

Observatoire astronomique
de Strasbourg

Rapport de stage

Réalisation d'interface graphique pour création fichier d'ingestion de base de données

ROYER Félix

28 mars 2016 – 4 juin 2016



Maître de stage :
Michel SALOMON

Tuteur de stage :
Gilles LANDAIS

Remerciements

Je tiens particulièrement à remercier Gilles LANDAIS, mon maître de stage, pour le temps qu'il m'a accordé en me conseillant et en m'aidant. Il a répondu à mes questions et m'a apporté quotidiennement son avis sur le travail effectué et des indications pour la suite du développement. Son aide m'a été précieuse tout au long du stage que ce soit pour le développement ou pour la rédaction de ce rapport.

Je tiens également à remercier Monsieur André SCHAAFF ainsi que l'ensemble de l'équipe du CDS pour avoir pris le temps de nous présenter, à mes camarades stagiaires et à moi-même, le travail qu'ils réalisaient et sur quels points de ce travail nous allions intervenir. Cela nous a permis de mieux comprendre en quoi nos stages allaient aider le CDS.

Je tiens enfin à remercier Monsieur Michel SALOMON mon référent de stage à l'IUT pour avoir effectué le suivi de mon stage, pour avoir pris le temps de visiter le CDS et pour avoir répondu à mes questions relatives à la rédaction des rapports ou à des problèmes administratifs.

Sommaire

1 – Introduction	p 6
2 – Présentation de l'entreprise	p 8
2.1 – Les équipes de recherche	p 9
2.2 – Les services du CDS	p 10
2.3 - L'IVOA	p 12
3 – Présentation du sujet	p 13
4 – Cahier des charges	p 15
5 – Mise en œuvre	p 17
5.1 – Étude des bibliothèques JGoodies et JavaFX	p 18
5.1.1 – Contexte de la demande	p 18
5.1.2 – Comparaison des bibliothèques	p 18
5.1.3 – Choix de la bibliothèques	p 20
5.2 – Utilisation de l'API Saada	p 20
5.2.1 – Récupération des données à partir des fichiers FITS	p 20
5.2.2 – Récupération des en-têtes FITS	p 21
5.2.3 – Vérification de la validité des mappings	p 21
5.3 – Accès aux bases de données du CDS	p 22
5.4 – Gestion des mappings	p 23
5.4.1 – Ouverture des mappings	p 23
5.4.2 – Édition des mappings	p 24
5.4.3 – Prévisualisation	p 26
5.4.4 – Sauvegarde	p 26
5.5 – Difficultés rencontrées	p 27
5.5.1 – Problèmes avec l'API Saada	p 27
5.5.2 – Tutoriaux JavaFX	p 28
6 – Bilan	p 28
7 – Conclusion	p 30

Liste des abréviations

- API : Application Programming Interface (Interface de programmation applicative);
- CDS : Centre de Données astronomiques de Strasbourg;
- FITS : Flexible Image Transport System (Système flexible de transport d'image);
- IVOA : International Virtual Observatory Alliance (Alliance international de l'observatoire virtuel);
- WCS : World Coordinate System (Système Mondial de Coordonnées).

1 – Introduction

En vue de la validation de mon DUT, j'ai effectué un stage de 10 semaines à l'Observatoire Astronomique de Strasbourg, et plus particulièrement au Centre de Donnée astronomique de Strasbourg (CDS). Ce stage est un premier contact avec le monde du travail ainsi qu'une première expérience professionnelle. C'est la concrétisation pratique des connaissances acquises au cours des 2 années d'étude à l'IUT. Durant ce stage j'ai été encadré par Monsieur Gilles LANDAIS, ingénieur d'étude au sein du CDS.

L'intitulé du stage était : « Conception d'une interface utilisateur d'ingestion dans le système d'information Vizier des données images et spectres en astronomie » et proposait de réaliser une interface graphique en Java à destination des documentalistes du CDS. Cette interface a pour but de les aider dans leur travail de génération de fichiers d'ingestion qui, pour le moment, était fait par ligne de commande. Ce projet est le premier du genre pour le service Vizier qui entend renouveler ces outils de travail pour des interfaces moins austères. L'interface se doit donc d'être complète et simple pour garantir un travail efficace et rapide.

Ce stage requiert donc des connaissances en interface homme-machine ainsi qu'en Java et ses bibliothèques graphiques. Ce travail s'est avéré plus intéressant que je ne le croyais en postulant, car il m'a également offert des opportunités de recherche et de développement sur des domaines informatiques que je ne pensais pas aborder.

Étant moi même féru d'astronomie et amateur des interfaces graphiques Java, avec lesquels je suis à l'aise pour travailler, j'ai profité de l'occasion dès que j'ai eu connaissance de cette offre de stage.

Ce rapport a pour but de résumer et de retranscrire au mieux la manière avec laquelle s'est déroulé le stage, selon la structure suivante :

Dans un premier temps, je présenterais l'Observatoire Astronomique de

Strasbourg, ses équipes et ses services puis le sujet précis du stage. Cela a pour but de donner une idée du contexte dans lequel le stage s'est déroulé.

Je décrirai ensuite les principales fonctionnalités de l'application finale en expliquant leur fonctionnement,

Je dresserai finalement un bilan des 10 semaines de stages sur le travail réalisé ainsi que sur les améliorations possibles à apporter.

2 – Présentation de l'entreprise

L'observatoire astronomique de Strasbourg est un institut de recherche dont les origines remontent au 17^{ème} siècle. C'est en 1881 que l'infrastructure actuelle est construite au cœur de l'université de Strasbourg. Il est désormais intégré au Jardin des Sciences de l'Université de



Figure 1 : bâtiment de la coupole

Strasbourg. Aujourd'hui, l'observatoire a le statut d'unité mixte de recherche du CNRS et de l'université de Strasbourg. Sa direction est assurée par Hervé Wozniak.

Ses principaux objectifs sont :

- l'acquisition de données d'observation ;
- le développement et l'exploitation de moyens appropriés à son fonctionnement ;
- l'élaboration des outils théoriques nécessaires à son fonctionnement.

Il est également chargé :

- de fournir des services liés à son activité de recherche ;
- d'assurer la formation des étudiants et des personnels de recherche ;
- d'assurer la diffusion des connaissances ;
- d'assurer des activités de coopération internationale.

Son activité consiste donc principalement à rendre accessibles des données et relevés astronomiques à l'échelle mondiale. A cette fin, il est structuré en 3 équipes de recherches : Galaxies, Hautes-énergies et le Centre de Données astronomiques de Strasbourg (CDS).

2.1 – Les équipes de recherche

- Galaxies :

Le rôle de l'équipe « Galaxies » est d'étudier la formation des galaxies ainsi que le recensements de leurs populations stellaires. Son but est d'étudier la dynamique qui est à l'origine de la formation des galaxies et le comportement des étoiles et des matériaux qui s'y trouvent. L'objectif global est de réunir une masse d'information qui permettrait de dresser un historique des événements clés de la vies des galaxies.

L'équipe participe également à des missions stellaires telles que Gaia, mission lancée en décembre 2013 par l'Agence Spatiale Européenne qui a pour but de fournir des relevés sur la Voie Lactée.

- Hautes-énergies :

L'équipe « Hautes-énergies » s'intéresse aux sources d'émission de rayon X : les objets compacts - étoiles à neutrons par exemple - ou encore les noyaux actifs des galaxies.

Elle fait partie du Survey Science Center d'XMM (SSC-XMM), un consortium de laboratoires internationaux dont le rôle est de fournir des catalogues complets d'objets observés par le satellite européen XMM-Newton.

L'équipe participe également à des projets communautaires tels que le projet européen Arches, notamment en collaboration avec le CDS.

- Le Centre de Données de Strasbourg :

Cette équipe étant celle qui m'a accueilli pour effectuer mon stage, je vais développer un peu plus avant à son propos.

Lors de la création du service, en 1972, le sigle « CDS » correspondait à « Centre de Données Stellaires » avant de devenir « Centre de Données astronomiques de Strasbourg » en 1983.

La principal travail du CDS est de collecter, répertorier et rendre accessibles des données provenant d'observatoires ou publiées dans des revues scientifiques¹. Reconnu internationalement, le CDS propose des services pour les astronomes qui sont sollicités par environ 1 000 000 de requêtes par jour.



Figure 2 : logo du CDS

Le CDS compte une trentaine de personnes réparties entre les astronomes, les documentalistes et les informaticiens, auxquels s'ajoutent régulièrement des stagiaires et autres personnes présentes temporairement dans le service. Ces employés travaillent à différents niveaux sur les différents services du CDS.

Le CDS a reçu le DSA (Data Seal of Approval) pour les années 2014 et 2016. Ce label est dédié aux centres de données scientifiques ; il garantit que les moyens mis en œuvre pour la préservation et la dissémination des données sont en accord avec les exigences du label.

2.2 – Les services du CDS

- Simbad :

Simbad est une base de données objets, qui regroupe près de 8 000 000 d'objets astronomiques, répondants à 23 000 000 d'identifiants, regroupés en 316 000 bibliographies, les données de Simbad provenant uniquement de publications. Ces différents objets sont répertoriés selon les flux d'observation infrarouges, visibles, ultraviolets, etc ,, en 250 colonnes pour un total de 60 GO de données.

Simbad est une base de référence qui permet l'identification et la bibliographie des objets astronomiques.



Figure 3 : logo de Simbad

¹ <http://cdsweb.u-strasbg.fr/index-fr.gml>

Ce service est sollicité par 500 000 requêtes par jours, provenant d'astronomes en majeure partie situés aux États-Unis.

- VizieR :

VizieR est la seconde base de données proposée par le CDS. Elle regroupe des catalogues de données astronomiques sous forme tabulaires, des données provenant d'articles issus des journaux de référence en Astronomie, de télescopes ou de missions d'observations. Sur un total de 15 000 catalogues, 30 000 tables au contenu hétérogène et aux tailles de table très variables, entre 10 et 2 000 000 000 d'entrées.



Figure 4 : logo de VizieR

VizieR propose un accès aux données, via des services web conformes aux standards définis par l'IVOA (International Virtual Observatory Alliance). L'IVOA, ou l'Observatoire virtuel est un consortium international dont le but est l'interopérabilité des ressources en astronomie : il y est établis des standards de formats (XML), d'échange (protocoles) ou des langages d'accès aux données.

VizieR dispose de miroirs (Royaume-Unis, Canada, États-Unis (Harvard+Hawaii), Inde, Japon, Chine, Afrique du sud) dans lesquels sont répartis/dupliqués les données : données originales accessibles par FTP, bases de données, formats dédiés pour les grandes volumétries.

VizieR est sollicité par environ 300 000 requêtes par jour avec des pics occasionnels à quelques millions.

Au cours de l'année 2015, le nom VizieR a été cité 337 fois dans 14 articles scientifiques à travers le monde.

Les données tabulaires de VizieR sont de plus en plus fréquemment accompagnées d'images ou de spectres. Ces données, tout comme les tables sont indexées et documentées par des méta-données (région du ciel ciblé, longueur d'onde observée, télescope utilisée etc). Ces méta-données sont incluses dans des fichiers de mappings qui sont joints aux documents dans VizieR.

L'ensemble de l'ingestion des données, incluant l'ajout des méta-données est opéré par les documentalistes du CDS et supervisé par les astronomes.

- Aladin :

Aladin est un atlas interactif du ciel, regroupant les images provenant des plusieurs sources d'observations et permettant aux astronomes de localiser, comparer et d'analyser les données issues de la plupart des serveurs astronomiques. La carte en trois dimensions est divisée en losanges, selon le modèle HEALPix - ce qui limite les déformations de l'image notamment aux pôles - eux même subdivisés en losanges, qui se chargent à mesure que l'utilisateur zoom sur l'image. Ce processus permet une grande précision et une limitation de la consommation de mémoire.

Depuis l'interface d'Aladin, il est également possible de charger les tables de VizieR ou de Simbad concernant l'objet ciblé.

Chaque jour, 300 000 de ces tuiles sont chargées depuis le service Aladin. Il existe également une version en ligne, facilement intégrable sur un site web : Aladin-Lite.



Figure 5 : logo d'Aladin

2.3 – L'international Virtual Observatory Alliance

L'International Virtual Observatory Alliance est une organisation internationale regroupant les projets d'Observatoire Virtuel de plus de 20 pays (France, Allemagne, Chine, États-Unis, Brésil, Australie, . . .). Le but de l'IVOA est d'établir, au niveau mondial, des standards ainsi que de définir les stratégies à adopter. Cela permet de faciliter les échanges internationaux au même titre que la collaboration entre les différents pays.

Le CDS participe activement à ce projet par ces trois services cités auparavant, mais aussi par son implication au sein des groupes de travail de l'IVOA. En effet, plusieurs de ces groupes de travail qui composent l'IVOA sont présidés ou vice-présidés par des membres du CDS.

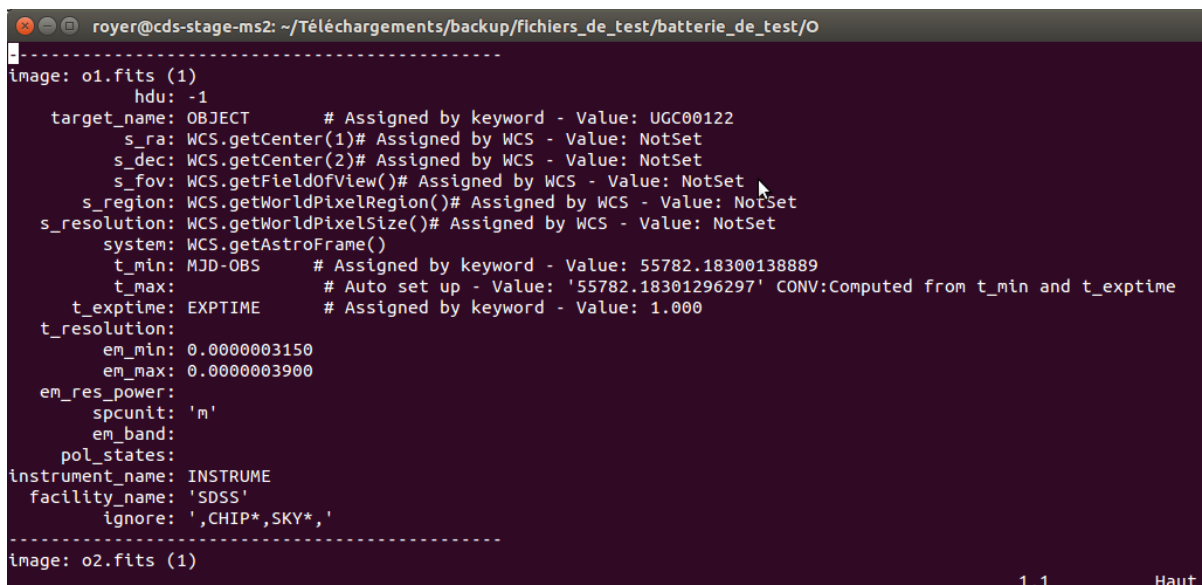


Figure 6 : logo de l'IVOA

3 – Présentation du sujet

Dans le cadre de son objectif de diffusion de données astronomiques, le service VizieR voit de plus en plus d'images spatiales et de spectres s'ajouter à ces catalogues. Pour les rendre visible parmi la masse de documents préservé au CDS, les ressources sont indexées et documentées selon des standards de la communauté (IVOA). L'utilisation de méta-données standard, ainsi que l'implémentation des services d'accès permettent d'augmenter la visibilité des ressources dans une base de données de l'Observatoire virtuel. Ils deviennent ainsi a porté des outils développés dans la discipline. Parmi les méta-données se trouvent la longueur d'onde observée, le télescope utilisé, la région ciblée, etc. Les images et spectres sont sous forme de fichier FITS (une extension spécifique a l'astronomie) et les métadonnées les accompagnant sont stockées dans un fichier texte de mapping au format spécial, qui contient les métadonnées ainsi que les fichiers FITS qu'elles concernent.

Les métadonnées sont sous la forme "clé-valeur" pour chacun des attributs qui décrivent l'image ou le spectre. Jusqu'à présent ces fichiers de mappings étaient créés en lignes de commandes, par les documentalistes du CDS, grâce a la commande "vi".



```
royer@cds-stage-ms2: ~/Téléchargements/backup/fichiers_de_test/batterie_de_test/O
-----
Image: o1.fits (1)
  hdu: -1
  target_name: OBJECT      # Assigned by keyword - Value: UGC00122
    s_ra: WCS.getCenter(1)# Assigned by WCS - Value: NotSet
    s_dec: WCS.getCenter(2)# Assigned by WCS - Value: NotSet
    s_fov: WCS.getFieldOfView()# Assigned by WCS - Value: NotSet
    s_region: WCS.getWorldPixelRegion()# Assigned by WCS - Value: NotSet
  s_resolution: WCS.getWorldPixelSize()# Assigned by WCS - Value: NotSet
  system: WCS.getAstroFrame()
    t_min: MJD-OBS        # Assigned by keyword - Value: 55782.18300138889
    t_max:                # Auto set up - Value: '55782.18301296297' CONV:Computed from t_min and t_exptime
  t_exptime: EXPTIME      # Assigned by keyword - Value: 1.000
  t_resolution:
    em_min: 0.0000003150
    em_max: 0.0000003900
  em_res_power:
  spcunit: 'm'
  em_band:
  pol_states:
  instrument_name: INSTRUME
  facility_name: 'SDSS'
  ignore: ',CHIP*,SKY*,'
-----
Image: o2.fits (1)
1,1 Haut
```

Figure 7 : fichier de mapping ouvert dans un terminal grâce à la commande "vi"

L'objectif du stage était de créer une interface graphique en Java pour proposer une alternative aux documentalistes peu à l'aise avec les lignes de commandes.

En terme d'existant, le CDS possède déjà une API Java spécialisée dans la gestion des mappings et des fichiers FITS, nommée Saada. Elle permet également de très nombreuses manipulations pour le service Vizier. En l'occurrence, Saada est utilisée avec l'interface créée durant le stage, pour permettre de récupérer des données provenant les collections FITS sur lesquels on souhaite appliquer le mapping ainsi que l'ouverture et la sauvegarde des fichiers mappings.

Une collection de fichiers FITS peut être soit un fichier FITS, soit un fichier texte contenant une liste de fichiers FITS, soit un répertoire contenant des fichiers FITS (dans le cas d'un fichier liste ou d'un répertoire, par défaut, le fichier de mapping créé affectera tous les fichiers FITS dans la collection).

Les données récupérables sont les valeurs de certains attributs déjà contenues dans le fichier FITS, ainsi qu'un certain nombre de couples "clé-valeur" contenues dans l'entête des fichiers FITS. Ces couples fournissent une banque dans laquelle on pourra piocher pour affecter une valeur à un attribut, grâce au mot clé contenu dans le couple. Ce mot-clé sera ensuite reconnu lors de la sauvegarde du mapping par Saada, qui l'utilisera pour déterminer la valeur de l'attribut en question pour ce mapping. Les données sont renvoyées sous forme de JSON au programme par l'API.

L'avantage de cette API fournie est que la plupart des travaux de manipulation existent déjà, et que le développement peut donc se concentrer sur l'interface graphique. Mais l'API est en elle-même complexe parce que complète et peut être initialement dure à maîtriser en raison de son utilisation de nombreuses bibliothèques Java. Qui plus est, le programme final est destiné à des documentalistes, donc des utilisateurs n'ayant pas nécessairement des connaissances en informatique et qui ont déjà des réflexes et des habitudes de



Figure 8 : logo de Saada

travail acquises avec l'ingestion par ligne de commande. L'interface graphique finale se doit donc d'être accessible, complète et pas trop perturbante pour les personnes ayant l'habitude de travailler avec une autre méthode.

En utilisant Saada, le but était de créer une interface conviviale qui permette de voir en temps réel les modifications apportées au mapping pendant la création afin de faciliter le travail des documentalistes pour l'ingestion de données dans la base de VizierR.

Pour ce faire, la bibliothèque Swing du langage Java avait d'abord été pressentie. Mais après l'avoir comparée aux autres possibilités disponibles en Java, il a été décidé de se tourner vers la bibliothèque JavaFX, successeur de Swing.

4 – Cahier des charges

Le format FITS (Flexible Image Transport System) est un format de fichier propre à l'astronomie depuis 1981. Le format permet d'encapsuler des images, des spectres, des séries temporelles ou des tables en accompagnant les données par des méta-données. Parmi ces méta-données, l'instrument, l'objet visé, des dates et des positions sont inclus. Les données sont stockés généralement en binaire et les méta-données sont exposés par des couples mots-clés – valeurs en entête des données. La manière de sérialisé les données ainsi que l'ensemble des mots clés FITS n'est pas standardisé malgré des notes de « bon usage »: celles -ci ne sont pas toujours pris en compte

La documentation et l'indexation des données associées (spectres , images) est sérialisé dans des fichiers ASCII. Ces fichiers contiennent les méta-données qui établissent le mapping entre les entêtes des fichiers FITS et le model ObsCore de l'IVOA. Ce model est un standard définit par l'IVOA , qui a été discutés par les centres de données pour définir les méta-données de bases pour la description d'une observation.

Le mapping est généré soit par les auteurs qui utilisent un service web de

téléchargement de données ou bien par les documentalistes de VizieR.

Les fichiers FITS et les mappings sont ingérés dans la base de données VizieR. Actuellement, les fichiers de mappings sont générés et ingérés grâce à un certain nombre d'outils dont dispose le CDS : différentes API et des applications en mode texte exécutés dans une console. Une GUI destinée au CDS et en particulier aux documentalistes du CDS permettrait une meilleure prise en main de ce mapping.

L'application a accès aux données nécessaires à son fonctionnement grâce aux outils fournis par le CDS :

- les données provenant des entêtes FITS sont récupérées par l'API du CDS;
- la liste des bandes spectrales et les couples télescope / instrument / bande spectrale est accessible par requête aux bases de données du CDS;
- l'API Saada est utilisée pour tester le mapping et récupérer les éventuelles erreurs dans sa conception;
- L'API VizieR, qui est une surcouche à l'API Saada sert pour la sérialisation des mappings.

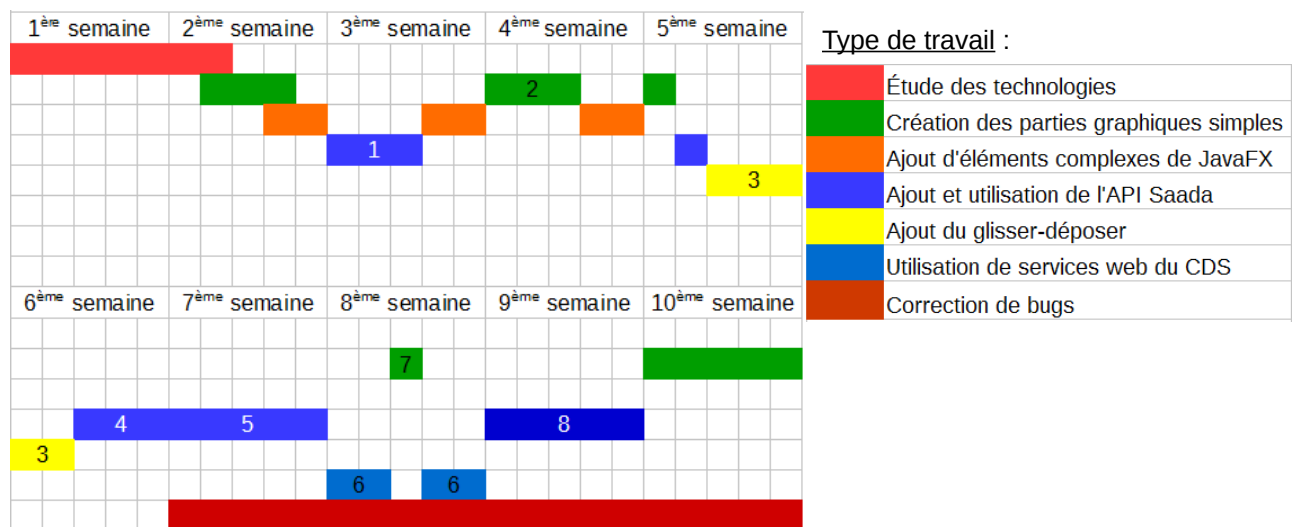
D'un point de vue fonctionnel, l'application permet :

- de rechercher les ressources présentes dans un répertoire contenant des fichiers FITS, dans un fichier liste contenant des adresses de fichiers FITS, dans un fichier FITS;
- de proposer un mapping pour toutes les collections citées au dessus;
- de charger et d'éditer un mapping déjà existant ;
- de sauvegarder un mapping;
- d'afficher les entêtes FITS pour les documents mappés, permettant d'ajouter ces mots-clés dans les valeurs du mapping;
- de proposer une prévisualisation du mapping généré.

Techniquement, le développement doit être en Java, en revanche la bibliothèque graphique utilisée est libre, du moment qu'elle est adaptée au type d'interface demandée.

5 – Mise en œuvre

Au cours du stage, j'ai eu la possibilité d'organiser mon temps de travail à ma guise. Mon tuteur et moi faisons le point quotidiennement sur ce qui avait été fait et ce sur quoi je devais me concentrer pour la suite. Ainsi, en dehors de l'ordre des fonctionnalités à développer j'ai été libre de mon emploi du temps. Ci dessous voici mon emploi du temps global jour par jour durant mon stage :



- Fonctionnalités principales implémentées :**
- 1 : création et édition de mappings depuis des collections FITS
 - 2 : gestionnaire des en-têtes FITS
 - 3 : glisser-déposer
 - 4 : ouverture des mappings existants
 - 5 : sauvegarde des mappings
 - 6 : utilisation des bases de données du CDS par service web
 - 7 : prévisualisation
 - 8 : tests sur les mappings

Figure 9 : représentation de l'emploi du temps des 10 semaines de stage

5.1 – Études des bibliothèques Jgoodies et JavaFX

5.1.1 – Contexte de la demande

Initialement, le développement était censé se faire en Java avec la bibliothèque Swing pour la partie graphique. Mais à mon arrivé, mon tuteur m'a proposé d'étudier les différentes possibilités pour réaliser l'interface. D'autres membre du CDS qui avaient déjà travaillé sur des interfaces graphiques Java proposaient deux bibliothèques à étudier : JavaFX, qui est le successeur de Swing, une bibliothèque proposée par Oracle qui a été créée en 2008 ; et Jgoodies qui est un supplément à Swing pour mieux gérer la mise en forme des différents éléments graphiques dans la fenêtre. La raison de cette étude venait de la faiblesse de Swing sur ce dernier point.

J'ai donc étudié et comparé les deux technologies, voici mes conclusions.

5.1.2 – Comparaison des bibliothèques

- Jgoodies utilise le même principe que Swing à ceci près que l'agencement de la fenêtre est défini par rapport à des axes verticaux et horizontaux qui régissent les bords des éléments. Ce qui est pratique puisque ainsi les éléments conservent leurs positions relatives lors du redimensionnement de la fenêtre. En revanche, cela oblige la fenêtre a avoir sa configuration définitive dès le début du développement, au risque de devoir tout reprendre si l'on veut ajouter un élément.
- JavaFX, quant à lui, utilise des fichiers spécifiques dans un format spécial XML (le FXML) pour gérer les fenêtres., Ceci réduit considérablement le code à écrire puisque tous les éléments graphiques avec lesquels l'utilisateur n'interagit pas ne sont pas définis dans le code. De plus, l'agencement de la fenêtre n'as pas à être défini dans ce dernier puisqu'il l'est déjà dans la structure même du FXML.

Pour créer ces fichiers FXML il existe deux possibilités :

- soit le code Java peut être utilisé pour définir les éléments, ce code sera ensuite interprété pour générer le fichier FXML. Cette méthode ne présente que peu d'intérêt par rapport à Swing puisque le temps de codage et la masse du code sont très proches ;
- on peut aussi utiliser un logiciel spécialement conçu pour ça : SceneBuilder. Ce logiciel permet de créer des interfaces graphique de manière visuelle. Pour ce faire, il suffit de glisser et déposer sur le centre de la fenêtre les différents éléments graphiques souhaités.

Si on utilise la seconde option, le temps de développement s'en trouve fortement réduit pour un meilleur résultat.

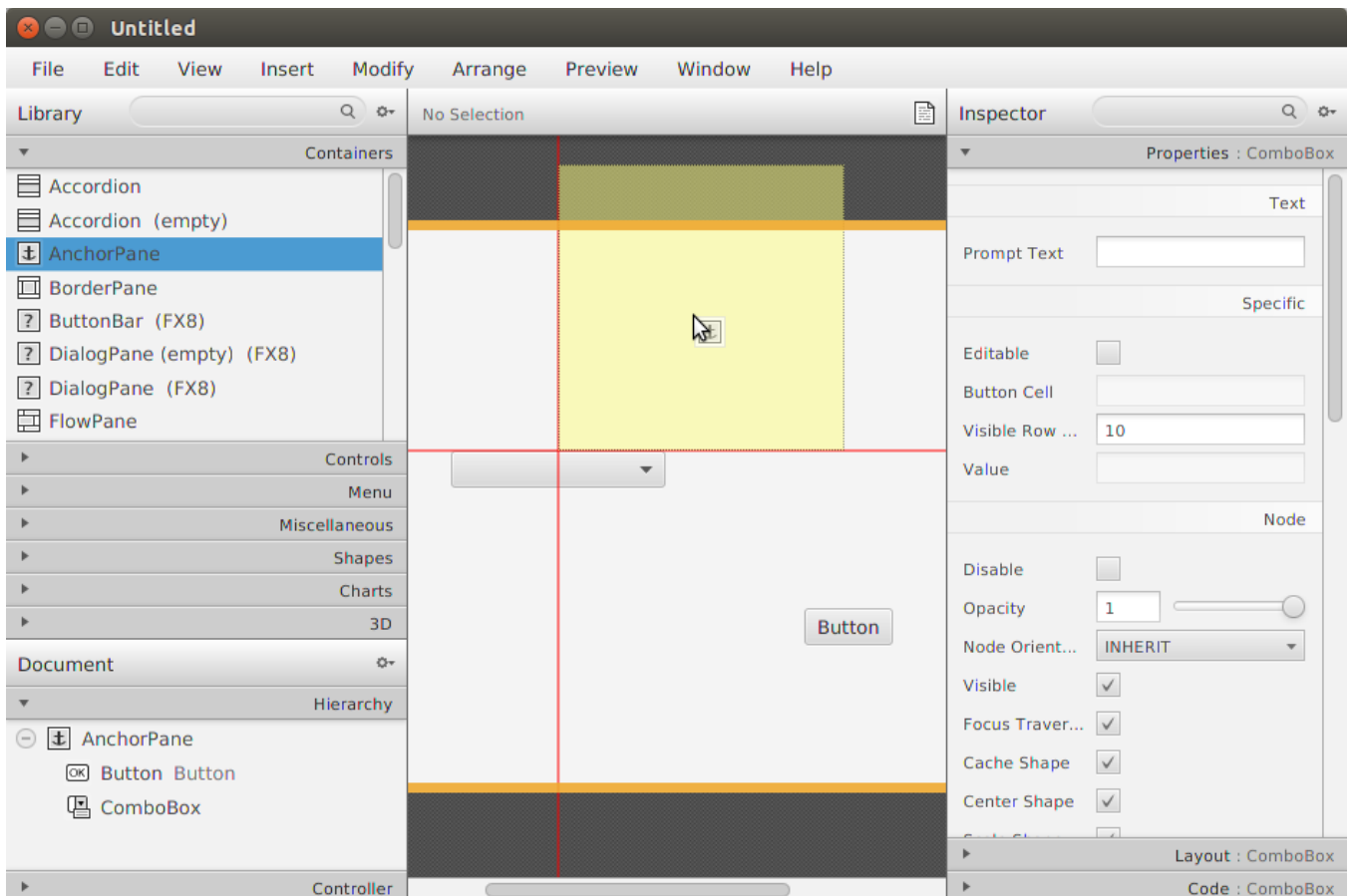


Figure 10 : interface du logiciel SceneBuilder

5.1.3 – Choix de la bibliothèque

Malheureusement, le développeur indépendant à l'origine de Jgoodies a décidé de rendre la prochaine version de la bibliothèque payante. Ce qui est rédhibitoire pour le CDS qui ne peut se permettre de faire évoluer onéreusement son interface dans le temps. C'est donc la bibliothèque JavaFX, plus simple, plus efficace et gratuite qui a été retenue. J'ai donc utilisé SceneBuilder pour les parties graphique et Eclipse pour le code, avec l'ajout de e(fx)clipse, qui permet à l'éditeur de reconnaître le code JavaFX.

5.2 – Utilisation de l'API Saada

Le CDS utilise la librairie fournie par le logiciel Saada. Saada est un générateur de bases de données pour l'astronomie qui est développé à l'Observatoire de Strasbourg dans l'équipe des hautes Energies (et donc pas au CDS). Elle est utilisée par le programme créé lors du stage.

5.2.1 – Récupération des données à partir des fichiers FITS

D'abord, l'API Saada permet de récupérer les données contenues dans les fichiers FITS lors de la création du mapping pour les dits fichiers. L'API Saada dispose d'une base de mots clés FITS qui permet d'établir le mapping. Il s'agit de mots-clés définis dans la note FITS : ce sont des mots clés qui sont proposés mais nullement standardisés. Parmi ces mots clés la position des objets est interprétée pour les projections les plus standards. Lors de la création du mapping, l'utilisateur sélectionne une collection FITS (fichier FITS, répertoire ou fichier liste) et l'API propose un mapping suivant les méta-données incluses dans les fichiers FITS. Cela permet outre de gagner du temps sur les données à rentrer dans l'application. L'application permet aussi dans le cas où un même mapping est utilisé par une collection de fichiers, de tester la cohérence du mapping. Il se peut que les fichiers soient très différents. Dans ce cas, les données récupérées depuis les fichiers FITS seront également très différentes et ne pourront être ajoutées dans le mapping,

puisqu'elles ne correspondront pas à tous les fichiers sources. L'application avertira alors l'utilisateur et précisera quelles données n'ont pas pu être ajoutées, en pointant leurs différences. L'utilisateur pourra ainsi constater la différence entre les fichiers FITS et donc adapter son mapping en fonction.

5.2.2 – Récupération des en-têtes FITS

Ensuite, l'API Saada sert à la récupération du contenu des en-têtes FITS. Ces en-têtes contiennent un certain nombre de couples « clé-valeur ». Ces couples dépendent du fichier en question et sont potentiellement des valeurs qui correspondent à l'un des attributs du mapping pour ce fichier. Ces couples sont donc récupérés grâce à l'API et stockés dans une banque où le documentaliste pourra venir piocher pour affecter une valeur à un attribut. Cela permet également d'accélérer et de faciliter le travail des documentalistes.

5.2.3 – Vérification de la validité des mappings

Enfin, l'API est utilisé pour tester la validité des mappings. Avant de sauvegarder le mapping, l'utilisateur peut vérifier qu'il est correcte. Pour cela, l'application envoie les données rentrées dans l'interface à l'API qui vérifie deux points : la cohérence des données et la présence des données requises. En effet, plusieurs attributs sont définis les uns par rapports aux autres, Si leurs valeurs ne sont pas cohérentes (par exemple, si les limites de la bande spectrale observée ne correspond pas à la dénomination de la bande), l'API signale l'erreur à l'utilisateur. De même, certaines données n'ont de sens que si d'autres sont présentes pour les utiliser. Dans ce cas (par exemple, si la zone ciblée par l'observation est renseignée mais le système de calcul de cette zone ne l'est pas) l'API signale une erreur. Si l'un ou l'autre des types d'erreur est signalée, la sauvegarde du mapping est impossible. Cette vérification se fait de deux manières : soit l'utilisateur le demande en cliquant sur un bouton, soit automatiquement lors de la sauvegarde du mapping.

5.3 – Accès aux bases de données du CDS

En plus de l'API Saada, le CDS fournit des bases de données contenant des références qui sont utilisées dans l'application.

Premièrement, il est possible pour le documentaliste de renseigner automatiquement la bande spectrale observée. Directement accessible sur l'interface, une liste déroulante propose toutes les bandes référencées par le CDS (et donc par l'IVOA). Lorsque l'une d'elles est sélectionnée, les limites de la bande (longueur d'onde minimale et maximales) sont renseignées également. Il a été préféré de remplir cette liste déroulante en récupérant les bandes depuis la base de données plutôt que de les avoir directement dans le programme. La raison en est simple, comme le CDS respecte les standards internationaux de l'IVOA, si l'une des valeurs change ou si une nouvelle valeur apparaît à l'échelle mondiale, il est bien plus simple de ne modifier la liste des bandes qu'une seule fois. Ainsi, la liste des bandes reste en permanence à jour. Pour faire cette requête, l'application utilise un protocole HTTP, grâce à une URL.

Deuxièmement, l'ajout des valeurs pour le télescope ou l'instrument utilisé lors de l'observation peut s'effectuer grâce à une fenêtre à part. Ces requêtes sont faites sur Vizier et le résultat sont tous les noms qui existent déjà dans la base. Lorsque l'utilisateur clique sur un bouton, une fenêtre s'ouvre, dans laquelle figure la liste des différents télescopes connus du CDS. Une fois un télescope sélectionné, l'application dresse la liste des instruments du télescope ainsi que la bande spectrale et les limites de cette bande pour chacun des instruments sont proposés. Une fois l'instrument souhaité du télescope sélectionné est choisi, l'utilisateur peut cliquer sur un bouton et renseigner le nom du télescope, le nom de l'instrument ainsi que les informations de la bande spectrale. Ainsi, 5 valeurs pourront être renseignées en une seule action, ce qui représente un gain de temps dans la création du mapping. Pour avoir les données concernant les télescopes et leurs instruments, l'application fait deux requêtes à la base de Vizier :

- une première requête permet d'obtenir la liste des télescopes référencés sur Vizier au moment où le documentaliste effectue sa

demande et appuie sur le bouton de l'interface principale qui ouvre la première fenêtre ;

- une seconde requête récupère la liste des instruments, ainsi que les informations de bandes spectrales associées, lorsque que l'utilisateur sélectionne l'un des télescopes.

Ce processus permet encore un fois de garder les listes à jour, mais aussi de limiter la mémoire utilisée par ces données, ne limitant l'utilisation qu'au minimum requis.

5.4 – Gestion des mappings

5.4.1 – Ouverture des mappings

D'abord, l'utilisateur peut ouvrir des mappings déjà existants qui sont dans des fichiers dans un format texte spécial. Ces mappings sont issus soit d'une sauvegarde d'un travail fait par un documentaliste, soit issu du mapping fourni par les auteurs (astronome) qui ont versé leur données via le service web de soumission de Vizier.

Ces fichiers contiennent les données qui peuvent être renseignées selon trois méthodes différentes qui permettent d'avoir les valeurs pour chaque fichiers FITS :

- la méthode par valeur : dans ce cas, la valeur de l'attribut en question est la valeur rentrée. Elle s'applique alors à tous les fichiers FITS de la collection sur laquelle s'applique le mapping. Elle est entourée par des simple quotes ;
- la méthode par mot-clé : dans ce cas, la valeur de l'attribut dépend du fichier FITS. En effet, le mot-clé sera interprété lors de l'utilisation du mapping. Donc, pour un même attribut, deux fichiers FITS peuvent avoir deux valeurs différentes avec un même mot-clé en raison des différences entre les fichiers FITS qui conduisent à des résultats différents lors de l'interprétation ;
- la méthode WCS (World Coordinate Systems) : WCS est un standard qui défini des mots-clés et usages spécifique aux fichier FITS pour les descriptions astronomiques. WCS (World Coordinate System) permet de préciser les coordonnées spatiale ou spectrale en prenant en compte différentes projections sur la sphère. Dans le mapping, un attribut avec cette

méthode sera un appel d'une fonction de WCS permettant d'obtenir une valeur pour cet attribut en fonction du fichier FITS concerné.

Lors de l'ouverture d'un mapping existant, la méthode avec laquelle l'attribut est renseigné doit être prise en compte. Si l'attribut est renseigné par valeur, dans ce cas c'est cette valeur qui est affichée dans l'interface. Si la méthode est « par mot-clé », dans ce cas, le mot-clé sera affiché, et la valeur venant de l'interprétation du mot-clé apparaîtra à titre informatif. Enfin, pour la méthode WCS, l'interface affichera « WCS » pour l'attribut et le résultat de l'appel à la fonction de WCS.

L'interface pourra également afficher laquelle des trois méthodes a été utilisée pour renseigner chacun des attributs

5.4.2 – Édition des mappings

Une fois le mapping ouvert, l'utilisateur peut l'éditer, soit en remplissant lui-même la valeur en toute lettre, soit en utilisant un des mots-clé provenant de l'entête FITS. Dans le premier cas, la méthode de l'attribut deviendra « par valeur » et dans le second elle deviendra « par mot-clé ».

- Une fois le mapping ouvert, le documentaliste peut éditer chacun des attributs indépendamment à la main. Il a plusieurs possibilités pour le faire :
 - il peut entrer une valeur directement dans le formulaire. Dans ce cas, la méthode sera « par valeur » et l'application l'affichera directement ;
 - il peut aussi entrer un mot-clé. Si la syntaxe des mot-clé est respectée (toutes les lettres en majuscules accompagnées uniquement de « _ » et de « - »), l'interface affichera le mot-clé. Elle comparera ensuite le mot clé avec la liste de mots-clés qu'elle possède. Si le mot-clé en fait partie, elle modifiera la méthode de l'attribut pour « par mot-clé » ;
 - le documentaliste peut également entrer une fonction de WCS. Cette fonction a la forme : « WCS.nomDeLaFonction() ». Si l'interface reconnaît le mot « WCS » dans l'expression de l'attribut, elle changera la méthode pour « par WCS » et affichera « WCS ».

- Pour gagner du temps et simplifier l'édition, l'utilisateur a un autre moyen de renseigner une valeur pour un attribut. En effet, nous avons vu que l'application peut récupérer le contenu des en-têtes FITS pour former une banque où l'utilisateur pourra venir piocher des couples clés-valeur contenant des mots-clés déjà fournis. Une fois ces couples récupérés, l'application les liste dans un tableau. Le documentaliste peut naviguer verticalement dans le tableau ou taper le mot-clé dans la barre de recherche, ce qui supprimera du tableau tous les couples ne contenant pas le caractère rentré dans la barre de recherche. Une fois le mot-clé souhaité trouvé, l'utilisateur aura juste à glisser la ligne du tableau et à la déposer sur le champ de l'attribut auquel il souhaite affecter cette valeur. Une fois ce glisser-déposer effectué, le champ prendra pour valeur le mot-clé et la méthode de l'attribut en question deviendra « par mot-clé ».

Enfin, en plus de ces deux manières de procéder, comme nous l'avons vu précédemment, l'utilisateur a d'autres moyens pour accélérer l'édition du mapping.

D'abord l'édition des bandes spectrales a été améliorée. En effet, pour rentrer les valeurs de la bande spectrale observée, l'utilisateur peut soit les rentrer lui même, soit choisir l'une des bandes de la liste déroulante sur l'interface principale. Auquel cas les limites de la bande seront également renseignées automatiquement. Si ensuite le documentaliste modifie une de ces limites, la bande sera désélectionnée de la liste déroulante.

Enfin, grâce à la fenêtre de gestion de télescope, l'utilisateur peut facilement et rapidement entrer le nom du télescope, le nom de l'instrument et les limites de la bande automatiquement. Il pourra également naviguer dans les listes des télescopes ou des instruments avec la molette de la souris, ou chercher le nom qu'il veut grâce aux deux barres de recherches pour les deux listes, avec le même principe que pour les barres de recherches de la liste de mot-clé des entêtes FITS.

5.4.3 – Prévisualisation

A tous moments au cours du travail sur le mapping, l'utilisateur peut prévisualiser le résultat dans une fenêtre prévue à cette effet contenant un tableau avec une colonne pour chacun des attributs du mapping. Il s'agit ici d'une preview, préfigurant les meta-données tels qu'ils seront disponible après l'ingestion. Pour ce faire, il lui suffira de cliquer sur le bouton « Preview » de la fenêtre principale. L'interface enverra alors les valeurs entrées par l'utilisateur dans l'API Saada qui générera les résultats pour les fichiers FITS concernés. Pour chacun des fichiers FITS, le tableau de la fenêtre de prévisualisation générera une ligne qui aura les valeurs de chaque attributs en fonction de son expression dans le mapping et du fichiers FITS en question (pour les mots-clés et les fonctions WCS, le résultat sera affiché et non l'expression).

5.4.4 – Sauvegarde

Une fois l'édition terminée, l'utilisateur peut créer le fichier de mapping. Il sera créé automatiquement par l'application pour peu que les vérifications de validité de mappings aient été faites. Lorsque que l'utilisateur clique sur le menu « Save » ou utilise le raccourci « Ctrl + s », une fenêtre s'ouvre dans laquelle il pourra choisir le nom et l'emplacement où il veut sauvegarder le mapping. Les expressions des attributs du mappings sont envoyées dans l'API avec les méthodes auxquelles elles ont été renseignées.

Selon la méthode, l'API Saada vas interpréter différemment l'expression, comme expliqué plus haut.

Voila les principales fonctionnalités de l'application développée. Afin de faciliter et d'accélérer le travail des documentalistes du CDS, beaucoup de fonctionnalités ont été ajoutées dans ce but. Et chacune d'entre elle a un raccourci clavier pour encore accélérer le processus. En plus de cela l'accent a été mis sur l'aspect visuel de l'interface, dans le sens où elle permet de visualiser rapidement et facilement le mapping dans sa globalité.

Voici sur la figure 12 l'interface finale implémentant toutes ces fonctionnalités.

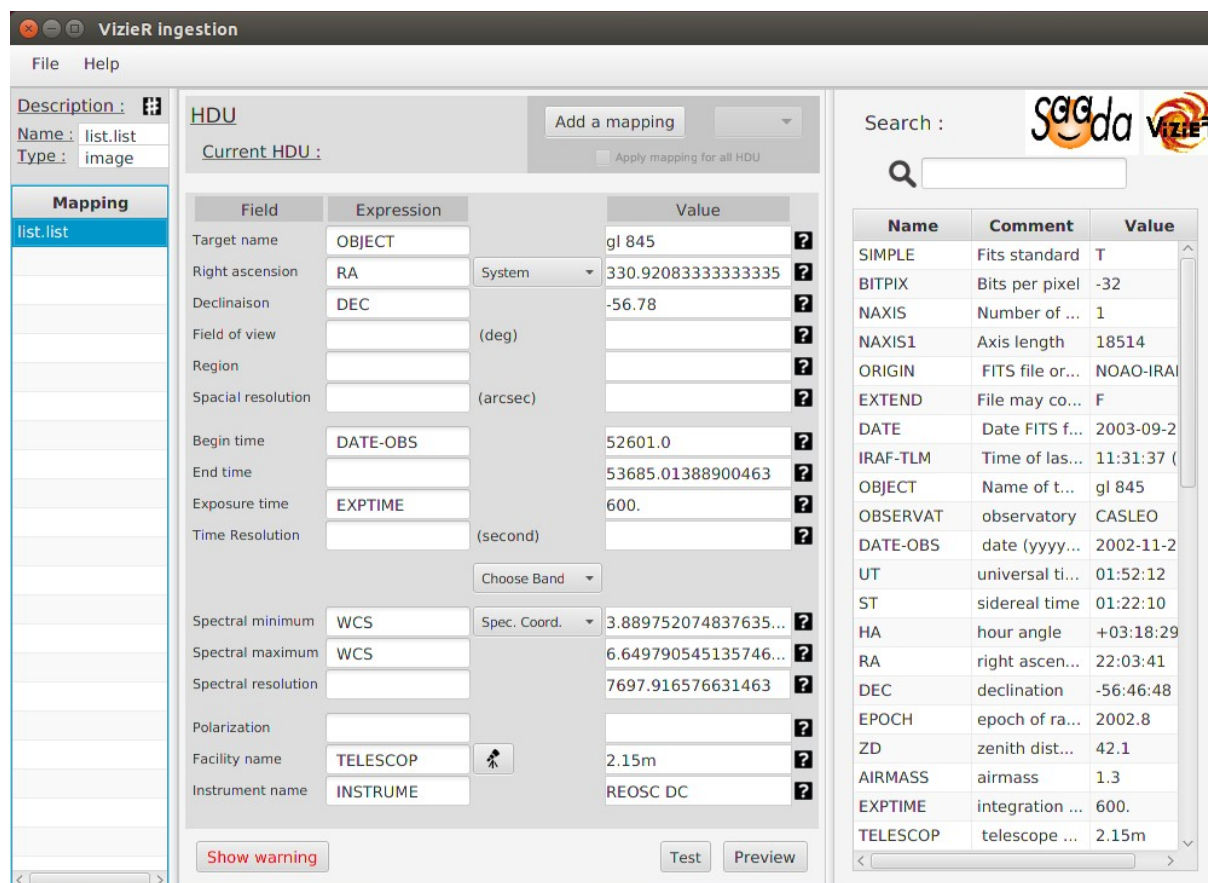


Figure 11 : interface finale de l'application

5.5 – Difficultés rencontrées

Le développement de l'application n'était pas spécialement compliqué et si certains points m'ont posés quelques difficultés, je ne suis pas resté bloqué très longtemps sur un problème. La plupart du temps la difficulté venait de deux facteurs : ma méconnaissance de l'API Saada et la pauvreté des tutoriaux en ligne pour JavaFX.

5.5.1 – Problèmes avec l'API Saada

Concernant l'API Saada, le majeur problème est sa taille et sa complexité. En effet, cette API regroupe près de 70 classes qui interagissent entre elles. Bien souvent lorsqu'il y a une erreur avec l'utilisation de l'API, il est très difficile de savoir d'où vient le problème sans parcourir tout le code, ce qui est extrêmement long.

On est donc bien souvent limité à ne pouvoir lire que quelques classes pour trouver notre erreur. Cette API ne disposant pas de documentation, j'ai été bien souvent obligé de demander de l'aide à mon tuteur pour sa compréhension.

5.5.2 – Tutoriaux JavaFX

Quant aux tutoriaux JavaFX, le problème vient du fait que cette bibliothèque est encore relativement récente. Par conséquent, il n'existe pas encore de solide communauté supportant l'application. La majorité des développeur continuent à travailler avec Swing. En plus du faible nombre de tutoriaux, la méthode employée par leurs créateurs pose aussi problème. En effet, des deux méthode pour développer en JavaFX, celle qui est proche de Swing est la plus souvent utilisée et donc celle qui est le plus souvent employée pour faire les tutoriaux. Par exemples, les tutoriaux officiels d'Oracle pour JavaFX sont intégralement fait suivant cette méthode. Cela ma posé un problème en particulier pour l'implémentation du glisser-déposer. En effet, si l'utilisation des deux méthodes pour deux fonctionnalités différentes est possible au sein d'un même programme, par souci de cohérence et afin de faciliter la maintenance de l'application par les informaticiens du CDS, j'ai préféré créer cette fonctionnalité en utilisant SceneBuilder, comme je l'avais fait pour toute les autres. En fait, la traduction des tutoriaux d'une méthode à l'autre a été long et délicat. Le glisser-déposer est la partie la plus longue et difficile que j'ai eu a faire pour cette interface.

6 – Bilan

L'interface développée au cours du stage possède dès à présent les fonctionnalités nécessaires à son utilisation et elle sera transmise prochainement aux documentalistes qui l'utiliseront d'ici cet été. L'application tel qu'elle est actuellement correspond aux attentes de l'entreprise en terme de fonctionnalité et d'ergonomie. Mais bien sûr, de nombreuses améliorations sont possibles, notamment pour automatiser complètement la génération des fichiers de mapping et leur envoi dans le base VizieR.

Au CDS, les trois corps de métiers astronomes, documentalistes et informaticiens travaillent ensembles, et j'ai eu l'opportunité, au cours de mon stage, de côtoyer les trois.

J'aborderai d'abord mes relations avec les informaticiens, pour des raisons évidentes étant donné que je travaille avec eux. De plus, les informaticiens organisent régulièrement des réunions dites "infusion". Durant ces réunions, un membre de l'équipe présente aux autres une technologie ou une technique inconnue ou non maîtrisée par le reste des informaticiens. Ces réunions, outre leur côté instructif, étaient l'occasion d'échanger avec ceux avec qui nous n'avions pas l'occasion de travailler. Qui plus est, mon poste de travail était dans la bibliothèque du CDS avec trois autres stagiaires. Nous avons pu nous entraider sur nos projets respectifs à plusieurs reprises.

Ensuite, compte tenu de mon sujet de stage, mon travail était destiné aux documentalistes, c'est à dire à des personnes qui ne travaillent pas dans l'informatique. Donc, lors des présentations de mon travail, les recommandations n'étaient pas faites en termes informatiques et je devais les interpréter en termes de fonctionnalités informatiques à implémenter.

Enfin, tous les vendredis, l'observatoire organise un séminaire sur l'astronomie en anglais, qui nous permettait de voir comment les astronomes travaillent notamment grâce aux outils développés par les informaticiens.

Au cours de ce stage, j'ai eu l'occasion de travailler sur de nombreux problèmes, dont certains que je ne pensais pas aborder. Si le centre du travail reposait sur une interface homme-machine, j'ai pu également mettre à profit mes connaissances de langage objet. Parmi les points auxquels je ne m'attendais pas, j'ai beaucoup travaillé avec le format JSON pour la réception des données de l'API ou des bases de données. L'implémentation de l'API dans l'interface m'a aussi fait utiliser une API que je ne connaissais pas, ce qui est une expérience particulière puisqu'il faut pouvoir résoudre ses problèmes en ne se basant que sur la partie connue du code. En termes d'acquis au cours du stage, la plus grosse part est représentée par JavaFX que je ne connaissais pas du tout au départ et que j'arrive aujourd'hui à maîtriser convenablement.

7 – Conclusion

Ce stage de 10 semaines a été ma première expérience dans le monde du travail et je l'ai grandement appréciée. Outre les nouvelles technologies que j'ai apprises à maîtriser, j'ai eu l'occasion de travailler avec des informaticiens expérimentés dans un service qui est une référence mondiale dans son domaine. J'ai également dû collaborer avec d'autres corps de métier afin d'adapter au mieux l'interface à leur demande. L'application reste perfectible, mais les fonctionnalités demandées dans le cadre du stage sont présentes et l'interface peut être servir pour une utilisation basique de création de fichiers de mapping.

Intéressant et très formateur, ce stage conforte complètement mon choix de carrière et a été une excellente conclusion à mes deux années de formation à l'IUT. En effet, il m'a permis de mettre en pratique les connaissances que j'ai acquises durant ces deux ans, dans un cadre qui me passionne.

Au cours des différents projets auxquels j'ai participé pendant ma formation, j'ai souvent réalisé l'interface graphique des applications et logiciels développés, Je l'ai fait parfois plus par nécessité que par choix, mais j'apprécie désormais le travail des interfaces en Java et ce stage conforte cette idée. Je pense que travailler dans ce domaine me conviendrait parfaitement et ces 10 semaines ont été un excellent préambule en la matière. Étant donnée la masse de travail qu'il reste pour terminer l'application, le CDS m'a offert un CDD de un mois pour compléter l'interface au-delà des fonctionnalités demandées dans le cadre du stage.

Table des illustrations

Figure 1 : bâtiment de la coupole	page 8
Figure 2 : logo du CDS	page 10
Figure 3 : logo de Simbad	page 10
Figure 4 : logo de VizieR	page 11
Figure 5 : logo d'Aladin	page 12
Figure 6 : logo d'IVOA	page 12
Figure 7 : fichier de mapping ouvert dans un terminal grâce à la commande « vi »	page 13
Figure 8 : logo de Saada	page 14
Figure 9 : représentation de l'emploi du temps des 10 semaines de stage	page 17
Figure 10 : interface du logiciel ScenBuilder	page 19
Figure 11 : interface finale de l'application	page 27

