpes, Projet WAM - Maison des Sciences de l'Homme-Alpes, Projet Portail du Réseau des Maisons des Sciences de l'Homme.

[vincent.kober@inrialpes.fr](mailto:vincent.kober@inrialpes.fr)

date : 14 octobre 2003

Daniel Weck

INRIA Rhône-Alpes, Projet WAM - LimSee2

[daniel.weck@inrialpes.fr](mailto:daniel.weck@inrialpes.fr)

Cécile Roisin

INRIA Rhône-Alpes, Projet WAM - Université Pierre Mendès France Grenoble

[cecile.roisin@inrialpes.fr](mailto:cecile.roisin@inrialpes.fr)

## Résumé :

*Cet article présente le résultat d'une collaboration entre une équipe de recherche de l'INRIA qui travaille notamment sur la présentation de documents multimédia synchronisés et une équipe d'historiens ayant des besoins dans ce domaine. Sans être une étude exhaustive d'un langage, nous montrons à travers l'étude des besoins exprimés par les chercheurs historiens, comment l'utilisation des technologies et des outils liés au langage SMIL (Synchronized Multimedia Integration Language) répond aux besoins de mise en oeuvre de ce type de présentation. Cet article décrit également les principaux outils permettant d'écrire et de jouer des documents au format SMIL en insistant sur l'outil auteur LimSee développé au sein du projet WAM.*

## Mots clefs :

SMIL, Présentation multimédia, Edition de documents, LimSee, Web, PREALP

## 1 Introduction

PREALP (Peintures murales des Regions ALPines - <http://www.msh-alpes.prd.fr/prealp/>) programme européen interdisciplinaire, s'attache à l'étude exhaustive des décors monumentaux médiévaux de l'ensemble de l'arc alpin, du Léman à la Méditerranée et de la Savoie au Frioul. Depuis décembre 1999, la base de données de PREALP est installée à Grenoble, à la Maison des Sciences de l'Homme-Alpes. PREALP a des besoins de représentation et de synchronisation de ses données multimédia (principalement images et textes pour l'instant) dans des documents divers destinés à des chercheurs ou jeunes doctorants en histoire et permettant de visualiser de façon très détaillée, et de naviguer à travers les décors qu'ils étudient.

Le projet WAM (Web Adaptation et Multimédia - <http://wam.inrialpes.fr/>) est un projet de l'INRIA Rhône-Alpes qui aborde les problèmes posés par les évolutions du Web. Il se focalise sur la transformation de documents considérée comme un type de traitement générique des documents du Web. Il considère plus particulièrement les documents multimédia qui intègrent étroitement des média statiques (texte, images, équations) et dynamiques (vidéo, son, animations). Il applique ses résultats en particulier à l'adaptation des documents et à l'indépendance des appareils d'accès au Web. WAM développe entre autres un outil auteur de documents multimédia au format SMIL

Ces deux projets ont établi une collaboration depuis juillet 2002. L'objectif est de mettre en place des outils permettant de représenter l'ensemble des monuments étudiés dans PREALP par des documents multimédia mais aussi de pouvoir lier ceux-ci entre eux, fournissant ainsi aux chercheurs des éléments de documentation indispensables.

Dans cet article, la présentation de cette collaboration va servir de base à la description du langage choisi pour la mise en oeuvre de ce projet ainsi qu'à la description des principaux outils permettant l'édition et la visualisation de ces documents multimédia. L'expérience du projet WAM sur les documents structurés, notamment multimédia, et leur utilisation, l'implication de certains de ses membres dans la création du langage SMIL ont fait que celui-ci est très vite apparu comme le mieux adapté à répondre aux différents besoins exprimés.

Cet article est structuré de la manière suivante :

Dans une première partie, sans vouloir faire une description exhaustive des possibilités du langage SMIL, nous présenterons les besoins principaux des historiens quant à la création ou la lecture de leurs documents de présentation multimédia ainsi que les manières possibles d'y répondre grâce à ce langage. Dans une seconde partie, nous présenterons les principaux outils existant dans ce domaine. L'analyse détaillée des besoins de visualisation permettra de poser la problématique de l'utilisation du langage standard et de ses dérivés (HTML + time : Timed Interactive multimedia Extensions for HTML notamment). Ensuite, les outils auteur représentatifs seront décrits et illustreront les différents moyens de créer des documents en SMIL. Finalement, nous détaillerons de façon plus précise LimSee2, l'éditeur de documents SMIL développé au sein du projet WAM. LimSee2 se place dans la catégorie des environnements auteurs multi-vues. La présentation de ses principales fonctionnalités permettra de pointer les difficultés particulières rencontrées lors de l'édition de documents.

Comme nous le verrons, l'éventail relativement limité des outils (tant pour la présentation que pour la création), l'existence des deux versions de SMIL (SMIL1.0 et SMIL2.0), ainsi que l'opportunité de n'utiliser qu'une partie du langage (modularisation et profilage) rendent encore plus difficile la création de documents permettant d'être lus sur l'ensemble des systèmes existants.

Nous concluons sur l'hétérogénéité des plates-formes de visualisation disponibles et donc sur la nécessité, pour l'auteur de documents multimédia, de prendre en compte ces contraintes de présentation dans la démarche de production.

## 2 Expression des besoins

Les besoins rencontrés pour la production de documents multimédia peuvent se classer en deux catégories : les besoins immédiats d'expression liés à la présentation des média (disposition spatiale des média, organisation temporelle) et les besoins à plus long terme liés à l'évolution de ce type de documents (réutilisation, adaptation à de nouveaux média (PDA, pages html...)).

Les cas concrets rencontrés au cours de ce projet vont permettre d'illustrer les besoins d'une application multimédia et de justifier le choix de SMIL comme étant un langage adapté à ce type d'application.

### 2.1 Besoins d'expression des présentations multimédia

#### – Structuration

La structuration d'un document multimédia est importante car celui-ci n'est pas uniquement une suite de slides avec quelques animations. Il est en fait composé de différents média (textes, sons, images, vidéo, animation...) présentés dans des régions, chacun de ces média pouvant, à un moment ou à un autre avoir une interaction plus ou moins grande avec tout autre partie du document. Cette interaction se concrétise par des id's et des références à ces id's.

Cela implique une nécessaire structuration du document permettant d'éviter les redondances de code.

Le document SMIL est donc composé de plusieurs parties, l'entête comprenant notamment la description des régions et des animations, le corps du document regroupant le scénario proprement dit.

La richesse d'un document SMIL dépend aussi de la capacité du langage, comme la plupart des langages dérivés d'XML d'inclure la notion d'espaces de nom (namespaces) permettant d'inclure, dans un document SMIL d'autres langages (SVG, formules mathématiques, documents html). Cette richesse étant limitée, comme nous le verrons plus loin, dans la section sur les outils de visualisation, par les implémentations de ces derniers.

Il est possible d'illustrer ce besoin par l'écriture du code concernant une animation de fondu enchaîné entre images qui peut être utilisée à plusieurs endroits dans le document (Figure 1).

- L'animation ( `<transition.../>` ) est tout d'abord définie dans l'entête ( `<head>...</head>` ) en lui spécifiant un identifiant ( `'id=fade_3'` )

- Cet identifiant peut ensuite être utilisé dans la suite du document comme une valeur d'un attribut de transition ('transIn=fade\_3') d'un élément dans le document

```

<!--smil xmlns="http://www.w3.org/2001/SMIL20/Language"
xmlns:rn="http://features.real.com/2001/SMIL20/Extensions"
xmlns:cv="http://features.real.com/systemComponent" -->
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <layout>
      <root-layout id="presente" width="960" height="840"/>
      <region id="logo" left="10" top="20" width="163" height="89"/>
      .
      .
    </layout>
    <!-- animation définissant une transition d'une durée
de 2 seconde par fondu -->
    <transition id="fade_3" dur="2s" type="fade"/> 1 - définition de l'animation
  </head>
  <body>
    <par>
      <excl id="excl01" dur="indefinite">
        <!-- insertion de l'image en utilisant la transition fade_3 définie
dans l'entête -->
         2 - réutilisation de l'animation
      .
      .
    </par>
  </body>
</smil>

```

Figure 1 – Exemple de structuration d'un document multimédia

## – Placement Spatial

Le placement spatial se caractérise par l'identification de zones à l'écran dans lesquelles les média vont se placer.

Dans le projet de présentation multimédia de PREALP, il est important de pouvoir identifier les différentes zones (zone principale, zone de navigation, zone d'information (voir Figure 2) et de s'y conformer pour garder une bonne lisibilité du document.

La possibilité de réutiliser ces zones pour d'autres média, des les placer relativement à leur conteneur, de prendre en compte la notion de profondeur grâce à l'attribut `z-index` permettent au langage SMIL de répondre à quasiment l'ensemble des besoins.

Cependant, des spécificités comme le placement d'une région par rapport à une autre de même niveau (2 régions soeurs ayant le même conteneur) restent plus complexes à réaliser.

## – Navigation

A la navigation interne à un monument et aux différents détails qui le composent s'ajoutent des besoins de navigation entre documents sur des thématiques particulières (même sujet iconographique (apparition de la Vierge de Miséricorde dans des chapelles différentes par exemple), datation de peintures, corpus des inscriptions (avec la nécessité de pouvoir replacer précisément les textes dans leur contexte)...).

Le langage doit ainsi prendre en compte non seulement la notion hypertexte liée au lien proprement dit, mais aussi la donnée temporelle, tant à la source (le lien peut n'exister que pendant un temps défini dans une vidéo par exemple), qu'à la destination (la cible du lien est un point précis temporel du document de destination avec l'ensemble des autres média synchronisés sur ce point).

SMIL2.0 permet aussi de prendre en compte la notion d'évènement, notamment pour les attributs `begin` et `end`.

Dans l'exemple de la Figure 2, le passage de la souris sur la région de l'image en bas à droite, correspondant à la partie de chapelle que l'on souhaite visiter, permet l'affichage d'une miniature de la portion de mur ainsi que du nom de celle-ci accompagné d'une colorisation du plan. Le code correspondant est détaillé Figure 3.

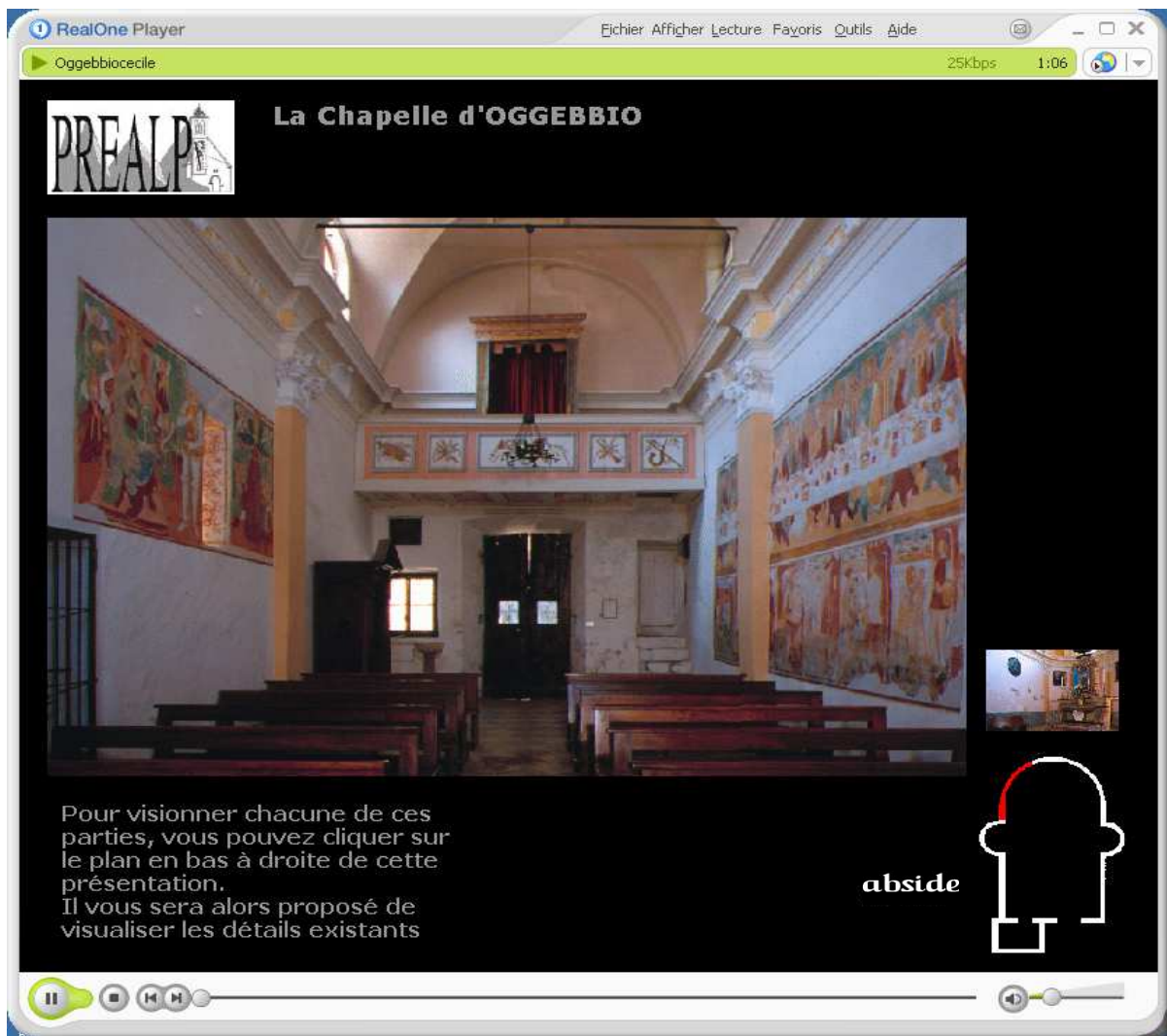


photo : AD Studio Silvano Ferraris (21 juillet 1999)

Figure 2 – Gestion des évènements pour la navigation dans la chapelle

Dans cet exemple, l'image ayant pour id 'bouton2' passe d'un fond blanc à un fond rouge grâce à un élément fils : l'élément `set` qui est un des éléments du module BasicAnimation de SMIL2.0.

Celui-ci est composé :

- 2 attributs permettant de déterminer de début et la fin de l'animation :
  - `begin` et `end` ayant respectivement pour valeur : `'bouton2.inBoundsEvent'` pour le focus de la souris sur l'image et `'bouton2.outOfBoundsEvent'` pour la perte de focus.
- 2 attributs permettant de définir le type d'animation :
  - `attributeName` définissant le nom de l'attribut à animer et `to` permettant de définir sa nouvelle couleur.
  - 1 attribut déclarant l'élément sur lequel va s'effectuer l'animation :
    - `targetElement` qui a pour valeur le nom de l'image.

Les éléments `text` et `img` qui suivent apparaissent eux aussi en fonction des évènements affectés à leurs attributs `begin` et `end`.

```

<!-- bouton de navigation permettant de détailler le mur sud -->

  <set targetElement="bouton2" attributeName="backgroundColor" to="red"
    begin="bouton2.inBoundsEvent" end="bouton2.outOfBoundsEvent" />
  <set targetElement="bouton2" attributeName="backgroundColor" to="white"
    begin="bouton2.outOfBoundsEvent" end="bouton2.inBoundsEvent" />
</img>
<!-- texte s'affichant au passage du curseur de la souris sur le bouton -->
<text src="textes/legende_12.txt" region="textebasdroitgauche"
  begin="bouton2.inBoundsEvent" end="bouton2.outOfBoundsEvent">
  <param name="charset" value="iso-8859-1"/>
  <param name="fontFace" value="Verdana"/>
  <param name="fontWeight" value="bold"/>
  <param name="fontSize" value="3"/>
  <param name="fontColor" value="#999999"/>
  <param name="backgroundColor" value="black"/>
  <param name="hAlign" value="right"/>
  <param name="vAlign" value="center"/>
</text>
<!-- image s'affichant au passage du curseur de la souris sur le bouton -->


```

Figure 3 – Détail du code : gestion des évènements

### – Placement temporel et synchronisation

La gestion des différents média temporels (animations, vidéos, sons) nécessite une réflexion approfondie, particulièrement sur la granularité de leur synchronisation.

En effet, s'il est relativement simple de faire apparaître un média dans un document, il est beaucoup plus complexe de le synchroniser finement avec les autres éléments du document.

On distinguera 3 types de synchronisation :

- La synchronisation « gros grain » gérée par les éléments <par> et <seq> et <excl> permettant de définir le placement dans le temps des différents média.
- La synchronisation liée à l'utilisation des attributs de synchronisation begin ou end. Ceux-ci permettent d'exprimer le début d'un élément de plusieurs façons (temps d'horloge ou temps où se produit un évènement).
- Ce type de synchronisation peut encore s'affiner par l'utilisation par exemple :
  - de l'attribut clipBegin qui permet de commencer un média, non pas au début, mais à un instant décalé par rapport à son début.
  - de l'élément area permettant de séparer un objet en sous-parties temporelles, en utilisant des attributs comme begin et end. Les valeurs des attributs begin et end étant relatives au début de l'objet média conteneur. L'élément area permet de faire d'une sous-partie de l'objet média la destination d'un lien, en utilisant ces attributs temporels et l'attribut id.

## 2.2 Besoins de gestion des présentations multimédia

Ces besoins regroupent principalement les problématiques de réutilisation des documents à long terme, que ce soit en terme de réutilisation du document dans le temps, ou d'adaptation à d'autres outils de lecture.

SMIL héritant de l'ensemble des technologies XML, il peut répondre parfaitement à ces exigences.

### – Réutilisation du document

Ce problème est important. En effet, il est fréquent de voir des documents réalisés par une application version N qui ne soient plus lisibles par la même application version N+2 voire illisibles parce que l'application n'existe plus.

La séparation, dans SMIL, des média et de la structure du document permet de faire évoluer les média sans avoir à remettre en cause l'ensemble du document.

Dans le contexte de PREALP, il était important de prendre en compte l'exploitation de média déjà existants et la possibilité d'utiliser ceux-ci pour d'autres applications :

- utilisation sur le site de PREALP par exemple, à l'intérieur d'une page HTML

- prévision d'utilisation pour la réalisation d'un CDRom de présentation servant de vitrine au projet.

Il est aussi important de noter que SMIL permet de faire référence plusieurs fois à la même image dans le même document en la présentant de multiples façons (taille de la région d'accueil de l'image, type de remplissage de la région (attribut fill)...)

D'autre part, SMIL, comme tout document XML, est décrit dans un format textuel ce qui le rend beaucoup plus facile à traiter.

#### – Adaptation - transformation

La commande initiale exprimée dans le cadre du projet PREALP a été la réalisation d'un document multimédia consultable sur PC (soit en direct sur une machine, soit par une consultation web). Ce besoin est cependant susceptible d'évoluer dans le temps, soit pour répondre à une demande de consultation sur un autre support (PDA par exemple), soit pour une réutilisation partielle de la présentation sur un site web.

SMIL offre, là aussi, des solutions par :

- **des fonctions syntaxiques** (switch par exemple) permettant d'adapter le document en fonction, par exemple, de la taille de l'écran du support de lecture.

On peut voir, Figure 4, le détail du code permettant au lecteur SMIL de choisir la taille de l'image qu'il va lire en fonction de la taille de l'écran du système sur lequel il se trouve. L'auteur du document peut donc ainsi prévoir l'adaptation de son document à plusieurs supports.

```
<switch>
  
  
  
  
</switch>
```

Figure 4 – Utilisation de SMIL pour l'adaptation d'une image à la taille de l'écran

- **le format XML** permettant d'utiliser l'ensemble des outils de transformation et de traitement liés à XML (XSLT, Xquery, Xpath...).

Des travaux de recherche sur une adaptation en temps réel du document aux capacités du support final, sans que l'auteur ait à le prévoir dans la phase d'édition, sont en cours. Ils sont consultables sur le site du projet WAM. (<http://wam.inrialpes.fr/publications/index.fr.html#Adapted>)

La transformation d'un document SMIL peut, quant à elle, être envisagée quant, à partir des mêmes sources, on souhaite effectuer une présentation multimédia, mais aussi des présentations de nature plus statique (pages html par exemple). Elle peut aussi servir à transformer un document SMIL en un document HTML + Time (ou un autre profil de SMIL) afin de pouvoir offrir une plus grande plage de possibilités de lecture.

Ces transformations pourront se mettre en place dans le cadre de modèles de structure bien déterminés, sur des objectifs de transformation ciblés (exemple : remplacement d'une vidéo de fichier SMIL par une image dans la page HTML) et n'auront d'intérêt que sur un nombre important de documents de même type, permettant ainsi un gain de productivité, ou sur des documents fréquemment modifiés afin d'en faciliter le processus de mise à jour.

Un exemple de transformation de document XML en document SMIL ou HTML peut être consulté à cette adresse :

<http://wam.inrialpes.fr/people/kober/DemoXslt/>

Le document explicatif concernant la procédure est le suivant :

<http://wam.inrialpes.fr/people/kober/DemoXslt/README.TXT>

et l'ensemble des fichiers nécessaires peuvent aussi être téléchargés :

<http://wam.inrialpes.fr/people/kober/DemoXslt/DemoXslt.zip>

Ces transformations peuvent bien entendu être réalisées dynamiquement (transformation via des pages JSP ou PHP au niveau du serveur web par exemple), ou statiquement.

#### – Création de documents multilingue

Le programme PREALP étant un programme européen, des solutions permettant de gérer de façon efficace l'aspect multilingues des présentations sont à envisager. Là aussi, le langage SMIL va pouvoir facilement adapter le document à ce type de besoins :

- d'une part par l'utilisation d'un attribut `switch` capable de définir le langage à fournir au navigateur en tenant compte de la configuration de ce dernier (Figure 5 : adaptation au langage par défaut).
- d'autre part par une mise à jour facilitée du fait justement de ce format textuel : il est facilement envisageable de prévoir par exemple une interface web susceptible de pouvoir faire modifier ces fichiers par un tiers étranger (facilitant ainsi le caractère collaboratif d'une application).

```

- <switch>
  <audio id="sound" src="../media/audioVersions/russian.mp3" begin="1" dur="30" systemLanguage="ru" />
  <audio id="sound" src="../media/audioVersions/japanese.mp3" begin="1" dur="30" systemLanguage="jp" />
  <audio id="sound" src="../media/audioVersions/english.mp3" begin="1" dur="30" />
</switch>
- <switch>
  <text src="Privyet. Menya zovout Dominique." region="lyrics1" dur="30" systemLanguage="ru" />
  <text src="Konnichi-wa! Boku wa Dominiku desu." region="lyrics1" dur="30" systemLanguage="jp" />
  <text src="Hello, my name is Dominique." region="lyrics1" dur="30" />
</switch>
- <switch>
  <text src="Ya zhivou v Kanade." region="lyrics2" dur="30" systemLanguage="ru" />
  <text src="Kanada ni sunde-imasu." region="lyrics2" dur="30" systemLanguage="jp" />
  <text src="I live in Canada." region="lyrics2" dur="30" />
</switch>

```

Figure 5 – adaptation du document SMIL au langage du navigateur

## – Synthèse

A travers l'exemple d'une présentation multimédia demandée par une équipe d'historiens, nous avons identifiés les points forts d'une approche s'appuyant sur la technologie SMIL/XML : modularité, pouvoir d'expression, réutilisation, adaptation.

Il reste maintenant à étudier la mise en oeuvre d'un tel projet, c'est à dire les conditions de création et de diffusion de ces documents et l'étude des outils à utiliser pour leur édition et leur visualisation.

## 3 Outils et expériences

Nous allons distinguer dans cette section les outils de visualisation et les outils auteur de documents SMIL. Il ne s'agit en aucun cas d'un état de l'art exhaustif des solutions existantes. Nous nous sommes focalisés sur les solutions les plus diffusées ou les plus à même de répondre à nos besoins.

Avant de détailler les différents outils, il m'a paru important de repréciser que le langage SMIL était organisé autour de modules et de profils et que, même s'il est un standard, les implémentations d'outils de visualisation permettant de lire ce type de fichiers sont peu nombreuses et très souvent limitées à une partie seulement de la recommandation du W3C (World Wide Web Consortium : <http://www.w3.org/>).

Il faut donc tenir compte de cette impératif, dès la phase de conception du document final, afin d'adapter la représentation des média aux différents outils de visualisation que l'on va utiliser.

### 3.1 Le standard SMIL et son organisation

La recommandation du W3C sur SMIL1.0 date du 15 juin 1998 (<http://www.w3.org/TR/REC-smil/>). Elle définit le langage SMIL comme un langage permettant d'intégrer des objets multimédia indépendant dans une présentation multimédia synchronisée.

Elle a été suivie le 7 août 2001 de la recommandation pour le langage SMIL2.0 (<http://www.w3.org/TR/smil20/>).

Comme le précise le W3C, la stratégie prise en compte pour intégrer des fonctionnalités respectives avec d'autres langages basés sur XML est fondée sur les concepts de modularisation et de profilage.

La modularisation est une approche selon laquelle une fonctionnalité de balisage est spécifiée comme un ensemble de modules qui contiennent des éléments, des attributs et des valeurs d'attributs sémantiquement liés. Le profilage est la création d'un langage basé sur XML par la combinaison de ces modules, dans le but de fournir la fonctionnalité requise par une application particulière.

Généralement, il y a un profil de langage principal qui inclut pratiquement tous les modules associés à un même espace de nommage. Par exemple, le profil de langage SMIL 2.0 utilise la plupart des modules SMIL 2.0.

D'autres profils de langage peuvent être spécifiés comme sous-ensemble du profil SMIL2.0, ou comme incorporant

un mélange de modules associés à différents espaces de nommage. SMIL 2.0 Basic est un exemple du premier, XHTML+SMIL du dernier.

## 3.2 Les outils de visualisation

Nous avons choisi de citer, dans un premier temps, les 3 outils les plus diffusés (Real, Internet Explorer et GriNS). Dans un deuxième temps, nous présentons des outils plus expérimentaux mais dont le développement et la diffusion nous paraissent prometteurs. Nous terminons par le point le plus délicat des présentations multimédia en SMIL, le rendu du texte, qui dépend très fortement des implémentations choisies par les différents outils de visualisation.

### 3.2.1 Les classiques

- **Real** (<http://www.real.com/realone/index.html>), avec **Real Player (compatible SMIL1.0)** et **Real One (compatible SMIL2.0)** semble s'imposer comme le système de présentation le plus universel puisqu'il est possible de l'utiliser aussi bien sous Linux (Preview Release) que sous Windows ou Mac OS X.

On peut préciser aussi l'importance du travail de Real sur les possibilités de streaming des documents multimédia, grâce à des attributs liés à l'utilisation d'un espace de nom spécifique à Real. Cela permet d'améliorer nettement les performances de lecture de ces documents sur le web.

Il faut cependant noter que toute utilisation d'un namespace particulier rendra le document SMIL moins lisible pour des outils de visualisation qui n'auront pas la possibilité de décoder ce namespace.

Il faut malheureusement insister sur les différences d'implémentation de la version Linux du système de présentation RealOne qui ne permet pas à l'heure actuel un rendu correct de textes incluant des éléments param alors que la version sous Windows le fait correctement.

- **Internet Explorer** (<http://www.microsoft.com/windows/ie/default.asp>) permet de lire des documents html+time. Ce format est une implémentation spécifique à Microsoft du profil xhtml+smil. Celui-ci est décrit dans une note du W3C (<http://www.w3.org/TR/2001/WD-XHTMLplusSMIL-20010807/>)

- **GriNS** (<http://www.oratrix.com/>) quant à lui, est un outil payant et permet la lecture de documents au format SMIL2.0.

### 3.2.2 et les autres...

- **Xsmiles** (<http://www.x-smiles.org/>) est un système de présentation intéressant, puisqu'il est open-source. Celui-ci a cependant pour défaut de ne pas implémenter complètement la norme SMIL2.0, même s'il propose des fonctionnalités intéressantes comme l'intégration de documents HTML formatés comme média source de l'élément texte ou l'affichage de SVG animé à l'intérieur d'une page SMIL. Il est en outre un navigateur web capable de lire les formulaires réalisés grâce au standard Xform's.

- **PocketSMIL** (<http://wam.inrialpes.fr/software/pocketsmil/>) s'adresse aux utilisateurs d'assistants personnel PocketPC et implémente le profil SMIL2.0 Basic

- **Helix community** (<https://www.helixcommunity.org/>) a été créé suite à la décision de Real d'ouvrir le code de ses outils de visualisation et de production au monde open-source. On peut espérer que cela permette à terme la création d'un système de présentation SMIL2.0, favorisant ainsi la reconnaissance de SMIL comme le standard des présentations multimédia sur le web.

- **Une liste d'outils de visualisation** (QuickTime (SMIL1.0), SMIL Player ( SMIL Basic Profile)...) peut aussi être trouvée sur la page du W3C (<http://www.w3.org/AudioVideo/>) ainsi qu'un certain nombre d'exemples sur la création de documents au format SMIL.

### 3.2.3 Le problème particulier du rendu des textes par les outils

Comme nous l'avons précisé ci-dessus, le point le plus délicat dans la création de fichiers au format SMIL est le formatage du texte. En effet, à part xhtml+smil où l'outil de lecture est de toute façon Internet Explorer, il est très



difficile de concevoir un document multimédia présentant du texte formaté sans tenir compte de l'outil qui jouera ce document.

Real a contourné ce problème en proposant l'utilisation de fichiers au format propriétaire RealText (.rt). Cela permet un rendu très correct du texte mais celui-ci ne pourra être lu que dans des outils supportant cette spécification.

Dans la Figure 6 nous pouvons voir un appel à un fichier texte (au format RealText) détaillé dans la Figure 7

```
<text src="textes/presente_04.rt" region="textehaut" begin="0s" dur="300s"/>
```

Figure 6 – Appel au fichier RealText dans un élément texte du fichier SMIL

```
<window bgcolor="black" width="300" height="599">
<font charset="iso-8859-1" face="helvetica" size="5" color="#999999">
<b>
La fresque du mur Sud
</b>
</font>
</window>
```

Figure 7 – Le fichier RealText correspondant (presente\_04.rt)

Une autre solution pour représenter le texte est de faire appel à un fichier textuel source (.txt) dans un élément SMIL `<text>` et de le formater en lui associant des attributs `<param>` (voir Figure 8)

```
<text src="textes/legende_12.txt" region="textebasdroidgauche">
<param name="charset" value="iso-8859-1"/>
<param name="fontFace" value="Verdana"/>
<param name="fontWeight" value="bold"/>
<param name="fontSize" value="3"/>
<param name="fontColor" value="#999999"/>
<param name="backgroundColor" value="black"/>
<param name="hAlign" value="right"/>
<param name="vAlign" value="center"/>
</text>
```

Figure 8 – Appel à un fichier source au format txt et formatage

Dans ce cas, on aura un texte formaté lisible avec RealOne (version 2) mais qui ne fournira qu'un texte standard sur fond blanc dans les versions antérieures ou ne sera pas lisible sur d'autres outils de visualisation (GriNS ou XSmiles par exemple).

Deux autres solutions existent pour insérer du texte dans un document SMIL :

`<text src='texte/foo.html' region='bar'>` : permet d'inclure un fichier html dans une présentation mais cette implémentation n'est acceptée que par Xsmiles. Elle est cependant très pratique pour des portions de texte structurées mais beaucoup plus lourde à mettre en oeuvre pour de simples chaînes textuelles.

`<text src='data:text/plain, ceci est un texte d'essai' region='bar'>` : permet d'écrire le texte directement dans le document SMIL mais est cependant limité aux caractères ASCII.

### 3.2.4 Le problème particulier du rendu des vidéos

Un des problèmes posés par l'hétérogénéité des outils est leur faculté à jouer les différents formats de vidéo. En effet, si RealOne peut sans problème lire des vidéos au format RealMédia (.rm) ou mpeg (selon l'encodage), il n'en va pas de même pour Xsmiles ou LimSee2. Ceux-ci s'appuyant sur JMF (Java Media Framework), ils peuvent lire des vidéos au format .mov ou mpeg mais la possibilité de jouer des fichiers .rm repose sur l'installation d'un plugin supplémentaire sur le client.

### 3.3 Les outils auteur

Comme nous l'avons vu, l'hétérogénéité des solutions de visualisation entraîne la nécessité pour l'éditeur, soit de se spécialiser dans une partie du langage, soit de proposer plusieurs formats de fichiers finaux.

On peut classer les outils d'édition en 4 catégories fonctionnelles (même si ces fonctions peuvent se retrouver dans plusieurs éditeurs) :

#### 3.3.1 Outils utilisant SMIL comme un format d'export

C'est le cas par exemple de certains outils comme Real Slide Show, les nouvelles versions de PowerPoint ou LimSee2 qui permet de lire des fichiers PowerPoint pour les transformer en documents SMIL.

#### 3.3.2 Outils d'édition basés sur le code source

- Emacs ou Xemacs (ou leurs homologues...) restent encore sûrement les outils les plus utilisés pour l'édition de documents SMIL. Les colorations syntaxiques permettant une gestion aisée des documents XML.

- Quelques outils spécifiques de création et de management de documents XML, qu'ils soient open-source comme Xerlin ou dans le domaine commercial comme XMLSpy permettent, eux aussi, d'éditer du SMIL en proposant des fonctionnalités intéressantes comme le contrôle de la DTD ou de l'édition de feuilles de transformation XSLT.

- d'autres solutions, un peu plus difficiles à utiliser mais toutes aussi efficaces, sont les Interfaces de développement comme Netbeans ou Eclipse (ou autres produits commerciaux de même type) qui offrent des options de traitement de documents XML relativement intéressantes (vérification syntaxique, validation, transformations XSLT...).

#### 3.3.3 Outils d'édition à base de template

On peut classer dans cette catégorie les logiciels comme PowerPoint ou Real Slide Show, qui, sur la base de modèles, permettent de créer des documents multimédia. La richesse de ces présentations étant relativement faible notamment en terme de granularité de synchronisation des différents média entre eux.

#### 3.3.4 Outils d'édition de haut niveau incluant des structures temporelles

- S'il existe dans le monde de la vidéo des outils de haut niveau permettant de gérer finement les synchronisations des média, les outils de ce type permettant de générer du code SMIL sont plus rares.

GrINS Editor est certainement le produit commercial reconnu comme étant le leader dans ce domaine. Celui-ci permet notamment de créer des présentations multimédia et de les exporter de façon optimisée (notamment en terme de Streaming – voir figure 10) soit pour le système de présentation GrINS, soit pour le système de présentation Real.

- Une alternative à ce produit est un éditeur développé dans le cadre du projet WAM : LimSee.

La version 1 de cet outil permettait d'éditer des documents en SMIL1.0. Elle bénéficiait notamment d'une vue temporelle très riche. La seconde version, LimSee2, est sortie officiellement en juin 2003. Elle est plus conviviale et dispose notamment d'une vue layout permettant un placement fin des média. Elle n'est cependant qu'en version beta et sa vue temporelle demande encore à être travaillée.

La section suivante va présenter dans le détail ce dernier outil.

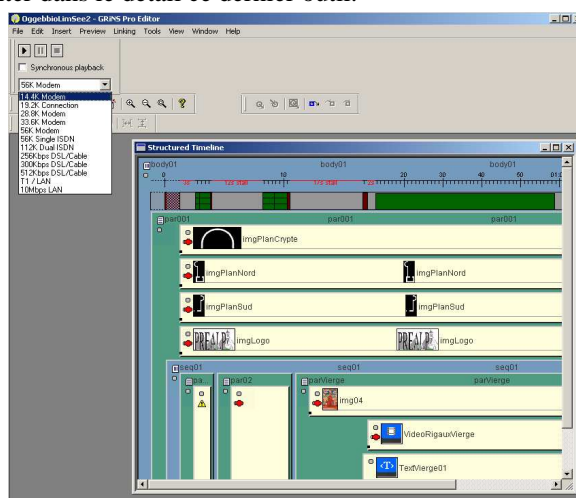


Figure 10 – Détail de la fenêtre d'édition de Grins (gestion de la bande passante)

## 4 Présentation de LimSee2

Les auteurs de documents SMIL sont très souvent plutôt des producteurs de contenus que des spécialistes du multimédia. Ils peuvent donc ne pas se sentir à l'aise avec les outils d'édition existants.

La difficulté principale, pour l'auteur, dans la manipulation d'un document SMIL est de visualiser, en cours d'édition, comment le document se comportera en phase de présentation. La solution retenue par la majorité des outils d'édition multimédia est de proposer une vue temporelle. Les objets média étant représentés sous la forme de boîtes dont la longueur représente la durée de l'objet et la position dans l'axe du temps représente son début et sa fin. LimSee suit cette approche.

LimSee, qui en est à sa seconde version (LimSee2 : <http://wam.inrialpes.fr/software/limsee2>), est distribué sous la forme d'un projet open-source depuis juin 2003. Il s'agit d'une application Java (testée sous Linux RedHat, Windows 2000/XP) qui exploite la librairie JMF (Java Media Framework) développée par Sun afin de prendre en charge les formats de média les plus courants. LimSee2 est distribué sous la forme d'un installateur convivial mais peut aussi être lancé 'en ligne' grâce à Java Web Start.

### 4.1 fonctionnalités

Cet outil d'édition de haut niveau a pour objectif de faciliter la création de documents SMIL grâce à une interface multi-vues (voir Figure 11 ci dessous).

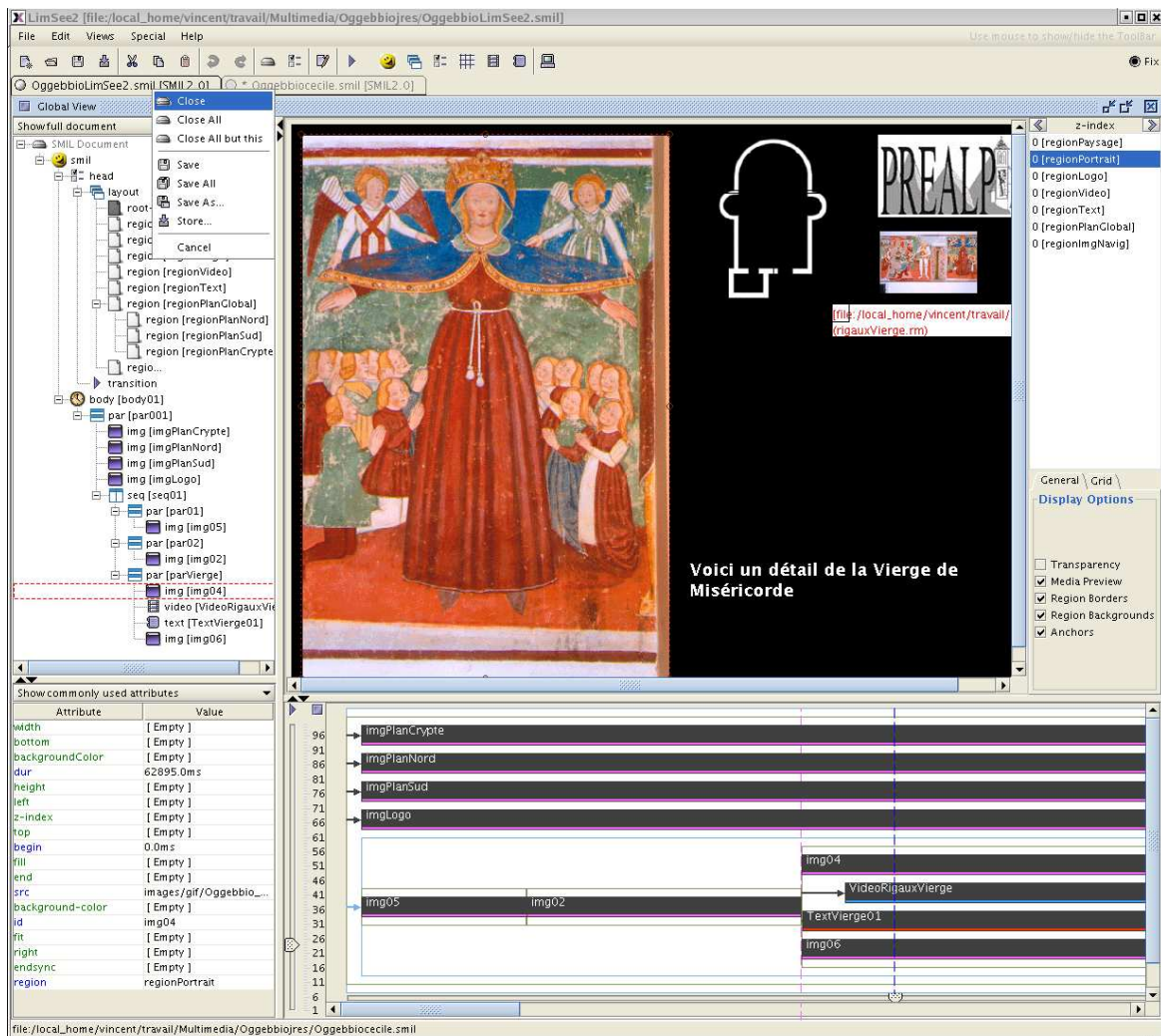


Figure 11 – Capture d'écran de l'application LimSee2

Les différentes vues proposées par l'application sont les suivantes :

- **une vue hiérarchique** (en haut à gauche) permet de manipuler la structure (XML) du document.
  - Un filtre d'affichage peut être utilisé afin de focaliser l'édition sur un sous-arbre particulier, par exemple :
    - <body> pour l'édition des éléments de synchronisation temporelle,
    - <layout> pour l'édition des caractéristiques spatiales.
  - L'insertion de nouveaux éléments est guidée par le profil utilisé (DTD SMIL1.0 ou SMIL2.0)
  - Cette vue supporte le copier-coller ainsi que le glisser-déposer
- **une vue attributs** (en bas à gauche) permet d'éditer les attributs de l'élément couramment sélectionné (La sélection d'un élément peut s'effectuer dans une des autres vues).

Cette vue complète naturellement la vue hiérarchique en permettant l'édition des attributs associés à un élément particulier. En effet, le profil du document édité intervient à nouveau, puisqu'il est possible non seulement de modifier les attributs existants, mais aussi de manipuler l'ensemble des attributs autorisés par la DTD. Ainsi, à la manière de la vue hiérarchique, un filtre d'affichage permet de limiter l'édition à un ensemble d'attributs en particulier :

  - tous les attributs autorisés par la DTD utilisée,
  - attributs renseignés (déjà spécifiés pour l'élément),
  - attributs vides (autorisés par la DTD et non-spécifiés pour l'élément),
  - attributs les plus couramment utilisés (SMIL2.0, en particulier, offre beaucoup d'attributs qui ne sont pas souvent utilisés, d'où l'intérêt de les filtrer).
- **une vue spatiale** (en haut au centre) permet l'édition conviviale (à la souris) des caractéristiques spatiales de la présentation :
  - les régions peuvent être placées et dimensionnées sur une grille paramétrable,
  - les z-index peuvent être mis à jour par glisser-déposer,
  - les média peuvent être affichés directement dans leur région respective en tenant compte de la propriété 'fit' (manière dont le média va être mis à l'échelle),
  - les liens peuvent être affichés et les documents correspondants ouverts par simple clic,
  - la visualisation est paramétrable : transparence, dessin des contours de région, affichage des couleurs de fond...
- **une vue temporelle** (en bas au centre) permet de manipuler la structure temporelle du document par l'intermédiaire d'une représentation graphique de type 'timeline' :
  - des blocs graphiques répartis horizontalement sur l'axe du temps représentent les média ainsi que les conteneurs temporels (<par> et <seq> par exemple),
  - les flèches liant les blocs entre-eux représentent les relations de synchronisation,
  - un facteur d'échelle peut être appliqué sur le temps (axe horizontal) afin d'adapter la taille de la vue à l'espace disponible sur l'écran,
  - un curseur temporel peut être positionné sur l'axe horizontal, ce qui a pour effet de visualiser dans la vue spatiale l'état des objets média à l'instant t donné.

Cette fonctionnalité s'avère essentielle pour la réalisation de scénarios d'édition très répandus, comme par exemple le calage de sous-titres sur une vidéo.

LimSee2 propose aussi quelques fonctionnalités intéressantes :

- l'intégration d'un outil de présentation (Xsmiles.org) permet la prévisualisation du document en cours d'édition sans avoir besoin de le sauver au préalable,
- un outil permettant l'aspiration complète d'une présentation SMIL : le fichier SMIL (XML), les fichiers média exploités (vidéo, images, sons, textes...), ainsi que tous les documents SMIL liés (<a>, <anchor>...) peuvent être copiés localement dans un répertoire unique (la procédure est récursive pour chaque document lié).

L'arborescence de fichier ainsi créée peut ainsi être facilement compressée puis sauvegardée,
- un utilitaire d'importation de documents PowerPoint existants génère automatiquement un squelette SMIL de type diaporama et en permet l'édition directe,
- la gestion de l'encodage des fichiers SMIL (UTF8, ISO-LATIN-1, US-ASCII...),
- l'ouverture incrémentale des fichiers XML mal-formés (par corrections d'erreurs successives jusqu'à l'obtention d'un fichier bien formé)
- la validation des documents selon le profil d'édition (SMIL1.0, SMIL2.0).

## 4.2 Evolutions prévues

Cette application est en cours de développement et évolue vers plus d'utilisabilité (intégration de modèles, prise en compte des scénarios d'édition les plus courants), et vers une prise en charge plus complète des fonctionnalités SMIL2.0 (switch, indéterminisme, animations et transitions).

L'indéterminisme reste une des caractéristiques les plus délicates à traiter car les paradigmes d'édition qui y sont associés sont complexes et difficiles à mettre en oeuvre. SMIL2.0 permet en effet l'utilisation de valeurs temporelles déterminables uniquement lors de la présentation (clic souris), mais aussi d'un conteneur nondéterministe <excl> (dont un seul des fils peut être joué à un instant donné) et d'une valeur temporelle indéterminée 'indefinite'.

L'intégration des outils de visualisation les plus utilisés (RealOne, WindowsMediaPlayer, QuickTime...) est au programme mais cela relève plutôt de problèmes techniques liés à la plate-forme d'exécution.

## 5 Conclusion

Comme nous avons pu le voir, les difficultés rencontrées lors de l'édition de documents SMIL sont principalement dues à l'hétérogénéité des possibilités de visualisation et donc à la quasi impossibilité de réaliser un document SMIL lisible par l'ensemble des outils de visualisation du marché.

Gageons que l'appropriation de ce standard par les producteurs de documents engage les éditeurs de logiciels à implémenter des outils de visualisation respectant strictement le standard.

Il restera cependant toujours quelques points d'achoppement, et notamment la visualisation du texte formaté pour lequel la DTD de SMIL2.0 reste très ouverte.

Par ailleurs, l'édition de présentations multimédia dont la structure s'appuie principalement sur des médias indéterministes et des événements sera toujours aussi difficile à concevoir et à mettre au point.

A partir de l'expérience décrite ici, nous pouvons formuler une recommandation pour les personnes qui réalisent des présentations en SMIL : celle d'essayer de suivre au maximum le standard en tenant compte des outils de visualisation usuellement utilisés actuellement (Real notamment) tout en se disant que certaines fonctionnalités (rendu du texte) sont inopérantes sur certaines plates-formes : donc faire à minima en espérant que les prochains outils nous permettront d'introduire dans nos documents l'ensemble des possibilités offertes par le standard.

Nous conseillons, bien sûr, d'effectuer une veille technologique pointue sur l'évolution des outils, notamment en suivant ce qui se fait autour de la communauté Helix.

L'utilisation d'un standard du W3C pour l'écriture de ce type de document est donc un pari sur l'avenir en espérant que, le nombre de documents SMIL disponibles augmentant, des solutions soient rapidement trouvées pour garantir les possibilités de lecture de tels documents sans passer par des solutions moins ouvertes (xhtml + time par exemple).

## 6 Remerciements

Nous tenons à remercier particulièrement :

- Dominique Rigaux, responsable du programme PREALP à la Maison des Sciences de L'Homme-Alpes à Grenoble et Professeur d'Histoire du Moyen Age à l'Université Pierre Mendès-France de Grenoble, pour le temps qu'elle a passé avec nous à décrire les besoins de son équipe, et la fourniture des média nécessaires à nos travaux.
- Nabil Layaida et Tayeb Lemlouma, du projet WAM, pour leurs précieux conseils sur SMIL2.0 et l'adaptation de documents SMIL.
- Silvano Ferraris, auteur des photographies insérées dans ce document.

## 7 Références

### **Etat de l'art des outils d'édition :**

SMIL Authoring Tools : a state of the art

<http://aristote1.aristote.asso.fr/Presentations/SMIL-2003/P/Bulterman/SmilEurope-Bulterman.pdf>

*Dick Bulterman (Oratrix)*

### **Présentations de la conférence SMILEurope 2003 (12-13 et 14 février 2003)**

<http://aristote1.aristote.asso.fr/Presentations/SMIL-2003/PresentationTalks.htm>

### **Adaptation de Contenu :**

Adapted Content Delivery for Different Contexts :

<http://wam.inrialpes.fr/publications/index.fr.html#Adapted>

*T. Lemlouma, N. Layaïda*

### **Recommandation SMIL1.0 :**

<http://www.w3.org/TR/REC-smil/>

### **Recommandation SMIL2.0 :**

<http://www.w3.org/TR/smil20/>

### **W3C (Multimédia Synchronisé)**

<http://www.w3.org/AudioVideo/>

### **Limsee2 :**

<http://wam.inrialpes.fr/software/limsee2>

### **PocketSMIL:**

<http://wam.inrialpes.fr/software/pocketsmil/>

### **Helix community :**

<https://www.helixcommunity.org/>

### **Xsmiles.org :**

<http://www.x-smiles.org/>

### **RealOne :**

<http://www.real.com/realone/index.html>

### **Internet Explorer :**

<http://www.microsoft.com/windows/ie/default.asp>

### **Sun (Java Media Framework)**

<http://java.sun.com/products/java-media/jmf/>

### **Sun (Machine Virtuelle) :**

<http://java.sun.com/j2se/>

### **Sun (Java Web Start) :**

<http://java.sun.com/products/javawebstart/>