

Rapport de Stage

Centre de Données Astronomique de Strasbourg

28 juin 2007

pour l'obtention du

Master II de l'Université Henri Poincaré – Nancy I
(Spécialité Ingénierie du logiciel)

par

Ammari Fethi

Composition du jury

M. Nacer Boudjlida Professeur responsable
M. Pascal Urso
M . André Schaaff Tuteur du stage

Table des matières

I	Présentation du Laboratoire	3
1	Présentation du Laboratoire	4
1.1	Structure Juridique	4
1.2	Présentation Générale	4
1.2.1	Haute énergie	4
1.2.2	Etoiles et systèmes stellaires	4
1.2.3	Galaxies	5
1.2.4	Le Centre de Données de Strasbourg (CDS)	5
1.3	Organisation du personnel	5
1.4	Centre de données astronomique de Strasbourg (CDS)	5
1.5	L'Observatoire Virtuel	6
1.6	Les Services du CDS	6
1.6.1	Simbad	6
1.6.2	VizieR	7
1.6.3	Aladin	7
1.6.4	Les autres services	7
II	Analyse de la situation	9
2	Analyse de la situation	10
2.1	Analyse de l'existant	10
2.2	Déroulement du stage	10
III	Etat de l'art de la technolgie AJAX	12
3	Etat de l'art de la technolgie AJAX	13
3.1	Introduction	13
3.2	Qu'est-ce que AJAX?	13
3.3	Pourquoi AJAX?	14

3.4	Les manques dans les applications Web actuelles	14
3.5	Petit historique du web	14
3.5.1	Histoire du Web	14
3.6	Buts d'AJAX	14
3.7	Trois principes d'Ajax	14
3.8	Les éléments clés d'AJAX	15
3.8.1	Javascript	15
3.8.2	Cascading Style Sheets (CSS)	15
3.8.3	Document Object Model (DOM)	15
3.8.4	L'objet XMLHttpRequest	15
3.9	Clients riche Ajax dans le monde réel	15
IV	Etude d'AJAX	16
4	Etude d'AJAX	17
4.1	Etude de la mise en place	17
4.1.1	Communication asynchrone	17
4.2	Méthode des cadres cachés	19
4.3	Méthode utilisant l'objet XMLHttpRequest	20
4.3.1	Cycle de vie de l'objet XMLHttpRequest	20
4.3.2	Fonctionnement	20
4.4	Mise en place côté serveur	22
4.5	Mise en place côté client	22
4.6	Apports d'AJAX	22
4.6.1	Les trois principes définissant le terme Ajax	22
4.7	Avantage et Désavantages d'Ajax	23
4.7.1	Ajout de nouvelles fonctionnalités	23
4.7.2	Gain de bande passante	23
4.7.3	Amélioration de l'interactivité	23
4.7.4	Ergonomie	23
4.7.5	Adaptabilité	24
4.8	Service Web	24
4.8.1	Pourquoi la page ne parlerait-elle qu'à son propre site?	24
4.8.2	Contraintes de sécurité sur le navigateur	24
4.8.3	Couche proxy sur votre serveur	24
V	Les Frameworks AJAX	26
5	Les Frameworks	27
5.1	Définition	27
5.1.1	Les différents types Frameworks	27

5.2	Avantages d'un Frameworks	27
5.2.1	Simplifier de mise en oeuvre	27
5.2.2	Maintenance	27
5.2.3	Ajout de fonctionnalité	27
5.3	Choisir une/des librairies de développement	27
5.4	Description de quelques Frameworks existants	28
5.4.1	Sondage	28
5.4.2	Les types de librairies et plateformes	28
5.5	Choix des Frameworks	31
5.5.1	Nos besoins	32
5.5.2	L'éditeur	32
5.5.3	Projet existant ou nouveau projet	32
5.5.4	Les performances	32
5.5.5	Frameworks retenus	33
VI	Conception	34
6	Conception	35
6.1	Introduction	35
6.2	Application de base	35
6.2.1	Contraintes	35
6.2.2	Objectif de l'Autocomplétion	35
6.2.3	Diagramme de séquence	35
VII	Réalisation	37
7	Réalisation	38
7.1	Mise en oeuvre	38
7.1.1	Autocomplétion	38
7.2	Mise en place	39
7.2.1	La classe XMLHttpRequest	40
7.2.2	Dojo	40
7.2.3	Prototype	43
7.2.4	Script.aculo.us	43
7.2.5	Rialto	45
7.2.6	DWR	47
7.2.7	Conclusion	47

VIII	Service VizieR	48
8	Service VizieR	49
8.1	Carte VizieR Mine	49
8.1.1	VizieR	49
8.2	Etude	50
8.2.1	Systèmes de coordonnées	50
8.3	Phase d'analyse	52
8.3.1	Analyse de l'existant	53
8.4	Phase de développement	53
8.4.1	Objectifs	53
8.4.2	Phase d'implémentation	55
8.5	Test et Validation	56
IX	Conclusion	58
9	Conclusion	59
	Liste des tableaux	60
	Bibliographie	61
X	Annexes	62
10	Annexes	63
10.1	Comparatif des Frameworks	63
10.1.1	Tableau comparatif	68
10.2	Installation du Framework ATF	68
10.2.1	Outils HTML	68
10.2.2	Bibliothèque Dojo	68
10.2.3	Biblioth(é)que DWR	68
10.3	Carte VizieR Mine	74

Table des figures

1.1	Page Web permettant d'effectuer une requête sur Simbad	7
1.2	Page Web permettant d'effectuer une requête sur Vizier	7
1.3	Exemple d'utilisation d'Aladin	8
4.1	Comparaison des mécanismes d'échanges client-serveur synchrones et asynchrones	17
4.2	Le modèle traditionnel pour les applications Web (gauche) comparé au modèle Ajax (droite).	18
4.3	Technique des cadres cachés	19
4.4	Technique XMLHttpRequest	21
4.5	Proxy sur votre serveur	25
4.6	XMLHttpRequest et les services Webs	25
5.1	Résultat du sondage Ajaxian 2006 :Frameworks AJAX les plus populaires	28
5.2	Résultat du sondage Ajaxian 2006 :plateformes AJAX les plus populaires	29
6.1	Diagramme de séquence : Autocomplétion	36
7.1	AJAX ToolKit Framework	40
7.2	Application : Autocomplétion	41
7.3	Gestion de modules	42
8.1	Architecture de la base Vizier	49
8.2	Orbite circulaire de deux corps de masses différentes autour d'un barycentre.	50
8.3	Ascension droite	51
8.4	Déclinaison	51
8.5	Position d'un observateur sur la sphère terrestre	52
8.6	Système de coordonnées équatoriales	52
8.7	The Vizier Mine	53
8.8	La selection des catalogues	54
8.9	VizieR Search Pages	54
8.10	VizieR Result Pages	55
8.11	Systèmes de coordonnées céleste	56
8.12	Analyse des catalogues	57
8.13	The Vizier Mine	57
10.1	Tableau comparatif	68
10.2	Outil ATF	69
10.3	Création d'un projet Tomcat	69
10.4	Création d'une appliaction Dojo	70
10.5	Palette Dojo	70
10.6	fonctionnement de DWR	71
10.7	DWR dwr.xml en mode debugge	71
10.8	web.xml du serveur tomcat	72
10.9	DWR dwr.xml en mode debugge	72

10.10fonctions DWR	73
10.11Test de fonctions de DWR	73
10.12The Vizier Mine	74

Remerciements

Avant de présenter mon travail, il m'est particulièrement agréable d'adresser mes vifs remerciements à tous ceux qui m'ont facilité la tâche pour mener à bien mon étude.

Je désire, tout d'abord, exprimer ma très grande reconnaissance à l'égard de M. André Schaaff, M. Thomas Boch Ingénieurs de Recherche et M. Sébastien Derriere, M. Francois Ochsenbein Astronomes qui mon proposé ce sujet et qui mon diriger pendant ces 6 mois. Je n'ai cessé de profiter de ces précieux conseils et directives, je les remercie pour leur rigueur et pour le temps qu'il m'ont consacré tout au long de ce travail.

Je remercie M. Nacer Boudjlida mon tuteur pour ce stage de fin d'étude. Je lui suis très reconnaissant pour sa disponibilité, ses enrichissantes discussions, qu'il trouve ici l'expression de ma haute considération.

Mes remerciements les plus respectueux vont au membres de jury : M. Nacer Boudjlida et M. Pascal Urso.

Qu'il me soit permis d'exprimer ici, ma plus vive reconnaissance à Madame Françoise Genova, Directrice du CDS pour avoir mis à disposition tous les éléments matériels pour la réussite de mon projet et de m'avoir accueilli avec beaucoup de confiance, sans oublier tous les cadres et le personnel de l'unité. Je remercie, par la même occasion ceux qui m'ont dispensé leur enseignement, tout au long de mes années d'études et qui m'ont prodigué toutes les connaissances.

Pour terminer que tous ceux qui m'ont aidé, trouvent ici l'expression de ma plus haute considération.

Introduction

Dans le cadre de fin d'études en Master II Informatique spécialité Ingénierie du Logiciel à l'Université Henri Poincaré, j'ai effectué un stage informatique d'une durée de 6 mois au sein du Centre de Données Astronomique de Strasbourg.

Les buts d'un stage de fin d'étude sont multiples. Tout d'abord, il s'agit de confronter les connaissances acquises pendant la formation à un problème concret. Ce stage est l'occasion de travailler sur un projet qui s'intégrera dans les besoins de l'entreprise et qui apportera une valeur ajoutée. Il s'agit ensuite de travailler dans un environnement professionnel et d'avoir une meilleure idée sur les méthodes de travail et sur la gestion de projet dans une entreprise, afin de mieux se préparer au métier d'ingénieur. Enfin, un tel stage permet d'acquérir de nouvelles compétences tant sur le plan technique que sur le plan humain.

Le Centre de Données astronomique de Strasbourg (CDS) développe des services de référence à forte valeur ajoutée très utilisés par les astronomes du monde entier. Placé sous la responsabilité de l'Institut National des sciences de l'Univers (INSU) et de l'Université Louis Pasteur (ULP), il fait partie de l'Observatoire astronomique (UMR7550). Le CDS participe activement, en collaboration avec de nombreux centres de recherche internationaux, au projet d'Observatoire Virtuel Astronomique. Le but de mon stage est d'explorer les possibilités offertes par la technologie AJAX¹ pour le développement des interfaces aux principaux services en ligne du CDS (SIMBAD², VizieR³, Aladin⁴).

Ce stage a commencé le 20 Février 2007 et finira le 20 juillet 2007, et nous faisons ici le point fin juin. Dans ce rapport, je débiterai par une présentation du centre de recherche dans lequel j'ai travaillé, puis je décrirai mon stage, de son contexte à son déroulement. J'apporterai des précisions sur les notions abordées avant de détailler les outils testés. Puis nous réaliserons un prototype pour les différentes bibliothèques choisies. Nous verrons par la suite ce qu'il a été nécessaire de réaliser pour obtenir une carte VizieR Mine pratique et fonctionnelle. Et enfin, je terminerai par un bilan du stage et sa conclusion.

¹Asynchrone Javascript and XML

²<http://simbad.u-strasbg.fr/simbad/sim-fid>

³<http://vizier.u-strasbg.fr/>

⁴<http://aladin.u-strasbg.fr/aladin.gml>

Première partie

Présentation du Laboratoire

Chapitre 1

Présentation du Laboratoire

1.1 Structure Juridique

L'Observatoire de Strasbourg est une unité de Formation et de Recherche (UFR) de l'Université Louis Pasteur. Il est également une Unité Mixte de Recherche du CNRS et de l'Université Louis Pasteur (UMR 7550).

1.2 Présentation Générale

L'Observatoire est situé sur le campus de l'Esplanade, ses bâtiments font partie du campus historique de l'université de Strasbourg.

Enseignements dispensés :

- Master 2 Analyse et Traitement des Données sur les Milieux Astronomiques. ;
- DEUG Sciences et autres DEUG (enseignement d'ouverture).
- Licence de Géosciences.
- Maîtrise de Géosciences, Maîtrise de Physique, Maîtrise de Sciences Naturelles.
- Préparation au CAPES et à l'Agrégation.
- Master Applications des Technologies.
- Diffusion de la Culture.

La partie publique de l'Observatoire, le Planetarium, est destiné à la vulgarisation de l'astronomie. L'Observatoire se compose de quatre équipes de recherche :

1.2.1 Haute énergie

L'équipe Astrophysique des Hautes énergies a pour thème l'étude des astres et sites de l'univers émetteurs de photons de haute énergie. Cette thématique générale recouvre des aspects variés, comme l'étude des astres compacts en fin d'évolution, la physique de leur activité, les phénomènes de haute énergie intéressant les étoiles jeunes ou le soleil, ou l'étude de ces phénomènes à l'échelle galactique. Ces recherches se sont largement appuyées sur les données acquises par le satellite ROSAT et s'appuieront à l'avenir sur celles des satellites X de nouvelle génération, tout spécialement XMM.

1.2.2 Etoiles et systèmes stellaires

Les recherches menées par l'équipe "Etoiles et systèmes stellaires" recouvrent un domaine étendu, incluant les étoiles, les milieux interstellaires, la Galaxie et les galaxies proches de l'étoile, objet individuel,

l'intérêt s'est porté sur les groupes d'étoiles, témoins de l'évolution stellaire mais traceurs des grandes structures de la Voie Lactée.

1.2.3 Galaxies

Les activités de l'équipe sont centrées sur les problèmes de la structure du Groupe Local, de ses populations stellaires et sur la dynamique gravitationnelle. De plus, l'équipe possède un savoir-faire pour les outils statistiques d'analyse de données et sur les méthodes inverses non paramétriques. Un des objectifs consiste à combiner les informations d'évolution des populations stellaires et celles de dynamique afin de reconstituer les événements déterminants liés aux processus de formation et d'évolution galactique.

1.2.4 Le Centre de Données de Strasbourg (CDS)

L'activité de recherche du CDS est centrée sur l'étude de la dynamique galactique et des populations d'étoiles binaires, sur une participation importante à la mission HIPPARCOS de l'Agence Spatiale Européenne, ainsi que sur le développement de méthodologies nouvelles applicables à l'analyse et au traitement de données astronomiques.

1.3 Organisation du personnel

Le CDS est un laboratoire de l'Institut National des Sciences de l'Univers (INSU), rattaché au CNRS. L'Observatoire de Strasbourg est un institut de l'Université Louis Pasteur. Le personnel permanent du CDS comprend 10 chercheurs, 7 ingénieurs de recherche, 6 ingénieurs d'étude, 3 techniciens, et plusieurs collaborateurs à contrat temporaire (projet européens, etc.) et des invités.

1.4 Centre de données astronomique de Strasbourg (CDS)

J'ai effectué mon stage au sein du Centre de Données astronomiques de Strasbourg (CDS) qui est un centre de données dédié à la collection et à la distribution dans le monde entier de données astronomiques.

Le CDS héberge la base de données Simbad, la base de référence mondiale pour l'identification d'objets astronomiques. Le but du CDS est de :

- rassembler toutes informations utiles concernant les objets astronomiques, disponibles sous forme informatisée : données d'observations produites par les observatoires du monde entier, au sol ou dans l'espace ;
- mettre en valeur ces données par des évaluations et des comparaisons critiques ;
- distribuer les résultats dans la communauté astronomique ;
- conduire des recherches utilisant ces données ;

Le CDS joue, ou a joué un rôle dans d'importantes missions astronomiques spatiales : contribuant aux catalogues d'étoiles guides, aidant à identifier les sources observées ou organisant l'accès aux archives, etc. Le CDS contribue au XMM Survey Science Center, sous la responsabilité de l'équipe "Hautes-Energies" de l'Observatoire de Strasbourg.

Le CDS a signé des accords d'échanges internationaux avec les organismes suivants :

- NASA,
- National Astronomical Observatory (Tokyo, Japon),
- l'Académie des Sciences de Russie,
- le réseau PPARC Startlink au Royaume-Uni,

- l'Observatoire de Beijing (Chine),
- l'Université de Porto Alegre au Brésil,
- l'Université de La Plata en Argentine,
- InterUniversity Center for Astronomy and Astrophysics (Inde).

Il est membre de la Fédération des Services d'Analyse de Données Astrophysiques et Géophysiques.

Le CDS coopère aussi avec l'Agence spatiale Européenne (transfert au CDS du service de catalogues du projet ESIS : Le projet VizieR), et avec la NASA : il abrite en particulier une copie miroir du Système de Données Astrophysiques (ADS) et ADS abrite une copie miroir de Simbad. Le CDS contribue aussi au projet NASA AstroBrowse. Il abrite les copies miroirs Européennes des journaux de l'American Astronomical Society (AAS)

1.5 L'Observatoire Virtuel

Jusqu'à une époque récente, un grand nombre de projets ne prévoyaient pas de rendre accessibles les données issues des missions (spatiales, terrestres). Depuis l'avènement des nouvelles technologies, Internet et les réseaux rapides, d'importants efforts ont permis de mettre à disposition des astronomes de nombreuses sources de données (on peut citer par exemple les services VizieR, Simbad et Aladin du CDS). Depuis quelques années, l'interopérabilité entre ces services est devenu une priorité car elle permet un accès aisé à l'information, un croisement des données de toutes natures et elle conduit au concept d'Observatoire Virtuel. L'astronome disposera à terme de nouveaux instruments utiles à sa recherche au travers d'une simple interface Web. Afin d'y parvenir, de nombreuses collaborations sont indispensables afin d'aboutir aux consensus indispensables à la pose des bases nécessaires à la "construction" de l'Observatoire Virtuel.

Les projets nationaux (OVF pour la France, GAVO pour l'Allemagne, le NVO pour les Etats-Unis, ...) et transnationaux (ESA VO, ESO VO, ...) sont réunis au sein de l'International Virtual Observatory Alliance. Ils participent à l'élaboration de Recommandations dans divers domaines (Data Model, Accès aux données, Sémantique, Grilles, etc.) au travers de groupes de travail qui se réunissent semestriellement. A titre d'exemple significatif, la première recommandation, "VOTable", décrit la formalisation de tables de données astronomiques au format XML. Celle-ci a été adoptée par de nombreux fournisseurs de données et intégrée dans les outils de l'OV.

1.6 Les Services du CDS

1.6.1 Simbad

Simbad est une base de données de référence pour les identificateurs et la bibliographie d'objets astronomiques. Simbad contient plus de 7,5 millions d'identificateurs pour plus de 2,8 millions d'objets différents. Pour chaque objet figurent dans la base quelques mesures (position, magnitude dans différents domaines de longueurs d'ondes), ainsi que les références bibliographiques où l'objet est cité (plus de 110 000 articles sont concernés). L'utilisateur peut choisir le format du fichier où seront entreposés les résultats de la requête (Fig. 1.1). Simbad peut générer des fichiers HTML, XML ou XLS (fichiers Excel).

Cet ensemble de données résulte d'un long travail d'identification croisée entre de nombreux catalogues, listes d'objets et articles de journaux, entrepris au début des années 1980, et constamment développé et mis à jour depuis.

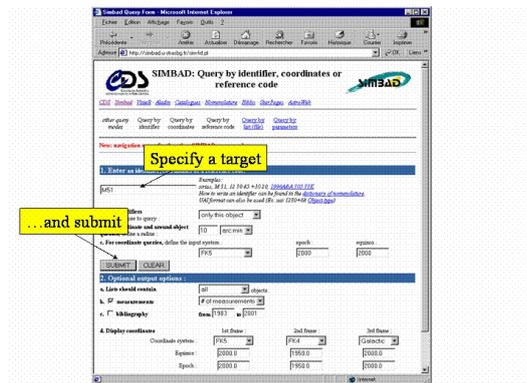


FIG. 1.1 – Page Web permettant d’effectuer une requête sur Simbad

1.6.2 VizieR

VizieR est une base de données rassemblant plusieurs milliers de catalogues astronomiques sous un format homogène. Une description standardisée du contenu des catalogues permet leur inclusion dans un système de gestion de base de données (SGBD) relationnel. Un ensemble de liens, entre les tables de VizieR, et avec des services externes (bibliographiques, archives externes, serveur d’images), permettent de naviguer entre les données des catalogues et d’autres données associées (Fig. 1.2). Il faut noter que les très grands catalogues (plus de 107 enregistrements) ne peuvent pas être gérés par un SGBD relationnel pour des raisons de performances, c’est pourquoi des outils spécifiques doivent être utilisés.

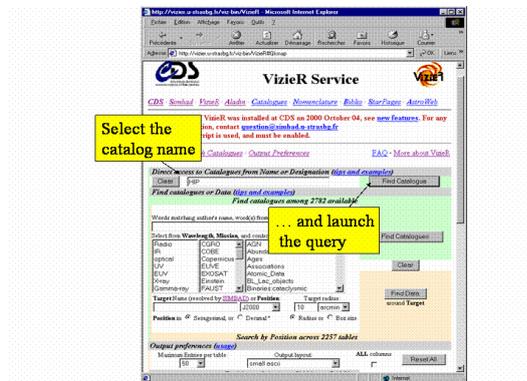


FIG. 1.2 – Page Web permettant d’effectuer une requête sur VizieR

1.6.3 Aladin

Aladin est un atlas interactif du ciel permettant d’accéder simultanément à des images numérisés du ciel, ainsi qu’à des catalogues et bases de données astronomiques. Cet outil permet de superposer, sur des images du ciel optique, les objets présents dans Simbad, des sources de catalogues contenus dans VizieR, mais aussi d’autres données, locales ou situées sur des serveurs distants (archive,HST,...).

1.6.4 Les autres services

Parmi les autres services offerts par le CDS, on peut citer le dictionnaire de nomenclature, les services bibliographiques et les services pages jaunes(AstroWeb, Sar’s family, AstroGlu).

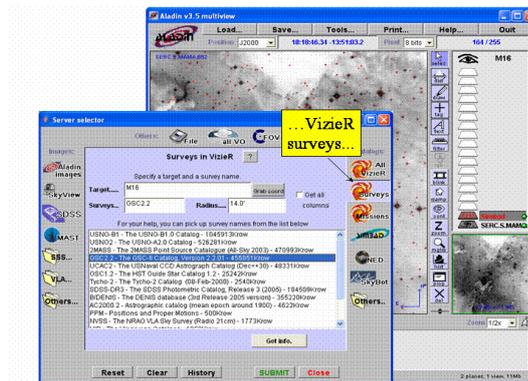


FIG. 1.3 – Exemple d'utilisation d'Aladin

Le CDS est également impliqué dans le développement d'outils et de méthodes d'intérêt général pour l'échange de données et la standardisation.

Deuxième partie

Analyse de la situation

Chapitre 2

Analyse de la situation

2.1 Analyse de l'existant

L'observatoire de Strasbourg dispose de services qu'il met à disposition de l'ensemble de la communauté astronomique internationale. Cette dernière comprend des chercheurs en astronomie et astrophysique mais également des amateurs (les services étant publics). Les principaux services sont proposés à travers un portail Web. Il sont assez variés puisque le CDS proposent aussi bien des publications que des accès aux bases de données (VizieR et Simbad qui contiennent respectivement des catalogues astronomiques et des identificateurs pour des objets astronomiques) ou un atlas astronomique (Aladin qui permet à n'importe qui de pouvoir afficher chez lui des images astronomiques issues du serveur et de manipuler (rotations, déplacements, zooms, fusions de plusieurs images,...)). Ce genre de service nécessite beaucoup d'interactivité et d'interrogation sur des millions d'enregistrements.

2.2 Déroulement du stage

Le stage a débuté par les Services Web du Centre de Données astronomique, afin de me familiariser avec ces outils. Suite à une réunion avec mes responsables, j'ai commencé par un état de l'art visant à réaliser une recherche approfondie des éléments à prendre en compte pour mener à bien le projet. L'objectif de cette réunion était de passer en revue les outils, librairies, sites Web en rapport avec AJAX certains aspects étant à considérer en priorité : la fiabilité (outils maintenus), la documentation, le type de licence, la facilité de mise en oeuvre, etc. J'ai commenté au fur et à mesure les résultats de ces tests sur le TWiki⁵ du CDS afin que mes collègues puissent consulter le résultat de mes recherches. Le mois de mars m'a permis de me familiariser avec les solutions existantes, et surtout de comprendre la technologie AJAX. À la fin de ce mois, une réunion avec mes responsables et d'autres personnes du CDS intéressées par cette nouvelle technologie, eu lieu afin d'abordé tous les points que nous nous étions fixés lors de notre première réunion afin de faire les orientations possible.

Pendant le mois suivant, je me suis consacré à l'exploration de toutes les solutions retenues. Durant cette période, j'ai testé plusieurs bibliothèques AJAX, en créant une application (Autocomplétion) client-serveur retenue lors de notre dernière réunion. Ce mois fût fort intéressant car j'ai pu me familiariser avec les différentes bibliothèques et de voir les aspects essentiels de toutes ces bibliothèques pour la suite du projet tous en mettant à jour TWiki du CDS. À la fin du mois, j'ai exposé mon travail et j'ai expliqué le fonctionnement des différentes types de bibliothèques.

Durant le mois de mai, j'ai abordé le projet VizieR Mine afin d'améliorer l'interactivité de ce service, tous en utilisant les outils étudiés durant le mois précédent. Il y avait plusieurs objectifs à atteindre, qui

⁵le principe du wiki est de fournir un environnement collaboratif

ont été tous réalisés. Nous avons fait plusieurs tests de cette application. D'autres utilisateurs l'ont testé et leurs remarques ont été prises en compte.

Durant les deux mois qui me restent à faire, l'objectif était d'enrichir les services Web de l'Observatoire astronomique, en ajoutant des parties dynamiques et innovantes mettant à profit la technologie AJAX.

Troisième partie

Etat de l'art de la technologie AJAX

Chapitre 3

Etat de l'art de la technologie AJAX

3.1 Introduction

Idéalement l'application doit permettre à l'utilisateur d'arriver à ses fins de manière transparente, en l'assistant efficacement durant son utilisation. Nous sommes habitués à utiliser fréquemment des interfaces Web peu optimisées souvent en raison des limites des technologies de base du Web.

Le Web est en train de subir une telle évolution. En effet, les technologies de base du Web pour l'affichage de contenu on déjà été poussées très loin.

3.2 Qu'est-ce que AJAX ?

AJAX, pour Asynchronous Javascript And XML, et un nom apparu en février 2005 dans un article publié par Jess James Garret ⁶. AJAX est une approche du développement d'interfaces Web riches. Construire une interface client riche est un peu plus compliqué que créer de simples pages Web. Le mot riche définit le modèle d'interaction avec le client. Un modèle d'interaction est celui qui supporte un grand nombre de possibilités en entrées en répondant de manière intuitive et de façon très dynamique.

AJAX n'est pas une technologie, c'est une appellation qui désigne une architecture c'est-à-dire une façon de concevoir et de développer un service en ligne, et fait usage des éléments de base que sont XHTML, CSS, Javascript, DOM, XML, XSLT et XMLHttpRequest. C'est en fait Microsoft qui le premier a lancé le concept, car à la base l'objet XMLHttpRequest était un ActiveX d'Internet Explorer 4 (en 1999). Cet ActiveX a ensuite été intégré sous cette forme sur tout les navigateurs modernes sous l'impulsion de Mozilla.

Google est le premier à l'avoir massivement employé, et donc popularisé, avec ses services Google Suggest (l'outil de suggestion des requêtes), Gmail ⁷ (le webmail de Google⁸) ou encore Google Maps ⁹ (l'outil de consultation de cartes), même si les technologies sur lesquels AJAX repose existaient depuis plus de 5 ans et qu'une multitude de sites faisaient déjà de l'AJAX sans le savoir.

Les services Web deviennent de plus en plus complexes et les applications Web classiques commencent à ne plus répondre aux exigences. AJAX, en alliant diverses technologies ayant fait leurs preuves et qui sont présentent sur la majorité des ordinateurs, permet de combler ce trou par la création de clients plus riches et plus intelligents.

⁶<http://www.adaptivepath.com/publications/essays/archives/000385.php>

⁷<http://www.gmail.com>

⁸<http://google.com>

⁹<http://maps.google.com>

3.3 Pourquoi AJAX ?

3.4 Les manques dans les applications Web actuelles

Les utilisateurs sont de plus en plus exigeants vis à vis des temps de réponse des applications Web. Après chaque action il faut patienter jusqu'à l'affichage du résultat. Prenons l'exemple d'une liste d'éléments sur une page. Si l'on veut changer leur ordre, la méthode consiste à placer des flèches cliquables permettant de faire monter ou descendre l'élément d'un niveau. Dans le cas d'un client riche, l'interaction est bien plus complète, en faisant par exemple un glisser-déposer, permettant de rapidement et simplement faire changer l'élément de plusieurs niveaux.

Le cas d'un traitement de texte fournissant un correcteur orthographique ou encore le tri sur un tableau, le tout agissant instantanément sont d'autres exemples.

3.5 Petit historique du web

Le Web a beaucoup évolué depuis ses débuts, nous sommes passés des pages statiques et difficilement éditables aux applications dynamiques et aux portails collaboratifs en quelques années seulement. Voici une rétrospective sous forme d'un tableau synthétique

3.5.1 Histoire du Web

	Web 1.0	Web 1.5	Web 2.0
Type de Web	statique	dynamique	collaboratif
Période	1994-1997	1997-2003	2003-...
Technologie associé	HTML GIF	ASP-PHP-Perl-CSS	AJAX
Nom associé	Netscape	Yahoo!, Amazon, eBay	Google,netvibes ¹⁰

TAB. 3.1 – Historique du Web.

3.6 Buts d'AJAX

L'objectif d'AJAX est de mettre à disposition des applications utilisables à l'aide d'un navigateur Web et satisfaisant les buts de productivité, de réseau, du moindre effort et de la maintenance centralisée d'une application. Pour atteindre ce niveau, il faut commencer par penser différemment la façon de créer les pages Web ainsi que les applications.

3.7 Trois principes d'Ajax

1. Le serveur délivre les données, pas la présentation.
2. Le navigateur accueille l'application.
3. L'utilisateur interagit continuellement avec l'application, les requêtes aux serveurs sont implicites.

3.8 Les éléments clés d'AJAX

3.8.1 Javascript

Javascript¹¹ est un langage de script créé pour être utilisé à l'intérieur d'applications. Dans un navigateur Web il permet l'interaction avec les fonctionnalités de celui-ci. Les applications AJAX sont écrites en Javascript.

3.8.2 Cascading Style Sheets (CSS)

Le CSS¹² permet de définir un aspect graphique réutilisable pour des éléments d'une page Web. Dans une application AJAX, l'interface utilisateur peut être modifiée dynamiquement grâce à ces feuilles de style.

3.8.3 Document Object Model (DOM)

Le DOM¹³ représente la structure de la page sous forme d'une collection d'objets pouvant être manipulés avec Javascript. Elle nous permet de modifier l'interface utilisateur en redessinant des parties de la page.

3.8.4 L'objet XMLHttpRequest

L'objet XMLHttpRequest¹⁴ permet aux programmeurs Web d'obtenir des données du serveur Web en tâche de fond. Le format des données est typiquement de l'XML mais cela fonctionne aussi avec n'importe quelles données textuelles.

3.9 Clients riche Ajax dans le monde réel

C'est Google qui a le plus popularisé ce nouveau profil d'application. C'était au début 2004, avant que le terme AJAX n'existe, grâce à son webmail nommé Gmail. Ils ont en effet été les premiers à proposer une interface graphique efficace, fournissant les mêmes fonctions que les webmail d'entreprise tel que Microsoft Outlook. Cela a permis d'avoir un webmail puissant multi-plateformes et multi-endroits car il n'y avait plus besoin de préconfigurer le poste client pour avoir un outil de qualité.

¹¹<http://www.ecma-international.org/publications/standards/Ecma-262.htm>

¹²www.w3.org/TR/REC-CSS2

¹³www.w3.org/DOM

¹⁴<http://www.xulplanet.com/references/objref/XMLHttpRequest.html>

Quatrième partie

Etude d'AJAX

Chapitre 4

Etude d'AJAX

4.1 Etude de la mise en place

Le but de cette analyse est de déterminer quels moyens généraux sont nécessaires pour déployer une architecture AJAX aussi bien dans la partie serveur que cliente.

4.1.1 Communication asynchrone

Principe

Dans le modèle des pages Web "classiques" encore largement utilisé aujourd'hui, la communication entre le navigateur et le serveur Web est dite synchrone.

En effet, à partir d'un certain état d'une page, l'utilisateur ne peut effectuer qu'une seule requête aboutissant à un rechargement complet de celle-ci, le contenu restant statique avant d'être à nouveau mis à jour en intégralité (Fig 4.1).

Le rechargement complet d'une page s'accompagne souvent d'une réinitialisation des scrollbars et d'une brève disparition du contenu limitant fortement l'impression d'interactivité pour l'utilisateur, et rendant très difficile l'utilisation d'éléments standards dans des interfaces utilisateurs à contenu dynamique, tels que les éléments "Drag and Drop".

La communication asynchrone permet de résoudre ce problème en permettant à l'utilisateur, depuis un certain état d'une page, de lancer un nombre arbitraire de requêtes pour mettre à jour uniquement des éléments internes à celle-ci (Fig 4.1).

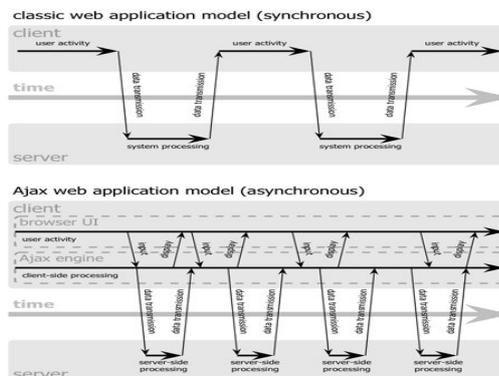


FIG. 4.1 – Comparaison des mécanismes d'échanges client-serveur synchrones et asynchrones

Implémentation avec Ajax

L'implémentation d'un concept de communication asynchrone passe par une adaptation au niveau client (qui doit être capable de lancer des requêtes de manière asynchrone), et éventuellement au niveau du serveur. En effet, les données envoyées n'étant plus forcément des pages Web mais des "données utilisateur brutes" (pour par exemple mettre à jour un formulaire), on peut attendre de ce dernier qu'il soit capable de traiter différents formats de données.

AJAX fonctionne différemment des applications Internet classiques, comme on peut le constater sur la figure (Fig 4.2) suivante

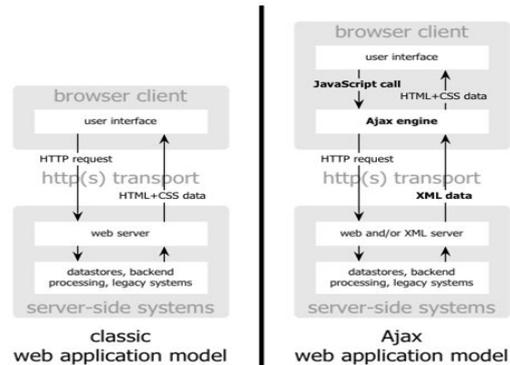


FIG. 4.2 – Le modèle traditionnel pour les applications Web (gauche) comparé au modèle Ajax (droite).

Une des différences majeures entre le Web classique et le Web basé sur AJAX est l'apparition d'un moteur chargé au démarrage du client (navigateur). Toujours présentés chronologiquement, voici les différentes étapes des transactions avec modèle basé sur AJAX :

- Le client engendre une action qui génère une requête Javascript vers le moteur AJAX.
- Le moteur AJAX génère une requête vers le serveur.
- Le serveur exécute des traitements (calculs, requêtes SQL ...) et retourne des données XML au moteur du client.
- Le moteur AJAX transforme les données XML en données HTTP et envoie ce résultat au client.
- Le client interprète le code HTML et affiche la page.

Remarque : la page n'est pas renvoyée entièrement. Seules les données demandées au serveur sont incluses dans la page.

Le moteur AJAX

- Une application AJAX élimine les transactions "start-stop-start-stop.." en ajoutant un moteur AJAX entre l'utilisateur et le serveur. Au démarrage, le navigateur charge un moteur AJAX (écrit en javascript) dans une frame cachée.
- Ce moteur est responsable du rendu vu, de la communication avec le serveur et permet de rendre les interactions et les communications avec les serveurs asynchrones.
- Toutes les actions utilisateurs nécessitant une requête HTTP prennent la forme d'un appel Javascript à la machine AJAX
- Toutes les actions qui ne nécessitent pas de traitement de la part du serveur sont prises en charge par le moteur AJAX (simple validation de données, édition de données en mémoire et même certaines navigations.)
- Si le moteur a besoin de données ou de nouveaux codes d'interface, il soumet ces requêtes au serveur de manière asynchrone.

4.2 Méthode des cadres cachés

La méthode des cadres cachés peut techniquement être implémentée depuis la sortie de Netscape Navigator 2.0 en septembre 1995. Ce n'est toutefois qu'en 2001-2002 que les premiers articles ont releté ce principe. On peut notamment citer un article paru sur le site d'Apple à cette époque et qui s'intitule *Remote Scripting with IFRAME*¹⁵.

Le principe des cadres cachés consiste en l'intégration à une page html d'une frame (cadre) dont la taille est à 0, ce qui permet de disposer d'un cadre caché dans la page. C'est à l'intérieur de celui-ci que l'on place le code Javascript qui appellera le serveur. Le schéma ci-dessous (Fig 4.3) décrit les différentes étapes du processus.

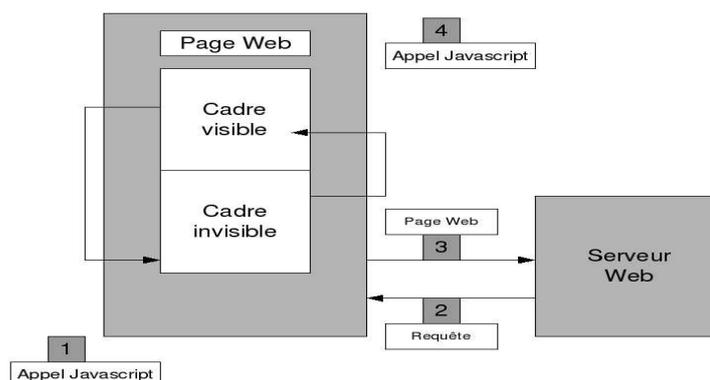


FIG. 4.3 – Technique des cadres cachées

1. Suite à une action de l'utilisateur, le cadre visible appelle en Javascript une fonction qui se trouve dans le cadre caché.
2. Le cadre caché envoie une requête au serveur Web, en lui passant les paramètres nécessaires.
3. Le serveur renvoie une page Web au cadre caché. Cette page contient les données répondant à la requête, ainsi que le code Javascript qui lui permettra de mettre à jour le cadre visible.
4. La page renvoyée prend place dans le cadre caché, lors de son chargement elle appelle en Javascript une fonction contenue dans le cadre visible, en lui passant les paramètres nécessaires. Le cadre visible peut alors mettre à jour les éléments voulus dans la page.

Les étapes une à trois ne sont pas perçues par l'utilisateur. En effet c'est uniquement le cadre caché qui appelle le serveur et se rafraîchit intégralement. A l'étape quatre, le cadre caché est entièrement rechargé, puisqu'une nouvelle page prend place dans celui-ci. La technique des cadres cachés a l'avantage de conserver l'historique de navigation : les boutons "précédent" et "suivant" du navigateur actif, ce n'est pas le cas en utilisant XMLHttpRequest. Il faut toutefois noter que l'historique n'est pas conservé par le navigateur "FireFox" dans le cas d'utilisation d'iframes

¹⁵developer.apple.com/internet/webcontent/iframe.html

Cette méthode a plusieurs inconvénients. L'un d'eux est l'absence de remontée des erreurs. Etant donné que le code se trouve dans une partie cachée, l'utilisateur n'est pas informé en cas d'erreur, la page restera figée sans afficher un message. Il est alors difficile de savoir ce qui se passe en arrière plan.

4.3 Méthode utilisant l'objet XMLHttpRequest

L'objet XMLHttpRequest permet d'appeler une page en Javascript et de récupérer son contenu. Pour utiliser l'objet XMLHttpRequest en Javascript, il faut commencer par l'instancier. Cette étape peut paraître simple, mais c'est durant celle-ci qu'il faut gérer les différentes versions du navigateur.

Une fois l'objet créé, il faut :

- l'initialiser.
- lui spécifier quelle page il doit appeler.
- lui spécifier quelle méthode il doit appeler à la fin de son traitement, lorsque la page a été récupérée.
- lancer la requête.

4.3.1 Cycle de vie de l'objet XMLHttpRequest

En mode asynchrone, l'instance de la classe XMLHttpRequest utilisée supporte les différents états récapitulés dans le tableau (Tab 4.1).

Valeur	Etat	Description
0	Uninitialized	Etat initial
1	Open	La méthode open a été exécutée avec succès
2	Sent	La requête a été correctement envoyée, mais aucune donnée n'a encore été recue
3	Receiving	Des données sont en cours de réception
4	Loaded	Le traitement de requête est fini

TAB. 4.1 – Etats de l'instance de la classe XMLHttpRequest en mode asynchrone

4.3.2 Fonctionnement

Cette figure (Fig 4.4) nous montre, comment s'organise la méthode AJAX, basée sur l'utilisation de cet objet et qui décrit les différentes étapes du traitement.

1. En premier lieu l'évènement est déclenché dans la page, par une action quelconque
2. Une méthode Javascript est appelée. Celle-ci crée un objet XMLHttpRequest, lui spécifie la page à appeler et lance le traitement.
3. La page appelée sur le serveur peut être un simple fichier texte à récupérer, mais c'est en général une page qui contient du code permettant de générer un fichier XML ou HTML. Ce fichier peut contenir par exemple des données issues d'une base de données. Le serveur, après traitement, renvoie donc ce fichier à l'objet XMLHttpRequest appelant.
4. Une fois le fichier récupéré l'objet XMLHttpRequest appelle une méthode Javascript. Cette méthode vérifie le code d'erreur, et traite les données récupérées, en modifiant les parties de la page qui sont impactées.

L'avantage de cette méthode par rapport à celle des cadres cachés, est qu'elle permet d'écrire du code plus propre et plus efficace. Elle permet également de gérer correctement les erreurs, alors que la méthode des cadres cachés ne le permet pas.

Attribut	Description
onreadystatechange	Permet de positionner une fonction de rappel, qui est appelée lorsque l'événement readystatechange se produit. Cet événement est supporté en mode asynchrone et correspond à un changement d'état dans le traitement de la requête.
readyState	Code correspondant à l'état dans lequel se trouve l'objet tout au long du traitement d'une requête et ou sa réponse
responseText	Réponse reçue sous forme de texte
responseXML	Réponse reçue sous forme XML
status	Correspond au code du statut de retour HTTP de la réponse. Il n'est disponible en lecture que lorsque la réponse à la requête a été reçue.
statusText	Correspond au texte du statut de retour HTTP de la réponse. Il n'est disponible en lecture que lorsque la réponse à la requête a été reçue.

TAB. 4.2 – Attributs de la classe XMLHttpRequest

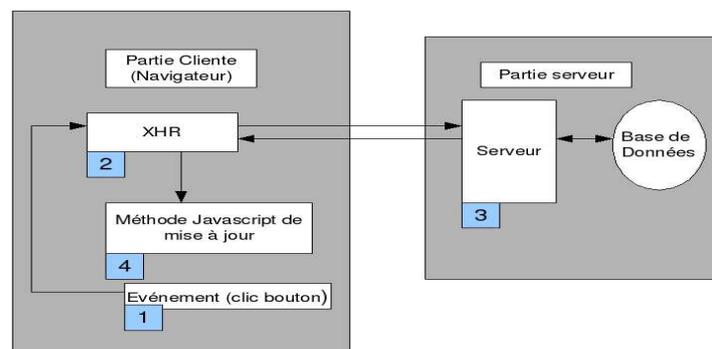


FIG. 4.4 – Technique XMLHttpRequest

En revanche, il faut signaler que l'objet XMLHttpRequest est compatible avec la plupart des navigateurs sauf dans certains cas :

- Navigateur trop ancien .
- Navigateur Internet Explorer, sur lequel les ActiveX sont désactivés.

4.4 Mise en place côté serveur

Les requêtes asynchrones étant traitées de manière classique, l'utilisation d'AJAX peut se faire de manière totalement indépendante du serveur. On aura néanmoins un avantage indéniable à choisir une plateforme de programmation (telle que Perl, Java, PHP, etc) correspondante à nos besoins. La plupart d'entre elles supportent de manière native un parseur XML permettant de manipuler plus facilement les données transmises.

4.5 Mise en place côté client

Tout comme la partie serveur, la mise en place d'AJAX côté client est relativement légère, elle se base principalement sur la présence de 2 fonctionnalités :

- La capacité d'envoyer des requêtes de type XMLHttpRequest pour la communication asynchrone.
- La capacité à traiter les réponses XML du serveur.

L'implémentation des objets XMLHttpRequest est native dans la plupart des navigateurs récents et ne nécessite donc pas de plugin supplémentaire.

4.6 Apports d'AJAX

Les divers technologies utilisées par AJAX ne sont pas nouvelles. En les utilisant de manière appropriée on peut envisager le développement Web de clients riches.

4.6.1 Les trois principes définissant le terme Ajax

Le navigateur héberge une application et non du contenu

Avec une application AJAX, une partie de la logique serveur est transférée vers le client. Le document va rester dans le navigateur tout au long de la session. Comme ce document est persistant différents états peuvent être stockés désormais dans le navigateur. Contrairement à une application Web classique chaque fois que l'utilisateur interagit avec un site, un nouveau document est envoyé vers le navigateur. Le navigateur remplace l'ancien document (page) par le nouveau.

Le serveur envoie des données et non du contenu

Dans une application Web classique le serveur fournit un mélange de données et de présentation (HTML). Par exemple dans un service de réservation d'hôtel, on a uniquement besoin du prix de la chambre. Une application AJAX enverrait une requête asynchrone au serveur, et en retour ne recevrait que la donnée "prix". Toute la présentation resterait inchangée du côté du navigateur.

L'interaction utilisateur avec l'application peut être fluide et continue

Effectué dans une applications classique de la soumission d'un formulaire : quand l'utilisateur fait un submit de la page, la page courante reste encore invisible un moment. L'utilisateur est invité à attendre que la page entière soit rafraîchie. Dans l'exemple du caddie, notre application AJAX envoie des requêtes asynchrones, l'utilisateur peut continuer de son côté à rajouter des éléments au panier. Si le code client est robuste, il va supporter ces échanges très facilement, et l'utilisateur pourra continuer à faire ce qui lui plait sans avoir besoin d'attendre quoi que ce soit.

4.7 Avantage et Désavantages d'Ajax

Les Avantages et inconvénients d'AJAX sont liés à la technique du XMLHttpRequest.

4.7.1 Ajout de nouvelles fonctionnalités

AJAX ouvre la porte à des fonctionnalités que l'on n'aurait jamais imaginé pouvoir implémenter dans un site Web auparavant. Il devient possible de déplacer des zones d'une page d'un endroit à un autre, d'afficher des informations provenant du serveur au survol d'un élément ou de faire des recherches en arrière plan lorsque l'utilisateur saisit des caractères au clavier (Google Suggest¹⁶).

4.7.2 Gain de bande passante

Du côté technique, on peut mettre en avant la diminution du volume des données qui transitent par le réseau. Auparavant toute la page était envoyée à chaque évènement. Maintenant elle est chargée au moment où l'utilisateur arrive sur celle-ci, et ensuite ce sont seulement les données nécessaires qui sont récupérées lors des évènements. Cela peut représenter un gros avantage pour les sites qui accueillent beaucoup d'utilisateurs, ou ceux qui payent un hébergeur en fonction de la qualité des données échangées.

4.7.3 Amélioration de l'interactivité

C'est le point essentiel d'AJAX, qui permet à l'utilisateur de naviguer presque comme s'il était sur une application installé directement sur son poste. Les échanges entre le client et le serveur se font oublier puisqu'il s'effectuent en arrière-plan . Par conséquent l'utilisateur n'est plus coupé dans sa navigation, a des vitesses d'affichage beaucoup plus courtes, et les sites qui lui sont proposés sont beaucoup plus riches qu'auparavant (dynamisation des éléments de pages).

4.7.4 Ergonomie

Impossible d'indexer les pages

Les robots d'indexation, permettent aux moteurs de recherche de retrouver les pages en fonction des critères de recherche. Le fonctionnement est le suivant : le robot va trouver une page, indexer son contenu et parcourir ensuite toutes les pages référencés par la première au travers d'un lien hypertexte. Dans un site AJAX, le contenu initial des pages (état affiché lorsqu'un utilisateur y accède) va donc être indexé. Mais dans une page AJAX, si l'utilisateur effectue une action, ce n'est pas une autre page qui est appelée, mais le contenu de la page courante qui change.

Gestion des marques-pages

De la même manière, puisqu'une même page peut accueillir plusieurs contenus différents, il n'est pas possible pour l'utilisateur de se créer un "Favori" pointant sur la page affichée. Si un favori est créé, c'est la page dans son état initial qui apparaîtra lorsqu'il sera ouvert.

¹⁶<http://google.com>

4.7.5 Adaptabilité

Compatibilité

Il ne faut pas oublier que pour une minorité de personnes qui utilisent d'anciens navigateurs ne possédant pas l'objet XMLHttpRequest les sites AJAX ne fonctionnent pas. Il faut prévoir un double développement, avec une version qui n'utilise pas AJAX pour que ces personnes puissent quand même accéder au site. Cela implique un surcoût.

Utilisation de Javascript

Les internautes utilisent souvent un navigateur ne supportant pas Javascript (indispensable à l'utilisation d'AJAX) ou ayant désactivé le support de celui-ci (le plus souvent par soucis de sécurité).

Cross-browsing

Les implémentations de Javascript varient grandement entre les navigateurs, de ce fait certains composants AJAX propres aux applications riches, sont susceptibles d'être dépendant d'une plateforme cible. Cependant la majorité des Frameworks courants affiche une bonne compatibilité inter-navigateur

4.8 Service Web

4.8.1 Pourquoi la page ne parlerait-elle qu'à son propre site ?

Tout simplement pour des raisons techniques. Conceptuellement, il est parfaitement acceptable que notre page fédère du contenu qu'elle obtient dynamiquement depuis des services hébergés sur d'autres serveurs. C'est fondamentalement le rôle des services Web. Aujourd'hui, tout service fournit des services Web.

4.8.2 Contraintes de sécurité sur le navigateur

Des personnes aux intentions peu honorables pourraient contourner des mesures classiques de sécurité. Imaginons par exemple que vous soyez sur votre poste de travail en entreprise, et que vous accédiez à l'intranet, lequel contient sans doute des informations confidentielles. Et la, vous accédez une page dotée d'un script qui tente de récupérer des informations sur [http ://intranet/]. Pour ensuite les transmettre à un site distant.

C'est pour cette raison que pour certains navigateurs, Javascript interdit l'accès à d'autres domaines que celui dont la page est issue. Il peut s'agir d'un script. Il peut s'agir d'un script (fourni par frame ou iframe) issu d'un domaine B. Dans notre cas , il s'agit d'un objet XMLHttpRequest qui n'a pas le droit de communiquer avec un domaine tiers. Suivant le navigateur, cette contrainte est plus au moins prononcée. Il est souvent possible d'accéder à un domaine parent. Par exemple, un script exécuté par une page de cds.exemple.com peut avoir des ressources de exemple.com. Certains navigateurs ne mettent aucune restriction, par exemple MISE (qui n'est pas à une faille de sécurité près).

4.8.3 Couche proxy sur votre serveur

La seule solution véritablement portable, qui a de surcroît l'avantage de ne rien demander à l'utilisateur, consiste à mettre en place une couche intermédiaire sur notre serveur, qui se contente d'effectuer la requête qu'on lui donne et de transmettre le résultat tel quel au script. En effet notre couche serveur n'est pas soumise aux contraintes de sécurité du navigateur : elle peut émettre des requêtes où bon lui semble.

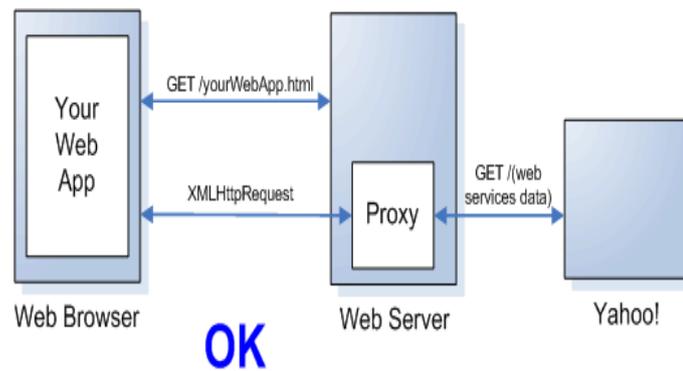


FIG. 4.5 – Proxy sur votre serveur

Il existe tout de même une différence entre l'utilisation d'une simple fonction de proxy et le modèle traditionnel de traitement sur la couche serveur. Le traitement doit se faire côté client : c'est plus léger pour le serveur.

Comme nous le montre la figure (Fig 4.6) on ne peut pas envoyer des requêtes XMLHttpRequest sur un autre domaine. La classe XMLHttpRequest va envoyer un message d'erreur lors de l'envoi.

Mettez l'image .

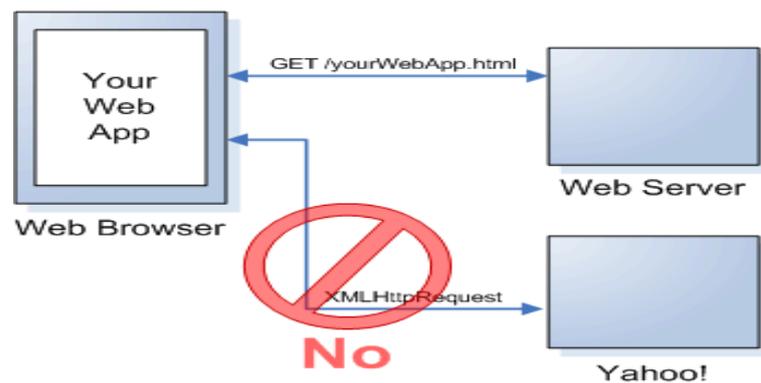


FIG. 4.6 – XMLHttpRequest et les services Webs

Cinquième partie

Les Frameworks AJAX

Chapitre 5

Les Frameworks

5.1 Définition

Un Framework est un ensemble de bibliothèques facilitant le développement. Un Framework AJAX est donc un ensemble de classes, d'outils, qui permet le développement d'une applications AJAX plus simplement et plus rapidement.

5.1.1 Les différents types Frameworks

Un ensemble de fichiers Javascript contenant des méthodes déjà écrites que l'on peut utiliser pour le développement (appelé Framework "client"). Un ensemble de quelques composants AJAX évolués que l'on peut intégrer à une application. Exemple : un composant sous forme d'une zone de texte. Un cadre de développement complet, qui imbrique toutes les fonctionnalités AJAX.

5.2 Avantages d'un Frameworks

5.2.1 Simplicité de mise en oeuvre

L'utilisation d'un Framework diminue très nettement la complexité de mise en oeuvre d'AJAX. La gestion des différentes versions des navigateurs, qui doit être faite rigoureusement, n'est plus à la charge du développeur, puisque le Framework s'en charge.

5.2.2 Maintenance

Les pages de développement sont plus claires et mieux structurées. Il est plus facile de maintenir l'application.

5.2.3 Ajout de fonctionnalité

Un Framework, en plus de simplifier la mise en oeuvre du moteur AJAX, fournit dans bien des cas des composants évolués qui permettent l'implémentation de fonctionnalités intéressantes.

5.3 Choisir une/des librairies de développement

- Sur quelle base voulons-nous évaluer les librairies/plateformes ?
- Que voulons-nous exactement ?
- Qu'est-ce qu'une application Web riche ?
- Qu'est ce qu'une application Web 2.0 ?
- Des applications client-serveur ?

- API externes (intégration facile.)
- Interactivité proche d'applications client-serveur traditionnelles.
- Réponse perçue plus rapide.
- Facilité de maintenance.
- Productivité des développeurs, intégration IDE.
- Minimum de normalisation.
- Performances.

5.4 Description de quelques Frameworks existants

5.4.1 Sondage

Un sondage Ajaxian (Fig. 5.1) mené auprès de 865 personnes révèle que Prototype est le Framework le plus utilisé, suivi de Script.aculo.us et Dojo et que Moo.fx, JQuery et YUI sont moins fréquemment employés tandis qu'Atlas, MochiKit et GWT sont relégués en bas de tableau.

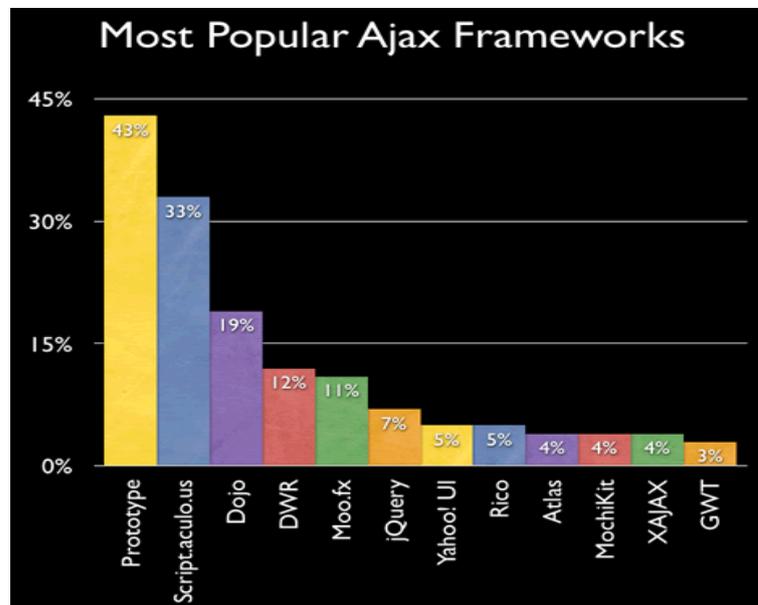


FIG. 5.1 – Résultat du sondage Ajaxian 2006 : Frameworks AJAX les plus populaires

Par ailleurs le sondage comme nous le montre la figure (Fig 5.2) montre également que la plateforme de développement la plus utilisée avec AJAX est PHP, ce qui est peu surprenant au vu de l'essor des plateformes LAMP (Linux + Apache + MySQL + PHP). Rails n'obtient pas un score satisfaisant malgré l'intégration native du toolkit Prototype.

5.4.2 Les types de librairies et plateformes

Prototype

Prototype peut être considéré comme une extension des fonctionnalités natives de Javascript. La bibliothèque se base sur un modèle objet inspiré de RUBY (Ruby est un langage de programmation interprété orienté objet). Développée par Sam Stephenson¹⁷, elle sert de base à plusieurs Frameworks Javascript.

¹⁷<http://sam.conio.net/>

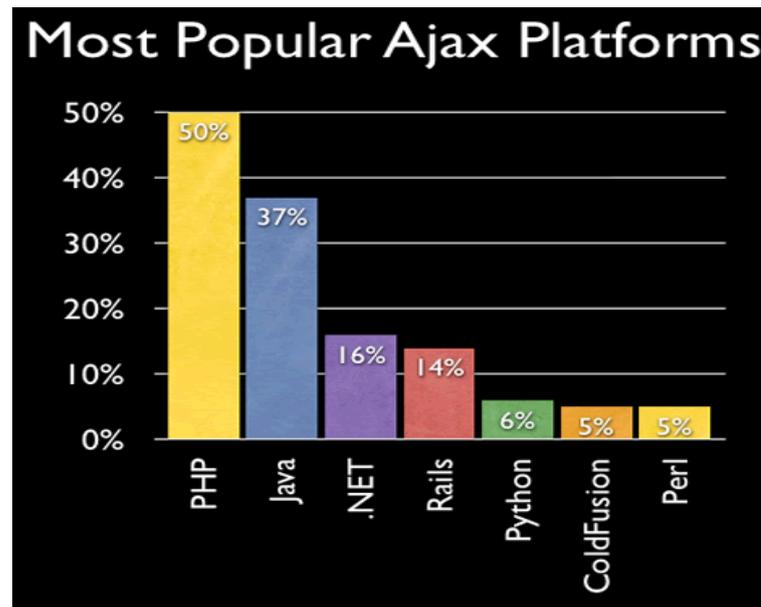


FIG. 5.2 – Résultat du sondage Ajaxian 2006 :plateformes AJAX les plus populaires

Script.aculo.us

Script.aculo.us est une bibliothèque AJAX basée sur Prototype (version 1.5rc), elle fournit :

- Un constructeur d'objets DOM (builder.js)
- Une large palette d'effets visuels (effects, silder.js).
- Des fonctionnalités de contrôle d'éléments HTML tels que l'autocomplétion le drag-and-drop et l'édition (dragdrop.js,control.js).
- Des méthodes pour procéder à des tests unitaires(unitest.js)

Dojo

Dojo Toolkit est un Framework de développement AJAX permettant une écriture simplifiée de l'interface interactive pour les applications Web. Il est née avec le site Jot.com qui l'utilise au sein de son système de wiki Jotpot¹⁸. Le toolkit inclut de nombreux composants personnalisables : Form Widgets, Layout widgets, Storage, Drag et Drop et Effects.

Dojo définit un ensemble de classes organisées en packages à la façon de Java. Ceci signifie donc qu'il est possible d'importer des classes dans une application Web au niveau du code HTML des pages (un équivalent de import en Java).

IBM et SUN ont récemment rejoint la fondation Dojo : ces deux acteurs participeront vraisemblablement à l'internationalisation de la documentation.

DWR

DWR est une librairie Java facilitant grandement l'utilisation d'AJAX dans les applications Web en Java. Il suffit d'écrire le code Java, de faire appel aux codes javascript auto-générés par DWR dans la JSP (utilise des objets java côté serveur et javascript côté client).

DWR permet à des objets Java d'être utilisés à distance selon un principe proche de RMI (Remote Method Invocation) .

¹⁸www.jotpot.com/

Le code Javascript au sein du navigateur client communique avec le serveur Web et met à jour dynamiquement la page et la servlet Java communique avec le serveur qui réalise les requêtes et renvoie les réponses à l'utilisateur .

Moo.fx

Moo.fx est une bibliothèque Javascript qui permet de créer des effets visuels : la bibliothèque ne dispose que de peu de composants. La bibliothèque étend sa richesse fonctionnelle avec les trois composants :

- Moo.fx : Composant graphique et effets visuels.
- Moo.Dom : Fonction de manipulation de DOM.
- Moo.Ajax : support relatif à la mise en oeuvre de la technologie AJAX.

Par ailleurs l'intérêt majeur de Moo.fx réside dans l'utilisation d'une version allégée de Prototype (Prototype.lite.js). Partie d'une initiative d'optimiser Prototype pour une utilisation personnelle, Moo.fx s'affirme à ce jour comme un compétiteur de talent attirant de plus en plus de contributeurs. Puis Moo.fx a été amélioré en janvier 2007, et renommé en MooTools.

MooTools est un Framework Javascript de dernière génération. Concrètement, il s'agit d'une bibliothèque Javascript modulaire permettant aux développeurs Web d'écrire de façon rapide des scripts basés sur une approche orientée objet, sans se soucier des éventuelles différences d'implémentation entre les différents navigateurs Web modernes.

JQuery

JQuery est une bibliothèque Javascript axée sur les standards

- DOM
- CSS 3
- XPATH (est une syntaxe (non XML) pour désigner une portion d'un document XML)

JQuery est une bibliothèque conçue pour changer la manière d'écrire en Javascript.

Yahoo! UI

Yahoo! User Interface Library est un exemple de projet Open Source de qualité : la documentation fournie est centrée sur les exemples et les méthodes possibles permettant d'exploiter les composants.

La bibliothèque se compose de trois collections :

- Utilités : manipulation du DOM, gestion d'évènements, Drag and Drop, utilisation XMLHttpRequest.
- Controls : widgets pour la plupart graphique (autocomplète, calendrier, container..)
- CSS : gestion des tailles, polices de caractères, styles prédéfinis.

Yahoo! UI librairie possède des Design Pattern Library qui est un ensemble de directives de conception pour des pages Web et des applications. Ces design pattern nous montrent comment bien mettre en oeuvre la bibliothèque de YUI dans des applications.

Riatlo

Riatlo (Rich Internet Application Toolkit) est une librairie Javascript multi-navigateurs axée Ajax de widgets. Ce toolkit peut être encapsulé dans PHP, ou .NET. Pour l'instant il supporte du pure Javascript et du JSP. L'idéal de Riatlo est un développeur n'a pas besoin de connaître ni HTML, ni AJAX, ni le codage DOM. Les cibles de Riatlo sont les applications Web d'entreprises et non des sites Web Internet.

Open Rico

Open Rico a été développé par une entreprise Américaine (Sabre Airline Solutions¹⁹) qui fournit des solutions informatiques à des compagnies aériennes. Le terme "Rico" signifie riche en espagnol, par allusion aux interfaces riches. Ce Framework fournit plusieurs éléments :

- Un moteur AJAX, pour communiquer avec le serveur en asynchrone.

¹⁹www.sabreairlinesolutions.com/

- Une librairie pour gérer le glisser -déposer.
- Une librairie pour gérer les animations, une autre pour les comportements.

Il se présente sous forme de deux bibliothèques Javascript téléchargeables gratuitement, et modifiables si nécessaire. Les méthodes fournies par ces bibliothèques sont prêtes à l'emploi. Il suffit d'intégrer les deux fichiers au site Web, et d'appeler les méthodes voulues.

Atlas

Microsoft Atlas est un Framework AJAX conçu pour s'intégrer au Framework .NET. Ses bibliothèques Javascript sont construites et liées de manière complexe ce qui rend difficile leur utilisation sans les outils Microsoft (Visual Studio). De fait, Atlas prévoit des mécanismes exploitant la puissance de son Framework .NET (par exemple l'accès via C# à une base de données).

MochiKit

Mochikit est une bibliothèque complète fournissant des méthodes pour travailler plus facilement avec Javascript notamment sur les structures de données et les données elles-mêmes. Cette bibliothèque est largement utilisée et testée, compte tenu de la couverture fonctionnelle limitée, Mochikit est assez facile à intégrer dans un code existant.

XAJAX

XAJAX est une librairie PHP permettant de créer facilement de puissantes applications Web AJAX. Les fonctions PHP côté serveur peuvent être appelées de manière asynchrone et ainsi permettre la mise à jour du contenu sans rechargement de la page.

GWT

Google Web ToolKit (GWT) est un Framework AJAX atypique :

- Le développement se fait intégralement sous Java 1.4 (bibliothèque graphique Swing ou AWT).
- La compilation traduit le code Java en Javascript, cette translation de code est une solution élégante pour développer des applications AJAX sans connaissance de ce dernier

Echo2

Echo2 tente de s'approcher des clients riches à travers une plateforme Java de développement Web. Le Framework s'intègre à un moteur de servlets pour convertir côté client les requêtes HTTP en Javascript. NextApp ²⁰ l'éditeur d'Echo2 vend ses atouts en précisant qu'aucune ligne de code Javascript, HTML, ou XML n'est écrite lors de son utilisation notamment grâce à son IDE Echo studio et au plugin Eclipse. Le Framework fournit une API riche (composants d'interface utilisateur, événements/écouteurs.)

5.5 Choix des Frameworks

Nous allons nous pencher sur la solution qui sera la plus souvent retenue, l'utilisation d'un Framework existant. Il est impossible de conseiller de prime abord un Framework existant qui sera parfait pour tous les projets. Chaque Framework AJAX a ses avantages et ses inconvénients. Voici quelques critères qui permettront de faire un choix :

²⁰<http://www.nextapp.com/platform/echo2/echo>

5.5.1 Nos besoins

Le langage

Le Framework AJAX doit donc être choisi en fonction du langage du projet (PHP, JAVA, .NET,...). Ce qui réduit fortement la liste de Framework AJAX utilisables. S'il n'existe aucun Framework adapté à la technologie utilisée pour le projet, il faudra se tourner vers un Framework "côté" client, développé uniquement en Javascript, et donc indépendant de la technologie utilisée pour le projet. En règle générale il est préférable d'utiliser un Framework "côté" serveur adapté au langage.

Documentations

Un point très important est le niveau de documentation (fonctions documentées, exemple, etc.) du Framework, nécessaire pour ne pas gaspiller trop de temps à faire des essais et pour comprendre pleinement l'implémentation des mécanismes asynchrones.

Maturité

Il est conseillé en règle générale d'utiliser un Framework qui soit déjà bien implanté et suffisamment documenté. Il est préférable d'utiliser un Framework qui a déjà fait ses preuves, et qui soit suffisamment utilisé pour être sûr qu'il ne sera pas laissé à l'abandon par ses créateurs.

5.5.2 L'éditeur

Si on choisit un Framework développé par une société importante, le risque de voir le Framework abandonné est moins important. Par exemple, le Framework ATF, soutenu par IBM, Borland,...et intégré à Eclipse a de fortes chances de devenir l'un des plus utilisés. Il en est de même, pour le Framework développé par Microsoft.

5.5.3 Projet existant ou nouveau projet

Il faut savoir si l'on veut seulement pouvoir recharger une partie de la page sans recharger le reste, ou si l'on a besoin de fonctionnalités vraiment évoluées, qui nécessitent des composants divers. Le Framework retenu ne sera pas forcément le même pour un nouveau projet ou pour un projet existant. Certains Frameworks sont très complets et pratiques à utiliser, mais peuvent difficilement être intégrés à un projet existant.

5.5.4 Les performances

Un des critères qui peut être important si le site est très visité est le poids des pages. Nous avons vu qu'AJAX ajoute du code Javascript dans les pages, ce qui peut alourdir plus ou moins le code client. En règle générale, ce critère est lié aux fonctionnalités que propose le Framework. Plus les fonctionnalités offertes sont nombreuses, plus la quantité de Javascript est importante, plus les pages lentes à charger. Si les fonctionnalités requises par le site sont minimales, il est donc préférable de ne pas utiliser un Framework trop riche.

Fonctionnalités supplémentaires(non-AJAX)

Pour pouvoir se concentrer principalement sur le code AJAX, il nous sera très utile de posséder des fonctions fournies pour des applications riches (telles que des arbres, des barres de progression, etc.) que nous pourrions utiliser telles quelles.

5.5.5 Frameworks retenus

Après une présentation et une analyse basés sur les exemples, nous avons aborder le choix des bibliothèques susceptibles d'enrichir les services Web de l'observatoire. Les bibliothèques ayant retenus notre attention sont les suivantes :

- Dojo
- Prototype
- Riatlo
- Script.aculo.us
- Open Rico
- DWR

Sixième partie

Conception

Chapitre 6

Conception

6.1 Introduction

La deuxième étape du stage devrait nous permettre de soutenir les différents points abordés dans l'analyse au travers de l'implémentation d'une application client-serveur (Autocomplétion) en utilisant les Frameworks/toolkit retenus précédemment. Le choix repose d'une part sur le fait qu'une telle application est bien adaptée en s'interfaçant efficacement avec AJAX, et d'autre part sur le fait que cette fonctionnalité nous servira ensuite pour enrichir les services Web de l'Observatoire astronomique.

6.2 Application de base

Cette application est particulièrement importante, car dans les services Web de l'Observatoire les utilisateurs sont confrontés à ce type d'application, ne savent pas quoi taper pour les abréviations astrophysiques afin d'obtenir leur définition puis de faire leur requête sur les bases de données.

Autocomplétion : L'autocomplétion est un mécanisme très pratique permettant de suggérer une liste de choix pertinente et restreinte dans une entrée texte standard. Dans notre cas, elle permettrait de sélectionner facilement nos mots en tapant uniquement les premières lettres de ceux-ci.

6.2.1 Contraintes

Le texte libre reste le meilleur moyen pour des humains de communiquer avec des ordinateurs. La tendance semble aller vers plus de texte que précédemment à cause des systèmes instantanés. Lorsqu'elles se présentent devant une zone de texte libre, les personnes ne savent pas toujours ce qu'elles sont supposées taper. Bien beaucoup d'utilisateurs sont maintenant à l'aise avec un clavier, il y a toujours beaucoup d'utilisateurs pour lesquels le Web consiste en une partie de clics de souris. La vitesse de frappe reste un goulet d'étranglement même pour ceux qui tapent vite, la plupart pensant plus vite qu'ils ne peuvent taper. Les personnes peuvent commettre des erreurs lorsqu'elles tapent.

6.2.2 Objectif de l'Autocomplétion

Pour certaines personnes le principal bénéfice d'une suggestion est de taper moins de caractères, mais ce n'est pas le cas. En effet, c'est souvent plus lent d'entrer un simple mot en utilisant suggestion, puisque cela induit une distraction et requiert probablement une manipulation de la souris. Le principal bénéfice est d'offrir un ensemble fixe de choix pour une liste d'options standards.

6.2.3 Diagramme de séquence

- L'évènement est déclenché dans la page lorsque l'utilisateur tape le mot.

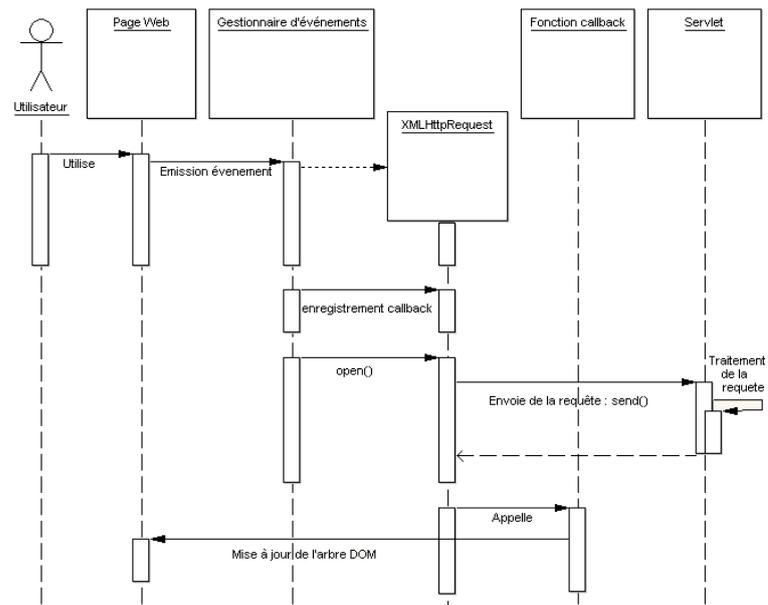


FIG. 6.1 – Diagramme de séquence : Autocomplétion

- Une méthode Javascript est appelée. Celle-ci crée un objet XMLHttpRequest, lui spécifie l'URL, le début de l'autocomplétion que l'utilisateur a tapé et lance le traitement.
- Le serveur après traitement, renvoie le fichier HTML ou XML selon un choix défini qui contient tous les mots qui commencent par ce que l'utilisateur a tapé.
- Une fois le fichier récupéré l'objet XMLHttpRequest appelle une méthode Javascript. Cette méthode vérifie le code d'erreur, puis affiche le contenu.

Septième partie

Réalisation

Chapitre 7

Réalisation

7.1 Mise en oeuvre

7.1.1 Autocomplétion

Solution

Suggérer des mots ou des phrases susceptibles de compléter ce que l'utilisateur est en train de taper. L'utilisateur peut alors sélectionner la suggestion et éviter de la taper en entier. Les résultats viennent d'XMLHttpRequest, l'entrée partielle va être téléchargée sur le serveur et ce dernier répond avec une collection de résultats probables.

Suggestion tire ses racines des "combo boxes" des champs de formulaires d'interface utilisateur traditionnelles qui combinent une entrée de texte avec une liste déroulante. Les éléments sont synchronisés. L'utilisateur est libre de taper n'importe quel texte et la sélection courante dans la liste va tracer ce qui a été tapé. L'utilisateur est également libre de choisir un élément de la liste qui sera mis en retour dans le champs de texte. Dans certains cas, la liste contraint ce que l'utilisateur tape ; dans d'autres cas, elles est là pour fournir uniquement des suggestions. C'est ce dernier style qui est devenu populaire ces dernières années. L'autocomplétion est devenue populaire avec Internet explorer 5 qui auto-complète des champs basés sur des historiques de l'utilisateur.

Les mécanismes de suggestion fonctionnent habituellement comme suit :

- Un champs standard est utilisé. Au même moment, un élément initial invisible est créé pour contenir les suggestions comme elles apparaissent. Le champ a besoin d'un gestionnaire d'évènements pour suivre le texte qu'il contient de façon à s'assurer que la liste surligne toujours la suggestion qui correspond.
- Au lieu de réaliser une requête de suggestion à chaque fois que l'on presse une touche, la solution la plus adoptée est que toutes les 100 millisecondes où à peu près le navigateur vérifie bien si rien n'a changé depuis la dernière requête. Si c'est le cas, le serveur passe la requête partielle comme un XMLHttpRequest Call basé sur un GET
- Le serveur utilise alors un algorithme pour produire une liste ordonnée de suggestions.
- En retour, le navigateur traite les suggestions et les affiche à l'utilisateur dans un format qui autorise la sélectionnés. Chaque entrée aura un gestionnaire d'évènements de manière a ce que si on clique sur l'entrée, le champ soit modifié.

7.2 Mise en place

Pour la réalisation de cette fonction, les moyens suivants ont été utilisés :

- L'IDE Eclipse WTP.²¹
- Le plugin ATF pour Eclipse : AJAX TOOLKIT Framework²²
- Le moteur de servlet : Apache Tomcat 4.1.24 ²³
- Le plugin Tomcat pour Eclipse.²⁴
- Le plugin Debug de Mozilla.²⁵

L'intégration des bibliothèques AJAX dans Eclipse a posé quelques problèmes, car le Toolkit ATF est encore en version beta. Mais par la suite j'ai pu les intégrer. L'installation est quasi identique pour tous les Frameworks sauf pour la bibliothèque DWR que je détaillerai par la suite.

Eclipse est un environnement de développement (Integrated Development Environment) dont le but est de fournir une plate-forme modulaire pour réaliser des développements informatiques. I.B.M est à l'origine du développement d'Eclipse qui est d'ailleurs toujours le cœur de son outil Websphere Studio Workbench (WSW) lui-même à la base de la famille des derniers outils de développement en Java d'I.B.M. Tout le code a été donné à la communauté par I.B.M.

Eclipse Web Tools Platform (WTP) une des caractéristiques d'applications Web est la multiplication des formats de fichiers que les développeurs sont amenés à manipuler. Le projet WebTools simplifie ces manipulations en fournissant des outils, notamment d'édition, pour les principaux types de fichiers dans des applications Web. Ces outils sont indépendants de J2EE²⁶ et peuvent être utilisés dans d'autres contextes. WTP est organisé en deux projets :

- Web Standard Tools (WST) : les applications J2EE utilisent des techniques qui ne sont pas propres au monde Java. Le but du projet WST est de fournir les outils pour manipuler ces technologies : XHTML, DOM, CSS, XML, XLT, Javascript, ..
- J2EE Standard Tools (JST) : Eclipse répond au besoin de base des développeurs en Java. Dans le cadre de développements J2EE des outils améliorant la productivité sont nécessaires. Le projet JST fournit des outils supportant le modèle applicatif de J2EE (Servlet), prenant en compte le modèle de déploiement et permettant de déboguer les applications J2EE de façon intégrée à Eclipse.

Courant 2006, un projet a rejoint le projet WebTools, AJAX Toolkit Framework (ATF), rattaché au projet WST, le sous-projet ATF a pour objectif de développer des outils utiles pour les développeurs d'applications AJAX.

ATF ce sous-projet a pour objectif d'ajouter des outils pour aider les développeurs d'applications AJAX. Les premiers travaux (Fig 7.1) :

- Amélioration de l'éditeur de Javascript notamment vérification de la syntaxe lors de la saisie.
- Ajout d'un débogueur de Javascript.

²¹<http://www.eclipse.org/webtools/main.php>

²²<http://www.eclipse.org/atf/>

²³<http://jakarta.apache.org/>

²⁴<http://www.sysdeo.com/eclipse/tomcatpluginfr>

²⁵<http://www.mozilla.org/>

²⁶Java 2 Edition Entreprise

- Intégration de Mozilla dans la zone éditeur Eclipse.
- Ajout de plusieurs outils permettant d'obtenir des informations sur la page affichée par le navigateur Mozilla embarqué
 - vue "DOM" inspector' qui détaille la structure de la page affichée.
 - vue "CSS" qui affiche les propriétés du style qui s'applique à l'élément sélectionné dans la vue "DOM inspector"
 - vue "DOM Source" qui affiche le code de l'élément sélectionné dans la vue "DOM inspector" et permet de modifier ce code et de voir le rendu immédiatement dans le navigateur mozilla embarqué.
- "XHR Monitor" cette vue affiche les requêtes asynchrones émises par l'application AJAX

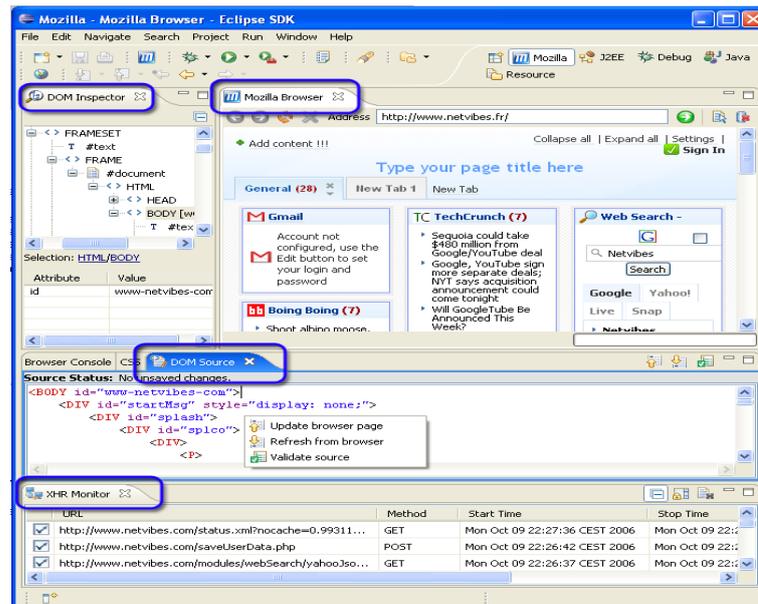


FIG. 7.1 – AJAX ToolKit Framework

TOMCAT est un serveur d'applications développé en Open source et qui implémente les technologies des Servlets et des JSP (Java Server Page). Les spécifications de ces technologies sont définies et sont rigoureusement respectées par Tomcat. La technologie des servlets consiste à exécuter sur un site Web du code Java précompilé. La servlet est écrite en Java avant d'être compilé puis placée dans un contexte d'exécution de Tomcat.

Dans cette section de réalisation nous allons juste aborder les différentes technique d'AJAX selon les bibliothèques. Le résultat de la réalisation de l'application Autocomplétion est la même pour toutes les bibliothèque comme nous le montre la figure (Fig 7.3).

7.2.1 La classe XMLHttpRequest

Nous avons utilisé cette classe afin de traiter tous les cas selon les différents navigateurs, car comme nous l'avons vu précédemment l'incompatibilité des navigateurs pose problème lors de la création de l'objet XMLHttpRequest voir figure (Fig 4.4).

7.2.2 Dojo

La bibliothèque Dojo est une boîte à outils Open Source visant à faciliter l'utilisation des technologies Javascript et DHTML. Dojo ne regroupe pas uniquement une collection de bibliothèques Javascript. Elle

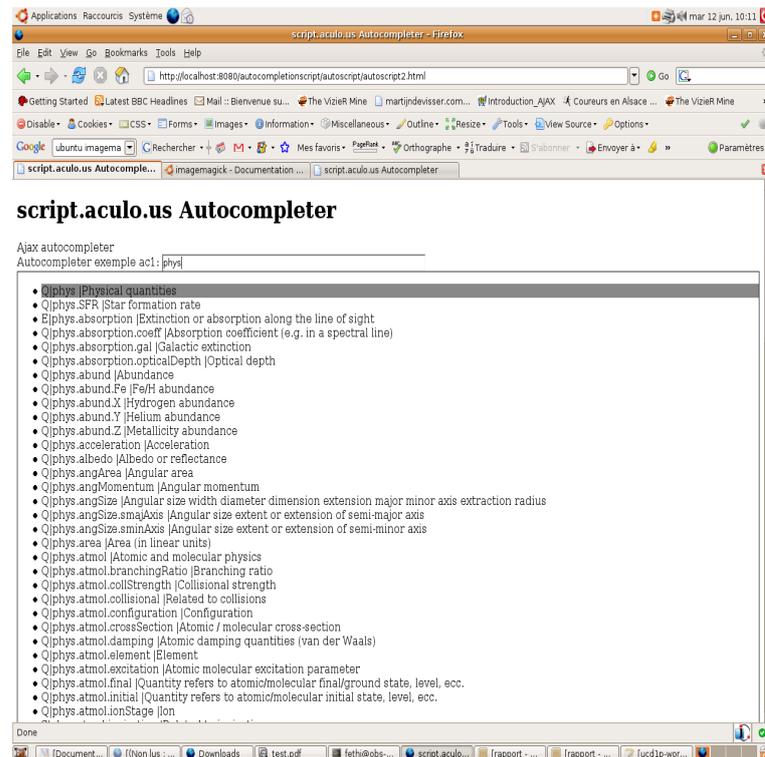


FIG. 7.2 – Application : Aut complétion

offre un gestionnaire évolué de modules ainsi qu'un cadre pour structurer les applications Web riches fondées sur la technologie Javascript.

Mécanisme de base

La bibliothèque est structurée en modules. Elle met en oeuvre un mécanisme de chargement à la demande de ces modules afin d'alléger le code Javascript de la bibliothèque utilisée au niveau navigateur. Ce mécanisme est récursif. Un module peut nécessiter le chargement d'autres modules et met en oeuvre la correspondance entre les noms de modules et leurs fichiers source ainsi que le transfert de ces fichiers et le chargement du code Javascript qu'ils contiennent comme nous le montre les schéma suivants :

Support des techniques AJAX

La bibliothèque dojo offre un intéressant support afin de mettre en oeuvre simplement les techniques AJAX. La fonction `dojo.io.bind` correspond à l'entité centrale du support d'AJAX par la bibliothèque dojo. Elle intègre tous les mécanismes permettant d'envoyer une requête à partir d'une page sans la recharger et de recevoir la réponse correspondante. Elle prend en paramètre un tableau associatif, qui fournit les différents attributs de configuration ainsi que les fonctions de rappel de la requête. Le tableau (Tab 7.1) récapitule les principaux attributs supportés par la fonction `dojo.io.bind`

Traitement du résultat d'une requête Les paramètres passés à la requête permettent de spécifier des fonctions de rappel afin de traiter les données envoyées. En cas de succès de la requête, dojo offre la possibilité d'utiliser la fonction de rappel `handle` ou `load`.

La fonction `load` permet de traiter exclusivement une requête exécutée avec succès. Cette fonction prend en paramètres les données et éventuellement une instance correspondant à l'évènement.

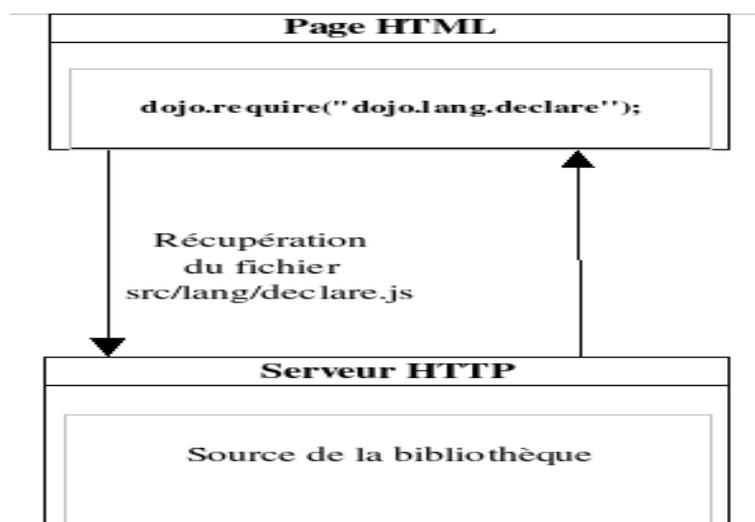


FIG. 7.3 – Gestion de modules

Attribut	Description
content	Contenu de la requête sous forme de tableau associatif .
formNode	Élément correspondant au formulaire à soumettre.
method	Méthode HTTP à utiliser afin d'envoyer la requête
mimetype	Type de contenu renvoyé par la ressource. La fonction gère de façon différente les différents types de contenus.
status	Spécifie si la requête est multipart, c'est-à-dire si elle contient plusieurs parties, pour l'envoi d'un formulaire. Cette fonctionnalité n'est pas supportée par toutes les couches de transport.
sync	Spécifie si la requête est synchrone. Les valeurs possibles sont true ou false.
timeoutSeconds	Délai d'attente maximal durant lequel la réponse doit être reçue.
transport	Spécifie explicitement la couche de transport à utiliser.
url	URL de la ressource.
useCache	Spécifie si un cache doit être mis en oeuvre.

TAB. 7.1 – Principaux attributs supportés par la fonction dojo.io.bind

Fonction	Paramètre	Description
error	Le type d'évènement et l'erreur survenue	Appelée en cas d'erreur lors de l'envoi de la requête.
handle	Le type d'évènement, les données et éventuellement l'instance de transport utilisée.	Elle appelée en cas de succès ou d'erreur de la requête dans le cas où les méthodes error et load ne sont pas spécifiées.
load	Le type d'évènement, les données et éventuellement l'instance de transport utilisée	Appelée en cas de succès de la requête lorsque les résultats de la réponse sont reçus
timeout	-	Appelée lorsque la réponse n'a pas été reçue dans le délai d'attente maximale spécifié. Elle doit être utilisée conjointement avec l'attribut timeoutSeconds

TAB. 7.2 – Fonctions de rappel supportées par la fonction dojo.io.bind

Conclusion La bibliothèque Dojo fournit d'intéressants mécanismes afin de mettre en oeuvre des applications Javascript évoluées. Intégrant un mécanisme de chargement à la demande des modules souhaités, elle permet d'alléger la quantité de code Javascript chargé dans le navigateur. La bibliothèque est structurée en modules adressant une fonctionnalité particulière mise à disposition correspondent a des fonctions facilitant la mise en oeuvre de Javascript, les types de base et les collections ainsi que les technologies DOM et AJAX.

7.2.3 Prototype

La bibliothèque Prototype offre de très intéressants mécanismes afin de faciliter l'utilisation de Javascript. Développée par Sam Stephenson²⁷, elle concerne différents aspects du langage Javascript, allant des éléments de base du langage jusqu'au support des collections et d'AJAX.

Support d'AJAX

La bibliothèque prototype offre un certain nombre de fonctionnalités afin de mettre en oeuvre AJAX. Elle permet notamment de communiquer de manière transparente avec un serveur via un canal HTTP, et ce de manière synchrone ou asynchrone. Pour cela, prototype utilise un objet d'envoi pour échanger des données.

Exécution de requêtes AJAX L'envoi et la réception de données en utilisant le support AJAX de prototype se fondent sur la classe Ajax.Request. Cette classe supporte les communications aussi bien synchrones qu'asynchrones. Elle détecte automatiquement l'objet à utiliser en fonction du navigateur pour le transport des requêtes et réponses. Le tableau (Fig 7.3) détaille la liste des options de la classe Ajax.Request.

Conclusion La bibliothèque prototype offre d'intéressants mécanismes visant à faciliter et alléger l'écriture d'applications Javascript. Le spectre des mécanismes mis en oeuvre s'étend de l'extension de classes fournies par le langage de base jusqu'au support d'AJAX.

7.2.4 Script.aculo.us

Cette bibliothèque Javascript réussit le véritable tour de force de mettre à disposition de petits objets Javascript faciles d'emploi, des effets visuels et des éléments d'interface utilisateur complexes, et cela, sur la plupart des navigateurs. Thomas Fuchs, est l'auteur de Script.aculo.us. On trouve une petite

²⁷<http://sam.conio.net/>

Options	Valeur	Description
asynchronous	true (par défaut) ou false	Permet de spécifier le mode d'exécution de la requête(synchrone ou asynchrone)
method	get ou post (par défaut)	Permet de spécifier la méthode avec laquelle est envoyée la requête AJAX.
onException	Fonction possédant deux paramètres, le premier de type Ajax.Request et le second de type Object	Permet de spécifier la fonction de rappel à utiliser lors de la levée d'une exception au cours du traitement d'une requête AJAX
onFailure	Fonction possédant deux paramètres, le premier de type XMLHttpRequest et le second de type Object	Permet de spécifier la fonction de rappel à utiliser en cas d'échec de la requête. Sa détermination s'appuie sur la propriété status de l'objet de transport. Cette option n'est utilisable qu'en mode asynchrone
onXXX	Fonction possédant deux paramètres, le type XMLHttpRequest et le second de type Object	Permet de spécifier des observateurs sur les différents événements du cycle de vie de la requête AJAX. Ce mécanisme n'est utilisable qu'en mode asynchrone. Les valeurs de "XXX" peuvent être Uninitialized, Loading, Loaded, Interactive et Complete.
onSuccess	Fonctions possédant deux paramètres, le premier type XMLHttpRequest et le second de type object	Permet de spécifier la fonction de rappel à utiliser en cas de succès de la requête. Sa détermination s'appuie sur la propriété status de l'objet de transport. Cette option n'est utilisable qu'en mode asynchrone
parameters	" " par défaut	En cas d'utilisation de la méthode HTTP get, permet de spécifier les paramètres utilisés pour la requête. La chaîne doit être au format suivant :[param1=valeur1& param2=valeur2]
requestHeaders	undefined	Permet de spécifier les en-têtes dans un tableau
postBody	undefined	En cas d'utilisation de la méthode HTTP post, permet de spécifier la chaîne à utiliser pour le corps de la requête

TAB. 7.3 – Option d'initialisation de la classeAjax.Request

innovation intéressante comme la bibliothèque Dojo la gestion de modules offerts pour exprimer les souhaits de chargement parmi ces modules.

Complétion automatique de texte Il existe des classes dédiées à ce domaine : `Ajax.AutoComplete` et `Autocomplete.Local`. Nous nous intéressons ici à la première, la seconde permettant une complétion à partir de données stockées côté client. Voici la liste (Tab 7.4) des options prises en charge par notre `Ajax.AutoComplete` :

Option	Description
<code>paraname</code>	Nom du paramètre à envoyer au serveur.
<code>tokens</code>	Permet la complétion incrémentale
<code>frequency</code>	intervalle d'examen pour la complétion, en secondes. Par défaut 0.4.
<code>minChars</code>	Nombre minimum de caractères à saisir avant de déclencher une complétion
<code>indicator</code>	ID ou référence directe sur l'élément
<code>parameters</code>	Permet de définir des paramètres supplémentaires à envoyer au serveur pour complétion, afin d'ajouter généralement un contexte de recherche
<code>asynchronous</code>	Détermine si la requête est asynchrone (true, par défaut) ou non (false).

TAB. 7.4 – Options prises en charge par `Ajax.AutoComplete`

7.2.5 Rialto

La bibliothèque Rialto offre la possibilité de développer des applications Web riches en se fondant sur une collection de composants graphiques. Cette bibliothèque se focalise toutefois davantage sur les applications Web pour l'informatique de gestion par l'intermédiaire de divers mécanismes dédiés. Réalisée initialement au sein de l'Institut Gustave Roussy (IGR), premier centre européen de lutte contre le cancer, à Villejuif, la bibliothèque Rialto a permis de bâtir une application de consultation des dossiers des patients accessible via le Web et désormais utilisée par l'ensemble du personnel médical de l'IGR. L'objectif principal de ce projet était de réaliser une application Web facilement consultable au sein de l'hôpital ainsi que de permettre un accès à distance aux informations d'un patient. Le second objectif de Rialto était d'offrir aux utilisateurs l'ergonomie riche à laquelle ils étaient habitués avec les applications de gestion en mode client-serveur déjà déployées sur le site.

Support des techniques AJAX La bibliothèque Rialto fournit la classe `rialto.io.AjaxRequest` afin de faciliter la mise en oeuvre des techniques Ajax fondées sur la classe `XMLHttpRequest`. La classe `rialto.io.AjaxRequest` englobe tous les mécanismes et le cycle de ce type de requête en mode asynchrone. Le tableau (Tab 7.5) récapitule les différents paramètres du constructeur de la classe `rialto.io.AjaxRequest`

Conclusion La bibliothèque Rialto offre d'intéressants mécanismes afin de mettre en oeuvre des applications de gestion aux interfaces Web riches. La bibliothèque propose une importante collection de composants graphiques. Cette bibliothèque facilite l'utilisation des techniques AJAX afin de soumettre les données de formulaires.

Paramètres	Type	Description
callbackObjectOnFailure	Un objet	Correspond à l'objet sur lequel est exécutée la fonction de rappel
callbackObjectOnSuccess	Un objet	Correspond à l'objet sur lequel est exécutée la fonction de rappel lors d'un succès
canBeCancel	un booléen	Spécifie si la requête peut être annulée au cours de son exécution
method	Une chaîne de caractères	Correspond à la méthode HTTP avec laquelle la requête est exécutée
OnFailure	Une fonction	Correspond à la fonction de rappel appelée lors de l'échec de la requête
OnSuccess	Une fonction	Correspond à la fonction de rappel appelée lors de succès de la requête
url	Une chaîne de caractères	Correspond à l'adresse de la ressource sur laquelle est exécutée la requête.
withWaitWindow	Un booléen	Spécifie si une fenêtre d'attente doit être affichée lors de l'exécution de la requête.

TAB. 7.5 – Paramètre du constructeur de la classe riatlo.io.AjaxRequest

7.2.6 DWR

Il est possible de développer des applications AJAX sans se soucier directement de l'objet XMLHttpRequest. Un kit de développement open source, Direct Web Remoting (DWR), délivre une couche logicielle pour la programmation de pages web du traitement de l'objet de requête. DWR propose également aux développeurs Java de créer des classes, pour accéder ensuite à des objets Java côté serveur depuis le code Javascript côté client (d'où le Web Remoting, un pilotage à distance). L'un des avantages de DWR est de ne plus soucier de l'objet XMLHttpRequest. Ce Framework inclut également des techniques simples pour déployer des éléments Web intégrant des données du serveur, sans pour autant nécessiter une connaissance approfondie de la programmation en Document Object Model. Le principal inconvénient de DWR réside dans l'obligation de disposer de composants côté serveur en Java, dans la mesure où DWR s'appuie sur des servlets et des objets Java pour fonctionner.

Mais à ce prix, on dispose d'un puissant outil permettant de tisser des liens entre des objets Java et du code Javascript. En d'autres termes, on peut mettre en place la logique de son application à l'aide d'objets Java sur le serveur, puis appeler les méthodes de ces objets grâce à du code Javascript quand le besoin s'en fait sentir. Par abus de langage, on dit qu'on fait "remoting" d'objets, ou plus exactement qu'on lance des appels à distance de méthodes Java à l'aide de d'objets Javascript (qui sont donc intrinsèquement liés aux objets Java du serveur, comme nous l'avons vu). Pour y voir plus clair, je détaille la mise en place de DWR et son intégration dans une application web.

Intégrer DWR à Eclipse Le code du Direct Web Remoting est disponible sous forme décompressée ou sous archive Java (JAR),dwr.jar. Télécharger DWR²⁸. Pour bien démarrer avec DWR, vous devez tout d'abord le mettre en place ; créer une application Tomcat avec Eclipse puis copiez le fichier dwr.jar /WEB-INF/lib de votre application Web Java sur le serveur, puis redémarrer ou relancer l'application.

7.2.7 Conclusion

Notre objectif principal était d'analyser la situation, la viabilité et la mise en oeuvre de l'ensemble des technologies AJAX. L'objectif secondaire était d'appuyer nos constatations à l'aide d'une applications Auto-complétion.

Au terme de cette application, j'ai pu me forger une vue d'ensemble pratiquement exhaustive sur le sujet, grâce à la lecture d'une quantité volumineuse d'articles, de tutoriaux et de livres. J'ai également acquis un savoir-faire grâce à la mise en pratique.

AJAX suscite un grand engouement mais a aussi ses détracteurs, comme c'est souvent le cas lors d'un "chamboulement" de concepts établis. Les attentes des utilisateurs évoluant avec le temps, la plate-forme Web est en train de rattraper son retard.

²⁸<http://www.getahead.ltd.uk/dwr/download.html>

Huitième partie

Service VizieR

Chapitre 8

Service VizieR

8.1 Carte VizieR Mine

8.1.1 VizieR

Les catalogues astronomiques

Un catalogue astronomique est un listing de descriptions d'objets célestes. Le catalogue est constitué de deux parties :

- une partie "métadonnées" qui indique l'auteur du catalogue, la période où les mesures ont été effectuées, la description de structures des données.
- une partie "données" qui liste les différentes mesures physiques

Ces catalogues se présente sous forme de fichiers ascii compressés. Leurs tailles varient entre quelques méga-octets et quelques giga-octets. Une application spécifique est associée à chaque catalogue pour la décompression et le traitement des informations.

Architecture existante

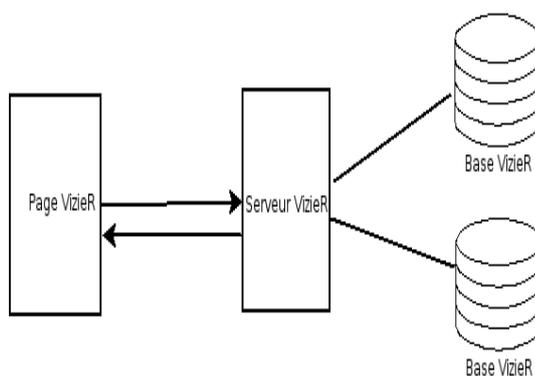


FIG. 8.1 – Architecture de la base VizieR

VizieR est une architecture logicielle qui permet à un utilisateur d'effectuer des requêtes des catalogues traitent de l'astronomie par le biais d'une page Web. Les catalogues sont triés suivant leur taille. En effet, les grands catalogues (dépassant 20 millions de tuples) ne sont pas dans la base de données Sybase.

8.2 Etude

8.2.1 Systèmes de coordonnées

Avant de rentrer dans la phase de Conception et de réalisation, un petit cours pour les débutants en astronomie s'impose afin de mieux comprendre la démarche du projet

La sphère Celeste

La sphère céleste est une sphère imaginaire de rayon quelconque et dont le centre est occupé par la Terre. Ce concept astronomique, hérite de l'antiquité, permet de représenter tous les astres²⁹ tels qu'on les voit depuis la Terre. Ainsi, il est possible de positionner les astres dans le ciel en leur attribuant des coordonnées uniques.

Equateur Celeste

En astronomie, l'équateur céleste est un grand cercle tracé sur la sphère céleste qui est la projection de l'équateur terrestre.

Ecliptique

L'écliptique est le grand cercle sur la sphère céleste représentant la trajectoire annuelle du soleil vue de la Terre. Il s'agit de l'intersection de la sphère céleste avec le plan écliptique, qui est le plan géométrique qui contient l'orbite de la Terre autour du Soleil.

Point vernal

Le point vernal est un des deux points de la sphère céleste où l'équateur et l'écliptique se croisent.

L'orbite :

L'orbite est la trajectoire que décrit dans l'espace un corps autour d'un autre corps sous l'effet de la gravitation.

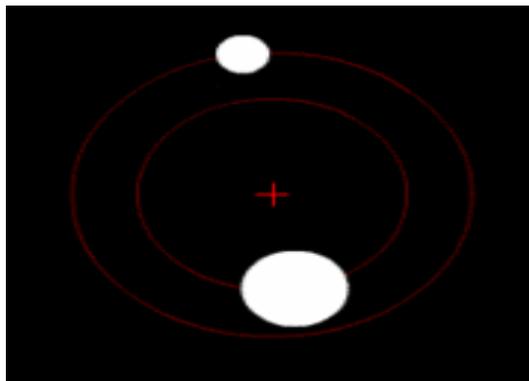


FIG. 8.2 – Orbite circulaire de deux corps de masses différentes autour d'un barycentre.

²⁹un astre, ou objet céleste désigne un corps céleste, ou plus généralement un objet de l'Univers.

L'ascension droite

cela correspond à la latitude pour des coordonnées terrestres. Il s'agit de l'angle entre la direction d'un astre et le plan de l'équateur comme nous le montre la figure (Fig 8.2) :

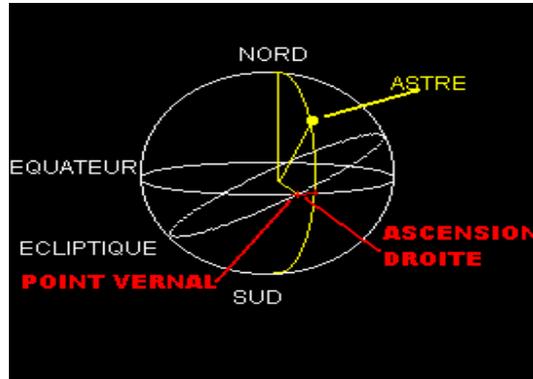


FIG. 8.3 – Ascension droite

Cette coordonnée est analogue à la longitude pour des coordonnées terrestres. Elle correspond donc à l'angle entre un point de référence (qui correspond à Greenwich sur Terre), le point vernal et la position de l'astre projeté sur l'équateur comme nous le montre (Fig 8.3) la figure ci-dessous

Déclinaison

Cela correspond à la latitude pour des coordonnées terrestres. Il s'agit de l'angle entre la direction d'un astre et le plan de l'équateur.

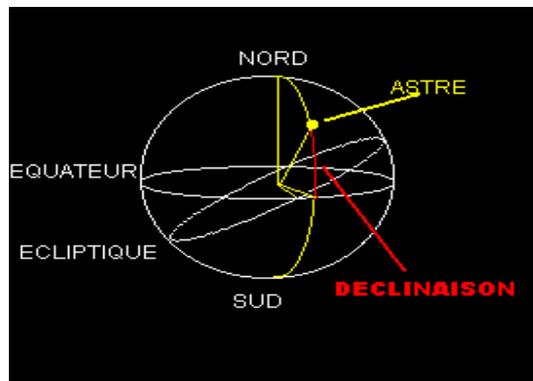


FIG. 8.4 – Déclinaison

Transformation en coordonnées galactiques

Le système de coordonnées galactiques est un système de coordonnées célestes qui prend en compte la rotation de la Galaxie sur elle-même. On parle ici de longitude et de latitude galactiques, notées respectivement l et b . Comme nous le montre la figure (Fig 8.4) suivant :

Transformation en coordonnées équatoriales.

On utilise le système de coordonnées équatoriales afin de repérer la position d'une étoile dans le ciel. Ce système utilise comme plan de référence la projection sur la sphère céleste de l'équateur de la terre.

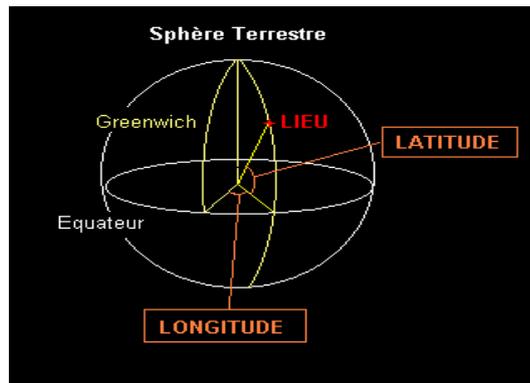


FIG. 8.5 – Position d'un observateur sur la sphère terrestre

On appelle cette projection l'équateur céleste. Le point de référence de ce cercle est le point vernal. Ce cercle est divisé en 24 heures (soit des divisions de 15 degrés chacune). L'angle mesuré entre la projection de l'objet sur ce cercle et le point vernal (en partant vers l'est de ce point) s'appelle l'ascension droite. Elle s'exprime en heures, minutes, secondes. L'angle entre cet équateur céleste et l'objet s'appelle la déclinaison. Elle se mesure en degrés, positive pour les objets situés dans l'hémisphère nord et négative pour les autres. L'axe des pôles de ce système coïncide donc avec l'axe de rotation de la Terre. Comme nous le montre le schéma ci-dessous (Fig 8.5)

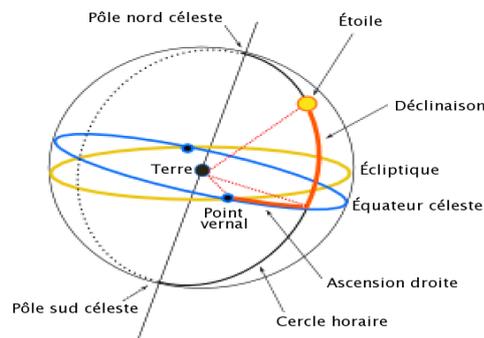


FIG. 8.6 – Système de coordonnées équatoriales

8.3 Phase d'analyse

The VizieR Mine Cette image du ciel est une représentation de la densité des objets accessibles dans VizieR, dans des coordonnées galactiques (le centre galactique est au milieu de l'image, la longitude galactique augmente vers la gauche). Chaque pixel de cette image représente un ensemble de catalogues. Les utilisateurs souhaitent voir apparaître quelque évolution afin de faciliter leurs recherches et de diminuer le temps de réponse afin d'obtenir les catalogues d'un point de l'image du ciel.

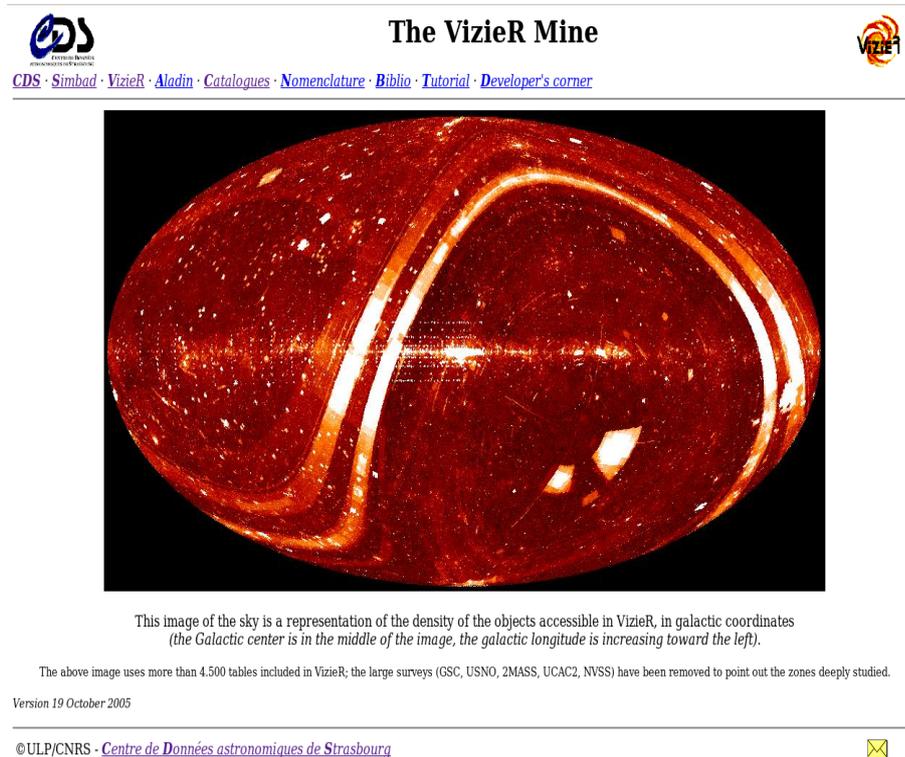


FIG. 8.7 – The VizieR Mine

8.3.1 Analyse de l'existant

Architecture actuelle

Fonctionnement de l'application

Lors du clic d'un utilisateur sur l'image, une requête est envoyée au service VizieR afin d'obtenir toutes les informations concernant ce point et nous recevons en retour une page HTML afin d'obtenir les catalogues. Nous pouvons consulter les catalogues comme nous le montre l'image ci-dessous :

Puis pour consulter un des catalogues il faut cliquer sur l'un des boutons comme nous le montre la figure qui nous donne une page VizieR Search figure 8.7 avec les coordonnées équatoriales, puis sur cette page nous pouvons interroger le Service VizieR avec d'autres options d'interrogation afin d'obtenir des informations sur ce catalogue comme nous le montre la figure 8.8

8.4 Phase de développement

8.4.1 Objectifs

Après cette phase d'analyse, on peut constater que l'utilisateur passe par plusieurs étapes afin d'obtenir des informations concernant le catalogue du point de l'image. Le but est d'améliorer l'interactivité de la carte en utilisant la technologie AJAX afin que l'utilisateur ait un ensemble de catalogues vues sur l'image et de calculer les coordonnées galactiques et équatoriales.

Catalogue Selection Page

CDS · Simbad · VizieR · Aladin · Catalogues · Nomenclature · Biblio · Tutorial · Developer's corner

Tokyo, Japan · IUCAA, India · CADAC, Canada · Cambridge, UK · CFA/Harvard, USA · UKIRT/Hawaii, USA · INASAN, Russia · Beijing Obs., China [Help]

97 catalogues found (obsoleted catalogues discarded)
having potential matches within 15' of 194.2323-58.2343(G) (J2000=02:54:59.1:13:44:13)

Reset All Show selected Catalogues OR Query selected Catalogues

<input type="checkbox"/>	I11	
<input type="checkbox"/>	I297	[53] ^(C) NOMAD Catalog (Zacharias+ 2005) (readMe)
<input type="checkbox"/>	I284	[51] ^(C) The USNO-B1.0 Catalog (Monet+ 2003) (readMe)
<input type="checkbox"/>	I305	[49] ^(C) The Guide Star Catalog, Version 2.3.2 (GSC2.3) (STScI, 2006) (readMe)
<input type="checkbox"/>	I252	[34] ^(C) The USNO-A2.0 Catalogue (Monet+ 1998) (readMe)
<input type="checkbox"/>	I271	[34] ^(C) The GSC 2.2 Catalogue (STScI, 2001) (readMe)
<input type="checkbox"/>	I246	[34] ^(C) 2MASS All-Sky Catalog of Point Sources (Cutri+ 2003) (readMe)
<input type="checkbox"/>	I254	[8] ^(C) The HST Guide Star Catalog, Version 1.2 (Lasker+ 1996) (readMe)
<input type="checkbox"/>	Bidenis	[8] ^(C) The DENIS database (DENIS Consortium, 2005) (readMe)
<input type="checkbox"/>	I289	[7] ^(C) UCAC2 Catalogue (Zacharias+ 2003) (readMe)
<input type="checkbox"/>	I304	[7] ^(C) Carlsberg Meridian Catalog 14 (CMC14) (CMC, 2006) (readMe)
<input type="checkbox"/>	I275	[3] ^(C) The AC 2000.2 Catalogue (Urban+ 2001) (readMe)
<input type="checkbox"/>	I297A	[2] ^(C) Tycho Input Catalogue, Revised version (Egret+ 1992) (readMe)
<input type="checkbox"/>	I259	[2] ^(C) The Tycho-2 Catalogue (Hog+ 2000) (readMe)
<input type="checkbox"/>	I280A	[2] ^(C) All-sky Compiled Catalogue of 2.5 million stars (Kharchenko 2001) (readMe)
<input type="checkbox"/>	VII05	[2] ^(C) 1.4GHz NRAO VLA Sky Survey (NVSS) (Condon+ 1998) (readMe) [Image/jpeg]
<input type="checkbox"/>	I98A	[1] ^(C) NLTT Catalogue (Luyten, 1979) (readMe)
<input type="checkbox"/>	I119	[1] ^(C) Southern Durchmusterung (Schoenfeld+ 1886) (readMe)
<input type="checkbox"/>	I31A	[1] ^(C) SAO Star Catalog J2000 (SAO Staff 1966; USNO, ADC 1990) (readMe)
<input type="checkbox"/>	I141	[1] ^(C) Yale Zone Catalogues Integrated (Yale Univ 1939:1983; ADC 1989) (readMe)
<input type="checkbox"/>	I171	[1] ^(C) ...

FIG. 8.8 – La selection des catalogues

VizieR Search Page

CDS · Simbad · VizieR · Aladin · Catalogues · Nomenclature · Biblio · Tutorial · Developer's corner

Tokyo, Japan · IUCAA, India · CADAC, Canada · Cambridge, UK · CFA/Harvard, USA · UKIRT/Hawaii, USA · INASAN, Russia · Beijing Obs., China [Help]

I/297 NOMAD Catalog (Zacharias+ 2005) [\(readMe\)](#) [\[Similar Catalogues\]](#)

1. I/297/out Example of the an output of NOMAD-1 (1117612732 rows)

The Naval Observatory Merged Astrometric Dataset (NOMAD) contains astrometric and photometric data for over 1 billion stars derived from the Hipparcos (I/239), Tycho-2 (I/259), UCAC2 (I/289), and USNO-B1.0 (I/284) catalogs for astrometry and optical photometry, supplemented by 2MASS (I/246) near-infrared photometry. An efficient remote query program `findnomad1` is available in the `cdsclient` package, for Unix/Linux platforms

Query Setup (usage)

Maximum Entries per table: 50 Output layout: HTML Table Output Order: [Reset All]

Query by Position on the Sky (Adapt Form to use a List of targets)

Target Name (resolved by SIMBAD) or Position: 211.0816+38.1554 Galactic Target dimension: 15 arcmin Submit Query

Position in Sexagesimal, or Decimal * Radius or Box size

Output preferences for Position:

Compute	r	x,y	Position	Galactic	J2000	B1950
<input type="checkbox"/>						

r and x,y are the distance to the Target. Position is in the same coordinate system as Target.

Query by Constraints applied on Columns

Show	Sort	Column	Constraint	Explain
<input type="checkbox"/>	<input type="checkbox"/>	NOMAD1	(char)	NOMAD-identifier (Note 1)
<input type="checkbox"/>	<input type="checkbox"/>	USNO-B1	(char)	Identifier in USNO-B1 (Cat. I/284) (Note 1)
<input type="checkbox"/>	<input type="checkbox"/>	UCAC2		^(a) Number in UCAC2 (Cat. I/289)
<input type="checkbox"/>	<input type="checkbox"/>	Tycho-2	(char)	Identification in Tycho-2 or Hipparcos (Note 2)
<input type="checkbox"/>	<input type="checkbox"/>	f_Tycho-2	(char)	[*] when distance NOMAD1/Tycho-2 > 1arcsec
<input type="checkbox"/>	<input type="checkbox"/>	YM	(char)	[YM.] Existence in YB6 or 2MASS (Cat. I/246)
<input type="checkbox"/>	<input type="checkbox"/>	RAJ2000	deg	Right ascension in ICRS, Ep=J2000
<input type="checkbox"/>	<input type="checkbox"/>	DEJ2000	deg	Declination in ICRS, Ep=J2000
<input type="checkbox"/>	<input type="checkbox"/>	r	(char)	[BCMYTH] Source of position (Note 3)
<input type="checkbox"/>	<input type="checkbox"/>	e_RAdeg	mas	Mean error on RAdeg (at epRA)

FIG. 8.9 – VizieR Search Pages

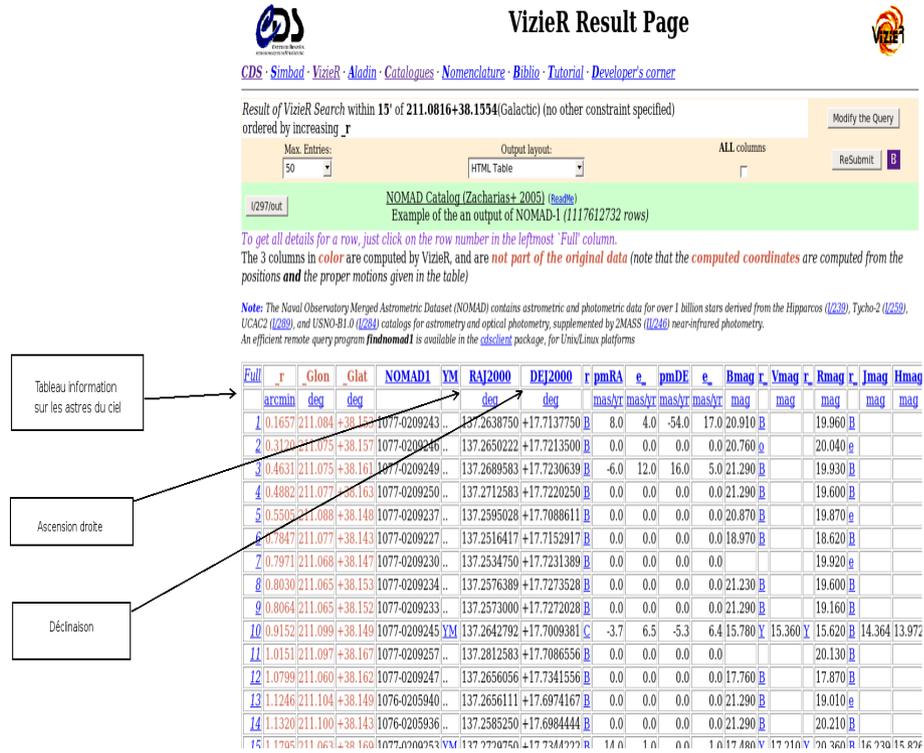


FIG. 8.10 – VizieR Result Pages

8.4.2 Phase d’implémentation

Pour concevoir cette application et après une analyse de la technologie AJAX, notre choix s’est basé sur la bibliothèque Dojo pour tout ce qui est IHM (Interface Homme Machine) et une base de données Sybase pour stocker les différents catalogues.

Choix technologiques

Le choix de la bibliothèque retenue pour cette application est la bibliothèque Dojo pour différentes raisons :

- Gestion des modules.
- Composant graphiques.
- Modules debugue.
- Modules profile.

Interfaces principales

Dans cette partie nous allons étudier la fenêtre principale ainsi que les opérations qu’effectuent ces dernières.

Coordonnées du système Dans cette partie le but est d’afficher les coordonnées galactiques et équatoriales qui sont calculées par rapport au pixel de l’image afin de permettre à l’utilisateur de connaître la position du point dans la sphère céleste. Dès que l’utilisateur bouge la souris sur l’image du ciel on obtient les coordonnées célestes comme nous le montre la figure suivante :

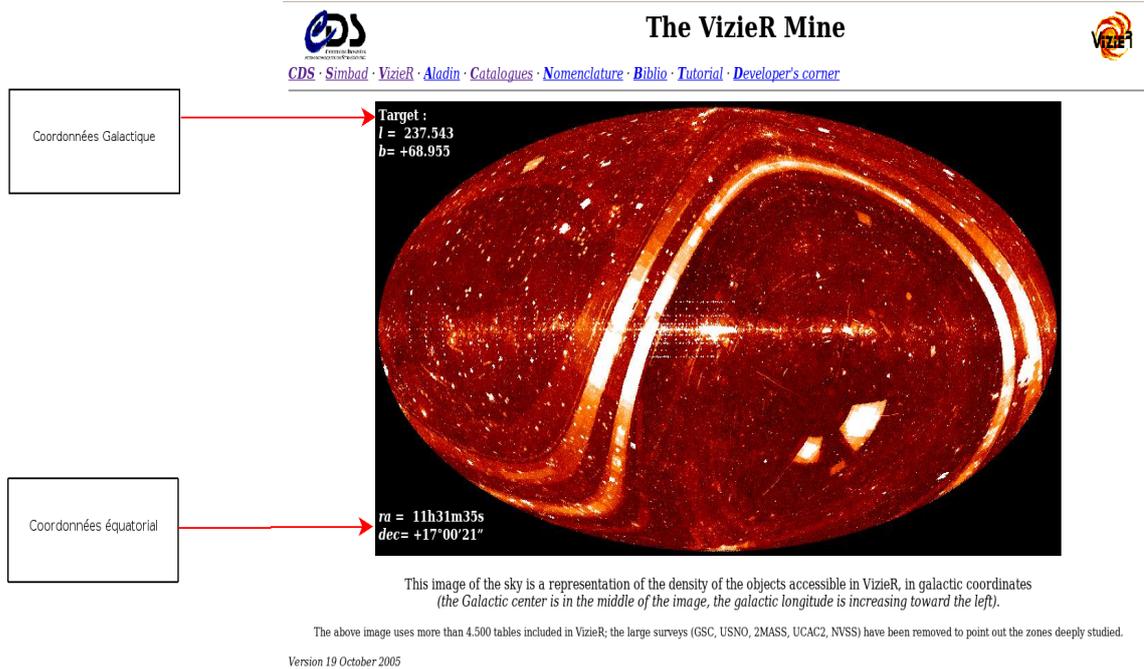


FIG. 8.11 – Systèmes de coordonnées céleste

Catalogues astronomiques Le second objectif est de récupérer toutes les informations du lieu où a cliqué l'utilisateur afin d'afficher les catalogues correspondant à partir du service VizieR. Comme nous le montre la figure

8.5 Test et Validation

En parallèle à la phase de développement, il a été nécessaire d'effectuer différentes sortes de tests et d'essais. Car l'un des problèmes majeur était l'incompatibilités des navigateurs. L'implémentation d'une fonction a été systématiquement suivie d'une phase de tests de non régression. Ce genre de test nous permet, grâce au debugger FireFox, au debugger et le profiler de la bibliothèque Dojo de localiser d'éventuels problèmes. Puis nous avons fait des tests sur presque tous les navigateurs les plus utilisés : Internet Explorer, FireFox, Opera, Camino, Konqueror . Nous avons décidé de conserver l'ancienne version de VizieR Mine pour les personnes possédant un navigateur incompatibles ce qui induit un surcoût de maintenance.

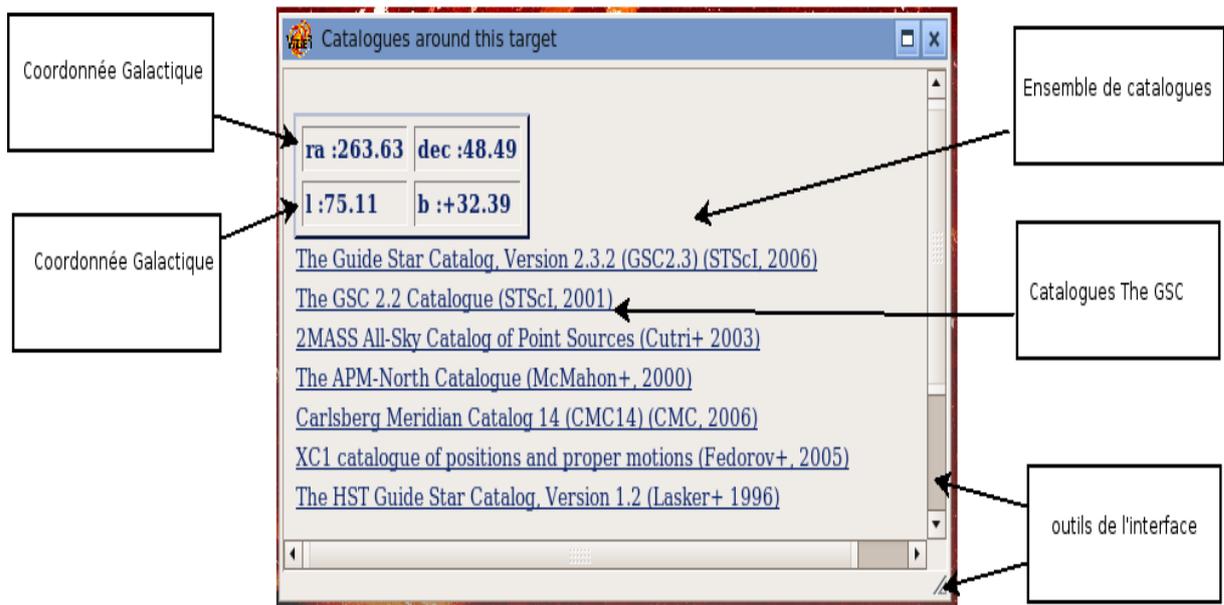


FIG. 8.12 – Analyse des catalogues

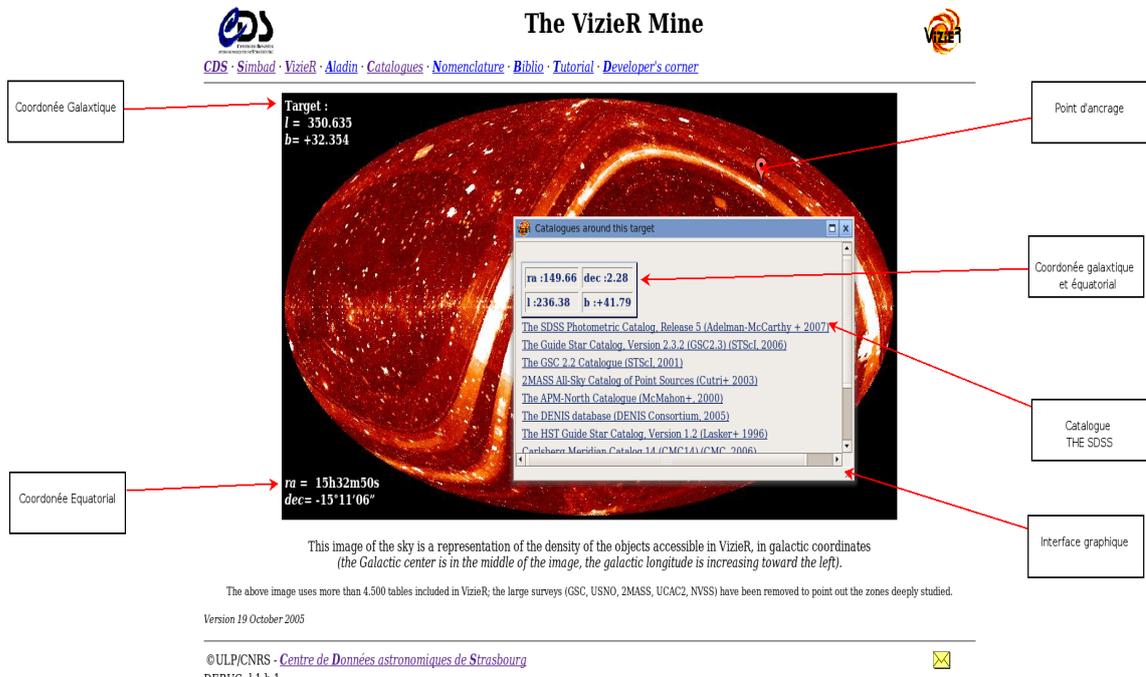


FIG. 8.13 – The VizieR Mine

Neuvième partie

Conclusion

Chapitre 9

Conclusion

Ce stage a été riche d'enseignement pour différentes raisons.

D'abord il m'a permis d'appliquer mes connaissances acquise lors de mon cursus universitaire afin de traiter différents problèmes informatiques. Ensuite ce stage a été l'occasion pour moi de découvrir :

- de nombreux aspects sur la technologie AJAX avec ses avantages et ses inconvénients.
- le choix des Framework AJAX.
- de nombreux aspects de l'architecture, du fonctionnement et de l'administration des logiciels tel qu'Apache et Tomcat.
- l'installation du ToolKit AJAX TOOLKIT Framework pour Eclipse avec ses différents plugins.
- analyse de différentes bibliothèques.

Aussi, le fait d'avoir eu à traiter un problème concret est une expérience très enrichissante qui me servira sans nul doute pour par la suite.

En ce qui concerne la gestion de projet, ce stage a été l'occasion pour moi de compléter mes connaissances. En effet, ayant tout au long du stage pu assister à des séminaires en astronomie et de faire des points d'avancement avec mes responsables.

Ce stage a également été bénéfique à titre personnel, car il m'a permis :

- de développer mon sens du relationnel et mon anglais, en étant en contact avec des personnes d'horizons et de nationalités différents.
- De voir le fonctionnement et les problématiques du Centre de Données de Strasbourg.
- La gestion de projet et l'organisation, le projet VO.

Sur le plan culturel j'ai été initié au monde de l'astronomie qui m'était inconnu. Ce domaine est très intéressant pour un informaticien car il fait appel à toute la puissance d'un système d'information aussi bien sur le plan matériel que logiciel. Plutôt habitué à des bases de données de l'ordre d'un millier de tuples alors que pour le domaine astronomique il est nécessaire d'enregistrer des dizaines de milliards de tuples.

Pour finir, je voulais encore remercier messieurs André Schaaff, Sébastien Derrière et M. Thomas Boch qui m'ont permis d'effectuer ce stage aux sein de leur équipe et aussi pour leurs conseils dans le domaine astronomique et Informatique, ainsi que M. François Ochsenbein M. Gilles Landais.

Liste des tableaux

3.1	Historique du Web.	14
4.1	Etats de l'instance de la classe XMLHttpRequest en mode asynchrone	20
4.2	Attributs de la classe XMLHttpRequest	21
7.1	Principaux attributs supportés par la fonction dojo.io.bind	42
7.2	Fonctions de rappel supportées par la fonction dojo.io.bind	43
7.3	Option d'initialisation de la classe Ajax.Request	44
7.4	Options prises en charge par Ajax.AutoComplete	45
7.5	Paramètre du constructeur de la classe riatlo.io.AjaxRequest	46
10.1	Informations concernant la bibliothèque Prototype	63
10.2	Informations concernant la bibliothèque Dojo	64
10.3	Informations concernant la bibliothèque Rico	64
10.4	Informations concernant la bibliothèque Sript.aculo.us	64
10.5	Informations concernant la bibliothèque JQuery	65
10.6	Informations concernant la bibliothèque Moo.fx	65
10.7	Informations concernant la bibliothèque MochiKit	65
10.8	Informations concernant la bibliothèque DWR	66
10.9	Informations concernant la bibliothèque Yahoo	66
10.10	Informations concernant la bibliothèque GWT	66
10.11	Informations concernant la bibliothèque Atlas	67
10.12	Informations concernant la bibliothèque Echo2	67

Bibliographie

- [AJA06] C. AJAX. *Survey of AJAX/JavaScript Libraries*. 2006.
- [App06] Apple. *Remote Scripting with IFRAME*. 2006.
- [Fet07] A. Fethi. *AJAX dans le contexte de services astronomiques*. Centre de Donnée de Strasbourg, Strasbourg, 2007.
- [Gar06] J. J. Garrett. *Ajax : A New Approach to Web Applications*. 2006.
- [JRD07] F. B. et Antoine Cailliau et Jean-Remy Duboc. *CSS Le guide complet*. Micro Application, 47 bis, rue des Vinaigries, 2007.
- [Mah07] M. Mahemoff. *Design Pattern*. O'REILLY, Paris, 2007.
- [McC06] P. McCarthy. *Ajax for Java developers : Ajax with Direct Web Remoting*. 2006.
- [NF06] B. C. et Nicolas Faugout. *Ajax Le guide complet*. Micro Applications, Paris, 2006.
- [Ste06] R. Steyer. *Ajax et PHP*. CampusPress, 47 bis, rue des Vinaigries, 2006.

Dixième partie

Annexes

Chapitre 10

Annexes

10.1 Comparatif des Frameworks

Nom	Prototype
Site Internet	http://www.prototypejs.org/
Auteur	Sam Stephenson (http://sam.conio.net/)
Licence	MIT (Massachusetts Institute of Technologie)
Compatibles avec les navigateurs	Internet Explorer 6.0 et plus, Mozilla FireFox1.0 Mozilla 1.7 et plus, Apple Safari 1.2 et plus
Documentation	Prototype API Documentation (http://www.prototypejs.org/api)
Tarification	Gratuite
Présentation	Synthaxe (http://particletree.com/features/quick-guide-to-prototype/)
Exemples d'utilisation	Prototype Windows Class (http://prototype-window.xilinus.com/samples.html)

TAB. 10.1 – Informations concernant la bibliothèque Prototype

Nom	Dojo
Site Internet	http://www.dojotoolkit.org/
Auteur	Alex Russell (http://alex.dojotoolkit.org/) Dylan Schiemann(http://www.dylanschiemann.com/)
Licence	Academic Free Licence
Version(s) Stable(s)	0.4
Compatibles avec les navigateurs	Internet Explorer 6.0 et plus, Mozilla FireFox1.0 Mozilla 1.7 et plus, Apple Safari 1.2 et plus
Documentation	API Documentation (http://www.dojotoolkit.org/api)
Tarification	Gratuite
Exemples d'utilisation	http://dojotoolkit.org/demos

TAB. 10.2 – Informations concernant la bibliothèque Dojo

Dojo fournit un ensemble de bibliothèques pour travailler plus facilement avec Javascript, DOM et HTML et autres :

Dojo.lang : Suite de fonctions pour rendre le Javascript plus facile à utiliser.

Dojo.io : Support relatif à la mise en oeuvre de la technologie AJAX.

Dojo.string : Fonctions de manipulation de DOM.

Dojo.style : Fonctions de manipulation de CSS.

Dojo.html : Fonctions de manipulation de HTML basée sur DOM.

Dojo.event : Gestion des événements (bibliothèque très utilisée pour la gestion des widgets)

Dojo.profile : Support permettant de mettre en oeuvre des tarces applicatives.

Dojo fournit de plus un ensemble de conteneur de données équivalent à ceux existant en Java

Dictionary : Objet dictionnaire (hash).

ArrayList

Queue : Objet FIFO (First In First Out)

Nom	OpenRico
Site Internet	http://www.openrico.org/
Auteur	Sabre Airline Solutions (http://www.sabreairlinesolutions.com/)
Licence	Apache 2.0 Licence
Version(s) Stable(s)	version (http://openrico.org/rico/downloads.page)
Compatibles avec les navigateurs	Internet Explorer 6.0 et plus, Mozilla FireFox1.0 Mozilla 1.7 et plus, Apple Safari 1.2 et plus
Documentation	API Documentation (http://www.openrico.org/rico/docs.page)
Tarification	Gratuite
Exemples d'utilisation	exemple (http://openrico.org/rico/demos.page)

TAB. 10.3 – Informations concernant la bibliothèque Rico

Nom	Scriptaculous
Site Internet	http://www.Script.aculo.us
Auteur	Thomas Fuchs [http://mir.aculo.us/]
Licence	MIT
Version(s) Stable(s)	1.6.4 [http://prototype.conio.net/dist/]
Compatibles avec les navigateurs	Internet Explorer 6.0 et plus, Mozilla FireFox1.0 Mozilla 1.7 et plus, Apple Safari 1.2 et plus
Documentation	API Documentation (http://wiki.script.aculo.us/scriptaculous/)
Tarification	Gratuite
Exemples d'utilisation	exemple [http://wiki.script.aculo.us/scriptaculous/show/Demos]

TAB. 10.4 – Informations concernant la bibliothèque Sript.aculo.us

Nom	JQuery
Site Internet	http://www.JQuery.com
Auteur	John Resig [http://ejohn.org/]
Licence	MIT/GPL
Version(s) Stable(s)	1.6.4 [http://docs.jquery.com/DownloadingjQuery]
Compatibles avec les navigateurs	Internet Explorer 6.0 et plus, Mozilla FireFox1.0 Mozilla 1.7 et plus, Apple Safari 1.2 et plus
Documentation	API Documentation (http://docs.jquery.com/Tutorials)
Tarification	Gratuite
Exemples d'utilisation	Exemple [http://mehdi.gimp4you.eu.org/BAS/jquery/tests/echange-contenu.html]

TAB. 10.5 – Informations concernant la bibliothèque JQuery

Nom	Moo.fx
Site Internet	http://www.JQuery.com
Auteur	Valerio Proietti [http://mad4milk.org/]
Licence	MIT
Version(s) Stable(s)	moo.fx.js+moo.fx.pack.js+prototype.lite.js
Compatibles avec les navigateurs	Internet Explorer, FireFox, Safari
Documentation	API Documentation (http://moofx.mad4milk.net/old/documentation/)
Tarification	Gratuite
Exemples d'utilisation	exemple [http://www.mad4milk.net/examples/mooajax/]

TAB. 10.6 – Informations concernant la bibliothèque Moo.fx

Nom	MochiKit
Site Internet	http://www.mochikit.com
Auteur	Bob Ippolito
Licence	MIT/Academic free License version 2.1
Version(s) Stable(s)	1.31
Compatibles avec les navigateurs	Internet Explorer 6.0 et plus , FireFox, Safari 2.0.2, Opera 8.5
Documentation	API Documentation (http://mochikit.com/download.html)
Tarification	Gratuite
Exemples d'utilisation	Exemple [http://mochikit.com/demos.html]

TAB. 10.7 – Informations concernant la bibliothèque MochiKit

Nom	DWR
Site Internet	http://getahead.ltd.uk/dwr/
Auteur	Joe Walker
Licence	Apache Software License v2
Version(s) Stable(s)	1.31 [http://getahead.ltd.uk/dwr/download]
Compatibles avec les navigateurs	Internet Explorer 5.5 et plus , FireFox 1.0, Mozilla 1.7 et plus, Safari 1.2 et plus, Opera 8.5
Documentation	API Documentation (http://getahead.ltd.uk/dwr/documentation)
Tarification	Gratuite
Exemples d'utilisation	exemple (http://getahead.org/dwr/examples/table)

TAB. 10.8 – Informations concernant la bibliothèque DWR

Nom	Yahoo! UI Librairy
Site Internet	http://www.developer.yahoo.com/yui
Auteur	Yahoo
Licence	BSD
Version(s) Stable(s)	2.2.0
Compatibles avec les navigateurs	Internet Explorer, FireFox, Safari, Opera
Documentation	API Documentation (http://developer.yahoo.com/yui/docs/)
Tarification	Gratuite
Exemples d'utilisation	exemple (http://developer.yahoo.com/yui/reset/)
support & communauté	http://tech.grups.yahoo.com./group/ydn-javascript

TAB. 10.9 – Informations concernant la bibliothèque Yahoo

Nom	Google Web ToolKit
Site Internet	http://www.code.google.com/webtoolkit
Auteur	Google
Licence	BSD
Version(s) Stable(s)	1.1.1.0
Compatibles avec les navigateurs	Internet Explorer 6 et plus, FireFox 1.0, Safari 1.2 , Opera
Documentation	API Documentation (http://code.google.com/webtoolkit/documentation/gwt.html)
Tarification	Gratuite
Exemples d'utilisation	Google Earth & Gmail

TAB. 10.10 – Informations concernant la bibliothèque GWT

Nom	Atlas
Site Internet	http ://www.atlas.asp.net
Auteur	Microsoft
Licence	BSD
Version(s) Stable(s)	2.2.0
Compatibles avec les navigateurs	Internet Explorer, FireFox, Safari, Opera
Documentation	API Documentation (http ://ajax.asp.net/)
Tarification	Gratuite
Exemples d'utilisation	PagesFakes (http ://www.pageflakes.com/)

TAB. 10.11 – Informations concernant la bibliothèque Atlas

Nom	Echo2
Site Internet	http ://www.nextapp.com/platform/echo2/echo
Auteur	NextApp
Serveur requis	Java
Licence	Mozilla Public Licence
Version(s) Stable(s)	2.2.0
Compatibles avec les navigateurs	Internet Explorer, FireFox, Safari, Opera
Documentation	API Documentation (http ://nextapp.com/platform/echo2/echo/doc/)
Tarification	commerciale
Exemples d'utilisation	exemple[http ://demo.nextapp.com/ExtrasTest/app]
support & communauté	http ://forum.nextapp.com/forum
outils	Echo Studio Visual Development Environnement

TAB. 10.12 – Informations concernant la bibliothèque Echo2

Tableau récapitulatif

	Prototype	Dojo	OpenRico	Script.aculo.us	JQuery	Moo.fx	Mookchit	DWR	Yahoo ! UI Library	GWT	Echo2	Atlas
Moteur AJAX	X	X	X	X	X	X	X	X	X			
Bibliothèque Framework de développement		X	X	X		X	X	X	X	X	X	X
Open Source	X	X	X	X	X	X	X	X	X		X	X
Propriétaire												
Principale caractéristique	Extension de javascript	Widgets	interface riche	effets visuels	Manipulation de DOM	Effets visuels	XML	traduction Java vers javascript	interface riche	traduction java vers javascript	traduction Java vers javascript	interface riche
Exécuté coté client		X	X	X	X	X	X	X		X		
Exécuté coté serveur											X	
Programmation Javascript	X	X	X	X	X	X			X			X
Programmation Objet								X		X	X	X

FIG. 10.1 – Tableau comparatif

10.1.1 Tableau comparatif

10.2 Installation du Framework ATF

10.2.1 Outils HTML

Créer des pages HTML, mettre au point des feuilles de style CSS ou encore développer du code JavaScript sont des tâches courantes lors du développement d'applications Web. WST propose des éditeurs de code source pour chacun de ces langages.

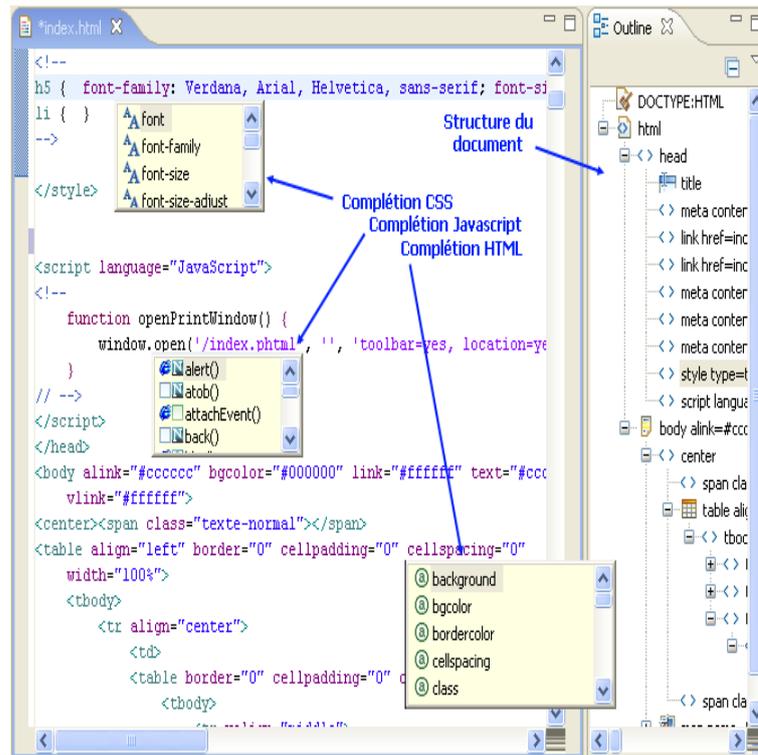


FIG. 10.2 – Outil ATF

10.2.2 Bibliothèque Dojo

Pour la création d'une application avec Dojo en utilisant le IDE Eclipse voici les étapes

10.2.3 Biblioth (e)que DWR

Configurer l'application Pour faire interagir DWR avec votre Javascript, vous devez d clarer dans web.xml une servlet Java que DWR va exploiter. Voici les instructions que vous devez ajouter   web.xml. Si ce dernier inclut d j des servlets, ajoutez DWR directement parmi celles-ci (il en va de m me pour l' l ment servlet-mapping) :

Remarque : red marrer votre application web Java pour le conteneur de servlets cr e une nouvelle instance de la servlet de DWR.

Cr er  galement un simple fichier XML pour d clarer les classes Java qu'on souhaite utiliser depuis le code Javascript c t  client. Maintenant nous allons voir comment exploiter les objets Javascript li s aux classes Java. Ce fichier doit s'intituler dwr.xml et on doit le ranger dans le dossier /WEB-INF/, aux c t  de web.xml :



FIG. 10.3 – Création d'un projet Tomcat

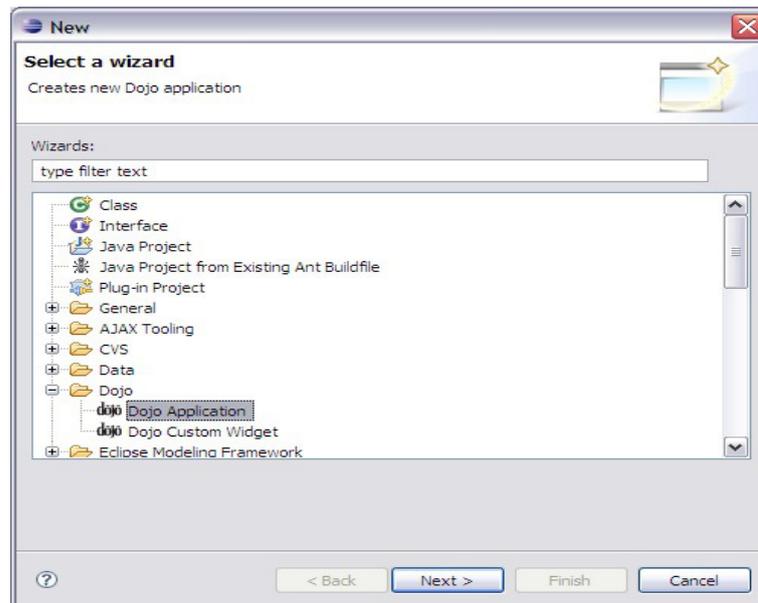


FIG. 10.4 – Création d'une appliaction Dojo

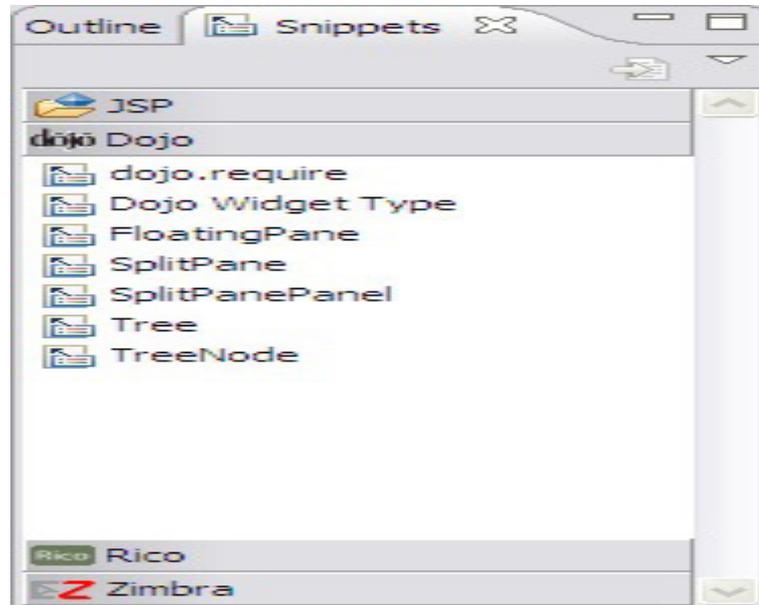


FIG. 10.5 – Palette Dojo

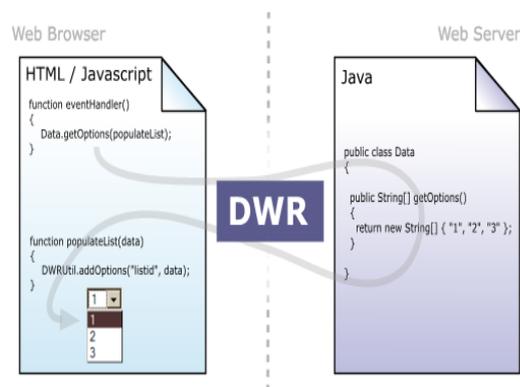


FIG. 10.6 – fonctionnement de DWR

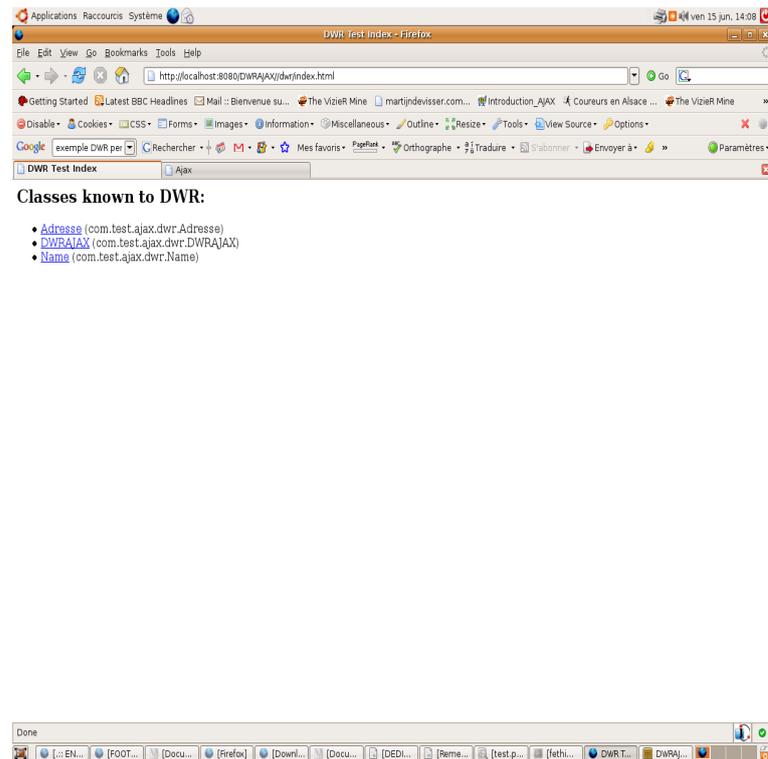


FIG. 10.7 – DWR dwr.xml en mode debugge

Ce XML permet au Javascript côté client d'appeler deux classes Java distantes. Les objets Javascript qui lient le code côté client aux classes Java s'intitulent ici . Parmi les opérations que qu'on doit mener côté serveur, on doit développ  la classe Java "com.AutoComplete" et l'avoir install e dans votre application. L'objet "Autocompl te" en sera alors une instance. La classe "Auto", en revanche, fait une partie du Java Software Development Kit : elle n'est pas de votre ressort.

Le fichier XML relie les noms Javascript aux objets Autocomplete, pour y acc der directement depuis le code c t  client. Ainsi, on peut appeler en Javascript toutes les m thodes publiques de ces objets Java. Mais comment le Javascript de la page web locale parvient-il acc der   distance aux instances Java qui tournent sur le serveur ? La figure montre l'appel   distance des m thodes Java par le Javascript, gr ce   DWR.

```

<servlet>
  <servlet-name>cgi</servlet-name>
  <servlet-class>org.apache.catalina.servlets.CGIServlet</servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
  </init-param>
  <init-param>
    <param-name>cgiPathPrefix</param-name>
    <param-value>WEB-INF/cgi</param-value>
  </init-param>
  <load-on-startup>5</load-on-startup>
</servlet>

  <servlet>
    <servlet-name>dwr-invoker</servlet-name>
    <display-name>DWR Servlet</display-name>
    <servlet-class>uk.ltd.getahead.dwr.DWRServlet</servlet-class>

    <init-param>
      <param-name>debug</param-name>
      <param-value>>true</param-value>
    </init-param>
<!-- changement pour la configuration de dwr.xml -->
    <init-param>
      <param-name>cgiPathPrefix</param-name>
      <param-value>WEB-INF/dwr.xml</param-value>
    </init-param>
<!-- The servlet mappings for the built in servlets defined above. Note -->
  </servlet>
<!-- ===== Built In Servlet Mappings ===== -->

```

FIG. 10.8 – web.xml du serveur tomcat

```

- <dwr>
- <allow>
- <create creator="new" javascript="DWRAJAX">
  <param name="class" value="com.test.ajax.dwr.DWRAJAX"/>
</create>
  <convert converter="bean" match="com.test.ajax.dwr.Personne"/>
  <param name="include" value="nom,datenaissance,adresse,numerotelephone"/>
- <create creator="new" javascript="Name">
  <param name="class" value="com.test.ajax.dwr.Name"/>
</create>
- <create creator="new" javascript="Adresse">
  <param name="class" value="com.test.ajax.dwr.Adresse"/>
</create>
</allow>
</dwr>

```

FIG. 10.9 – DWR dwr.xml en mode debugge

Methods For: DWRAJAX (com.test.ajax.dwr.DWRAJAX)

To use this class in your javascript you will need the following script includes:

```
<script type='text/javascript' src='/DWRAJAX/dwr/interface/DWRAJAX.js'></script>
<script type='text/javascript' src='/DWRAJAX/dwr/engine.js'></script>
```

In addition there is an optional utility script:

```
<script type='text/javascript' src='/DWRAJAX/dwr/util.js'></script>
```

Replies from DWR are shown with a yellow background if they are simple or in an alert box otherwise. The inputs are evaluated as Javascript so strings must be quoted before execution.

There are 15 declared methods:

- test();
- getInclude("");
- setPersonne();
- informationPersonnes("");
- getPersonnes();
- appelle();
- hashCode();
- getClass();
- wait(0);
- wait(0, 0);
- wait();
- equals();

(Warning: hashCode() is excluded: Methods defined in java.lang.Object are not accessible. See [below](#))

(Warning: No Converter for java.lang.Class. See [below](#))

(Warning: getClass() is excluded: Methods defined in java.lang.Object are not accessible. See [below](#))

(Warning: wait() is excluded: Methods defined in java.lang.Object are not accessible. See [below](#))

(Warning: overloaded methods are not recommended. See [below](#))

(Warning: wait() is excluded: Methods defined in java.lang.Object are not accessible. See [below](#))

(Warning: overloaded methods are not recommended. See [below](#))

(Warning: wait() is excluded: Methods defined in java.lang.Object are not accessible. See [below](#))

(Warning: overloaded methods are not recommended. See [below](#))

(Warning: wait() is excluded: Methods defined in java.lang.Object are not accessible. See [below](#))

(Warning: No Converter for java.lang.Object. See [below](#))

(Warning: equals() is excluded: Methods defined in java.lang.Object are not accessible. See [below](#))

FIG. 10.10 – fonctions DWR

Methods For: DWRAJAX (com.test.ajax.dwr.DWRAJAX)

To use this class in your javascript you will need the following script includes:

```
<script type='text/javascript' src='/DWRAJAX/dwr/interface/DWRAJAX.js'></script>
<script type='text/javascript' src='/DWRAJAX/dwr/engine.js'></script>
```

In addition there is an optional utility script:

```
<script type='text/javascript' src='/DWRAJAX/dwr/util.js'></script>
```

Replies from DWR are shown with a yellow background if they are simple or in an alert box otherwise. The inputs are evaluated as Javascript so strings must be quoted before execution.

There are 15 declared methods:

- test();
- getInclude("arit"); Qjarith |Arithmetic quantities
Sjarith.diff |Difference between two quantities described by the same UCD
Pjarith.factor |Numerical factor
Pjarith.grad |Gradient
Pjarith.rate |Rate (per time unit)
Sjarith.ratio |Ratio between two quantities described by the same UCD
Qjarith.zp |Zero point
- setPersonne();
- informationPersonnes("arit");
- getPersonnes();
- appelle();
- hashCode();
- getClass();
- wait(0);
- wait(0, 0);

(Warning: hashCode() is excluded: Methods defined in java.lang.Object are not accessible. See [below](#))

(Warning: No Converter for java.lang.Class. See [below](#))

(Warning: getClass() is excluded: Methods defined in java.lang.Object are not accessible. See [below](#))

(Warning: wait() is excluded: Methods defined in java.lang.Object are not accessible. See [below](#))

(Warning: overloaded methods are not recommended. See [below](#))

(Warning: wait() is excluded: Methods defined in java.lang.Object are not accessible. See [below](#))

(Warning: overloaded methods are not recommended. See [below](#))

(Warning: wait() is excluded: Methods defined in java.lang.Object are not accessible. See [below](#))

FIG. 10.11 – Test de fonctions de DWR

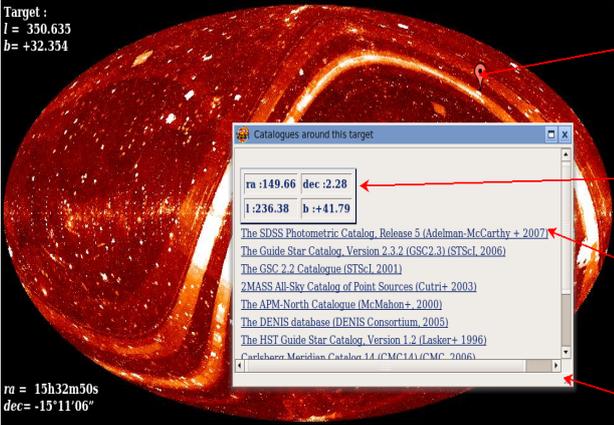
10.3 Carte VizieR Mine


The VizieR Mine


[CDS](#) · [Simbad](#) · [VizieR](#) · [Aladin](#) · [Catalogues](#) · [Nomenclature](#) · [Biblio](#) · [Tutorial](#) · [Developer's corner](#)

Coordonnée Galactique

Point d'ancrage



Coordonnée Equatorial

Coordonnée galactique et équatorial

Catalogue THE SDSS

Interface graphique

Target :
 $l = 350.635$
 $b = +32.354$

$ra = 15h32m30s$
 $dec = -15^{\circ}11'06''$

This image of the sky is a representation of the density of the objects accessible in VizieR, in galactic coordinates
 (the Galactic center is in the middle of the image, the galactic longitude is increasing toward the left).

The above image uses more than 4.500 tables included in VizieR; the large surveys (GSC, USNO, 2MASS, UCAC2, NVSS) have been removed to point out the zones deeply studied.

Version 19 October 2005

© ULP/CNRS - [Centre de Données astronomiques de Strasbourg](#)

FIG. 10.12 – The VizieR Mine