

# Technologies du Web

## Contenus multimédias riches

Pierre Senellart ([pierre.senellart@telecom-paristech.fr](mailto:pierre.senellart@telecom-paristech.fr))



Mastère spécialisé *Management et nouvelles technologies*,  
24 octobre 2008

# Plan du cours

- 1 Introduction
- 2 AJAX
- 3 Applets Java
- 4 Adobe Flash
- 5 Alternatives
- 6 Son et vidéo sur le Web
- 7 Application

# Applications Web

- Le Web n'est plus seulement utilisé pour présenter des information textuelles, mais héberge de véritables **applications**.

## Exemples

- ▶ Messagerie électronique
  - ▶ Applications bureautiques
  - ▶ Cartographie
  - ▶ Jeux
  - ▶ Contenu multimédia
- En général, applications conçues uniquement pour fonctionner avec des **navigateurs graphiques**.
  - HTML, CSS, et même JavaScript n'ont pas **du tout** été conçus pour ça : d'autres technologies sont (souvent) nécessaires.

# Web 2.0

- **Web 2.0**, terme marketing ne correspondant pas à un concept technique précis, mais recouvrant :

**Phénomène social.** Un Web dans lequel l'utilisateur **contribue à la production de contenu** (wikis, blogs et systèmes de commentaires, partage de fichiers multimédias, réseaux sociaux. . .)

**Phénomène technique.** Des interactions de plus en plus riches **à l'intérieur du navigateur** : applications Web similaires aux applications sur ordinateur de bureau, complétion automatique, etc.

- En particulier, interactions utilisant souvent l'ensemble de technologies **AJAX**.

# Dangers potentiels

- **Accessibilité restreinte** . . .
- . . . justifiable quand il s'agit de construire des applications complexes, mais **pas pour présenter du contenu textuel** !
- Perte des fonctionnalités habituelles d'un navigateur (marque-pages, historique, gestion des mots de passe, etc.)
- Quid des autres agents Web, en particulier **robots des moteurs de recherche** ?
- **Complexification** du développement Web (empilement de technologies), entraînant potentiellement bugs, problèmes de conception et de maintenance
- Plug-ins à ajouter à un navigateur : multiplication des **failles de sécurité potentielles**

# Plan du cours

- 1 Introduction
- 2 AJAX**
- 3 Applets Java
- 4 Adobe Flash
- 5 Alternatives
- 6 Son et vidéo sur le Web
- 7 Application

# AJAX

- **A**synchronous **J**avascript and **X**ML
- Pas une technologie, mais un ensemble de technologies, comme DHTML :
  - DHTML HTML+CSS+JavaScript
  - AJAX HTML+CSS+JavaScript+XML+XMLHttpRequest
- Partie importante : la classe JavaScript **XMLHttpRequest**
- Permet des échanges à l'intérieur d'une même page Web entre code JavaScript et programme tournant sur un serveur
- **Asynchrone** : données traitées quand elles arrivent, de manière non bloquante
- **XML** : données renvoyées par le serveur en général en XML (cf. cours ultérieur)
- Utile pour autocomplétion, Webmails, rafraîchissement des informations d'une partie de la page, etc.

# Support dans les navigateurs

- Introduit par Microsoft dans IE5, sous une forme un peu instable
- Très bonne idée ! D'autres manières de faire la même chose (p. ex., `<iframe>` ), mais beaucoup moins pratique.
- Repris par l'ensemble des navigateurs graphiques depuis, sous une forme simplifiée et standardisée (y compris IE7)
- Travail en cours du W3C pour la normalisation (a posteriori)

# Principe d'utilisation

- On crée un objet **XMLHttpRequest** en lui donnant l'URL d'un script (avec éventuellement des paramètres) à contacter.
- On associe à cet objet une fonction qui traitera les données récupérées. En général, les données récupérées sont sous forme **XML**, et on les traite en JavaScript avec les fonctions de **manipulation DOM**, comme on traiterait du HTML.
- On envoie la requête.
- Une fois la requête terminée, la fonction de traitement s'exécute (de manière **asynchrone**).

# Exemple d'utilisation de XMLHttpRequest

## Exemple

```
request = new XMLHttpRequest();
request.open("GET", url);

request.onreadystatechange = function() {
  if (request.readyState == 4 && request.status == 200) {
    result=request.responseXML;
    /* faire quelque chose avec responseXML */
  }
};
request.send(null);
```

## Compatibilité avec Internet Explorer 5 et 6

```
request=null;
if(typeof XMLHttpRequest != "undefined")
    request = new XMLHttpRequest();
else {
    try {
        request = new ActiveXObject("Msxml2.XMLHTTP.6.0");
    } catch(e) {
        try {
            request = new ActiveXObject("Msxml2.XMLHTTP.3.0");
        } catch(e) {
            try {
                request = new ActiveXObject("Msxml2.XMLHTTP");
            } catch(e) {
                request = new ActiveXObject("Microsoft.XMLHTTP");
            }
        }
    }
}
```

## Fonctionnalités JavaScript utiles

- Minuterie en JavaScript (utile pour retarder l'exécution d'une requête, en attendant de voir si une autre requête ne doit pas être lancée tout de suite après)

```
timer=window.setTimeout("fonction_js()",200); /* 200ms */  
window.clearTimeout(timer);
```

- `document.createTextNode(data)` pour créer un nouveau nœud textuel
- `document.importNode(element, true)` pour pouvoir transférer un élément d'un document à un autre (d'un document XML au document HTML courant par exemple)
- Des **variables globales** peuvent être déclarées et utilisées à travers plusieurs fonctions en mettant (p. ex.) `toto=null` en dehors de toute fonction.

# Plan du cours

- 1 Introduction
- 2 AJAX
- 3 Applets Java**
- 4 Adobe Flash
- 5 Alternatives
- 6 Son et vidéo sur le Web
- 7 Application

# Le langage Java

- **Java** : langage de programmation **orienté objet**, utilisé pour écrire des logiciels complets
- Syntaxe proche de JavaScript et PHP, mais plus rigoureuse (langage fortement typé, toutes les variables doivent être déclarées, etc.)
- Langage **compilé** (et non interprété comme JavaScript ou PHP) vers du code pour une **machine virtuelle**
- Nécessite donc un environnement de développement ou **JDK** (pour transformer le code source en code pour machine virtuelle) et un environnement d'exécution ou **JRE** (pour exécuter le code de la machine virtuelle)
- Implémentation de référence : **Sun**, récemment devenue libre
- JRE disponibles pour de très nombreuses plate-formes, y compris téléphones mobiles basiques
- Assez lourd !

## Applets Java

- Programme Java à l'intérieur du navigateur dans une zone de taille fixe
- Intégration via un plugin de navigateur, à installer séparément (disponible pour tous les navigateurs graphiques modernes)
- Intégration avec HTML :

```
<applet code="toto.class" width="300" height="400">  
  <param name="param" value="value">  
  <!-- Contenu alternatif -->  
</applet>
```

- `<applet>` est normalement déprécié en faveur de `<object>`, mais pas de manière simple de faire fonctionner des applets Java avec `<object>` dans tous les navigateurs graphiques
- Accès à toute la bibliothèque de fonctions standards du langage Java (conséquente) et possibilité d'utiliser les très nombreuses bibliothèques supplémentaires existantes

# Système de confiance

Deux niveaux de confiance en une applet :

- Par défaut, une applet (tout comme JavaScript, Flash. . .) est confinée dans un **bac à sable**, n'a pas accès à l'extérieur du navigateur (système d'exploitation, contenu du disque dur de l'ordinateur, etc.)
- Une **trusted applet** (demande de confirmation faite par le navigateur) aura par contre les mêmes droits d'accès qu'un programme classique s'exécutant sur la machine locale.

## Intérêts/inconvénients

- Assez ancien, et initialement conçu comme **la** façon d'avoir une application complexe à l'intérieur d'un navigateur
- Bon support pour divers navigateurs et plates-formes, mais à condition d'installer le plugin et le JDK
- Assez lourd à mettre en place, les applets mettent en général du temps à démarrer
- Trusted applets : Bonne manière de déployer une application lourde sur plusieurs plates-formes (cf. Impôt sur le revenu)
- Possibilités graphiques existantes, mais non mises en avant

# Plan du cours

- 1 Introduction
- 2 AJAX
- 3 Applets Java
- 4 Adobe Flash**
- 5 Alternatives
- 6 Son et vidéo sur le Web
- 7 Application

# Flash

- Adobe Flash (anciennement, Macromedia Flash)
- À l'origine prévu pour des **animations vectorielles** : images vectorielles + **ActionScript** (basé sur ECMAScript)
- Comme les applets Java, s'exécute dans une zone délimitée du navigateur
- Programmes flash compilés dans un format binaire (SWF)
- Support correct mais pas exceptionnel :
  - ▶ Pas de support des Unix exotiques : AIX, HP-UX, OpenBSD, etc.
  - ▶ Pas de support pour de nombreux téléphones mobiles
  - ▶ Pas de support pour les architectures 64bits; critique pour les distributions Linux sur ordinateurs récents
- La description du format SWF est publique, mais jusque mai 2008, Adobe interdisait de l'utiliser pour créer des lecteurs Flash : pas d'alternative libre crédible au plugin officiel Adobe Flash.
- Environnement de développement : payant (novembre 2008 : 699\$).

# Adobe Flex

- Langage de programmation permettant de créer des applications Flash, de manière programmatoire (et non plus graphique comme traditionnellement)
- Rien ne change pour l'utilisateur
- Possibilité
- Compilateur (mais pas environnement de développement) libre mis à disposition par Adobe

# Intégration avec HTML

Avec la balise `<object>` :

```
<object data="toto.swf"  
  type="application/x-shockwave-flash"  
  width="300" height="400" >  
  <param name="param" value="value">  
  Contenu alternatif  
</object>
```

Méthode standard, peut nécessiter des workarounds pénibles pour Internet Explorer

# Plan du cours

- 1 Introduction
- 2 AJAX
- 3 Applets Java
- 4 Adobe Flash
- 5 Alternatives**
- 6 Son et vidéo sur le Web
- 7 Application

# Silverlight

- Technologie Microsoft assez récente (première version décembre 2006)
- Alternative à Flash, fonctionne sur le même principe et suivant le même modèle
- Support encore très léger (Windows et MacOS X uniquement, pas tous les navigateurs), mais développement de plugins encore en cours
- Programmation reposant sur l'architecture .NET de Microsoft

# ActiveX

- Technologie Microsoft
- Plugins du navigateur permettant une interaction très forte avec le système d'exploitation
- Pas de système de bac à sable, mais l'exécution des plugins doit être confirmée par l'utilisateur
- Uniquement sous Windows, principalement sous Internet Explorer
- Un peu passé de mode maintenant, tout comme les applets Java qu'ActiveX était censé concurrencer
- Utilisé pour Windows Update, etc.

# Canvas+JavaScript

- Balise `<canvas>` définissant une zone permettant de faire du dessin bitmap
- JavaScript permet de rendre le contenu dynamique
- Supporté dans Safari, Opera, Firefox
- Exemple : <http://www.abrahamjoffe.com.au/ben/canvascape/>

# SVG/VML+JavaScript

- Formats d'images vectorielles, manipulables en JavaScript
- Scalable Vector Graphics
- Vector Markup Language
- SVG : cf. cours sur XML, supporté (à des degrés variés) dans Firefox, Opera, Safari, Konqueror ; Adobe SVG Plugin pour Internet Explorer
- VML : Internet Explorer uniquement
- Encore un peu expérimental
- SVG+JavaScript : Peut être vu comme une alternative à Flash, dans des formats ouverts et standards
- Utilisé par exemple dans Google Maps (VML dans IE, SVG ailleurs)
- Exemple : <http://isthis4real.com/orbit.xml>

Et aussi...

**Shockwave** initialement une contrepartie à Flash dans l'offre de  
Macromedia, maintenant utilisée surtout pour des jeux  
interactifs et environnements 3D

**VRML** Images et modèles 3D

...

# Plan du cours

- 1 Introduction
- 2 AJAX
- 3 Applets Java
- 4 Adobe Flash
- 5 Alternatives
- 6 Son et vidéo sur le Web**
- 7 Application

# HTML et contenu multimédia

- HTML ne propose que la représentation de texte (structuré) et d'images ( `<img>` )
- Rien pour le son et la vidéo ! (à part un `<bgsound>` , non standard, à oublier)
- Solution standard : recourir au mécanisme général de `<object>`

## Conteneurs et codecs

**Codecs audio** MPEG-1/I, MPEG-1/II, MPEG-1/III (MP3), AAC, AC3, RealAudio. . .

Manière dont des données sonores sont compressées (avec ou sans perte) par blocs

**Codecs vidéo** MPEG-1, MPEG-2, H.264, Theora, RealVideo, DivX. . .

Manière dont des séquences vidéo sont compressées (avec ou sans perte) par blocs

**Conteneurs** WAV, ASF. . . (son), AVI, MPEG-2, MPEG-4, MOV, OGG, RealMedia, DivX. . . (son+vidéo)

Manière dont données sonores et séquences vidéo compressées par des codecs sont agencés dans un fichier, avec des fonctionnalités permettant l'entrelacement son/vidéo, le positionnement à un endroit aléatoire du fichier, etc.

**Beau mic-mac!** Difficile de trouver un logiciel supportant tous ces formats. . .

## Brevets logiciels et restrictions d'utilisation

- Nombreux **brevets logiciels** protégeant l'encodage ou le décodage selon tel ou tel codec (licence accordée contre redevances)
- Peu clair quels brevets sont vraiment valides ! De nombreuses entreprises prétendent avoir un brevet sur la compression/décompression MP3.
- **Note** : les brevets logiciels n'existent (normalement) dans l'UE que s'ils décrivent un **processus industriel**. Peu clair si ça s'applique à l'encodage/décodage son et vidéo.
- **Conséquence** : un logiciel libre comme vlc est peut-être illégal ! Peu de chance de voir un support vidéo et audio dans des navigateurs libres
- **Exempts** de ces problèmes :
  - ▶ Les **vieux** formats (RIFF, MPEG-1...)
  - ▶ Les formats **conçus comme libres** (Ogg, Vorbis, Theora); mais certaines entreprises ont peur que ces formats violent quand même des brevets existants non encore connus !

## Alternatives pour son et vidéo sur le Web (1/2)

- Balise `<embed>` : non standard, vieillerie, à oublier
- Utilisation d'un plugin spécifique à un type de média donné (ne marche pas toujours avec IE, nécessite le plugin !)

```
<object type="video/quicktime" data="test.mov"
  width="320" height="240">
  <a href="test.mov">test.mov</a>
</object>
```

- Utilisation d'un contrôle ActiveX spécifique (IE seulement)

```
<object classid="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"
  width="320" height="240">
  <param name="src" value="test.mov" >
  <a href="test.mov">test.mov</a>
</object>
```

## Alternatives pour son et vidéo sur le Web (1/2)

- Simple lien vers le contenu (fonctionne si un logiciel correspondant est installé, mais **pas d'intégration au navigateur**)

```
<a href="test.mov">test.mov</a>
```

- Balises `<audio>` et `<video>` (HTML 5, **non encore implémentées**, avenir incertain)

```
<video src="test.mov" width="320" height="240">  
  <a href="test.mov">test.mov</a>  
</video>
```

- Intégré à une application Flash ; fonctionne... si Flash fonctionne (**moins mauvaise solution actuellement**)

# Plan du cours

- 1 Introduction
- 2 AJAX
- 3 Applets Java
- 4 Adobe Flash
- 5 Alternatives
- 6 Son et vidéo sur le Web
- 7 Application**

# Application

Reproduire le formulaire avec auto-complétion depuis le site Web du cours.