

AJAX

Etat de l'art d'Interface Web Riche

Open Source

Baris Ulucinar

baris.ulucinar@unifr.ch

Octobre 2006



Université de Fribourg, Suisse
Sous la direction de Prof. O. Abou Khaled

Table des matières

1. Introduction.....	3
2. RIA : Riche Internet Application.....	3
3. Pourquoi avons-nous besoin d'Ajax ?.....	3
3.1 Les principes d'Ajax.....	4
3.2 Web synchrone / asynchrone.....	4
3.3 Implémentations avec Ajax.....	4
3.3.1 L'objet XMLHttpRequest.....	4
3.3.2 Mini Exemple.....	5
3.4 Ajax avantages, désavantages.....	6
3.4.1 Avantages.....	6
3.4.2 Problèmes et solutions.....	6
3.5 Frameworks.....	8
3.5.1 Cross-browser libraries	9
3.5.1.1 Prototype.....	9
3.5.1.2 Sarissa.....	9
3.5.2 Widgets and widget suites	9
3.5.2.1 Script.aculo.us.....	10
3.5.2.2 Open Rico.....	10
3.5.3 Application frameworks	10
3.5.3.1 DWR: Easy AJAX for JAVA.....	10
3.5.3.2 Sajax.....	11
3.5.3.3 Xajax.....	11
3.5.3.4 Echo2.....	11
3.5.3.5 Ruby on Rails.....	12
3.5.3.6 Ajax Tags Library.....	12
3.5.3.7 Swato	12
3.5.3.8 WebWork.....	13
3.5.3.9 Dojo.....	13
3.6 Fonctionnalités et utilisations communes d'AJAX.....	13
3.6.1 Widgets.....	13
3.6.2 Architecture de la page.....	14
3.6.3 Effets visuels.....	14
4. Quelques Applications et Sites Web	15
4.1 Portails Web et Pages Personnalisables.....	15
4.1.2 Todo's.....	15
4.1.3 Gestionnaires de Musiques.....	16
4.1.4 Gestionnaires d'Images.....	16
4.2 Web Applications.....	16
4.2.1 Word Processing.....	16
4.3 Autres.....	16
5. Conclusion.....	17
6. Références.....	18

1. Introduction

Ajax ou Asynchronous Javascript And XML est un acronyme apparu en 2005 dans un article publié par Jesse James Garrett sur le site AdaptivePath. Ajax n'est pas une nouvelle technologie, mais une technique qui combine un ensemble de techniques et technologies déjà existantes connues qui se complètent. C'est donc un mélange d'un certain nombre de technologies standard déjà familières : *Javascript* et *Dom* permettent d'ajouter du scripting dynamique, d'afficher, interagir avec les données, *XHTML* : structures sémantiques des données, *CSS* : présentation des données, *XML*, *XSL* et l'objet *XMLHttpRequest* pour échanger et manipuler les informations avec le serveur d'une manière asynchrone. Ajax est une approche au développement d'interface Web riches. Plupart des sites utilisent Ajax, sans le savoir. Microsoft Outlook Web Access utilise cet ensemble de technologie depuis 1998, mais AJAX est venu sur les feux de la rampe dès la sortie de Google Suggest et Google Maps en 2005.

2. RIA : Riche Internet Application,

Les Riche Internet Applications sont des applications Web qui partagent les caractéristiques des applications autonomes sur l'ordinateur¹. Dans les applications classiques seulement le Serveur et le Client existent : les traitements sur le premier et le second juste pour la représentation de l'information. Avec les RIA, une partie du traitement de l'information se fait dans le côté du Client, donc localement. Cela permet d'avoir des interfaces clients riches, et ne pas avoir les côtés négatifs d'applications réels : pas d'installation nécessaire, peut être utilisé partout (besoin d'une connexion Internet). Voyons maintenant plus en détail une des méthodes pour arriver à une applications Internet riche : AJAX.

3. Pourquoi avons-nous besoin d'Ajax ?

Avec Ajax, les pages Internet ressemble de plus en plus à des applications autonomes de l'ordinateur personnel. Car Ajax permet aux applications de travailler en arrière fond et de montrer les données demandées si nécessaire. Puisque Ajax permet d'avoir des RIA, cela amène une meilleure ergonomie pour les utilisateurs, qui ne se sentiront pas dépaycé. Nous pouvons voir cela avec les webmails qui remplace très bien les applications standard tels que Mozilla Thunderbird, ou Microsoft Outlook Express. Puisqu'il n'y a rien à installer, et que la plupart des navigateurs acceptent Javascript, cela permet d'avoir une maintenance centralisée

¹ http://fr.wikipedia.org/wiki/Rich_Internet_Application

sans un trop grand effort et créer des applications, satisfaisant les buts de productivité, de mise en réseau. De plus l'application ne sera pas bloquée par un pare-feu.

3.1 Les principes d'Ajax

Il existe trois principes d'Ajax, d'abord, premier point : le navigateur Web (browser) doit accueillir l'application, et pas le contenu, car deuxième point : le contenu est livré par le serveur et troisième point : l'utilisateur interagit avec l'application, les requêtes se font de manières implicites, en arrière plan sans perturber l'utilisation d'autres éléments de la page.

3.2 Web synchrone / asynchrone

Avant l'utilisation de l'objet XMLHttpRequest, la communication entre le navigateur Web (formulaire par exemple) et le serveur se faisait de manière synchrone. C'est-à-dire qu'il fallait attendre chaque fois le rechargement entier de toute la page. Le contenu était donc statique et se mettait à jour lors du rechargement de la page.

Avec l'utilisation de cet objet, le navigateur peut lancer plusieurs requêtes et pendant ce temps l'utilisateur peut utiliser d'autres éléments de la page, il ne sera donc pas frustré. Les requêtes sont envoyées d'une façon asynchrone.

3.3 Implémentations avec Ajax

Il faut d'abord choisir une plateforme de programmation : Java, Php, Ruby on Rails, pour subvenir aux besoins des requêtes dans le serveur. Dans le côté client, il faut pouvoir envoyer des requêtes de types XMLHttpRequest et pouvoir « parser » les réponses XML du serveur. L'implémentation de XMLHttpRequest et le parseur XML existent dans tous les navigateurs récents, intégré ou avec bibliothèques spéciales (Microsoft Internet Explorer).

3.3.1 L'objet XmlHttpRequest

Pour utiliser Ajax, il faut donc créer en « Javascript » l'objet « XmlHttpRequest » (qui possède son propre API²) qui va permettre de faire des requêtes « Http » pour échanger du « Xml ». Il faut tenir compte dans la création et initialisation de l'objet de tester le navigateur Web du client. Selon que c'est Microsoft Internet Explorer ou les autres : Mozilla Firefox, Safari, Opera. Pour tous les navigateurs sauf MSIE il faut instancier l'objet par

² Ajax Hacks, mars 2006, Bruce W. Perry, O'Reilly.

```
var requester = new XMLHttpRequest();
et pour MSIE de la façon suivante :
var requester = new ActiveXObject("Microsoft.XMLHTTP");
```

3.3.2 Mini Exemple

Le mini exemple a pour but d'aller chercher un fichier texte qu'il affiche de manière asynchrone. Il se compose de 3 fichiers : « mini.html » est une page Web qui va montrer les résultats, et « search.txt », « perso.txt » qui feront état de catégorie de liens.

3.3.2.1 mini.html

```
<html>
<head><title>Mini Exemple</title>
<script language="javascript">
    var XMLHttpRequestObject = false;

    if(window.XMLHttpRequest) {
/* creation de l'objet pour les navigateurs:
safari - netscape - firefox - opera */
        XMLHttpRequestObject = new XMLHttpRequest();
    }else if(window.ActiveXObject) {
/* pour internet explorer */
        XMLHttpRequestObject = new ActiveXObject("Microsoft.XMLHTTP");
    }

    function getData(dataSource, divID){
        if(XMLHttpRequestObject) {
            var obj = document.getElementById(divID);
            XMLHttpRequestObject.open("GET", dataSource);
/* onreadystatechange: detient le nom du event handler */
            XMLHttpRequestObject.onreadystatechange = function() {
/* readyState: detient L-etat de la requete
4 = complete */
/* status: detient le code de statut du HTTP retourne
par la requete 200 = OK */
/* pour controler que tout s'est bien passe */
if (XMLHttpRequestObject.readyState == 4 && XMLHttpRequestObject.status ==
200) {
                obj.innerHTML = XMLHttpRequestObject.responseText;
            }
            XMLHttpRequestObject.send(null);
        }
    }
</script>
</head>
<body>
<!-- ici on appelle la fonction qui va chercher le fichier de categories et
qui va l-afficher dans l-element dont l-id est targetDiv -->
<a href="#" onclick="getData('search.txt', 'targetDiv')">Search</a><br>
<a href="#" onclick="getData('perso.txt', 'targetDiv')">Perso</a>
<!-- où les informations seront affichees -->
<div id="targetDiv" style="border:1px solid blue;height:80px">Chosissez une
categorie</div>
</body>
```

</html>

3.3.2.2 *search.txt*

```
<p>  
<a href="http://google.ch">Google</a><br>  
<a href="http://search.ch">Search.ch</a>  
</p>
```

3.3.2.3 *perso.txt*

```
<p>  
<a href="http://baris.ulucinar.org">Perso</a><br>  
<a href="http://ulucinar.org">Site village</a>  
</p>
```

3.4 Ajax avantages, désavantages

3.4.1 *Avantages*

Lorsque AJAX est utilisé, la majeure partie de l'application se trouve dans le browser, c'est-à-dire dans le côté client. Les données sont envoyées par le serveur. Son utilisation permet de créer des applications Web se rapprochant de plus en plus d'applications de bureaux standard. Cela permet une réduction des coûts [1]. Les IFrames n'ont plus besoin d'être, si l'application est bien implémentée, cela permet de réduire le temps du premier chargement [4]. Ajax permet de la Validation, et amène plus de possibilités de contenu aux pages jusque là sans vie : intégration avec Google Maps, de client chat, lecteur multimédia, de lecteur de flux Rss. Il permet de faire une personnalisation complète de la page, qui amène une meilleure interaction, ergonomie, et fidélise les visiteurs. Le visiteur se croira chez lui. Il existe différents frameworks demandant une base solide de Javascript, ou pas.

3.4.2 *Problèmes et solutions*

Il existe différents problèmes liés à l'utilisation d'AJAX : d'abord il y a les gros problèmes des marque-pages, qui ne marche pas car l'utilisateur a sous les yeux toujours la même page. L'utilisation abusive d'Ajx amène une surcharge du serveur, à cause par exemple de grand nombre de requêtes. Une autre source de problème est le manque de Back Button, qui est très utilisé par les internautes, ce bouton utilise l'objet history du navigateur, pour revenir en arrière. Puisque avec l'utilisation de Ajax il n'y a qu'une seule page, alors l'objet ne peut revenir en arrière. Pour remédier à cela il faut créer son propre Javascript Back Button.

Puisque les traitements et autres opérations se passent en arrière plan, il se peut que l'utilisateur s'impatiente et risque de quitter la page. Alors il faut mettre des animations visuelles, ou changement de couleurs signifiant par là qu'il y a quelque chose qui se passe, ou que un contenu spécifique a changé. Un bon exemple est le bouton de chargement en haut à droite dans Gmail.

Si mal programmé, l'application bombarde le Serveur de requête, et envoie des informations parfois sensibles sans demander l'avis de l'utilisateur. En outre il faut veiller à donner des moyens pour que la personne puisse corriger ses erreurs.

Il faut tenir compte des différents navigateurs, puisque les méthodes et variables change souvent suivant le navigateur: la « Figure 1 : Navigateurs » montre le partage des différents navigateurs pour le site : <http://ulucinar.org>, le graphique est fournit par sitemeter.com; Mozilla Firefox 14%, Internet Explorer : 83%, les autres navigateurs rapporte 1% chacun.

Les grandes applications font ralentir l'ordinateur (grand usage du CPU, mémoire), c'est pourquoi, il faut utiliser Ajax si besoin, pas par un effet de mode.

Un autre grand problème est que Javascript doit être activé chez le client. Si cela n'est pas le cas, il faut avoir un plan de secours, car un nombre non négligeable d'utilisateur n'ont pas Javascript ou l'on simplement désactivé pour des raisons de sécurités. Javascript n'est pas utilisé par 4-13% des utilisateurs. La « Figure 2 : JavaScript » montre que différents versions sont utilisés, en outre chez 1% des personnes il est désactivé.

Il n'existe pas encore de conventions d'utilisation pour les interfaces utilisateurs. Les utilisateurs ne savent pas au premier abord (il n'est pas évident) qu'ils peuvent prendre un objet le mettre là, le faire disparaître, etc.

Une autre complication peut être que le contenu ne sera pas accessible par les robots des moteurs de recherche : il faut utiliser les tags <meta> pour informer les robots.

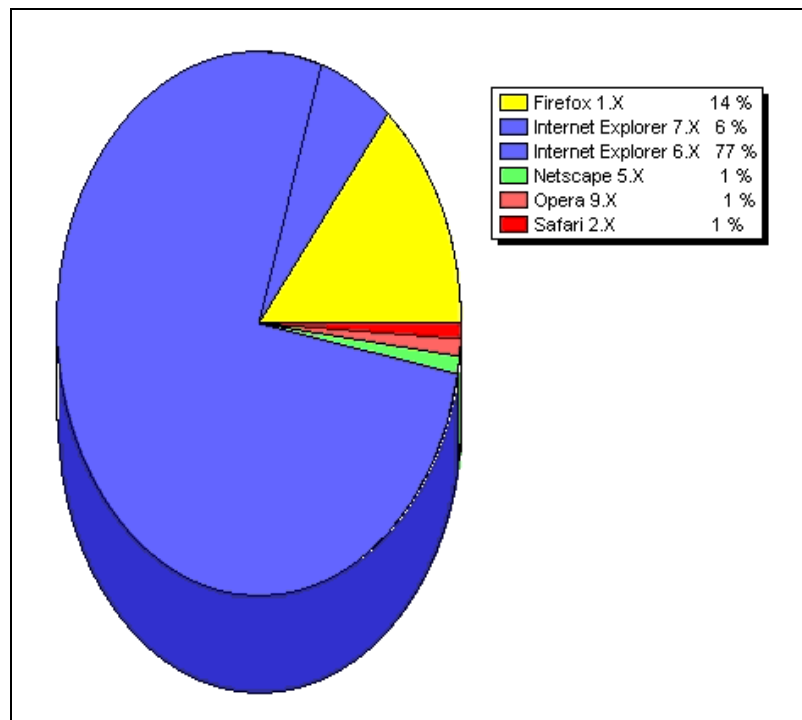


Figure 1 : Navigateurs

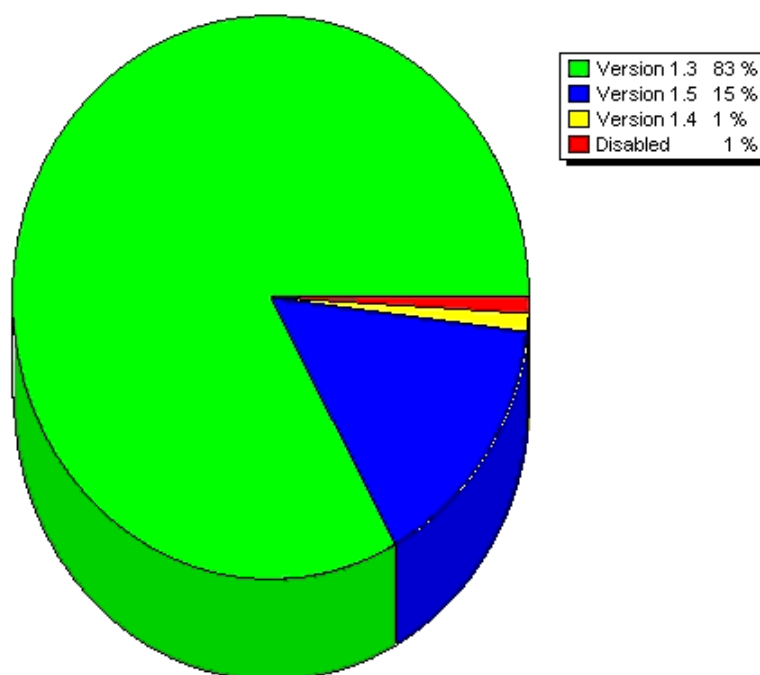


Figure 2 : JavaScript

3.5 Frameworks

Il existe de nombreux framework sur le marché, mais ce papier s'intéresse aux frameworks Open Source. En générale, Javascript est utilisé chez le côté client, et PHP ou JSP du côté

serveur. Il existe 3 types de librairies : la première est une librairie graphique, amenant des jolies effets « woaaaa ! » visuels, et le second est plutôt généraliste. C'est-à-dire qu'il permet de simplifier le développement et l'utilisation de l'objet XMLHttpRequest, et le dernier est un framework d'application.

Il en existe 3 types, « *Cross-browser libraries* » est une librairie généraliste (indépendant du navigateur) qui permet de simplifier l'utilisation de l'objet XMLHttpRequest, le second : « *Widgets and widget suites* », est un ensemble collections d'effets visuels et de widgets. Le dernier « *Application frameworks* » est un framework complet pour créer des applications Web riches.

3.5.1 Cross-browser libraries

Les “Cross-browser librairies” se composent généralement d'un voir deux-trois petits fichiers, qui regroupent des méthodes très utiles pour aider au développement de la page sans tenir compte du navigateur utilisé tels.

3.5.1.1 Prototype

Prototype³ est une librairie open source très connue, qui donne aux développeurs des outils pour créer des applications Ajax. Il est souvent la librairie de base pour d'autres projets open source : script.aculo.us, rico, ruby on rails, etc.

3.5.1.2 Sarissa

Sarissa⁴ est une librairie multi navigateur pour API XML. Il permet la création et manipulation de documents XML, chargement de fichier XML depuis un URL, ou chaîne de caractère, des transformations XSLT (html,etc.), des requêtes XPath pour extraire des données. Sarissa utilise l'objet XMLHttpRequest pour télécharger du XML en utilisant Ajax, il est très utile car il permet de prendre en charge facilement le fichier XML téléchargé.

3.5.2 Widgets and widget suites

Ce type est utilisé pour la créer plus facilement des interfaces utilisateur, car il a un ensemble de widget à disposition: bouton, effets, sliders, etc.

³ <http://prototype.conio.net/>

⁴ <http://sarissa.sourceforge.net/projects/sarissa>

3.5.2.1 Script.aculo.us

Script.aculo.us⁵ permet la création d'interface utilisateur riche pour page Web 2.0 ou application, par l'utilisation de superbes effets et animations. Il possède des animations de base, qui peuvent être combinés pour rendre des effets plus complexes. Le Drag-and-Drop, les Slider, autocompletion, in place editing, sont les quelques propriétés offertes.

3.5.2.2 Open Rico

Open Rico⁶ est un librairie JavaScript open source pour créer des applications Web riches. Se base sur Prototype, fournit tous les fonctionnalités d'Ajax, la gestion par Drag-and-Drop, un librairie d'effets visuelles et des comportements (behaviors : effet accordion, livegrid). Il est composé de 3 fichiers : Prototype.js, Rico.js et Util.js.

3.5.3 *Application frameworks*

Ceux sont des frameworks complets qui permettent de faire des applications Web riches professionnelles.

3.5.3.1 DWR: Easy AJAX for JAVA

Pour travailler avec Ajax, sans toucher à l'objet XMLHttpRequest. DWR est un librairie open source (direct web remoting), avec lequel nous pouvons créer des sites Ajax. Il permet d'avoir une couche supplémentaire au dessus de l'objet pour n'avoir pas le programmer directement l'objet de requête. Avec DWR, les développeurs peuvent créer des classes Java, et utiliser les objets se trouvant dans le côté Serveur (Java Servlet) avec du code Javascript se trouvant dans le côté Client (Javascript/Html, cf. Figure 3 : DWR) pour envoyer des requêtes, pour donc mettre à jour dynamiquement la page.⁷ Donc il existe deux parties : le code dans le serveur pour les traitements, opérations divers et le code dans le navigateur pour représenter les données plus facilement.

⁵ <http://wiki.script.aculo.us/scriptaculous/show/CombinationEffectsDemo>

⁶ <http://openrico.org/rico/home.page>, <http://openrico.org/rico/demos.page>

⁷ <http://getahead.ltd.uk/dwr/overview/dwr>

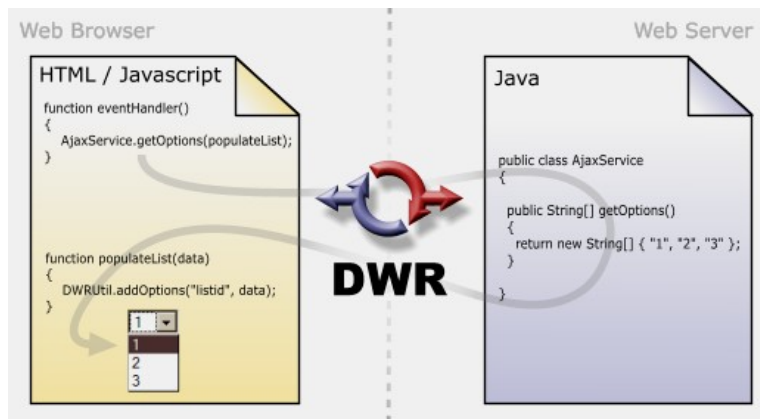


Figure 3 : DWR

DWR génère dynamiquement le code javascript du code java, en fait l'exécution se passe dans le Serveur, et seulement les données sont envoyées reçues par lui-même.

DWR est contenu dans un java archive se trouvant : <http://getahead.ltd.uk/dwr/>.

Puisqu'il utilise AJAX, les développeurs ont accès à la puissance de Java, car Java est beaucoup plus puissant que Javascript. DWR permet de faire plus que le PHP. Le seul inconvénient, c'est qu'il faut avoir un « java-based » serveur comme par exemple : Apache Tomcat.

3.5.3.2 Sajax

Sajax permet de créer du Javascript AJAX dans le navigateur avec l'utilisation de différents langages : ASP, Perl, Python, PHP, etc. Par exemple : une méthode Javascript dans une page Web connectée à une méthode PHP, se trouvant dans le serveur, qui traite les données et les envoie à une autre méthode Javascript pour la représentation. Lorsque l'utilisateur accède à la page, Sajax génère tout le code Javascript pour traiter les opérations AJAX.

3.5.3.3 Xajax

Xajax est une librairie de classe PHP open source qui permet de créer facilement des applications AJAX en utilisant HTML, CSS, JavaScript et PHP. L'objet `$xajax` génère du code Javascript.

3.5.3.4 Echo2

Echo2⁸ permet la construction d'applications Web, et est la nouvelle génération du Framework Echo Web, se rapprochant le plus d'applications classiques de l'ordinateur

⁸ Echo2, <http://www.nextapp.com/platform/echo2/echo/>, demo: <http://demo.nextapp.com/Demo/app>

personnel. Il faut en outre même pas avoir des connaissances du HTML, http ou Javascript. Les applications sont créées par l'utilisation de librairie open source et de l'IDE EchoStudio2. L'EchoStudio2 est un outil de développement d'application rapide basé sur Eclipse⁹ 3.1. Cet outil n'est malheureusement pas gratuit.

3.5.3.5 Ruby on Rails

Ruby on Rails¹⁰ est un framework permettant le développement d'applications Web écrit en Ruby. Rails utilise Ajax pour réaliser des pages dynamiques utilisant Javascript et XML pour envoyer des requêtes au serveur sans recharger la page. ROR supporte AJAX et simplifie son utilisation en offrant différentes méthodes.

3.5.3.6 Ajax Tags Library

Le code Javascript est construit par l'utilisation de tags JSP dans le serveur. De plus la création de tag personnel est possible. Quelques tags déjà existants sont (les tester : <http://ajaxtags.no-ip.info/>):

Autocomplete : cherche une liste d'item possible qui convient à ce que l'utilisateur est en train d'entrer.

Callout : permet d'avoir des pop up ballons, pour montrer les informations de manière conviviale.

Select/drop down : rend le contenu d'un contrôle drop down à un contrôle select.

Toggle : permet de changer des images de deux trois sources, le plus utilisé pour donner des scores.

Form Update: mise à jour de champ texte en fonction de ce que l'utilisateur rentre dans un autre champ.

HtmlContent : remplit le contenu d'un tag Div, span en utilisant une autre source.

Tab Panel : permet d'avoir des tabs.

3.5.3.7 Swato

Swato¹¹ est un framework open source pour développer plus facilement des applications Web utilisant AJAX. Des Servlets sont utilisés pour le côté serveur, une librairie Javascript basée

⁹ Eclipse, <http://www.eclipse.org/>

¹⁰ ROR, <http://www.rubyonrails.org/>

¹¹ Swato, <https://swato.dev.java.net/>

sur prototype¹². JSon¹³ peut être utilisé pour rassembler les données POJO¹⁴ dans le serveur. De plus il fournit une interface simple en Javascript pour l'interaction avec des POJO distants. La configuration se fait d'une manière simple grâce l'utilisation de <servlet>, <filter> dans le fichier de configuration <web.xml>¹⁵, en outre il faut aussi parler de l'intégration Spring¹⁶. Il existe quelques composants déjà existants et réutilisable : Auto-suggest Textbox, JS Template, JS Logge, cela permet d'aider au développement rapide de site.

3.5.3.8 WebWork

WebWork¹⁷ s'est spécialisé pour la création d'application Web riche professionnelle. Avec l'utilisation de WebWork Ajax peut être intégré avec d'autres technologies tels que Flash. Dans le côté serveur l'utilisation de .Net et Java est possible.

3.5.3.9 Dojo

Dojo¹⁸ est un ensemble de librairies open source (nWidgets, Burstlib, f(m)) permettant de développer des applications Web professionnelles mieux, plus facilement et plus rapidement. En outre les pages créées seront plus utilisables, réactives et fonctionnelles en utilisant des widgets, animations, etc. L'importance est donnée à l'interactivité.

3.6 Fonctionnalités et utilisations communes d'AJAX

3.6.1 Widgets

Les Widgets peuvent être selon wikipédia, un élément de base d'interface utilisateur ou un petit outil permettant d'obtenir des informations. Dans les applications riches utilisant Ajax, nous pouvons avoir des « *Sliders* » pour que l'utilisateur puissent indiquer une valeur dans une certaine domaine, un « *Progress Indicator* » pour lui indiquer qu'il y a un traitement en cours. Avec le « *Drill-Down* » l'utilisateur pourra visiter les sous menu dans un menu. Le « *Data Grid* » permet d'explorer les données, qui sont représentées sous forme de table, avec laquelle un utilisateur peut interagir par filtrage. Le « *Rich Text Editor* » permet aux utilisateurs de rentrer du contenu texte riche. Le widget « *Suggestion* » suggère des mots à

¹² Prototype, <http://prototype.conio.net>

¹³ JavaScript Object Notation : utilise la notation des objects JavaScript pour transmettre de l'information structurée, <http://fr.wikipedia.org/wiki/JSON>

¹⁴ Plain Old Java Object, <http://fr.wikipedia.org/wiki/POJO>

¹⁵ web.xml Deployment Descriptor Elements, http://e-docs.bea.com/wls/docs81/webapp/web_xml.html

¹⁶ Spring, "the leading full-stack Java/J2EE application framework", <http://www.springframework.org/>

¹⁷ WebWork, <http://www.opensymphony.com/webwork/>

¹⁸ Dojo Toolkit, <http://dojotoolkit.org/>

l'utilisateur pour lui venir en aide. Par exemple lorsque l'utilisateur tape « A » le widget pourrait suggérer Armand, Ali, Aslan, pour nom propre, etc. Le « *Live Search* », fait une recherche dynamique, c'est-à-dire que les résultats sont toujours rechargés et mise à jour selon les données supplémentaire que l'utilisateur rentre dans un champ texte. Le « *Live Form* » comme son nom l'indique est une sorte de formulaire dynamique qui ajoute ou enlève des champs du formulaire par rapport à l'information rentrée, avec ce widget, les données peuvent être validées directement chez le client pendant qu'il les tape comme le courriel, noms utilisateur de login unique, l'existence ou non de date, de url. Les « *Layout Widgets* » représentent l'information sous forme de menu, tab, arbre, animations accordions, et fenêtres. « *Slide Show* » comme son nom l'indique fait un cycle à travers les contenus, plus souvent des images.

3.6.2 Architecture de la page

La page peut contenir différent styles d'interaction et stratégie pour structurer son contenu. Le « *Drag-and-Drop* » est un moyen d'interaction par lequel, un utilisateur peut prendre un élément et le laisser dans un autre endroit de la page. Avec le « *Sprite* » le contenu visuel sera flexible, par exemple dans une application pour rechercher des appartements à louer, selon les données récoltées par un formulaire sur le vif, une carte pourrait afficher les appartements convenants à ces données. Le « *Popup* » permet de représenter de l'information supplémentaire. Le « *Malleable Content* » permet de rendre les éléments de la page malléable, par exemple le visiteur peut se faire une page sur mesure. Le « *Microlink* » représente l'information sans gêner les autres éléments composant la page, pour naviguer rapidement dans le contenu. Les « *Portlets* » sont des mini-applications.

3.6.3 Effets visuels

Les effets en plus d'être « jolies à voir », permettent d'attirer l'attention de l'utilisateur et l'aider à comprendre que fait l'application. Les différents effets ont pour effets d'animer la page, par des effets graphiques tels que dégradation, et animations divers. L'effet « *Spotlight* » permet d'attirer l'attention du client vers un élément, partie de la page. « *Mutation* » est un effet qui fait par exemple changer la grandeur de l'objet, ceci peut être utilisée dans des albums photos pour gagner de l'espace et attirer l'attention de l'utilisateur. L'effet « *Highlight* » peut montrer que des éléments sont sélectionnés. L'effet « *Fade* » est le plus souvent utilisés dans les albums, pour animer les images et changer leur opacité, des animations divers existent tels que « *Pulse* » pour redimensionner automatiquement un objet

plus petits, « *Slide* » pour bouger un contenu, « *Explode/Inplode* » pour faire apparaître disparaître du contenu, « *Wipe* »,

4. Quelques Applications et Sites Web

4.1 Portails Web et Pages Personnalisables

Ce sont des pages Web très ergonomiques, dans lesquelles les visiteurs peuvent configurer les éléments et interagir.

Windows Live [<http://www.live.com/>] est un page personnalisable et un moteur de recherche en un. En cliquant à un lien un haut à droite, l'utilisateur peut ajouter des pages à ajouter : Accueil, Actualités, Sports, Divertissements. Offrant en outre la météo, le contrôle de la boîte de réception et divers informations. Autres page de même genre : [<http://www.google.com/ig>], [<http://www.netvibes.com/>], [<http://www.start.com/>], [<http://metasearch.franceevasion.com/my/>]

Protopage [<http://protopage.com/v2>] est un page de démarrage personnel gratuit, qui peut contenir : des News, Sticky Notes, todo, signets, namingit(..baris), partager une partie de la page, des images. Protopage est hautement interactif, l'utilisateur peut l'éditer à son souhait, et ajouter des panels, et des pages supplémentaires.

Google Page Creator [<http://pages.google.com/>] permet de créer une page personnelle, il faut pour cela, avoir un compte gmail. L'utilisateur peut utiliser une interface pour créer des pages et les éditer directement pour ensuite les publier.

Google suggest [<http://www.google.com/webhp?complete=1&hl=en>] est un moteur de recherche qui suggère une liste de mots possible pour aider l'utilisateur.

4.1.2 Todo's

Tadalist [<http://www.tadalist.com/>] permet la création de to-do listes personnelle, l'utilisateur doit s'enregistrer pour pouvoir créer des listes, et les choses à faire. L'utilisateur pourra atteindre son liste par le lien [<http://nomUtilisateur.tadalist.com/>]. Sites de même genre : [<http://voo2do.com/>], [<http://www.rememberthemilk.com/>].

4.1.3 Gestionnaires de Musiques

Zicabox [<http://www.zicabox.com/>] permet le partage et l'écoute de musique. Car la musique personnelle doit être accessible de partout.

Jamendo [<http://www.jamendo.com/fr/>] propose un nouveau stratégie pour faire connaître, diffuser et rémunérer les artistes pour leur travail. Ce sont les artistes qui diffusent leur travail en acceptant une licence de diffusion.

4.1.4 Gestionnaires d'Images

Flickr [<http://flickr.com/>] est un gestionnaire très connu et utilisé qui permet de partager, de rechercher des images. Autres sites : [<http://www.zoto.com/>], [<http://www.zimagez.com/>].

4.2 Web Applications

Des sites plus sophistiqués sont à proprement parler des applications riches, qui n'ont rien à envier aux applications de bureau.

4.2.1 Word Processing

Avec Docs & Spreadsheets de Google [<http://docs.google.com/>] l'utilisateur peut créer des documents complets qu'il peut ensuite sauver sous format divers : doc, rtf, pdf, html et odt.

Ajax Write [<http://www.ajax13.com/fr/ajaxwrite/>] est un traitement de texte sur le Web complet, le seul inconvénient est qu'il requiert Mozilla Firefox v1.5 (ou plus).

Ajax XLS: <http://www.ajax13.com/fr/ajaxxls/index.html> est un tableur.

Ajax Scetch [<http://www.ajax13.com/fr/ajaxsketch/index.html>] est un éditeur graphique.

Autres éditeurs: [<http://zohowriter.com/Home.do>], [<http://numsum.com/>]

4.3 Autres

Meebo [<http://www.meebo.com/>] est client de chat Web IM, donne la possibilité de se connecter à son compte MSN, Yahoo Messenger, ICQ, Gtalk.

Google maps [<http://maps.google.com/>] est une page permettant de consulter une carte du monde.

Gmail [<http://gmail.com>] est un webmail très utilisés qui remplace très bien les applications mail de bureau.

Ajax Tunes [<http://www.ajax13.com/fr/ajaxtunes/index.html>] permet d'écouter de la musique.

Buttonator [<http://www.buttonator.com/>] permet la création de bouton Web personnalisé.

Map Chat
[<http://www.themidnightcoders.net/examples/messageserver/chat/mapchatajax.htm>] permet le chat en utilisant le Google Maps.

Ajax OS [<http://www.ajax13.com/fr/ajaxos/index.html>] est une sorte un système d'exploitation (visuellement). Les utilisateurs sont dans un bureau avec différents outils. Un autre site de même genre est You OS [<https://www.youos.com/>].

5. Conclusion

Ajax fut un acronyme créé par Garrett en 2005, mais pour autant il existait depuis longtemps, et était utilisé par de nombreux développeurs. Ajax ouvre une nouvelle façon d'appréhender Internet, non plus comme représentation de l'information mais une nouvelle domaine de programmation. Où nous pourrions créer des applications Web riches, qui déjà commencent à remplacer les applications de bureau standard, même si elles ont de gros handicaps comme le temps de réaction. Ajax est une domaine où les bibliothèques pullulent, de simple généralistes qui sont simples : un voire deux trois fichiers de quelques kilo octets. Vers des plateformes de développement complexe qui aident le développeur, même au-delà, car il en existe même où il ne faut même pas de connaissance de Javascript ou de Html. Pourtant il existe un gros problème : les bibliothèques sont souvent mal documentées et le développeur doit être très doué en Javascript. Cependant il est très important de préciser qu'une page utilisant Ajax, n'a pas à installer de plugin, Javascript est accepté par le plupart des navigateurs. Ajax permet une maintenance centralisée des applications sans un grand effort. En l'utilisant le développeur peut créer des applications, satisfaisant les buts de productivité, de mise en réseau.

6. Références

- [1] A new approach to Wep Applications, Jesse James Garrett,
<http://www.adaptivepath.com/publications/essays/archives/000385.php>
- [2] Dépoussiérez vos applications Web! (2006), Omar Abou Khaled, Chanton, Revertera, Vonlanthen, Bapst.
http://sawww.epfl.ch/SIC/SA/SPIP/Publications/article.php3?id_article=1058
- [3] Allez plus loin avec AJAX et XMLHttpRequests, (2006), Siddh,
<http://siddh.developpez.com/articles/ajax/>
- [4] http://www.dhtmlnirvana.com/downloads/ajax_tutorial1.zip, Edward Traversa
- [5] Asynchronous Javascript and XML, Wikipedia.org,
http://fr.wikipedia.org/wiki/Asynchronous_JavaScript_And_XML
- [6] *Ajax For Dummies*, (2006), Steve Holzner, *Wiley*
- [7] *Ajax Design Patterns*, (2006), Michael Mahemoff, *O'Reilly*
- [8] *Ajax in Action*, (2006), Dave Crane, Eric Pascarello, Darren James, *Manning*
- [9] Ajax Mistakes, (2005) A. Bosworth,
http://sourcelabs.com/ajb/archives/2005/05/ajax_mistakes.html
- [10] OpenAjax Alliance White Paper, (2006),
<http://www.openajax.org/OpenAjax%20Alliance%20White%20Paper.pdf>
- [11] *Ajax Hacks*, (2006), Bruce W. Perry, *O'Reilly*.