



Anne Camille Simon



Rapport de stage
effectué du 19 mars au 22 juin



VizieR : interface Javascript, Jquery



Maître de stage :
M. Gilles Landais

Responsable pédagogique :
M. Emmanuel Medina



Lp I2M 2011-2012

Remerciements

Avant de commencer la présentation de mon travail durant ces 3 mois de stage, je tiens à remercier les personnes grâce auxquelles cela fut possible.

Je désire, en premier lieu, adresser mes sincères remerciements à M. Gilles Landais, mon tuteur, qui m'a proposé ce stage et qui m'a encadré au quotidien. J'ai pu profiter de ses conseils et de ses connaissances sur la programmation mais également sur l'astronomie, domaine dont j'ai grandement approfondi mes connaissances durant ce stage.

Je remercie également M. Grégory Mantelet pour sa patience et qui m'a apporté son aide lorsque j'en ai eu besoin.

Merci à M. Abdellatif Bourjij, le responsable enseignant de l'IUT de Saint-Dié-des-Vosges, pour les conseils donnés lors de sa visite à l'observatoire.

Enfin je tiens à remercier l'ensemble des membres du CDS* et de l'Observatoire qui m'ont accueilli chaleureusement.

Les termes suivis d'une astérisque sont expliqués dans le glossaire.

Sommaire

Introduction	4
1. L'Observatoire	5
1.1. Présentation générale	5
1.2. Centre de Données astronomiques de Strasbourg (CDS)	5
1.2.1. Présentation	5
1.2.2. Organisation des services du CDS*.	7
1.3. Les services du CDS*	8
1.3.1 Simbad	8
1.3.2 VizieR	9
1.3.3 Aladin	9
1.3.4 Autres services	10
1.4. L'Observatoire Virtuel	10
2.VizieR	12
2.1 Présentation	12
2.2. Utilisation du service VizieR	13
2.3 Problématiques de VizieR	14
2.3.1. la volumétrie	14
2.3.2. L'homogénéité	14
2.3.3. Les évolutions technologiques	15
2.3.4. durées des requêtes.	15
3.L'interface	17
3.1. Résumé du besoin	17
3.2. Les technologies utilisées	17
3.2.1. Javascript/JQuery	17
3.2.2. ADQL*	18
3.2.3. TAP*	18
3.2.4. MVC	18
3.2.5. Java	19
3.3 La solution mise en place	19
3.1.1. Présentation des résultats et sélection des tables	20
3.1.2. Aide à la création de la requête	22
3.1.3. La requête	26
3.1.4. Exécution des requêtes et gestion des résultats	26
3.4. Les problèmes et leur résolution	28
3.3.1. International et miroir	28
3.3.2. La durée des requêtes	29
3.3.3. Volumes des résultats	29
3.3.4. Variété des tables	30
Conclusion	31
Glossaire	32
Annexes	34
Annexe A : Systèmes de coordonnées célestes	34
Annexe B : Cross-match	35
Annexe C : serveur TAP	36
Annexe D : MCD du meta-catalogue de la base de données de TAP VizieR	37
Annexe E : ADQL vers SQL	38

Introduction

Je devais effectuer un stage de 18 semaines en entreprise pour compléter ma licence professionnelle Internet et Média Mobiles. Ayant de plus grandes affinités avec le développement de site web qu'avec la création d'application pour Android ou iOS, c'est vers ce domaine que je souhaitais orienter mon stage. Le Centre de Données astronomiques de Strasbourg recherchant des stagiaires pour le développement d'interfaces web, c'est donc tout naturellement que j'ai postulé au sein de leur entreprise.

Le service VizieR du CDS* propose aux chercheurs un accès simple et facile aux tableaux de données astronomiques. Suite à la création des nouveaux outils pour interroger les bases de données avec des fonctions de géométrie céleste, il était nécessaire de mettre en place une interface pour permettre aux utilisateurs de les exploiter sur VizieR.

Le sujet de mon stage a donc été de concevoir une interface pour aider à la création de requêtes.

Après une présentation de l'Observatoire Astronomique de Strasbourg et, plus particulièrement, des services du CDS*, je présenterai VizieR. Pour finir je présenterai la solution qui a été mise en place, en expliquant la raison des choix que j'ai fait et les différents obstacles rencontrés.

1. L'Observatoire

1.1. Présentation générale



*Photographie de l'Observatoire de
Astronomie de Strasbourg*

L'Observatoire de Strasbourg est une Unité de Formation et de Recherche (UFR) de l'Université de Strasbourg. Il s'agit également d'une Unité Mixte de Recherche du CNRS* (Centre National de la Recherche Scientifique) et de l'Université Louis Pasteur (UMR* 7550). Jusqu'en 2008, le Planétarium représentait la partie publique de l'Observatoire pour la vulgarisation de l'astronomie.

Il est structuré en trois équipes de recherche et deux Services d'Observation de l'Institut National des Sciences de l'Univers (INSU). Les équipes sont :

- Hautes Énergies, étudiant l'astrophysique des hautes énergies, telles que les rayons-X. L'équipe Hautes Énergies est partie prenante du Survey Science Center du satellite XMM*-Newton ;

- Galaxies : les activités de l'équipe couvrent des problèmes variés concernant l'histoire et la formation des galaxies ainsi que l'étude des populations stellaires qui constituent ces galaxies;

- le Centre de Données astronomiques de Strasbourg (CDS*), présenté dans la partie 1.2 CDS;

Les deux services d'observation sont :

- le Survey Science Centre d'XMM*-Newton (SSC-XMM) : Le Survey Science Center (SSC) de la mission XMM*-Newton a pour objectif d'assurer le retour scientifique en réalisant une analyse systématique des observations, et de proposer des identifications des sources X.;

- le CDS* ;

1.2. Centre de Données astronomiques de Strasbourg (CDS)

1.2.1. Présentation

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. Il est à la fois une équipe de



recherche et un service d'observation au sein de l'Observatoire de Strasbourg. Parmi les bases de données gérées par le CDS*, on compte :

- Simbad, la base de données de référence pour l'identification et la bibliographie des objets astronomiques en dehors du système solaire ;
- VizieR, base de données permettant aux utilisateurs de naviguer à travers plusieurs milliers de catalogues* et de tables publiées ;
- Aladin, l'atlas du ciel.

Il maintient et distribue également le dictionnaire de la nomenclature des objets astronomiques. 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, organisant l'accès aux archives, etc. En outre, son but est de :

- rassembler toutes les informations utiles concernant les objets astronomiques disponibles sous forme numérique (données d'observation 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.

Par ailleurs, le CDS* contribue au XMM* Survey Science Center sous la responsabilité de l'équipe Hautes Énergies. Il a également signé des accords d'échanges internationaux avec les organismes suivants :

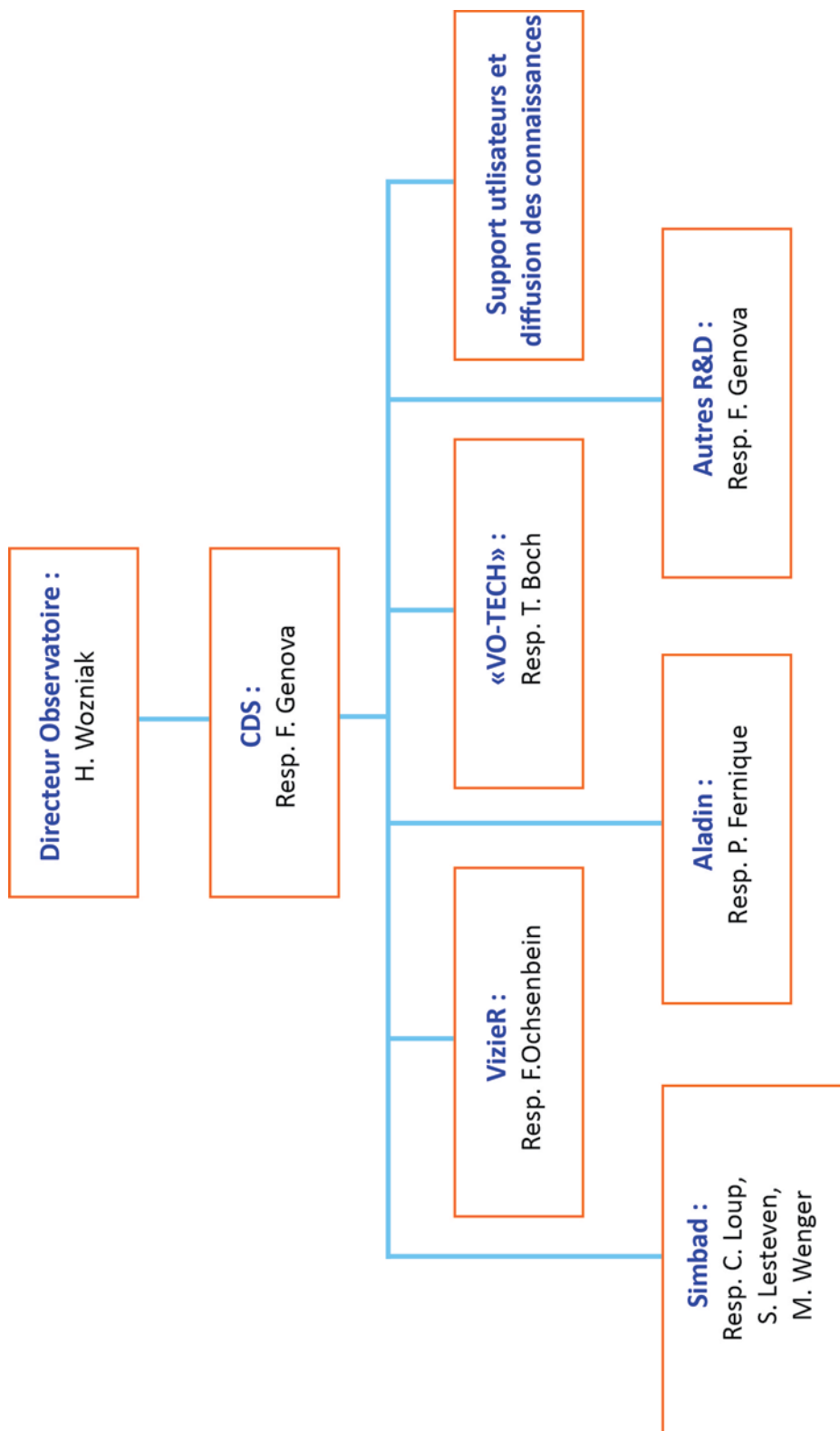
- la NASA (National Aeronautics and Space Administration) ;
- le National Astronomical Observatory (Tokyo, Japon) ;
- l'Académie des Sciences de Russie ;
- le réseau PPARC Startlink au Royaume-Uni ;
- l'Observatoire de Pékin (Chine) ;
- l'Université de Porto Allegre (Brésil) ;
- l'Université de La Plata (Argentine) ;
- InterUniversity Center for Astronomy and Astrophysics (Inde).

Il est membre de la Federation of Astronomical and Geophysical Data Analysis Services (FAGS), ou Fédération des Services d'Analyse de Données Astrophysiques et Géophysiques, dont les tâches consistent à recueillir en continu des observations, des informations et des données liées à l'astronomie, la géodésie, la géophysique et des sciences connexes, d'analyser, synthétiser et en tirer des conclusions; pour distribuer des données et publier les résultats obtenus. Il accueille des copies miroir de la base de données bibliographique NASA ADS (Système de Données Astrophysiques) ainsi que celles, européennes, des

1. L'Observatoire

journaux de l'American Astronomical Society (AAS). Il participe activement à l'élaboration de l'Observatoire astronomique virtuel.

1.2.2. Organisation des services du CDS*.



Organigramme minimaliste des différents services du CDS*

Le CDS* est composé d'une trentaine de personnes, travaillant dans un ou plusieurs services.

On compte quatre catégories de poste :

- Les experts scientifiques qui s'assurent de la cohérence et de la justesse des données stockées et mises à dispositions du public ;
- Les documentalistes dont le rôle est de saisir les données publiées dans des revues scientifiques, que ce soient des tableaux, des graphiques, des références ou bien des nomenclatures ;
- Le poste « logiciel » chargé du développement des sites et des applications proposés par le CDS* ;
- Les personnes en charge des images astronomiques ;

Ces groupes interagissent les uns avec les autres avec beaucoup d'efficacité permettant ainsi une véritable complémentarité entre les outils mis à la disposition du public pour la recherche de données astronomiques.

Grâce à ses performances, le CDS* est devenu, fin 2008, un TGIR (Très Grande Infrastructure de Recherche), ce qui le range au même niveau que des infrastructures internationales comme l'ESO* (European Southern Observatory) ou RENATER* (REseau NATIONAL de télécommunications pour la Technologie l'Enseignement et la Recherche) à l'échelon national.

1.3. Les services du CDS*

1.3.1 Simbad



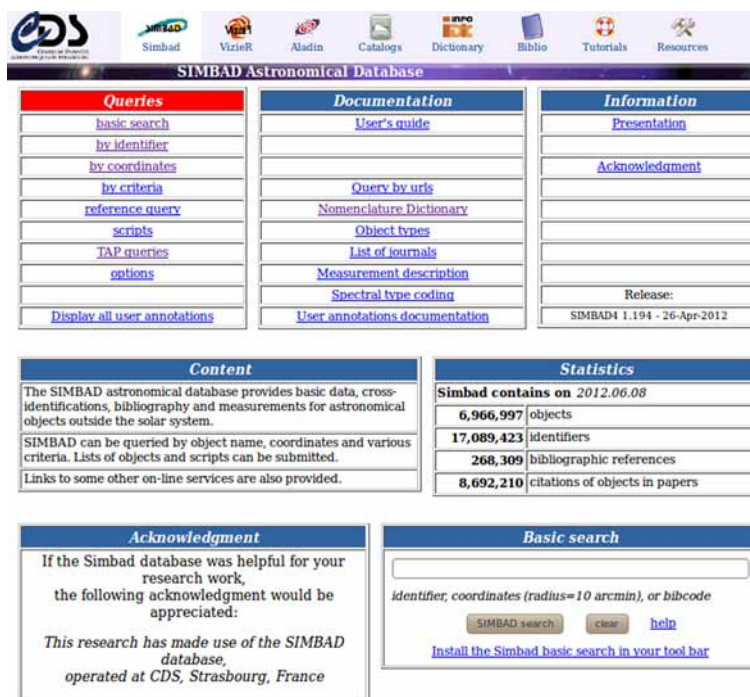
SIMBAD, Set of Identifications, Measurements and Bibliography for Astronomical Data, contient des informations sur plusieurs millions d'objets stellaires et non stellaires, excepté ceux de notre système solaire. Pour chaque objet figure les différents identificateurs connus, les références bibliographiques où cet objet est cité ainsi que quelques mesure (les coordonnées dans les différents systèmes, les mouvements propres, le type d'objet...). Il y a plusieurs mode de recherche : par critères, par position, par identifiants...

L'utilisateur peut choisir le format de sortie des résultats de la requête.

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, constamment développé et mis à jour depuis.

Disponible à cette adresse : <http://simbad.u-strasbg.fr/simbad/>

1. L'Observatoire



Page d'accueil du service Simbad

1.3.2 VizieR

VizieR donne accès à la bibliothèque la plus complète de catalogues* astronomiques et tableaux de données publiés et disponibles en ligne, organisée en une base de données auto-documenté. Des outils de requête permettent à l'utilisateur de sélectionner les tables de données pertinentes et d'en extraire et formater les résultats correspondant à des critères donnés.

Les données sont entrées dans un format homogène. De plus, une description standardisée du contenu des catalogues* permet leur inclusion dans un système de gestion de base de données (SGBD) relationnel.

Jusqu'à très récemment, en raison des performances, les grands catalogues* de plusieurs milliards de lignes ne pouvaient pas être traités au moyen d'un SGBD. Mais ils sont petit à petit insérés dans VizieR pour pouvoir les interroger comme tous les autres catalogues* ou tableau de données.

Il s'agit du service que j'ai intégré. Il est détaillé dans la partie 2. VizieR.

Disponible à cette adresse : <http://vizier.u-strasbg.fr/viz-bin/VizieR>

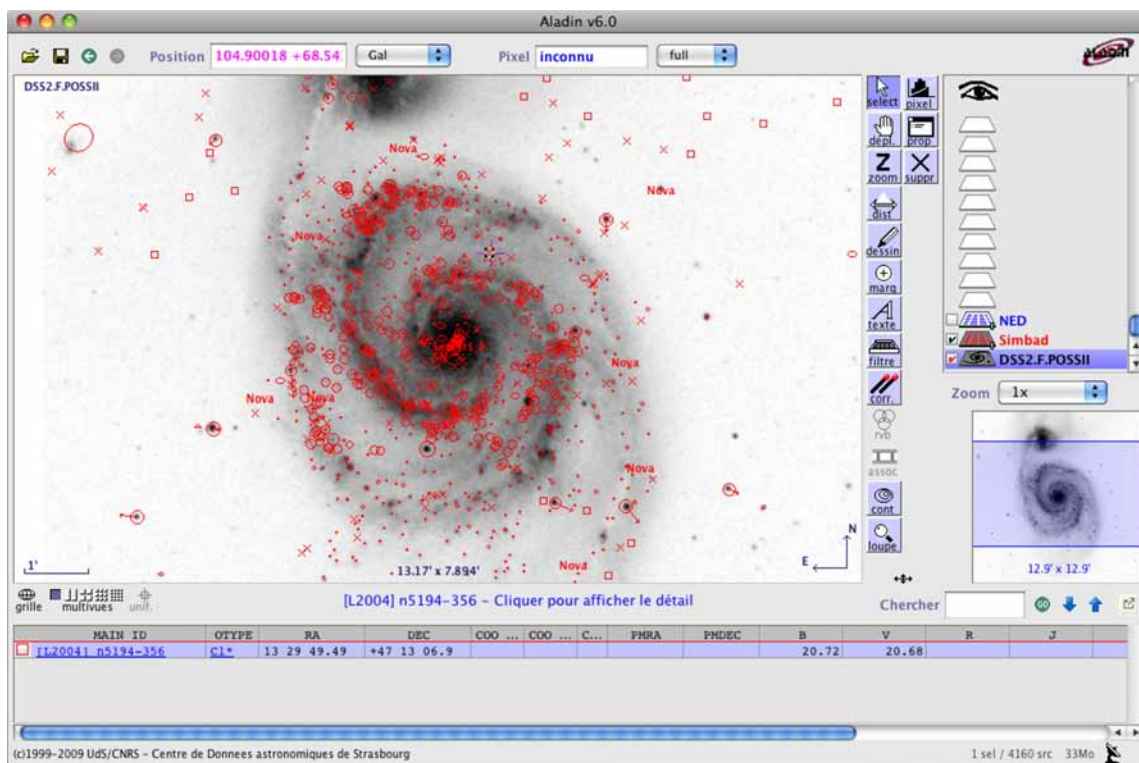
1.3.3 Aladin

Le logiciel Aladin, Access to Library and Database Information Network, est un atlas interactif du ciel. Il permet aux utilisateurs de visualiser des images digitalisées du ciel et d'y superposer des



informations issues de catalogues* ou de bases de données astronomiques. Il permet ainsi d'accéder de façon interactive à la plupart des données images et catalogues* accessibles par l'Internet et notamment aux données de SIMBAD et VizieR mais aussi d'autres données, locales ou situées sur des serveurs distants. Il est interconnectable avec d'autres outils de visualisation et d'analyse de données astronomiques.

Disponible à cette adresse : <http://aladin.u-strasbg.fr/aladin.gml>



Logiciel Aladin, vue du couple de galaxies M51 superposée de données Simbad en rouge

1.3.4 Autres services

Le CDS* offre d'autres services, par exemple les services bibliographiques et les services de type pages jaunes (Dictionnaire de nomenclature, Cross-match service, SimPlay...). Les ingénieurs travaillent activement sur le développement d'outils et de méthodes d'intérêt général pour l'échange de données et la standardisation.

1.4. 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

1. L'Observatoire

car elle permet un accès aisé à l'information, un croisement des données de toutes natures et elle conduit au concept d'Observatoire Virtuel (OV). 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'OV.

Les projets nationaux (Observatoire Virtuel Français (OBF) pour la France, German Astrophysical Virtual Observatory (GAVO) pour l'Allemagne, le National Virtual Observatory (NVO) pour les Etats-Unis...) et transnationaux (ESA* VO*, European Southern Observatory (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. À 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.

2. VizieR

2.1 Présentation

Le service VizieR, pour lequel j'ai créé une interface, regroupe la majorité des données tabulaires astronomiques qui sont publiées. Elle contient également les informations, analysées, qui ont été transmises par les grands télescopes ou par les satellites artificiels. Depuis sa création en 1996, VizieR est devenu un point de référence pour les astronomes du monde entier engagés dans la recherche, qui viennent accéder aux données régulièrement publiées dans des revues astronomiques.



Find catalogs among 10049 available

Clear Find...

Expand search

Wavelength **Mission** **Astronomy**

Radio	AKARI	AGN
IR	ANS	Abundances
optical	ASCA	Ages
UV	BeppoSAX	Associations
EUV	CGRO	Atomic_Data
X-ray	COBE	BL_Lac_objects
Gamma-ray	Chandra	Binaries:cataclysmic

Search by Position across 10430 tables

Target Name (resolved by [Sesame](#)) or Position: Clear J2000

Target dimension: 2 arcmin

Radius Box size

[More about VizieR](#)

Browsing modes: [Designation](#), [Acronyms](#), [Favorites](#), [Dates](#), [Image.spectra](#), [Kohonen](#)
Or list [the large surveys](#)

Thanks for acknowledging the VizieR Service ©UDS/CNRS Contact: ✉

Page d'accueil du service VizieR

La grande force de VizieR est l'homogénéisation des données insérées ainsi qu'une description standardisée des catalogues*, permettant leur insertion dans un système de gestion de base de données (SGBD) relationnelle. En effet les catalogues* ou les tableaux ne sont pas publiés par les mêmes chercheurs, et n'ont donc pas forcément les mêmes notations, les même noms de colonnes ou bien les même systèmes de coordonnées céleste*. Grâce à un système de metadata, des données décrivant d'autres données, il est possible de connaître le sujet d'un catalogue* ou bien la description du contenu d'une colonne, l'échelle utilisée...

Ces métadonnées sont stockées dans 3 tables principales et une trentaine de tables annexes :

- METAcat détaille les auteurs, les références, le titre et les explications de chaque catalogue* stocké.
- METAtab décrit chaque tableau de données stockées dans VizieR: légende de la table, le nombre de lignes, comment accéder aux données réelles, l'équinoxe et l'époque des coordonnées entre autre.
- METAcod détaille chacune des colonnes actuellement stockées dans VizieR: nom

2. VizieR

de la colonne ou l'étiquette, l'explication textuelle du contenu des colonnes, les types de données (numérique ou caractère) et le mode de stockage dans la base de données (entier ou à virgule flottante, longueur maximale des chaînes, etc.), unités dans lesquelles les données sont stockées dans la base de données et les unités dans lesquelles les données sont présentées à l'utilisateur, des formats édition, et quelques drapeaux utilisés pour les recherches (par exemple la colonne utilisée comme clé primaire) ou la présentation des données (par exemple la colonne à afficher dans la présentation par défaut du résultat).

Comme l'ensemble des tables META est lui-même décrit dans VizieR, le méta-dictionnaire peut être consulté et interrogé comme l'un des catalogues* stockés dans VizieR, permettant de localiser facilement par exemple des tables avec un grand nombre de lignes. (modèle MCD* en annexe D).

Dans le service VizieR actuel, les tables et les catalogues* de moins de 10 millions de lignes sont stockés dans une base de données tandis que les très grands catalogues* sont sauvegardés sous forme de fichier. Dans le nouveau service TAP* de VizieR, que je détaillerai plus tard, toutes les tables et les catalogues* stockés, le seront dans une seule base de données PostgreSQL*.

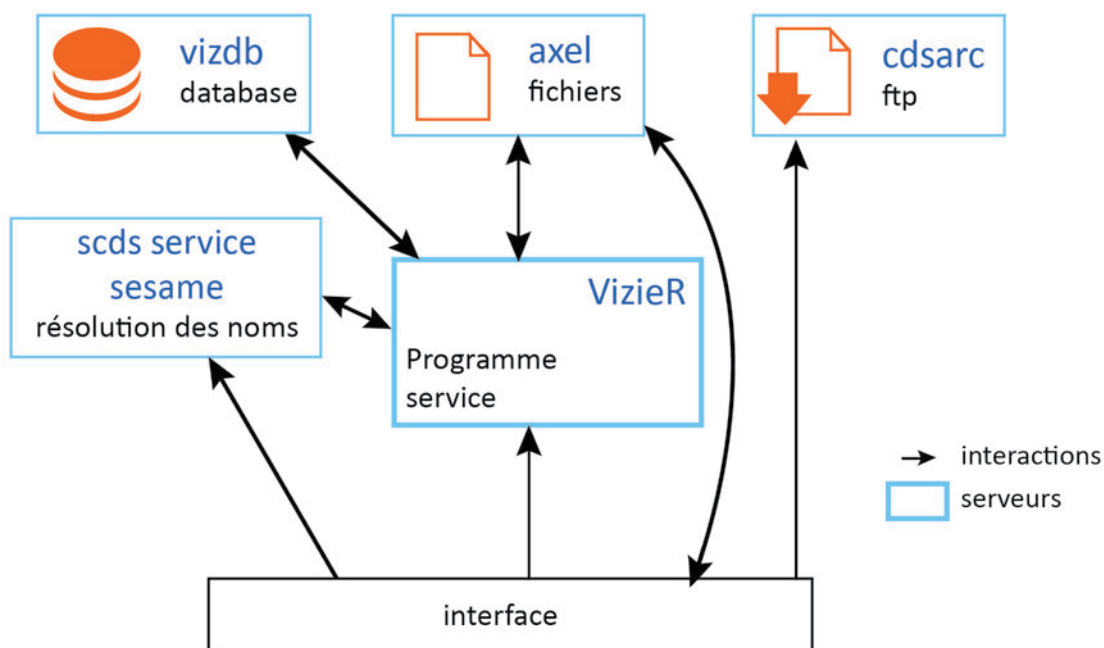


Schéma général de l'architecture du service VizieR

2.2. Utilisation du service VizieR

VizieR est mondialement connu. Il est donc fréquemment utilisé. Ainsi en moyenne, il y a 150 000 connexions par jour avec certains pics pouvant aller jusqu'à 4 millions. Cependant toutes ces connexions ne sont pas réalisées par des visiteurs du site ou de ces miroirs. Des

2. VizieR

programmes peuvent également se connecter pour interroger le service. Environ 1 000 visiteurs par jour se connectent au site pour une moyenne de 8 000 requêtes envoyées.

Il existe 8 miroirs de VizieR, répartis sur toute la surface du globe.



Utilisation de VizieR à travers le monde et position des miroirs.

2.3 Problématiques de VizieR

Du fait de la grande variété des données stockées, le service TAP* VizieR pose plusieurs problématiques majeures.

2.3.1. la volumétrie

La principale problématique posée par la base VizieR est la volumétrie.

Actuellement il y a plus de 10 000 catalogues* disponibles et plus de 20 000 tables de données. Cependant en raison des limites matérielles, les très grands catalogues* n'étaient pas entrés sous forme de table mais sous forme de fichier à télécharger. Maintenant ils sont petits à petits insérés dans VizieR, augmentant considérablement son volume. Ainsi alors que l'ancienne base de données fait 60 Gb, la nouvelle devrait atteindre les 3,5 Tb.

De plus, la taille des tables varie elle aussi. En effet le nombre d'entrées varie de quelques lignes (ex. : pour le catalogue* II/248, la table c2248t8 n'a que 6 lignes) à quelques milliards (ex. : pour le catalogue* I/284, la table usnob1 fait 1 045 913 669 lignes). Le nombre de colonne de chaque table n'est pas non plus fixe. De fait, la table c1231notes du catalogue* I/231 possède 3 colonnes alors que la table c9037xmm1src du catalogue* IX/37 a 242 colonnes. Il n'est donc pas étonnant de voir des catalogues* faire plusieurs centaines de Giga byte une fois inséré dans la base de données. C'est le cas par exemple de 2MASS qui atteint les 400 Gb et ce n'est pas le plus volumineux.

2.3.2. L'homogénéité

2. VizieR

En stockant des données provenant de partout, le CDS* se heurte à l'hétérogénéité des sources et de leur contenu.

Bien que le CDS* ait signé des partenariats avec des revues scientifiques pour avoir un format numérique standard dans les tables de données astronomiques, certains irréductibles s'obstinent à les publier sous le format de leur choix, obligeant les documentalistes à les adapter à la base VizieR.

Les différentes échelles de mesures posent également problèmes. En effet en fonction de l'analyse et de l'étude effectuée pour obtenir les résultats, il est possible d'avoir la même information sur différentes tables avec des échelles différentes, voire des unités différentes.

Enfin, un autre problème, mais non des moindre, est issu de l'époque de création des catalogues*. En raison du mouvement des étoiles, plusieurs systèmes de coordonnées* ont été définis. Il a donc été nécessaire de convertir les coordonnées en un même système, surtout pour les grands catalogues*, pour n'avoir qu'un seul type de coordonnées indexé. Un algorithme permet de créer deux nouvelles colonnes comportant les positions dans le système ICRS*, lors de l'insertion dans la base. Malheureusement ceci n'est pas fait pour toutes les tables. Lors de cross-match* sur des objets avec deux systèmes de coordonnées* différents, il faut convertir les positions en ICRS*, pour obtenir un résultat cohérent.

2.3.3. Les évolutions technologiques

Depuis sa création en 1996, VizieR n'a cessé d'évoluer en même tant que la technologie.

Tout d'abord, les évolutions concernent les catalogues* d'observation. En effet, la puissance et les capacités des outils utilisés de nos jours pour les observations permettent d'analyser plus de résultats et de repérer plus d'objets. Ainsi les grands catalogues* ont vu leur taille augmentée, passant de plusieurs millions de résultats à quelques milliards. Il a donc fallu gérer ces catalogues* d'une manière particulière.

Ensuite, l'équipe de VizieR s'est adaptée aux évolutions des langages pour insérer les tables de données dans la base. Il y a par exemple des descriptions de tables ou de catalogues* écrite en LaTeX*, avec la mise en forme incluse dans le texte.

Enfin, appartenant à l'IVOA*, la base de données doit s'adapter à certains standards mis en place par l'alliance et donc ajouter des informations sur les tables...

2.3.4. durées des requêtes.

Au vu de la volumétrie de la base ainsi que du nombre important de lignes par tables, les requêtes peuvent être très longues, surtout si l'on fait des jointures. Il est possible que leur exécution dépasse le quart d'heure par exemple pour un cross-match d'une table de 400 millions de lignes et une autre de 100 000.

Lorsque les tables de données ont beaucoup de lignes certaines de leurs colonnes sont indexées, notamment celle définissant les coordonnées de l'objet, sa magnitude* ou bien

2. VizieR

sa désignation au sein du catalogue*. En effet, en indexant les informations les plus utilisées par les chercheurs, l'exécution des requêtes est plus rapide. De même l'indexation des positions permet de faire des jointures par coordonnées ou cross-match*. Néanmoins, un cross-match* sur deux tables n'ayant pas le même système de coordonnées célestes* entraîne une conversion, si besoin, des positions dans le système ICRS*, annulant ainsi l'utilisation des indexes positionnels.

Actuellement pour les cross-match* sur les très grands catalogues*, il est possible de les effectuer grâce au site <http://cdsxmatch.u-strasbg.fr/xmatch>, un service du CDS* permettant l'identification croisée des sources entre deux catalogues*. Pour les catalogues* plus modestes, il n'y a pas de solution efficace.

3. L'interface

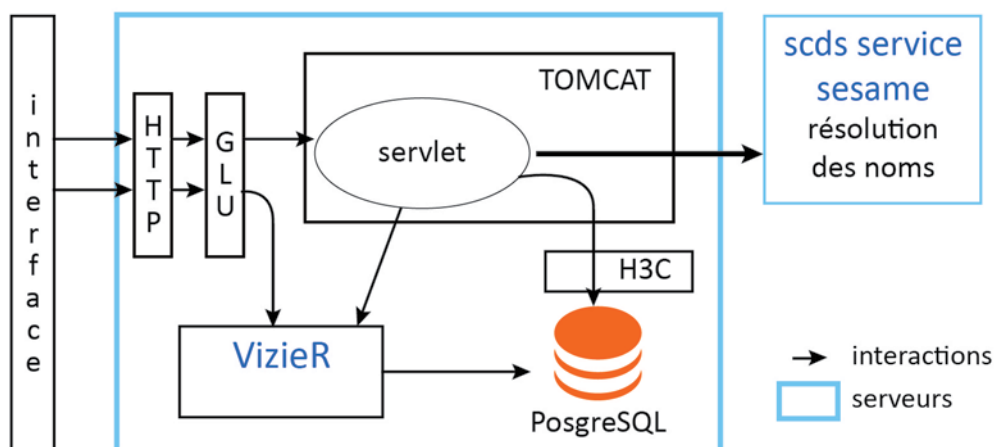
3.1. Résumé du besoin

Avec la création de l'ADQL*, détaillé en 3.2.2., les chercheurs ont eu la possibilité d'interroger les bases de données en utilisant des fonctions propres à l'astronomie. Quelques centres de données à travers le monde propose un service permettant d'écrire des requêtes en ADQL* pour leur bases de données. C'est le cas par exemple de GAVO (German Astrophysical Virtual Observatory) avec son service « ADQL* Query » http://dc.zah.uni-heidelberg.de/_system/_adql/query/form.

VizieR étant une référence mondiale, la demande pour un service de ce genre est grande.

La cible principale de ce service étant des chercheurs, nous voulions privilégier l'aspect professionnel de l'interface et avons décidé de lui donner un aspect relativement simple pour se concentrer sur les performances et les résultats.

L'architecture du VizieR actuelle n'est pas celle utilisée par le service TAP* VizieR. Un nouveau serveur, hébergeant entre autre la base de données, a été mis en place.



Architecture pour le service TAP VizieR. (cf annexe C)

3.2. Les technologies utilisées

3.2.1. Javascript/JQuery

Pour l'interface utilisateur du service TAP* VizieR, j'ai utilisé le langage de programmation JavaScript qui est interprété par le navigateur. En effet il ne pose pas de trop gros problème de compatibilité avec les différents navigateurs et permet une manipulation du DOM* très facile.

JQuery, une librairie JavaScript, a également été utilisée afin de pouvoir utiliser certains plugin mais aussi et avant tout pour la simplicité des requêtes Ajax et de l'asynchrone.

3. L'interface

3.2.2. ADQL*

Pour l'interrogation de la base de données, il faut utiliser un langage spécial, conçu pour l'astronomie.

L'ADQL*, ou Astronomical Data Query Language (langage de requête de données astronomiques), est utilisé par le VO* pour envoyer des requête astronomiques aux services de l'IVOA*. Ce langage est basé sur de SQL auquel a été ajouté de nombreuses fonctions géométriques.

Cela signifie qu'il est possible de créer des requêtes en SQL basiques, avec des select, des jointures, des contraintes sur les colonnes et des colonnes calculées. Mais il est également possible de créer des zones du ciel définies par des points. Cela permet de n'effectuer la recherche que sur l'aire définie, plutôt que sur tout le ciel, ou bien de faire des cross-matches. Ce langage existe dans sa version stable depuis le deuxième semestre 2008. Il est très récent et assez peu connu.

Pour le service VizieR, la librairie Java utilisée pour gérer les requêtes côté serveur a été développée par le CDS*. Elle peut être trouvée à l'adresse : <http://cds.u-strasbg.fr/resources/doku.php?id=adqlib> Elle sert à transformer la requête ADQL créée par l'utilisateur en une requête SQL interprétée par le système de gestion de la base de données. (Cf Annexe E : ADQL vers SQL)

3.2.3. TAP*

Le protocole d'accès au tables (TAP*) définit un protocole de service pour accéder de façon générale aux données de tables, y compris les catalogues* astronomiques ainsi que des tables de bases de données générales. Il supporte les requêtes synchrones et asynchrones. Ce protocole permet plusieurs langage de requêtes dont l'ADQL*. Également développé au sein de l'IVOA*, il a pour but de standardiser les accès aux bases de données afin de permettre une utilisation croisée des multiples services TAP*.

Pour VizieR, la librairie Java TAP*, créée au sein du CDS*, est composé de deux API :

- l'API ADQL*, (cf 3.2.2. ADQL*) ;
- la librairie UWS (*Universal Worker Service*), également un standard de l'IVOA*, permet de gérer les requêtes asynchrones grâce à au protocole REST.Elle aussi est développée au sein du CDS* et peut être trouvée à l'adresse : <http://cds.u-strasbg.fr/resources/doku.php?id=uwslib>

Cette librairie peut être trouvée à l'adresse suivante : <http://cds.u-strasbg.fr/resources/doku.php?id=taplib>

3.2.4. MVC

Pour gérer les différentes interactions de l'utilisateur avec l'interface, un Modèle-Vue-Contrôleur a été mis en place. Un patron spécial a été mis en place pour les requêtes

3. L'interface

asynchrones.

3.2.5. Java

Le serveur est en Java, un langage de programmation orienté objet. Il s'agit donc du langage utilisé pour les servlets*. Ces dernières sont des composants web conçus sous forme d'une classe Java, qui est la structure d'un objet, permettant de créer dynamiquement des données au sein d'un serveur HTTP.

3.3 La solution mise en place

The screenshot displays the Tap VizieR service interface. At the top, there are navigation icons for ODS, SIMBAD, VizieR, Aladin, Catalogs, Dictionary, Biblio, Tutorials, and Resources. Below these is the title "Tap VizieR" and a description: "The TAPVizieR service provides VizieR tables using the Astronomical Data Query Language (an SQL extension in Astronomy). Search by catalog, author's name, word(s) from title, position (resolved by *Sesame*), ... e.g.: Veron, 2Mass, redshift, HD2012...".

A search bar contains the text "redshift" and a "search" button. Below it, there are tabs for "all", "by wavelength", "by mission", "by astronomy", and "hide results". A list of expandable sections includes "columns and constraints", "sky area", "joins", and "unit change of coordinates".

At the bottom of the search section, there are radio buttons for "Max records" (set to "all") and a "limit" input field set to "100", along with a "create query" button.

The query editor shows the following SQL query:

```

1 -- output format : votable
2 SELECT TOP 100 c341404873.z, c341404873.Q_E, c341404873.Nee, c341404873.RAJ2000, c341404873.OHAG, c34
3 c341404873.Sep, c341404873.dar, c341404873.DEL2000, c341404873.Xr, c341404873.recno, c341404873.Type
4 FROM c341404873
5 WHERE c341404873.z:=1

```

Below the query editor, there is a "Query name" field with "tap Query", an "Output format" dropdown set to "votable", and buttons for "run", "Quickview", and "reset".

The "List of your TAP queries" section includes "Refresh", "Abort", and "Destroy" buttons. A search bar is present above a table with the following data:

ID	name	phase	start	destruction	results
1338980982530A	tap Query	COMPLETED	undefine	undefine	result
1338988899042A	c341404873	COMPLETED	15:21:39 on 06/06/2012	15:21:39 on 11/06/2012	result

At the bottom, it shows "Showing 1 to 2 of 2 entries" and navigation buttons: "First", "Previous", "Next", "Last".

Page d'accueil du service Tap VizieR

Disponible à l'adresse : <http://tapvizier.u-strasbg.fr/adql/>

3. L'interface

L'interface retenue est intuitive et permet à des néophytes mais également à des personnes connaissant bien l'ADQL* d'interroger la base de données du service VizieR.

3.1.1. Présentation des résultats et sélection des tables

Dans la base de données, catalogues* possèdent, entre autre, une description ainsi que des mots clés de trois catégories différentes : « wavelength » ou longueur d'onde, « mission » et « astronomy » qui représente les différents termes astronomiques attachés au catalogue*. Les tables sont caractérisées par une description, le nombre de lignes, la présence ou non de positions principales* et éventuellement une image dans le service Alaska. Ce service de Aladin permet d'obtenir des footprints (ou empreintes) de la zone du ciel concerné par la table.

Il a donc fallu présenter toutes ces informations de façon claire et cohérente.

Une fois la recherche effectuée, avec une requête Ajax, les données sont retournées sous forme d'un tableau associatif JSON*, format plus léger que le XML*. Cela permet de pouvoir manipuler le contenu selon nos besoins.

Les données à afficher

Pour les informations à présenter à l'utilisateur, nous avons choisi uniquement les plus pertinentes :

- chaque catalogue* est représenté par son nom, sa description ainsi que les mots clés qui lui sont attachés ;
- chaque table est présentée avec son nom, sa description, son nombre de ligne ainsi que la présence ou non de positions célestes.

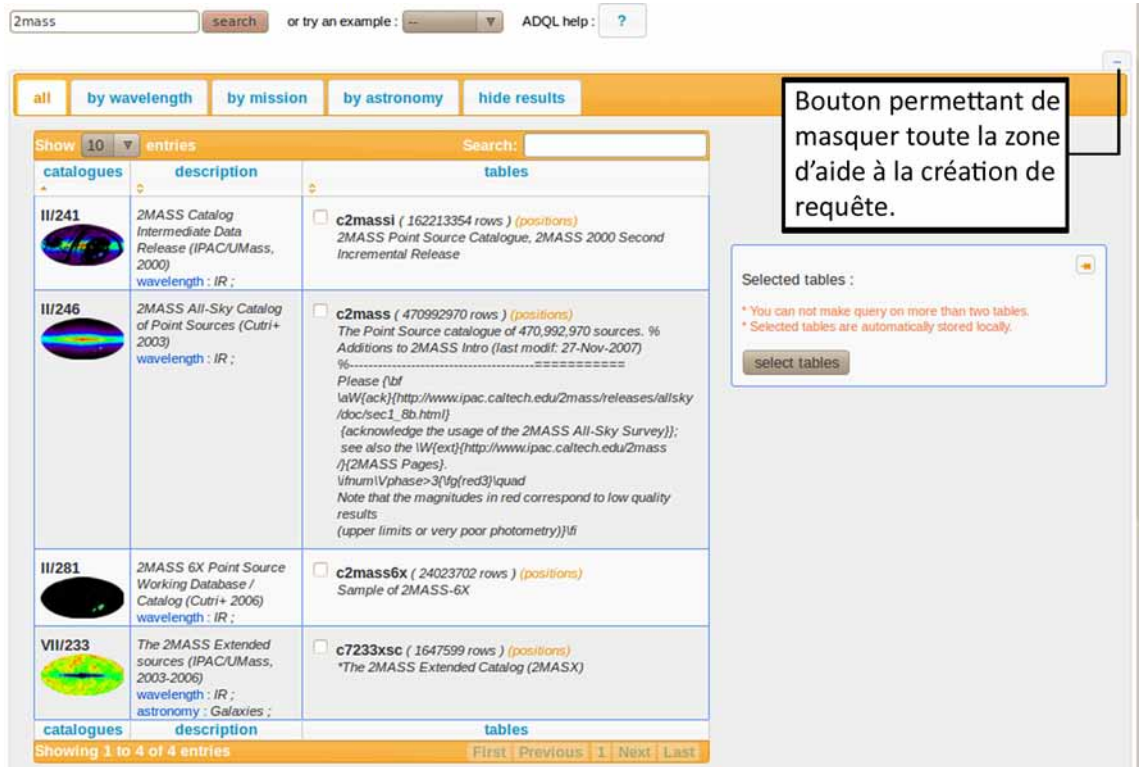
Il a été décidé qu'un seul footprint serait affiché par catalogue*, quand bien même chacune des tables qui le composent en possède.

L'affichage de ces données

Nous avons opté pour des onglets, quatre permettant d'afficher les résultats et un dernier pour les masquer.

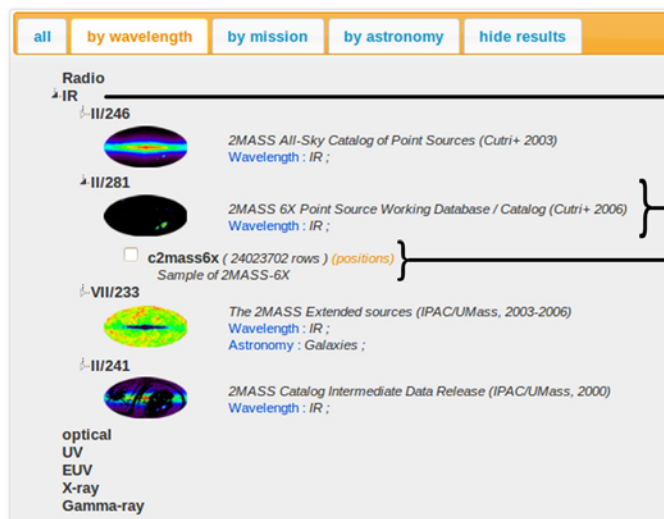
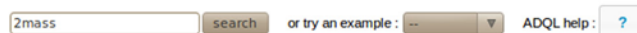
Dans le premier onglet « all » la présentation des résultats se fait sous la forme d'une table de données triable et paginable. Il est généré grâce au plugin dataTables et permet en plus d'effectuer une recherche par mot clé.

3. L'interface



Affichage des résultats pour le mot clé «2mass» dans l'onglet «all»

Les onglets « by wavelength », « by mission », « by astronomy » organisent les résultats de la façon à avoir une arborescence avec au premier niveau les mots clés, ensuite les catalogues* et leurs informations et au dernier niveau les tables et leurs données. Cet arbre est généré avec le plugin jqtree.



Affichage des résultats pour le mot clé «2mass» dans l'onglet «by wavelength»

Sélection des tables

Les tables sont sélectionnables à l'aide de check-box. Le choix de l'utilisateur s'affiche

3. L'interface

dans la box à droite de l'écran. Il récapitule le nom du catalogue* ainsi que le nom de la table et un lien sur le catalogue* conduit au service VizieR.

The screenshot shows the VizieR interface for selecting tables. On the left, a list of tables is shown with checkboxes. The selected table is **c7233xsc** (1647599 rows). On the right, a panel displays the selected tables and a 'select tables' button. Annotations point to various elements: 'pins permettant de bloquer la fenêtre' (pins allowing window locking), 'Selected tables' (the list of selected tables), 'table actuellement sélectionnée' (currently selected table), 'permet de supprimer la table' (allows deleting the table), and 'tables stockées localement' (locally stored tables).

sélection d'une table avec deux tables précédemment sélectionnées

La précédente sélection de l'utilisateur est stockée localement. Elle apparaît également ici et il est possible de les supprimer de la mémoire en cliquant sur l'icone de la poubelle. Le nombre de ligne de la table est également indiqué. Cela peut être utile, en cas de jointure notamment.

3.1.2. Aide à la création de la requête

Le langage ADQL* étant assez peu connu, il a été décidé que ce service devait permettre à un utilisateur novice de créer des requêtes. Il faut donc une interface qui permette de bien voir les différentes composantes d'une requête ainsi que leur fonction. Lors que les tables sont sélectionnées, le serveur est interrogé en Ajax et renvoie là encore un tableau associatif en JSON* qui contient toutes les informations sur les colonnes de chaque table.

Ceci étant à destination des néophytes, nous nous sommes concentré sur les composantes principales de la requête. De plus il n'est possible de choisir que deux tables maximum.

Columns and constraints

Cette partie présente à l'utilisateur les colonnes ainsi que leur description, leurs UCD*, l'unité etc. dans un tableau triable.

The screenshot shows two tables of columns and constraints for the tables 'c2mass' and 'c2massi'. The tables are organized into columns for table name, column name, data type, and other attributes.

tableaux des colonnes pour les tables « c2mass » et « c2massi »

3. L'interface

Chaque colonne peut être sélectionnée ou non pour être intégrée dans la requête. Il est également possible de lui appliquer une contrainte.

Un champ « computed col » permet au novice d'écrire une colonne calculée.

Le bouton « merge tables », visible uniquement si deux tables sont sélectionnées, permet de fusionner les tables. Cela permet de voir si des noms de colonnes sont identiques ou bien si les unités d'une valeur sont les mêmes.

Les tables ont chacune une couleur différente. Cela permet de bien les différencier lors de la fusion.

columns and constraints

separate tables

computed col : ?

c2massi, c2mass

Show All entries Search:

selected	table_name	column_name	description	ucd	unit	index	main	pos	constraint
<input checked="" type="checkbox"/>	c2mass	"_MASS"	(designation) Source designation (1)	meta.id;meta.main		1	0		
<input checked="" type="checkbox"/>	c2mass	Aflg	[0,1] (mp_flg) Association with asteroid or comet (11)	meta.code		0	0		
<input checked="" type="checkbox"/>	c2mass	Bflg	(bl_flg) JHK components fit to source (7)	meta.code		0	0		
<input checked="" type="checkbox"/>	c2massi	bl_flg	[012] JHK components fit to source (5)	meta.id;stat.fit		0	0		
<input type="checkbox"/>	c2massi	Bmag	? Blue mag of associated optical source	phot.mag;em.opt.B	mag	0	0		
<input type="checkbox"/>	c2mass	Bmag	?(b_m_opt) Blue magnitude of associated optical source	phot.mag;em.opt.B	mag	0	0		
<input checked="" type="checkbox"/>	c2massi	cc_flg	[UNPCDSRGO] Artifact contamin./confusion (6)	meta.note		0	0		
<input checked="" type="checkbox"/>	c2mass	Cflg	(cc_flg) Artifact contamination, confusion (8)	meta.code		0	0		
<input type="checkbox"/>	c2mass	Cntr	(pts_key) Unique source identifier in	meta.id;meta.table		0	0		

Fusion des deux tableaux de colonnes

Sky area

sky area

unit : ICRS FK4 (1950, 1950) GALACTIC SUPER GALACTIC

cone

center :

radius : arcsec

box

center :

width : arcsec

height :

menu pour définir la zone du ciel traité

Cette possibilité n'apparaît que s'il y a des positions principales* dans au moins une des tables choisies.

Elle permet de définir la zone du ciel concernée par la recherche. Cela peut alors être un « cone », un cône définit par un centre et par un radius, ou bien une « box », un quadrilatère définit par un centre, une hauteur

3. L'interface

et une largeur.

Il est aussi possible de définir les unités du système céleste dans lequel on souhaite faire la requête.

Les coordonnées du centre peuvent être entrées de différentes façons :

- en notation décimale (ex. : 10.2+12.5 ou 56-95.2) ;
- en notation sexagésimale : (ex. : 00 01 23.0 + 01 23 30.2 [hh mm ss +/- dd mm ss]);
- en donnant le nom de l'objet ;

Pour l'instant, seul le premier format est traité.

Il s'agit d'une clause WHERE. Par exemple:

- pour un « cone »: `WHERE 1=CONTAINS (POINT('ICRS', c41150097t1.RAJ2000, c41150097t1.DEJ2000), CIRCLE('ICRS', 0, 0, 12/3600.))`, vérifie si l'objet de la table c41150097t1 de coordonnées RAJ2000 et DEJ2000 est compris dans le cercle de centre 0,0 et de radius 12 arcsec. La fonction CONTAINS est un booléen qui renvoie 1 si le cas est validé, d'où le 1=... .
- pour un « box » : `WHERE 1=CONTAINS (POINT('ICRS', c41150097t1.RAJ2000, c41150097t1.DEJ2000), BOX('ICRS', 0, 10, 12/3600., 24/3600.))`, vérifie si le même objet que pour l'exemple précédent est présent dans un quadrilatère de centre 0,12, de hauteur 12 arcsec et de largeur 24 arcsec.

Joins

N'apparaissant que si deux tables sont sélectionnées, « joins » est obligatoire. Ce menu sert à définir une jointure des deux tables afin d'éviter un produit cartésien inexploitable et bien souvent inutile.

menu pour créer une jointure

Il est possible de faire deux types de jointures :

- « by coordinates », soit par coordonnées ou cross-match*, en indiquant un radius. Exemple : `WHERE 1=CONTAINS (POINT('ICRS', tableA.RAJ2000, tableA.DEJ2000), CIRCLE('ICRS', tableB.`

`RAJ2000, tableB.DEJ2000, 1))`, vérifie si l'objet de la table `tableA` est compris dans le cercle centré sur l'objet astronomique de `tableB` et de rayon 1 degré ;

- « by column names ». Les documentalistes essaient au maximum d'homogénéiser les noms de colonnes des tables. Il est donc possible de proposer à l'utilisateur de choisir une jointure en se basant sur le nom des colonnes. Cependant ceci est limité. En effet certaines jointures par nom de colonnes sont inexploitable soit par ce que le contenu est propre à chaque table soit parce que les unités du contenu sont différentes. Cette possibilité est avant tout pour montrer ce qu'y est faisable. Exemple : `WHERE c41150097t1.Name=c41150097t2.Name`, il s'agit d'une jointure standard en SQL.

Unit change of coordinates



menu pour obtenir les coordonnées dans un autre système

Cette partie permet à l'utilisateur d'obtenir les positions principales* des objets astronomiques dans un autre système de coordonnées*. Ceci n'est

pas une fonction d'ADQL* à proprement parler. Il s'agit en réalité du détournement de la fonction de création d'un point céleste. En effet à l'origine la fonction `POINT('système de coordonnées', ascension droite*, déclinaison*)` sert à définir un point avec l'ascension droite* et la déclinaison* exprimé dans le système de coordonnées* indiqué. Ici on l'utilise pour obtenir les coordonnées dudit point dans le système indiqué.

Elle appartient à la clause `SELECT`.

Exemple : `POINT('GALACTIC', c2mass.RAJ2000, c2mass.DEJ2000)` permettra d'obtenir les coordonnées de l'objet astronomique dans le système `GALACTIC`.

Max records

Cette dernière partie sert à limiter le nombre de résultats souhaités. Contrairement au SQL, ici la clause se place juste après le terme `SELECT` et est indiqué par le mot `TOP`. Exemple : `SELECT TOP 100 * from c2mass`, cette requête retournera les cent premiers résultats de la table `c2mass`.

Il est donc relativement aisé d'obtenir ce genre de requête :

```
SELECT TOP 100 c34140487t2.z, c34140487t2.Name,
c34140487t2.DEJ2000, c34140487t3.z, c34140487t3.RAJ2000,
c34140487t3.OMAG, c34140487t3.DEJ2000,
c34140487t2.RAJ2000,
POWER(OMAG, 3),
POINT('SUPER GALACTIC', c34140487t2.RAJ2000, c34140487t2.
```

```

DEJ2000), POINT('SUPER GALACTIC', c34140487t3.RAJ2000,
c34140487t3.DEJ2000)
FROM c34140487t2,c34140487t3
WHERE 1 = CONTAINS ( POINT ( 'ICRS' , c34140487t2.
RAJ2000,c34140487t2.DEJ2000), CIRCLE('ICRS',c34140487t3.
RAJ2000,c34140487t3.DEJ2000, 1.))
AND c34140487t3.z > 1

```

En plus de tout cela, j'ai rédigé une aide sur l'ADQL* reprenant les bases. Elle explique les différentes possibilités offertes par le SQL basique mais aussi par l'ADQL* que ce soit les colonnes calculées, les fonctions arithmétiques, le tri ou bien le filtrage des résultats.

3.1.3. La requête

```

1  -- output format : votable
2  SELECT TOP 100 c2massi.Id_opt, c2massi.Jmag, c2massi.RAJ2000, c2massi.e_Kmag, c2massi.e_Hmag, c2massi.e_Jmag, c2massi.bl_flg,
3  c2massi.DEJ2000, c2massi.mp_flg, c2massi.extd_flg, c2massi.cc_flg, c2massi.rd_flg, c2massi.Kmag, c2massi.Hmag, c2mass.e_Kmag,
4  c2mass.Aflg, c2mass.Xflg, c2mass.Cflg, c2mass.Bflg, c2mass.Rflg, c2mass.Oflg, c2mass.e_Hmag, c2mass.Jmag,
5  c2mass."_MASS", c2mass.e_Jmag, c2mass.RAJ2000, c2mass.DEJ2000, c2mass.Kmag, c2mass.Hmag
6  FROM c2massi,c2mass
7  WHERE 1=CONTAINS(POINT('ICRS',c2massi.RAJ2000,c2massi.DEJ2000),CIRCLE('ICRS',c2mass.RAJ2000,c2mass.DEJ2000, 123/3600.))

```

Edition de la requête

Ceci est la zone d'édition de la requête soit en cliquant sur le bouton « create query » soit en l'écrivant directement. La coloration syntaxique est immédiate. Elle s'e fait à l'aide du plugin CodeMirror avec une modification, développée par le CDS*, pour le rendre compatible avec l'ADQL*. Cela offre une bonne vue d'ensemble des différents éléments à un novice mais permet aussi à un utilisateur d'éviter quelques erreurs de frappes.

3.1.4. Exécution des requêtes et gestion des résultats

Nom et Format

Une fois que la requête est rédigée, plusieurs possibilités s'offrent à l'utilisateur.

Il peut tout d'abord nommer sa requête, ce qui sera utile quand de nombreux jobs auront été créés. De même il peut choisir le format de sortie des résultats :

- VOTable*, qui est un standard IVOA* en XML* ;
- CSV (Comma Separated Values), pour obtenir un fichier ouvrable dans un tableur ;
- JSON*;

QuickView

Le bouton « Quickview » permet d'obtenir un aperçu des résultats que retournera la requête en se limitant uniquement aux 10 premiers résultats. Une requête Ajax asynchrone

3. L'interface

renvoie le résultat de la requête ainsi que des informations importantes indiquant notamment les modifications de système de coordonnées ou l'utilisation de colonnes non indexées. Si la requête contient une erreur, elle n'est pas exécutée et le message d'erreur est renvoyé. Un timeout permet de tuer assez rapidement la requête Ajax. En effet puisqu'il s'agit d'un aperçu rapide, il n'est pas nécessaire d'attendre l'exécution complète des très longues requêtes.

Cela ouvre une fenêtre de dialogue créer avec JQuery. Les programmeurs peuvent également voir la requête SQL générée. Il est possible d'ouvrir plusieurs vues « QuickView » afin de pouvoir comparer les résultats par exemple.

id_opt	jmag	raj2000	e_kmage	hmage	jmag	bl_flg	dej2000	mp_flg	extd_flg	cc_flg	r
U	13.406	265.0166	0.048	0.034	0.045	111	-37.9226	0	0	000	:
U	13.406	265.0166	0.048	0.034	0.045	111	-37.9226	0	0	000	:
U	13.406	265.0166	0.048	0.034	0.045	111	-37.9226	0	0	000	:
U	13.406	265.0166	0.048	0.034	0.045	111	-37.9226	0	0	000	:
U	13.406	265.0166	0.048	0.034	0.045	111	-37.9226	0	0	000	:
U	13.406	265.0166	0.048	0.034	0.045	111	-37.9226	0	0	000	:
U	13.406	265.0166	0.048	0.034	0.045	111	-37.9226	0	0	000	:
U	13.406	265.0166	0.048	0.034	0.045	111	-37.9226	0	0	000	:
U	13.406	265.0166	0.048	0.034	0.045	111	-37.9226	0	0	000	:
U	13.406	265.0166	0.048	0.034	0.045	111	-37.9226	0	0	000	:
id_opt	jmag	raj2000	e_kmage	hmage	jmag	bl_flg	dej2000	mp_flg	extd_flg	cc_flg	r

show sql query

aperçu du résultat de la requête

Run

En click sur « run », l'utilisateur lance la requête dans le format de sortie voulu. Une première requête Ajax synchrone permet de vérifier la requête ADQL* envoyée. Si cette dernière contient des erreurs, le processus s'arrête là et renvoie les erreurs. S'il n'y a aucun problème, une seconde requête Ajax asynchrone envoie la requête à la Servlet* qui crée alors le job en lui attribuant un identifiant unique.

List of your jobs

Ce tableau triable permet de gérer tous les jobs créés par l'utilisateur. Il est possible de les détruire avec le bouton « destroy » ou bien d'avorter les jobs qui ne sont pas encore terminés grâce au bouton « abort ». « Refresh » permet de mettre à jour cette liste. Toutes ces actions s'effectuent à l'aide de l'API TAP* développée au sein du CDS*. Pour la mise à jour de la liste, l'API transforme en JSON* le document XML* renvoyé par le serveur.

3. L'interface

List of your TAP queries :

Show entries Search:

ID	name	phase	start	destruction	results
133898889042A	c3414048713	COMPLETED	15:21:39 on 06/06/2012	15:21:39 on 11/06/2012	result
1339161965071A	c2massi	COMPLETED	15:26:05 on 08/06/2012	15:26:05 on 13/06/2012	result
1339161998682A	query 2mass	COMPLETED	15:26:38 on 08/06/2012	15:26:38 on 13/06/2012	result
1339162152411A	tap Query	EXECUTING	15:29:12 on 08/06/2012	15:29:12 on 13/06/2012	please refresh

Showing 1 to 4 of 4 entries [First](#) [Previous](#) [Next](#) [Last](#)

tableau récapitulatif de la liste des jobs de l'utilisateur

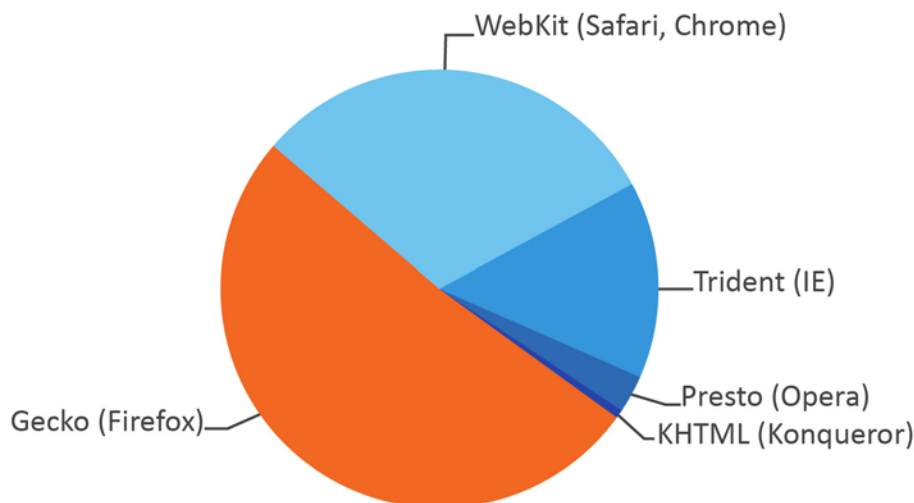
Chaque job est éphémère, c'est-à-dire qu'au bout de 5 jours il sera automatiquement détruit.

La liste de jobs est stockée côté serveur, mais pour des raisons de standard il est impossible de sauvegarder les dates de créations et de destruction. La liste est donc enregistrée en localStorage et mise à jour lors du refresh.

3.4. Les problèmes et leur résolution

3.3.1. International et miroir

VizieR étant mondialement connu, il a été obligatoire de créer une interface compatible avec les grands navigateurs et avec un maximum de résolutions d'écrans.



Graphique représentant les différents moteurs de rendu* utilisés pour accéder au service VizieR

Grâce à JQuery, le site s'adapte automatiquement à la taille du navigateur et globalement il n'y a pas de problème de style. Nous avons rencontré nos plus gros soucis sur Internet Explorer. En effet, le navigateur de Microsoft parcourt les documents XML* à sa manière et il a fallu si adapter. Ensuite pour la mise à jour de la liste de jobs, la librairie TAP* interrogeait, à l'origine, le serveur sans envoyer de paramètres donc en utilisant toujours la même URL. Or s'il n'y pas de changement d'URL, Internet Explorer 8 va interroger le cache pour obtenir les informations et non recharger la page. Pour contourner ce désagrément,

3. L'interface

nous avons dû envoyer une variable unique, le timestamp, à chaque rafraichissement de la liste de jobs.

Un deuxième point important à prendre en compte est le mirroring. Il existe de nombreux miroirs à travers le monde : Tokyo (Japon), Pune (Inde), Moscow (Russie), Hawaii (Etats-Unis)... En raison de cela il n'est pas possible de mettre des URL vers des images ou des services hébergés ailleurs, celle-ci risquant de changer. J'ai utilisé des tags « GLU » pour y remédier. Ces derniers doivent être intégrés dans la page qui sera chargée par le serveur, ce dernier les interprète et les transforme en en tag HTML ou bien en texte. Néanmoins, la majorité des actions de l'utilisateur sur l'interface sont exécutées côté client et l'intégration de tags GLU après le chargement de la page est inefficace. Les informations dont on a besoin pour créer les liens vers le service VizieR ou bien la petite icône qui les accompagnent dépendent de ces tags GLU. Elles sont donc écrites dans des balises div masquées. De même, le menu et le bas de la page sont des tag GLU.

3.3.2. La durée des requêtes

Certaines requêtes peuvent être très longues principalement lors de la recherche, parce que le terme est trop vague, ou bien lors du QuickView, parce que la requête est trop complexe. Afin de ne pas bloquer le service, dans ces cas de figure, nous avons choisi d'utiliser des requêtes Ajax asynchrones et de définir un timeout à l'échéance duquel la requête est arrêtée. L'asynchronicité des requêtes Ajax permet de ne pas bloquer l'utilisation du site.

3.3.3. Volumes des résultats

Dans un premier temps, la recherche retournait un tableau associatif en JSON* contenant les catalogues*, les tables correspondantes mais également leurs colonnes et toutes les informations nécessaires. Outre le fait que le tableau était très grand, le temps d'exécution de toutes les requêtes pour l'obtention de toutes les données aboutissait en timeout pour les termes trop vagues. J'ai donc créé une Servlet* qui retourne les informations sur colonnes de chaque table passée en paramètres. Cela a beaucoup diminué le temps d'exécution de la recherche. Néanmoins le nombre de résultats obtenu à soulever un deuxième problème très important.

Avant la création de la Servlet* pour les colonnes, le nombre de résultats ne dépassait pas le millier. Maintenant, il est possible d'obtenir un très grand nombre de catalogues*, c'est le cas par exemple avec le terme « galaxies » qui en retourne 2 604 ou encore « stars » qui en retourne 4 357. L'affichage d'autant de données n'est pas aisé. C'est dû au fait que, pour chaque catalogue*, il y a une description, au moins une table et sa description et éventuellement une image. L'insertion dans le DOM* de la page d'autant de contenu génère un message d'erreur indiquant que l'exécution du script JQuery est trop lente. L'onglet pour lequel cela pose le plus de problème est « by astronomy ». En effet chaque catalogue* possède au minimum un mot clé dans cette catégorie et il apparaîtra dans l'arbre à tous les termes le caractérisant en augmentant considérablement le contenu à insérer dans le DOM*.

3. L'interface

Le problème étant lié aux performances de l'ordinateur utilisé, il constituera une nouvelle piste d'amélioration futur pour VizieR

3.3.4. Variété des tables

Le grand nombre de tables et leurs différences m'ont compliqué la tâche. Les données contenues n'étant pas toujours les mêmes il a fallu chercher les cas particuliers, les exceptions et tous les cas qui posaient problèmes. Ce qui n'a pas été simple puisqu'on effectue généralement la même recherche à chaque fois.

Cela a surtout été nécessaire pour les positions principales* et pour l'affichage. Dans le premier cas, les colonnes utilisées pour indiquer les positions principales* sont généralement décrite par un UCD* l'indiquant, mais ce n'est pas toujours vrai. Il y a les tables ne possédant pas d'UCD* car elles sont très anciennes ou bien celles avec des colonnes de positions calculées qui ont alors 4 colonnes indiquées par cet UCD*. Par exemple, la table c41390575t2 du catalogue J/A+AS/139/575 ne possède aucun UCD.

Pour la création de la Servlet* renvoyant les données sur les colonnes, il a fallu prendre tous ces cas en considération pour retourner un résultat juste. Certaines descriptions de catalogues*, de tables ou même de colonnes possèdent des caractères spéciaux tels que des antislashes ou bien des balises ouvrantes et fermantes. Je les ai donc soit modifiés soit échappés pour pouvoir les intégrer dans la page HTML ou pour pouvoir les insérer dans un tableau associatif en JSON*.

Conclusion

Tout au long de ces 18 semaines, j'ai créé une interface web permettant à un néophyte en ADQL de créer simplement et facilement une requête pour interroger la base de données du service TAP VizieR. Ne connaissant pas ce langage moi même au début de mon stage, il m'a été aisé de comprendre et d'analyser la demande et les besoins des utilisateurs.

L'agencement des différents éléments a été facilement trouvé. Le design est dans l'ensemble définitif, mise à part des détails afin de répondre au mieux aux demandes des chercheurs. La partie de l'interface qui demande encore des améliorations est liée aux performances pour éviter les écueils dus à l'exécution trop lente d'un script, comme par exemple lors de l'affichage d'une arborescence pour les données.

Ce stage s'est bien déroulé. Il m'a offert la possibilité de vraiment lier mon DUT Services et Réseaux de Communication, grâce à l'apparence de l'interface, et ma licence, avec la programmation en JavaScript et jQuery.

Une interface d'administration est prévue. En effet, ce service permettant à un utilisateur de lancer des requêtes nombreuses et parfois lente, mon maitre de stage m'a proposé un CDD afin de réaliser la partie administrable. Elle devra permettre de gérer les requêtes, notamment celles étant encore en cours d'exécution, et fournir des informations sur l'état du serveur.

J'ai appréciée de travailler au sein du CDS*. On ne se sent pas abandonnée, l'équipe n'hésite pas à partager ses connaissances. J'ai pu, de ce fait, assimiler relativement facilement des notions d'astronomies et comprendre la nécessité de l'IVOA* et des outils informatiques pour la recherche scientifique.

ADQL : *Astronomical Data Query Language*, ce langage de requête de données astronomiques est un standard de l'IVAO* basé sur un SQL enrichi de fonctions géométriques célestes.

Ascension droite : right ascension (RA), équivaut à la longitude

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

- une partie «méta-données» qui indique l'auteur du catalogue, la période où les mesures ont été effectuées, la description de la structure des données, etc ... ;
- une partie «données» qui liste les différents objets avec leurs différentes mesures physiques.

Il est généralement composé de plusieurs tables. Les très grands catalogues font plus de 10 millions de lignes.

CDS : Le *Centre de Données astronomiques de Strasbourg* est un centre de données dédié à la collecte et la distribution mondiale des données astronomiques et des informations connexes. (<http://cds.u-strasbg.fr/>)

CNRS : Le *Centre National de la Recherche Scientifique* est un organisme public de recherche (Etablissement public à caractère scientifique et technologique, placé sous la tutelle du Ministère de l'Enseignement supérieur et de la Recherche). Il produit du savoir et met ce savoir au service de la société. (<http://www.cnrs.fr/>)

Cross-match : jointure par coordonnées de 2 tables. Cf annexe B.

Déclinaison : declination (DE), équivaut à la latitude.

DOM : *Document Object Model*, permet aux programmes et aux scripts d'accéder dynamiquement et mettre à jour le contenu, la structure et le style des documents WWW comme les page HTML ou XML.

ESA : *European Space Agency*, agence spatiale européenne. (<http://www.esa.int/esaCP/France.html>)

ESO : *European Southern Observatory*, l'Observatoire Européen Austral est la première organisation intergouvernementale pour l'astronomie en Europe et l'observatoire astronomique le plus productif au monde. Il emploie environ 730 personnes. (<http://www.eso.org/public/france.html>)

IVOA : *International Virtual Observatory Alliance*, alliance internationale de l'observatoire virtuel. (<http://www.ivoa.net/>)

ICRS : *International Celestial Reference System* ou système de référence céleste international. cf annexe A

Jointure : Une jointure permet de combiner les colonnes de plusieurs tables.

JSON : *JavaScript Object Notation*, c'est un format de données textuel, générique.

LaTeX : c'est un système très puissant pour composer du texte, utilisé pour les documents scientifiques notamment.

Magnitude : C'est une mesure de l'éclat d'un objet. Plus la valeur est grande, moins l'objet est lumineux. Le Soleil a une magnitude de -20 environ, la Lune de -12. On considère que l'œil humain ne peut percevoir un objet astronomique d'une magnitude supérieur à 6.

MCD : un *Modèle Conceptuel des Données* permet de représenter la structure du système d'information, du point de

vue des données, et définit également les dépendances ou relations entre ces différentes données.

Positions principales : Coordonnées que l'on va utiliser en priorité dans VizieR. Quand une table comporte plusieurs systèmes de coordonnées, on utilisera d'abord le ICRS*, puis cas échéant le FK4 et enfin le GALACTIC.

PostgreSQL : Outil libre de système de gestion de base de données relationnelle. (<http://www.postgresql.fr/>)

RENATER : *REseau NATional de télécommunications pour la Technologie, l'Enseignement et la Recherche*. Il s'agit du réseau informatique français reliant les différentes universités et les différents centres de recherche. Il a une trentaine d'employés. (<http://www.renater.fr/>)

Servlet : c'est une application Java fonctionnant du côté. Elle permet donc de gérer des requêtes HTTP et de fournir au client une réponse HTTP dynamique (donc de créer des pages web dynamiques).

Système de coordonnées célestes : En astronomie, un système de coordonnées céleste est un système de coordonnées, permettant de déterminer une position dans le ciel. Cf annexe A.

TAP : *Tables Access Protocol*, protocole standardisé par l'IVOA pour l'accès au contenu des tables de données.

UCD : *Unified Content Descriptors*, il catégorise chaque colonne en fonction de ses paramètres.

UMR : Une *Unité Mixte de Recherches* regroupe un ou plusieurs organismes de recherches et un ou plusieurs établissement(s) d'enseignement supérieur.

VO : *Virtual Observatory*, un observatoire virtuel utilise Internet pour bâtir un environnement de recherche scientifique dans lequel les programmes de recherche en astronomie pourront être conduits.

VOTable : c'est un standard XML pour l'échange de données représentées comme un ensemble de tables.

XML : *Extensible Markup Language* (langage de balisage extensible) est un langage informatique de balisage générique.

XMM : XMM-Newton est un satellite artificiel d'observation des rayons X. L'abréviation XMM signifie X-ray Multi-Mirror. Et Newton est en hommage à l'astronome Isaac Newton.

Annexe A : Systèmes de coordonnées célestes



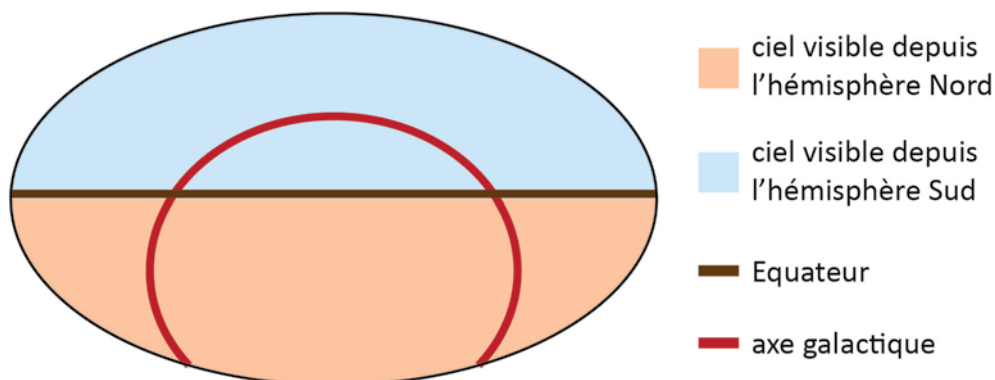
footprint visible sur la page d'accueil de VizieR

Il existe plusieurs systèmes utilisant une grille de coordonnées projetée sur la sphère céleste, de manière analogue aux systèmes de coordonnées géographiques utilisés à la surface de la Terre. Les systèmes de coordonnées célestes* diffèrent seulement dans le choix du plan de référence, qui divise le ciel en deux hémisphères le long d'un grand cercle (le plan de référence du système de coordonnées géographiques est l'équateur terrestre). Dans le service TAP VizieR, il n'y en a que deux de proposés.

a. le système de coordonnées équatoriale, liés à la Terre :

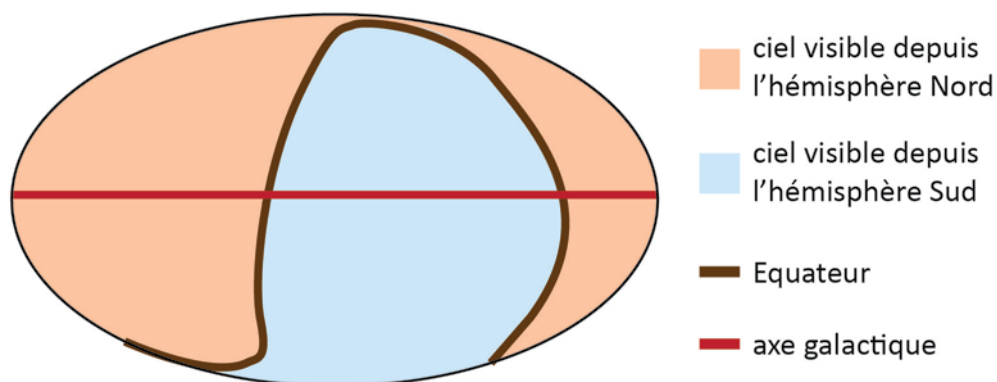
L'ICRS*, pour International Celestial Reference System (système de référence céleste international), est la normalisation du système de coordonnées* équatoriale. Il est défini par l'orientation des pôles célestes, qui sont les points de la sphère céleste vers lesquels pointe l'axe de rotation de la Terre, et d'un troisième point.

FK (Fundamental Katalog) : Il s'agit d'un catalogue* de référence d'objet de faible magnitude* pour mesurer plus facilement la position des autres objets astronomiques. Aujourd'hui, il en existe 6 versions dont FK5 qui coïncide avec l'ICRS*, et FK4, proposée dans l'interface, qui correspond à la position de ces étoiles en 1950.



Schematisation de l'univers observable en système de coordonnées équatoriale

b. le système de coordonnées galactique, reposant sur notre galaxie.

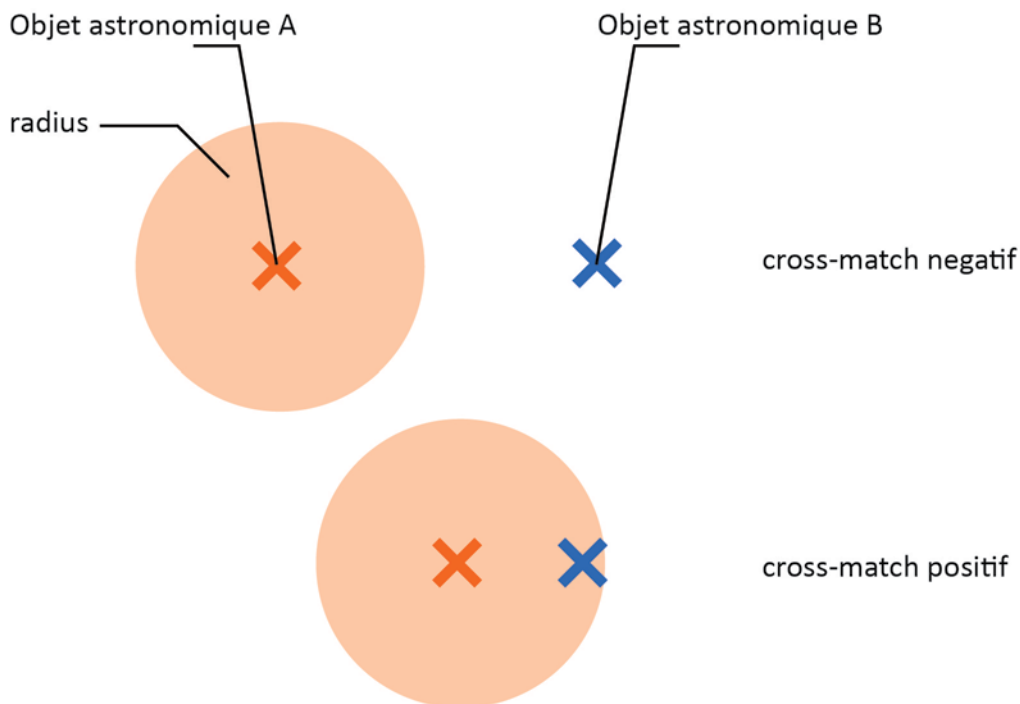


Schematisation de l'univers observable en système de coordonnées galactique

Ce système comporte le type Galactic et Super Galactic. Il s'agit du système utilisé pour réaliser les footprints avec Alaska.

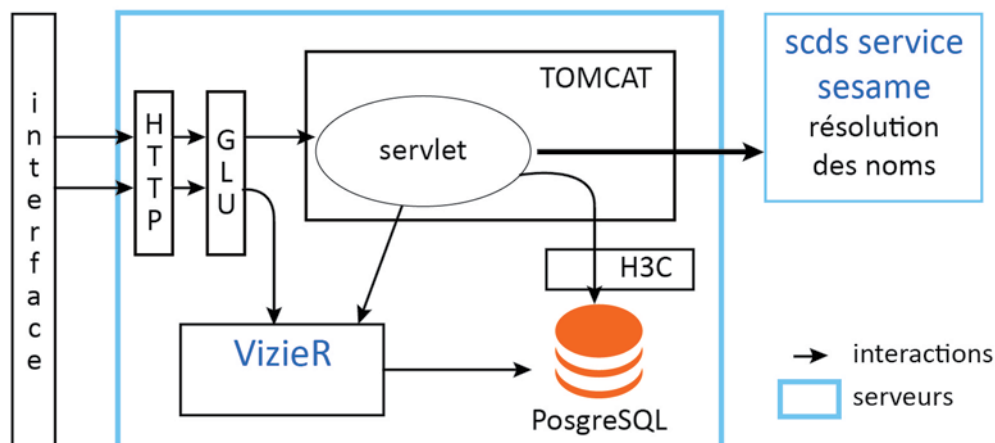
Annexe B : Cross-match

Il s'agit d'une jointure par coordonnées. Pour chaque source du catalogue* A, on vérifie si les sources du catalogue* B se trouvent dans un certain rayon autour de la première source.



*Schema du principe d'un cross-match**

Annexe C : serveur TAP



Architecture pour le service TAP VizieR.

HTTP : logiciel de serveur HTTP. Pour ce serveur il s'agit d'Apache.

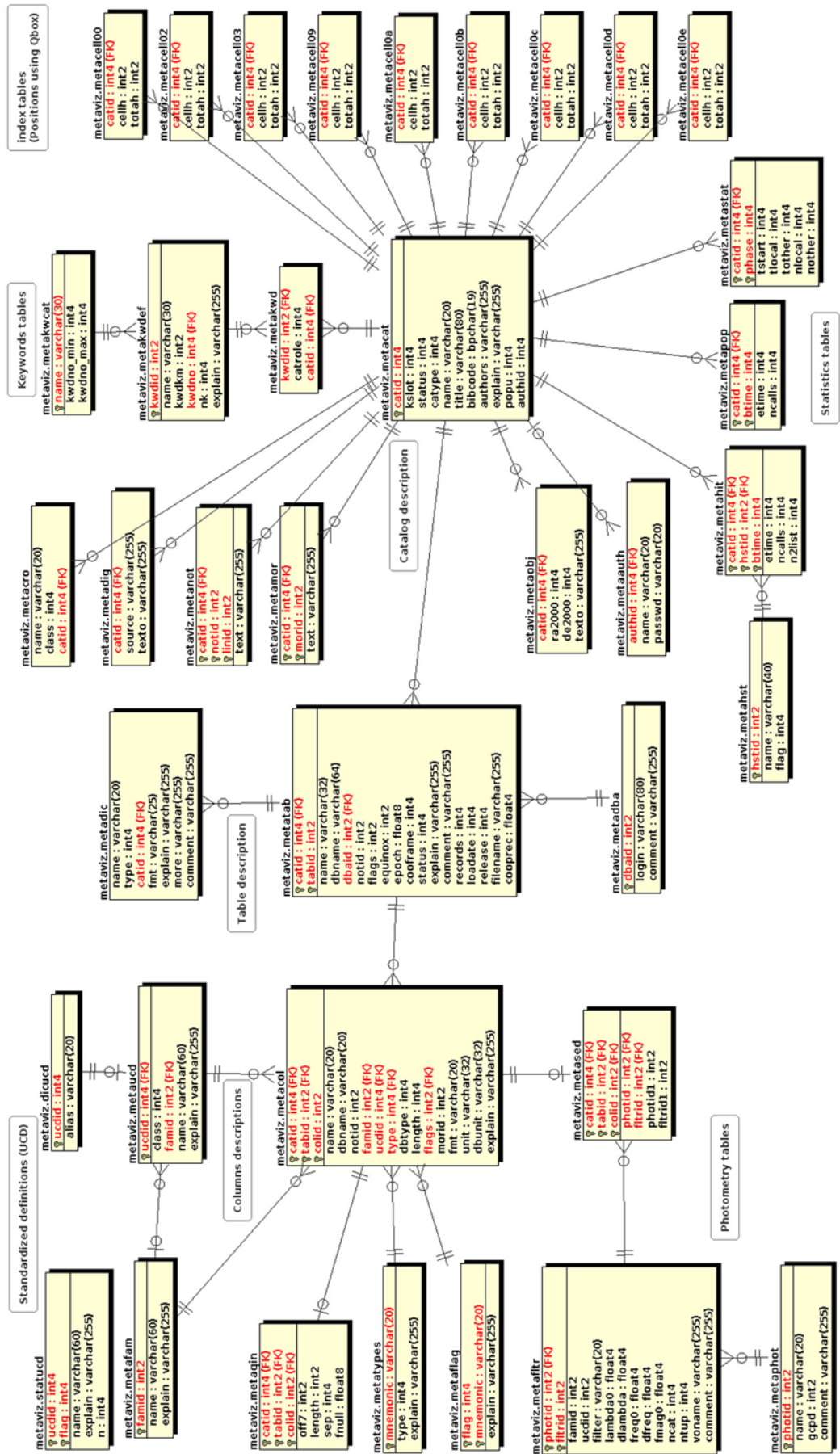
GLU : Programme permettant de faire fonctionner conjointement d'autres programmes.

TOMCAT : Tomcat est un conteneur web. Il permet, entre autre, l'implémentation des servlets.

H3E : Healpix Tree C , programme l'utilisation des des index sur la base de données ainsi que des fonctions de géométrie céleste.

Annexe D : MCD du meta-catalogue de la base de données de TAP VizieR

Annexes



Annexe E : ADQL vers SQL

requête ADQL :

```
-- output format : votable
SELECT TOP 100 c1239hip_main.pmDE, c1239hip_main.HIP,
c1239hip_main.RA_ICRS_, c1239hip_main.RAhms, c1239hip_main.
pmRA, c1239hip_main.e_Plx, c1239hip_main.DE_ICRS_, c1239hip_
main.DEdms, c1239hip_main.Notes, c1239hip_main.Vmag,
c1239hip_main.Plx, c1239tyc_main.B_V, c1239tyc_main.pmDE,
c1239tyc_main.TYC, c1239tyc_main.RA_ICRS_, c1239tyc_main.
RAhms, c1239tyc_main.pmRA, c1239tyc_main.BTmag, c1239tyc_
main.DE_ICRS_, c1239tyc_main.DEdms, c1239tyc_main.VTmag,
c1239tyc_main.Vmag
FROM c1239hip_main, c1239tyc_main
WHERE 1=CONTAINS(POINT('ICRS', c1239hip_main.RA_
ICRS_, c1239hip_main.DE_ICRS_), CIRCLE('ICRS', c1239tyc_main.
RA_ICRS_, c1239tyc_main.DE_ICRS_, 30/3600.))
```

même requête en SQL :

```
SELECT round((c1239hip_main.pmDE/100.0)::numeric,2) AS
pmDE , c1239hip_main.HIP AS HIP , round((c1239hip_main.
RA_ICRS_)::numeric,8) AS RA_ICRS_ , c1239hip_main.RAhms AS
RAhms , round((c1239hip_main.pmRA/100.0)::numeric,2) AS
pmRA , round((c1239hip_main.e_Plx/100.0)::numeric,2) AS
e_Plx , round((c1239hip_main.DE_ICRS_)::numeric,8) AS DE_
ICRS_ , c1239hip_main.DEdms AS DEdms , c1239hip_main.Notes
AS Notes , round((c1239hip_main.Vmag/1000.0)::numeric,2)
AS Vmag , round((c1239hip_main.Plx/100.0)::numeric,2) AS
Plx , round((c1239tyc_main.B_V/1000.0)::numeric,3) AS
B_V , round((c1239tyc_main.pmDE/100.0)::numeric,2) AS
pmDE , c1239tyc_main.TYC AS TYC , round((c1239tyc_main.
RA_ICRS_)::numeric,8) AS RA_ICRS_ , c1239tyc_main.RAhms AS
RAhms , round((c1239tyc_main.pmRA/100.0)::numeric,2) AS
pmRA , round((c1239tyc_main.BTmag/1000.0)::numeric,3) AS
BTmag , round((c1239tyc_main.DE_ICRS_)::numeric,8) AS DE_
ICRS_ , c1239tyc_main.DEdms AS DEdms , round((c1239tyc_main.
VTmag/1000.0)::numeric,3) AS VTmag , round((c1239tyc_main.
Vmag/1000.0)::numeric,2) AS Vmag FROM c1239hip_main AS
c1239hip_main CROSS JOIN c1239tyc_main AS c1239tyc_main
WHERE '1' = h3c_join(c1239hip_main.ra_icrs_ , c1239hip_
main.de_icrs_ , c1239tyc_main.ra_icrs_ , c1239tyc_main.de_
icrs_ , 1.0*30/3600.) Limit 10
```