Langages de publication temporalisée

Ingénierie Documentaire
http://doc.crzt.fr

Stéphane Crozat

14 septembre 2016

Table des matières

Introduction à SMIL

I

1. Qu'est ce que SMIL ?

Définition

SMIL (*Synchronised Multimedia Integration Language*, prononcé *smile* à l'anglaise, pour "sourire") est un langage XML de présentation multimédia fondé sur la synchronisation de contenus spatiaux et temporels.

SMIL est un standard (*W3C recommendation*) du W3C, dans sa version 3 depuis 2008.

http://www.w3.org/TR/SMIL3/smil-timing.html

Méthode : SMIL est un format de publication

SMIL permet de spécifier des présentations multimédias : c'est un langage de de mise en forme spatiale et temporelle.

Les présentations SMIL sont engendrées la plupart du temps par une transformation XSLT à partir de contenus XML et de ressources binaires (vidéos, sons, photos).

Attention : Pour comprendre plus que pour faire...

SMIL est un concurrent standard et déclaratif de Flash. Mais SMIL reste peu utilisé dans les faits, peu d'outils le lisent, aucun ne l'exécute parfaitement, et le succès de Flash n'a pas favorisé son développement.

SMIL a probablement peu d'avenir, mais reste intéressant comme approche pédagogique de la synchronisation de contenus multimédias.

Exemple : Lecteurs SMIL : Real et Ambulant

- *Real One Player*, le lecteur commercial le plus avancé, mais qui ne couvre pas l'ensemble du standard, et avec de nombreux soucis d'implémentation.
- *Ambulant Player*, une alternative *open source*, plus fidèle au standard.

Complément : LimSee2 : Un éditeur WYSIWYG pour SMIL

http://limsee2.gforge.inria.fr

Complément : Historique

SMIL est un formalisme puissant, couvrant les principaux besoins de représentation de documents temporels.

Différentes versions de la norme ont vu le jour, la première, SMIL 1.0, fut élaborée en 1998 et permit de poser les fondements de ce langage. Un effort important de structuration des différentes composantes d'une publication multimédia a été apporté par la version 2.0. La version actuelle est la version 3.0.

La norme se structure en plusieurs modules permettant d'obtenir le profil complet du langage SMIL. Chaque module traite un aspect de la publication multimédia. Il est ainsi possible de ne sélectionner que quelques sous-modules afin de satisfaire des besoins spécifiques ne nécessitant pas l'utilisation de l'intégralité du standard.

Complément : Les modules SMIL

- Les modules d'objets média (ils précisent les types de médias supportés et les attributs associés).
- Le module de méta-information.
- Le module de structure (il précise l'ossature d'un document SMIL).
- Le module de disposition (il permet l'agencement des zones graphiques d'affichage des médias).
- Le module de temporisation et de synchronisation des médias.
- Les modules de lien.

2. Structure de base d'un contenu SMIL

Attention : Simplification volontaire

Les structures présentées sont simplifiées à des fins pédagogiques afin de permettre une appréhension rapide des principes du langage.

Il existe de nombreux autres éléments dans le standard, en conséquence les schémas proposés ne sont pas les schémas standards, mais ils permettent de créer des documents valides par rapport aux schémas standards (les schémas proposés sont inclus dans les schémas standards).

Syntaxe : smil

Un document SMIL comprend une entête, `<head>` et un corps `<body>`.

```
1 <!ELEMENT smil (head?,body)>
```

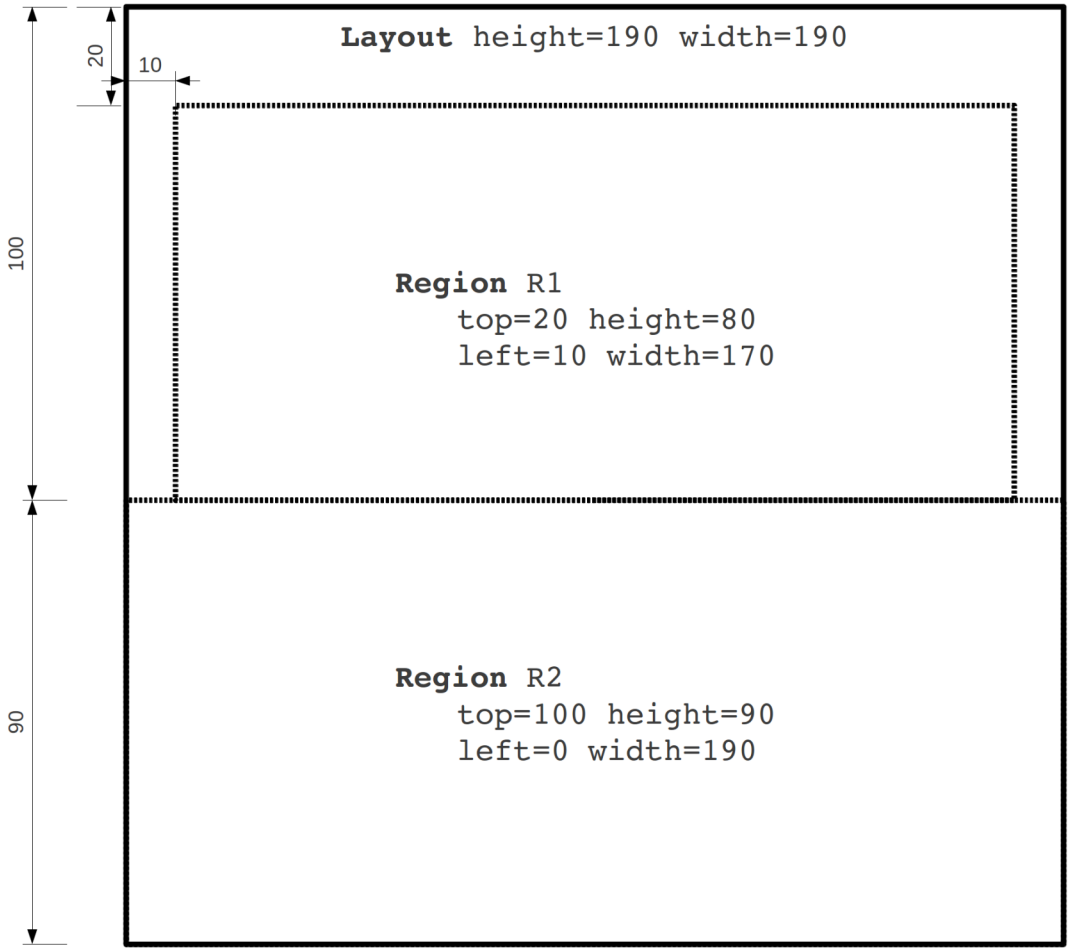
Syntaxe : head

L'entête permet (notamment) de déclarer :

- des méta-données `<méta>`
- une structure générale de l'écran de présentation découpé en zones `<layout>`

```
1 <!ELEMENT head (meta*,layout?)>
2 <!ELEMENT meta EMPTY>
```

```
<!ATTLIST meta
name (title | author) #REQUIRED
content CDATA #REQUIRED>
<!ELEMENT layout (root-layout, region*)>
<!ELEMENT root-layout EMPTY>
<!ATTLIST root-layout
width CDATA #REQUIRED
height CDATA #REQUIRED>
<!ELEMENT region EMPTY>
<!ATTLIST region
id ID #REQUIRED
width CDATA #REQUIRED
height CDATA #REQUIRED
left CDATA #IMPLIED
top CDATA #IMPLIED>
```



Exemple de définition de layout

Syntaxe : body

Le corps permet de déclarer une séquence `<seq>` d'éléments parallèles `<par>` : Chaque `<par>` est un ensemble de ressources synchronisées (textes, images, sons, vidéos, etc.).

Chaque ressource est associée à :

- une région `region` du *layout*

- une durée `dur` en secondes
- éventuellement une date de début `begin` en seconde

```
<!ELEMENT body (seq)>
<!ELEMENT seq (par+)>
<!ELEMENT par (text | img | audio | video)*>
<!ELEMENT text EMPTY>
<!ATTLIST text
src CDATA #REQUIRED
region CDATA #REQUIRED
dur CDATA #REQUIRED>
<!ELEMENT img EMPTY>
<!ATTLIST img
src CDATA #REQUIRED
region CDATA #REQUIRED
dur CDATA #REQUIRED>
<!ELEMENT audio EMPTY>
<!ATTLIST audio
src CDATA #REQUIRED
region CDATA #REQUIRED
dur CDATA #REQUIRED>
<!ELEMENT video EMPTY>
<!ATTLIST video
src CDATA #REQUIRED
region CDATA #REQUIRED
dur CDATA #REQUIRED>
```

Exemple

```
<?xml version="1.0" encoding="UTF-8"?>
<smil>
    <head>
        <meta name="title" content="Titre de mon projet"/>
        <meta name="author" content="Stéphane Crozat"/>
    <layout>
        <root-layout width="600" height="600"/>
            <region id="image" width="550" height="400" left="25" top="0"/>
            <region id="texte" width="600" height="200" left="0" top="400"/>
        </layout>
    </head>
    <body >
        <seq>
            <par>
                <img src="image01.jpg" region="image" dur="2"/>
                <text src="texte01.rt" region="texte" dur="2"/>
            </par>
            <par>
                <img src="image02.jpg" region="image" dur="2"/>
                <text src="texte02.rt" region="texte" dur="2"/>
            </par>
        </seq>
    </body>
</smil>
```

Complément : Layout

Une publication multimédia nécessite de spécifier un agencement spatial particulier pour les différentes ressources convoquées. L'approche adoptée par SMIL consiste :

1. À déclarer un canevas général (*layout*) puis à y positionner les différentes régions d'affichage des zones d'affichage.
   Chaque région est :
   - nommée (attribut `id`),
   - positionnée (attributs `left` et `top`),
   - dimensionnée (attribut `width` et `height`),
   - et éventuellement ordonnée (attribut `z-index` permettant d'exprimer l'empilement).
2. À associer à chaque média, dans le corps de la présentation, une de ces zones, par l'attribut `region`..

```
<head>
  <meta name="title" content="Demo"/>
  <layout>
    <root-layout width="800" height="600"/>
    <region id="rg_video1" z-index="1" left="36" top="138" width="378" height="294"/>
  </layout>
</head>
```

```
<body>
  <video .... region="rg_video1" src="MaVideo.mpg" >
</body>
```

Complément : Pour aller plus loin...

"Publier à partir de XML : SMIL" (par Bruno Bachimont) (cf. nf29p2009_9a.pdf)

Complément : Tutoriels en ligne

http://real-and-smil.com/tutorialsmil.php

3. Médias externalisés et temporalisés dans une présentation SMIL

⚠ *Attention : Médias externalisés*

SMIL ne décrit pas de contenu, mais vise à expliciter une scénarisation de médias.

Aucun contenu n'est donc présent dans un document SMIL, tous sont adressés via les mécanismes d'URL.

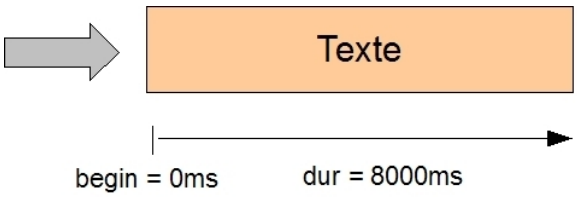
⚠ *Attention : Médias temporalisés*

Toutes les ressources sont temporalisés dans le flux de la présentation : attributs `dur`, `begin` et `end`.

Ainsi un texte possède une durée d'affichage tout comme une vidéo.

Exemple : Texte

```
1 <text begin="0ms" dur="8000ms" region="rg_TxtFlagBas" src="txt/UnBlocdeTexte.html" >
```



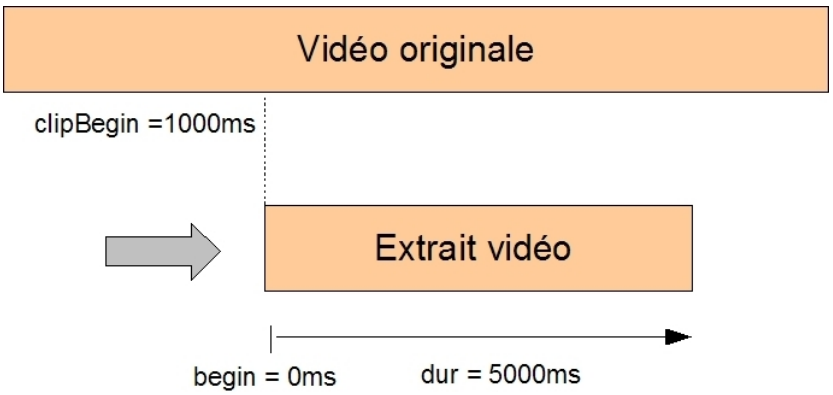
Texte temporalisés

Remarque : clipBegin

Les documents intrinsèquement temporels possèdent une temporalité par défaut qui peut être altérée : attribut `clipBegin`.

Exemple : Vidéo

```
1 <video begin="0ms" clipBegin="1000ms" dur="5000ms" region="rg_video1" src="MaVideoOriginale.mpg"
  />
```



Vidéo re-temporalisée

Complément : Types de médias supportés

SMIL permet de décrire des présentations multimédias complexes mêlant :

- des textes : `<text>`
- des images : `<image>`
- des sons : `<audio>`
- des vidéos : `<video>`
- ...

Complément : textSmil

`smilText` permet de gérer du texte internalisé.

Texte internalisé SMIL 3.0 (smilText) (cf. p.11)

4. Agencement temporel des médias

Fondamental

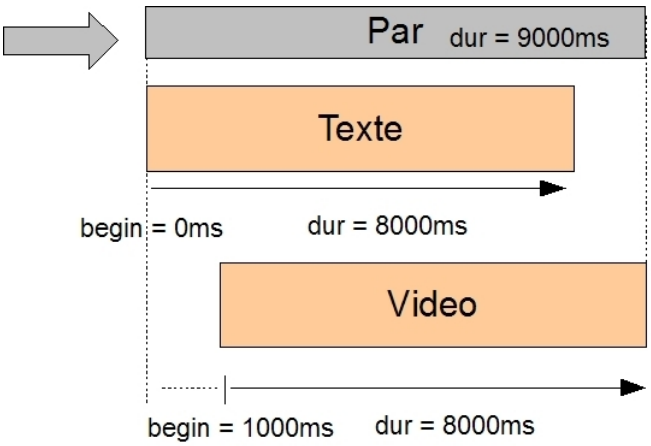
La fonction principale de SMIL est d'organiser l'agencement temporel des médias au moyen de deux éléments conteneurs permettant d'exprimer :

- la mise en *parallèle* : balise `<par>`
- ou en *séquence* : balise `<seq>`

Ces conteneurs sont temporalisés et peuvent s'imbriquer à volonté.

Syntaxe : <par>

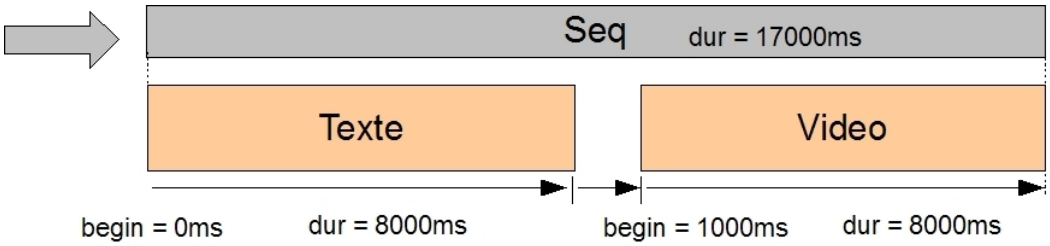
L'élément `<par>` implique que le lancement des fils sera effectué en parallèle.



élément <par>

Syntaxe : <seq>

L'élément `<seq>` implique que les fils seront séquencés.

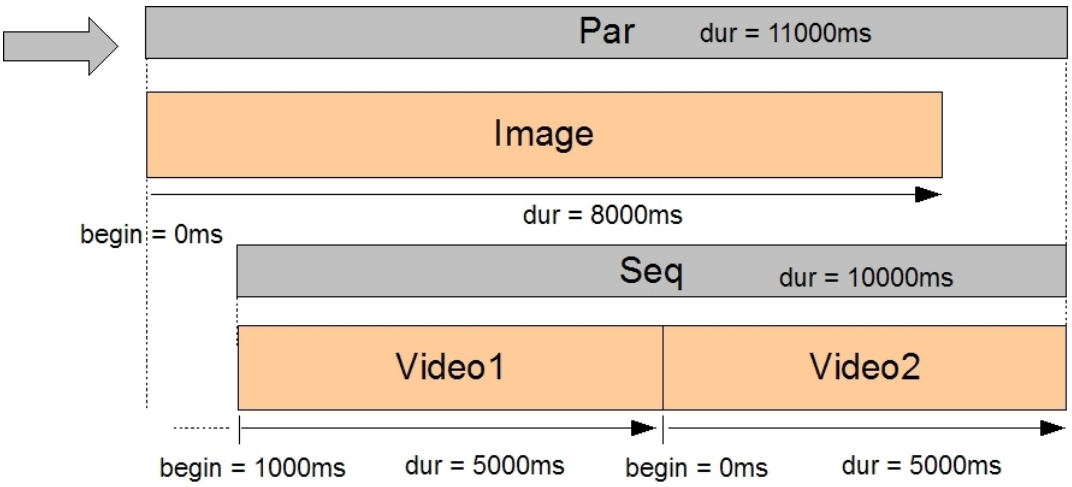


Élément <seq>

Exemple

```
1 <par begin="0ms" dur="11000ms" >
2   <image begin="0ms" dur="8000ms" region="rg_ImgDte" src="portait.jpg" />
3   <seq begin="1000ms" dur="10000ms" >
```

```
4     <video begin="0ms" clipBegin="0ms" dur="5000ms" region="rg_video1" src="v1.mpg" / >
5     <video begin="0ms" clipBegin="10000ms" dur="5000ms" region="rg_video1" src="v2.mpg" / >
6   </seq>
7 </par>
```



Temporalisation avec <par> et <seq>

5. Texte externalisé avec Real (RealText)

Les textes utilisables dans SMIL et lisibles dans Real PLayer sont écrits selon le format Real Text (rt).

Ce format est un "sous" XML, c'est à dire un langage XML qui doit néanmoins respecter des contraintes de syntaxe supplémentaires par rapport à XML.

⚠ *Attention : Un "sous" XML*

1. Les texte RT ne doivent pas avoir de prologue `<?xml?>`, ils commencent donc directement par l'élément racine.
2. Les espaces sont pris en compte, l'indentation n'est donc pas neutre pour la mise en forme

Exemple : Texte RT

```
1 <window height="200" width="600" bgcolor="yellow" >
2 <font face="arial" color="blue">
3 <br/>
4 Première ligne<br/>
5 Seconde ligne
6 </font>
7 </window>
```

Syntaxe : Texte temporalisé

Il est possible de temporiser l'affichage du texte en encadrant ce texte dans un élément `<time begin="x">`.

Exemple : Texte RT temporalisé

```
<window height="200" width="600" bgcolor="yellow" >
<!-- NB : Durée du texte = 2-->
<font face="arial" color="blue">
<br/>
Première ligne<br/>
Début de la seconde ligne <time begin="1">suite de la seconde ligne <br/></time>
<time begin="2">Troisième ligne</time>
</font>
</window>
```

Complément : Pour aller plus loin

http://www17.real.com/help/library/guides/realtext/realtext.htm

6. Texte internalisé SMIL 3.0 (smilText)

`smilText` a été ajouté à SMIL 3.0 afin de gérer du texte internalisé dans les fichiers SMIL (sans recours à un fichiers externe).

SMIL 3.0 smilText, W3C Recommendation

http://www.w3.org/TR/2006/WD-SMIL3-20061220/smil-text.html

Syntaxe

`smilText` pose les éléments suivants :

- `<smilText region="...">...</smilText>` : balise encadrante du texte
- `<br/>`

Section 8.14 "SMIL 3.0 BasicText Module", *http://www.w3.org/TR/2006/WD-SMIL3-20061220/smil-text.html#SMILtextNS-BasicText*

Syntaxe : Temporalisation du texte

```
<smilText dur="2s" ...
```

Syntaxe : Stylage

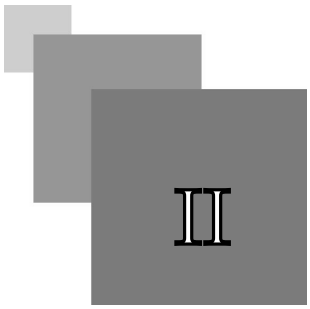
- `<smilText textColor="green" ...` : Stylage du bloc de texte
- `<span ...` : Stylage *inline*

Section 8.5 "SMIL 3.0 TextStyling Module", *http://www.w3.org/TR/2006/WD-SMIL3-20061220/smil-text.html#SMILtextNS-TextStyling*

Exemple

```
<?xml version="1.0" encoding="UTF-8"?>
<smil>
<head>
```

```
  <meta name="title" content="Titre du document"/>
  <meta name="author" content="Stéphane Crozat"/>
  <layout>
    <root-layout width="600" height="600"/>
    <region id="image" width="550" height="400" left="25" top="0"/>
    <region id="texte" width="600" height="200" left="0" top="400"/>
  </layout>
</head>
<body>
  <seq>
    <par>
      <img src="image01.jpg" region="image" dur="2"/>
      <smilText region="texte" dur="2s">
        Le texte de la première <span textFontWeight='bold'>ligne</span>
        <br/>
        Celui de la seconde ligne.
      </smilText>
    </par>
    <par>
      <img src="image02.jpg" region="image" dur="2"/>
      <smilText region="texte" dur="2s" >
        Le second texte.
      </smilText>
    </par>
  </seq>
</body>
</smil>
```



Environnements JavaScript de temporalisation

1. HTML5 + Timesheet

Définition

HTML5 + Timesheet (*http://wam.inrialpes.fr/timesheets*) est une solution JavaScript développée par l'INRIA dans le cadre du projet de recherche C2M (*http://www.utc.fr/ics/c2m*).

L'objectif est d'étendre HTML5 avec des attributs SMIL pour intégrer la synchronisation, notamment de ressources audiovisuelles.

Il permet de définir des *Timesheets* (feuilles de temps), qui dans l'esprit des feuilles de styles, permettent de mutualiser les scénarisations récurrentes.

Syntaxe : Documentation

http://wam.inrialpes.fr/timesheets/docs/

Exemple : Bannière

```
1 ...
2 <script type="text/javascript" src="timesheets.js"/>
3 <link href="banner.smil" rel="timesheet"
4 type="application/smil+xml"/>
5 <div id="banner">
6 <img src="image1.png"/>
7 <img src="image2.png"/>
8 <img src="image3.png"/>
9 </div>
10 ...
```

```
1 <timesheet xmlns="http://www.w3.org/ns/SMIL">
2 <!-- banner.smil -->
3 <seq repeatCount="indefinite">
4 <item select="#banner img" dur="3s"/>
5 </seq>
6 </timesheet>
```

http://wam.inrialpes.fr/timesheets/markup/

Exemple : Diaporama

```
<!DOCTYPE html>
<html>
<head>
[...]
<link href="timesheet.smil" rel="timesheet" type="application/smil+xml">
<script type="text/javascript" src="timesheets.js"></script>
<script type="text/javascript" src="timecontroller.js"></script>
</head>
<body>
<div id="slideshow">
<header id="slide01"> [...] </header>
<div id="slide02"> [...] </div>
<div id="slide03"> [...] </div>
<div id="slide04"> [...] </div>
<div id="slide05"> [...] </div>
<div id="slide06"> [...] </div>
<div id="slide07"> [...] </div>
<div id="slide08"> [...] </div>
<div id="slide09"> [...] </div>
<div id="slide10"> [...] </div>
<div id="slide11"> [...] </div>
<footer id="slide12"> [...] </footer>
</div>
<audio id="talk" autoplay preload>
<source type="audio/ogg" src="media/joinMozilla.ogg"/>
<source type="audio/mp4" src="media/joinMozilla.m4a"/>
</audio>
<nav id="timeController" class="smil-timeController">
[...]
</nav>
</body>
</html>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- timesheet.smil -->
<timesheet xmlns="http://www.w3.org/ns/SMIL">
<excl mediaSync="#talk" controls="#timeController">
<item select="#slide01" begin="00:00"/>
<item select="#slide02" begin="00:09"/>
<item select="#slide03" begin="00:50"/>
<item select="#slide04" begin="01:18"/>
<item select="#slide05" begin="01:48"/>
<item select="#slide06" begin="02:23"/>
<item select="#slide07" begin="03:55"/>
<item select="#slide08" begin="06:20"/>
<item select="#slide09" begin="07:17"/>
<item select="#slide10" begin="09:11"/>
<item select="#slide11" begin="10:47"/>
<item select="#slide12" begin="13:03"/>
</excl>
</timesheet>
```

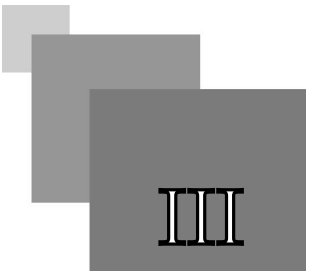
http://wam.inrialpes.fr/timesheets/slideshows/audio.html

2. Popcorn.js

Popcorn.js est un *framework* JavaScript pour HTML5 destiné à la création d'animation multimédia Web.

http://popcornjs.org/

Exercices



1. Travaux pratiques SMIL et HTML+Timesheets

1.1. Exercice : Photos de vacances commentées [60 min]

Écrire un fichier SMIL permettant d'enchaîner des photos de vacances avec pour chacune une phrase d'illustration.

- Les photos devront être de taille 640*480 (on ne gérera que des photos en "paysage")
- Le texte s'affichera dans une fenêtre de 640*200
- Chaque partie (photo + texte) s'affichera durant 3 secondes.

Question 1

Récupérer 2 ou 3 photos de vacances et les mettre au format 640*480.

Question 2

Définir le `layout`.

Question 3

Créer un fichier SMIL ne contenant que les photos, le jouer dans *Real Player* et *Amulant Player*.

Indice :

http://ambulantplayer.org/

http://www.real.com/

Question 4

Ajouter les textes de commentaires au fichier SMIL en utilisant `smilText`.

Jouer le nouveau fichier avec *Amulant Player*.

Question 5

Écrire des fichiers textes de commentaires pour *Real Player* (un fichier pour chaque photo).

Écrire et jouer le fichier SMIL avec texte pour Real Player.

Question 6

Écrire un schéma RelaxNG et un fichier XML valide permettant de représenter des photos de vacances commentées.

Question 7

Écrire une XSLT permettant de transformer ce fichier XML en fichier SMIL.

1.2. Exercice : Webradio [2h]

Soit le contenu multimédia *HTML5 + Timesheet* ci-après.

Télécharger le contenu en ligne.

[cf. Exemple de contenu HTML5+Timesheet]

Question 1

Exécuter ce contenu, l'analyser pour en comprendre les principes de fonctionnement.

Indice :

HTML5 + Timesheet (cf. p.13)

Question 2

Réaliser une chaîne XML permettant la publication de tels contenus :

- Schéma
- Instances exemples
- Publication temporalisée
- Éditeur
- Publication paginée